

Aprende a crear sitios Web con HTML5 y CSS3

¿Os interesa hacer sitios web? En este curso aprenderemos a usar HTML5 y CSS3, los dos lenguajes de programación en los que están basados todos los sitios web.

Es posible que penséis que no estáis hechos para aprender el lenguaje informático, pero no os preocupéis: El HTML y el CSS son lenguajes simples. Las últimas versiones de estos lenguajes, HTML5 y CSS3, nos ofrecen actualmente posibilidades increíbles. Con ellos podréis añadir texto a un sitio web, construir un menú de navegación, añadir imágenes, efectos gráficos y mucho más.

• Parte 1 - Fundamentos del HTML5

1. [1. ¿Cómo se construyen los sitios web?](#)
2. [2. Vuestra primera página web en HTML](#)
3. [3. Organizar el texto](#)
4. [4. Crear enlaces](#)
5. [5. Imágenes](#)
- [_Test: Quiz 1](#)
- Actividad: Crea tu propio CV en HTML

• Parte 2 - Los placeres de dar formato con CSS

0. [1. Configurar CSS](#)
1. [2. Dar formato al texto](#)
2. [3. Color y fondo](#)
3. [4. Bordes y sombreado](#)
4. [5. Crear apariencias dinámicas](#)
- [_Test: Quiz 2](#)
- Actividad: Añade estilos a tu CV

• Parte 3 - Diseño de la página web

0. [1. Estructurar tu página](#)
1. [2. El modelo de cajas](#)
2. [3. Posicionamiento en CSS](#)
3. [4. Ejercicio práctico: creación de una página web paso por paso](#)
- [_Test: Quiz 3](#)
- Actividad: CV horizontal

• Parte 4 - Características avanzadas

- 0. [1. Tablas](#)
- 1. [2. Formularios](#)
- 2. [3. Vídeo y audio](#)
- 3. [4. Diseño de página adaptable con consultas multimedia](#)
- 4. [5. Hay que seguir](#)
 - [_Test: Quiz 4](#)
 -
- **Parte 5 - Anexos**

- 0. [1. Envío de tu sitio web en internet](#)
- 1. [2. Recuerdo de las etiquetas HTML](#)
- 2. [3. Recuerdo de las propiedades CSS](#)
- 3. [4. Manejar la compatibilidad entre navegadores](#)

¿Cómo se construyen los sitios web?

¡Bueno, aquí tenemos el primer capítulo de este curso que os va a enseñar a diseñar un sitio web!

Vamos a pasar un rato juntos.

Os sugiero que empecemos con lo más simple, pero también con lo más importante, pregunta: **¿cómo funcionan los sitios web?**

¿Cómo funcionan los sitios web?

¿Cómo funcionan los sitios web?

¡Es muy importante que hablemos un poco sobre esto antes de que nos metamos de lleno en el diseño web!

Estoy seguro de que visitáis sitios web a diario. Para hacer esto ejecutáis un programa que se denomina navegador web haciendo clic en uno de estos iconos que se muestran en la figura siguiente.



Los navegadores web son programas que muestran sitios web

Podéis visitar cualquier sitio web con un navegador. Aquí, por ejemplo tenemos un navegador que muestra la famosa página web de Wikipedia (ver la figura siguiente).



¡Estoy seguro de que estáis acostumbrados a usar un navegador web! Actualmente todo el mundo sabe como navegar en la Red..., pero ¿quién sabe realmente cómo funciona la Red? ¿Cómo diseñar sitios web como este?

Hemos oído hablar del HTML y de CSS. ¿Tienen algo que ver con la forma en la que funcionan los sitios web?

¡Totalmente!

Existen **lenguajes informáticos** para crear sitios web. Todos los sitios web se basan en estos lenguajes que son fundamentales actualmente y se usan en todos los lugares. Son los verdaderos cimientos de la Red. El lenguaje HTML lo inventó Tim Berners-Lee en 1991...

Tim Berners-Lee aún sigue de cerca las evoluciones de la Red. El creó el *World Wide Web Consortium* (W3C) que define las nuevas versiones de estos lenguajes. Más recientemente él también creó la *World Wide Web Foundation* que analiza y monitoriza las evoluciones de la Red.

Mucha gente confunde erróneamente Internet con la Red. Deberías saber que la Red es *parte de* Internet.

Internet es, por lo tanto, un gran sistema que incluye entre otras cosas la Red, los correos electrónicos, los mensajes instantáneos, etc.

Así que Tim Berners-Lee no inventó Internet, sino «únicamente» la Red.

Los lenguajes HTML y CSS son los cimientos del funcionamiento de todos los sitios web. Cuando visitáis un sitio web con un navegador deberíais saber que detrás de bastidores las ruedas giran para permitir que el sitio web se visualice. El ordenador se basa en lo que se le explicó en HTML y CSS para saber que mostrar (ver la figura siguiente).

[illegible]

¿Probablemente os estaréis preguntando por qué necesitáis conocer dos lenguajes para crear sitios web? ¡Voy a responder a esto de inmediato!

Para crear un sitio web, tenéis que darle instrucciones al ordenador. No basta con teclear el texto que queráis poner en el sitio web (como lo haríais en un procesador de texto como Word, por ejemplo). También tenéis que mostrarle donde poner el texto, insertar las imágenes, crear los enlaces entre las páginas, etc.

La función de HTML y de CSS.

Para explicarle al ordenador lo que queréis hacer, tenéis que usar un lenguaje que él entienda. ¡Y es aquí donde la cosa se pone difícil porque *tenéis que aprender dos lenguajes!*

¿Por qué se crearon dos lenguajes? ¿No habría bastado con uno?

Probablemente estéis pensando que manejar dos lenguajes va a ser doblemente complicado y va a llevar el doble de tiempo aprenderlos..., ¡pero no es así! Os puedo asegurar que si hay dos lenguajes se diseñó así para hacer las cosas más fáciles. Estaremos lidiando con dos lenguajes que *se complementan* entre ellos debido a que tienen funciones distintas:

- **HTML** (*HyperText Markup Language (lenguaje de marcas de hipertexto)*): apareció por primera vez en 1991 cuando se lanzó la Red. Su función es gestionar y organizar el contenido. Por lo tanto, usaréis el HTML para escribir lo que queréis que muestre la página: texto, enlaces, imágenes, etc. Podéis decir por ejemplo: «Este es el título, este es el menú, aquí está el texto principal de la página, aquí está una imagen a mostrar, etc.».
- **CSS** (*Cascading Style Sheets (hojas de estilo en cascada)*): la función del CSS es gestionar la apariencia del sitio web (trazado, posicionamiento, diseño, color, tamaño del texto, etc.). Este lenguaje se introdujo para complementar al lenguaje HTML en 1996.

También puede que hayáis oído sobre el lenguaje XHTML. Este es una variante del HTML que es más rigurosa y por tanto un poco más complicada de manejar.

En pocas palabras el HTML apareció por primera vez en 1991. A comienzos de 2000, el W3C lanzó el XHTML y declaró que este representaba el futuro..., pero el XHTML no tuvo tanto éxito como se esperaba. Así que se volvió a lo básico en 2009, el W3C abandonó el XHTML y decidió volver al HTML y actualizarlo.

Hay mucha confusión en relación a estos lenguajes porque son muy parecidos. Ninguno de ellos es realmente mejor que el otro. Son solo dos formas diferentes de hacer las cosas. En este curso vamos a trabajar con la última versión del HTML (HTML5) que es ahora el lenguaje del futuro que todo el mundo se anima a usar.

Podéis crear bastante bien un sitio web solo con HTML, pero no se va a ver muy bonito, se mostrará la información «granulada». Esta es la razón por la que el lenguaje CSS siempre se usa para complementarlo.

Para daros una idea de esto la figura siguiente muestra como se ve la misma página sin CSS y luego con CSS.

HTML
(no CSS)



HTML + CSS



Con o sin CSS

El HTML define el contenido (como podéis ver, ¡no está muy pulido!). El CSS se usa para colocar el contenido y definir la presentación: el color, la imagen de fondo, los márgenes, el tamaño del texto, etc.

Como os podéis imaginar el CSS necesita una página en HTML para funcionar. Esta es la razón por la que primero vamos a aprender los fundamentos del HTML antes de pasar a lidiar con el aspecto decorativo en CSS.

Como resultado vuestras primeras páginas no tendrán un aspecto muy bonito, pero ¡a quién le importa! Esto no va a durar mucho tiempo.

Las diferentes versiones del HTML y del CSS

Con el paso del tiempo los lenguajes HTML y CSS han cambiado significativamente. En la primera versión del HTML (HTML 1.0), ¡ni siquiera se podía mostrar imágenes!

Aquí tenemos un pequeño historial muy breve de los lenguajes para vuestros conocimiento general...

Versiones del HTML

- **HTML 1:** esta fue la primera versión que creó Tim Berners-Lee en 1991.
- **HTML 2:** la segunda versión del HTML apareció en 1994 y finalizó en 1996 con la publicación del HTML 3.0. Esta fue la versión que sentó las bases para las versiones futuras del HTML. Las reglas y la operatividad de esta versión las otorgó el W3C (aunque la primera versión la creó un solo hombre).
- **HTML 3:** apareció en 1996 y esta versión nueva del HTML añadió muchas posibilidades al lenguaje, como tablas, *applets*, *scripts*, posicionamiento de texto alrededor de las imágenes, etc.
- **HTML 4:** esta es la versión más común del HTML (más específicamente es HTML 4.01). Apareció por primera vez en 1998 y propuso el uso de tramas (estas seccionan una página web en varias partes), tablas más complejas, mejoras en los formularios, etc. Lo más importante es que esta versión permitió por primera vez el uso de hojas de estilo, ¡nuestro famoso CSS!
- **HTML 5:** esta es la última versión. Aún no está muy extendida y llama mucho la atención porque proporciona muchas mejoras, como la capacidad para incluir vídeos fácilmente, un diseño mejorado del contenido, nuevas funciones para los formularios, etc. Esta es la versión que vamos a estudiar en detalle.

Versiones del CSS

- **CSS 1:** la primera versión del CSS se usó a partir de 1996. Sentó las bases de este lenguaje que permite mostrar las páginas web con colores, márgenes, tipos de letra, etc.
- **CSS 2:** apareció en 1999 y luego se completó con la versión CSS 2.1. Esta versión nueva del CSS añadió numerosas opciones. Ahora podemos usar con bastante exactitud las técnicas de colocación para mostrar objetos en el lugar que queremos que estén en la página.
- **CSS 3:** esta es la última versión, esta incluye funciones muy esperadas, como bordes redondeados, degradado, sombras, etc.

Tened en cuenta que HTML5 y CSS3 no son versiones que el W3C haya finalizado completamente. Sin embargo, aunque puede que se hagan cambios menores en estos lenguajes, de verdad os recomiendo que empecéis con estas versiones nuevas hoy. Hacen muchas contribuciones y merecen la pena. También muchos sitios web profesionales actualmente están contruidos con estas últimas versiones.

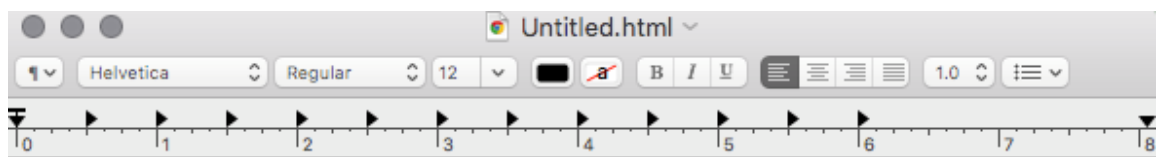
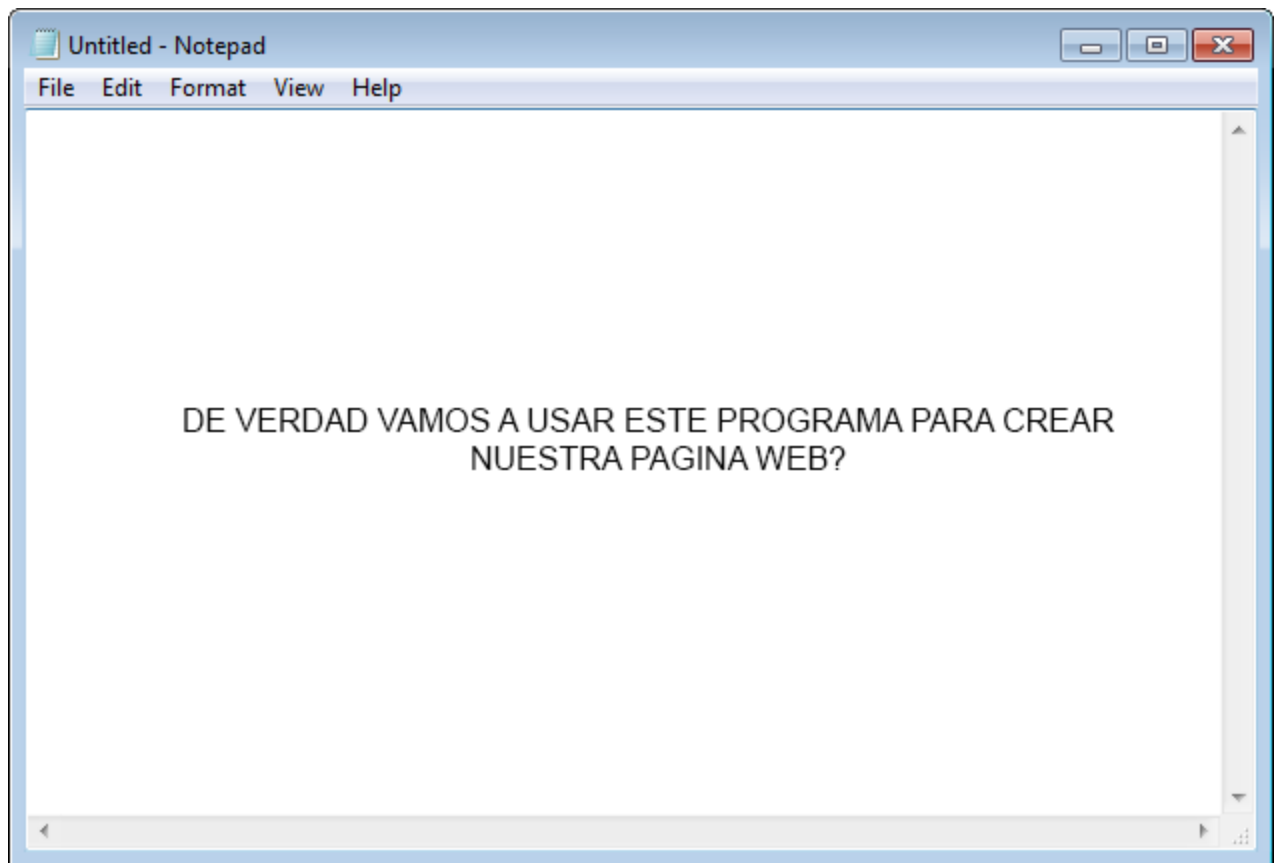
El editor de texto

¿Qué software voy a necesitar para construir un sitio web?

¿Voy a tener que comprar un software muy complejo que voy a tardar meses en entender?

Hay de hecho un gran número de programas informáticos para el diseño de sitios web. Pero os puedo asegurar que no vais a necesitar pagar ni un duro por ellos. ¿Por qué pagar por un software informático complicado cuando ya tenéis en casa todo lo que necesitáis?

Sí, coged aire porque todo lo que necesitáis es... ¡El el Editor de Texto para usuarios MAC o el bloc de notas para los de WINDOWS! (ver la figuras siguientes)



DE VERDAD VAMOS A USAR ESTE PROGRAMA PARA CREAR
NUESTRA PAGINA WEB?

Extraño pero cierto: podéis diseñar bastante bien una página web usando simplemente el Bloc de notas, el programa de edición de texto que viene incluido de serie en Windows.

Sin embargo, hay software más potente disponible actualmente y realmente nadie usa ya el Bloc de notas. Estos programas para **el diseño web** se pueden clasificar en dos categorías:

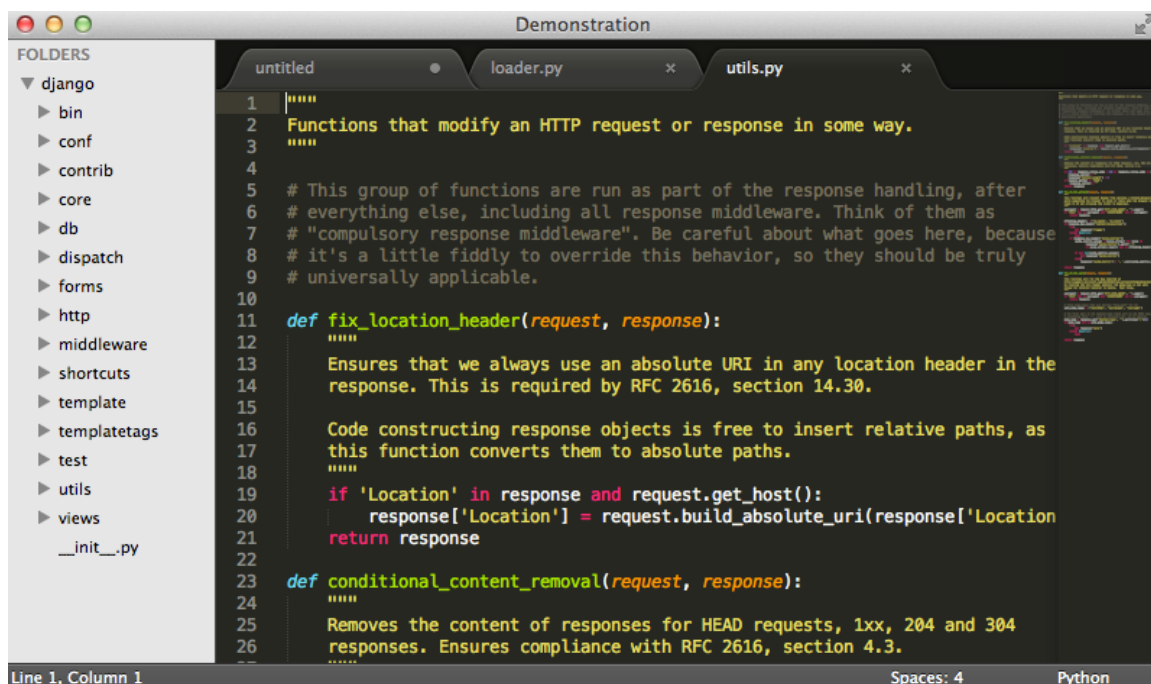
- **WYSIWYG** (*What You See Is What You Get* (lo que ves es lo que hay)): estos programas son muy fáciles de usar y permiten crear sitios web sin aprender ningún lenguaje en particular. Entre los más conocidos están: Nvu, Microsoft Expression Web, Dreamweaver... ¡e incluso Word! Su desventaja principal es la calidad del código HTML y CSS que generan automáticamente estas herramientas, el cual tiene a menudo una calidad bastante mala. Un buen diseñador de páginas web va a tener que saber tarde o temprano HTML y CSS, y esta es la razón por la que no recomiendo el uso de estas herramientas.
- **Editores de texto**: estos son programas para escribir código. Generalmente se usan para otros lenguajes y no únicamente para HTML y CSS. ¡Han demostrado que son aliados poderosos para los diseñadores de páginas web!

Como os habréis dado cuenta, os voy a pedir que uséis un editor de texto en este curso. En todos los sistemas operativos podéis usar los siguientes editores de texto:

- [Sublime Text](#)
- [Atom](#)
- [Brackets](#)

Con estos tres editores que os he recomendado, que son los más usados entre los desarrolladores web, estaréis más que cubiertos.

Cuando tengáis el software instalado, sólo tendréis que crearos vuestro proyecto con la estructura que queráis e ir creando poco a poco ficheros que en nuestro caso serán en HTML y CSS. Sin embargo estos editores os permiten jugar con infinidad de lenguajes de programación.



The screenshot shows a code editor window titled "Demonstration". On the left, there is a "FOLDERS" panel showing a Django project structure with folders like bin, conf, contrib, core, db, dispatch, forms, http, middleware, shortcuts, template, templatetags, test, utils, views, and __init__.py. The main editor area shows the content of "utils.py". The code is written in Python and includes docstrings and function definitions. The status bar at the bottom indicates "Line 1, Column 1", "Spaces: 4", and "Python".

```
1  """
2  Functions that modify an HTTP request or response in some way.
3  """
4
5  # This group of functions are run as part of the response handling, after
6  # everything else, including all response middleware. Think of them as
7  # "compulsory response middleware". Be careful about what goes here, because
8  # it's a little fiddly to override this behavior, so they should be truly
9  # universally applicable.
10
11 def fix_location_header(request, response):
12     """
13     Ensures that we always use an absolute URI in any location header in the
14     response. This is required by RFC 2616, section 14.30.
15
16     Code constructing response objects is free to insert relative paths, as
17     this function converts them to absolute paths.
18     """
19     if 'Location' in response and request.get_host():
20         response['Location'] = request.build_absolute_uri(response['Location'])
21     return response
22
23 def conditional_content_removal(request, response):
24     """
25     Removes the content of responses for HEAD requests, 1xx, 204 and 304
26     responses. Ensures compliance with RFC 2616, section 4.3.
27     """
```

De momento no os preocupéis por lo que significa todo este galimatías. Solo quería daros una visión general de lo que puede hacer el software.

Navegadores

Por qué un navegador es importante

Como expliqué anteriormente, el trabajo de un navegador es interpretar el código HTML y el CSS para mostrar un resultado visual en la pantalla. Si vuestro código CSS dice que «los títulos están en rojo» el navegador mostrará los títulos en rojo. ¡La función del navegador es por tanto primordial!

Aunque no lo creáis un navegador es un programa extremadamente complejo. Entender el código HTML y el CSS no es de hecho una tarea fácil. El problema principal como os daréis cuenta rápidamente es que *¡los navegadores no muestran todos los sitios web exactamente de la misma manera!* Os tendréis que acostumbrar a esto y habituaos a comprobar con frecuencia que vuestro sitio web funciona correctamente en la mayoría de los navegadores.

Los navegadores en los ordenadores

Descarga de navegadores

Hay muchos navegadores diferentes. Los principales que deberíais conocer se muestran en la figura siguiente.

Navegador	OS	Descargar	Comentarios
Google Chrome  Google Chrome	Windows Mac Linux	<u>Descargar</u>	El navegador de Google que es muy fácil de usar y muy rápido. Es el navegador que uso todos los días.
Mozilla Firefox  Firefox	Windows Mac Linux	<u>Descargar</u>	El navegador famoso y renombrado de la Fundación Mozilla. Lo uso con frecuencia para probar los sitios web.
Internet Explorer	Windows	<u>Descargar</u> (Preinstalado en Windows)	El navegador de Microsoft que está instalado en todos los PC de Windows. Lo uso con frecuencia para probar los sitios web.

Navegador	OS	Descargar	Comentarios
 IE9			
Safari  Safari	Windows Mac	Descargar (Preinstalado en Mac OS X)	El navegador de Apple que viene instalado en todos los Macs.
Opera  Opera	Windows Mac Linux	Descargar	El eterno <i>desconocido</i> . Aunque ofrece muchas funciones se usa menos.

Es recomendable que instaléis varios navegadores en vuestro ordenador para que os aseguréis que vuestro sitio web funciona correctamente en cada uno de ellos. En general, os recomiendo que probéis vuestro sitio web con frecuencia al menos en Google Chrome, Mozilla Firefox e Internet Explorer.

Tomad en cuenta que Safari y Google Chrome muestran los sitios web casi de la misma forma. No tenéis que probar vuestro sitio web en Safari y Google Chrome, aunque es siempre más seguro.

En la siguiente figura podéis ver una visión general de algunos de los navegadores principales en la página de inicio de Google.



Entender la diferencia entre navegadores

Como he mencionado anteriormente, los navegadores no siempre muestran los sitios web *exactamente* de la misma forma. ¿Por qué? Esto se debe a que los navegadores no siempre conocen las últimas funciones del HTML y del CSS. Por ejemplo Internet Explorer se ha quedado por mucho tiempo rezagado en relación a algunas funciones del CSS (y paradójicamente ha estado por delante de otras).

Para complicar el asunto hay varias versiones de los navegadores que coexisten:

- Firefox 2, Firefox 3.5, Firefox 3.6 y Firefox 4;
- Internet Explorer 6, Internet Explorer 7, Internet Explorer 8 y Internet Explorer 9;
- Chrome 8, Chrome 9 y Chrome 10;
- ...

Aunque cada versión es compatible con las nuevas funciones, esto se convierte en un problema para **los administradores web** como vosotros que crean sitios web cuando los usuarios no actualizan los navegadores.

Chrome ha resuelto en gran medida este problema con el establecimiento de actualizaciones automáticas que no requieren la intervención del usuario. Muchos de los usuarios de Firefox se olvidan de actualizar e Internet Explorer se esfuerza para conseguir que los usuarios actualicen a la última versión que también requiere una actualización de Windows (por ejemplo Internet Explorer 9 no está disponible para Windows XP).

Sitios como normansblog.de y caniuse.com proporcionan una lista actualizada con las funciones del CSS que soportan las diferentes versiones de cada navegador (ver figura siguiente).

Browser Rendering Engine	 Firefox Gecko					 Safari Webkit			 Chrome Webkit		 Internet Explorer Trident			 Opera Presto	
Version	3.6	4	5	6	7b – 8a	4	5	5.1	9	10 – 14	6 – 8	9	10a3	10.6 – 11	11.1 – 11.5
Animations	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✗	✗
Background Gradients	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✗	▲
Background Size	✓	✓	✓	✓	✓	▲	✓	✓	✓	✓	✗	✓	✓	✓	✓
Border Image	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓
Border Radius	▲	▲	▲	▲	▲	▲	✓	✓	✓	✓	✗	✓	✓	✓	✓
Box Shadow	✓	✓	✓	✓	✓	▲	✓	✓	▲	✓	✗	✓	✓	✓	✓
Columns	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✗	✓
Font Face	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	▲	✓	✓	✓	✓
HSLa	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓
Hyphens	✗	✗	✗	✓	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗
Multiple Backgrounds	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓
Opacity	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓

[Lista de funciones compatibles por navegadores](#)

Como podéis ver, es... complicado.

El problema más grande se encuentra con más frecuencia en las versiones más antiguas de Internet Explorer (IE6, IE7 y IE8). Deberíais comprobar cómo se ve vuestro sitio web en estas versiones antiguas... ¡Esperad lo inesperado! Especialmente aseguraos de que vuestro sitio web se muestra sin errores sin intentar obtener exactamente el mismo resultado en las versiones antiguas de estos navegadores.

Navegadores en teléfonos móviles

Además de los navegadores que he presentado se pueden encontrar versiones de estos navegadores para teléfonos móviles, particularmente para **smartphones**.

Ya que más y más gente visita actualmente los sitios web con sus móviles tenemos que tener por lo menos alguna idea de cómo funcionan los navegadores en los teléfonos móviles.

De hecho, no os perderéis: la mayoría de los navegadores de los smartphones son los mismos que los de los ordenadores en una versión más ligera apta para teléfonos móviles. Todo depende el tipo de teléfono móvil.

- **iPhone:** en el iPhone de Apple se usa el navegador Safari Mobile. Esta es una versión *ligera*, pero aún así muy completa del Safari para ordenadores.
- **Android:** Los teléfonos móviles Android usan el navegador Chrome Mobile. Una vez más, esta es una versión para teléfonos móviles.
- **Teléfono Windows:** El teléfono Windows incluye... ¡Internet Explorer Mobile! El principio es el mismo al igual que los navegadores anteriores: es una versión dedicada a teléfonos móviles.
- **Blackberry:** Las Blackberry's son la excepción a la regla porque tienen su propio navegador (no hay equivalente en ordenadores). Sin embargo, las versiones más recientes de este navegador están basadas en el núcleo común de Safari y Chrome (el motor de renderizado Webkit). Como resultado la visualización es normalmente similar a la de Safari y Chrome.

Los navegadores para teléfonos móviles soportan la mayoría de las últimas funciones del HTML y del CSS. Además, el sistema de actualización automático de los teléfonos móviles garantiza que los usuarios tengan la última versión más a menudo.

Sin embargo, deberíais ser conscientes de que hay diferencias entre estos navegadores para teléfonos móviles y ¡es aconsejable que también probéis el sitio web en estos dispositivos! Concretamente, debido a que la pantalla es mucho más estrecha os deberíais asegurar que vuestro sitio web se muestra correctamente.

Los dispositivos táctiles como las tabletas tienen los mismos navegadores, pero simplemente la pantalla es más amplia. Por tanto, el iPad viene de serie con Safari Mobile por ejemplo.

En resumen

- La Red la inventó Tim Berners a principios de los 90.
- Para crear sitios web usamos dos lenguajes informáticos:
 - HTML que nos permite escribir y organizar el contenido de la página (párrafos, títulos...);
 - CSS que nos permite dar formato a la página (color, tamaño...).
- Ha habido varias versiones de HTML y de CSS. Las últimas son HTML5 y CSS3
- Un navegador web es un programa que muestra sitios web. Interpreta los lenguajes HTML y CSS para saber que es lo que tiene que mostrar.
- Existen un montón de navegadores web: Google Chrome, Mozilla Firefox, Internet Explorer, Safari, Opera... Cada uno de ellos mostrará los sitios web de una forma ligeramente diferente.
- En este módulo aprenderemos a usar los lenguajes HTML y CSS. Trabajaremos con un programa llamado «editor de texto».

Vuestra primera página web en HTML

Entonces, ¿habéis instalado todo el software? Deberíais tener ahora un o varios editores de texto para *crear vuestro sitio web* (Sublime Text, Atom o Brackets) y varios navegadores para *probar* el sitio (Mozilla Firefox, Google Chrome, Internet Explorer, etc.).

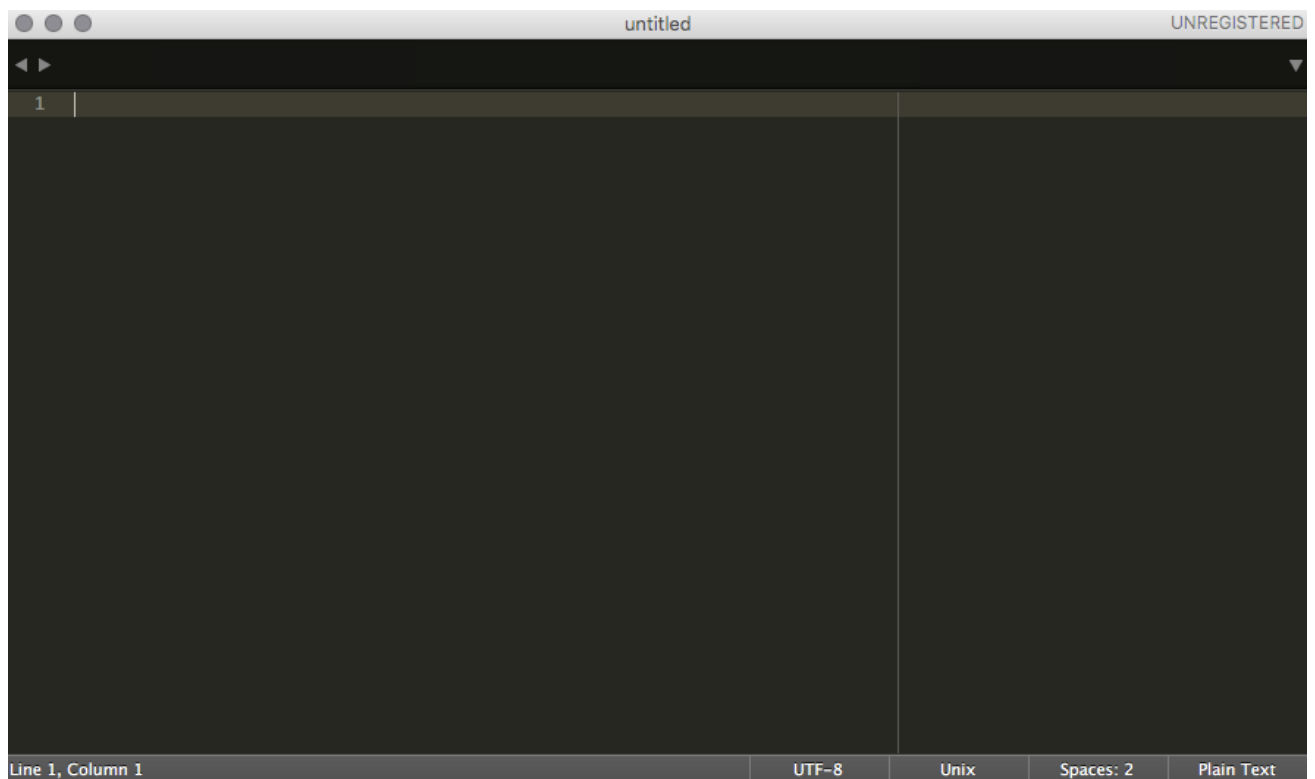
¡Pasaremos a la acción en este capítulo! ¡Vamos a descubrir los fundamentos de HTML y a guardar nuestra primera página web!

Por supuesto, no os creáis que vais a producir una página web excepcional justo en el capítulo dos, pero esperad y veréis... ¡vendrá!

Crear una página web con el editor

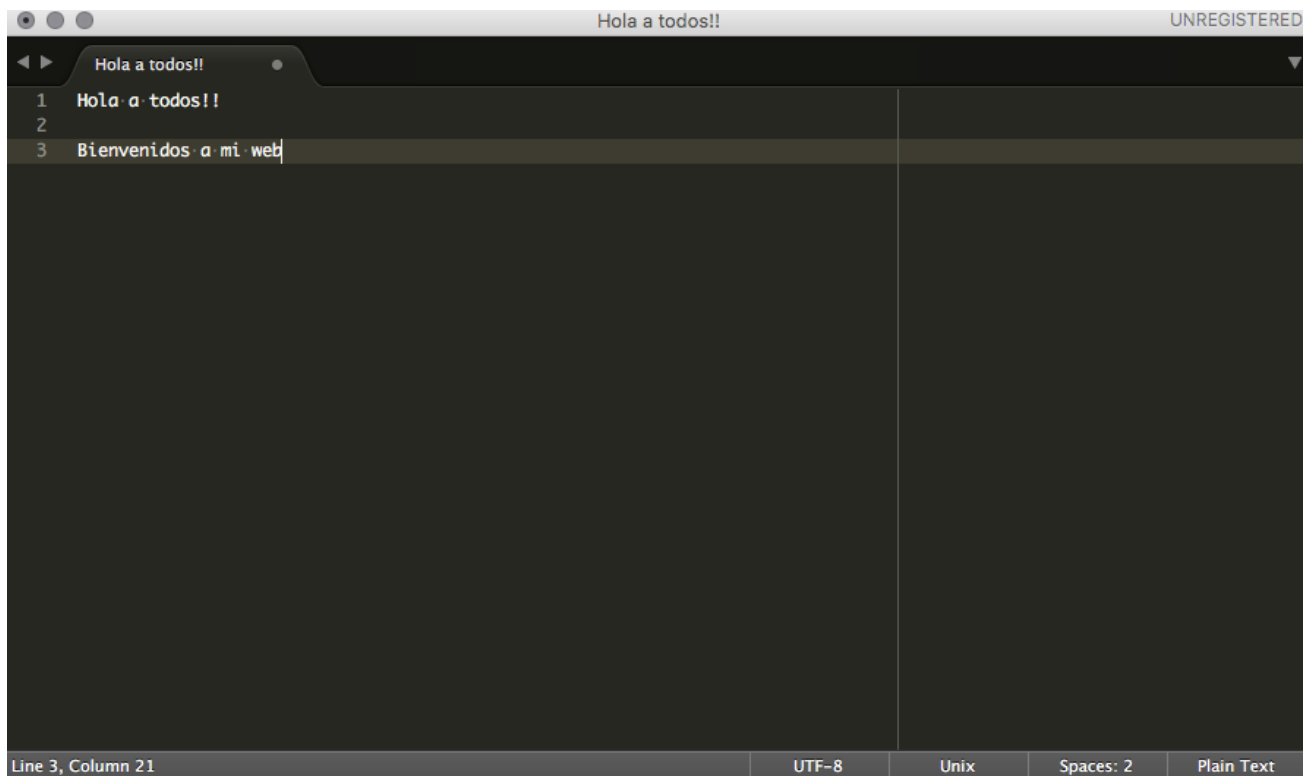
Entonces, ¡sigamos adelante! Como acabo de decir vamos a crear nuestro sitio web en un editor de texto. Deberíais haber instalado uno como os aconsejé en el primer capítulo: no importa qué editor de texto sea. Estos programas tienen un propósito muy simple, ¡permiten que tecleéis el texto!

En este ejemplo, con Sublime Text. Así que vamos a abrirlo (ver figura siguiente).

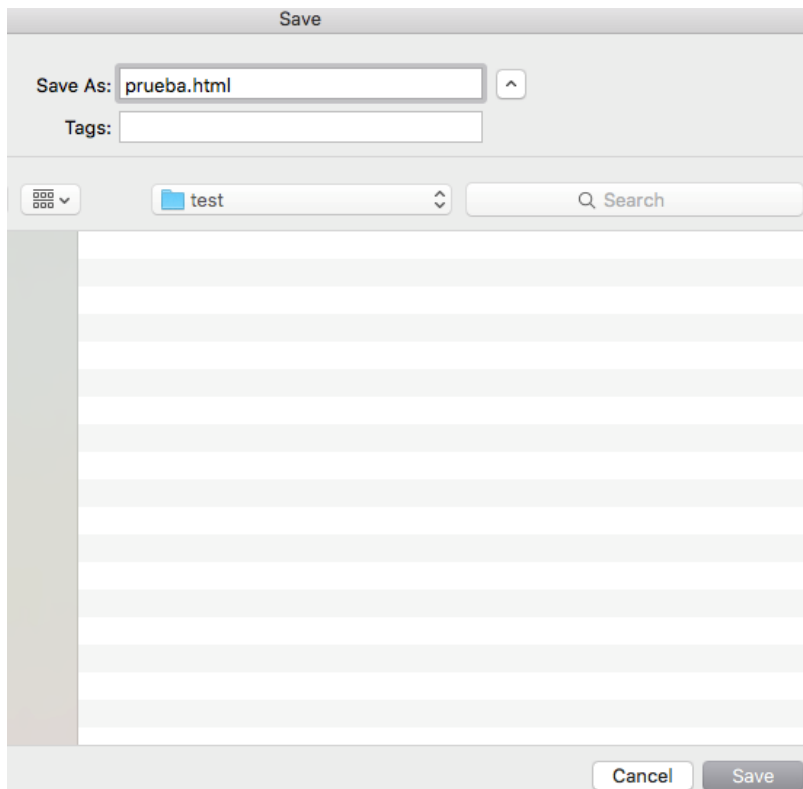


VALE, ¿entonces qué hacemos ahora? ¿Qué vamos a escribir en esta página en blanco?

Hagamos una pequeña prueba. Simplemente escribid los que os venga a la mente.

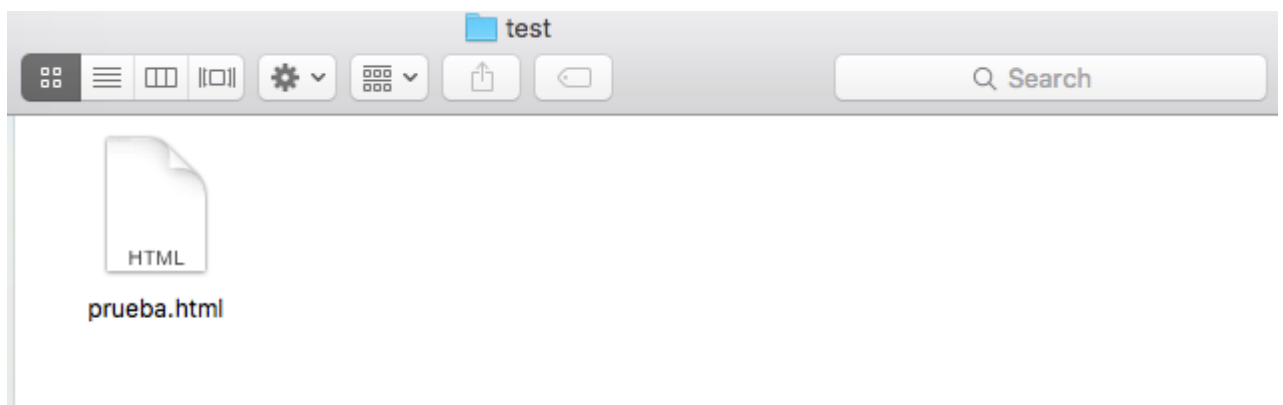


Ahora vamos a guardar este archivo. Hacer esto es muy sencillo como en todos los programas tenéis un menú **Archivo > Guardar como**. Un cuadro de diálogo os pregunta dónde queréis guardar el archivo y qué nombre tiene. Guardadlo donde queráis. Dadle al archivo cualquier nombre que queráis que termine con `.html`, por ejemplo `prueba.html` como se muestra en la figura siguiente.



Os recomiendo que creéis una carpeta nueva en vuestros documentos que contenga los archivos para el sitio.

Ahora abrid el explorador de archivos en la carpeta en la que hayáis guardado la página. Veréis el archivo que acabáis de crear (figura siguiente).



Os puede salir un icono dependiendo del sistema operativo que uséis que represente el archivo va en función del navegador web que tengáis predeterminado. Por ejemplo estos son los iconos que aparecen cuando vuestro navegador predeterminado es Firefox o Internet Explorer (ver la figura siguiente).

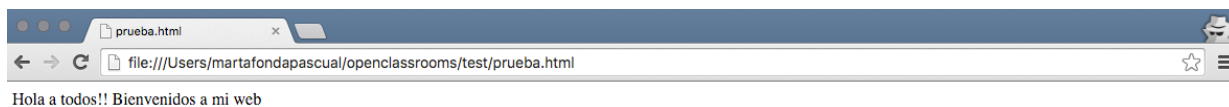


Icono del archivo con Firefox



Icono del archivo con Internet Explorer

Simplemente haced doble clic en este archivo y... vuestro navegador se abrirá y mostrará el texto que habéis escrito como se ve en la figura siguiente.



Visualización de la página web

¡Parece que no funciona muy bien! ¿¿Todo el texto se muestra en la misma línea aunque escribimos dos líneas de texto diferentes!?

¡De hecho, buena observación!

El texto se muestra en la misma línea aunque pedimos que se tecleara en dos líneas diferentes. ¿Qué está pasando?

De hecho, para crear una página web no basta solo con teclear el texto como acabamos de hacer. Además de este texto también tenéis que teclear lo que se conoce como **etiquetas** que le darán instrucciones al ordenador como «ir a la línea siguiente», «mostrar una imagen», etc.

Etiquetas y sus atributos

Bueno, todo eso fue demasiado fácil. No basta «solo» con teclear el texto en el editor también tenéis que darle instrucciones al ordenador. Para hacer esto en HTML se usan **etiquetas**.

Etiquetas

Las páginas HTML están llenas de las llamadas etiquetas. Aunque estas etiquetas no las pueden ver aquellos que visiten vuestro sitio, ellas permiten que el ordenador entienda que es lo que tiene que mostrar.

Las etiquetas son fáciles de identificar. Están rodeadas por «corchetes angulares» en otras palabras los símbolos < y >, como este: <etiqueta>

¿Para qué sirven? Indican el tipo de texto al que rodean. Quieren decir por ejemplo: «esto es el título de la página», «esto es una foto», «esto es un párrafo», etc.

Hay dos tipos de etiquetas: etiquetas en pares y etiquetas huérfanas.

Etiquetas en pares

Se abren, contienen texto y se cierra posteriormente. Esta es la apariencia que tienen:

```
<p>Esto es un párrafo</p>
```

Vais a ver una etiqueta que abre (<párrafo>) y una etiqueta que cierra (</párrafo>) que indica el final del párrafo. Para el ordenador, esto significa que todo lo que *no* esté entre esas dos etiquetas no es un párrafo.

```
Esto no es un párrafo <p>Esto es un párrafo</p> Esto no es un párrafo
```

Etiquetas huérfanas

Estas son las etiquetas que se usan con más frecuencia para insertar un elemento en un punto determinado (por ejemplo una imagen). No es necesario definir el comienzo y el final de la imagen. Solo tenéis que decirle al ordenador que «inserte una imagen aquí»

Una etiqueta huérfana se escribe así:

```
<img></code>
```

Tomad en cuenta que el final / no es obligatorio. Podríais únicamente teclear `<img>`. Para evitar confundirlas con el primer tipo de etiqueta, sin embargo, los administradores web recomiendan que esto / (barra) se debería añadir al final de las etiquetas huérfanas. Así que veréis que añado una / a las etiquetas huérfanas y os recomiendo que hagáis lo mismo; es una buena práctica.

Atributos

Los atributos son como las opciones de la etiqueta. Las complementan para proporcionar información adicional. El atributo se coloca después del nombre de la etiqueta que abre y frecuentemente es un valor como este:

```
<NombreDeLaEtiqueta AtributoDeLaEtiqueta="valor">
```

¿Para qué sirve? Vamos a usar la etiqueta `<img />` que acabamos de ver. Por sí misma no tiene mucha utilidad. Podemos añadir un atributo que indique el nombre de la imagen a mostrar:

```
</code>
```

El ordenador entonces entiende que tiene que mostrar la imagen que contiene el archivo `foto.jpg`.

En el caso de una etiqueta que funciona como una «pareja», los atributos únicamente se colocan en la etiqueta de apertura y no en la de cierre. Por ejemplo este código indica que la cita es de Neil Armstrong y data del 21 de julio de 1969:

```
<blockquote author="Neil Armstrong" date="2107/07/1969">
```

```
Es un pequeño paso para el hombre, pero un gran paso para la humanidad.</blockquote>
```

Todas las etiquetas que acabamos de ver son imaginarias. Vamos a descubrir las etiquetas reales en unos instantes.

Estructura básica de una página HTML5

Volvamos a nuestro editor de texto (en este caso Sublime Text)

He añadido espacios al comienzo de algunas líneas para «cambiar» las etiquetas. Aunque esto no es obligatorio y no afecta a la forma en la que se muestra la página hace que el código fuente sea más legible. Esto se llama **indentation (sangría)**. En vuestro editor simplemente presionad la tecla `Tab` para obtener el mismo resultado.

El resultado que se obtiene cuando se copió dentro del Sublime Text se muestra en la figura siguiente.



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>My Title</title>
6 </head>
7 <body>
8
9 </body>
10 </html>
11
```

El código HTML5 mínimo en Sublime Text

Notaréis que las etiquetas se abren y se cierran en un orden específico. Por ejemplo, la etiqueta `<html>` es la primera que se abre y también la última que se cierra (al final del código con `</html>`). *Las etiquetas se tienen que cerrar en el orden inverso en el que se abrieron.* Por ejemplo:

- `<html><body></body></html>`: **correcto**. Una etiqueta que se abrió dentro de otra etiqueta también se tiene que cerrar dentro de esa etiqueta.
- `<html><body></html></body>`: **incorrecto**, las etiquetas están entrecruzadas.

¡No os asustéis viendo todas esas etiquetas de un golpe, os voy a explicar lo que hacen!

doctype

```
<!DOCTYPE html>
```

La primera línea se denomina el **doctype**. Es fundamental porque indica que esto es efectivamente una página web HTML.

No es realmente una etiqueta como las otras (empieza con un signo de exclamación) y se puede considerar como una excepción que confirma la regla.

Esta línea doctype antiguamente era extremadamente compleja. Memorizarla era imposible. En XHTML 1.0 se tenía que escribir:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">.
```

Para el deleite de los administradores web se decidió simplificarla en el HTML5. Cuando veáis una etiqueta doctype corta (`<!DOCTYPE html>`) significa que la página está escrita en HTML5.

La etiqueta `</html>`

```
<html>

</html>
```

Esta es la etiqueta principal del código. Incluye el contenido completo de la página. ¡Como podéis ver la etiqueta `</html>` de cierre está situada a la derecha del final del código!

La cabecera `<head>` y el cuerpo `<body>`

Una página web se compone de dos partes:

- La cabecera `<head>`: esta sección proporciona información general sobre la página como su título, la codificación (para la gestión de caracteres especiales), etc. Esta sección suele ser bastante corta. La información que contiene la cabecera no se muestra en la página y es simplemente información general concebida para el ordenador. ¡Sin embargo, es muy importante!
- El cuerpo `<body>`: es aquí donde se encuentra la parte principal de la página. Todo lo que tecleemos aquí se mostrará en la pantalla. La mayor parte de nuestro código se tecleará en el cuerpo.

Por ahora el cuerpo está vacío (volveremos a esto más adelante). Pero vamos a echar un vistazo a las dos etiquetas que contiene la cabecera...

La codificación (`charset`)

```
<meta charset="utf-8" ></code>
```

Esta etiqueta indica la codificación que se usa en el archivo `.html`.

Sin entrar en detalles porque se complica muy rápido, la codificación especifica cómo se guarda el archivo. Es lo que determina la forma en la que se van a mostrar los caracteres especiales (acentos, caracteres japoneses y chinos, caracteres árabigos, etc.)

Hay varias técnicas de codificación con nombres raros que se usan en función del idioma: ISO-8859-1, OEM 775, Windows-1253, etc. No obstante, actualmente solo se debería usar una: UTF-8. ¡Este método de codificación permite que todos los símbolos de todos los idiomas del mundo se muestren virtualmente sin ningún problema! Esta es la razón por la que indiqué `utf-8` en esta etiqueta.

Vuestro archivo se tiene que guardar efectivamente en UTF-8. Podréis ver la codificación en la fila de abajo del editor.

En Sublime Text, si queréis cambiar la codificación seleccionad la codificación asignada en la fila de abajo, haced click en `Set Encoding` y elegid el formato que preferáis para que vuestro archivo se guarde en ese formato desde el principio. Esto únicamente afecta al archivo que esté abierto en el momento.

Si tenéis un problema más adelante cuando se muestren los acentos en vuestra página web significa que hay un problema de codificación. Aseguraos de que la pestaña efectivamente indica UTF-8 y que vuestro archivo se guardó en formato UTF-8 (esto lo puede indicar vuestro editor de texto).

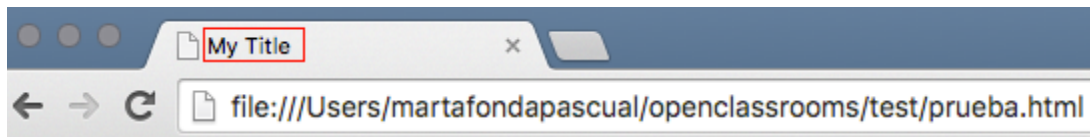
El título principal de la página

```
<title>
```

Este es el título de vuestra página, ¡lo cual es probablemente el elemento más importante! Todas las páginas tienen que tener un título que describe lo que contienen.

Es aconsejable que el título se mantenga bastante corto (normalmente menos de 100 caracteres).

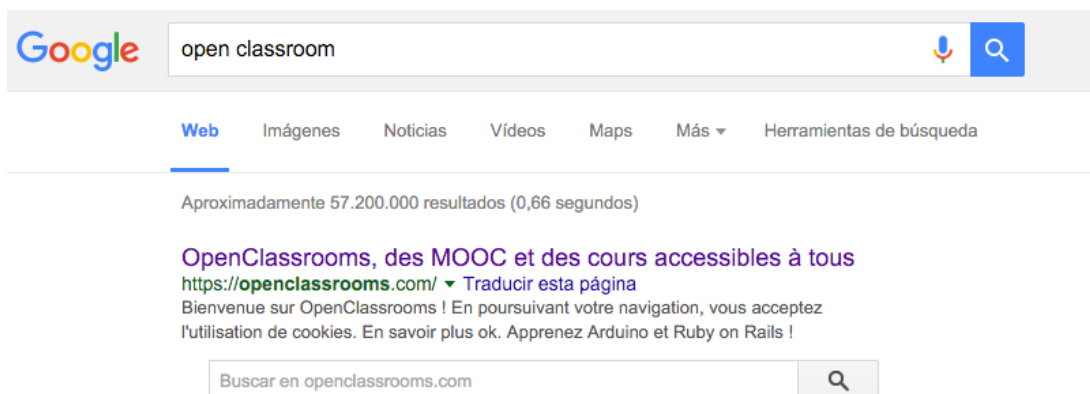
El título no se muestra en la página sino en su parte superior (habitualmente en la pestaña del navegador). Guardad vuestra página web y abridla en el navegador. Veréis que el título se muestra en la pestaña como se ve en la figura siguiente.



El

título de la página figura en la parte superior del navegador

Deberíais saber que el título también aparece en los resultados de búsqueda, como en Google (figura siguiente).



El

El título de la página figura en las búsquedas en Google

¡Así que elegir el título es importante!

Comentarios

En este capítulo aprendimos cómo crear nuestra primera página HTML *real*. Antes de concluir me gustaría añadir una o dos observaciones.

Un **comentario** en HTML es un texto que es simplemente una nota. No se muestra, el ordenador no lo interpreta y no modifica lo que se visualiza en la página.

En pocas palabras, ¿no sirve para nada?

¡Bueno, claro que sí!

Lo usáis *vosotros* y aquellos que lean el código fuente de la página. Podéis usar los comentarios para dejar instrucciones sobre la forma en la que funciona la página.

¿Qué finalidad tiene esto? Os ayudará a recordar cómo funciona la página cuando volváis al código fuente después de que haya pasado un tiempo sin verlo. No es broma, esto le pasa a todos los administradores web.

Insertar un comentario

Un comentario es una etiqueta HTML con un formato muy especial:

```
<!-- Esto es un comentario -->
```

Lo podéis poner en la parte que queráis en el código fuente, no afecta a la página, pero lo podéis usar como ayuda para que os situéis en el código especialmente si es largo.

```
<!DOCTYPE html>

<html>

  <head>

    <!-- Cabecera de la página -->

    <meta charset="utf-8" />

    <title>Título</title>

  </head>

  <body>

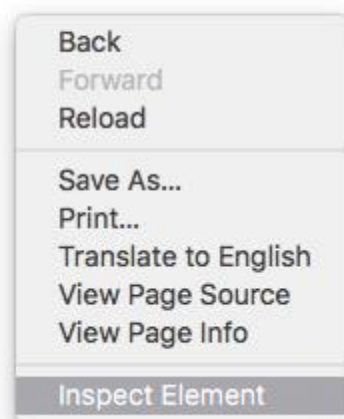
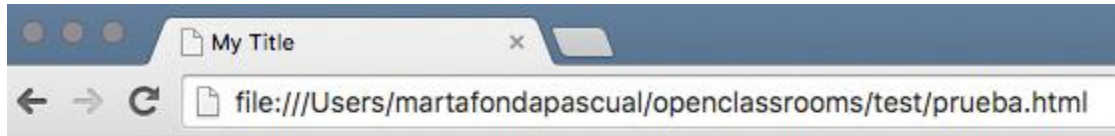
    <!-- Cuerpo de la página -->

  </body>
```

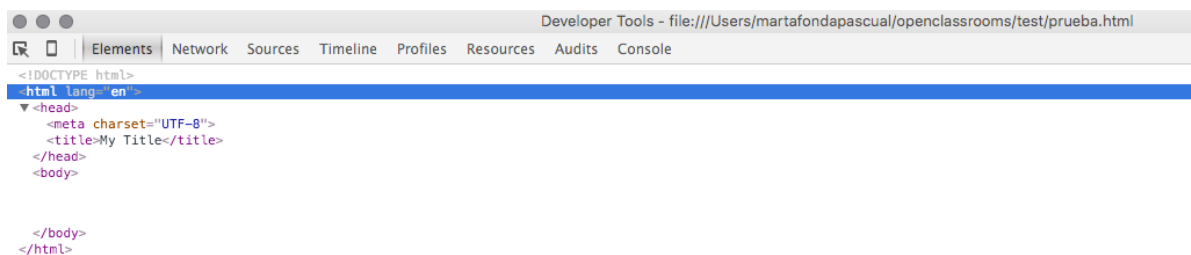
```
</html>
```

¡Todo el mundo puede ver vuestros comentarios... y vuestro código HTML completo!

Voy a concluir con una observación importante: *todo el mundo puede ver el código HTML de vuestra página*, ya que está *online* en la Red. Simplemente haced clic derecho en la página y seleccionad «Inspeccionar elemento o Ver código fuente» (en Chrome, ya que el texto exacto puede variar en función del navegador) como se muestra en la figura siguiente.



A continuación aparece el código fuente (figura siguiente).



Podéis probar esto en cualquier sitio web, ¡funciona! 100 % garantizado. Esto tiene una explicación sencilla, el navegador *necesita* el código HTML para saber que mostrar. El código HTML de todos los sitios por consiguiente es público.

¿La moraleja de la historia? Todo el mundo puede ver vuestro código fuente HTML y no podéis hacer nada al respecto. En consecuencia, no incluyáis información sensible en los comentarios, como contraseñas...

Cuando echéis un vistazo al código de algunos sitios web no os desaniméis si os parece largo. No todos los sitios web están programados en HTML5 ni mucho menos, ¡y el código fuente que algunos administradores web han mal escrito no es un buen ejemplo a seguir!

En resumen

- Un editor de texto (Sublime Text, Brackets, Atom, bloc de notas) se usa para crear un archivo con la extensión `.html` (p. ej. `prueba.html`). Esto será nuestra página web.
- Este archivo se puede abrir en el navegador web simplemente haciendo doble clic sobre él.
- Tecleamos el contenido de la página junto con las etiquetas HTML dentro del archivo.
- Las etiquetas pueden tener varias formas:
 - `<etiqueta> </etiqueta>`: se abren y se cierran para delimitar el contenido, por ejemplo el comienzo y el final de un título.
 - `<etiqueta />`: las etiquetas huérfanas (solo se introduce una etiqueta) se usan para insertar un objeto en un punto determinado, por ejemplo una imagen.
- Las etiquetas van acompañadas algunas veces con atributos que proporcionan información adicional (p. ej. `<img src=«foto.jpg» />`).
- Una página web se compone de dos secciones principales: una cabecera (`<head>`) y un cuerpo (`<body>`).
- El código fuente de cualquier página web se puede visualizar al hacer clic derecho y seleccionando «Inspeccionar Elemento o Ver código fuente» en Google Chrome. En el resto de los navegadores el texto varía pero se accede de la misma forma.

Organizar el texto

Bueno, la página en blanco es bonita, pero si dejáis vuestra página web así es probable que no tenga mucho éxito.

En este capítulo vamos a conocer un buen número de etiquetas HTML. Algunas etiquetas se usan desde la primera versión de HTML y otras se han introducido más recientemente en HTML5.

En este capítulo vamos a ver en orden sucesivo:

- cómo teclear párrafos;
- cómo organizar la página con títulos;
- cómo resaltar palabras determinadas en el texto;
- cómo organizar la información en listas.

Párrafos

El texto en una página web se teclea habitualmente en párrafos. En el lenguaje HTML la etiqueta `<p>` se usa para delimitar los párrafos.

```
<p>¡Hola y bienvenidos a mi sitio!</p>
```

- `<p>` significa "comienzo del párrafo";
- `</p>` significa "fin del párrafo";

Como os conté en el capítulo anterior tecleamos el contenido del sitio web entre las etiquetas `<body></body>`. Así que tenemos que poner nuestro párrafo dentro de estas dos etiquetas y ¡finalmente tendremos nuestra primera página web con texto!

Así que voy a coger exactamente el mismo código del capítulo anterior y lo voy a añadir al párrafo:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Párrafos</title>

</head>


<body>

<p>¡Hola y bienvenidos a mi sitio!</p>

</body>

</html>
```

¡Probadlo, veréis los resultados!

Vale, no es la panacea, pero es un buen comienzo.

Pero no vamos a dejarlo cuando las cosas están yendo tan bien. Ahora vamos a ver algo que es un pelín especial en HTML: el salto de línea. Parece simple, pero realmente no funciona como un procesador de texto común...

Crear una línea nueva

En HTML si presionáis la tecla `Enter` no se crea una línea nueva en la forma en la que estáis acostumbrados. Así que probad este código:

```
<!DOCTYPE html>

<html>
```

```
<head>

<meta charset="utf-8" />

<title>Pruebas de salto de línea</title>

</head>


<body>

<p>¡Hola y bienvenidos a mi sitio web!

Esta es mi primera prueba así que por favor, tomadlo en cuenta, estoy aprendiendo
poco a poco cómo funciona esto.

De momento está un poco vacío, ¡pero volved en 2 o 3 días cuando haya aprendido
algunas cosas más y os puedo asegurar que os sorprenderéis!</p>

</body>

</html>
```

¡¡Todo el texto se muestra en la misma línea aunque haya un salto de línea en el código!! Darle a la tecla del Enter frenéticamente en el editor de texto, por lo tanto, no sirve de nada.

Como podríais esperar, sin embargo, existe por supuesto una forma de empezar una línea nueva en HTML.

De hecho, si queréis teclear un párrafo nuevo solo necesitáis usar una segunda etiqueta `<p>`. ¡Así que al final vuestro código HTML debería estar lleno de etiquetas de párrafos!
¡Así que al final vuestro código HTML debería estar lleno de etiquetas de párrafos!

Un ejemplo:

```
<html>

<head>

<meta charset="utf-8" />

<title>Párrafos</title>

</head>


<body>
```

```
<p>¡Hola y bienvenidos a mi sitio!
```

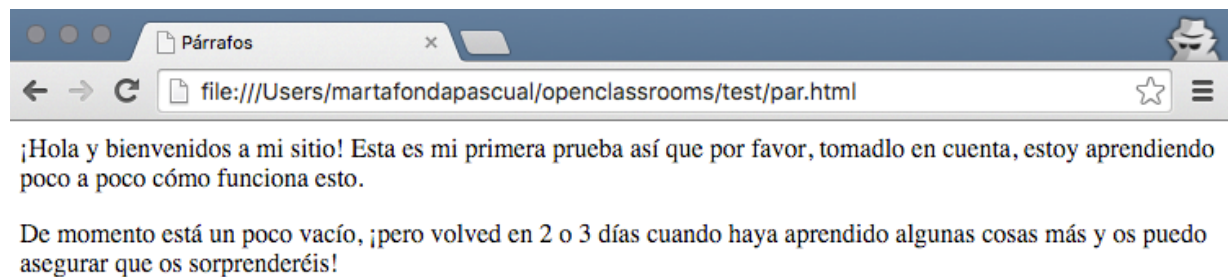
```
Esta es mi primera prueba así que por favor, tomadlo en cuenta, estoy aprendiendo poco a poco cómo funciona esto.</p>
```

```
<p>De momento está un poco vacío, ¡pero volved en 2 o 3 días cuando haya aprendido algunas cosas más y os puedo asegurar que os sorprenderéis!</p>
```

```
</body>
```

```
</html>
```

El resultado se muestra en la figura siguiente.



Dos párrafos con 2 etiquetas <p>

Sí, ¿pero qué pasa si solo quiero pasar a la línea siguiente en un párrafo dando comienzo a una línea nueva?

Bueno, ¿adivina qué? ¡Hay una etiqueta que se llama «salto de línea»!

Es una etiqueta **huérfana** que solo se usa para indicar el salto de línea:
. La tenéis que colocar *dentro de un párrafo*.

Aquí tenemos la forma de usarla en un código:

```
<html>
```

```
<head>
```

```
<meta charset="utf-8" />
```

```
<title>Saltos de línea</title>
```

```
</head>
```

```
<body>
```

```
<p>¡Hola y bienvenidos a mi sitio web!<br />
```

```
Esta es mi primera prueba así que por favor, tomadlo en cuenta, estoy aprendiendo poco a poco cómo funciona esto.</p>
```

```
<p>De momento está un poco vacío, ¡pero volved en 2 o 3 días cuando haya aprendido algunas cosas más y os puedo asegurar que os sorprenderéis!</p>
```

```
</body>
```

```
</html>
```

Teóricamente podéis poner varias etiquetas `<br />` una detrás de la otra para hacer varios saltos de línea, pero esto se considera una mala práctica porque dificulta el mantenimiento del código. Para cambiar un texto con más precisión se usa CSS, que es un lenguaje que complementa el HTML y hablaremos de él un poco más adelante.

Entonces,

- `<p> </p>`: para organizar el texto en párrafos;
- `<br />`: para hacer un salto de línea.

Ahora ya sabemos cómo teclear párrafos, vamos a ver cómo crear títulos.

Títulos

Ya que el contenido de vuestra página incluye varios párrafos, va a ser difícil que vuestros visitantes se sitúen. Es aquí donde los títulos ganan utilidad.

En HTML podéis seleccionar seis niveles de títulos diferentes. Con esto quiero decir que podéis decir «este es un título muy importante», «este es un título un poco menos importante», «este es un título mucho menos importante», etc. Por consiguiente, tenéis seis etiquetas de título diferentes:

- `<h1> </h1>`: significa «título muy importante». En general se usa para mostrar el título de la página al comienzo de la página.
- `<h2> </h2>`: significa "título importante".
- `<h3> </h3>`: asimismo, esto significa un título un poco menos importante (podéis decir «subtítulo» si queréis).
- `<h4> </h4>`: el título es incluso menos importante.
- `<h5> </h5>`: el título no es importante.
- `<h6> </h6>`: el título no es nada importante.

Advertencia: ¡no confundáis esto con la etiqueta `<title>`! La etiqueta `<title>` muestra el título de la página en la barra de título del navegador como hemos visto. Los títulos `<h1>` y similares se usan para crear títulos que se mostrarán *en* la página web.

Qué no os desconcierten todas estas etiquetas. De hecho, seis niveles de títulos son muchos. En la práctica usaremos dos o tres `<h1>`, `<h2>` y `<h3>` y muy raras veces las otras (normalmente no se necesitan seis niveles de títulos diferentes). El navegador mostrará el título realmente

importante en caracteres muy grandes, los títulos que son un poco menos importante en caracteres menos grandes, etc.

¡No seleccionéis la etiqueta del título en función del tamaño que se le dé al texto! Es fundamental que estructuréis vuestra página correctamente y que empecéis con un título nivel 1 (<h1>), luego un título nivel 2 (<h2>), etc. ¡No tiene que haber ningún subtítulo sin un título principal! Si queréis cambiar el tamaño del texto, vamos a aprender cómo se hace esto en CSS más adelante.

Intentad crear una página web con títulos para ver cómo se ve:

```
<html>

<head>

    <meta charset="utf-8" />

    <title>Niveles de títulos</title>

</head>

<body>

    <h1>Título muy importante</h1>

    <h2>Título importante</h2>

    <h3>Título un poco menos importante (subtítulo)</h3>

    <h4>No es un título importante</h4>

    <h5>Título sin importancia</h5>

    <h6>Título sin ninguna importancia</h6>

</body>

</html>
```

Ejemplo de cómo usar los títulos en una página web:

```
<html>

<head>

<meta charset="utf-8" />

<title>Presentación de mi sitio web</title>
```

```

</head>

<body>

    <h1>¡Hola y bienvenidos a mi sitio web!</h1>

    <p>¡Hola y bienvenidos a mi sitio web: SdZ.<br /> ¿Cuál es el contenido del
sitio web SdZ?</p>

    <h2>Cursos de formación para principiantes</h2>

    <p>El sitio web SdZ proporciona cursos de formación (tutoriales) para
principiantes; ¡no se requiere ningún conocimiento para hacer estos cursos!</p>

    <p>¡Por lo tanto, podéis aprender, sin tener ningún conocimiento previo, a
crear un sitio web y a programar y construir entornos 3D!</p>

    <h2>Una comunidad activa</h2>

    <p>¿Tenéis algún problema o hay algo en el curso que no entendáis?
¿Necesitáis ayuda con la creación de vuestro sitio web?<br /> ¡Visitad los foros!
Descubriréis que no estáis solos en esto e indudablemente encontraréis a alguien que
os ayudará encantado a solucionar vuestro problema.</p>

</body>

</html>

```

¡Esta es una página web que está cogiendo forma!

¡Sí, pero quiero centrar el título, escribirlo en rojo y subrayarlo!

Haremos todo eso cuando aprendamos CSS. Me gustaría indicar que `<h1>` no significa «Times New Roman, tamaño 16 pt,...», sino « *título importante* ».

Cuando uséis el lenguaje CSS podréis decir: «quiero de los títulos importantes estén centrados, sean rojos y estén subrayados». De momento solo estamos estructurando la página en HTML. *Estamos escribiendo el contenido antes de pasarlo bien dándole formato.*

Resaltar

Algunas palabras en los párrafos son algunas veces más importantes que otras y querréis que resalten. El HTML proporciona varias formas para resaltar el texto de la página.

Enfatizar

Para *enfatizar* el texto, tenéis que usar la etiqueta `<em>` `</em>`.

Es muy simple de usar, ¡todo lo que tenéis que hacer es encerrar las palabras a enfatizar dentro de estas etiquetas! Voy a volver a usar el ejemplo de hace un momento y a enfatizar algunas palabras:

```
<html>

<head>

<meta charset="utf-8" />

<title>Enfatizar</title>

</head>


<body>

<p>¡Hola y bienvenidos a mi sitio web!<br />

Esta es mi primera prueba así que por favor, <em>tomadlo en cuenta</em>, estoy
aprendiendo poco a poco cómo funciona esto.</p>

</body>

</html>
```

Como podéis ver al usar la etiqueta `<em>` esta tiene un efecto, coloca el texto en *cursiva*. De hecho, es el navegador el que elige la forma en la que se muestran las palabras. Le diremos que las palabras son bastante importantes y para enfatizar esta información cambia la apariencia del texto con el uso de cursiva.

Enfatizar considerablemente

Para enfatizar considerablemente un texto, podéis usar la etiqueta `<strong>` que significa «énfasis fuerte» o si preferís «importante». Se usa exactamente de la misma forma que `<em>`:

```
<html>

<head>

<meta charset="utf-8" />

<title>Énfasis fuerte</title>

</head>


<body>

<p>¡Hola y bienvenidos a mi sitio web!<br />
```

```
Esta es mi primera prueba así que por favor, <strong>tomadlo en cuenta</strong>,
estoy aprendiendo poco a poco cómo funciona esto.</p>

</body>

</html>
```

Sin lugar a dudas veréis que el texto aparecerá en negrita. También en este caso la negrita es un solo una *consecuencia* . El navegador eligió mostrar las palabras importantes en negrita para hacer que destaquen más.

La etiqueta `<strong>` no significa «negrita» sino «importante». Podéis decidir más adelante en CSS si queréis mostrar las palabras «importantes» de una forma diferente a la negrita.

Marcado de texto

La etiqueta `<mark>` se usa para resaltar visualmente una parte del texto. El extracto no se tiene que considerar necesariamente importante, pero queréis que destaque del resto del texto. Esto puede ser útil para resaltar texto que sea relevante, por ejemplo después de buscar en vuestra página web.

```
<html>

<head>

<meta charset="utf-8" />

<title>Marcado de texto</title>

</head>

<body>

<p>¡Hola y bienvenidos a mi sitio web!<br />

Esta es mi primera prueba así que por favor, <mark>tomadlo en cuenta</mark>, estoy
aprendiendo poco a poco cómo funciona esto.</p>

</body>

</html>
```

Por definición `<mark>` tiene el efecto de resaltar el texto. Podemos cambiar en CSS la forma en la que se muestra (decidir resaltar en otro color, poner el texto en un cuadro, etc.) Es el mismo principio como el que os conté para las etiquetas anteriores: indican la *relevancia* de las palabras y no la forma en la que deberían mostrarse.

No os olvidéis: HTML para el contenido, CSS para el estilo

Puede parecer que estoy rizando el rizo, pero es muy importante que entendáis esto correctamente, ya que los principiantes a menudo cometen este gran error en esta etapa. Ven las etiquetas `<em>`, `<strong>`, `<mark>`... y se cuentan a sí mismos: «¡Genial, ya sé cómo poner el texto en cursiva, negrita y subrayarlo en HTML!»

Y aún así... ¡esto no es para lo que se usan estas etiquetas! Ya sé, ya sé, me vais a decir: «sí, pero cuando uso `<strong>` el texto aparece en negrita, entonces se tiene que usar para poner en negrita». Aún así, es un error creer que esta etiqueta se usa para hacer eso.

La finalidad de las etiquetas es indicar la *relevancia* del texto. Por tanto, `<strong>` le dice al ordenador que «este texto es importante». Eso es todo. Y para *mostrar* que el texto es importante, el ordenador decide ponerlo en negrita, ¡pero podría estar escrito igualmente en rojo! Aunque la mayoría de los navegadores muestren el texto importante en negrita nada les exige que lo hagan así.

No lo entiendo. ¿Qué sentido tiene que el ordenador sepa que un texto es importante?
¡No es lo suficientemente inteligente para entender!

¡Pensadlo otra vez! Muchos programas analizan el código fuente de las páginas web y aún más importante los robots de los motores de búsqueda. Estos robots rastrean la Red y leen el código HTML de todos los sitios. Este es el caso de los robots de Google y Bing por ejemplo. Como ellos lo ven, las palabras clave «importantes» suelen tener más valor, así que si alguien hace una búsqueda con estas palabras es más probable que se encuentre con vuestro sitio web. Esto es por supuesto una explicación superficial y no deberíais concluir que el uso descontrolado de la etiqueta `<strong>` de alguna forma va a mejorar vuestro ranking de búsqueda. Solo tenéis que confiar en los ordenadores, ellos entienden lo que significa un texto «importante» y pueden usar esa información.

Entonces, ¿cómo ponéis concretamente un texto en negrita o lo escribís en rojo y todo eso?

Todo eso se hace en CSS. Recordad:

- HTML define el contenido (contenido lógico de elementos);
- CSS define el estilo (apariencia).

Veremos el CSS más adelante de momento nos centraremos en el HTML y sus etiquetas, cada una de ellas tienen un significado concreto.

Listas

Las listas normalmente nos permiten organizar el texto y ordenar la información. Vamos a descubrir dos tipos de listas aquí:

- listas no ordenadas o listas con viñetas;
- listas ordenadas o listas enumeradas.

Listas no ordenadas

Una lista no ordenada tiene esta apariencia:

- Fresas
- Frambuesas
- Cerezas

Es un sistema que nos permite crear una lista sin ningún concepto de orden (no hay «primero» o «último»). Crear una lista no ordenada es muy sencillo. Simplemente usad la etiqueta `<ul>` cerrada posteriormente con `</ul>`.

Entonces, empezad introduciendo esto:

```
<ul></ul>
```

Y ahora esto es lo que vamos a hacer: vamos a escribir cada elemento de la lista dentro de las etiquetas `<li></li>`. Cada una de estas etiquetas tienen que estar situadas entre `<ul>` y `</ul>`. Vais a entender esto de inmediato con este ejemplo:

```
<ul>

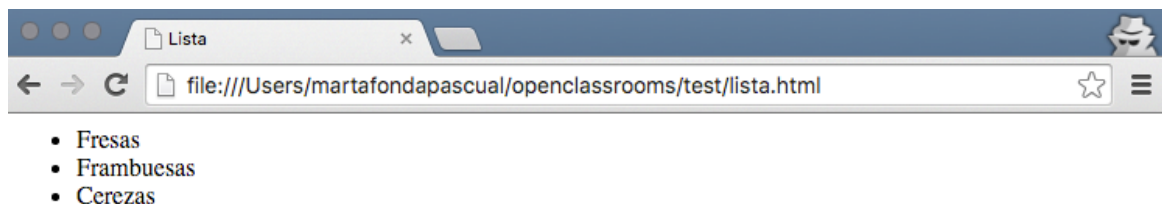
  <li>Fresas</li>

  <li>Frambuesas</li>

  <li>Cerezas</li>

</ul>
```

El resultado se muestra en la figura siguiente.



Una lista no ordenada

Tened en cuenta que la lista no se ha colocado dentro de `<body></body>`. Para mantener el código de la página legible de ahora en adelante no voy a incluir en él todo lo que veamos.

Así que recordad estas dos etiquetas:

- `<ul></ul>` delimita la lista completa;
- `<li></li>` define un elemento de la lista (una viñeta).

Podéis incluir todos los elementos que queráis, no únicamente tres.

¡Así que ahora sabéis cómo crear una lista con viñetas! No es difícil una vez que entendáis cómo anidar las etiquetas.

Para aquellos que tengáis que crear listas complejas, recordad que podéis *anidar* listas no ordenadas (crear una lista no ordenada **dentro** de una lista no ordenada). Si queréis hacer esto abrid una segunda etiqueta `<ul>` y poned **dentro** un elemento `<li></li>`. Si cerráis las etiquetas en el orden correcto no tendréis ningún problema. Sin embargo, tened cuidado porque esta técnica no es fácil dominar.

Lista ordenada

Una lista ordenada funciona de la misma forma únicamente cambia la etiqueta: tenéis que sustituir `<ul></ul>` por `<ol></ol>`.

No cambiéis nada dentro de la lista: siempre tenéis que usar las etiquetas `<li></li>` para delimitar los elementos de la lista.

El orden en el que colocáis los elementos en la lista es importante. El primero `<li></li>` será el elemento n.º 1, el segundo será el elemento n.º 2, etc.

Como es especialmente intuitivo os voy a dejar que admiréis la simplicidad de este ejemplo (los resultados se muestran en la figura de abajo):



Mi día

1. Me levanto.
2. Como y bebo.
3. Me vuelvo a dormir.

Una lista ordenada

Comparado con el ejemplo anterior todo lo que ha cambiado es la etiqueta `<ol>`.

Para vuestra información existe un tercer tipo de lista mucho menos frecuente: la lista de definiciones. Consiste en el uso de las etiquetas `<dl>` para delimitar la lista, `<dt>` para delimitar el término y `<dd>` para delimitar la definición del término.

En resumen

- HTML tiene muchas etiquetas que nos permiten organizar el texto de la página. Estas etiquetas proporcionan indicaciones, como «este es un párrafo», «esto es un título», etc.
- La etiqueta `<p> </p>` define los párrafos y la etiqueta `<br />` el salto de línea.
- Hay seis niveles de títulos, desde `<h1> </h1>` hasta `<h6> </h6>`, los cuales se usan en función de la importancia del título.
- Podéis resaltar determinadas palabras con las etiquetas `<strong>`, `<em>` y `<mark>`.
- Para crear listas, tenéis que usar la etiqueta `<ul>` (lista no ordenada con viñetas) o la etiqueta `<ol>` (lista ordenada). Los elementos se incluyen dentro de la lista con la etiqueta `<li>` por cada elemento.

Crear enlaces

En el capítulo anterior habéis aprendido cómo crear una página HTML sencilla. Vale, en realidad no era para tanto, pero era una página HTML real a pesar de todo.

Como sabréis un sitio web está formado por varias páginas. ¿Cómo vais de una página a otra? ¡Usando enlaces por supuesto! Entonces, la forma en la que se crean los enlaces entre las páginas es lo que vais a aprender en este capítulo.

Supongo que todos sabéis lo que es un enlace: es un texto en el que hacéis clic para ir a otra página.

Podéis crear un enlace para ir de la página `a.html` a la página `b.html`, pero también podéis crear un enlace a otro sitio web, por ejemplo `http://www.google.com`. Veremos que ambos casos funcionan de la misma forma.

Un enlace a otro sitio web

Estos enlaces se reconocen fácilmente en una página: están escritos de una forma diferente (de forma predeterminada en azul y subrayados) y un cursor que se parece a una mano aparece cuando se pone el ratón sobre ellos.

Os sugiero que intentéis codificar un enlace que os lleve a Google como se muestra en la figura siguiente.

Hola. ¿Queréis visitar el sitio web [Google](http://www.google.com)?
¡Es un buen sitio web! ;-)

Enlace a Google

Para insertar un enlace, la etiqueta que vamos a usar es muy fácil de recordar: `<a>`. Sin embargo, necesita un atributo, `href`, que indica la página a la que debería llevar el enlace.

Por ejemplo el código que está a continuación lleva a Google, localizado en la dirección `http://www.google.com`:

```
<a href="http://www.google.com">Google</a>
```

Vamos a incluir este enlace en un párrafo. Entonces, así es cómo reproducís el ejemplo en la figura siguiente:

```
<p>Hola. ¿Queréis visitar el sitio web <a href="http://www.google.com">Google</a>?<br />  
¡Es un buen sitio web! ;-)</p>
```

Por defecto el enlace se resalta en azul. Si ya habéis abierto la página el enlace se muestra en morado.

Veremos cómo cambiar este aspecto cuando estudiemos CSS.

Si queréis crear un enlace a otra página podéis simplemente copiar y pegar su dirección (llamada URL) en `http://`. Tomad en cuenta que algunas veces algunos enlaces empiezan con `https://` (sitios web seguros) u otros prefijos (`ftp://`, etc.).

Si creáis un enlace a un sitio web con una dirección un poco extraña no veréis la diferencia, pero es necesario que creéis una página web correctamente en HTML5.

Los enlaces que hemos visto se denominan **enlaces absolutos** debido a que muestran la dirección completa. Ahora vamos a ver que podemos escribir los enlaces de algún modo diferente, lo que será útil para crear enlaces entre las páginas de nuestro sitio web.

Un enlace a otra página de vuestro sitio web

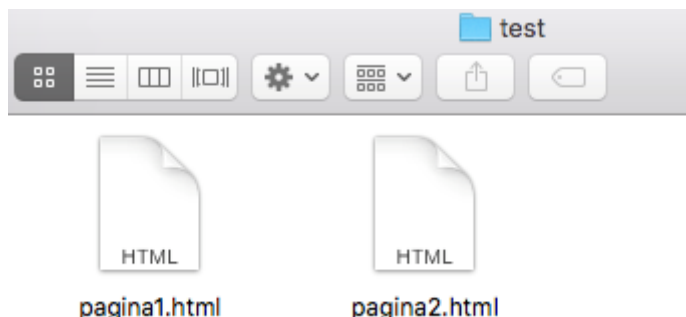
Acabamos de aprender la forma de crear enlaces a sitios web que existen. Pero estoy seguro de que os gustaría crear enlaces entre las páginas de vuestro sitio web, ¿verdad?

Sí, ¿qué tengo que hacer exactamente para crear un enlace a otra página de mi sitio web? ¡No sé su dirección `http://.....` ya que acabo de empezar a crear mi sitio web aquí! No tengo una dirección.

De hecho, de momento habéis creado vuestro sitio web en vuestro ordenador. Sois los únicos que podéis verlo y aún no tiene una «dirección web» que empiece con `http://` como la mayoría de los sitios web. Afortunadamente esto no impedirá que trabajemos.

Dos páginas ubicadas en la misma carpeta

Para empezar vamos a crear dos archivos que correspondan a dos páginas HTML diferentes. Como estoy inspirado os propongo que los llamemos `pagina1.html` y `pagina2.html`. Por tanto, tendremos estos dos archivos en nuestro disco duro *en la misma carpeta* (ver figura a continuación).



Varios archivos HTML en la misma carpeta

¿Cómo hacéis un enlace de la página 1 a la página 2 sin que tengáis una dirección en `http://`? De hecho, si ambos enlaces están ubicados en la misma carpeta es fácil, simplemente escribid el nombre del archivo al que queráis moveos como el enlace de destino. Por ejemplo: `<a href="pagina2.html">`. Esto se denomina **enlace relativo**.

Aquí tenemos el código que vamos a usar en nuestros archivos `pagina1.html` y `pagina2.html`.

pagina1.html

```
<p>Hola. ¿Queréis ver la <a href="pagina2.html">página2</a> ?</p>
```

`pagina2.html`

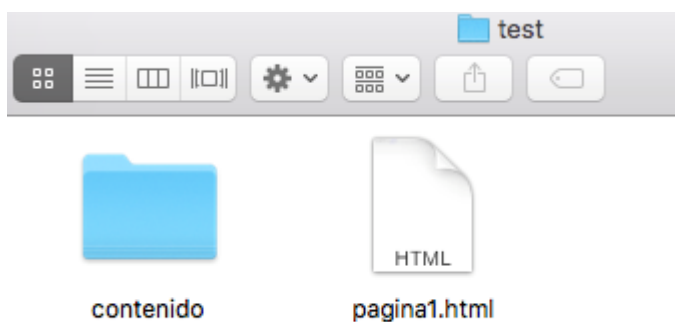
La página 2 (página de llegada) simplemente mostrará un mensaje que indicará que efectivamente habéis llegado a la página 2:

```
<h1>¡Bienvenidos a la página 2 !</h1>
```

Dos páginas ubicadas en carpetas diferentes

Las cosas se ponen un poco más difíciles si las páginas se encuentran en carpetas diferentes. Lo ideal sería que no estuvieran muy lejos las unas de las otras (por ejemplo en una subcarpeta).

Vamos a imaginar que la `pagina2.html` se encuentra en una subcarpeta denominada `contenido` como en la figura siguiente.



El archivo `página2.html` se encuentra en la carpeta `contenido`

En este caso el enlace debería estar escrito de esta forma:

```
<a href="contenido/pagina2.html">
```

Si hay varias carpetas deberíais escribir esto:

```
<a href="contenido/otracarpeta/pagina2.html">
```

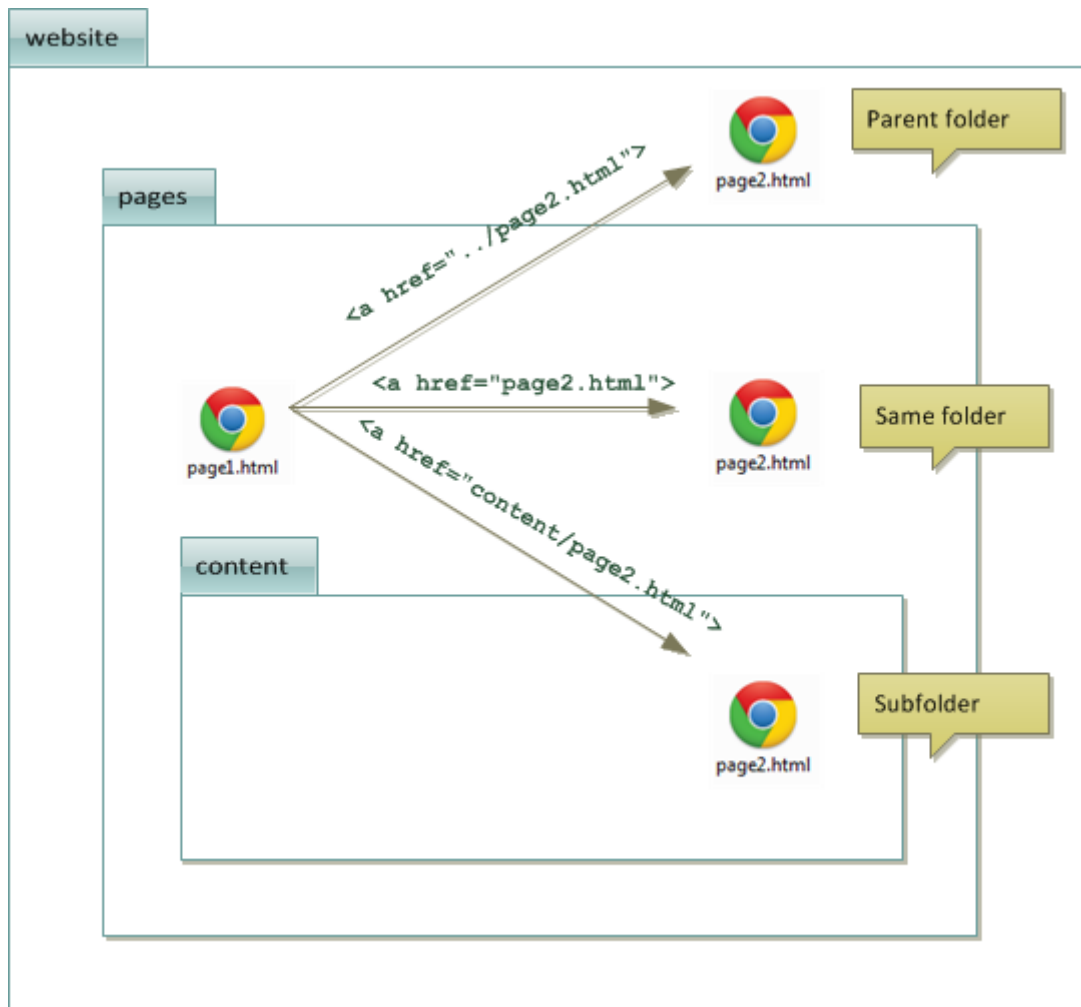
¿Y qué ocurre si el archivo no está ubicado en una subcarpeta sino en una carpeta raíz? ¿Qué hacemos entonces?

Si tu archivo de destino se encuentra en una carpeta ubicada "más arriba" en la estructura de árbol, tienes que escribir dos puntos como estos:

```
<a href="../pagina2.html">
```

Resumen gráfico

Los enlaces relativos no son muy difíciles de utilizar una vez que hayas entendido el procedimiento. Para saber cómo escribir tu enlace, solo tienes que mirar el "nivel de carpeta" en el que se encuentra el archivo de destino. La siguiente imagen resume los diferentes enlaces relativos posibles.



Enlaces relativos

Un enlace a un hipervínculo

Un **hipervínculo** es una especie de punto de referencia que puedes incluir en tus páginas HTML cuando son muy largas.

De hecho, pueden ser útiles para crear un enlace a un punto situado más abajo en la misma página para permitir a los visitantes saltar directamente a la parte que les interesa.

Para crear un hipervínculo nuevo, Puede ser cualquier etiqueta, por ejemplo un título.

Para dar un nombre al hipervínculo, se utiliza el atributo `id`. Entonces podremos usarlo para crear un enlace a este hipervínculo. Por ejemplo:

```
<h2 id="my_anchor">Título</h2>
```

Ahora solo tienes que crear un enlace como siempre, pero esta vez el atributo `href` incluirá una almohadilla (#) seguida del nombre del enlace. Ejemplo:

```
<a href="#my_anchor">Ir al hipervínculo</a>
```

Normalmente, si haces clic en el enlace, te llevará a un punto más abajo de la misma página (siempre que la página contenga texto suficiente para que las barras de desplazamiento se muevan automáticamente).

Este es un ejemplo de página con mucho texto que utiliza hipervínculos (he utilizado texto antiguo para rellenar la página):

```
<h1>>Mi gran página</h1>

<p>

Ir directamente a la parte que trata sobre:<br />

<a href="#cocina">La cocina</a><br />

<a href="#rodillos">Rodillos</a><br />

<a href="#tiro con arco">Tiro con arco</a><br />

</p>

<h2 id="cocina">La cocina</h2>

<p>... (mucho texto) ...</p>

<h2 id="rodillos">Rodillos</h2>

<p>... (mucho texto) ...</p>

<h2 id="tiro con arco">Tiro con arco</h2>

<p>... (mucho texto) ...</p>
```

Si no ocurre nada cuando hagamos clic sobre los enlaces, esto significa que no hay suficiente texto. En este caso, puedes optar por añadir texto antiguo a la página para que tenga (incluso) más texto, o bien reducir el tamaño de la ventana del navegador para que aparezcan las barras de desplazamiento en el lateral.

Para dar un nombre "único" a una etiqueta, se utiliza el atributo `id`. Y créeme, todavía no lo has oído todo sobre este atributo. Aquí se utiliza para crear un enlace a un hipervínculo, pero en CSS,

nos será muy útil para "marcar" una etiqueta específica. Ya lo veremos. Sin embargo, evita crear ids con espacios o caracteres especiales. Siempre que sea posible, utiliza solo letras y número para que el valor sea reconocido por todos los navegadores.

Enlace a un hipervínculo situado en otra página

Ok, estáis avisados. ¡Esto va a ser un auténtico megamix!

La idea es crear un enlace que abra otra página Y que te lleve directamente a un hipervínculo situado más abajo de esta página.

En la práctica es muy fácil hacerlo: escribe el nombre de la página, seguido por una almohadilla (#), seguido del nombre del hipervínculo.

Por ejemplo: `<a href="anchors.html#rodillos">`

... te llevará a la página `anchors.html`, directamente al hipervínculo llamado `rodillos`.

Esta página contiene tres enlaces, cada uno de los cuales lleva a uno de los hipervínculos de la página del ejemplo anterior:

```
<h1>Megamix</h1>

<p>

Referencia a otro lugar de la misma página:<br />

<a href="anchors.html#kitchen">La cocina</a><br />

<a href="anchors.html#rollers">Rodillos</a><br />

<a href="anchors.html#arc">Tiro con arco</a><br />

</p>
```

Ejemplos prácticos del uso de enlaces

Ahora voy a intentar mostraros varios ejemplos prácticos del uso de enlaces. Por ejemplo, ¿sabías que es muy fácil crear enlaces que inician una descarga? ¿Que crean un nuevo correo electrónico? ¿Que abren una ventana nueva?

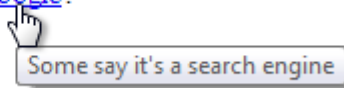
¿No? Bueno, pues vamos a ver todo esto.

Un enlace que muestra una descripción emergente

Puedes utilizar el atributo `title` que muestra una descripción emergente cuando coloques el curso sobre el enlace. Este atributo es opcional

Debes obtener un resultado similar al de la imagen siguiente.

Hello. Do you want to visit [Google](#)?



Una descripción emergente

La descripción emergente puede ser útil para proporcionar información a los visitantes incluso antes de que hayan hecho clic sobre el enlace. Aquí tenemos la forma de reproducir este resultado:

```
<p>Hola. ¿Queréis visitar <a href="http://www.google.com" title="Dicen que es un motor de búsqueda">Google</a> ?</p>
```

Un enlace que abre una ventana nueva

Un enlace se puede "obligar" a abrir una nueva ventana. Para hacer esto, se añade `target="_blank"` a la etiqueta `<a>`.

```
<p>Hola. ¿Queréis visitar <a href="http://www.google.com" target="_blank">Google</a> ?<br />
```

Este sitio web se mostrará en otra ventana.</p>

Dependiendo de la configuración del navegador, la página se abrirá en una ventana nueva o en una pestaña nueva. No puedes elegir entre que se abra en una ventana nueva o en una pestaña nueva.

Sin embargo, no se recomienda abusar de esta técnica porque afecta a la navegación. Los visitantes pueden decidir si quieren abrir el enlace en una ventana nueva. Pueden pulsar **Shift** + Clic sobre el enlace para abrirlo en una ventana nueva o **Comando** + Clic para abrirlo en una pestaña nueva.

Un enlace para enviar un correo electrónico

Si quieres que tus visitantes puedan enviarte un correo electrónico, puedes usar enlaces del tipo `mailto`. A nivel de etiqueta no cambia nada. Solo cambias el valor del atributo `href` así:

```
<p><a href="mailto:nombre@email.com">¡Envíame un correo electrónico!</a></p>
```

Así que solo tienes que empezar el enlace con `mailto:` e introducir la dirección de correo electrónico a la que quieres que te escriban. Si haces clic en el enlace, se abre un nuevo mensaje en blanco, listo para ser enviado a tu dirección de correo electrónico.

Un enlace para descargar un archivo

Muchos de vosotros os estaréis preguntando cómo funciona esto para descargar un archivo... En realidad, el procedimiento es exactamente el mismo que para enlazar a una página web, pero esta vez tenemos que indicar el nombre del archivo que queremos que se descargue.

Por ejemplo, imaginad que queréis descargar `myfile.zip`. Solo tenéis que colocar este archivo en la misma carpeta que vuestra página web (o en una subcarpeta) y crear un enlace a este archivo:

```
<p><a href="myfile.zip">Descargar el archivo</a></p>
```

¡Eso es todo! Cuando el navegador vea que no hay ninguna página web que mostrar, comenzará el proceso de descarga cuando el usuario haga clic sobre el enlace.

En resumen

- Los enlaces sirven para moverse de una página a otra y por defecto son texto subrayado de color azul.
- Para introducir un enlace, utiliza la etiqueta `<a>` con el atributo `href` para indicar la dirección de la página de destino. Ejemplo: `<a href="http://www.google.com">`.
- Puedes crear un enlace a otra página de tu sitio web con tan solo escribir el nombre del archivo: `<a href="pagina2.html">`.
- Los enlaces también se pueden usar para saltar a otras secciones de la misma página. Tienes que crear un hipervínculo con el atributo `id` para "marcar" una sección de la página y después crear un enlace al hipervínculo como este: `<a href="#anchor">`.

Imágenes

¿Insertar una imagen en el sitio web? Ya veréis que no es nada complicado... Bueno, casi nada. En los sitios web se pueden utilizar varios **formatos** de imágenes diferentes y no debemos elegirlos al azar. De hecho, a veces las imágenes son grandes y tardan mucho en descargarse, lo que ralentiza el tiempo de carga de la página (¡mucho más que el texto!).

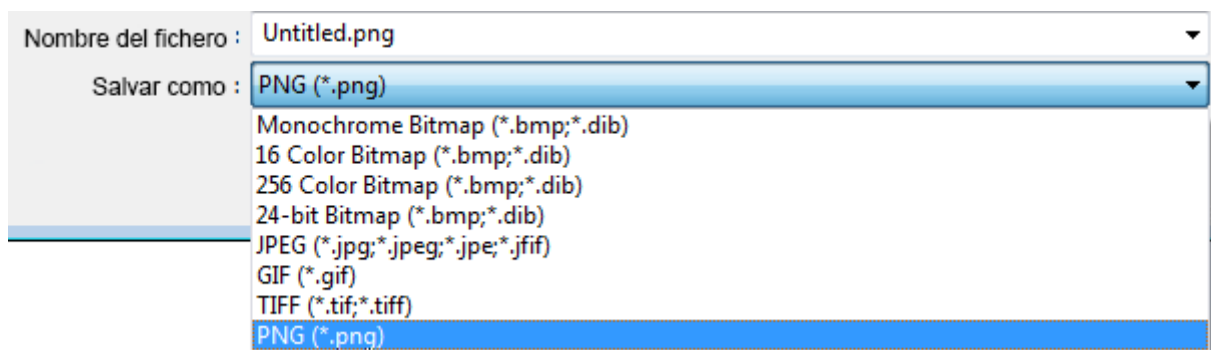
Entonces, para asegurarte de que tus páginas son legibles y rápidas de descargar, sigue mi consejo.

Formatos de imagen

¿Sabes qué es el formato de imagen?

Cuando tienes "acceso" a una imagen, puedes guardarla en varios "formatos" diferentes. El tamaño de la imagen (en KB o incluso MB) variará dependiendo del formato seleccionado y la calidad de la imagen cambiará.

Por ejemplo, el programa de dibujo Paint (aunque por supuesto no sea el mejor) te permite elegir entre diferentes formatos cuando guardas una imagen (imagen siguiente).



Formatos de imagen que ofrece Paint

Algunos formatos son mas adecuados que otros según la imagen (foto, dibujo, imagen en movimiento, etc.). Nuestro objetivo es repasar los formatos utilizados en la Red para que tengáis una idea de ellos y sepáis elegir el más adecuado para vuestra imagen. No os preocupéis. No existen muchos formatos diferentes, así que no tardaremos mucho.

Todas las imágenes publicadas en Internet tienen un punto en común: están **comprimidas**. Esto significa que el ordenador hace cálculos para ampliarlas y que se carguen más rápido.

JPEG

Las imágenes en formato JPEG (*Joint Photographic Expert Group*) son muy comunes en la Red. Este formato está diseñado para reducir el tamaño de almacenamiento de las fotos (en otras palabras, el tamaño del archivo asociado), que puede contener más de 16 millones de colores diferentes. La siguiente ilustración es una imagen guardada en formato JPEG.



La foto de una montaña en JPEG

Las imágenes JPEG se guardan con la extensión `.jpg` o `.jpeg`.

Tened en cuenta que el formato JPEG reduce la calidad de la imagen ligeramente, de un modo generalmente imperceptible. Esto es lo que le hace tan efectivo en la reducción del tamaño de almacenamiento de las fotos.

Cuando se trata de una foto, normalmente no se detecta la pérdida de calidad. Si no es una foto, la imagen puede parecer que está un poco borrosa. En este caso, es mejor usar el formato PNG.

PNG

PNG (*Portable Network Graphics*) es el formato más nuevo de todos. Este formato es adecuado para la mayoría de gráficos (me atrevería a decir que para "todo lo que no sea una foto"). El formato PNG tiene dos grandes ventajas: puede ser transparente y no afectar a la calidad de la imagen.

El formato PNG se inventó para competir con otro formato, GIF, en tiempos en los que había que pagar derechos de autor por el uso de GIF. Desde entonces, el formato PNG ha cambiado mucho y se ha convertido en el formato más potente para guardar la mayoría de las imágenes.

PNG está disponible en dos versiones, dependiendo del número de colores que incluya la imagen:

- PNG de 8 bits: 256 colores;
- PNG de 24 bits: 16 millones de colores (como una imagen JPEG).

La siguiente ilustración es una imagen PNG de 24 bits, que representa a nuestra famosa mascota Zozor.



Zozor en formato PNG

La versión antigua de Internet Explorer (IE6) no muestra correctamente imágenes PNG de 24 bits transparentes. Aunque este navegador cada vez se utiliza menos, hay que tener en cuenta esta información.

De hecho, si una imagen PNG de 24 bits puede mostrar el mismo número de colores que una imagen JPEG, y también puede hacerse más transparente sin afectar a la calidad de la imagen... ¿para qué queremos el formato JPEG?

La compresión de fotos es mas potente con JPEG. Una foto guardada en JPEG se cargará siempre más rápido que si se guarda en PNG. Por eso os recomiendo que siempre mantengáis el formato JPEG en vuestras fotos.

GIF

Aunque este formato es bastante antiguo, se ha utilizado mucho (y todavía se usa). El formato PNG es mucho mejor que el GIF en general: normalmente las imágenes tienen un tamaño de almacenamiento mucho menor y su transparencia es de mejor calidad. Por eso recomiendo que uséis PNG todo lo posible.

GIF está limitado a 256 colores (mientras que PNG puede llegar hasta varios millones de colores).

Sin embargo, el formato GIF tiene una ventaja que no tiene el PNG: puede ser animado (ver ilustración siguiente).



Una imagen GIF animada

Hay un formato apropiado para cada imagen

Para resumir, el formato se debe escoger dependiendo de la imagen que tengas, tal y como se indica a continuación:

- **Una foto:** usar JPEG.
- **Cualquier gráfico que no tenga muchos colores** (menos de 256): usar PNG de 8 bits o posiblemente GIF.
- **Cualquier gráfico con muchos colores** (menos de 256): usar formato PNG de 24 bits.
- **Una imagen animada:** usar GIF animado.

Errores a evitar

Evitar otros formatos

Otros formatos que no se nombran aquí, tales como el formato BITMAP (*.bmp), deben evitarse porque normalmente no están comprimidos y, por tanto, son demasiado grandes. No todos son apropiados para la Red. Aunque puedes incluirlos en tu sitio web, tardarán *muchísimo* tiempo en cargarse.

Elige bien el nombre de tu imagen

Si quieres evitarte problemas, acostúmbrate a guardar tus archivos con nombres en minúsculas, sin espacios ni tildes, por ejemplo: `my_image.png`.

Puedes sustituir los espacios por un *guión bajo* (" _ ") como he hecho yo aquí.

Insertar una imagen

Ahora volvamos al código HTML para descubrir cómo incluir imágenes en nuestras páginas web.

Insertar una imagen

¿Cuál es la famosa etiqueta que utilizamos para insertar una imagen? ¡Es... `<img />`!

Es una etiqueta **huérfana** (como `<br />`). Esto quiere decir que no tienes que introducir dos etiquetas como en el caso de la mayoría de etiquetas que hemos visto hasta ahora. De hecho, no tenemos que definir una porción de texto; lo único que queremos es colocar una imagen en un punto específico.

La etiqueta debe ir acompañada de dos atributos obligatorios:

- `src`: que indica la ubicación en la que quieres insertar la imagen. Puedes incluir una ruta absoluta (e.g. `http://www.website.com/flor.png`) o una ruta relativa (que es lo que suele hacerse). Así que si la imagen está en la subcarpeta `imagenes`, tienes que introducir: `src="imagenes/flor.png"`
- `alt`: que significa "texto alternativo". Un texto alternativo a la imagen, en otras palabras, un texto corto que describe lo que contiene la imagen, debe indicarse *siempre*. Este texto se mostrará en lugar de la imagen si esta última no se puede descargar (esto ocurre), o en los navegadores de personas con discapacidad visual que lamentablemente no pueden "ver" la imagen. También ayuda a los robots de motores de búsqueda a buscar imágenes. Para la flor, por ejemplo, se debería escribir lo siguiente: `alt="Una flor" ..`

Las imágenes deben colocarse obligatoriamente dentro de un párrafo (`<p></p>`). Aquí tenéis un ejemplo de imagen insertada:

```
<p>

  Esta es una foto que hice durante mis pasadas vacaciones en las montañas:<br />

</p>
```

Resumiendo, es muy fácil insertar una imagen siempre y cuando puedas indicar donde está ubicada, como aprendimos a hacer con los enlaces.

La mayor "dificultad" (si es que podemos llamarlo dificultad) es seleccionar el formato de imagen adecuado. En este caso es una foto, por lo que obviamente se utiliza el formato JPEG.

Lo diré de nuevo: debemos evitar las letras mayúsculas y los espacios a toda costa en los nombres de archivos y carpetas. Esta ruta causaría problemas:

`"Website imagenes/Mi súper imagen.jpg"`.

Los espacios se deben borrar (o reemplazados por el símbolo "_") y todo debe ir en letras minúsculas como aquí:

`"website_imagenes/mi_super_imagen.jpg"`.

No lo escribas mal, porque si la imagen no se muestra, es casi seguro porque la ruta es incorrecta. Elige los nombres más simples posibles para los nombres de tus carpetas y archivos, y todo irá bien.

Añadir una descripción emergente

El atributo utilizado para mostrar una descripción emergente es el mismo que para los enlaces: `title`. Este atributo es opcional (al contrario que `alt`).

Aquí tenemos un ejemplo:

```
<p>

Esta es una foto que hice durante mis pasadas vacaciones en las montañas:<br />



</p>
```

Coloca el puntero del ratón sobre la foto para que aparezca la descripción emergente.

Miniatura clicable

Si la imagen es muy grande, es recomendable mostrar su vista en miniatura en tu sitio web. Después añade un enlace a la vista en miniatura para que los visitantes puedan ver la imagen en su tamaño original.

Existen millones de programas para crear miniaturas de imágenes. Yo personalmente utilizo [Easy Thumbnails](#). Por tanto, tendré dos versiones de mi foto (como se muestra en la siguiente ilustración): la vista en miniatura y la imagen original.



Meto las dos en una carpeta llamada, por ejemplo, `img`. Muestro la versión `monte_peque.jpg` en mi página y creo un enlace a `monte.jpg` para que se muestre la versión ampliada al hacer clic sobre la vista en miniatura.

Aquí está el código HTML que utilizo para esto:

```
<p>

¿Quieres ver la imagen en su tamaño original? ¡Haz clic sobre ella!<br />

<a href="img/monte.jpg"></a>

</p>
```

A veces, algunos buscadores muestran un marco azul (o morado) muy poco atractivo alrededor de la imagen clicable.
Afortunadamente, podemos quitar rápidamente este marco con CSS.

Figuras

Al leer este libro, ya habrás visto varios ejemplos de **figuras**. Son elementos que realzan el texto para complementar la información de la página.

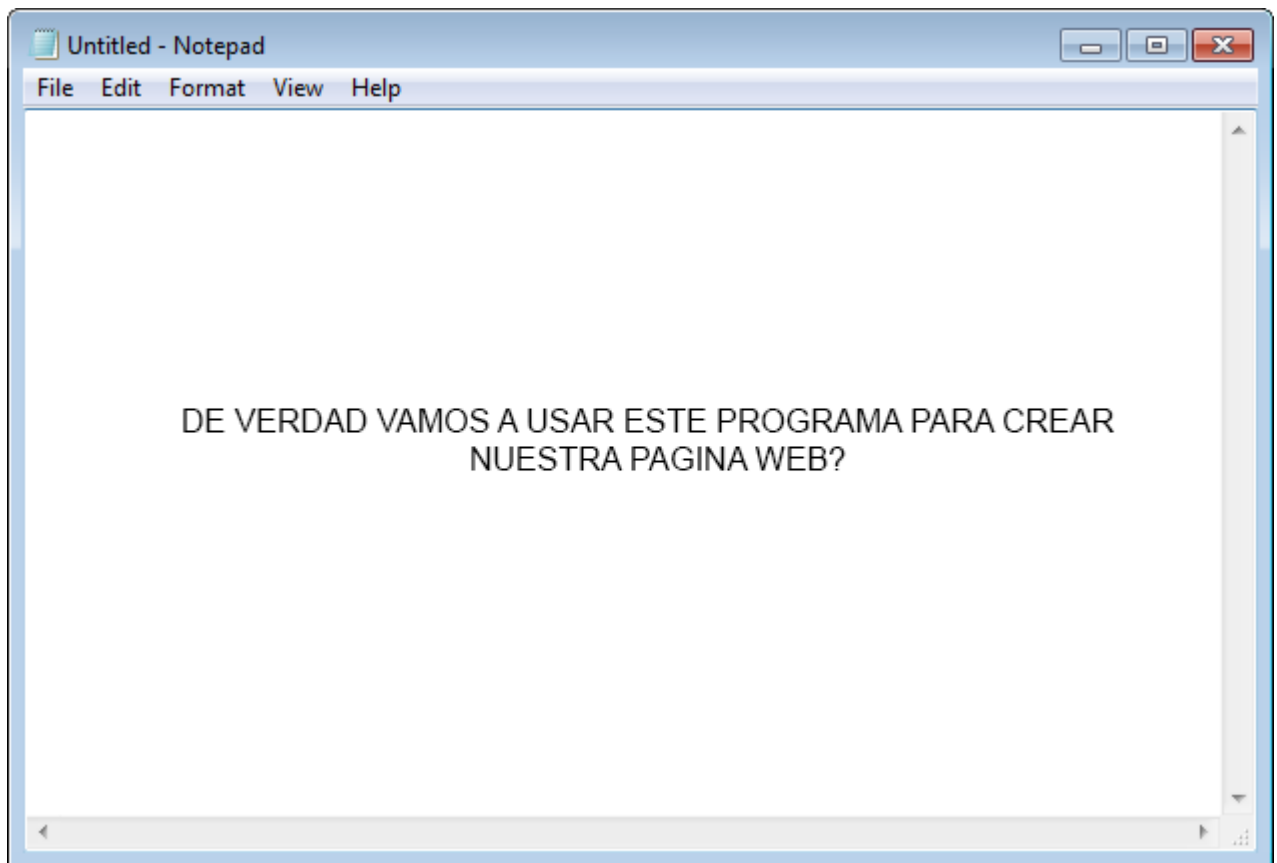
Las figuras pueden ser de diferentes tipos:

- imágenes
- código fuente;
- comillas;
- ...

Resumiendo, cualquier cosa que *ilustre* el texto es una figura. Aunque en este caso estamos usando imágenes, las figuras, al contrario de lo que muchos estaréis pensando, no tienen por qué ser imágenes *necesariamente*: un código fuente también ilustra el texto.

Crear una figura

Tomemos de nuevo el ejemplo de la captura de pantalla que vimos en el primer capítulo, que se muestra en la siguiente ilustración:



El programa Notepad

En HTML5, la etiqueta usada es `<figure>`. Aquí tenemos la forma de usarla:

```
<figure>



</figure>
```

Una figura suele ir acompañada la mayoría de las veces de una leyenda. Para añadir una leyenda, utiliza la etiqueta `<figcaption>` dentro de la etiqueta `<figure>`, así:

```
<figure>



<figcaption>El programa Notepad</figcaption>

</figure>
```

Entender la finalidad de las figuras

Anteriormente en este capítulo, os dije que las imágenes tenían que ir colocadas en párrafos (situadas dentro de una etiqueta `<p></p>`). No es totalmente cierto.

Si realizas una figura de tu imagen, la imagen se puede colocar fuera del párrafo.

```
<p>¿Conocéis el programa Notepad? ¡Se pueden hacer páginas web con él!</p>

<figure>



<figcaption>El programa Notepad</figcaption>

</figure>
```

No veo la diferencia. ¿Cuándo debo poner la imagen en un párrafo y cuándo debo ponerla en una figura?

¡Buena pregunta! Todo depende de lo que tu imagen proporcione al texto:

- Si no proporciona información (es simplemente una ilustración con fines decorativos): coloca la imagen en un párrafo.
- Si proporciona información: coloca la imagen en una figura.

La etiqueta `<figure>` tiene un papel principalmente **semántico**. Esto significa que le dice al ordenador que la imagen tiene un significado y que es importante para la comprensión del texto. Esto puede permitir a un programa recuperar todas las figuras del texto mencionarlas en una tabla de figuras, por ejemplo.

Por último, ten en cuenta que una figura puede tener varias imágenes. Este es un caso que lo justifica:

```
<figure>







<figcaption>Logos de varios navegadores</figcaption>

</figure>
```

En resumen

- Existen varios formatos de imagen apropiados para su uso en la Red:
 - JPEG: para fotos;
 - PNG: para todas las demás ilustraciones;

- GIF: similar a PNG, con un número de colores más limitado, pero puede ser animado.
- Las imágenes se insertan con la etiqueta `<img />`. Debe incluir al menos dos atributos: `src`(nombre de la imagen) y `alt` (descripción corta de la imagen).
- Si una imagen ilustra el texto (y no tiene una finalidad simplemente decorativa), es mejor ponerla dentro de una etiqueta `<figure>`. La etiqueta `<figcaption>` se utiliza para introducir la leyenda de la imagen.

Quiz 1

• Pregunta 1

¿Cuál es la función del navegador web?

- ☐ Escribir código HTML y CSS
- ☐ Convertir el HTML y el CSS en un resultado gráfico.
- ☐ Conectar a una red Wifi
- ☐ Proveer el website a los visitantes

• Pregunta 2

¿En código HTML, cuál de estos elementos va primero?

- ☐ `<html>`
- ☐ `<head>`
- ☐ `<!--BEGIN-->`
- ☐ `<!DOCTYPE html>`

• Pregunta 3

¿Qué lenguaje necesitas para decir "*Mi menú de navegación está arriba a la derecha*"?

- ☐ HTML
- ☐ CSS
- ☐ JavaScript
- ☐ PHP

• Pregunta 4

Una de las siguientes etiquetas se refiere al título principal de tu website. Es muy importante para los buscadores. ¿Qué etiqueta es?

- ☐ `<title>`
- ☐ `<meta>`
- ☐ `<h1>`
- ☐ `<h2>`

• Pregunta 5

¿Cuál de estas etiquetas indica el contenido de mayor importancia?

- ☐ `<strong>`
- ☐ `<p>`
- ☐ `<li>`
- ☐ `<em>`

• **Pregunta 6**

¿Cuál es el formato recomendado para almacenar fotos?

- ☐ PNG
- ☐ JPG
- ☐ GIF
- ☐ BMP

• **Pregunta 7**

¿Por qué se requiere el atributo "alt" para imágenes?

- ☐ Porque especifica el tamaño de la imagen
- ☐ Porque describe el contenido de la imagen para personas con visibilidad reducida y para buscadores
- ☐ El atributo "alt" no se requiere, sólo se requiere el atributo "src"

• **Pregunta 8**

¿Cómo consigues que un usuario de tu web se descargue un archivo mediante código HTML?

- ☐ creando un link al archivo con la etiqueta `<a>`
- ☐ Utilizando la etiqueta `<download>`
- ☐ Incluyendo el contenido del archivo directamente en el código HTML
- ☐ No puedes hacer esto con código HTML

• **Pregunta 9**

¿Qué etiqueta se utiliza para crear una lista numerada?

- ☐ `<dl>`
- ☐ `<ol>`
- ☐ `<li>`
- ☐ `<ul>`

• **Pregunta 10**

¿Qué link permite a un visitante ir a un lugar específico que está en la misma web y cuyo "id" es "details" ?

- ☐ `<a href=".details">`
- ☐ `<a href="details">`
- ☐ `<a href="/details">`
- ☐ `<a href="#details">`

html5es-part1 Assessment

¡Es hora de practicar! En este ejercicio, vas a crear tu primera página HTML basándote en lo que has aprendido en la primera parte de este curso. Tu misión consiste en crear tu propio CV online. No tiene por qué quedar perfecto, ya que seguiremos construyendo sobre él más tarde.

Esto es lo que necesitas incluir en tu página web:

- Tu nombre, como título principal
- Debajo de tu nombre, una frase breve que te describa, o una frase célebre
- Una foto pequeña, que mostrará una versión más grande cuando pinchemos sobre ella
- 3 secciones con los siguientes títulos:
 - Mi experiencia
 - Mis habilidades
 - Mi educación

No olvides utilizar las etiquetas para los títulos y párrafos, así como las listas de elementos allí donde sea necesario. También necesitarás incluir imágenes que hayas utilizado en tu archivo .html , y enviarlas junto con el ZIP.

Nota: Si no deseas mostrar tu nombre y foto real, no tienes por qué hacerlo. Puedes hacer un CV de muestra. Ten en cuenta que tendemos a poner más interés en ello cuando hablamos de nosotros mismos ;0)

Configurar CSS

Habiendo pasado la primera parte del curso trabajando solo en HTML, ahora vamos a centrarnos en CSS que he querido dejar a un lado de forma deliberada. CSS no es más difícil que HTML. Complementa a HTML para ayudarte a dar formato a tu sitio web.

El primer capítulo de CSS trata la teoría que hay detrás de él: ¿Qué es? ¿Cómo es? ¿Dónde se escribe el código CSS? Aunque estos aspectos teóricos no son muy complicados, es necesario que los conozcas porque son la base del CSS. Y es lo único que os voy a pedir que os aprendáis de memoria del CSS, ya que el resto podéis encontrarlo en el apéndice.

La teoría que hay detrás del CSS

Os advertí sobre esto justo al comienzo del curso: vamos a aprender dos lenguajes. Ya hemos descubierto muchas cosas sobre HTML, aunque todavía quedan algunas cosillas que aprender (volveremos a ello dentro de algunos capítulos). Sin embargo, ha llegado la hora de centrarnos en el lenguaje CSS.

CSS (*Cascading Style Sheets*) es otro lenguaje que complementa al lenguaje HTML.
¿Recuerdas cuál era su finalidad? *Gestionar el formato de tu sitio web.*

Breve recordatorio: ¿Para qué se utiliza CSS?

¿CSS? Es lo que te permite elegir el color de tu texto.
También te permite seleccionar la fuente usada en tu sitio web.
Y también te permite establecer el tamaño de texto, los bordes, el fondo, etc.

Y también es lo que utilizas para el diseño de tu página web. Puedes usarlo para decir: Quiero que el menú aparezca a la izquierda y que sea así de ancho. Quiero que la cabecera de mi página web esté alineada en la parte superior y esté siempre visible. Etc.

Recordemos la comparación que vimos en el primer capítulo (figura siguiente).

HTML
(no CSS)



HTML + CSS



Con y sin CSS

Usábamos HTML para escribir el contenido de nuestro sitio web pero era muy irregular por los bordes. CSS complementa este código para dar formato a todo el contenido y darle la apariencia que deseas.

CSS: un comienzo dudoso

Cabe destacar que el lenguaje CSS no existía en el momento en que nació la Red. De hecho solo existía HTML.

HTML fue creado en 1991 y CSS en 1996. Así que probablemente os estaréis preguntando: ¿Cómo daban formato al texto desde 1991 hasta 1996? Bueno, en realidad, solo con HTML. De hecho, había etiquetas HTML para dar formato. `<font color="#aab1c3">`, por ejemplo, se utilizaba para definir el color del texto.

Sin embargo, las páginas de HTML empezaron a hacerse muy complejas. Había cada vez más etiquetas de contenido y forma, lo que hacía que cada vez fuese más difícil actualizar las páginas web. Esa es la razón por la que se creó CSS.

CSS no fue adoptado de forma inmediata por los administradores de sitios web, en absoluto. Había que acabar con una serie de malos hábitos y eso lleva su tiempo. Incluso ahora, todavía podemos encontrar sitios web con etiquetas de formato HTML antiguas y obsoletas, como por ejemplo `<font>`.

CSS: soporte para navegadores

CSS ha cambiado, igual que HTML. Os he dicho que existían cuatro versiones principales de CSS:

- CSS 1;
- CSS 2.0;
- CSS 2.1;
- CSS 3.

De hecho, la versión CSS 3 no está totalmente acabada (todavía no es una versión oficial). De todos modos, se puede usar ya porque está muy avanzada y ya la soportan muchos navegadores.

Sería una lástima perdérsela porque CSS 3 añade muchas funciones a CSS (¡la mitad si lo comparamos con CSS 2.1!). Por tanto, en este curso usaremos CSS 3, que incorpora y complementa la mayoría de las funciones de CSS 2.1.

El trabajo más complejo lo realizan los navegadores web: tienen que *leer* el código CSS y *comprender* cómo mostrar la página.

Aunque Internet Explorer era el navegador más común a principios del 2000, la forma en que gestionó el lenguaje CSS fue mediocre (si no rotundamente mala) durante mucho tiempo. Esto sucedió en el apogeo de la versión 6 (IE6), que desgraciadamente sigue siendo utilizada hoy en día por una pequeña proporción de usuarios (afortunadamente, esta proporción está disminuyendo).

Desde entonces, muchos navegadores han salido a la palestra y se lo han puesto difícil a Internet Explorer: Mozilla Firefox por supuesto, pero también Google Chrome. Por no mencionar el éxito de los Mac y los teléfonos iPhone con su navegador Safari. Esto dio lugar a que Microsoft respondiera y lanzara (tras un largo periodo de inactividad) IE 7, IE 8 y después IE 9. Ya se está hablando de IE 10.

Bueno, tu lección de historia es bonita, ¿pero por qué ha de interesarme a mí hoy?

¿Qué podemos aprender de todo esto? Que los navegadores no conocen todas las propiedades de CSS existentes. Cuanto más viejo es el navegador, menos sabe de las funciones de CSS.

Voy a mostraros en este curso un número de funciones de CSS que no tienen por qué funcionar en navegadores viejos. No puedo evitarlo porque es así: *ningún navegador tiene un conocimiento completo de todas las funciones de CSS* de ninguna manera. Si el navegador no conoce una propiedad CSS, lo peor que puede pasar es que lo ignore y no de el formato, pero la página no se quedará colgada: por tanto, seguirá siendo legible.

Os recomiendo encarecidamente que añadáis a marcadores www.caniuse.com y normansblog.de, donde se muestran tablas de compatibilidad entre funciones de HTML y CSS en varios navegadores (y sus distintas versiones). En concreto, echad un vistazo a [las tablas de compatibilidad de CSS](#) de www.caniuse.com (siguiente ilustración).

CSS position:fixed - Recommendation									
Method of keeping an element in a fixed location regardless of scroll position									
Resources: Workaround for IE6 Article on mobile support									
Show all versions	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Opera Mobile	Android Browser
		3.6						10.0	2.1
	6.0	9.0				3.2		11.0	2.2
	7.0	10.0	17.0			4.0-4.1		11.1	2.3
	8.0	11.0	18.0	5.0		4.2-4.3		11.5	3.0
Current	9.0	12.0	19.0	5.1	11.6	5.0	5.0-6.0	12.0	4.0
Near future	10.0	13.0	20.0	5.2	12.0				
Farther future		14.0	21.0						
Note: Only works in Android 2.2+ by using the following meta tag: <meta name="viewport" content="width=device-width, user-scalable=no">									
Feedback									

Tabla de compatibilidad de CSS de caniuse.com

¿Dónde se escribe el código CSS?

Puedes elegir, ya que el código en lenguaje CSS se puede escribir en tres lugares diferentes:

- En un archivo con la extensión `.css` (es el método más recomendado);
- en la cabecera `<head>` del archivo HTML;
- directamente en las etiquetas del archivo HTML a través de un atributo `style` (es el método menos recomendado).

Aunque voy a describir los tres métodos, os puedo asegurar que el primero... es el mejor.

En un archivo `.css` (recomendado)

Como acabo de decir, lo más frecuente es escribir el código CSS en un archivo con la extensión `.css` (a diferencia de los archivos HTML que tienen extensión `.html`). Este es el método más cómodo y flexible. Evita mezclarlo todo en un solo archivo. Voy a utilizar esta técnica en lo que queda de curso.

¡Empecemos ahora mismo! Vamos a utilizar el siguiente archivo HTML:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<link rel="stylesheet" href="style.css" />

<title>Primeras pruebas de CSS</title>

</head>
```

```
<body>

<h1>Mi gran sitio web</h1>


<p>¡Hola y bienvenidos a mi sitio web!</p>

<p>Mi sitio web todavía está un poco <em>vacío</em>. ¡Pero espera un poco!</p>

</body>

</html>
```

Fijaos en el contenido de la línea 5, `<link rel="stylesheet" href="style.css" />`: esta es la línea que indica que este archivo HTML está asociado con un archivo llamado `style.css` que se encarga del formato.

Guardad este archivo con el nombre que queráis (por ejemplo, `page.html`). Hasta ahora, no hemos hecho nada especial excepto la nueva etiqueta que hemos añadido.

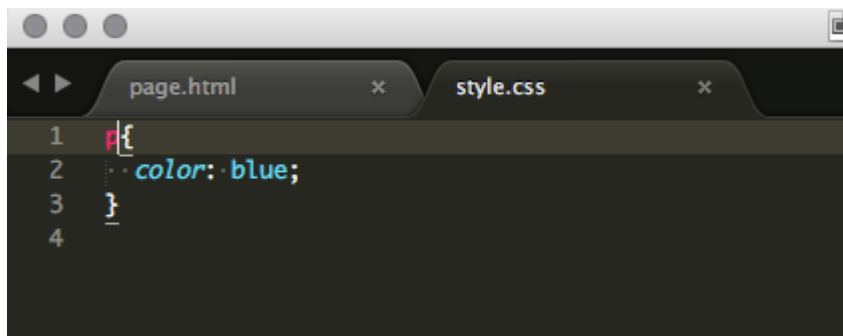
Ahora cread un archivo *nuevo* vacío en vuestro editor de texto (por ejemplo, Notepad++) y copiad este trozo de código CSS (no os preocupéis, os explicaré lo que significa más adelante):

```
p
{
  color: blue;
}
```

Para activar el color en Notepad++, seleccionad el menú `Language > C > CSS`.

Guardar el archivo con un nombre que acabe con `.css`, como por ejemplo `style.css`. Guardar este archivo `.css` en la misma carpeta que vuestro archivo `.html`.

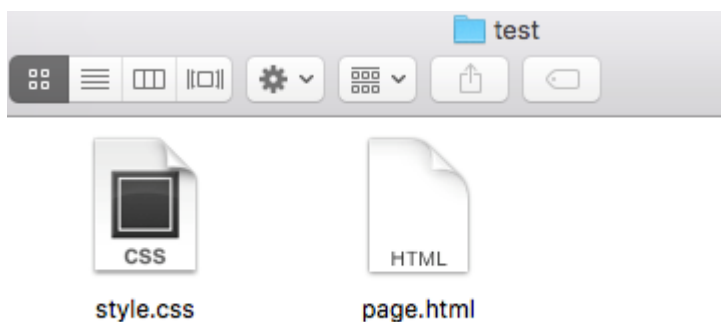
En Sublime, debéis obtener un resultado similar al de la imagen siguiente.



Archivos HTML y CSS

en Sublime Text

En el explorador de archivos, lo veréis mostrador un al lado del otro. El archivo `.html` a un lado y el archivo `.css` al otro lado, como se muestra en la ilustración.



Archivos HTML y CSS en el finder

Ahora abrid vuestro archivo de página `.html` en el navegador para probarlo, como siempre. Mirad, es mágico: los párrafos se muestran en azul, como en la siguiente ilustración.



Mi gran sitio web

¡Hola y bienvenidos a mi sitio web!

Mi sitio web todavía está un poco *vacío*. ¡Pero espera un poco!

El texto está escrito en

azul

No es necesario abrir el archivo `style.css` directamente en el navegador. Solo tenéis que abrir el archivo `page.html` (que llamará automáticamente al archivo `style.css`).

En la cabecera `<head>` del archivo HTML

Existe otra forma de utilizar CSS en los archivos HTML: esto implica insertar el código CSS directamente en una etiqueta `<style>` dentro de la cabecera `<head>`.

Así es como puedes obtener exactamente el mismo resultado con un solo archivo .html que contenga el código CSS (líneas 5-10):

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<style>

p

{

color: blue;

}

</style>

<title>Primeras pruebas de CSS</title>

</head>

<body>

<h1>Mi gran sitio web</h1>

<p>¡Hola y bienvenidos a mi sitio web!</p>

<p>Mi sitio web todavía está un poco <em>vacío</em>. ¡Pero espera un poco!</p>

</body>

</html>
```

Probadlo y veréis que el resultado es el mismo.

Directamente en las etiquetas (no recomendado)

Con el último método hay que tener cuidado: se puede añadir un atributo `style` a cualquier etiqueta. Hay que insertar el código CSS directamente en este atributo:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Primeras pruebas de CSS</title>

</head>


<body>

<h1>Mi gran sitio web</h1>

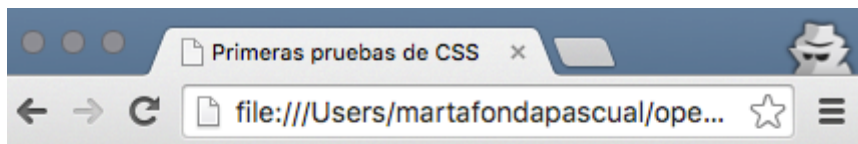

<p style="color: blue;">¡Hola y bienvenidos a mi sitio web!</p>

<p>Mi sitio web todavía está un poco <em>vacío</em>. ¡Pero espera un poco!</p>

</body>

</html>
```

Esta vez, solo el texto del primer párrafo (línea 11), que contiene el código CSS, aparecerá en color azul (siguiente ilustración).



Mi gran sitio web

¡Hola y bienvenidos a mi sitio web!

Mi sitio web todavía está un poco *vacío*. ¡Pero espera un poco!

escrito en azul

El primer párrafo está

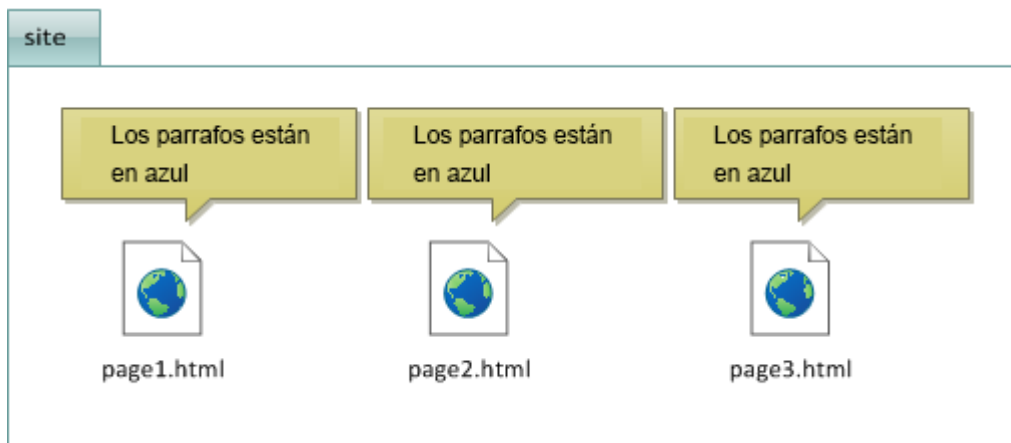
¿Qué método debes elegir?

Creo que el primer método que nos has recomendado es más complicado que los otros dos. ¿Por qué nos recomiendas crear dos archivos? ¡Me iba bien con tan solo un archivo .html!

Os recomiendo encarecidamente que os acostumbréis a trabajar con el primer método porque es el que utilizan la mayoría de administradores de sitios web... ¿Por qué?

De momento, vais a realizar las pruebas en un solo archivo HTML. Sin embargo, más tarde vuestro sitio web incluirá varias páginas HTML, ¿verdad?

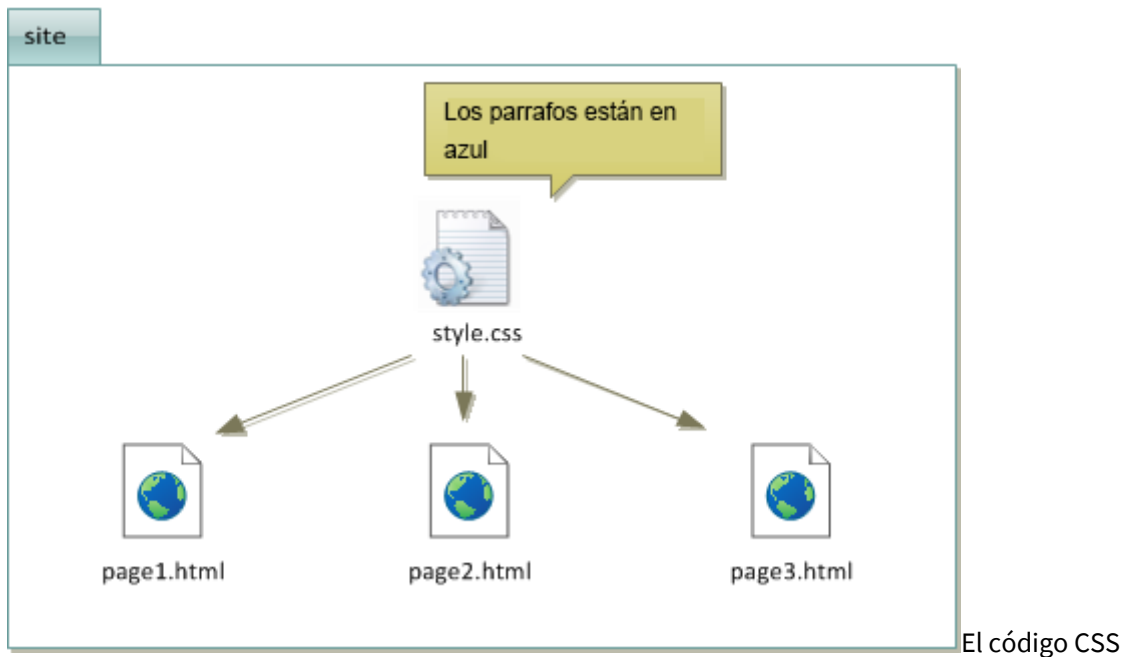
Imaginad esto: si ponéis el código CSS directamente en el archivo HTML, tendréis que copiar este código en todos los archivos HTML del sitio web. Y si mañana cambiáis de opinión y, por ejemplo, queréis los párrafos en color rojo en vez de azul, tendréis que cambiar todos los archivos HTML uno por uno, como se muestra en la siguiente ilustración.



El código

CSS se repite en todos los archivos HTML

Si trabajáis con un archivo CSS externo, solo tendréis que escribirlo una vez para toda la página web, como se muestra en la siguiente ilustración.



se escribe una sola vez en un archivo CSS

Aplicar un estilo: seleccionar una etiqueta

Ahora que ya sabemos donde poner el código CSS, estudiémoslo más detalladamente. Antes os dí un trozo de código CSS sin explicároslo:

```
p
{
color: blue;
}
```

El código CSS como éste contiene tres elementos diferentes:

- **Nombres de etiquetas:** se escriben los nombres de las etiquetas cuya apariencia se quiere cambiar. Por ejemplo, si queremos cambiar la apariencia de todos los párrafos `<p>`, tenemos que escribir `p`.
- **Propiedades CSS:** la página "efectos de estilo" está almacenada en propiedades. Por ejemplo, la propiedad `color` nos permite especificar el color del texto, y la propiedad `font-size` se usa para especificar el tamaño del texto, etc. Existen un montón de propiedades CSS y, como ya os he dicho, no voy a haceros que os las aprendáis todas de memoria.
- **Valores:** para cada propiedad CSS, tenemos que especificar un valor. Para la propiedad `color`, por ejemplo, tenemos que especificar el nombre del color. Para la propiedad `font-size`, especificaremos el tamaño que queremos, etc.

Por lo general, una hoja de estilo CSS tiene esta apariencia:

```
tag1
```

```
{  
  
property1: value1;  
  
property2: value2;  
  
property3: value3;  
  
}
```

tag2

```
{  
  
property1: value1;  
  
property2: value2;  
  
property3: value3;  
  
property4: value4;  
  
}
```

tag3

```
{  
  
property1: value1;  
  
}
```

En este fragmento de código, podemos ver las etiquetas, propiedades y valores que acabo de mencionar.

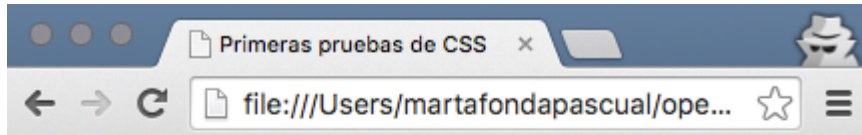
Como podéis ver, se escribe el nombre de la etiqueta (por ejemplo, `h1`) y se incluyen las propiedades y valores deseados entre corchetes. Se pueden incluir todas las propiedades que queráis dentro de los corchetes. Cada propiedad va seguida de "dos puntos" (":") y después el valor correspondiente. Por último, cada línea acaba con un punto y coma ";".

Voy a enseñaros una serie de propiedades en los siguientes capítulos. De momento, solo vamos a cambiar el color de los ejemplos como ejercicio.

El código CSS que hemos usado hasta ahora:

```
p
{
color: blue;
}
```

...que significa: " *Quiero que todos mis párrafos estén escritos en azul.*". El resultado se muestra en la ilustración siguiente.



Mi gran sitio web

¡Hola y bienvenidos a mi sitio web!

Mi sitio web todavía está un poco *vacío*. ¡Pero espera un poco!

Párrafos escritos en

azul

Intentad cambiar el nombre de la etiqueta afectada por el código CSS. Por ejemplo, si escribimos `h1`, el título estará escrito en azul. Editad vuestro archivo `style.css` así:

```
h1
{
color: blue;
}
```

Ahora abrid de nuevo la página HTML (recordad que es la página HTML que se abre en el navegador, no el archivo CSS): debe aparecer el título mostrado en azul (siguiente ilustración).



Mi gran sitio web

¡Hola y bienvenidos a mi sitio web!

Mi sitio web todavía está un poco *vacío*. ¡Pero espera un poco!

Título escrito en azul

Aplicar un estilo a varias etiquetas

Tomemos como ejemplo el siguiente código CSS:

```
h1
{
  color: blue;
}

em
{
  color: blue;
}
```

Significa que nuestros títulos `<h1>` y nuestro texto importante `<em>` se debe mostrar en azul. Sin embargo, es un poco repetitivo, ¿no creéis?

Afortunadamente, en CSS hay un modo de hacerlo más rápido si se le quiere dar la misma presentación a dos etiquetas. Simplemente tenemos que combinar la declaración, separando los nombres de las etiquetas con una coma, así:

```
h1, em
{
  color: blue;
}
```

El resultado se muestra en la ilustración siguiente.



Mi gran sitio web

¡Hola y bienvenidos a mi sitio web!

Mi sitio web todavía está un poco *vacio*. ¡Pero espera un poco!

Título y texto

importante escrito en azul

Esto significa: " Quiero el texto de mis etiquetas `<h1>` y `<em>` escrito en azul ".

Se pueden especificar todas las etiquetas que queremos una detrás de otra.

Comentarios en CSS

Como ocurre en HTML, se pueden incluir comentarios. Los comentarios no se muestran, simplemente se usan para especificar información destinada a vosotros, para encontrar, por ejemplo, algo que estamos buscando en un archivo CSS largo.

Y, como veréis, el archivo HTML generalmente es bastante corto y la hoja CSS bastante larga (si contiene todos los elementos de estilo de vuestro sitio web, es bastante normal). Tened en cuenta que se pueden crear varios archivos CSS para un sitio web si tenemos la necesidad de separar el código CSS de alguna forma (dependiendo de las diferentes secciones del sitio web, por ejemplo).

¿De qué estábamos hablando? Ah, sí. Comentarios en CSS.

Pues eso, incluir un comentario es muy fácil. Tenemos que introducir `/*` seguido por nuestro comentario, y después `*/` para completar el comentario.

Se pueden escribir comentarios en una o más líneas. Por ejemplo:

```
/*  
  
style.css  
  
-----  
  
Por Marta Fonda  
  
*/
```

```
p
{
    color: blue; /* Los párrafos serán azules */
}
```

Puedo utilizar comentarios en el resto del curso para daros explicaciones dentro de los mismos archivos `.css`.

Aplicar un estilo: class y id

Lo que os he enseñado hasta ahora tiene una limitación: significa, por ejemplo, que TODOS los párrafos tendrán la misma presentación (en este caso, estarán escritos en azul). ¿Qué tenemos que hacer para que solo algunos párrafos estén escritos de forma diferente? Pondremos el código CSS en un atributo `style` de la etiqueta en cuestión (esta es la técnica que os enseñé antes), pero como os he dicho, no se recomienda (es mejor usar un archivo CSS externo).

Para solucionar el problema, podemos usar estos atributos especiales *que funcionan con todas las etiquetas*:

- el atributo `class`;
- el atributo `id`.

Seamos claros desde el principio: los atributos `class` y `id` son prácticamente lo mismo. Hay solo una pequeña diferencia que os mostraré más abajo.

Por ahora, y para hacerlo más simple, solo vamos a centrarnos en el atributo `class`.

Como acabo de deciros, es un atributo que se puede poner en cualquier etiqueta, ya sea título, párrafo, imagen, etc.

```
<h1 class=""> </h1>

<p class=""> </p>

<img class="" />
```

Sí pero, ¿qué valor se incluye para el atributo `class`?

Tenemos que introducir un nombre que identifique a la etiqueta. El que queramos, siempre que el nombre empiece con una letra.

Por ejemplo, voy a asociar el atributo `class introduction` con mi primer párrafo (línea 12):

```
<!DOCTYPE html>

<html>
```

```

<head>

  <meta charset="utf-8" />

  <link rel="stylesheet" href="style.css" />

  <title>Primeras pruebas de CSS</title>

</head>


<body>

  <h1>Mi gran sitio web</h1>


  <p class="introduction">¡Hola y bienvenido a mi sitio web!</p>

  <p>Mi sitio web todavía está un poco <em>vacío</em>. ¡Pero espera un
poco!</p>

</body>

</html>

```

Una vez hecho esto, el párrafo ya está identificado. Tiene un nombre: `introduction`. Podremos volver a usar este nombre en el archivo CSS para decir: " *Quiero que se muestren en azul solo las etiquetas llamadas 'introduction'*".

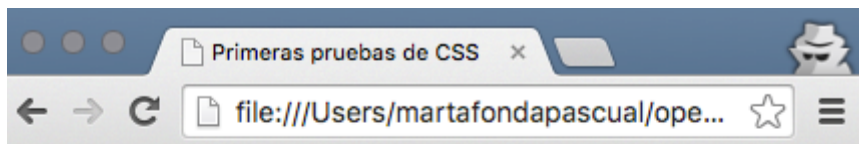
Para hacer esto en CSS, introducimos el nombre de nuestro atributo `class` con un punto al principio, como se muestra a continuación:

```

.introduction
{
  color: blue;
}

```

Probad el resultado: solo el párrafo llamado `introduction` se mostrará en azul (siguiente ilustración).



Mi gran sitio web

¡Hola y bienvenidos a mi sitio web!

Mi sitio web todavía está un poco *vacío*. ¡Pero espera un poco!

Solo el primer párrafo

se muestra en azul

¿Y el atributo `id`?

Funciona exactamente igual que el atributo `class`, con una excepción: solo se puede utilizar *una* vez en el código.

¿Qué finalidad tiene esto? No mucha, para ser honestos. Os sería útil si utilizarais JavaScript más adelante para reconocer ciertas etiquetas. Y ya hemos visto el atributo `id` en el capítulo sobre enlaces (para introducir hipervínculos). En la práctica, solo introducimos un atributo `id` sobre elementos que son únicos en la página, como el logo:

```
</code>
```

Si utilizamos atributos `id`, al definir sus propiedades en el archivo CSS, tenemos que poner una almohadilla (#) delante del nombre de `id`:

```
#logo

{

    /* Especificad las propiedades CSS aquí */

}
```

No os molestéis en probarlo, funciona exactamente igual que `class`.

Si confundís los atributos `class` y `id`, recordad que dos etiquetas pueden tener el mismo nombre con el atributo `class`. Y el nombre de `id`, sin embargo, debe ser único en la página HTML.

Etiquetas de propósito general

Quizás alguna vez tengamos que aplicar un atributo `class` (o un atributo `id`) a algunas palabras que originalmente no iban entre etiquetas.

De hecho, el problema de `class` es que es un atributo. Así que solo podemos introducirlo en una etiqueta. Si, por ejemplo, solo quiero cambiar "bienvenido" en el siguiente párrafo:

```
<p>¡Hola y bienvenido a mi sitio!</p>
```

Sería fácil hacerlo si "bienvenido" estuviera entre etiquetas, pero no lo está. Por suerte, alguien inventó... la etiqueta-sin-propósito.

De hecho, se inventaron dos etiquetas llamadas **de propósito general**, que no tienen un significado especial (no especifican que la palabra es importante, por ejemplo). Existe una pequeña (aunque importante) diferencia entre estas dos etiquetas:

- `<span>` `</span>`: es una etiqueta de tipo **inline**, es decir, una etiqueta situada dentro de un párrafo de texto, para seleccionar solo ciertas palabras. `<strong>` y `<em>` son de la misma familia. Esta etiqueta se utiliza en medio de un párrafo y es la etiqueta que vamos a usar para darle color a "bienvenido".
- `<div>` `</div>`: es una etiqueta de tipo **block**, entre la cual se encuentra un bloque de texto. `<p>`, `<h1>`, etc. son de la misma familia. Estas etiquetas tienen algo en común: crean un nuevo "bloque" en la página y, de este modo, insertan un salto de línea necesariamente. `<div>` es una etiqueta que se usa frecuentemente para crear un diseño, como veremos más adelante.

Por ahora, utilizaremos la etiqueta `<span>` en su lugar. La colocamos alrededor de "bienvenido", le añadimos un atributo `class` (el nombre que queráis), creamos el archivo CSS y ya lo tenemos.

```
<p>¡Hola y <span class="heybuddy">bienvenidos</span> a mi sitio web!</p>

.heybuddy

{

    color: blue;

}
```

Podéis observar el resultado en la siguiente ilustración.



Mi gran sitio web

¡Hola y bienvenidos a mi sitio web!

Mi sitio web todavía está un poco vacío. ¡Pero espera un poco!

está escrita en azul

La palabra bienvenido

Aplicar un estilo: los selectores avanzados

En CSS, lo más difícil es saber cómo dar en el clavo con el texto al que queremos cambiar el formato. Para seleccionar los elementos de la página que queremos cambiar, usamos los llamados **selectores**. Ya hemos usado varios de ellos anteriormente en este capítulo, así que vamos a resumirlos para comenzar.

Los selectores que ya conocemos

Estos selectores, como hemos visto más arriba, son los más usados con diferencia. Tenéis que aprendéroslos de memoria. Empecemos por el principio:

```
p
{
}

```

significa "quiero cambiar todos los párrafos". Después de eso, es decisión tuya lo que quieras hacerle a estos párrafos (escribirlos en azul, por ejemplo).

También vimos:

```
h1, em
{
}

```

... que significa "Todos los títulos y todo el texto importante". Hemos seleccionado dos etiquetas a la vez.

Y, por último, hemos visto cómo seleccionar etiquetas específicas a las que les hemos dado un nombre utilizando los atributos `class` y `id`:

```
.class
{
}

```

```
#id
```

```
{
```

```
}
```

¿Y sabes qué? Hay muchas más formas de seleccionar etiquetas en CSS. No vamos a verlas todas porque son muchas y algunas son bastante complejas, pero a continuación te doy algunos consejos a tener en cuenta para ser más efectivo en CSS.

Los selectores avanzados

*: selector de propósito general

```
*
```

```
{
```

```
}
```

Selecciona todas las etiquetas sin excepción. Se llama selector de propósito general.

A B: una etiqueta dentro de otra

```
h3 em
```

```
{
```

```
}
```

Selecciona todas las etiquetas `<em>` situadas dentro de una etiqueta `<h3>`. Ten en cuenta que no hay coma entre los nombres de las dos etiquetas.

Ejemplo de código HTML correspondiente:

```
<h3>Título con <em>texto importante</em></h3>
```

A + B: una etiqueta seguida de otra etiqueta

```
h3 + p
```

```
{
```

```
}
```

Selecciona la primera etiqueta `<p>` situada después de un título `<h3>`.

Ejemplo:

```
<h3>Título</h3>
```

```
<p>Párrafo</p>
```

A[attribut]: una etiqueta que tiene un atributo

```
a[título]
```

```
{
```

```
}
```

Selecciona todos los enlaces `<a>` que tienen un atributo `title`.

Ejemplo:

```
<a href="http://site.com" title="Descripción emergente">
```

A[attribute="Valor"]: una etiqueta, un atributo y un valor exacto

```
a[title="Clic aquí"]
```

```
{
```

```
}
```

Igual, pero el atributo también debe tener el valor exacto "Clic aquí".

Ejemplo:

```
<a href="http://site.com" title="Clic aquí">
```

A[attribute="Valor"]: una etiqueta, un atributo y un valor

```
a[title*="aquí"]
```

```
{
```

```
}
```

Igual, pero el atributo esta vez tiene que contener la palabra "aquí" en este valor (en cualquier posición).

Ejemplo:

```
<a href="http://site.com" title="Por aquí">
```

¡También existen otros selectores!

Aquí solo he descrito algunos selectores CSS y hay muchos más. Si queréis una lista completa, podéis encontrarla directamente en el [sitio web de W3C website](#). Es muy completa.

Descubriremos más selectores más adelante.

En resumen

- CSS es otro lenguaje que complementa al lenguaje HTML. Su función es dar formato a tu sitio web.
- Tienes que tener cuidado con la compatibilidad de los navegadores con algunas funciones recientes de CSS3. Cuando un navegador no conoce una instrucción de formato, simplemente la ignora.
- Podemos escribir código CSS en varios sitios diferentes, siendo el más recomendado crear un archivo separado con la extensión `.css` (por ejemplo: `style.css`).
- En CSS, seleccionamos qué partes de la página HTML queremos editar y después cambiamos su presentación con propiedades CSS:

```
tag1
```

```
{
```

```
property1: value1;
```

```
property2: value2;
```

```
}
```

- Hay muchas formas de seleccionar la sección de la página a la que queremos dar formato. Por ejemplo, podemos seleccionar:
 - todas las etiquetas del mismo tipo, con tan solo introducir su nombre (`h1` por ejemplo);
 - algunas etiquetas específicas, a las que hemos asignado etiquetas utilizando los atributos `class` o `id` (`.classname` o `#idname`);
 - solo etiquetas situadas dentro de otras etiquetas (`h3 em`).

Dar formato al texto

Hemos llegado a un capítulo que puede resultaros muy interesante.

No, "formatear texto" no tiene nada que ver con la destrucción de todos los datos de tu disco duro. Simplemente significa que la apariencia del texto se va a cambiar (o "formatear").

No hay nada particularmente inesperado: Todavía estamos tratando el CSS y vamos a volver a utilizar lo que hemos aprendido en el capítulo anterior. Vamos a trabajar directamente en el archivo `.css` que hemos creado.

Este capítulo nos dará la oportunidad de descubrir muchas propiedades CSS: vamos a ver cómo cambiar el tamaño del texto, cambiar la fuente, alinear el texto, etc.

Tamaño

Para cambiar el tamaño del texto, usamos la propiedad CSS `font-size`. ¿Pero cómo especificamos el tamaño del texto? Aquí es donde la cosa se complica, ya que hay varias técnicas a nuestra disposición:

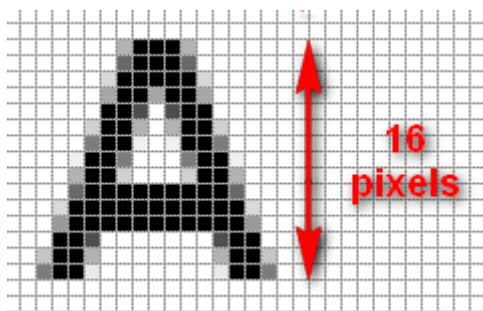
- Especificar un **tamaño absoluto**: en píxeles, pulgadas o milímetros. Aunque este método es muy preciso, se recomienda utilizarlo solo cuando sea absolutamente necesario, ya que podríamos especificar un tamaño demasiado pequeño para algunos lectores.
- Especificar un **tamaño relativo**: como un porcentaje, "em" o "ex". Esta técnica tiene la ventaja de ser más flexible. Se puede adaptar más fácilmente a las preferencias de tamaño de los visitantes.

Un tamaño absoluto

Para especificar un tamaño absoluto, normalmente se utilizan píxeles. Así que para obtener un texto de 16 píxeles de altura, tenemos que escribir esto:

```
font-size: 16px;
```

Las letras tendrán un tamaño de 16 píxeles, como se muestra en la ilustración de abajo.



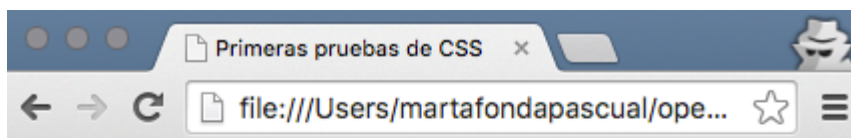
Una letra de 16 píxeles de altura

Aquí tenéis un ejemplo de cómo usarlo (poned este código en vuestro archivo `.css`):

```
p
{
    font-size: 14px; /* Párrafos de 14 píxeles */
}

h1
{
    font-size: 40px; /* Títulos de 40 píxeles */
}
```

Y el resultado se muestra en la siguiente ilustración.



Mi gran sitio web

¡Hola y bienvenidos a mi sitio web!

Mi sitio web todavía está un poco *vacío*. ¡Pero espera un poco!

Varios tamaños de

texto

Si queréis, también podéis establecer los tamaños en centímetros o milímetros. Cambiad "px" por "cm" o "mm". Sin embargo, estas unidades son menos adecuadas para las pantallas.

Un valor relativo

Este es el método recomendado porque el texto se adapta más fácilmente a las preferencias de todos los visitantes.

Existen varias formas de especificar un valor relativo. Puedes escribir el tamaño en palabras, así:

- `xx-small` : diminuto ;
- `x-small` : muy pequeño ;
- `small` : pequeño ;

- `medium` : mediano ;
- `large` : grande ;
- `x-large` : muy grande ;
- `xx-large` : enorme.

Puedes probar a utilizar estos valores en tu código CSS:

```
p
{
    font-size: pequeño;
}

h1
{
    font-size: grande;
}
```

Sin embargo, esta técnica tiene un inconveniente: solo hay siete tamaños disponibles (porque solo hay siete nombres). Afortunadamente, existen otras opciones. Mi técnica favorita es especificar el tamaño en "em".

- Si escribes `1em`, el texto tiene un tamaño normal.
- Si quieres que el texto sea más grande, puedes introducir un valor superior a 1, como `1.3em`.
- Si quieres que el texto sea más pequeño, puedes introducir un valor inferior a 1, como `0.8em`.

Atención: los números decimales se escriben con punto, no con coma. Por ejemplo, debes escribir « `1.4em` », no « `1,4em` ».

Ejemplo:

```
p
{
    font-size: 0.8em;
}

h1
```

```
{  
  
    font-size: 1.3em;  
  
}
```

Hay otras unidades disponibles. Podéis probar "ex" (que funciona con el mismo principio que em pero tiene una base más pequeña) y el porcentaje (80 %, 130 %, etc.).

Fuente

Ah... La fuente... Esto puede ser complicado

De hecho, hay un problema: para que una fuente se muestre correctamente, todos los usuarios de Internet tienen que tenerla. Si un usuario de Internet no tiene la misma fuente que tú, su navegador usará una fuente por defecto (una fuente estándar) que puede que no sea lo que esperabas.

La buena noticia es que desde CSS 3, ahora es posible que el navegador descargue una fuente automáticamente. Os explicaré cómo hacer esto más adelante.

Cambiar la fuente utilizada

La propiedad CSS que nos permite especificar la fuente a utilizar es `font-family`. Tienes que escribir el nombre de la fuente así:

```
tag  
  
{  
  
    font-family: fuente;  
  
}
```

Para evitar problemas si el usuario de Internet no tiene la misma fuente que tú, se pueden especificar *varios* nombres de fuente, separados por comas:

```
tag  
  
{  
  
    font-family: font1, font2, font3, font4;  
  
}
```

El navegador intentará usar primero la `font1`. Si no la tiene, lo intentará con la `font2`. Si no la tiene, lo intentará con la `font3`, y así sucesivamente. Pero os preguntaréis cuáles son las fuentes más comunes que tenéis "derecho" a utilizar.

Esta es una lista de fuentes que funcionan bien en la mayoría de los navegadores:

- Arial ;
- Arial Black ;
- Comic Sans MS ;
- Courier New ;
- Georgia ;
- Impact ;
- Times New Roman ;
- Trebuchet MS ;
- Verdana.

En la siguiente ilustración se muestran las diferentes fuentes.

Arial

Arial Black

Comic Sans MS

Courier New

Georgia

Impact

Times New Roman

Trebuchet MS

Verdana

Varias fuentes

Entonces, si escribo:

```
p
{
    font-family: Impact, "Arial Black", Arial, Verdana, sans-serif;
}
```

... esto significa: " Utilizar la fuente *Impact* o, si no se encuentra, *Arial Black*, o *Arial*, o *Verdana*, o si no funciona ninguna de éstas, utilizar una fuente estándar (*sans-serif*)".

En general, es una buena idea especificar una variedad de tres o cuatro fuentes (+ serif o sans-serif) para asegurarnos de que al menos una de ellas se encontrará en el ordenador del visitante.

Si el nombre de fuente incluye espacios, recomiendo ponerlo entre comillas, como he hecho yo con "Arial Black".

Como último recurso, se suele utilizar serif, que corresponde a la fuente predeterminada (que solo se aplica si no se ha encontrado ninguna otra fuente). También existe otra fuente predeterminada llamada sans-serif. La diferencia entre las dos es que la fuente sans-serif no tiene las pestañitas de conexión en la parte inferior de las letras. Sí, es casi imperceptible.

Utilizar una fuente personalizada con @font-face

A mi parecer la selección de fuentes es demasiado limitada.

¿Cómo puedo utilizar mi fuente favorita en mi página web?

Durante mucho tiempo, esto no era posible. Ahora, con CSS3, afortunadamente existe una forma de usar cualquier fuente en tu sitio web. Funciona bien con la mayoría de los navegadores.

Pero tened cuidado porque hay algunas complicaciones (de lo contrario, sería demasiado bonito para ser verdad):

- El navegador del visitante tendrá que *descargar* automáticamente el archivo fuente cuyo tamaño de almacenamiento podría ser superior a 1 MB...
- La mayoría de las fuentes tienen copyright, así que no es *legal* utilizarlas en nuestros sitios web. Por suerte, existen páginas web como fontquirrel.com y dafont.com que nos permiten descargar una serie de fuentes libres de copyright. En particular, recomiendo fontquirrel.com que te permite descargar paquetes para CSS3 listos para usar.
- Existen *varios formatos* de archivos fuente y no todos funcionan en todos los buscadores.

Estos son los formatos de archivo fuente existentes que debéis conocer:

- `.ttf`: *TrueType Font*. Funciona con IE9 y todos los demás navegadores.
- `.eot`: *Embedded OpenType*. Funciona solo con todas las versiones de Internet Explorer. Es un formato registrado, desarrollado por Microsoft.
- `.otf`: *OpenType Font*. No funciona con Internet Explorer.
- `.svg`: *SVG Font*. El único formato soportado por teléfonos iPhone y iPads por el momento.
- `.woff`: *Web Open Font Format*. Nuevo formato diseñado para la Red, que funciona con IE9 y todos los demás navegadores.

Para definir una fuente nueva en CSS, tienes que declararlo así:

```
@font-face {  
  
    font-family: 'MiSuperFuente';  
  
    src: url('MiSuperFuente.eot');  
  
}
```

El archivo fuente (en este caso *MiSuperFuente.eot*) debe estar ubicado en la misma carpeta que el archivo CSS (o en una subcarpeta si utilizáis una ruta relativa).

Yo creía que existían varios formatos de fuente.

Sí, y además `.eot` solo funciona con Internet Explorer. La solución ideal es proporcionar varios formatos para la fuente: el navegador descargará el que pueda leer. Aquí tenemos la forma de especificar varios formatos:

```
@font-face {  
  
    font-family: 'MiSuperFuente';  
  
    src: url('MiSuperFuente.eot') format('eot'),  
  
        url('MiSuperFuente.woff') format('woff'),  
  
        url('MiSuperFuente.ttf') format('truetype'),  
  
        url('MiSuperFuente.svg') format('svg');  
  
}
```

Para probar cómo funciona esto, recomiendo que descarguéis una fuente de Font Squirrel, por ejemplo [Learning Curve Pro](#). Haced clic en "@font-face Kit", que te permitirá descargar un paquete listo para usar con todos los formatos para esta fuente.

Tu archivo CSS al final será así:

```
@font-face { /* Define una nueva fuente llamada LearningCurveProRegular */  
  
    font-family: 'LearningCurveProRegular';  
  
    src: url('LearningCurve_OT-webfont.eot');  
  
    src: url('LearningCurve_OT-webfont.eot?#iefix') format('embedded-opentype'),  
  
        url('LearningCurve_OT-webfont.woff') format('woff'),
```

```

    url('LearningCurve_OT-webfont.ttf') format('truetype'),

    url('LearningCurve_OT-webfont.svg#LearningCurveProRegular') format('svg');
}

h1 /* Usa la fuente que acabas de especificar para los títulos */
{
    font-family: 'LearningCurveProRegular', Arial, serif;
}

```

La primera (gran) sección `@font-face` define un nuevo nombre de fuente que puede utilizarse en el archivo CSS. Entonces utilizamos este nombre de fuente con la propiedad `font-family`, que ya conocemos, para cambiar la apariencia de los títulos `<h1>`. Podéis observar el resultado en la siguiente ilustración.

Bienvenidos a mi website!

Visualización de una fuente personalizada

Descubriréis varias peculiaridades en el CSS generado por el sitio web Font Squirrel. El objetivo es esquivar ciertos errores en Internet Explorer porque las versiones antiguas no entienden cuándo se definen varios formatos. Esto explica el uso de `?#iefix` en el código.

Cursiva, negrita, subrayado...

CSS tiene una serie de propiedades de formato de texto estándar. Ahora vamos a descubrir cómo mostrar texto en cursiva, cursiva, subrayado, etc. y veremos que incluso es posible hacerlo parpadear.

Formato cursiva

¡Alto ahí! ¿Yo pensaba que la etiqueta `<em>` se utilizaba para mostrar texto en cursiva?

Yo nunca he dicho eso.

Si tenéis alguna duda, volved a los capítulos anteriores, pero yo *nunca* he dicho que la etiqueta `<em>` se usaba para poner el texto en cursiva (al igual que nunca he dicho que `<strong>` se utilice para poner el texto en negrita).

`<em>`, grabadlo a fuego en vuestras mentes, se utiliza para enfatizar palabras. Esto significa que las palabras del interior son muy importantes.

Para representar esta importancia, la mayoría de los navegadores muestran el texto en cursiva, pero no es obligatorio.

CSS, por su parte, te permite decir: "Quiero que este texto esté en cursiva". Por ejemplo, no hay nada que te impida decidir poner todos los títulos en cursiva.

Concretamente, para poner el texto en cursiva en CSS, se utiliza `font-style` que puede llevar tres valores:

- `italic`: el texto se mostrará en cursiva.
- `oblique`: el texto se mostrará oblicuamente (las letras se inclinan y el resultado es ligeramente diferente a la cursiva).
- `normal`: el texto se mostrará en formato normal (predeterminado). Esto permite cancelar la cursiva. Si quieres que el texto que va entre `<em>` deje de aparecer en cursiva, por ejemplo, tienes que escribir:

```
em
{
    font-style: normal;
}
```

En el siguiente ejemplo, he utilizado `font-style` para poner en cursiva todos mis títulos `<h2>`:

```
h2
{
    font-style: cursiva;
}
```

Negrita

¿Y qué ocurre con la negrita?

De nuevo, recordad que `<strong>` no pone el texto en negrita (su propósito es indicar que el texto es importante, y *como resultado* el navegador normalmente muestra el texto en negrita). En CSS, puedes aplicar la negrita en los títulos, algunos párrafos enteros, etc. Eso es decisión tuya.

La propiedad de CSS para poner en negrita es `font-weight` y tiene los siguientes valores:

- `bold`: el texto estará en negrita;
- `normal`: el texto se mostrará en formato normal (predeterminado).

Aquí tenéis un ejemplo de cómo escribir títulos en negrita:

```
h1
{
    font-weight: negrita;
}
```

Subrayado y otras decoraciones

La propiedad CSS asociada tiene un nombre muy acertado: `text-decoration`. Te permite subrayar texto, entre otras cosas, pero no solo eso. Estas son las diferentes posibilidades que ofrece:

- `underline`: subrayado.
- `line-through`: tachado.
- `overline`: línea encima.
- `blink`: parpadeante. No funciona en la mayoría de los navegadores.
- `none`: normal (predeterminado).

Este código CSS te permitirá probar los efectos de `text-decoration` usando `underline`:

```
h1
{
    text-decoration: underline;
}
```

Y el resultado se muestra en la siguiente ilustración.



Un texto subrayado

Alineación

El lenguaje CSS te permite establecer todas las alineaciones de texto conocidas: izquierda, centrado, derecha y justificado.

Es bastante sencillo. Utilizas la propiedad `text-align` y especificas la alineación deseada:

- `left`: el texto se alinearán a la izquierda (este es el ajuste predeterminado).
- `center`: el texto aparecerá centrado.
- `right`: el texto se alinearán a la derecha.
- `justify`: el texto aparecerá justificado. Justificar el texto hace que se ocupe todo el ancho posible sin dejar ningún espacio en blanco al final de las líneas. Los artículos de periódicos, por ejemplo, siempre se justifican.

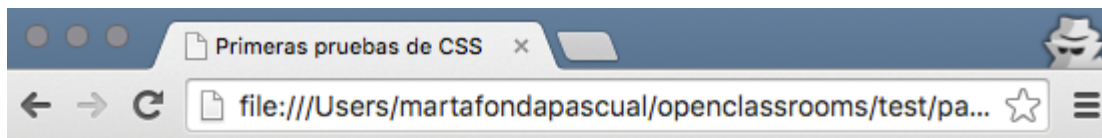
Echemos un vistazo a los diferentes tipos de alineación en este ejemplo:

```
h1
{
    text-align: center;
}

p
{
    text-align: justify;
}

.signature
{
    text-align: right;
}
```

El resultado se muestra en la siguiente ilustración.



Mi Título

Hola!

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Explicabo expedita error quia voluptatum repudiandae nam quo veniam, pariatur, consectetur neque autem magnam laborum voluptas natus harum libero sed consequuntur non.

Adios!

Alineación de texto

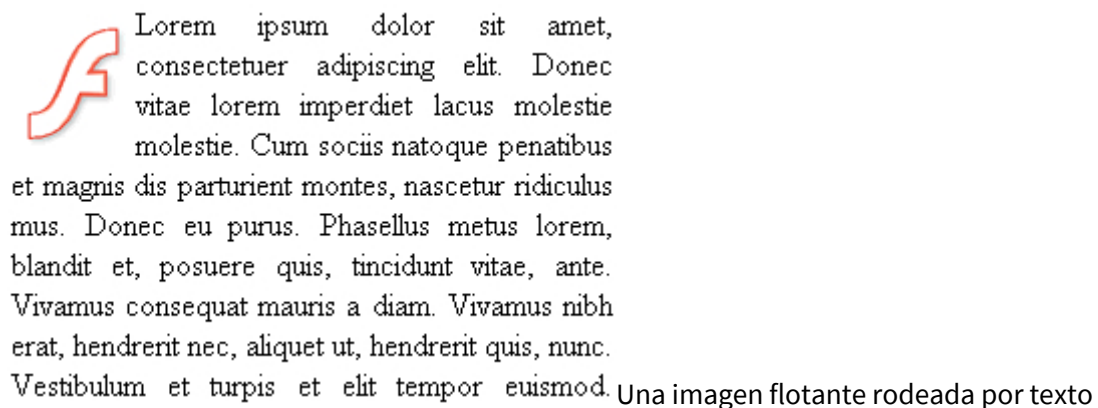
No se puede cambiar la alineación del texto de una etiqueta inline (como por ejemplo `<span>`, `<a>`, `<em>`, `<strong>`...). La alineación solo funciona en las etiquetas de tipo block (`<p>`, `<div>`, `<h1>`, `<h2>`, etc.), algo lógico si lo pensamos detenidamente: no se puede cambiar la alineación de unas cuantas palabras en medio de un párrafo.

Así que, en general, hay que alinear el párrafo entero.

Elementos flotantes

CSS nos permite colocar un elemento alrededor del texto. A esto también se le llama "wrapping".

Para que os hagáis una idea de lo que estamos hablando, la ilustración de abajo nos muestra lo que vamos a aprender a hacer.



Una imagen flotante rodeada por texto

Creo que la pregunta candente que está ahora en tu mente es: « *¿Pero cuál es la propiedad mágica que hace que las cosas floten?* ».

La respuesta es... `float`. Esta propiedad puede tener dos valores muy sencillos:

- `left`: el elemento flotará en la izquierda.
- `right`: el elemento flotará en la derecha. Sí, bien hecho.

Los elementos flotantes son muy sencillos de usar:

1. Sólo aplicas un `float` a una etiqueta.
2. Entonces continúas escribiendo el texto siguiendo de forma normal.

Pues usar una propiedad de flotación en las etiquetas de bloque e inline. Es habitual una imagen flotante con un texto rodeándola, como en el ejemplo anterior.

Hacer que una imagen flote

Aquí vamos a aprender cómo hacer que una imagen flote. Aquí está el código HTML que tenemos que introducir primero:

```
<p> Este es un  
texto de párrafo normal, escrito después de la imagen rodeada por él mientras la  
imagen flota.</p>
```

Primero tienes que colocar el elemento flotante en el código HTML. Si colocas una imagen después del párrafo, el efecto no funcionará.

Aquí está el trozo de código CSS que tienes que escribir para hacer flotar a la imagen en la izquierda:

```
.imagenflotante  
  
{  
  
    float: left;  
  
}
```

Inténtalo y haz que la imagen flote en la derecha, es muy fácil: sólo especifica el valor `right` ¡y ya está!

Parar un efecto de flotación

Cuando colocas una imagen flotante, el texto la rodea. Pero después de un rato, ¿qué pasa si quieres que el texto continúe *por debajo* de la imagen flotante? Podrías combinar varias etiquetas `<br />` en secuencia pero esto no sería muy ordenado o elegante...

Nos gustaría básicamente obtener el mismo resultado que la figura siguiente.



Este es un texto de párrafo normal,
escrito después de la imagen
rodeada por él mientras la imagen

flota.

Pero por debajo el texto sigue.

El texto debajo de la imagen ignora la

propiedad de flotación

CSS realmente tiene una propiedad que te permite decir: « *Para, este texto debería estar debajo del elemento flotante y no a su lado.* » Esta es la propiedad `clear`, que puede tener estos tres valores:

- `left`: el texto continúa debajo después de un `float: left`;
- `right`: el texto continúa debajo después de un `float: right`;
- `both`: el texto continúa abajo, tanto después de un `float: left`; o después de un `float: right`;

Para simplificar, siempre vamos a usar el `clear: both`, que funciona después de un flotar a la izquierda y después de un flotar a la derecha (funciona siempre). Para ilustrar cómo funciona, vamos a tomar el siguiente código HTML:

```
<p></p>

<p>Este texto está escrito al lado de la imagen flotante.</p>

<p class="below">Este texto está escrito debajo de la imagen flotante.</p>
```

Y este código CSS:

```
.imagenflotante
{
    float: left;
}

.debajo
{
    clear: both;
}
```

Y eso es todo.

Aplicas un `clear: both;` al párrafo que quieres ver continuar debajo de la imagen flotante ¡y ya está!

En resumen

- Cambias el tamaño del texto con la propiedad CSS `font-size`. Puedes especificar el tamaño en píxeles (16px) en «em» (1.3em), como porcentaje (110 %), etc.
- Cambias la fuente del texto con `font-family`. Ten cuidado, sólo algunas fuentes son reconocidas por todos los ordenadores. Sin embargo, puedes usar una fuente personalizada con la indicación `@font-face`: esto obligará al navegador a descargar la fuente de tu elección.
- Hay muchas propiedades de formato de texto: `font-style` para cursiva, `font-weight` para negrita, `text-decoration` para subrayar, etc.
- El texto se puede alinear con `text-align`.
- Una imagen puede ser envuelta («rodeada») por texto usando `float`.

Color y fondo

Continuemos viendo las propiedades CSS existentes. Ahora vamos a ver las propiedades directa o indirectamente relacionadas con el color. Vamos a ver entre otras cosas:

- cómo cambiar el color del texto;
- cómo añadir un color o una imagen de fondo;
- cómo añadir sombras;
- cómo variar los niveles de transparencia

¡CSS todavía tiene muchas sorpresas guardadas!

Color del texto

Vayamos al amplio tema del color.

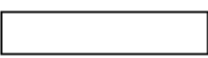
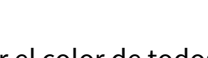
¿Qué quieres decir con «amplio»?

Ya conoces la propiedad usada para cambiar el color del texto: es `color`. Vamos a ver las diferentes formas de especificar el color, ya que hay varias.

Especificar el nombre del color

La manera más sencilla y más cómoda de elegir un color es escribir su nombre. El único problema con este método es que sólo hay dieciséis colores llamados «estándar». Aunque hay otros colores informales, no molestaré enseñándolos ya que no funcionan necesariamente de la misma manera en todos los navegadores.

La siguiente figura muestra los dieciséis colores que puedes usar simplemente escribiendo su nombre.

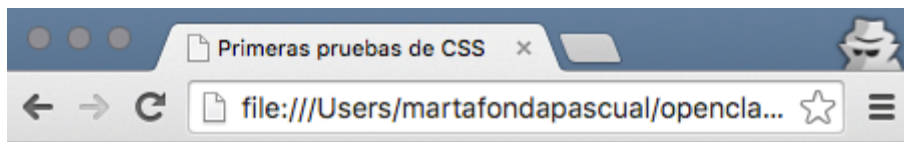
white	
silver	
gray	
black	
red	
maroon	
lime	
green	
yellow	
olive	
blue	
navy	
fuchsia	
purple	
aqua	
teal	

Los dieciséis nombres de colores pueden usarse en CSS

Para cambiar el color de todos los títulos a marrón, deberías ingresar:

```
h1
{
  color: maroon;
}
```

Puedes ver el resultado en la siguiente figura.



Mi gran sitio web

¡Hola y bienvenidos a mi sitio web!

Mi sitio web todavía está un poco *vacío*. ¡Pero espera un poco!

El título está
escrito en marrón

Notación hexadecimal

Pero dieciséis colores parecen un poco limitados cuando consideras que la mayoría de los monitores pueden mostrar dieciséis *millones*.

Por otro lado, ten en cuenta que si tuvieras que darle un nombre a cada uno de los dieciséis millones de colores...

Afortunadamente, hay varias maneras en CSS de elegir un color de entre los ya existentes. La primera manera que te enseñaré es en la notación hexadecimal. Aunque es muy común su uso en la web, hay también otro método que vamos a ver más tarde.

Así es como se ve el nombre de un color en hexadecimal: #FF5A28. Simplemente, se trata de una combinación de letras y números que especifican un color.

Deberías siempre empezar escribiendo una almohadilla (#) seguida de seis letras o dígitos del 0 al 9 y de la A a la F.

Estas letras o números funcionan en parejas. Los dos primeros dígitos especifican la cantidad de rojo, los siguientes dos, la cantidad de verde y los últimos dos la cantidad de azul. Mezclando estas cantidades (que son los componentes de los colores Rojo-Verde-Azul), puedes conseguir el color que quieras.

Así, #000000 corresponde al negro y #FFFFFF al blanco. Pero ahora no me preguntes qué combinación usar para producir el naranja «puesta de sol», porque no tengo ni idea.

Algunos programas de dibujo como Photoshop, [Gimp](#) y [Paint.NET](#) te dejan especificar los colores en hexadecimal. Puedes fácilmente copiar y pegar el código hexadecimal de un color en tu archivo CSS.

Aquí, por ejemplo, se muestra cómo aplicas el blanco a los párrafos en hexadecimal:

p

```
{  
  
  color: #FFFFFF;  
  
}
```

Ten en cuenta que hay una notación abreviada: puedes escribir un color con sólo tres caracteres. Por ejemplo: #FA3 es equivalente a escribir #FFAA33.

El método RGB

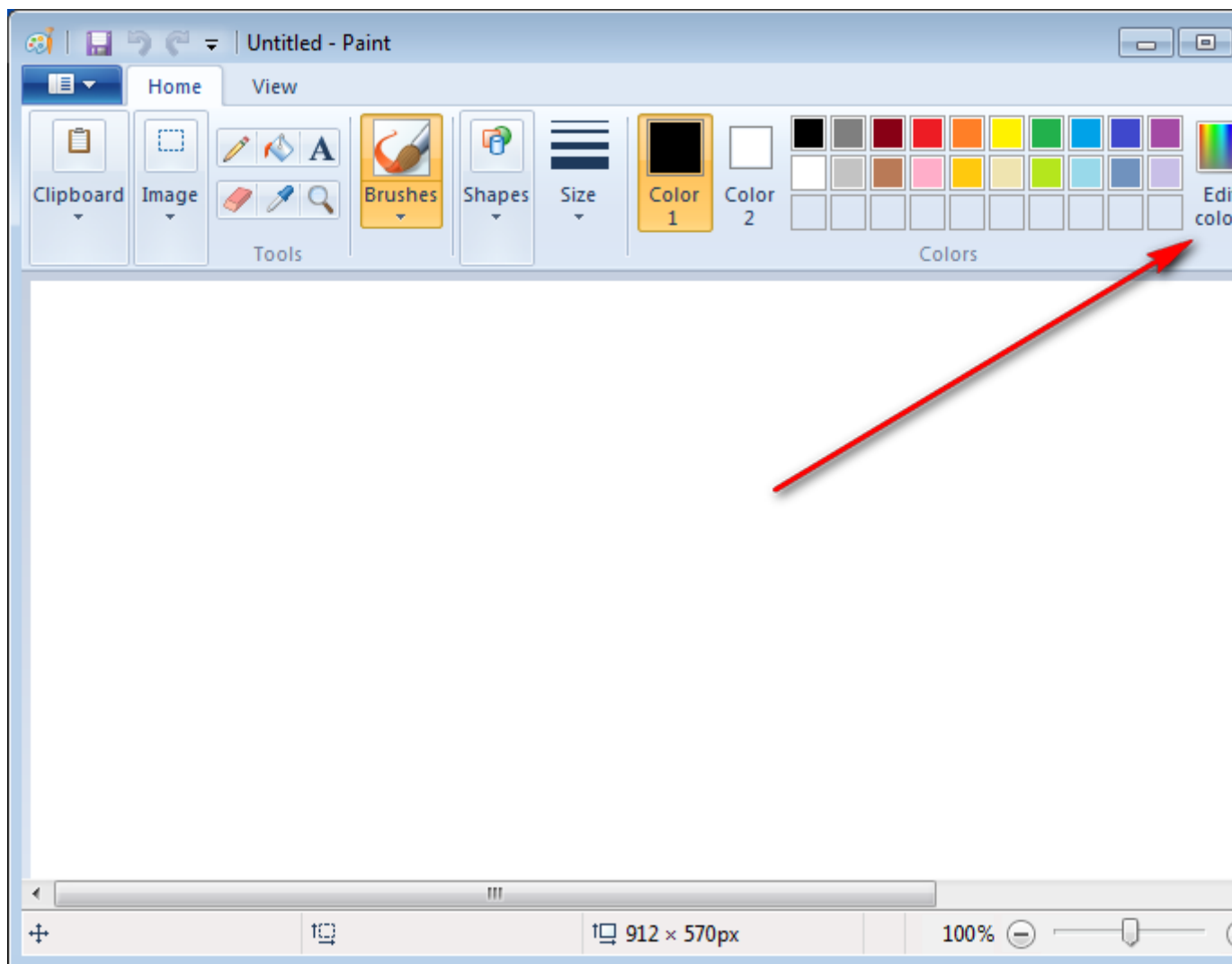
¿Qué significa RGB? *Rojo-Verde-Azul* (en inglés), que se abrevia como «RGB». Como en la notación hexadecimal, para elegir un color, tienes que definir una cantidad de rojo, verde y azul.

¿Esta historia rara de las cantidades de rojo-verde-azul otra vez?

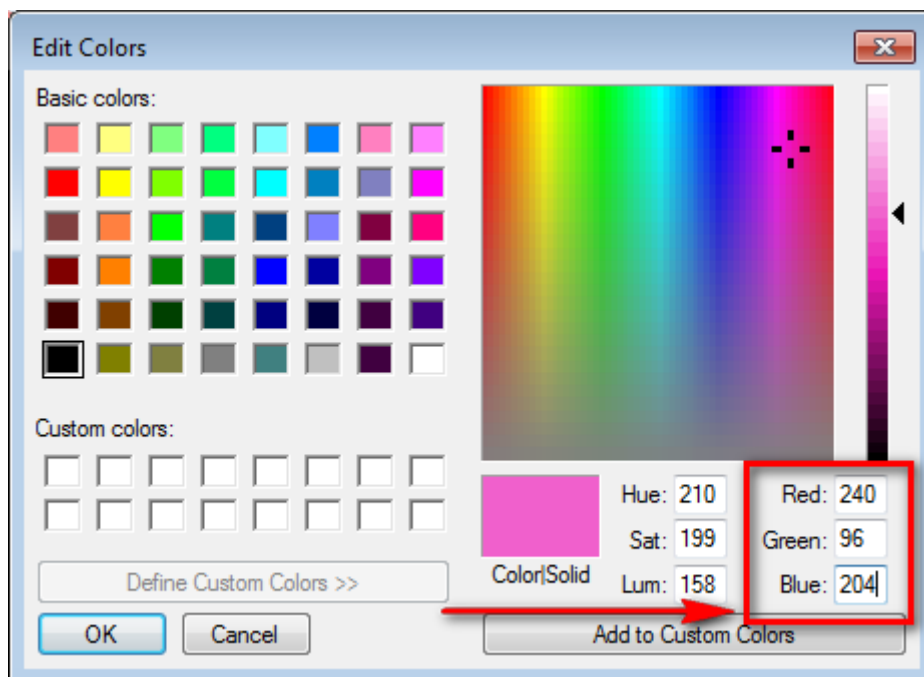
Sí, pero en este caso, verás que es mucho más práctico y que con un simple programa de dibujo como Paint puedes encontrar el color que quieres. Aquí están los pasos que hay que seguir:

1. Inicia el programa Paint desde el menú *Inicio*.
2. Ve a la sección de *Editar colores* como se muestra en la siguiente figura.
3. Se abre una ventana. En la caja que aparece en la derecha, mueve el deslizador para seleccionar el color que quieres. Supongamos que tienes un deseo loco de escribir tus títulos `<h1>` en rosa (sólo supongamos). Selecciona el color en la ventana, como muestra la siguiente figura.
4. Apunta las cantidades correspondientes de Rojo-Verde-Azul indicadas en la parte inferior derecha de la ventana (en este caso 240-96-204). Introduce estos valores en el mismo orden en el archivo CSS, como se muestra en el siguiente código.

```
p  
  
{  
  
  color: rgb(240,96,204);  
  
}
```



Cambiar los colores en Paint



color en Paint

Seleccionar un

Como puedes ver en el ejemplo, para usar el método RGB tienes que escribir `rgb(Red, Green, Blue)` reemplazando «Red, Green, Blue» por los números correspondientes. Para tu información, estas cantidades están siempre entre 0 y 255.

Color de fondo

Para especificar un color de fondo, usa la propiedad CSS `background-color`. Se usa de la misma manera que la propiedad `color`, en otras palabras, puedes escribir el nombre de un color, introducirlo en la notación hexadecimal o usar el método RGB.

Para especificar el color de fondo de la página web, tienes que trabajar con la etiqueta `<body>`. De hecho, `<body>` corresponde con la página web completa, así, cambiando su color de fondo, cambiarás el color de fondo de la página.

Mira con cuidado a este archivo CSS:

```
/* Estamos trabajando con la etiqueta del cuerpo, y por lo tanto con la página
COMPLETA */

body

{

    background-color: black; /* La página de fondo será negra */

    color: white; /*El texto de la página será blanco */

}
```

El resultado de este código se puede ver en la siguiente figura.



sobre fondo negro

Texto blanco

Pero pediste poner la etiqueta del texto del `<body>` en blanco, y todos los párrafos `<p>` y títulos `<h1>` cogerán este color. ¿Qué es eso?

En efecto, quería aprovechar esta oportunidad para hablarte sobre esto. Esto se conoce como **herencia**. Pero te lo puedo asegurar, nadie ha muerto.

CSS y herencia

En CSS, si aplicas un estilo a una etiqueta, todas las etiquetas que estén dentro cogerán el mismo estilo.

En realidad es fácil de entender e intuitivo. Como sabes, la etiqueta `<body>` tag contiene, entre otros, las etiquetas del párrafo `<p>` y del título `<h1>`.

Si aplico un fondo negro y una etiqueta de texto blanco al `<body>`, todos mis títulos y párrafos también tendrán un fondo negro y un texto blanco... Este fenómeno se llama herencia: las etiquetas que están dentro de otra etiqueta se dice que «heredan» sus propiedades.

Esto también vale para el nombre «CSS», que significa «*Cascading Style Sheets*» (Hojas de Estilo Cascada). Las propiedades CSS se heredan en cascada: si le das un estilo a un elemento, todos los subelementos tendrán el mismo estilo.

¿Esto significa que TODO el texto de mi página web estará necesariamente en blanco?

No, no necesariamente. Si después dices que quieres que tus títulos estén en rojo, este estilo tendrá preferencia y tus títulos estarán en rojo. Sin embargo, si no especificas nada en particular (como hicimos anteriormente), entonces tus títulos heredarán el color blanco.

Seamos claros, esto no sólo funciona con el color. Todas las propiedades CSS se heredan: por ejemplo, puedes asignar la negrita en la etiqueta `<body>` y todos tus títulos y párrafos estarán en negrita.

Ejemplo de herencia con la etiqueta `<mark>`

La gente tiende a pensar que sólo pueden cambiar el color de fondo de su página. Esto no es así: puedes cambiar el fondo de cualquier elemento: tus títulos, tus párrafos, algunas palabras, etc. En este caso, aparecerán destacadas (como si estuvieran marcadas con un subrayador).

Por ejemplo, ¿recuerdas la etiqueta `<mark>` que se usa para destacar algunas palabras? Lo usaremos otra vez aquí:

```
<h1>Quién apagó la luz?</h1>
```

```
<p>Brr, en esta página está todo negro es una atmósfera un tanto  
<mark>molesta</mark>, ¿no crees?</p>
```

Por defecto, el texto aparece con fondo amarillo. Puedes cambiar este comportamiento en CSS:

```
body
{
    background-color: black; /* La página de fondo será negra */

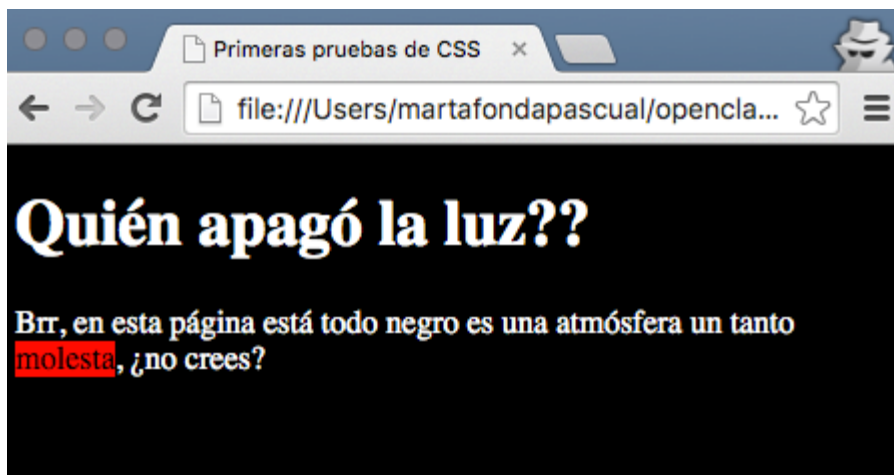
    color: white; /*El texto de la página será blanco */
}

mark
{
    /* El color del fondo anula el color de la página completa */

    background-color: red;

    color: black;
}
```

El color de fondo rojo se aplica al texto de la etiqueta `<mark>`. De hecho, incluso si el fondo de la página es negro, es la propiedad CSS del elemento más específico la que tiene prioridad (siguiente figura).



Un texto rojo
destacado sobre un fondo negro

¡El mismo principio se aplica a todas las etiquetas HTML y a todas las propiedades CSS! Si dices:

- mis párrafos tienen un tamaño de 1.2 em;

- mis textos importantes (`<strong>`) tienen un tamaño de 1,4 em

... podrías pensar que podría haber un conflicto. Si el texto importante es parte de un párrafo, ¿qué tamaño debería dársele? ¿1,2 em ó 1,4 em? CSS decide que la declaración más específica tiene prioridad: ya que `<strong>` se corresponde con un elemento más específico que los párrafos, el texto estará escrito en 1,4 em.

Imágenes de fondo

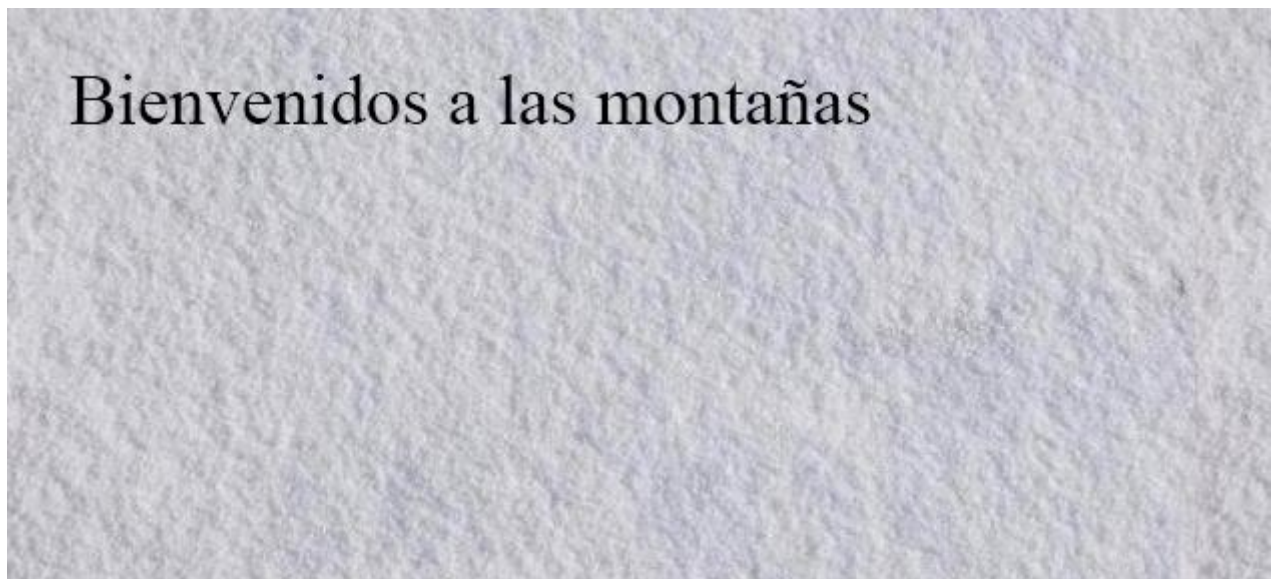
En los siguientes ejemplos, voy a cambiar la imagen de fondo de la página. Sin embargo, igual que para el color de fondo, no olvides que la imagen de fondo no se aplica necesariamente a toda la página. También puedes poner una imagen de fondo detrás de los títulos, párrafos, etc.

Aplicar una imagen de fondo.

La propiedad usada para especificar una imagen de fondo es `background-image`. Como valor, tienes que especificar `url ("nombreimagen1.png")` . Por ejemplo:

```
body
{
    background-image: url("snow.png");
}
```

Lo que da como resultado lo mostrado en la siguiente figura.



Una imagen de fondo en la página

Por supuesto, tu fondo no está necesariamente en PNG, pero también puede estar en formato JPEG o GIF.

La dirección que indica a ubicación de la imagen de fondo puede escribirse como una dirección completa (`http://...`) o como una dirección relativa (`fondo.png`).

¡Ten cuidado cuando escribas una dirección relativa en el archivo CSS! La dirección de la imagen debe ser especificada *relativa al archivo .css* y no relativa al archivo .html. Para simplificar las cosas, te aconsejo colocar la imagen en la misma carpeta que el archivo .css (o en una subcarpeta)

Opciones para la imagen de fondo

La propiedad de `background-image` que acabamos de ver puede suplementarse con otras propiedades que permitan que el comportamiento de la imagen de fondo cambie.

`background-attachment`: adjuntar el fondo

La propiedad `background-attachment` del CSS se usa para «adjuntar» el fondo. El efecto obtenido es interesante ya que puedes ver ahora el texto «slide» sobre el fondo. Hay disponibles dos valores:

- `fixed`: la imagen del fondo permanece fija;
- `scroll`: la imagen de fondo se desplaza con el texto (por defecto).

```
body
{
    background-image: url("snow.png");

    background-attachment: fixed; /* El fondo permanecerá fijo */
}
```

`background-repeat`: repetición del fondo

Por defecto, la imagen de fondo se repite en un patrón de mosaico. Puedes cambiar esto con la propiedad `background-repeat`:

- `no-repeat`: el fondo no se repetirá. La imagen será única en la página.
- `repeat-x`: el final se repetirá sólo en la primera línea, horizontalmente.
- `repeat-y`: el final se repetirá sólo en la primera columna, verticalmente.
- `repeat`: el fondo se repetirá en un patrón de mosaico (por defecto).

Ejemplo de uso:

```
body
{
```

```
background-image: url("sun.png");

background-repeat: no-repeat;

}
```

`background-position:` posición del fondo

Puedes especificar la posición requerida de la imagen de fondo con la `background-position`. Esta propiedad sólo es útil cuando se combina con `background-repeat: no-repeat;` (un fondo que no repite).

Tienes que introducir dos valores en píxeles para la `background-position` para especificar la posición de fondo relativa a la esquina superior izquierda de la página (o párrafo, si solicitas el fondo en un párrafo). Así, si introduces:

```
background-position: 30px 50px;
```

... tu fondo será situado a 30 píxeles de la izquierda y a 50 píxeles de la parte superior. También es posible usar estos valores:

- `top`: en la parte de arriba;
- `bottom`: en la parte de abajo;
- `left`: a la izquierda;
- `center`: centrado;
- `right`: a la derecha;

Estas palabras se pueden combinar. Por ejemplo, para alinear una imagen en la parte superior derecha, tendrías que escribir:

```
background-position: top right;
```

Así que si quiero mostrar un sol como imagen de fondo (figura siguiente), sólo una vez (`no-repeat`), siempre visible (`fixed`) y posicionado en la parte superior derecha (`top right`), escribiría esto:

```
body

{

background-image: url("sun.png");

background-attachment: fixed; /* El fondo permanecerá fijo */

background-repeat: no-repeat; /* El fondo no se repetirá */

background-position: top right; /* El fondo se mostrará en la parte superior
derecha */

}
```

}

Bienvenidos a las montañas



Ahora mismo está parcialmente nublado

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi eu lacus in libero sodales ultricies a at libero. Curabitur odio dolor, facilisis quis lacinia cursus, lobortis in nisl. Donec suscipit felis et libero euismod egestas. Aliquam a quam facilisis purus elementum laoreet vel id turpis. Vivamus ornare semper sapien ut malesuada. Nullam blandit nisi a purus auctor aliquet. Suspendisse porta felis sed nibh malesuada euismod. Nulla lacus leo, ornare at hendrerit eget, ullamcorper in velit. Nullam ornare consequat lectus id malesuada. Maecenas dapibus purus id odio posuere pharetra. Aenean congue auctor malesuada. Sed mi magna, rutrum semper vulputate ac, varius ut diam. Morbi urna metus, dictum vel feugiat a, suscipit nec mi. Etiam lobortis ligula eu massa vestibulum gravida. Maecenas iaculis ornare dictum. Curabitur vitae justo in elit vestibulum eleifend in nec odio. Duis rutrum nulla urna. Pellentesque ac orci eget dolor interdum suscipit. Aenean ullamcorper libero id ante consequat ut ornare risus consectetur. Cras aliquet hendrerit suscipit. Suspendisse at dapibus ipsum. Duis aliquet ligula at ligula pellentesque interdum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Aliquam tincidunt porttitor velit. Sed non justo eget turpis tempus bibendum nec non ligula. Duis velit augue, tempor dapibus lobortis et, luctus ac sem. Sed posuere ultrices eros non luctus. Etiam

Un sol como imagen de fondo en la esquina superior derecha

Propiedades de combinación

Si usas muchas propiedades en relación al fondo (como en el ejemplo anterior), puedes usar una especie de superpropiedad `background` cuyo valor puede combinar con varias propiedades ya vistas antes: `background-image`, `background-repeat`, `background-attachment` y `background-position`

Así podemos simplemente escribir:

```
body
{
    background: url("sun.png") fixed no-repeat top right;
}
```

Esta es la primera «superpropiedad» que te enseñé, pero habrá otras. Deberías saber lo siguiente:

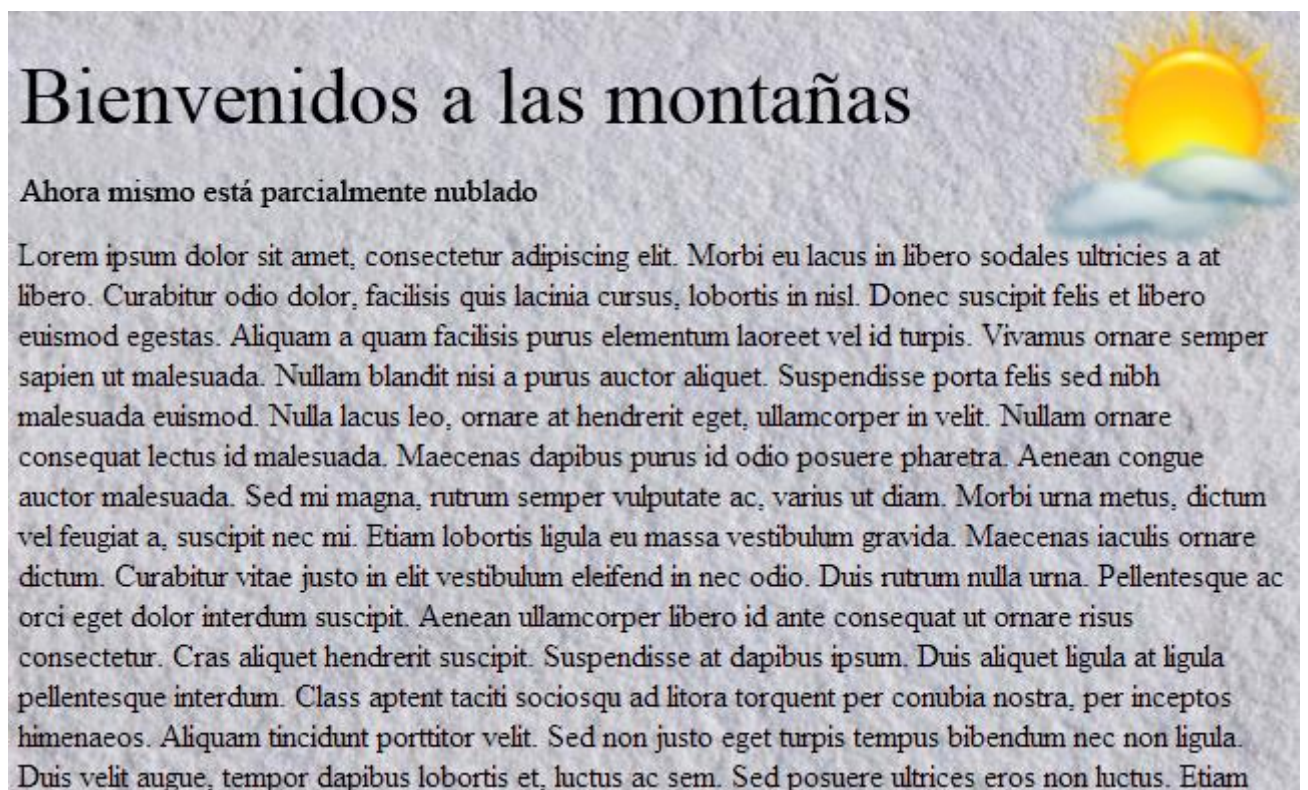
- El orden de los valores no importa. Puedes combinar los valores en cualquier orden.
- No tienes que introducir todos los valores. Si no quieres introducir `fixed`, puedes eliminarlo sin ningún problema.

Varias imágenes de fondo

Desde CSS3, puedes asignar varias imágenes de fondo a un elemento. Para hacer esto, simplemente separa las declaraciones por una coma así:

```
body{  
  
    background: url("sun.png") fixed no-repeat top right, url("snow.png") fixed;  
  
}
```

La primera imagen de esta lista se situará en la parte superior de las otras (siguiente figura). Ten cuidado, el orden de declaración de la imagen es importante: si intercambias el sol y la nieve en el código CSS anterior, ¡ya no verás el sol!



Fondo con múltiples imágenes

Ten en cuenta que múltiples imágenes de fondo funcionan en todos los navegadores excepto en las versiones más viejas de Internet Explorer, que sólo reconoce esta característica en la versión 9 (IE9).

Una última cosa antes de terminar con las imágenes de fondo: En todos estos ejemplos, he aplicado un fondo a la página completa (`body`). Sin embargo, esto no debería hacerte olvidar que puedes aplicar un fondo a cualquier elemento (un título, un párrafo, algunas palabras en un párrafo, etc.)

Para la práctica, te recomiendo intentar aplicar un fondo a tus títulos o párrafos. Si

tienes un poco de talento (no como yo), sin duda serás capaz de darle a tu página un aspecto genial.

Transparencia

¡CSS te permite muy fácilmente variar los niveles de transparencia de los elementos! Para hacer esto, usaremos las características CSS3: la propiedad `opacity` y la notación RGBA.

La propiedad de opacidad

Esta sencilla propiedad de `opacity` se usa para especificar el nivel de opacidad (lo contrario a transparencia).

- Con un valor de 1, el elemento será completamente opaco; este es el comportamiento por defecto.
- Con un valor de 0, el elemento será completamente transparente.

Así que tienes que elegir un valor entre 0 y 1. Con un valor de 0,6, tu elemento será un 60 % opaco... ¡y serás capaz de ver a través de él!

Aquí tienes cómo se puede usar:

```
p
{
    opacity: 0.6;
}
```

Aquí tienes un ejemplo que te dará una idea de transparencia. Puedes ver el resultado en la siguiente figura.

```
body
{
    background: url("snow.png");
}

p
{
```

```
background-color: black;

color: white;

opacity: 0.3;
}
```

Transparencia

Esto es un párrafo transparente. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi eu lacus in libero sodales ultricies a at libero. Curabitur odio dolor, facilisis quis lacinia cursus, lobortis in nisl. Donec suscipit felis et libero euismod egestas. Aliquam a quam facilisis purus elementum laoreet vel id turpis. Vivamus ornare semper sapien ut malesuada. Nullam blandit nisi a purus auctor aliquet. Suspendisse porta felis sed nibh malesuada euismod. Nulla lacus leo, ornare at hendrerit eget, ullamcorper in velit. Nullam ornare consequat lectus id malesuada. Maecenas dapibus purus id odio posuere pharetra. Aenean congue auctor malesuada. Sed mi magna, rutrum semper vulputate ac, varius ut diam. Morbi urna metus, dictum vel feugiat a, suscipit nec mi. Etiam lobortis ligula eu massa vestibulum gravida. Maecenas iaculis ornare dictum. Curabitur vitae justo in elit vestibulum eleifend in nec odio. Duis rutrum nulla urna. Pellentesque ac orci eget dolor interdum suscipit. Aenean ullamcorper libero id ante consequat ut ornare risus consectetur. Cras aliquet hendrerit suscipit. Suspendisse at dapibus ipsum. Duis aliquet ligula at ligula pellentesque interdum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Aliquam tincidunt porttitor velit. Sed non justo eget turpis tempus bibendum nec non ligula. Duis velit augue, tempor dapibus lobortis et, luctus ac sem. Sed posuere ultrices eros non luctus. Etiam adipiscing tellus vel purus accumsan ac dignissim dolor congue. Quisque mi justo, convallis luctus tempus nec, vulputate a augue. Cum sociis natoque penatibus et magnis dis parturient montes,

Un párrafo transparente

Ten en cuenta que la transparencia funciona en todos los navegadores, incluyendo Internet Explorer de IE9 en adelante.

Si aplicas la propiedad de `opacidad` a un elemento de la página, *todo* el contenido de este elemento se hará transparente (incluso imágenes, otros bloques dentro, etc.). Si sólo quieres hacer el color de fondo transparente, es mejor usar la notación RVAa de la que vamos a hablar ahora.

La notación RGBa

CSS3 nos ofrece otra manera de variar la transparencia: la notación RGBa. Esto es, en realidad, la notación RGB que vimos antes, pero con un cuarto ajuste: el nivel de transparencia (llamado canal alfa). Con un valor de 1, el fondo es completamente opaco igual que antes. Con un valor de menos de 1, es transparente.

```
p
{
```

```
background-color: rgba(255, 0, 0, 0.5); /* Rojo, fondo semitransparente */
}
```

Es tan simple como eso. Puedes conseguir exactamente el mismo efecto que con `opacidad` usando la notación `RGBA`. ¡Inténtalo!

Esta notación es reconocida por todos los navegadores más modernos, incluyendo Internet Explorer (de IE9 en adelante). Para los navegadores más viejos, se recomienda especificar la notación `RGB` convencional así como la `RGBA`.

Aunque para estos navegadores, el fondo no será transparente, al menos habrá color de fondo.

```
p
{
    background-color: rgb(255,0,0); /* Para los navegadores más viejos */
    background-color: rgba(255,0,0,0.5); /* Para los navegadores más nuevos */
}
```

En resumen

- Cambias el color del texto con la propiedad `color` y el color de fondo con `background-color`.
- Puedes especificar un color introduciendo su nombre (`black`, por ejemplo) en formato hexadecimal (`#FFC8D3`) o en notación `RGB` (`rgb(250, 25, 118)`).
- Puedes añadir una imagen de fondo con `background-image`. Puedes elegir adjuntar una imagen de fondo, mostrarla como mosaico o no, e incluso colocarla en cualquier lugar de la página.
- Puedes hacer una porción de la página transparente con la propiedad o con la notación `RGBA` (lo mismo que la notación `RGB`, con un cuarto valor que indica el nivel de transparencia).

Bordes y sombreado

Nuevo capítulo, nuevo conjunto de propiedades CSS. Vamos a ver los bordes y los efectos de sombreado que se pueden aplicar, tanto en el texto como en los bloques.

En particular, vamos a volver a usar nuestro conocimiento de los colores para elegir el color de nuestros bordes y sombreados.

¿Preparado para sorprenderte de nuevo?

Bordes estándar

CSS te ofrece una amplia selección de bordes para decorar tu página. Muchas propiedades CSS te permiten cambiar la apariencia de tus bordes: `border-width`, `border-color`, `border-style`...

Intento usar directamente la superpropiedad `border` que incluye todas estas propiedades. ¿Recuerdas la superpropiedad `background`? Funciona con el mismo principio: vamos a poder combinar varios valores.

Para el `borde`, puedes usar hasta tres valores para cambiar la apariencia del borde:

- **La anchura:** especifica la anchura de tu borde. Introduce un valor en píxeles (como `2px`).
- **El color:** es el color de tu borde. Como hemos aprendido, usa un nombre de color (`negro`, `rojo`, etc.) o un valor hexadecimal (`#FF0000`) o un valor RGB (`rgb(198, 212, 37)`).

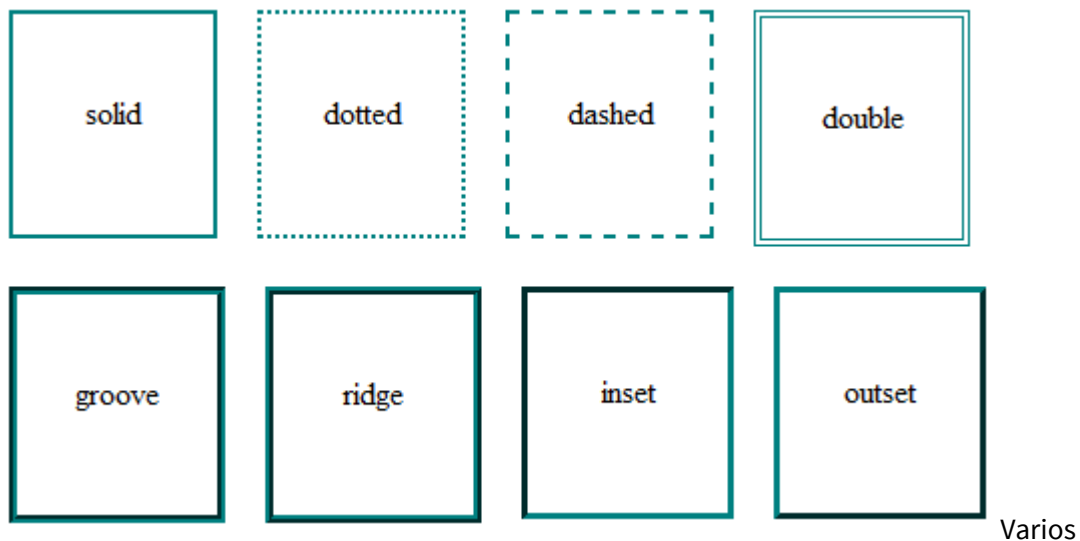
El tipo de borde: hay varias opiniones disponibles. Tu borde puede ser una única línea sólida, una línea de puntos o una línea discontinua. Aquí están los valores disponibles: `none`: sin borde (por defecto); `solid`: una sola línea sólida; `dotted`: una línea de puntos; `dashed`: una línea discontinua; `double`: con doble borde; `groove`: una línea estriada; `ridge`: una línea con crestas; `inset`: inset global efecto 3D; `outset`: outset global efecto 3D.

- **El tipo de borde:** hay varias opiniones disponibles. Tu borde puede ser una única línea sólida, una línea de puntos o una línea discontinua. Aquí están los valores disponibles:
 - `none`: sin borde (por defecto);
 - `solid`: una sola línea sólida;
 - `dotted`: una línea de puntos;
 - `dashed`: una línea discontinua;
 - `double`: con doble borde;
 - `groove`: una línea estriada;
 - `ridge`: una línea con crestas;
 - `inset`: inset global efecto 3D;
 - `outset`: outset global efecto 3D.

Para obtener un borde azul, una línea discontinua de 3 píxeles de grueso introduciré:

```
h1
{
    border: 3px blue dashed;
}
```

La siguiente figura muestra los diferentes estilos de bordes que puedes usar.



estilos de borde

En la parte superior, a la derecha, a la izquierda, en la parte inferior...

¿Quién dijo que tenías que aplicar el mismo borde a los cuatro lados de tu elemento?

Nada, si quieres poner bordes diferentes dependiendo del lado (arriba, abajo, derecha o izquierda), puedes hacerlo sin problema. En este caso, tienes que usar estas cuatro propiedades:

- `border-top`: borde superior;
- `border-bottom`: borde inferior;
- `border-left`: borde izquierdo;
- `border-right`: borde derecho;

Hay equivalentes para modificar cada detalle del borde si tú quieres: `border-top-width` para cambiar el grosor del borde superior, `border-top-color` para el color superior etc.

Son también superpropiedades que funcionan como `border`, pero se aplica a un solo lado.

Para añadir un borde a la izquierda o derecha de los párrafos, solo tenemos que introducir:

```
p
{
    border-left: 2px solid black;
    border-right: 2px solid black;
```

```
}
```

Puedes cambiar los bordes de cualquier elemento de la página. Lo hemos puesto en párrafos, pero también puedes cambiar el borde de imágenes, textos importantes como `<strong>` etc.

Bordes redondeados

Los bordes redondeados son un poco como los webmasters del Holy Grain que han estado esperando desde el milenio anterior (o eso parece). ¡Desde la llegada de CSS3, ahora es posible crearlos fácilmente!

La propiedad `border-radius` nos permitirá fácilmente redondear las esquinas de cualquier elemento. Simplemente especifica el tamaño («importancia») del radio en píxeles:

```
p
{
    border-radius: 10px;
}
```

El radio se ve especialmente si el elemento tiene bordes, como se muestra en la figura siguiente.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras ullamcorper sodales elit, sit amet pellentesque lectus aliquet quis. Etiam sem ipsum, rhoncus eu aliquam nec, mattis consectetur tortor. Mauris non lectus magna, vel interdum elit. Sed fermentum commodo commodo. Fusce imperdiet vestibulum neque, id pulvinar urna ultricies ullamcorper. Donec euismod, ipsum vehicula pretium tempor, mauris odio pellentesque metus, et ultrices arcu mauris sit amet leo. Curabitur ac scelerisque sem.

Bordes redondeados

... o si tiene un color de fondo, como en la figura siguiente.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras ullamcorper sodales elit, sit amet pellentesque lectus aliquet quis. Etiam sem ipsum, rhoncus eu aliquam nec, mattis consectetur tortor. Mauris non lectus magna, vel interdum elit. Sed fermentum commodo commodo. Fusce imperdiet vestibulum neque, id pulvinar urna ultricies ullamcorper. Donec euismod, ipsum vehicula pretium tempor, mauris odio pellentesque metus, et ultrices arcu mauris sit amet leo. Curabitur ac scelerisque sem.

Un fondo con esquinas redondeadas

También puedes especificar la forma del radio de cada esquina. En este caso, especifica cuatro valores:

```
p
{
    border-radius: 10px 5px 10px 5px;
}
```

Los valores corresponden a las siguientes esquinas en el orden indicado:

1. superior izquierda;
2. superior derecha;
3. inferior derecha;
4. inferior izquierda;

Finalmente, puedes definir la redondez de tus esquinas creando curvas elípticas como en la siguiente figura. En este caso, tienes que especificar dos valores separados por un carácter / (barra oblicua). Para ver el efecto, lo mejor es probarlo:

```
p
{
    border-radius: 20px / 10px;
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras ullamcorper sodales elit, sit amet pellentesque lectus aliquet quis. Etiam sem ipsum, rhoncus eu aliquam nec, mattis consectetur tortor. Mauris non lectus magna, vel interdum elit. Sed fermentum commodo commodo. Fusce imperdiet vestibulum neque, id pulvinar urna ultricies ullamcorper. Donec euismod, ipsum vehicula pretium tempor, mauris odio pellentesque metus, et ultrices arcu mauris sit amet leo. Curabitur ac scelerisque sem.

Bordes redondeados elípticos

Los bordes redondeados funcionan con todos los nuevos navegadores, incluyendo Internet Explorer desde su versión 9 (IE9).

Para las versiones más antiguas de Mozilla Firefox, Chrome y Safari, los llamados «prefijos del proveedor» tienen que ser usados, en otras palabras, tienes que escribir varias versiones de la propiedad (-moz-border-radius para Firefox, -webkit-border-radius para Safari, etc.) en el código CSS. Afortunadamente esto ya no es necesario, a menos que quieras soportar las versiones más antiguas de estos navegadores.

¿Quién dijo que tenías que aplicar el mismo borde a los cuatro lados de tu elemento? Nada, si quieres poner bordes diferentes dependiendo del lado (arriba, abajo, derecha o izquierda), puedes hacerlo sin problema. En este caso, tienes que usar estas cuatro propiedades:

- `border-top`: borde superior;
- `border-bottom`: borde inferior;
- `border-left`: borde izquierdo;
- `border-right`: borde derecho;

Hay equivalentes para modificar cada detalle del borde si tú quieres: `border-top-width` para cambiar el grosor del borde superior, `border-top-color` para el color superior etc.

Son también superpropiedades que funcionan como `border`, pero se aplica a un solo lado.

Para añadir un borde a la izquierda o derecha de los párrafos, solo tenemos que introducir:

```
p
{
    border-left: 2px solid black;

    border-right: 2px solid black;
}
```

Puedes cambiar los bordes de cualquier elemento de la página. Lo hemos puesto en párrafos, pero también puedes cambiar el borde de imágenes, textos importantes como `<strong>` etc.

Sombreado

Las sombras se incluyen en las últimas innovaciones ofrecidas por CSS3. ¡Ahora sólo necesitas una única línea de CSS para añadir sombras a tu página!

Aquí vamos a conocer dos tipos de sombras:

- sombras de caja;
- sombras de texto;

`box-shadow`: sombras de caja

La propiedad `box-shadow` se aplica a todo el bloque y tiene cuatro valores en el siguiente orden:

1. el offset horizontal de la sombra;
2. el offset vertical de la sombra;
3. la bajada de tono;
4. el color de la sombra.

Por ejemplo, para un negro, sombra de 6 píxeles, sin bajada de tono, introduciríamos:

```
p
{
    box-shadow: 6px 6px 0px black;
}
```

El resultado que esto tendría se muestra en la siguiente figura (añadí un borde al párrafo para mostrar mejor el efecto).

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras ullamcorper sodales elit, sit amet pellentesque lectus aliquet quis. Etiam sem ipsum, rhoncus eu aliquam nec, mattis consectetur tortor. Mauris non lectus magna, vel interdum elit. Sed fermentum commodo commodo. Fusce imperdiet vestibulum neque, id pulvinar urna ultricies ullamcorper. Donec euismod, ipsum vehicula pretium tempor, mauris odio pellentesque metus, et ultrices arcu mauris sit amet leo. Curabitur ac scelerisque sem.

Sombra bajo el párrafo

Bajemos el tono de la sombra usando el tercer ajuste (siguiente figura). La bajada de tono puede ser ligera (menos que el offset), normal (igual al offset) o fuerte (mayor que el offset). Vamos a probar un offset normal:

```
p
{
    box-shadow: 6px 6px 6px black;
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras ullamcorper sodales elit, sit amet pellentesque lectus aliquet quis. Etiam sem ipsum, rhoncus eu aliquam nec, mattis consectetur tortor. Mauris non lectus magna, vel interdum elit. Sed fermentum commodo commodo. Fusce imperdiet vestibulum neque, id pulvinar urna ultricies ullamcorper. Donec euismod, ipsum vehicula pretium tempor, mauris odio pellentesque metus, et ultrices arcu mauris sit amet leo. Curabitur ac scelerisque sem.

Una sombra redondeada bajo el párrafo

También puedes añadir un cuarto valor opcional: `inset`. En este caso, la sombra estará situada dentro del bloque, para dar un efecto de inset:

```
p
{
    box-shadow: 6px 6px 6px black inset;
}
```

Te dejo probarlo para que veas el resultado.

La propiedad `box-shadow` funciona en todos los navegadores nuevos, incluyendo IE9. Para algunos navegadores, especialmente navegadores móviles, todavía hay que añadir un prefijo. Así, tienes que introducir una versión `-webkit-box-shadow` para que funcione en los navegadores Android e iOS.

text-shadow: sombra del texto

Con `text-shadow`, puedes añadir una sombra directamente a los letras de tu texto. Los valores trabajan exactamente de la misma manera que la `box-shadow: offset`, bajada de tono y color.

```
p
{
    text-shadow: 2px 2px 4px black;
}
```

El resultado se muestra en la figura siguiente.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras ullamcorper sodales elit, sit amet pellentesque lectus aliquet quis. Etiam sem ipsum, rhoncus eu aliquam nec, mattis consectetur tortor. Mauris non lectus magna, vel interdum elit. Sed fermentum commodo commodo. Fusce imperdiet vestibulum neque, id pulvinar urna ultricies ullamcorper. Donec euismod, ipsum vehicula pretium tempor, mauris odio pellentesque metus, et ultrices arcu mauris sit amet leo. Curabitur ac scelerisque sem.

Texto con sombra

Esta propiedad es reconocida por todos los navegadores más nuevos excepto por Internet Explorer, que sólo la reconoce desde la versión IE10.

En resumen

- Puedes aplicar un borde a un elemento usando la propiedad `border`. Tienes que especificar la anchura del borde, el color, y el tipo (línea sólida, de puntos, etc.).
- Puedes redondear los bordes usando `border-radius`.

- Puedes añadir una sombra a los bloques de texto usando `box-shadow`. Tienes que especificar la sombra vertical y horizontal, su tono y su color.
- Al texto también se le puede dar una sombra usando `text-shadow`.

Crear apariencias dinámicas

Este es uno de sus puntos fuertes: CSS también nos permite combinar la apariencia de los elementos dinámicamente, en todas palabras, los elementos pueden cambiar la forma una vez que la página se ha cargado. Vamos a usar una característica poderosa del CSS: los pseudoformatos.

En este capítulo vamos a ver cómo cambiar la apariencia:

- al desplazarse;
- al hacer clic;
- al centrarse (elemento seleccionado);
- cuando se ha visitado un enlace;

¡Verás que CSS todavía tiene muchas sorpresas!

Al desplazarse

En este capítulo vamos a descubrir varios pseudoformatos CSS. Lo primero que te voy a mostrar se llama `:hover`. Como todos los otros pseudoformatos que vamos a ver, se trata de información añadida después del nombre de etiqueta (o clase) en el CSS, así:

```
a:hover  
  
{  
  
  
  
}
```

`:hover` puede interpretarse como: «Cuando el ratón se coloca encima del enlace» (cuando señalas hacia él).

Depende de ti definir la apariencia que el enlace tiene cuando haces clic sobre él. Dale rienda suelta a tu imaginación, no hay límite.

Aquí tienes un ejemplo de presentación de enlace, pero siéntete libre de inventar la tuya propia:

```
a /* Enlaces por defecto (cuando no se pasa por encima) */
```

```

{

    text-decoration: none;

    color: red;

    font-style: italic;

}

a:hover /* Apariencia de los enlaces cuando se pasa por encima */

{

    text-decoration: underline;

    color: green;

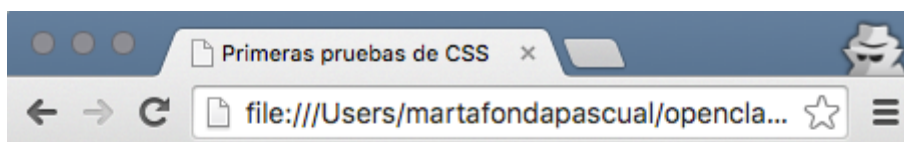
}

```

Hemos definido dos versiones de estilos para los enlaces:

- para enlaces por defecto (cuando no se pasa por encima)
- para enlaces cuando se pasa por encima.

El resultado se muestra en la figura siguiente.



Texto sombreado

Conoces *Google*? Es el buscador más usado del mundo

Conoces W3C? Son las personas que definen HTML y CSS

apariencia cuando se pasa por encima el ratón

Bonito, ¿no?

Aunque normalmente se usa en enlaces, puedes cambiar la apariencia de cualquier elemento. Por ejemplo, puedes cambiar la apariencia de párrafos cuando pasas por encima:

```
p:hover /* Cuando señalas un párrafo */  
  
{  
  
  
  
}
```

Al hacer clic y seleccionar

Puedes interactuar más en CSS. ¡Vamos a ver que podemos cambiar la apariencia de los elementos cuando hacemos clic sobre ellos o cuando los seleccionamos!

:active: cuando se hizo clic

El seudoformato `:active` te permite aplicar un estilo particular *cuando haces clic*. En la práctica, sólo se usa en enlaces.

El enlace no mantendrá esta apariencia durante mucho tiempo: de hecho, el cambio se produce cuando se pulsa la tecla del ratón. Claramente, esto no siempre será visible.

Puedes, por ejemplo, cambiar el color del fondo del enlace cuando lo has pulsado.

```
a:active /* Cuando el visitante hace clic en el enlace */  
  
{  
  
    background-color: #FFCC66;  
  
}
```

:focus: cuando se selecciona un elemento

En este caso, es un poco diferente. El seudoformato `:focus` aplica un estilo *cuando se selecciona un elemento*.

¿En otras palabras?

Una vez pulsado, el enlace permanece «seleccionado» (tiene un pequeño borde de puntos alrededor). Esa es la selección.

Este seudoformato se puede aplicar a otras etiquetas HTML que todavía no hemos visto, como los elementos de forma.

Intentémoslo en enlaces:

```
a:focus /* Cuando el visitante selecciona el enlace */  
  
{  
  
    background-color: #FFCC66;  
  
}
```

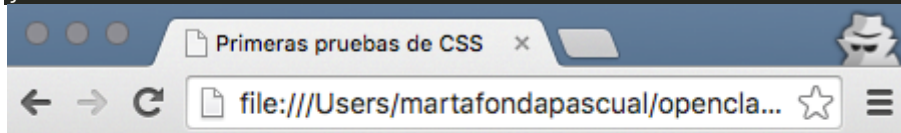
En Google Chrome y Safari, el efecto sólo se ve cuando pulsas la tecla Tab.

Cuando el enlace ya se ha visitado

Puedes aplicar un estilo a un enlace a una página que ya se ha visitado. Por defecto el navegador cambia el color del enlace a un púrpura bastante feo (en mi opinión, al menos).

Puedes cambiar esta apariencia con `:visited`. En la práctica, para los enlaces visitados, no puedes cambiar mucho aparte del color (siguiente figura).

```
a:visited /* Cuando el visitante ya ha visto la página correspondiente */  
  
{  
  
    color: #AAA; /* Aplicar un color gris */  
  
}
```



Texto sombreado

Conoces *Google*? Es el buscador más usado del mundo

Conoces *W3C*? Son las personas que definen HTML y CSS

en gris Enlaces visitados

Si no quieres que los enlaces ya visitados tengan otro color, tienes que aplicarles el mismo color que a los enlaces normales. Muchas páginas web hacen esto. Una excepción importante: Google... que es bastante manejable, como puedes ver en los resultados de búsqueda si has o no visitado las páginas web encontradas por Google.

En resumen

- En CSS, puedes cambiar la apariencia de algunas secciones dinámicamente, después de que la página se cargue, cuando suceden algunos eventos. Con este fin, se usan los pseudoformatos.
- El pseudoformato `:hover` se usa para cambiar la apariencia al pasar el ratón por encima (por ejemplo: `a:hover` para cambiar la apariencia de los enlaces cuando el ratón pasa por encima).
- El pseudoformato `:active` cambia la apariencia de los enlaces cuando haces clic en ellos, `:visited` cuando un enlace ya ha sido visitado.
- El pseudoformato `:focus` se usa para cambiar la apariencia de un elemento seleccionado.

Quiz 2

• Pregunta 1

¿Qué código CSS se podría utilizar para aplicar un estilo al nivel 2 y nivel 3 de los encabezados?

- ☐ `h2 + h3`
- ☐ `h2 h3`
- ☐ `.h2 .h3`
- ☐ `h2, h3`

• Pregunta 2

La `class` attribute tiene una característica especial relacionada con el `id` attribute. ¿Cuál es?

- ☐ En una web, un `id` puede ser utilizado sólo una vez, pero la misma `class` puede ser utilizada varias veces
- ☐ Versiones anteriores de HTML soportan la `class` attribute pero no el `id` attribute
- ☐ La `class` attribute puede ser utilizada en un archivo `.css`, el `id` attribute no.
- ☐ No hay ninguna diferencia entre estos dos atributos.

• Pregunta 3

¿Podrías utilizar la fuente que quieras en tu página web?

- ☐ ☐ Sí, sólo tendrías que utilizar `@font-face`
- ☐ ☐ Sí, pero sólo si utilizas una fuente alojada en Google Web Fonts
- ☐ ☐ Sí, pero el visitante tendrá que descargar la fuente e instalarla manualmente.
- ☐ ☐ De ninguna manera.

• Pregunta 4

¿Cuál de los siguientes trozos de código hacen que el texto ocupe todo el ancho disponible?

- ☐ ☐ Left
- ☐ ☐ Center
- ☐ ☐ Right
- ☐ ☐ Justify

• Pregunta 5

¿Cuál de estos códigos de color se refiere al blanco?

- ☐ ☐ #000000
- ☐ ☐ rgb(100,100,100)
- ☐ ☐ #FFFFFF
- ☐ ☐ purple

• Pregunta 6

Si establecemos la opacidad de un elemento a 1, ¿Qué ocurre?

- ☐ ☐ El elemento desaparece
- ☐ ☐ El elemento será medio transparente
- ☐ ☐ El elemento será ligeramente transparente
- ☐ ☐ El elemento se mostrará opaco.

• Pregunta 7

Un párrafo de texto puede tener imagen de fondo. ¿Verdadero o falso?

- ☐ Verdadero
- ☐ Falso

• Pregunta 8

¿Cuál de estas propiedades CSS te permite añadir sombreado a las letras?

- ☐ box-shadow
- ☐ text-shadow
- ☐ letter-shadow
- ☐ shadow

• Pregunta 9

¿Cuál de estas reglas CSS te permite cambiar la apariencia de un link que ha sido pinchado anteriormente?

- ☐ a:visited
- ☐ a.consulted
- ☐ a:focus
- ☐ a:consulted

• Pregunta 10

¿Qué ocurre si un elemento se establece como `float` ?

- ☐ Que se mueve en la página
- ☐ Que se establece en la región superior izquierda de la pantalla
- ☐ que el texto se envuelve alrededor del elemento flotante
- ☐ La propiedad float no existe

html5es-part2 Assessment

¡Ahora te toca mejorar el estilo del CV que comenzaste en el primer ejercicio!

Vas a trabajar en el fichero HTML que creaste con tu CV para:

- cambiar el color de un elemento de texto
- cambiar la alineación de un párrafo o título
- incluir una imagen de fondo de página
- utilizar una fuente personalizada via `@font-face`

- añadir bordes a un elemento
- añadir sombras a un elemento

Por supuesto puedes hacer algunos cambios en tu fichero HTML si lo necesitas. Pero debes utilizar un archivo externo CSS e incluirlo en tu fichero HTML en el ZIP que tienes que enviar. No olvides incluir tus imágenes y fuentes en el ZIP también.

Estructurar tu página

Estamos gradualmente llegando al punto. Si nuestra página web no se parece mucho a las páginas que conocemos es porque aún no sabemos cómo diseñar la página.

En general, una página web consiste en un encabezamiento (parte superior derecha), menús de navegación (en la parte superior o en los lados), varias secciones en el centro... y un pie de página (abajo a la derecha).

En este capítulo vamos a ver las nuevas etiquetas HTML para estructurar nuestra página. Estas etiquetas fueron introducidas por HTML5 (antes no estaban disponibles) y nos permitirán decir: «Este es mi encabezado», «este es mi menú de navegación», etc.

De momento no vamos a hacer ningún diseño de página. De hecho, vamos a preparar nuestro documento HTML para ser capaces de hablar del diseño de la página web en los siguientes capítulos.

Etiquetas estructurales HTML5

Voy a mostrarte las nuevas etiquetas introducidas por HTML5 para estructurar nuestras páginas. Verás que no cambiará el aspecto de nuestra página mucho, pero estará bien construida y perfecta para mostrar después.

`<header>`: *el encabezado*

La mayoría de las páginas web normalmente tienen un *encabezado*. La mayoría suelen contener un logo, un banner, el eslogan de tu web, etc.

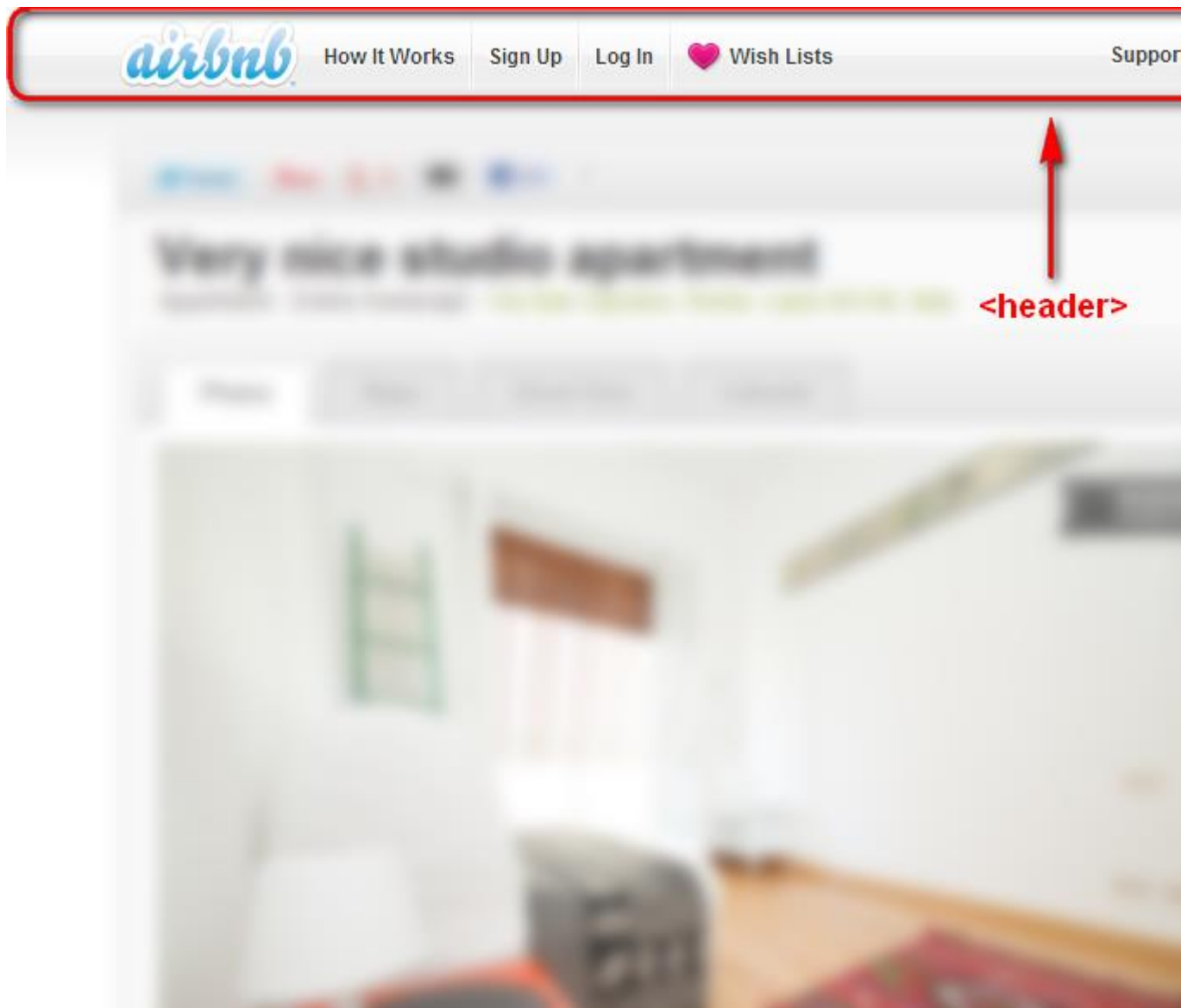
Tienes que colocar esta información dentro de la etiqueta `<header>`:

```
<header>

  <!-- Pon tu encabezado aquí -->

</header>
```

La figura siguiente, por ejemplo, muestra parte de la página de OpenClassrooms y puedes ver su encabezado.



Etiqueta de encabezado

El encabezado puede contener lo que quieras: imágenes, enlaces, textos, etc.

Tu página puede contener varios encabezados. Si se divide en varias secciones, cada sección puede tener su propio `<header>`.

`<footer>`*pie de página*

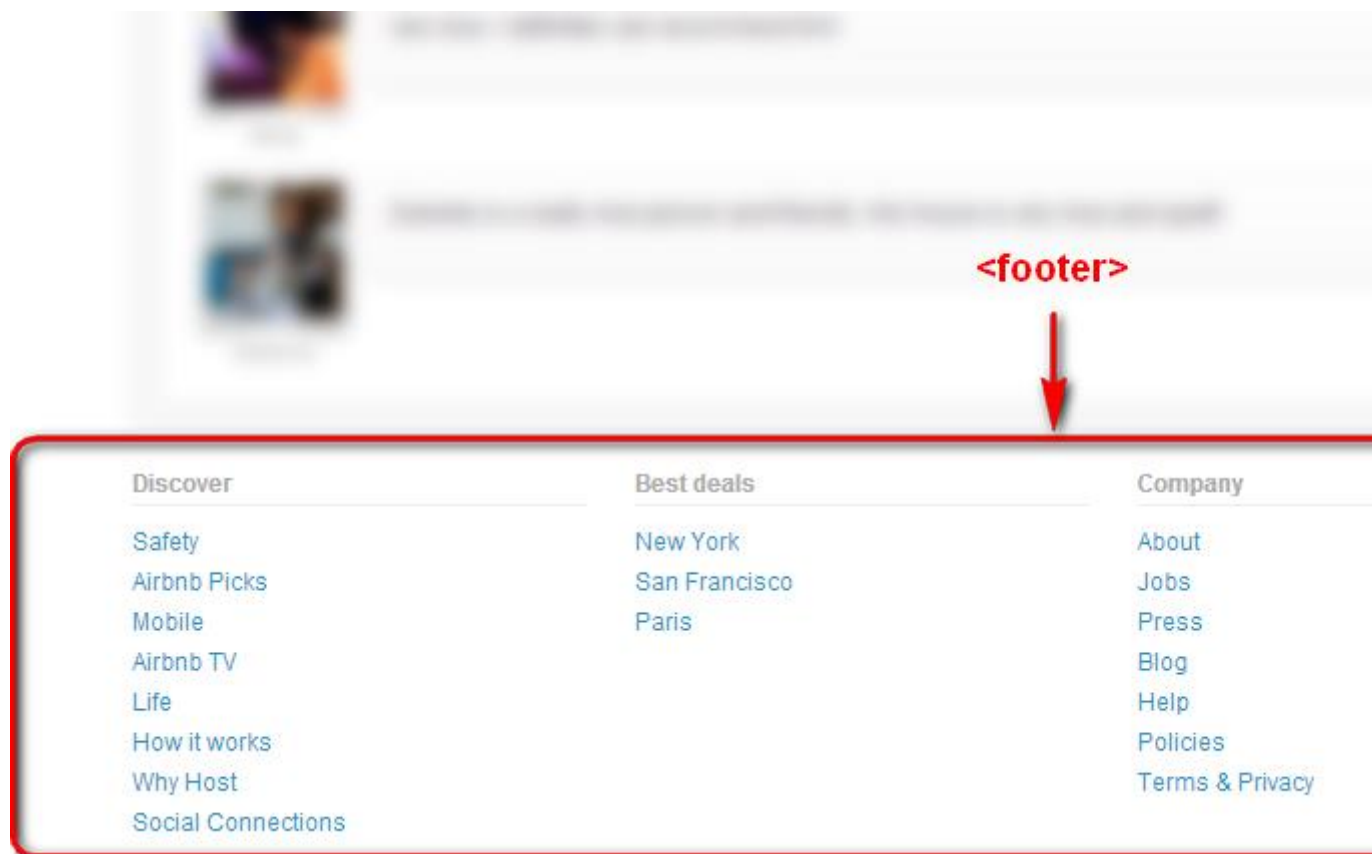
A diferencia del encabezado, el pie de página normalmente está en la parte inferior derecha del documento. Contiene información como enlaces de contacto, nombre del autor, informes legales, etc.

```
<footer>

  <!-- Pon tu pie de página aquí -->

</footer>
```

La figura siguiente muestra cómo es el el pie de página de la página de OpenClassrooms.



Etiqueta de pie de página

<nav>: enlaces de navegación principales

La etiqueta `<nav>` tiene que incluir todos los enlaces de navegación principales de la página. Por ejemplo, contendría el menú principal de tu página web.

El menú normalmente toma la forma de una lista con viñetas dentro de la etiqueta `<nav>`:

```
<nav>

  <ul>

    <li><a href="index.html">Inicio</a></li>

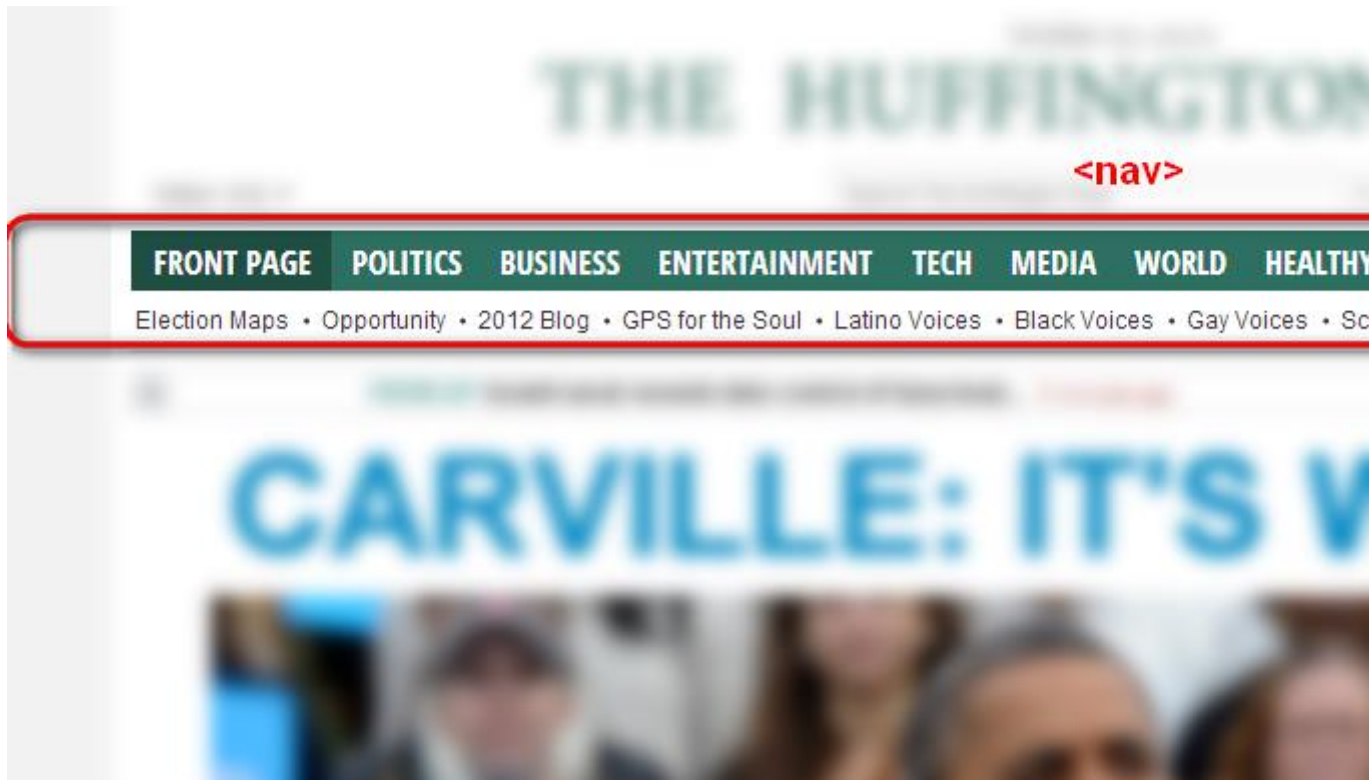
    <li><a href="forum.html">Foro</a></li>

    <li><a href="contact.html">Contacto</a></li>

  </ul>

</nav>
```

En la figura siguiente, puedes ver el menú de navegación principal, que podría usar la etiqueta `<nav>`.



Etiqueta Nav

`<section>`: una sección de la página

La etiqueta `<section>` incluye una porción del contenido según su tema. Generalmente cubre una porción del contenido en el centro de la página.

```
<section>

  <h1>Sección de mi página</h1>

  <p>Bla bla bla bla</p>

</section>
```

En la página de inicio de Wikipedia, hay varios bloques que podrían considerarse secciones de página (figura siguiente).

Today's featured article

<section>



Andjar Asmara (1902–1961) was a dramatist and filmmaker active in the cinema of the [Dutch East Indies](#). Born in [West Sumatra](#), he first found employment as a reporter in [Batavia](#) (modern-day Jakarta) before becoming a writer for the Padangsche Opera in [Padang](#). In Padang he developed a new, dialogue-centric style which later spread throughout the region. After returning to Batavia in 1929 he spent several years as a theatre and film critic before joining the [Dardanella touring troupe](#). He left Dardanella in 1936 to establish his own troupe and subsequently worked at a publisher, writing [serials](#) based on successful films. In 1940 he was asked to join [The Teng Chun's](#) company, Java Industrial Film, helping with marketing and working as a director for two productions. Andjar directed three films in the late 1940s and wrote four [screenplays](#) before spending the remainder of his life writing serials based on local films and publishing film criticism. Historians recognise him as a pioneer of theatre and one of the first [native Indonesian](#) film directors. ([more...](#))

Recently featured: [United States v. Wong Kim Ark](#) – Sarah Churchill, Duchess of Marlborough – Mercury dime

[Archive](#) – [By email](#) – [More featured articles...](#)

Did you know...

<section>

From Wikipedia's *newest content*:

- ... that the [lapis lazuli](#) inlays used in the 25th-century BC *[Statue of Ebih-II](#)* (*[pictured](#)*), discovered in ancient Mari in [Syria](#), were imported from as far east as [Afghanistan](#)?
- ... that the 1898 torpedo boat *[Storm](#)* was the only Norwegian warship to fire a [torpedo](#) at the invading Germans during the [1940 Norwegian Campaign](#)?
- ... that *[LW5/7 classified skiers](#)* use two skis and no ski poles while competing in skiing disciplines like the [downhill](#) and [Giant Slalom](#)?
- ... that *[Vehbi Dibra](#)* was the first chairman of the [Albanian senate](#)?
- ... that in Goya's *[Portrait of the Duchess of Alba](#)*, the duchess is wearing her mourning [clothes](#), as her husband died the year before?
- ... that German historian *[Hellmut G. Haasis](#)*, who won the [Schubart Literature Prize](#) for his biography of [Joseph Süß Oppenheimer](#), also performs for school children as "Druiknui, the fairy-tale clown"?



[Archive](#) – [Start a new article](#) – [Nominate an article](#)

In the new

- The discov...
impression...
closest to
- Hilary Mar...
time in fou
- Pakistani...
week, is s
- Norodom
- Austrian s...
balloon flig
- Syrian ci

On this da

October 20: E

- 1572 – Eig...
waded acr...
in water to
- Spanish N
- 1917 – The...
Petrograd...
the Russia
- 1951 – Afr...
victim of a...
football rul
- guards.
- 1961 – The...
launched...
Golf class
- 1967 – Ro...
at Six Rive

More annivers

Etiqueta de sección

Cada sección puede tener su nivel 1 título (`<h1>`), y el encabezado puede también incluir el título `<h1>`. Cada uno de estos bloques es independiente de los demás, y es habitual encontrar varios títulos `<h1>` en el código de la página web. Así «El título `<h1>` del `<header>`», «El título `<h1>` de esta `<section>`», etc.

<aside>: información adicional

La etiqueta `<aside>` está diseñada para contener información adicional al documento visto. Esta información está normalmente situada en el lado (aunque no tiene que ser así).

```
<aside>

  <!-- Pon aquí más información sobre tu contenido -->

</aside>
```

La página podría contener varios bloques `<aside>`.

En Wikipedia, por ejemplo, el artículo que se está viendo a menudo tiene un bloque de información adicional a la derecha. Así, la página que describe una estrella (siguiente figura) contiene un bloque dando las características de la estrella (dimensiones, peso, etc.).

Etiqueta aparte

<article>: un artículo independiente

La etiqueta `<article>` se usa para cubrir una porción de la página que generalmente es independiente. Es una parte de la página que podría ser rechazada en otra página. Este es el caso, por ejemplo, de las noticias (artículos en periódicos o blogs).

```
<article>

  <h1>Mi artículo</h1>

  <p>Bla bla bla bla</p>

</article>
```

Por ejemplo, un artículo en la página de OpenClassrooms se muestra en la siguiente figura.



Mars' Shiny Particles: Curiosity Rover Digs Up Unidentified Material, NASA Scientists Say

Posted: 10/18/2012 3:52 pm Updated: 10/18/2012 3:52 pm

Like 1,021 people like this. Sign Up to see what your friends like.



402

51

28

94

1172

share

tweet

+1

email

comment

GET SCIENCE ALERTS:

Enter email

SIGN UP

REACT: [Amazing](#) [Inspiring](#) [Funny](#) [Scary](#) [Hot](#) [Crazy](#) [Important](#) [Weird](#)

FOLLOW: [Video](#), [Planetary Science](#), [Curiosity](#), [Curiosity Shiny Particles](#), [Mars](#), [Mars Rocks](#), [Mars Shiny Particles](#), [Shiny Particles Mars](#), [Space](#), [Science News](#)

By: SPACE.com Staff

Published: 10/18/2012 10:14 AM EDT on SPACE.com

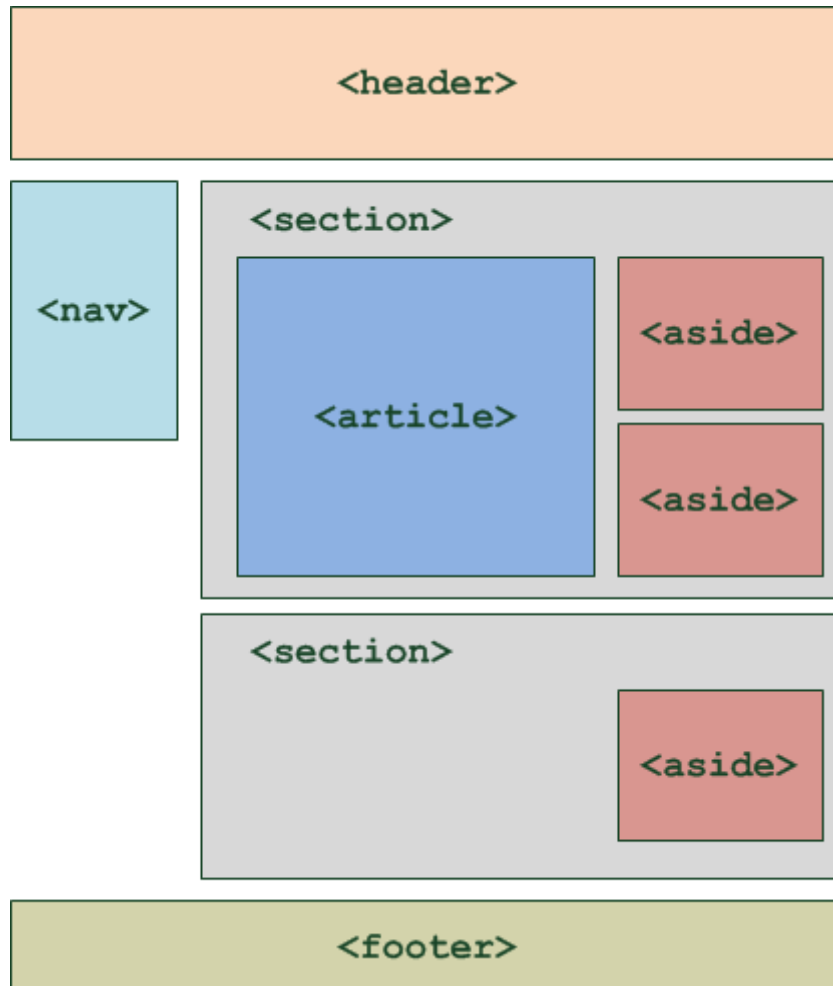
NASA's Mars rover Curiosity has found some more bright stuff on the Red Planet, scientists say.

Etiqueta de artículo

Resumen

Uf, son muchas etiquetas nuevas para recordar.

¡Afortunadamente, te he hecho un pequeño diagrama (figura siguiente) para ayudarte a recordar lo que hacen!



Secciones de página

identificadas por etiquetas

No cometas errores: este diagrama te ofrece un *ejemplo* de organización de página. Tú decides si pones tu menú de navegación en la derecha, o en la parte de arriba, o tus etiquetas **<aside>** arriba, etc.

Podías imaginar un segundo **<header>**, esta vez ubicado dentro de una **<section>**. En este caso, se consideraría la sección encabezado.

Finalmente, una sección no tiene que contener necesariamente una etiqueta de **<article>** y unas etiquetas **<aside>**. Usa solo estas etiquetas si las necesitas. Nada te impide crear secciones que contengan sólo párrafos, por ejemplo.

Ejemplo práctico del uso de etiquetas

Usemos las etiquetas que acabamos de descubrir en la estructura de nuestra página web. El siguiente código muestra todas las etiquetas que acabamos de ver, en una página web completa:

```
<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8" />

    <title>Zozor - La página web</title>

  </head>

  <body>

    <header>

      <h1>Zozor</h1>

      <h2>Diarios de viaje</h2>

    </header>

    <nav>

      <ul>

        <li><a href="#">Inicio</a></li>

        <li><a href="#">Blog</a></li>

        <li><a href="#">Resumen</a></li>

      </ul>

    </nav>

    <section>
```

```

<aside>

    <h1>Sobre el autor</h1>

    <p>¡Soy yo, Zozor! Nací el 23 de noviembre de 2005.</p>

</aside>

<article>

    <h1>Soy un gran viajero</h1>

    <p>Bla bla bla bla (texto del artículo)</p>

</article>

</section>


<footer>

    <p>Zozor Copyright - Todos los derechos reservados<br />

    <a href="#">¡Contacta conmigo!</a></p>

</footer>


</body>

</html>

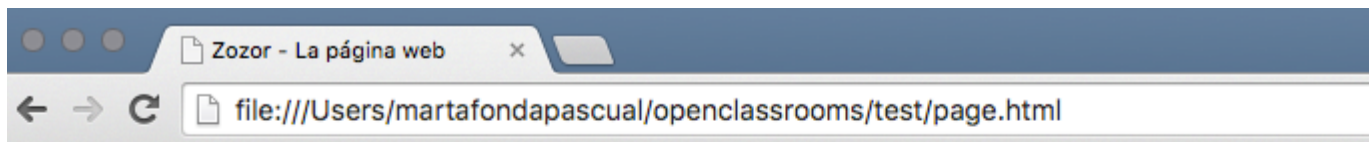
```

Este código puede ayudarte a comprender cómo las etiquetas deben colocarse. Puedes ver que contiene un encabezado, un menú de navegación, un pie de página, etc., y, en el centro, una sección con un artículo y un bloque `<aside>` que ofrece información sobre el autor del artículo.

¿A qué se se parece la página que acabamos de crear?

¡A nada!

Si compruebas el resultado, verás texto negro sobre fondo blanco (figura siguiente). ¡Es normal, no hay CSS! La página está, sin embargo, bien estructurada, lo que encontramos útil para el siguiente paso.



Zozor

Diarios de viaje

- [Inicio](#)
- [Blog](#)
- [Resumen](#)

Sobre el autor

¡Soy yo, Zozor! Nací el 23 de noviembre de 2005.

Soy un gran viajero

Bla bla bla bla (texto del artículo)

Zozor Copyright - Todos los derechos reservados
[¡Contacta conmigo!](#)

Una página que está bien estructurada pero sin CSS

Los enlaces son intencionadamente ficticios (por eso la presencia de un simple #) y por lo tanto llevan a ningún lado (bueno, es sólo una página demo).

No entiendo el objetivo de estas etiquetas. ¡Podrías obtener también el mismo resultado sin usarlas!

Es verdad. De hecho, estas etiquetas sólo están ahí para explicarle al ordenador «Este es el encabezado», «este es mi pie de página», etc. No especifican, al contrario de lo que podríamos pensar, dónde debería estar colocado el contenido. Eso es lo que hace CSS, como vamos a ver.

Para ser honesto, estas etiquetas no son muy útiles por ahora. También podrías usar las etiquetas genéricas `<div>` para incluir las diferentes porciones de nuestro contenido. De hecho, así es cómo fue hecho antes de la llegada de estas nuevas etiquetas HTML5.

Sin embargo, es muy posible que en el futuro cercano, los ordenadores empiecen a sacar beneficio de estas nuevas etiquetas. ¡Podrías imaginar, por ejemplo, un navegador que elija mostrar los enlaces de navegación `<nav>` para que siempre estuvieran visibles! Una vez que el ordenador «comprende» la estructura de la página, todo se vuelve posible.

Asegurar la compatibilidad con IE

Las nuevas etiquetas que acabamos de ver sólo han sido reconocidas por Internet Explorer desde su versión 9 (IE9). Esto va a ser un problema porque las versiones antiguas de IE no conocen la etiqueta... no gestionan la página correctamente (no puedes editar su CSS, por ejemplo).

Afortunadamente, esto puede ser resuelto fácilmente usando un script JavaScript. Los scripts son pequeños trozos de código que le permiten al usuario interactuar con la web y realizar algunas acciones. Como no nos vamos a preocupar por el JavaScript (no es materia de este curso), ten en cuenta que este tipo de script se llama en las páginas HTML de forma parecida a como los archivos CSS son llamados.

Los archivos JavaScript normalmente tienen la extensión `.js`. En el código HTML, normalmente estás citados en la etiqueta `<head>` con esta etiqueta:

```
<script src="myscript.js"></script>
```

La razón por la que te estoy dando una visión general de JavaScript es que un pequeño trozo de script con el bonito nombre `HTML5shiv` te permite asegurarte de que las etiquetas que acabamos de ver (`<header>`, `<footer>`, `<section>`...) se muestran correctamente en las versiones más antiguas de Internet Explorer (IE6, IE7, IE8).. En práctica, simplemente añades las siguientes líneas a tu código:

```
<!--[if lt IE 9]>

    <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>

<![endif]-->
```

Introduce este código en la etiqueta `<head>` como esta:

```
<!DOCTYPE html>

<html>

    <head>

        <meta charset="utf-8" />

        <!--[if lt IE 9]>

            <script
src=>http://html5shiv.googlecode.com/svn/trunk/html5.js<</script>

        <![endif]-->
```

```
<title>Inclusión de HTML5shiv</title>

</head>

<body>

</body>

</html>
```

¡Y aquí tienes, tu página web ahora se mostrará sin problemas en Internet Explorer!

Sin entrar en detalles, deberías saber que `<!--[if lt IE 9]>` es un **comentario condicional**. Este es un comentario especial que es leído por Internet Explorer. Asegura que el script sólo funciona en las versiones de Internet Explorer anteriores a IE9 (otros navegadores no lo necesitan e ignoran este comentario).

En resumen

- Varias etiquetas fueron introducidas con HTML5 para delienar las diferentes áreas al construir la página web:
 - `<header>`: encabezado;
 - `<footer>`: pie de página;
 - `<nav>`: principaes enlaces de navegación;
 - `<section>`: sección de página;
 - `<aside>`: información adicional;
 - `<article>`: artículo independiente.
- Estas etiquetas pueden anidarse unas con otras. Una sección puede tener así su propio encabezado.
- Estas etiquetas no están incluidas en el diseño de la página. Sólo le dicen al ordenador qué significa el texto que contienen. También podrías poner el encabezado en la parte baja de la página si quisieras.
- El código JavaScript `HTML5shiv` se usa para asegurar que estas etiquetas son reconocidas por versiones de Internet Explorer anteriores a IE9.

El modelo de cajas

Una página web puede ser vista como una sucesión y un conjunto de cajas llamadas «bloques». La mayoría de los elementos vistos en los capítulos anteriores son

bloques: `<header>`, `<article>`, `<nav>`... Pero ya conocemos otros bloques: párrafos `<p>`, títulos `<h1>`, etc.

En este capítulo, vamos a aprender cómo interactuar con estos bloques como si fuera cajas reales. Vamos a darles dimensiones, diseñarlos pero ajustando sus márgenes, y también aprenderemos cómo gestionar su contenido... ¡para evitar que el texto salga de estos bloques!

Estos son los conceptos básicos que necesitamos para diseñar nuestra página web... ¡Así que presta atención!

Bloquear y colocar entre líneas las etiquetas de tipo

En HTML, la mayoría de las etiquetas pueden colocarse en una o dos categorías:

- etiquetas **inline**: este es el caso, por ejemplo, de los enlaces `<a></a>`.
- etiquetas de **bloque**: este es el caso, por ejemplo, de los párrafos `<p></p>`.

En realidad hay varias categorías muy específicas, como las celdas de las tablas (`table-cell`) o viñetas (`list-item`). No nos vamos a preocupar por eso de momento, ya que estas etiquetas son una minoría.

¿Pero cómo indicas una etiqueta inline de una etiqueta de bloque?

En realidad es bastante fácil:

- **block**: una etiqueta de bloque en tu página web automáticamente crea un salto de línea antes y después. Sólo tienes que imaginar un bloque. Tu página consistirá, de hecho, en una serie de bloques, uno después del otro. Pero, además, ¡verás que es posible poder un bloque dentro de otro, lo que aumentará significativamente nuestras habilidades de crear el diseño de nuestra página web!
- **inline**: una etiqueta inline está necesariamente situada en una etiqueta de bloque. Una etiqueta inline no crea una vuelta de línea, así que el texto escrito dentro sigue desde el texto anterior en la misma línea (por eso se conoce como etiqueta «inline»).

Para poder visualizar el concepto correctamente, he creado un pequeño diagrama en la siguiente figura.

```
<h1>Title (block)</h1>
```

```
<p>Paragraph (block) text  
text text text text text text  
text <a>Inline link</a> text  
text text text text text  
text</p>
```

```
<p>One more paragraph text  
text text text</p>
```

Diferencia entre una etiqueta inline y otra

de bloque

- Las etiquetas de bloque se muestran con fondo azul.
- Las etiquetas inline se muestran con fondo amarillo.

Como puedes ver, los bloques están debajo uno de otro. Puedes ser anidados unos con otros (recuerda nuestra `<section>` de bloques contiene bloques `<aside>`, por ejemplo).

La etiqueta inline `<a></a>` está situada en una etiqueta de bloque y el texto está insertado en la misma línea.

Unos pocos ejemplos

Para ayudarte a entender qué etiquetas son inline y cuáles son de bloque, aquí hay una tabla con algunas etiquetas habituales.

Etiquetas de bloque	Etiquetas de inline
<code><p></code>	<code></code>
<code><footer></code>	<code></code>
<code><h1></code>	<code><mark></code>
<code><h2></code>	<code><a></code>
<code><article></code>	<code></code>
...	...

Esta table no es toda inclusiva, al contrario. Si quieres una lista completa de todas las etiquetas que existen y si ellas son inline o de bloque, mira el [anexo que te da una lista de todas las etiquetas HTML](#).

Etiquetas para objetivos generales

Ya las conoces porque te hablé de ellas hace varios capítulos. Estas son etiquetas que no tienen un significado en particular (a diferencia de `<p>` que significa «párrafo», `<strong>` que significa «importante», etc.).

La principal ventaja de estas etiquetas es que una `class` (o un `id`) puede aplicarse al CSS cuando no hay otra etiqueta adecuada.

Hay dos etiquetas genéricas y, coincidentemente, la única diferencia entre las dos es que una es inline y la otra en bloque:

- `<span></span>` (**inline**) ;
- `<div></div>` (**bloque**) ;

Adherir a la semántica

Aunque el objetivo general de las etiquetas es práctico, en algunos casos, ten cuidado de no exagerarlas. Ten en cuenta: muchos desarrolladores web usan muchas etiquetas `<div>` y `<span>` y se olvidan que hay otras etiquetas más adecuadas.

Aquí tienes un par de ejemplos:

- **Ejemplo de un apso innecesario:** `<span class=»importante»>`. Nunca debería ver esto en uno de tus códigos ya que la etiqueta `<strong>` está ahí para indicar importancia.
- **Ejemplo de un div innecesario:** `<div class=»título»>`. Es completamente absurdo, ya que hay etiquetas especialmente diseñadas para títulos (`<h1>`, `<h2>`, etc.).

Sí, dirás que al final el efecto visual es el mismo. Estoy de acuerdo. Pero las etiquetas genéricas no añaden significado a la página y no son comprendidas por el ordenador. Siempre usa otras etiquetas, más adecuadas, cuando sea posible. ¡El propio Google recomienda esto para ayudarte a mejorar la posición de tus páginas en los resultados de búsqueda!

Dimensiones

Aquí vamos a trabajar solamente con etiquetas de tipo bloque.

Para empezar, miremos el tamaño de los bloques. Al contrario que una etiqueta inline, un bloque tiene dimensiones específicas. Tiene anchura y altura. Así que tenemos dos propiedades CSS:

- `width`: es la anchura del bloque. Para especificarse en píxeles (px) o como porcentaje (%).

- `height`: es la altura del bloque. Que, de nuevo, se especifica en píxeles (px) o en porcentaje (%).

Para ser exactos, `width` y `height` representan la anchura y la altura del contenido de bloques. Si el bloque tiene márgenes (aprenderemos más sobre esto más adelante), se añadirán a la anchura y a la altura.

Por defecto, un bloque coge el 100 % de la anchura disponible. Esto se puede comprobar aplicando bordes o un color de fondo a nuestros bloques (figura siguiente).

Título

Párrafo

Los bloques cogen toda la anchura disponible

Ahora añadamos un poco de CSS para cambiar la anchura de los párrafos. El código CSS siguiente dice: «Quiero que todos mis párrafos tengan una anchura del 50 %»

```
p
{
    width: 50 %;
}
```

El resultado se muestra en la siguiente figura.

Título

Párrafo



Un párrafo con una anchura del 50 %

Los porcentajes serán útiles para crear un diseño que automáticamente se adapte a la resolución de pantalla del visitante.

Sin embargo, podrían necesitar crear bloques con un tamaño específico en píxeles.

```
p
{
    width: 250px;
}
```

Mínimo y máximo

Puedes especificar las dimensiones máximas y mínimas de un bloque. Esto es muy útil ya que nos permite definir dimensiones «límite» para permitirle a nuestra página adaptarse a las diferentes resoluciones de las pantallas de los visitantes.

- `min-width`: anchura mínima;
- `min-height`: altura mínima;
- `max-width`: anchura máxima;
- `max-height`: altura máxima.

Por ejemplo, puedes especificar que los párrafos deben ocupar en 50 % del ancho y tener al menos 400 píxeles de ancho en todos los casos.

```
p
{
    width: 50%;
    min-width: 400px;
}
```

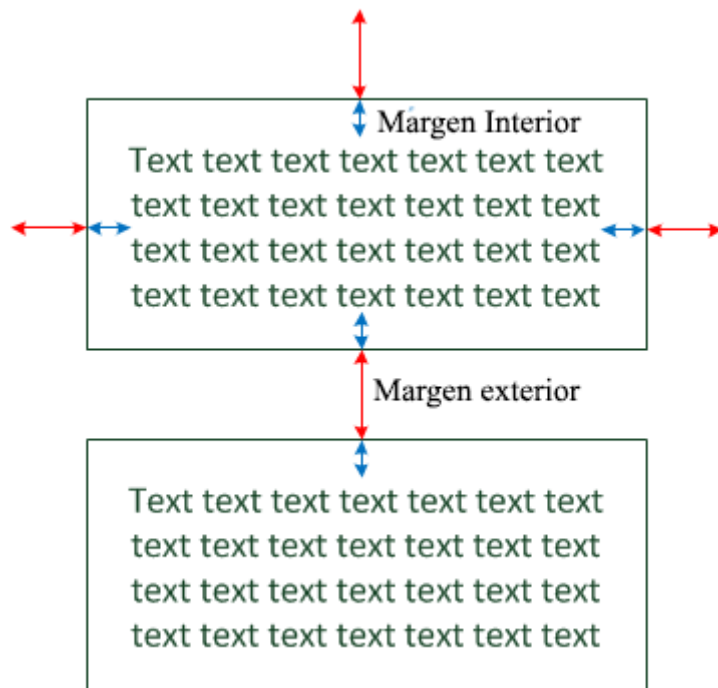
Ve el resultado cambiando la anchura de la ventana de tu navegador. Verás que, si es muy pequeño, el párrafo se ve forzado a tener una anchura de al menos 400 píxeles.

Márgenes

Deberías saber que todos los bloques tienen márgenes. Hay *dos tipos de márgenes*:

- márgenes interiores
- márgenes exteriores

Echa un vistazo en el diagrama en la figura siguiente.



He colocado un borde alrededor de este bloque para ver mejor dónde acaba.

En CSS, puedes cambiar el tamaño de los márgenes con las dos siguientes propiedades:

Las etiquetas inline también tienen márgenes. Así que puedes probar estos ajustes en este tipo de etiqueta.

```
p
{
    width: 350px;
    border: 1px solid black;
    text-align: justify;
```



Márgenes por defecto para los párrafos

Como puedes ver, por defecto no hay margen interior (`padding`). Sin embargo, hay un margen exterior (`margin`). Es debido a este margen por lo que dos párrafos no están pegados y parecen estar separados por una «línea de separación».

Los márgenes por defecto no son los mismos para todas las etiquetas del bloque. Intenta aplicar este CSS a las etiquetas `<div>` que contienen texto, por ejemplo: ¡verás que en este caso por defecto no hay ningún margen interior ni exterior!

Supongamos que quiero añadir un margen interno de 12 px a los párrafos (figura siguiente):

```
p
{
  width: 350px;
```

```

border: 1px solid black;

text-align: justify;

padding: 12px; /* Margen interior de 12px */
}

```



Un margen interior añadido a los párrafos

Ahora quiero que mis párrafos estén más separados. Añado la propiedad `margin` para especificar un margen de 50 px entre dos párrafos (siguiente figura):

```

p
{

width: 350px;

border: 1px solid black;
}

```

```

text-align: justify;

padding: 12px;

margin: 50px; /* Margen Exterior de 50px */
}

```



¿Pero? ¡Un margen también se añade a la izquierda!

En efecto, `margin` (como `padding`) se aplica a los cuatro lados del bloque. Si quieres especificar un margen superior, inferior, izquierdo y derecho diferente, tienes que usar unas propiedades más específicas... El principio es el mismo que para la propiedad `border`, como verás.

arriba, derecha, izquierda, abajo...

Tienes que recordar los siguientes términos:

- `top;`
- `bottom;`
- `left;`

- `right;`

Para que puedas mantener todas las propiedades en tu cabeza.

Igual, te haré una lista de propiedades para `margin` y `padding` para que entiendas el principio.

Aquí está la lista para `margen`:

- `margin-top`: margen exterior en la parte superior;
- `margin-bottom`: margen exterior en la parte inferior;
- `margin-left`: margen exterior en la izquierda;
- `margin-right`: margen exterior en la derecha.

Y la lista para `relleno`:

- `padding-top`: margen interior en la parte superior;
- `padding-bottom`: margen interior en la parte inferior;
- `padding-left`: margen interior en la izquierda;
- `padding-right`: margen interior en la derecha.

Hay otras maneras de especificar el margen con las propiedades `margin` y `padding`.

Por ejemplo:

`margin: 2px 0 3px 1px;` significa «margen superior de 2 px, 0 px a la derecha (el px es opcional en este caso), 3 px abajo, 1 px izquierda».

Otra notación acortada: `margin: 2px 1px;` significa «margen de 2 px arriba y abajo, margen de 1 px izquierda y derecha».

[Centrar bloques](#)

Los bloques pueden centrarse perfectamente. Y un diseño centrado es cómodo cuando no tienes la resolución del visitante.

Hay que seguir las siguientes reglas para centrar:

- dale una anchura al bloque (con la propiedad `width`);
- especifica que quieres unos márgenes exteriores automáticos así: `margin: auto;`.

Intentemos esta técnica en nuestros pequeños párrafos (líneas 3 y 4):

```
p
{
    width: 350px; /* Especificamos una anchura (obligatorio) */
    margin: auto; /* Podemos pedir que el bloque se centre con el auto */
}
```

```
border: 1px solid black;

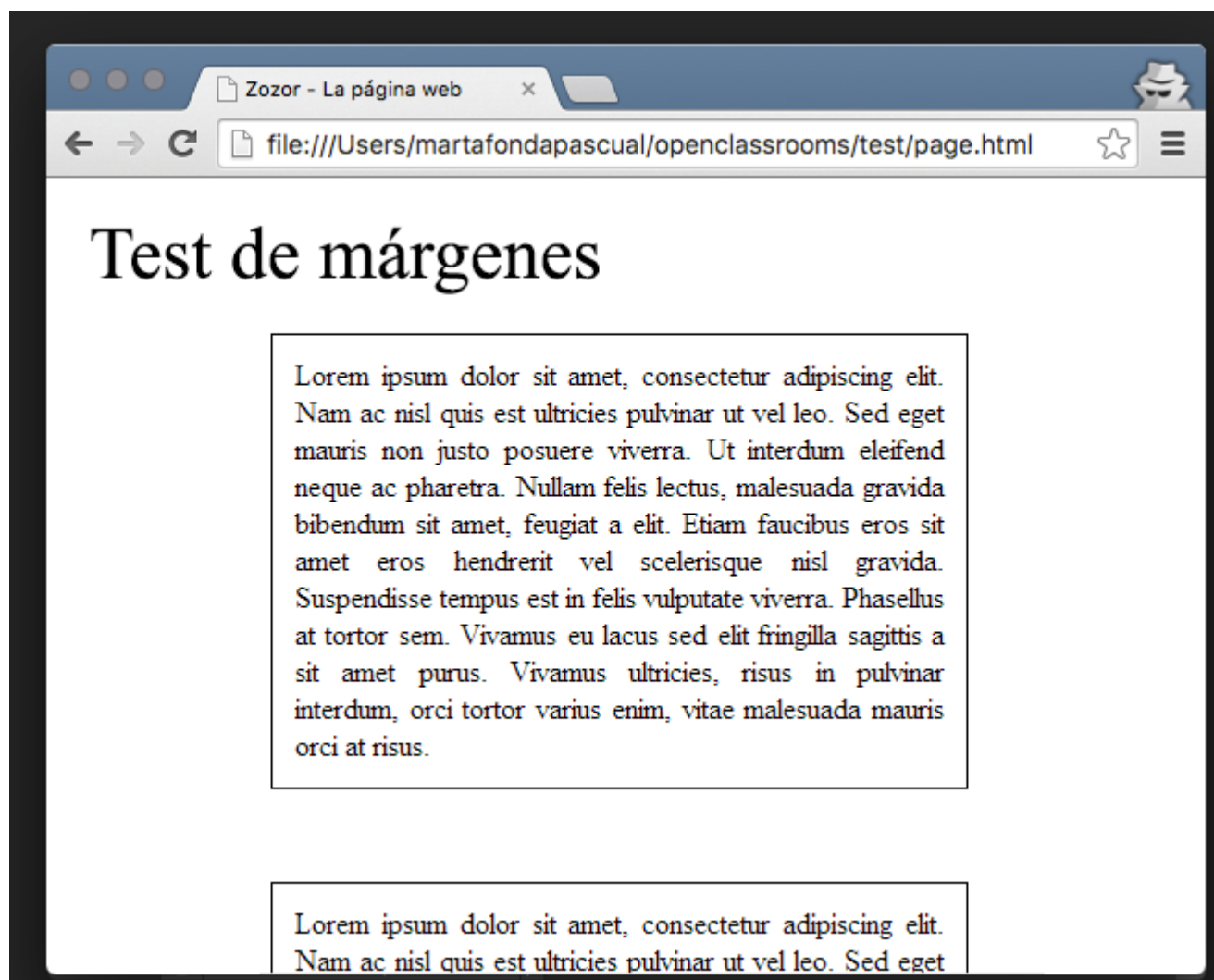
text-align: justify;

padding: 12px;

margin-bottom: 20px;

}
```

Y en la siguiente figura, el resultado.



¡Así el navegador centra automáticamente nuestros párrafos!

Un bloque no puede centrarse verticalmente con esta técnica. Solo el centrado horizontal está permitido.

Cuando las cosas se desbordan...

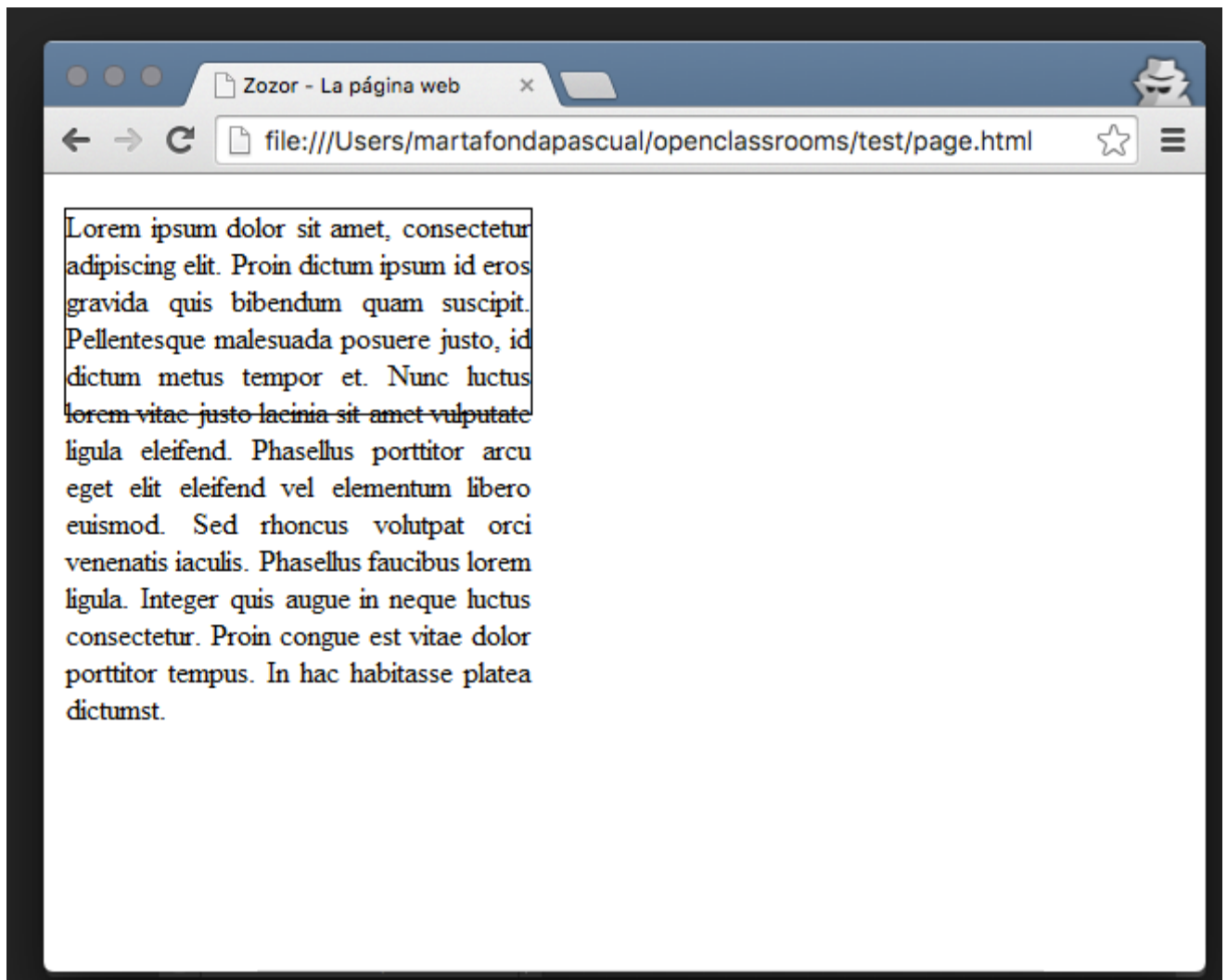
Cuando empezamos a definir las dimensiones exactas para nuestros bloques, como hemos hecho, podrían a veces parecer muy pequeños para el texto que contienen.

Las propiedades CSS que vamos a ver aquí han sido creadas precisamente para examinar el overflow... y decidir qué hacer si esto ocurre.

overflow: cortar un bloque

Supongamos que tienes un párrafo largo y quieres (por cualquier razón) que sea 250 px de ancho y 110 px de alto. Añadámosle un borde y rellenámoslo con texto... hasta el borde (figura siguiente).

```
p
{
    width: 250px;
    height: 110px;
    text-align: justify;
    border: 1px solid black;
}
```



El texto va más allá del bloque del párrafo

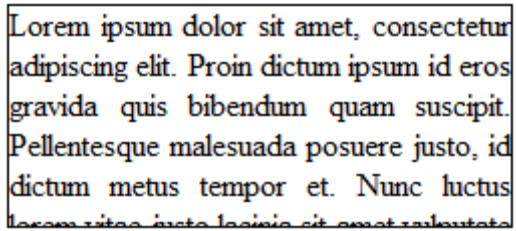
¡Oh, no! ¡El texto va más allá de los límites del párrafo!

¡En efecto! Preguntaste las dimensiones exactas, ¡las conseguiste! Pero... el texto no encaja en ese bloque tan pequeño.

Si quieres que el texto quede dentro de los límites del párrafo, tendrás que usar la propiedad `overflow`. Estos son los valores que puede llevar:

- `visible` (valor por defecto): si el texto excede los límites del tamaño, permanece visible y deliberadamente va más allá de los límites del bloque.
- `hidden`: si el texto excede los límites, simplemente se cortará. No seremos capaces de ver todo el texto.
- `scroll`: de nuevo, el texto será cortado si excede los límites. Excepto esta vez, el navegador creará barras de desplazamiento para que podamos leer el texto completo. Es como un marco dentro de la página.
- `auto`: es el modo «autopiloto». Básicamente, es el navegador el que decide si mostrar o no las barras de desplazamiento (sólo lo hará si es necesario). Este es el valor que recomiendo que uses más a menudo.

Mediante `overflow: hidden;` el texto queda cortado (no se puede ver qué hay debajo), como en la siguiente ilustración.

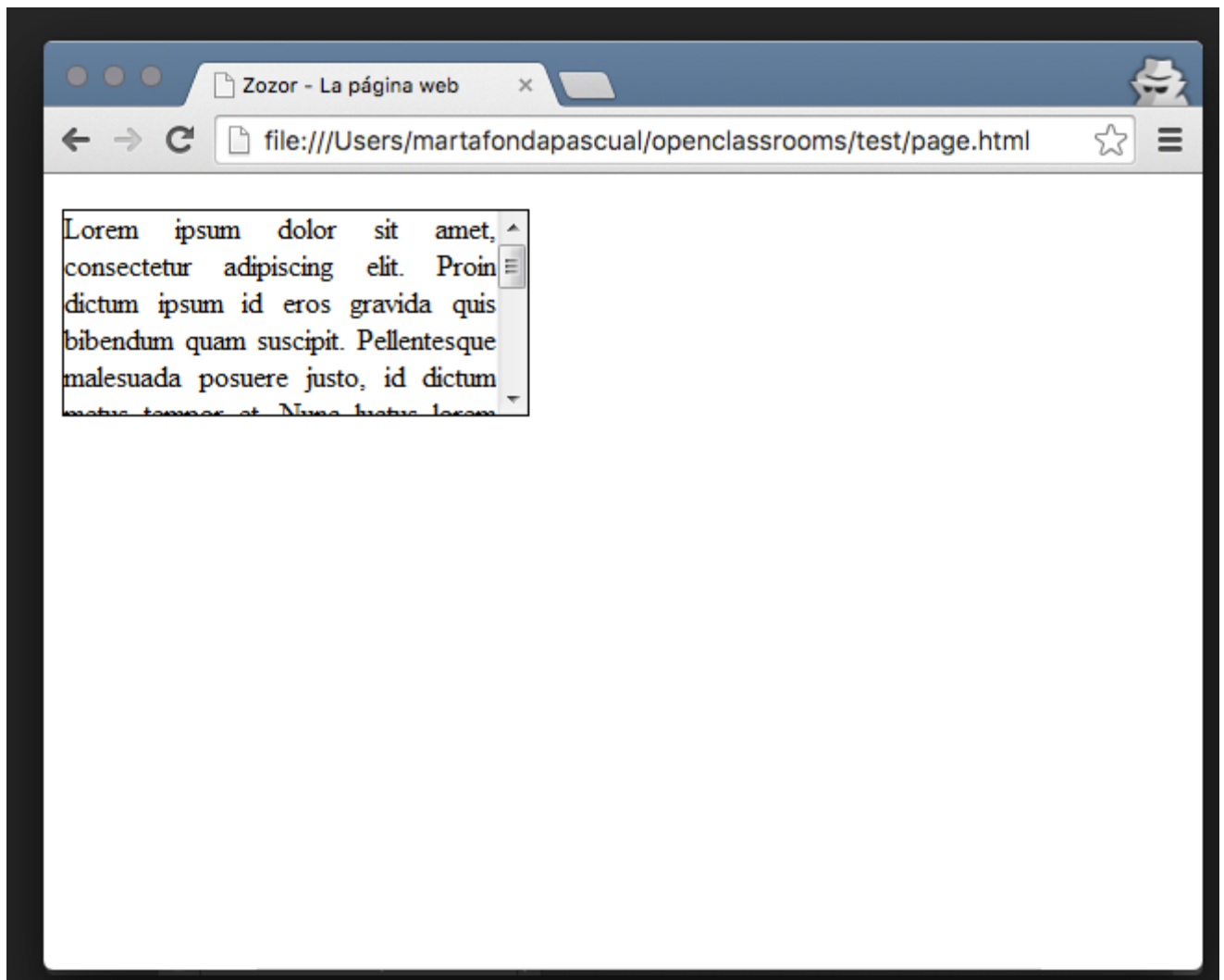


Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Proin dictum ipsum id eros
gravida quis bibendum quam suscipit.
Pellentesque malesuada posuere justo, id
dictum metus tempor et. Nunc luctus
lorem ipsum justo laoreet sit amet tristique.

El texto queda cortado en los límites del párrafo

Probemos ahora `overflow: auto;` con el código CSS mostrado a continuación (el resultado se puede comprobar en la siguiente imagen):

```
p
{
    width: 250px;
    height: 110px;
    text-align: justify;
    border: 1px solid black;
    overflow: auto;
}
```



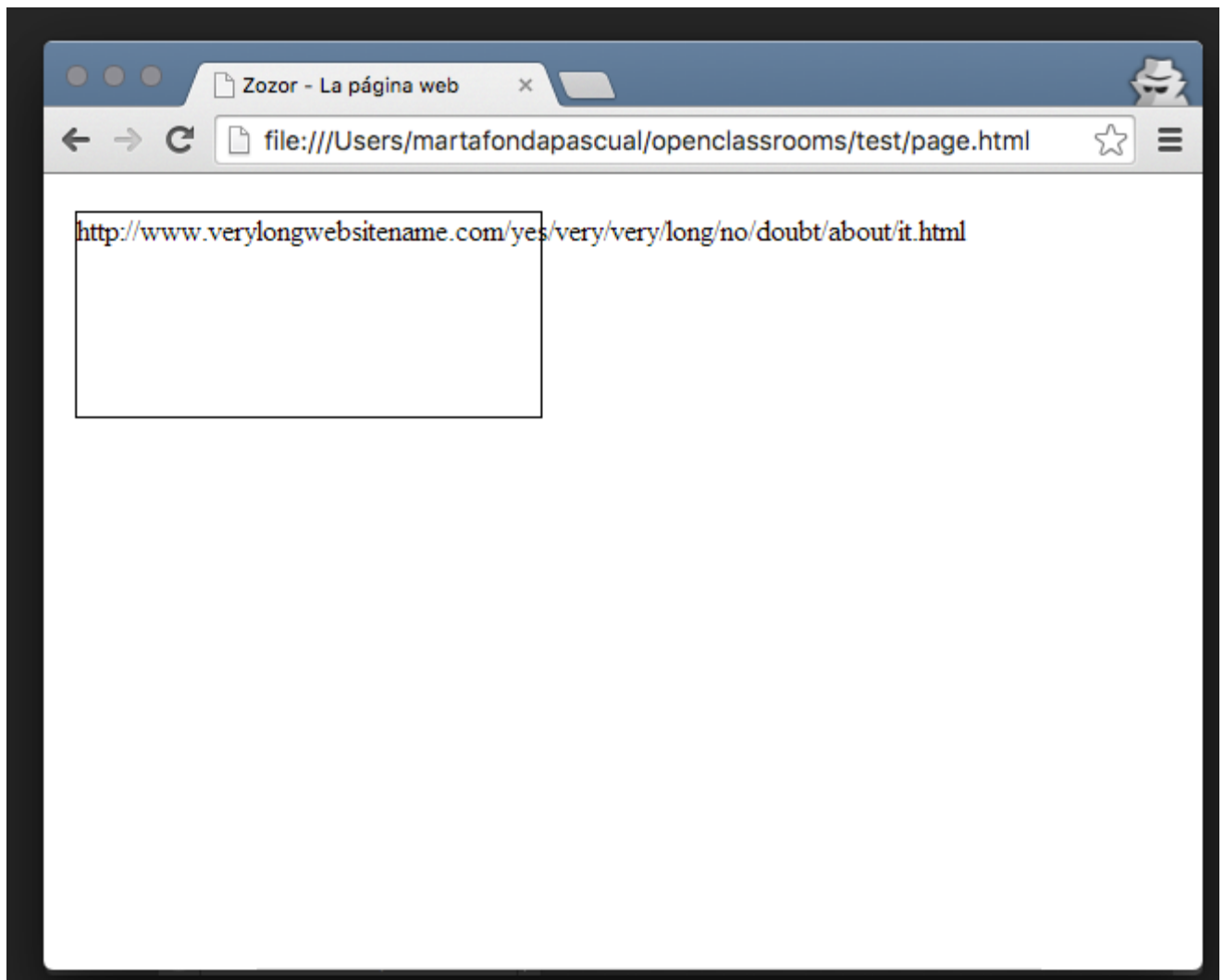
¡Eureka! Las barras de desplazamiento nos permiten ahora ver el contenido que no era visible.

Existe una vieja etiqueta HTML, `<iframe>`, que nos dará prácticamente el mismo resultado. Sin embargo, ya no es recomendable seguir usando esta etiqueta. Te permite cargar los contenidos completos de otra página HTML en tu página.

word-wrap: cortar textos demasiado grandes

Si necesitas colocar una palabra muy larga en un bloque y no cabe por ser demasiado ancha, te encantará `word-wrap`. Esta propiedad se utiliza para forzar la partición con guión de palabras muy largas (normalmente, direcciones bastante largas).

La siguiente ilustración muestra el posible resultado de introducir una URL bastante larga en un bloque.

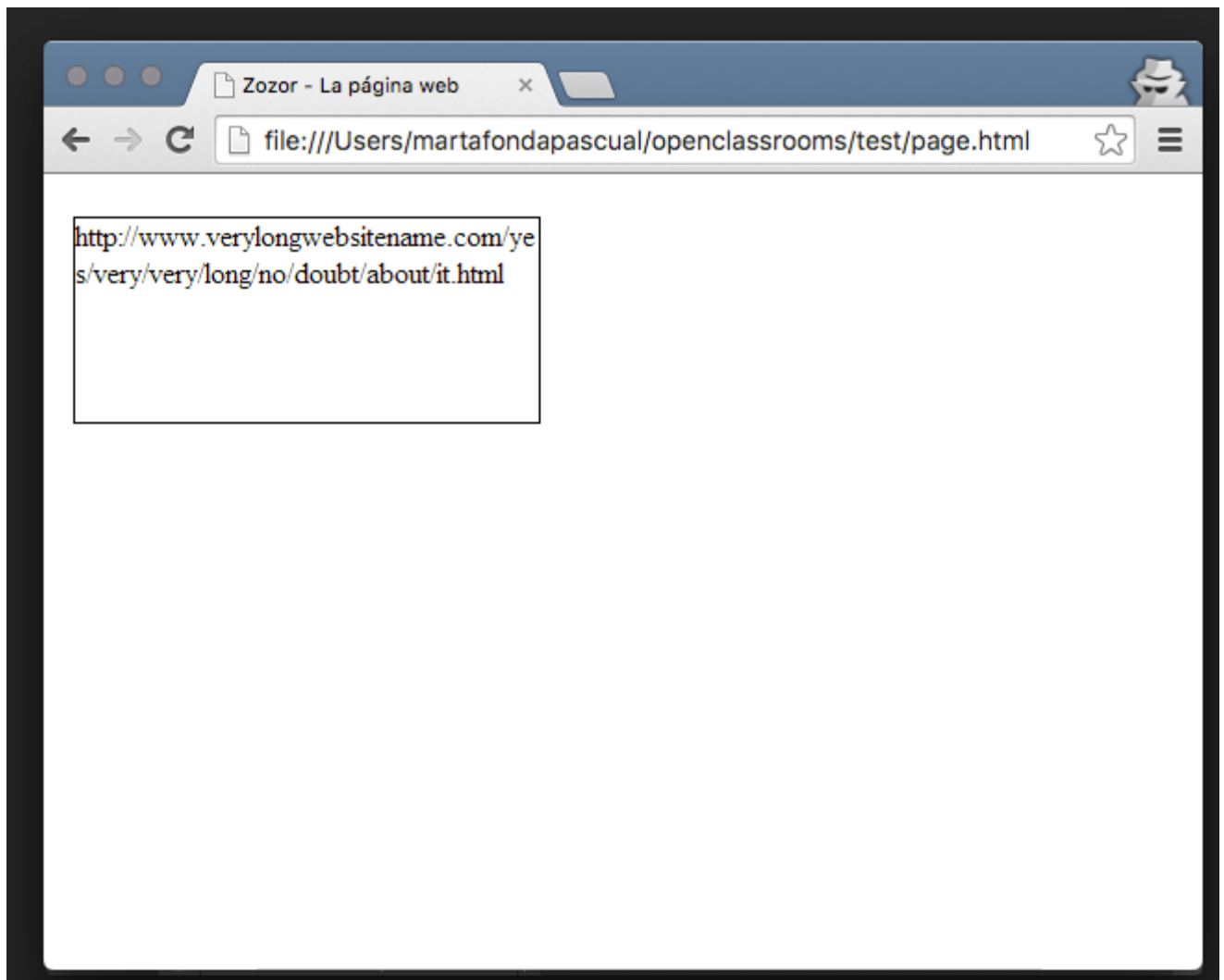


El texto excede el ancho disponible

El ordenador no puede «cortar» la dirección ya que no hay espacio o guión, y no sabe cómo dividirla con un guión.

Con el siguiente código, se forzará una división con guiones cuando el texto sea susceptible de exceder los límites (siguiente ilustración).

```
p
{
  word-wrap: break-word;
}
```



El texto se corta para que no exceda el espacio

Recomiendo usar esta función cuando un bloque pueda contener texto introducido por usuarios (por ejemplo, en los foros de tu futura página web). Sin este truco, el diseño de una página podría romperse fácilmente (al escribir series largas de «aaaaaaaaaaaa», por ejemplo).

En resumen

- Hay dos tipos principales de etiquetas en HTML:
 - Etiquetas de bloque (<p>, <h1>...): estas etiquetas crean un salto de línea por defecto y ocupan todo el ancho disponible.. Se suceden de arriba a abajo.
 - Etiquetas en línea (<a>, ...): estas etiquetas definen el texto en medio de una línea. Se suceden de izquierda a derecha.
- Puedes cambiar el tamaño de una etiqueta de bloque mediante las propiedades CSS `width` y `height`.
- Puedes establecer el mínimo y máximo permitido para el ancho y el alto: `min-width`, `max-width`, `min-height`, `max-height`.
- Los elementos de página tiene márgenes internos (`padding`) y márgenes externos (`margin`).

- Si hay demasiado texto dentro de un bloque de dimensiones fijas, el texto podría exceder los límites. En este caso, una buena opción sería añadir barras de desplazamiento mediante la propiedad `overflow` o utilizar la división con guiones mediante `word-wrap`.

Posicionamiento en CSS

Y aquí llega el ansiado momento: vamos a aprender cómo cambiar la posición de los elementos en nuestra página. Necesitaremos la teoría que veremos aquí para el próximo capítulo, donde diseñaremos nuestra primera página web paso a paso.

Veremos que hay varias técnicas para definir el diseño de página de un sitio web. Cada uno tiene sus pros y sus contras, así que dependerá de ti elegir el que creas más conveniente en función del caso.

Posicionamiento flotante

La técnica aquí mostrada tiene un fin informativo. Aunque actualmente se usa en la mayoría de páginas, tiene ciertas carencias. Enseñaré una técnica más efectiva, llamada posicionamiento `inline-block` (bloque-en línea), algo más adelante; recomiendo utilizarla siempre que sea posible.

¿Recuerdas la propiedad `float`? La hemos utilizado para situar una imagen alrededor del texto (siguiente ilustración).



Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Donec
vitae lorem imperdiet lacus molestie
molestie. Cum sociis natoque penatibus
et magnis dis parturient montes, nascetur ridiculus
mus. Donec eu purus. Phasellus metus lorem,
blandit et, posuere quis, tincidunt vitae, ante.
Vivamus consequat mauris a diam. Vivamus nibh
erat, hendrerit nec, aliquet ut, hendrerit quis, nunc.
Vestibulum et turpis et elit tempor euismod.

La imagen se sitúa alrededor del texto mediante la propiedad `float`.

Resulta que la mayoría de páginas web utilizan esta propiedad para... ¡el diseño de la página! De hecho, si quieres situar tu menú en la izquierda y el contenido de la página en la derecha *suele* ser una buena opción. Digo *suele* porque esta propiedad no se diseñó en un principio para aplicarse al diseño de una página y, como vamos a ver, tiene algunas pequeñas desventajas.

Vamos a usar el código HTML estructurado que escribimos unos capítulos atrás.

```
<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8" />

    <link rel="stylesheet" href="estilo.css" />

    <title>Zozor: la página web</title>

  </head>

  <body>

    <header>

      <h1>Zozor</h1>

      <h2>Diarios de viaje</h2>

    </header>

    <nav>

      <ul>

        <li><a href="#">Inicio</a></li>

        <li><a href="#">Blog</a></li>

        <li><a href="#">Continuar</a></li>

      </ul>

    </nav>

    <section>
```

```
<aside>

  <h1>Sobre el autor</h1>

  <p>¡Soy yo, Zozor! Nací el 23 de noviembre de 2005.</p>

</aside>

<article>

  <h1>Soy un gran viajero</h1>

  <p>Bla, bla, bla, bla (texto del artículo)</p>

</article>

</section>


<footer>

  <p>Copyright Zozor - Todos los derechos reservados

  <a href="#">¡Escríbeme!</a></p>

</footer>

</body>

</html>
```

Recuerda que sin CSS el diseño de la página quedará como en la siguiente ilustración.



Una página HTML sin CSS

Vamos a intentar situar el menú en la izquierda y el resto del texto en la derecha. Para ello, pondremos el menú en la izquierda con *float* y dejaremos que el resto del texto se coloque por sí solo a la derecha.

Queremos que el menú tenga 150 píxeles de ancho. También vamos a añadir un borde negro al menú y un borde azul alrededor del cuerpo (utilizando la etiqueta `<section>`) para resaltarlos:

```
nav
{
    float: left;

    width: 150px;

    border: 1px solid black;
```

```

}

section
{
    border: 1px solid blue;
}

```

La siguiente ilustración muestra el resultado. Aún nos queda trabajo.



El menú está posicionado correctamente, pero está pegado al texto

Hay dos fallos (aparte de que aún no es agradable visualmente):

- El cuerpo de la página toca el extremo del menú, así que necesita algo de margen.
- Y lo que es aún más molesto: ¡el resto del texto continúa debajo del menú!

Aunque queremos situar el pie de página («Copyright Zozor») en la parte inferior, bajo el menú, nos gustaría que la página entera fuera un solo bloque a la derecha.

Para solucionar estos dos problemas de una, hemos de añadir un margen exterior a la derecha de `<section>`, que además deberá ser *mayor* que el ancho del menú. Si nuestro menú tiene un ancho de 150 px, daremos, por ejemplo, un margen exterior izquierdo de 170 px (siguiente ilustración) a nuestra sección de página en la línea 10.

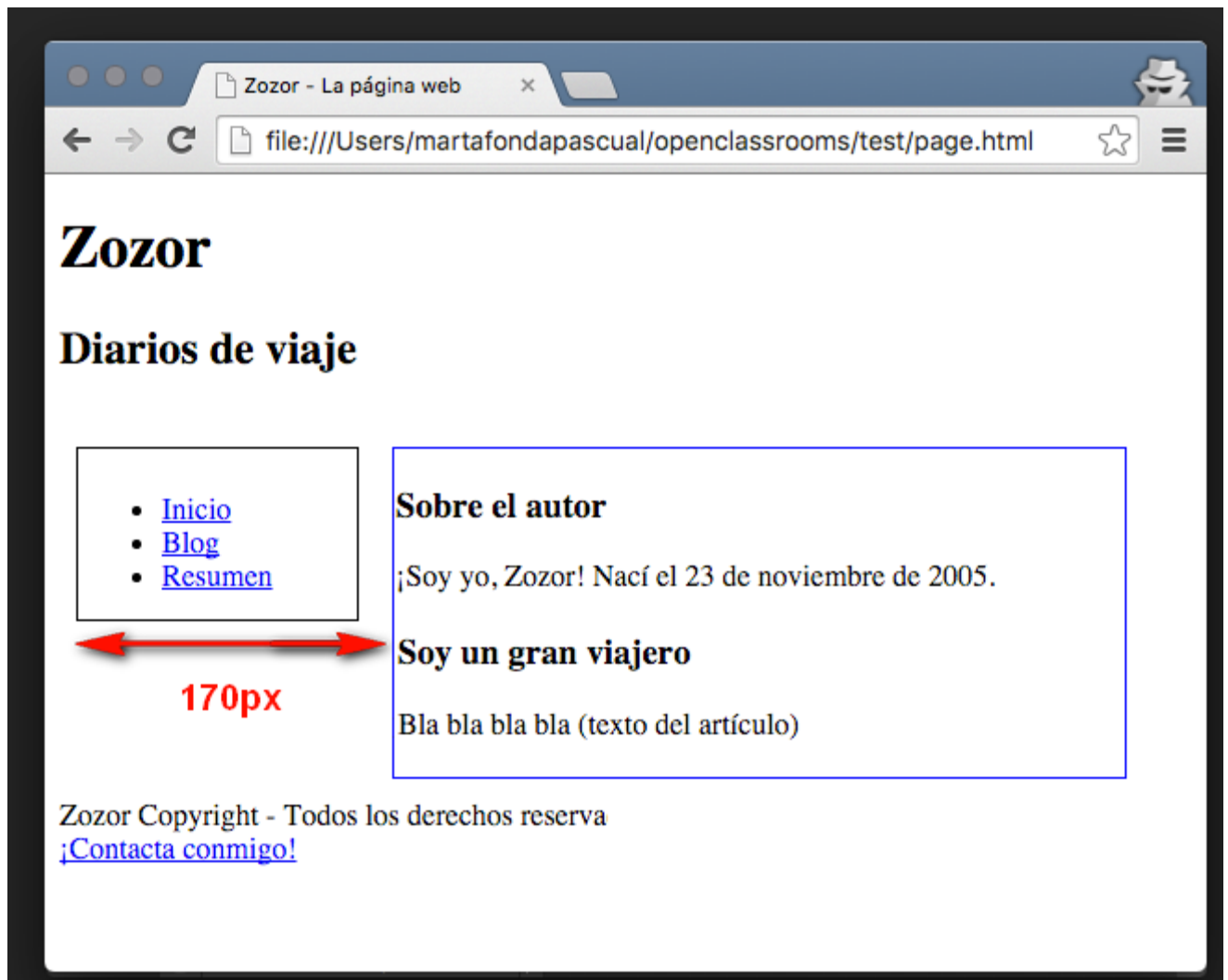
```
nav
{
    float: left;

    width: 150px;

    border: 1px solid black;
}

section
{
    margin-left: 170px;

    border: 1px solid blue;
}
```



El cuerpo de la página está alineado correctamente a la derecha del menú

Y ahí lo tienes, el contenido de la página está ahora correctamente alineado.

De manera alternativa, también puede darse que quieras ubicar un elemento bajo el menú. En este caso, tendrías que utilizar `clear: both;`, como ya hemos visto, para posicionar el resto del texto bajo el elemento que esté a su alrededor.

Cambiar tus elementos con *display*

Ahora pasaré a enseñar cómo modificar las reglas de CSS (¿eh, eh?) ¡Alto ahí!

CSS cuenta con una propiedad muy potente: `display`. Es capaz de *modificar* cualquier elemento de tu página de un tipo a otro. Con esta propiedad mágica puedo, por ejemplo, hacer que mis enlaces (que al principio eran elementos en línea) aparezcan en forma de bloque:

```
a
{
```

```
display: block;
}
```

Así los enlaces se posicionarán uno debajo del otro (como bloques normales) y entonces podrás modificar sus dimensiones.

A continuación, algunos de los principales valores que la propiedad `display` puede soportar en CSS (aunque también hay otros):

Valor	Ejemplos	Descripción
<code>inline</code>	<code><a></code> , <code></code> , <code></code> ...	Elementos en una línea. Se colocan uno al lado del otro.
<code>block</code>	<code><p></code> , <code><div></code> , <code><section></code> ...	Elementos en forma de bloque. Se encuentran uno debajo del otro y se pueden redimensionar.
<code>inline-block</code>	<code><select></code> , <code><input></code>	Elemento posicionados uno al lado del otro (como elementos en línea), pero que se pueden redimensionar (como bloques).
<code>none</code>	<code><head></code>	Elementos no mostrados.

Puedes escoger qué elemento de página ocultar completamente mediante esta propiedad. Por ejemplo, si quisiera ocultar los elementos que pertenecen a la clase «secreta», escribiría:

```
.secreta
{
    display: none;
}
```

Para mostrar estos elementos más adelante, tendrás que usar JavaScript. Algunas páginas web utilizan esta técnica de manera predeterminada para ocultar menús secundarios que solo aparecen al navegar por los menús.

¿Y este extraño tipo de elemento nuevo, `inline-block`? ¿Es una mezcla?

Sí, este tipo de elemento es de hecho una combinación de elemento en línea y de bloque. Es lo mejor de dos mundos en uno: los elementos se muestran uno al lado del otro y se pueden redimensionar.

Hay muy pocas etiquetas que se muestren por defecto así, y guardan relación principalmente con los elemento de forma (como `<input>`), de los que hablaremos

más tarde. Sin embargo, mediante la propiedad `display` podremos transformar otras etiquetas a `inline-block`, lo que nos ayudará a crear nuestro diseño.

Posicionamiento *inline-block*

Los pasos necesarios para la ubicación de elementos flotantes puede, a veces, llegar a ser algo difícil en páginas web complejas. En cuanto haya que dar formato a algo que no sea un menú, tendrás que recurrir a las etiquetas `clear:both`, lo que complicará en un momento el código de la página.

Aunque el posicionamiento flotante es de lejos el más usado hoy en día en Internet, existen otras técnicas, y muy pocos *webmasters* lo saben. Una de estas técnicas sorprendentemente eficientes ha pasado desapercibida por los diseñadores web a pesar de haber estado disponible desde CSS 2.1, o, en otras palabras, ¡desde hace más de diez años! Consiste en la transformación de tus elementos a `inline-block` mediante la propiedad `display`.

Aquí van algunas cosas que recordar sobre los elementos *inline-block*:

- Están colocados uno al lado del otro (¡justo como queremos colocar nuestro menú y el cuerpo de nuestra página!).
- Se les puede asignar dimensiones específicas (¡justo lo que queremos, de nuevo!).

Vamos a transformar los dos elementos que queremos situar uno al lado del otro en *inline-block*: el menú de navegación y la sección central de la página.

```
nav
{
    display: inline-block;

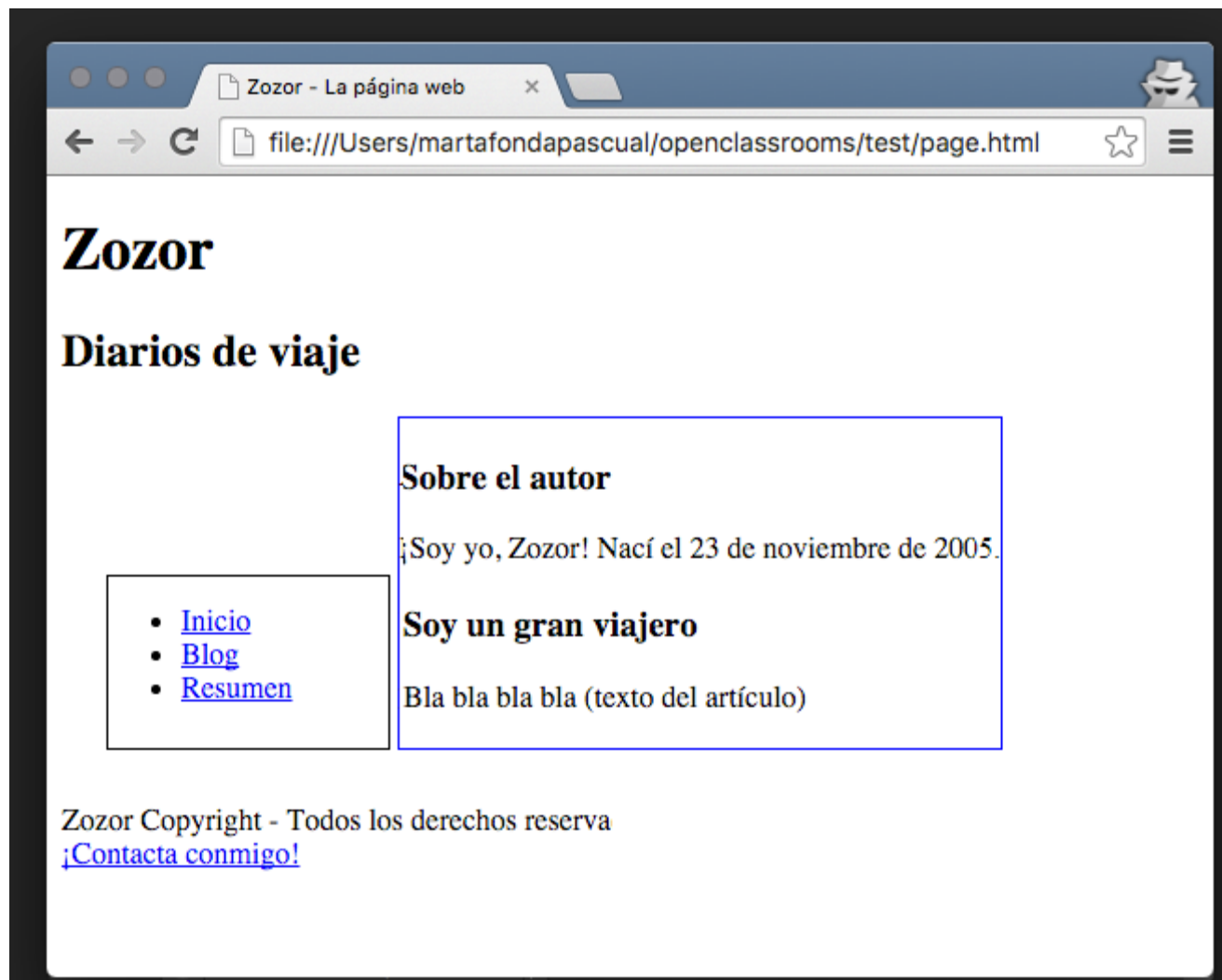
    width: 150px;

    border: 1px solid black;
}

section
{
    display: inline-block;
```

```
border: 1px solid blue;
}
```

El resultado se muestra en la siguiente ilustración.



El menú y el cuerpo están uno al lado del otro, ¡aunque colocados en la parte inferior!

¡Ups!

No es lo que intentábamos. Es normal, en realidad: los elementos *inline-block* están colocados en la misma **línea base**, en la parte inferior.

Por suerte, tras haber convertido los elementos en *inline-block*, podemos usar una nueva propiedad, normalmente reservada para tablas: `vertical-align`. Esta propiedad se usa para cambiar la alineación vertical de elementos. Aquí tenemos alguno de los posibles valores que esta propiedad acepta:

- `baseline`: alinea la base del elemento con la base del elemento padre (por defecto);
- `top`: lo alinea con la parte superior;

- `middle`: lo centra en vertical;
- `bottom`: lo alinea al pie;
- (valor en px o como %): lo alinea a una determinada distancia de la línea base.

A continuación solo hemos de alinear en la parte superior nuestros elementos (líneas 6 y 13), ¡y ya lo tienes!

```
nav
{
    display: inline-block;

    width: 150px;

    border: 1px solid black;

    vertical-align: top;
}

section
{
    display: inline-block;

    border: 1px solid blue;

    vertical-align: top;
}
```



El menú y el cuerpo se alinean en la parte superior uno al lado del otro

Fíjate en que el cuerpo (`<section>`) no ocupa todo el ancho. ¡Claro, si no es un bloque! La sección solamente ocupa el espacio que necesita.
Si no es lo que quieres en tu diseño, cambia el tamaño de la sección con `width`.

¡Y no hay más! Ni hay que preocuparse de los márgenes, ni existe riesgo de que el texto sobresalga del menú... En resumen, ¡que es perfecto!

¿...Cómo? Perdona, me acaba de llegar que hay cierto navegador que da algo de guerra...

`inline-block` y compatibilidad con Internet Explorer

El posicionamiento `inline-block` es totalmente compatible con Internet Explorer a partir de IE8.

En las versiones anteriores, concretamente en IE6 e IE7, *`inline-block`* funciona, pero solo en aquellos elementos de tipo en línea desde un principio.

Por suerte, podemos arreglar este problema con una solución alternativa: si el navegador es IE6 o IE7, puedes hacer que el elemento sea en línea y cambiar su comportamiento mediante el *diseño* (asignándolo con la propiedad `zoom: 1;`). Así, el tamaño del elemento se podrá modificar.

El **diseño** es uno de los modos de visualización algo especiales de Internet Explorer, y puede ser causa de muchos fallos. Como el asunto se pone bastante técnico, será mejor para todos dejarlo estar; si estás interesado en la materia, te sugiero que busques más información sobre la propiedad *HasLayout* de Internet Explorer.

Para empezar, tendrás que crear una hoja de estilo CSS especial para Internet Explorer 6 y 7. Para ello, usaremos el **comentario condicional** que ya hemos visto y que solo leen IE6 e IE7.

```
<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8" />

    <link rel="stylesheet" href="style.css" />

    <!--[if lte IE 7]>

    <link rel="stylesheet" href="style_ie.css" />

    <![endif]-->

    <title>Zozor: la página web</title>

  </head>
```

Si el navegador es IE6 o IE7, se cargará una segunda Hoja de Estilo en Cascada (CSS), `style_ie.css`, que se añadirá a la primera. En ella estarán todas nuestras «soluciones alternativas» para arreglar los problemas relacionados con las versiones antiguas de Internet Explorer y asegurar que nuestra página web siempre se visualiza correctamente, incluso con programas del pleistoceno ;-).

Simplemente añade en `style_ie.css` el siguiente código:

```
nav, section

{
```

```
display: inline;

zoom: 1;

}
```

Y ahí lo tienes, no hay más. En nuestra hoja CSS normal (`style.css`), puedes seguir usando `inline-block` como antes. Los que utilicen IE6 o IE7 *también* podrán leer este código ligeramente distinto, lo que les permitirá conseguir el mismo resultado visual.

Posicionamiento absoluto, fijo y relativo

Existen algunas otras técnicas bastante especiales que se emplean para posicionar elementos con precisión en la página:

- **Posicionamiento absoluto:** nos permite posicionar un elemento en cualquier parte de la página (superior izquierda, inferior derecha, centro, etc.).
- **Posicionamiento fijo:** es lo mismo que el posicionamiento absoluto, pero aquí el elemento permanece visible incluso si te desplazas hacia la parte inferior de la página. Es parecido, si hacemos memoria, al principio `background-attachment: fixed;`.
- **Posicionamiento relativo:** permite al elemento estar fuera de su posición normal.

Del mismo modo que los objetos flotantes, los posicionamientos absolutos, fijos y relativos también funcionan en etiquetas en línea.

Sin embargo, verás que es más frecuente usarlos en etiquetas de bloque que en etiquetas en línea.

En primer lugar tendrás que elegir una de las tres opciones de posicionamiento. Para ello, se utiliza la propiedad CSS `position` y se le da uno de estos valores:

- `absolute`: posicionamiento absoluto;
- `fixed`: posicionamiento fijo;
- `relative`: posicionamiento relativo.

Pasaremos a examinar cada una de estas opciones de posicionamiento.

Posicionamiento absoluto

El posicionamiento absoluto permite situar un elemento en cualquier lugar de la página. Para situar un elemento de manera absoluta has de introducir:

```
element
```

```
{  
  
    position: absolute;  
  
}
```

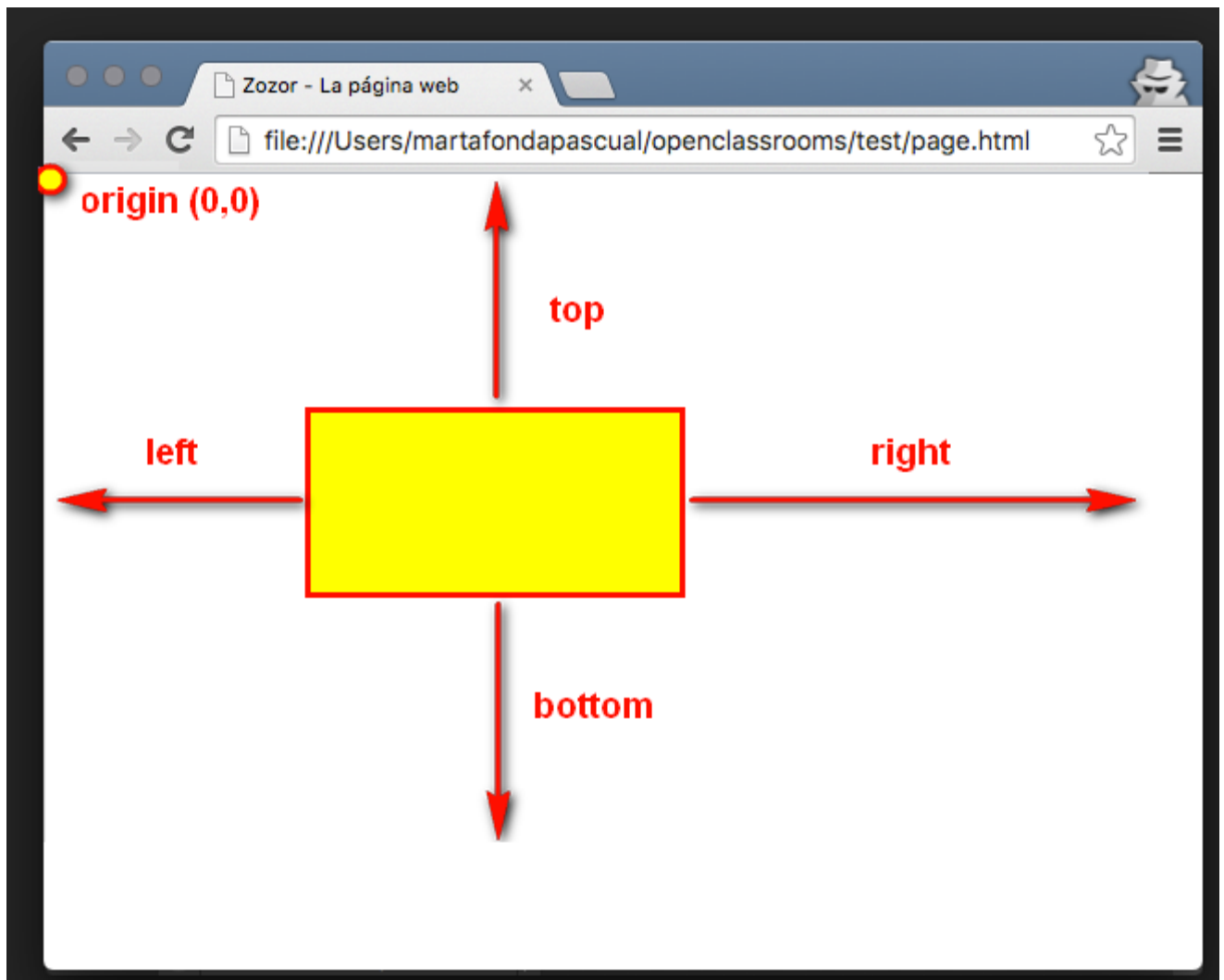
¡Pero no basta con esto! Aunque hemos especificado que el posicionamiento ha de ser absoluto, hemos de indicar *en qué parte de la página queremos que se sitúe el bloque*.

Para ello, vamos a utilizar cuatro propiedades CSS:

- `left`: posición relativa en la izquierda de la página;
- `right`: posición relativa en la derecha de la página;
- `top`: posición relativa en la parte superior de la página;
- `bottom`: posición relativa en el pie de la página;

Puedes darles un valor en píxeles, como 14px, o un porcentaje, como 50%.

Si todavía no ha quedado claro, la siguiente ilustración debería ayudarte a entenderlo.



Posicionamiento de elementos absolutos

Mediante esta propiedad deberías ser capaz de situar tu bloque correctamente.

Así pues, tendrás que usar la propiedad `position` correctamente y una de las cuatro propiedades anteriores como mínimo (`top`, `left`, `right` o `bottom`). Si por ejemplo introduces:

```
element
{
    position: absolute;
    right: 0px;
    bottom: 0px;
}
```

...significará que el bloque se situará a la derecha, en la parte inferior derecha de la página (0 píxeles tomando como referencia la derecha de la página, 0 tomando la parte inferior de la página),

Si intentamos poner nuestro bloque `<nav>` en la parte inferior derecha de la página, obtendremos el resultado mostrado en la siguiente ilustración.



El menú se sitúa en la esquina inferior derecha de la pantalla

Puedes, por supuesto, añadir un margen interior (*padding*) al menú para separarlo ligeramente del borde.

Los elementos posicionados de manera absoluta están sobre el resto de elementos de la página. Además, si sitúas dos elementos de manera absoluta en el mismo lugar es posible que se solapen. En estos casos, utiliza la propiedad `z-index` para especificar qué elemento debería aparecer encima de los otros elementos.

```
element
```

```
{
```

```
position: absolute;

right: 0px;

bottom: 0px;

z-index: 1;

}

element2

{

position: absolute;

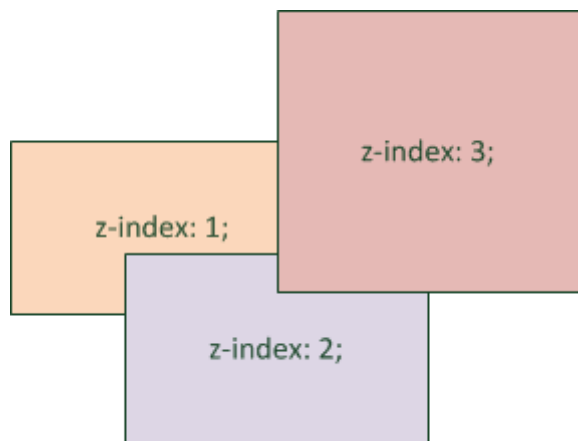
right: 30px;

bottom: 30px;

z-index: 2;

}
```

El elemento con el valor `z-index` más alto se situará por encima de los otros, tal y como se muestra en la siguiente ilustración.



Posicionamiento de elementos absolutos

Una pequeña aunque importante aclaración técnica: el posicionamiento absoluto no siempre funciona tomando como referencia la esquina superior izquierda de la ventana. Si utilizas el posicionamiento absoluto para situar un bloque A dentro de otro bloque B, que se ha posicionado de manera absoluta, relativa o fija, tu bloque A se posicionará tomando como referencia la esquina superior izquierda del bloque B. ¡Haz la prueba y compruébalo!

Posicionamiento fijo

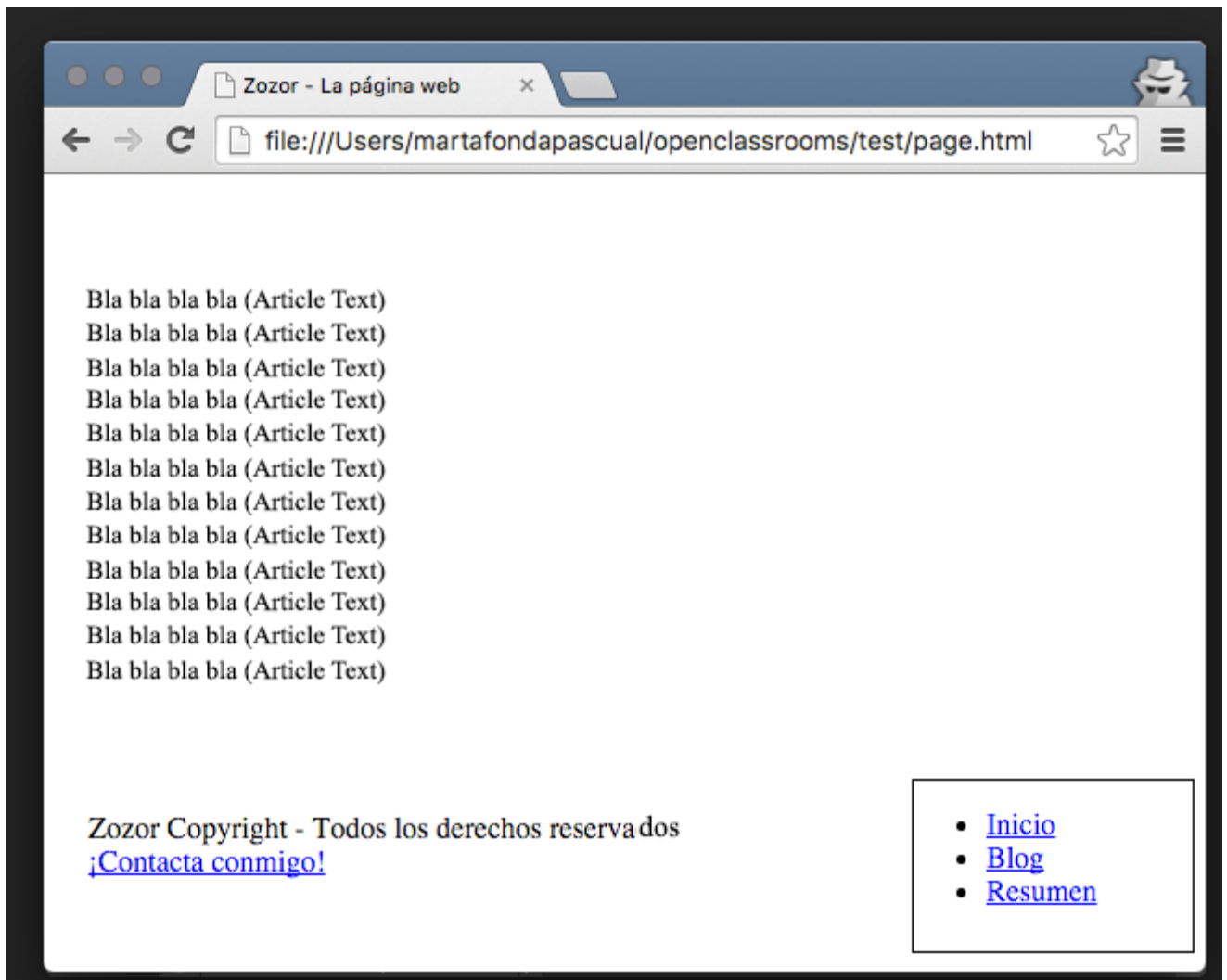
El principio es *exactamente el mismo* que el del posicionamiento absoluto, salvo que esta vez el bloque queda fijo en su posición incluso si te desplazas hacia el pie de la página

```
element
{
    position: fixed;

    right: 0px;

    bottom: 0px;
}
```

Inténtalo y comprueba los resultados. Verás que en este caso el menú sigue mostrándose en la parte inferior derecha incluso si desplazas la página hacia abajo (ilustración siguiente).



Posicionamiento relativo

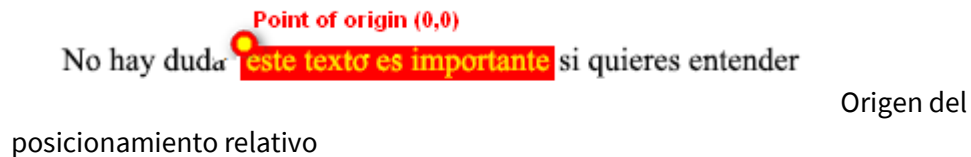
El posicionamiento relativo es algo más complicado y puede dar con facilidad algunas complicaciones. Este tipo de posicionamiento te permite realizar «ajustes»: el elemento está fuera de su posición normal.

Tomemos como ejemplo un texto importante situado entre dos etiquetas `<strong>`. Para empezar, lo pondré sobre un fondo rojo para poder identificarlo mejor.

```
strong
{
    background-color: red; /* Fondo rojo */
    color: yellow; /* Texto amarillo */
}
```

Esta vez, el procedimiento que he mostrado antes de posicionamiento absoluto y fijo no sirve. ¿Por qué? Porque el origen ha cambiado: el punto de coordenadas (0, 0) ya no se encuentra en la parte superior izquierda de la ventana como en el caso anterior. No, esta vez el origen está situado en la parte superior izquierda... de la posición actual de tu elemento.

¿A que es un poco enrevesado? Es el principio de la posición relativa. El diagrama de la siguiente ilustración debería ayudarte a entender dónde se sitúa el origen de los puntos.



Así pues, si introduces `position: relative;` y aplicas cualquiera de las propiedades `top`, `left`, `right` o `bottom`, el texto sobre fondo rojo se moverá tomando como referencia su posición actual.

Utilicemos un ejemplo: quiero desplazar mi texto 55 píxeles a la derecha y 10 píxeles hacia abajo; así, lo que le voy a indicar es que quiero que se sitúe a 55 píxeles del borde izquierdo y a 10 píxeles del borde superior (líneas 6 a 8):

```
strong
{
    background-color: red;

    color: yellow;

    position: relative;

    left: 55px;

    top: 10px;
}
```

El texto estará fuera de su posición inicial, tal como se indica en la siguiente ilustración.



si quieres entender

este texto es importante

Este texto está fuera de

su posición inicial

En resumen

- CSS se utiliza para el diseño de página de un sitio web. Tenemos a nuestra disposición diferentes técnicas.
- El posicionamiento flotante (utilizando la propiedad `float`) es una de las técnicas más utilizadas actualmente. Te permite, por ejemplo, situar un menú a la izquierda o a la derecha de la página. Esta propiedad, sin embargo, no se diseñó en un principio con este fin, y, de ser posible, será mejor evitar esta técnica.
- El posicionamiento `inline-block` consiste en asignar el tipo `inline-block` a nuestros elementos mediante la propiedad `display`. Se comportarán como elementos en línea (posicionamiento izquierda a derecha), pero podrán ser redimensionados como bloques (mediante `width` y `height`). Esta técnica es preferible a la de posicionamiento flotante.
- El posicionamiento absoluto te permite situar un elemento en cualquier lugar de la página, indicando la distancia en píxeles.
- El posicionamiento fijo es parecido al posicionamiento absoluto, pero el elemento permanece visible incluso si te desplazas hacia el pie de la página.
- El posicionamiento relativo permite que un bloque esté fuera de su posición normal.
- Un elemento A, posicionado de manera absoluta dentro de otro elemento B (a su vez posicionado de modo absoluto, fijo o relativo), se posicionará tomando como referencia este elemento B y no la esquina superior izquierda de la página.

Ejercicio práctico: creación de una página web paso por paso

Aquí estamos, por fin. Este capítulo es algo especial y bastante diferente de lo que hemos visto hasta ahora. Es, de hecho, un ejercicio práctico. Ahora ya no vale leerse mis capítulo medio dormido: tendrás que ensuciarte las manos conmigo.

Aunque hasta el momento has leído un montón de teoría, quizá te sigas preguntando cómo crean los webmasters páginas web tan elegantes. Quizá te sigas repitiendo que todavía te queda un largo trecho hasta conseguir el conocimiento necesario para crear toda una página web... ¡Pues te equivocas!

Esbozar el diseño

Me lo puedo imaginar. Estarás pensando «Estupendo, vamos a crear una página web entera», así que abrirás tu editor de texto (Notepad++, por ejemplo), me mirarás y dirás «Vale, ¿por qué línea de código empezamos?»

Y tendré que pararte los pies. Coge lápiz y papel: primera has de pensar qué página de web quieres crear. ¿De qué va a hablar? ¿Tienes algún tema u objetivo?

La experiencia me dice que la mayoría de vosotros solo queréis aprender en esta etapa. y puede que no tengáis ninguna idea concreta en mente. Si es el caso, os sugiero que creéis una página web para presentaros y trabajar vuestra presencia en Internet: esta página hablará sobre vosotros e incluirá vuestra trayectoria profesional, vuestros futuros proyectos y, por qué no, vuestro blog.

Por mi parte, en este ejercicio práctico crearé la página web de nuestra mascota Zozor, el famoso burro (siguiente ilustración). Zozor ha decidido embarcarse en un viaje por todo el mundo y su primera para será... ¡San Francisco! Por eso, quiere crearse una página web para que la gente lo conozca y siga su viaje por todo el mundo.



El burro Zozor

El primer paso será *esbozar el diseño* para seguir un modelo de página web a la hora de crearla. En este punto tenemos dos posibilidades:

- o eres un diseñador gráfico (o conoces a uno) acostumbrado a trabajar diseños con programas como Photoshop;
- no eres demasiado creativo y te falta inspiración, en cuyo caso puedes inspirarte en páginas web como <https://html5up.net/>, de donde los diseñadores pueden sacar ideas e ¡incluso utilizar los códigos HTML/CSS ya diseñados!

En mi caso, consulté a Jiyong Fan, el diseñador gráfico que sugirió el diseño (que por cierto me encanta) que podéis ver a continuación.



Travel diaries

HOME BLOG



Reflections on my holiday in the United States...



I'M A GREAT TRAVELLER

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam nec sagittis massa. Nulla facilisi. Cras id arcu lorem, et semper purus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Duis vel enim mi, in lobortis sem. Vestibulum luctus elit eu libero ultrices id fermentum sem sagittis. Nulla imperdiet mauris sed sapien dignissim id aliquam est aliquam. Maecenas non odio ipsum, a elementum nisi. Mauris non erat eu erat placerat convallis. Mauris in pretium urna. Cras laoreet molestie odio, consequat consequat velit commodo eu. Integer vitae lectus ac nunc posuere pellentesque non at eros. Suspendisse non lectus lorem.

Vivamus sed libero nec mauris pulvinar facilisis ut non sem. Quisque mollis ullamcorper diam vel faucibus. Vestibulum sollicitudin facilisis feugiat. Nulla euismod sodales hendrerit. Donec quis orci arcu. Vivamus fermentum magna a erat ullamcorper dignissim pretium nunc aliquam. Aenean pulvinar condimentum enim a dignissim. Vivamus sit amet lectus at ante adipiscing adipiscing eget vitae felis. In at fringilla est. Cras id velit ut magna rutrum commodo. Etiam ut scelerisque purus. Duis risus elit, venenatis vel rutrum in, imperdiet in quam. Sed vestibulum, libero ut bibendum consectetur, eros ipsum ultrices nisl, in rutrum diam augue non tortor. Fusce nec massa et risus dapibus aliquam vitae nec diam.

Phasellus ligula massa, congue ac vulputate non, dignissim at augue. Sed auctor fringilla quam quis porttitor. Praesent vitae dignissim magna. Pellentesque quis sem purus, vel elementum mi. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Maecenas consectetur euismod urna. In hac habitasse platea dictumst. Quisque tincidunt porttitor vestibulum. Ut iaculis, lacus at molestie lacinia, ipsum mi adipiscing ligula, vel mollis sem risus eu lectus. Nunc elit quam, rutrum ut dignissim sit amet, egestas at sem.

MY LAST TWEET

Hee-haw!

12/05 23:12

MY PICTURES



MY FRIENDS

- » Pupi the rab
- » Mr Baobab
- » Kaiwail
- » Perceval.eu

Diseño del borrador: Fan Jiyong

Este borrador es en realidad solo una foto del resultado que queremos obtener. Le pedí al diseñador gráfico que me facilitara los elementos que me ayudarían a crear el diseño; en otras palabras, los códigos de color usados, las imágenes recortadas (mostradas abajo) y las fuentes que iba a necesitar.

Descargar las imágenes y fuentes



Unas cuantas imágenes recortadas utilizadas en el diseño

¡Y todo lo que hemos de hacer es crear la página web! Lo haremos en dos fases:

1. Crear el boceto **HTML** de la página.
2. Formatearlo y hacer el diseño de la página mediante **CSS**.

Venga, ¡manos a la obra!

Organizar el contenido en HTML

El primer paso es diferenciar los bloques principales en el modelo. Estos bloques conformarán el borrador de nuestra página.

Para crear el borrador, utilizaremos varias etiquetas HTML:

- las etiquetas estructurales HTML5, que ya conocemos: `<header>`, `<section>`, `<nav>`, etc.;
- la etiqueta para fines generales `<div>` cuando no haya una etiqueta estructural adecuada.

¿Cómo sé qué etiqueta he de utilizar?

Eso es decisión tuya. Lo ideal sería utilizar una etiqueta lógica (como las etiquetas estructurales `<header>`, `<section>`, `<nav>`), pero si no hay ninguna que te parezca la adecuada, decántate por la etiqueta genérica `<div>`.

Observa la siguiente ilustración y verás mi sugerencia de estructura.



Travel diaries

<nav>

HOME BLOG

<header>



Reflections on my holiday in the United States...



I'M A GREAT TRAVELLER

<article>

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam nec sagittis massa. Nulla facilisi. Cras id arcu lorem, et semper purus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Duis vel enim mi, in lobortis sem. Vestibulum luctus elit eu libero ultrices id fermentum sem sagittis. Nulla imperdiet mauris sed sapien dignissim id aliquam est aliquam. Maecenas non odio ipsum, a elementum nisi. Mauris non erat eu erat placerat convallis. Mauris in pretium urna. Cras laoreet molestie odio, consequat consequat velit commodo eu. Integer vitae lectus ac nunc posuere pellentesque non at eros. Suspendisse non lectus lorem.

Vivamus sed libero nec mauris pulvinar facilisis ut non sem. Quisque mollis ullamcorper diam vel faucibus. Vestibulum sollicitudin facilisis feugiat. Nulla euismod sodales hendrerit. Donec quis orci arcu. Vivamus fermentum magna a erat ullamcorper dignissim pretium nunc aliquam. Aenean pulvinar condimentum enim a dignissim. Vivamus sit amet lectus at ante adipiscing adipiscing eget vitae felis. In at fringilla est. Cras id velit ut magna rutrum commodo. Etiam ut scelerisque purus. Duis risus elit, venenatis vel rutrum in, imperdiet in quam. Sed vestibulum, libero ut bibendum consectetur, eros ipsum ultrices nisl, in rutrum diam augue non tortor. Fusce nec massa et risus dapibus aliquam vitae nec diam.

Phasellus ligula massa, congue ac vulputate non, dignissim at augue. Sed auctor fringilla quam quis porttitor. Praesent vitae dignissim magna. Pellentesque quis sem purus, vel elementum mi. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Maecenas consectetur euismod urna. In hac habitasse platea dictumst. Quisque tincidunt porttitor vestibulum. Ut iaculis, lacus at molestie lacinia, ipsum mi adipiscing ligula, vel mollis sem risus eu lectus. Nunc elit quam, rutrum ut dignissim sit amet, egestas at sem.

MY LAST TWEET

<div>

Hee-haw!

12/05 23:12

MY PICTURES



MY FRIENDS

- » Pupi the rabbit
- » Mr Baobab
- » Kaiwail
- » Perceval.eu

Se nos pueden ocurrir otras maneras de vertebrar la estructura, ¡así que ten en cuenta que mi opción no es por fuerza la única!

Aunque no todas las etiquetas que vamos a utilizar aparecen en este modelo, te darán una idea de la disposición de elementos que sugiero.

En realidad, el código HTML no es la parte compleja de la construcción de la página web, así que si has entendido la disposición de etiquetas no deberías tener ningún problema al escribir un código similar al mío:

```
<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8" />

    <link rel="stylesheet" href="style.css" />

    <title>Zozor - Travel diaries</title>

  </head>

  <body>

    <div id="main_wrapper">

      <header>

        <div id="main_title">

          <h1>Zozor</h1>

          <h2>Travel diaries</h2>

        </div>

        <nav>

          <ul>
```

```
<li><a href="#">Home</a></li>

<li><a href="#">Blog</a></li>

<li><a href="#">Resume</a></li>

<li><a href="#">Contact</a></li>

</ul>

</nav>

</header>

<div id="banner_image">

    <div id="banner_description">

        Reflections on my holiday in the United States...

        <a href="#" class="red_button">See article </a>

    </div>

</div>

<section>

    <article>

        <h1>I'm a great traveller</h1>

        <p>Lorem ipsum dolor sit amet...</p>

        <p>Vivamus sed libero nec mauris pulvinar facilisis ut non
sem...</p>

        <p>Phasellus ligula massa, congue ac vulputate non, dignissim at
augue...</p>

    </article>

    <aside>
```

```

        <h1>About the author</h1>

        <p id="zozor_picture"></p>

        <p>Let me introduce myself: My name's Zozor. I was born on 23
November 2005.</p>

        <p>A bit meager, is it not? This is why I've now decided to write
my biography to let my readers know who I really am.</p>

        <p></p>

    </aside>

</section>

<footer>

    <div id="tweet">

        <h1>My last tweet</h1>

        <p>Hee-haw!</p>

        <p>12/05 23:12</p>

    </div>

    <div id="my_pictures">

        <h1>My pictures</h1>

        <p></p>

    </div>

    <div id="my_friends">

        <h1>My friends</h1>

```

```

        <ul>

            <li><a href="#">Pupi the rabbit</a></li>

            <li><a href="#">Mr Baobab</a></li>

            <li><a href="#">Kaiwaii</a></li>

            <li><a href="#">Perceval.eu</a></li>

        </ul>

        <ul>

            <li><a href="#">Ji</a></li>

            <li><a href="#">Super cucumber</a></li>

            <li><a href="#">Prince</a></li>

            <li><a href="#">Mr Fan</a></li>

        </ul>

    </div>

</footer>

</div>

</body>

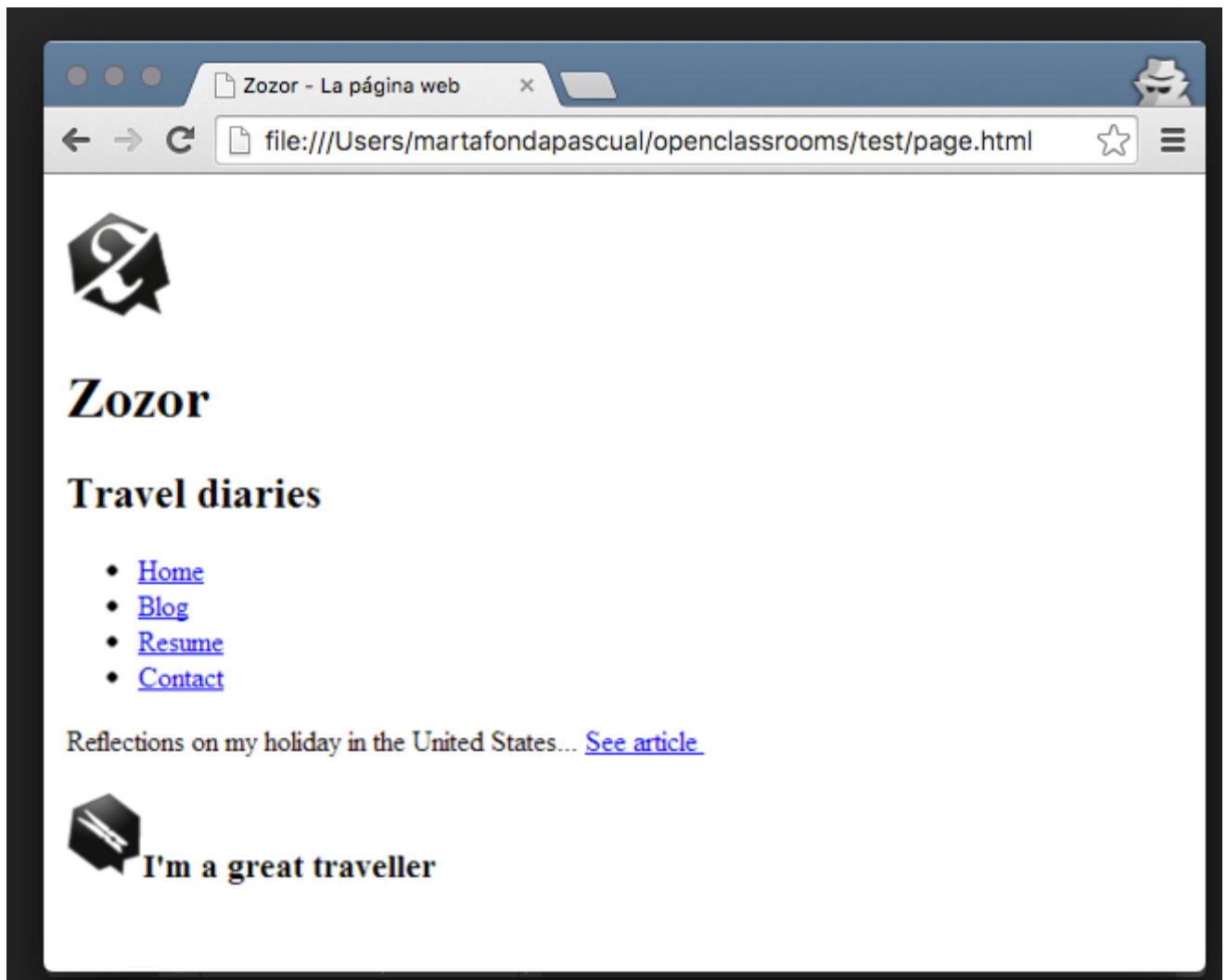
</html>

```

Un pequeño apunte: como ves, todo el contenido de la página está ubicado dentro de una larga etiqueta `<div>conidmain_wrapper`. Esta etiqueta, al alojar todo el contenido, nos permitirá determinar el tamaño de la página fácilmente y centrar nuestra página web en la pantalla.

Por lo demás, no hay ningún problema destacable. Verás que no había planeado todas las etiquetas desde el principio: cuando diseñes en CSS, te das cuenta de que tienes que incluir a veces algunas etiquetas de un bloque `<div>` para ayudarte a construir el diseño.

Estarás de acuerdo en que el aspecto de la página web aún no es nada del otro mundo, por decirlo suave. Puedes observar el resultado actual en la siguiente ilustración.



El aspecto de una página web al escribirla solo en HTML

Es el código CSS el que obra milagros.

Dar formato en CSS

Las cosas se complican algo más al llegar a la etapa de CSS. Bastante, porque lleva esfuerzo (y algunos trucos de vez en cuando) obtener un resultado similar al del modelo. Digo «similar a» porque nunca llegarás a un resultado idéntico píxel por píxel.

Hay que entender esto muy bien: el objetivo es obtener el resultado *más similar* y no perseguir la perfección. Incluso si logras la perfección que buscas en un navegador, ten por seguro que habrá diferencias en otro navegador (más antiguo) o en otro ordenador que no sea el tuyo. Así pues, simplemente nos esforzaremos al máximo, lo que no deja de ser bastante trabajo, como podrás comprobar.

Para dar formato al diseño seguiré varios pasos. Voy a tratar los siguientes elementos, en el orden indicado:

1. Fuentes personalizadas.
2. Definición de los estilos principales de la página (ancho de la página web, color de fondo, color de texto predeterminado).
3. Encabezado y enlaces de navegación.
4. Banner (que muestra el puente de San Francisco).
5. Sección principal del cuerpo de la página, en el centro.
6. *Pie de página*.

Fuentes personalizadas

Para el diseño, mi diseñador gráfico ha utilizado tres fuentes en el modelo:

- `Trebuchet MS`(fuente normal);
- `BallparkWeiner`(fuente exótica);
- `Day Roman`(fuente exótica).

Podrás encontrar estas fuentes en el archivo que he dejado para descargar más abajo. Si aún no lo has hecho, te recomiendo encarecidamente que lo hagas ahora.

La mayoría de ordenadores vienen con `Trebuchet MS` (aunque no necesariamente todos, en cuyo caso puedes descargarlo). Las otras dos fuentes, sin embargo, son un poco especiales, y es poco probable que los ordenadores de quien te visite las tengan. Las descargaremos.

Como ya sabes, es necesario tener varias versiones de estas fuentes para diferentes navegadores. `Dafont` solo ofrece el formato `.ttf` para descargar. Por el contrario, `FontSquirrel` ofrece un [generador de fuentes](#) para usar en CSS3 con `@font-face`: le mandas un `.ttf` y la herramienta convierte el archivo al formato necesario, e incluso te proporciona el código CSS ya preparado.

Yo lo he usado para generar las diferentes versiones de las dos fuentes exóticas que voy a utilizar. Después, añado el código que me ha proporcionado `FontSquirrel` a mi archivo CSS para especificar las nuevas fuentes:

```
/* Definir fuentes personalizadas */

@font-face
{
    font-family: 'BallparkWeiner';

    src: url('polices/ballpark.eot');

    src: url('polices/ballpark.eot?#iefix') format('embedded-opentype'),
```

```

        url('polices/ballpark.woff') format('woff'),

        url('polices/ballpark.ttf') format('truetype'),

        url('polices/ballpark.svg#BallparkWeiner') format('svg');

font-weight: normal;

font-style: normal;
}

@font-face
{

    font-family: 'Dayrom';

    src: url('polices/dayrom.eot');

    src: url('polices/dayrom.eot?#iefix') format('embedded-opentype'),

        url('polices/dayrom.woff') format('woff'),

        url('polices/dayrom.ttf') format('truetype'),

        url('polices/dayrom.svg#Dayrom') format('svg');

font-weight: normal;

font-style: normal;
}

```

Aparte de eso, tendrás que hacer que sea posible acceder a los archivos de las fuentes, claro está. Como ves, he creado una subcarpeta `polices` para las diferentes versiones de mis fuentes.

Definir los estilos principales

Ahora podemos definir unos pocos estilos generales para todo el diseño de nuestra página. Vamos a establecer una imagen de fondo, la fuente y el color del texto, y concretaremos el tamaño de nuestra página y la centraremos en la pantalla.

```
/* Elementos principales de la página */
```

```

body
{
    background: url('images/yellow_background.png');

    font-family: 'Trebuchet MS', Arial, sans-serif;

    color: #181818;
}

#main_wrapper
{
    width: 900px;

    margin: auto;
}

section h1, footer h1, nav a
{
    font-family: Dayrom, serif;

    font-weight: normal;

    text-transform: uppercase;
}

```

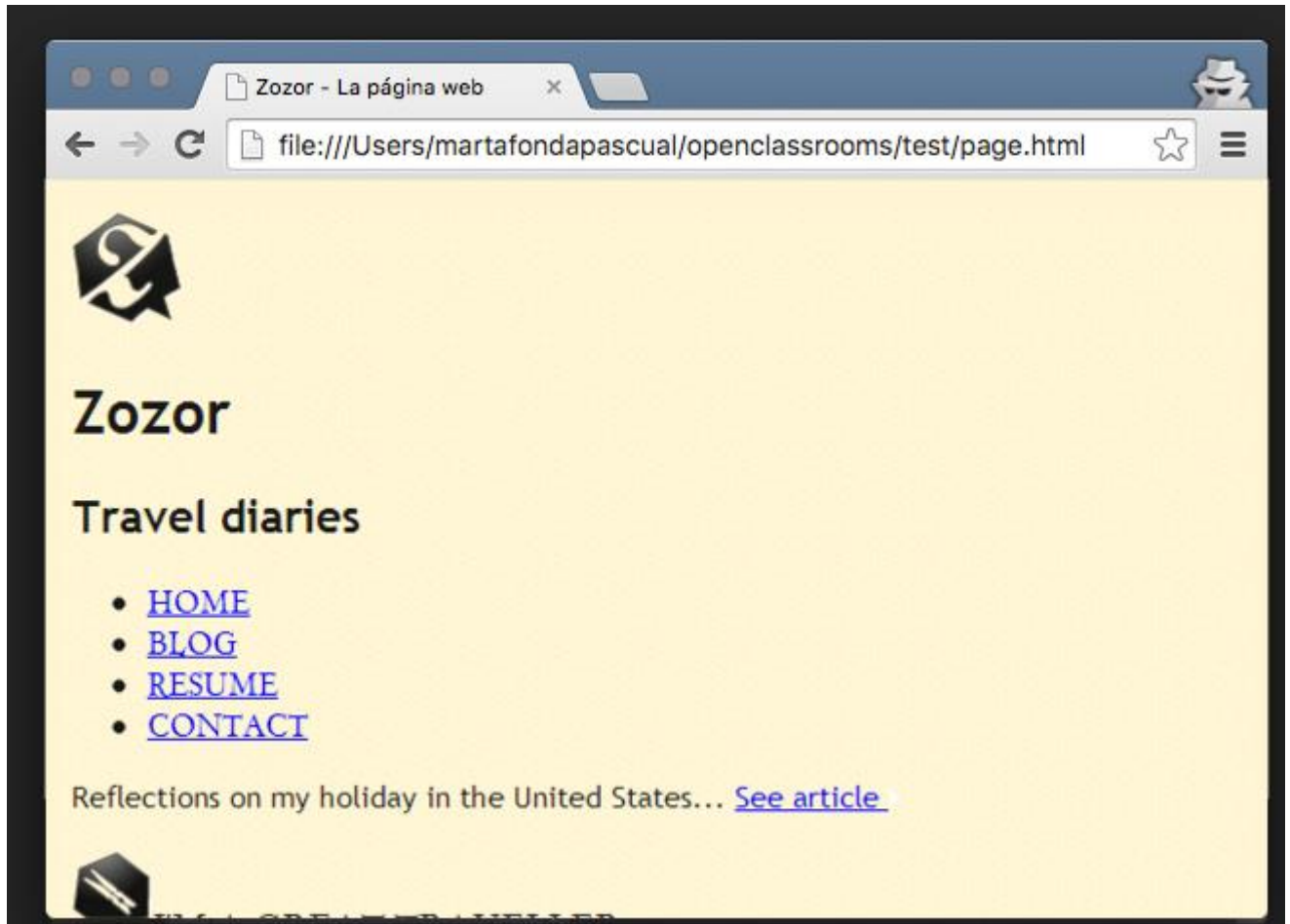
Con `#bloc_page`, el bloque ocupa toda la página, así que limito el ancho a 900 píxeles. Con márgenes automáticos, el diseño será centrado.

Si quieres crear un diseño que quepa en las dimensiones de la pantalla del visitante, establece el ancho como porcentaje en lugar de en píxeles.

He utilizado la propiedad CSS `text-transform: uppercase;` (que ya hemos visto) para asegurar que mis títulos estén siempre en mayúscula. Así pues, esta propiedad

transforma el texto a mayúsculas (también puede hacer lo contrario con `lowercase`). Date cuenta de que también podríamos haber escrito los títulos directamente en mayúsculas en el código HTML.

La siguiente ilustración te muestra lo que has conseguido hasta ahora con el código CSS. Aunque aún nos queda un buen trecho para tener el resultado final, ya parece que nos vamos acercando.



El fondo y los límites de la página comienzan a aparecer

Encabezado y enlaces de navegación

Siguiendo la estructura que he sugerido, el encabezado también contiene enlaces de navegación. Empezaremos estableciendo el encabezado, y, más concretamente, la imagen en la parte superior izquierda. A continuación, veremos cómo dar formato a los enlaces de navegación.

El encabezado

```
/* Encabezado */
```

```
header
```

```
{  
  
    background: url('images/line.png') repeat-x bottom;  
  
}
```

#main_title

```
{  
  
    display: inline-block;  
  
}
```

header h1

```
{  
  
    font-family: 'BallparkWeiner', serif;  
  
    font-size: 2.5em;  
  
    font-weight: normal;  
  
}
```

#logo, header h1

```
{  
  
    display: inline-block;  
  
    margin-bottom: 0px;  
  
}
```

header h2

```
{
```

```
font-family: Dayrom, serif;

font-size: 1.1em;

margin-top: 0px;

font-weight: normal;

}
```

Vamos a resaltar el encabezado frente al cuerpo de la página con una imagen de fondo. Los elementos están posicionados mediante `inline-block` y vamos a personalizar las fuentes y los tamaños. Nada del otro mundo por el momento.

Los enlaces de navegación

El formato de los enlaces de navegación ya es algo más interesante. Como ya has visto, he creado una lista no numerada para los enlaces, pero tal lista se muestra por lo general a lo largo y no a lo ancho. Por suerte, es fácil de cambiar, como verás:

```
/* Navegación */

nav

{

    display: inline-block;

    width: 740px;

    text-align: right;

}

nav ul

{

    list-style-type: none;

}
```

```

nav li
{
    display: inline-block;

    margin-right: 15px;
}

nav a
{
    font-size: 1.3em;

    color: #181818;

    padding-bottom: 3px;

    text-decoration: none;
}

nav a:hover
{
    color: #760001;

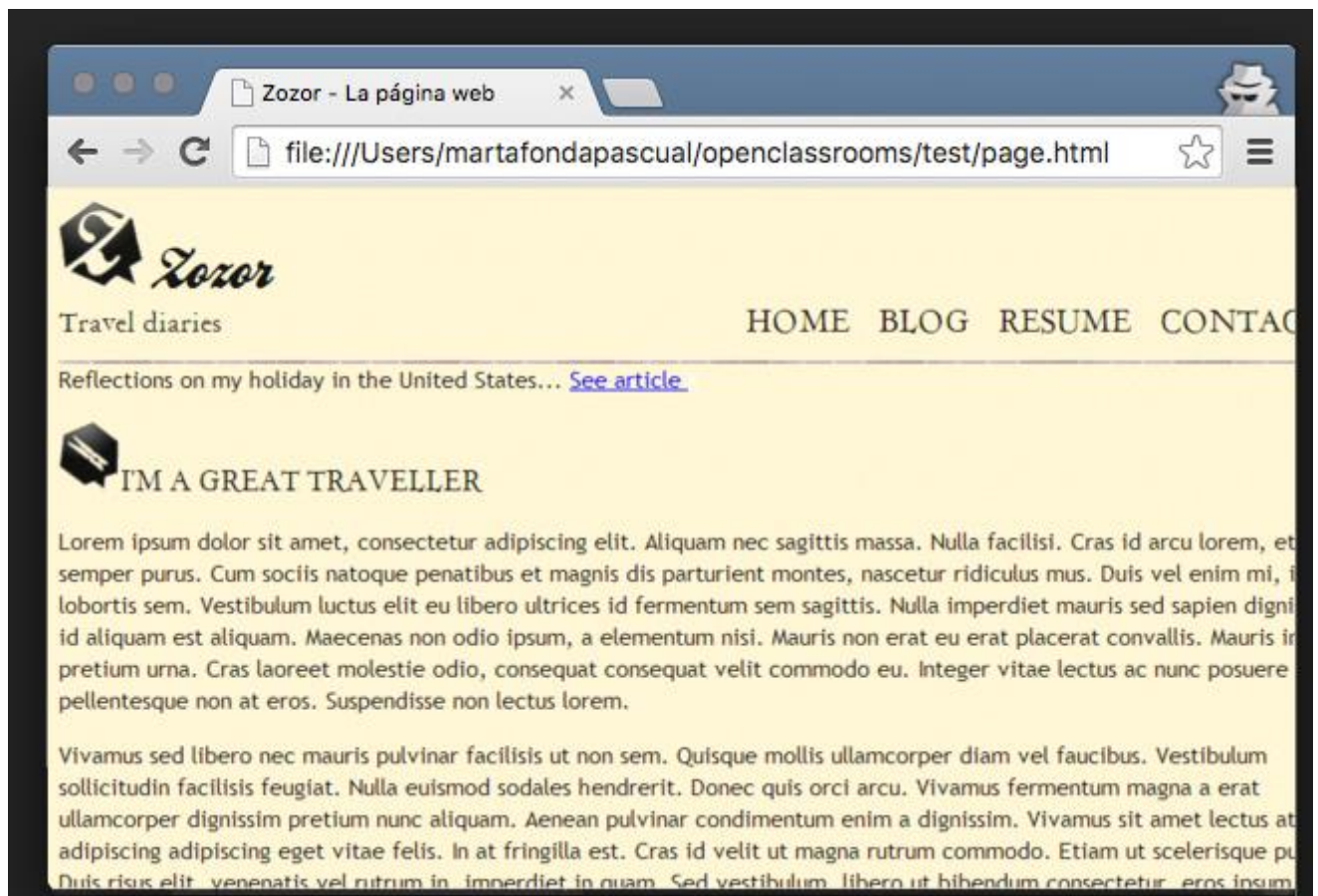
    border-bottom: 3px solid #760001;
}

```

El elemento nuevo aquí es la definición CSS `list-style-type: none;`, que elimina la figura redonda utilizada en la lista. Cada elemento de la lista (`<li>`) se coloca en `inline-block`, lo que nos permite ubicar los enlaces en horizontal a nuestro antojo.

Las demás definiciones no especifican nada fuera de lo normal: dimensiones, colores, bordes... Nada que no sepas ya. Si te das cuenta, puede que no dé con los valores correctos a la primera. A veces es necesario emplear el método de prueba y error para conseguir un aspecto parecido al del modelo original.

La siguiente ilustración muestra el resultado obtenido con los últimos retoques de CSS.



El encabezado ya está dispuesto

El banner

Ahora nos pondremos con otro ejercicio que es algo más complicado pero muy interesante: ¡el *banner*! Nuestro modelo contiene un bonito *banner* en el que aparece el puente de San Francisco). El *banner* de tu página web tal vez sea distinto. Como ejemplo, aquí tenemos el último blog de nuestro amigo Zozor, que acaba de pasar por San Francisco.

El *banner* es interesante por varios motivos:

- tiene bordes redondos;
- la descripción está escrita sobre un fondo ligeramente transparente;
- el enlace «See the article» está introducido en CSS, con esquinas redondas;
- el *banner* lleva un sombreado que le da volumen.

Aquí está el código que he usado para crear el *banner*:

```
/* Banner */
```

```
#banner_image

{

    margin-top: 15px;

    height: 200px;

    border-radius: 5px;

    background: url('images/sanfrancisco.jpg') no-repeat;

    position: relative;

    box-shadow: 0px 4px 4px #1c1a19;

    margin-bottom: 25px;

}
```

```
#banner_description

{

    position: absolute;

    bottom: 0;

    border-radius: 0px 0px 5px 5px;

    width: 99.5%;

    height: 33px;

    padding-top: 15px;

    padding-left: 4px;

    background-color: rgb(24,24,24); /* Los navegadores antiguos leerán esto */

    background-color: rgba(24,24,24,0.8); /* Los navegadores nuevos leerán esto */

    color: white;

    font-size: 0.8em;
```

```
}

.red_button
{
    display: inline-block;

    height: 25px;

    position: absolute;

    right: 5px;

    bottom: 5px;

    background: url('images/red_background.png') repeat-x;

    border: 1px solid #760001;

    border-radius: 5px;

    font-size: 1.2em;

    text-align: center;

    padding: 3px 8px 0px 8px;

    color: white;

    text-decoration: none;
}

.red_button img
{
    border: 0;
}
```

El código es bastante técnico y está lleno de elementos CSS. Es posiblemente la parte más complicada de crear esta página.

Verás que he optado por mostrar la imagen del puente como imagen de fondo en el bloque `<div>` del *banner*.

También le he aplicado una posición relativa al *banner* sin usar propiedades para cambiar su posicionamiento desplazado de su punto original... ¿Por qué? *Por lo general*, una posición relativa sin un posicionamiento desplazado de su punto original no tiene ningún sentido... Y aun así me ha resultado útil a la hora de situar el enlace «See article» en la parte inferior derecha del *banner*. De hecho, he colocado el enlace dentro, en una posición absoluta.

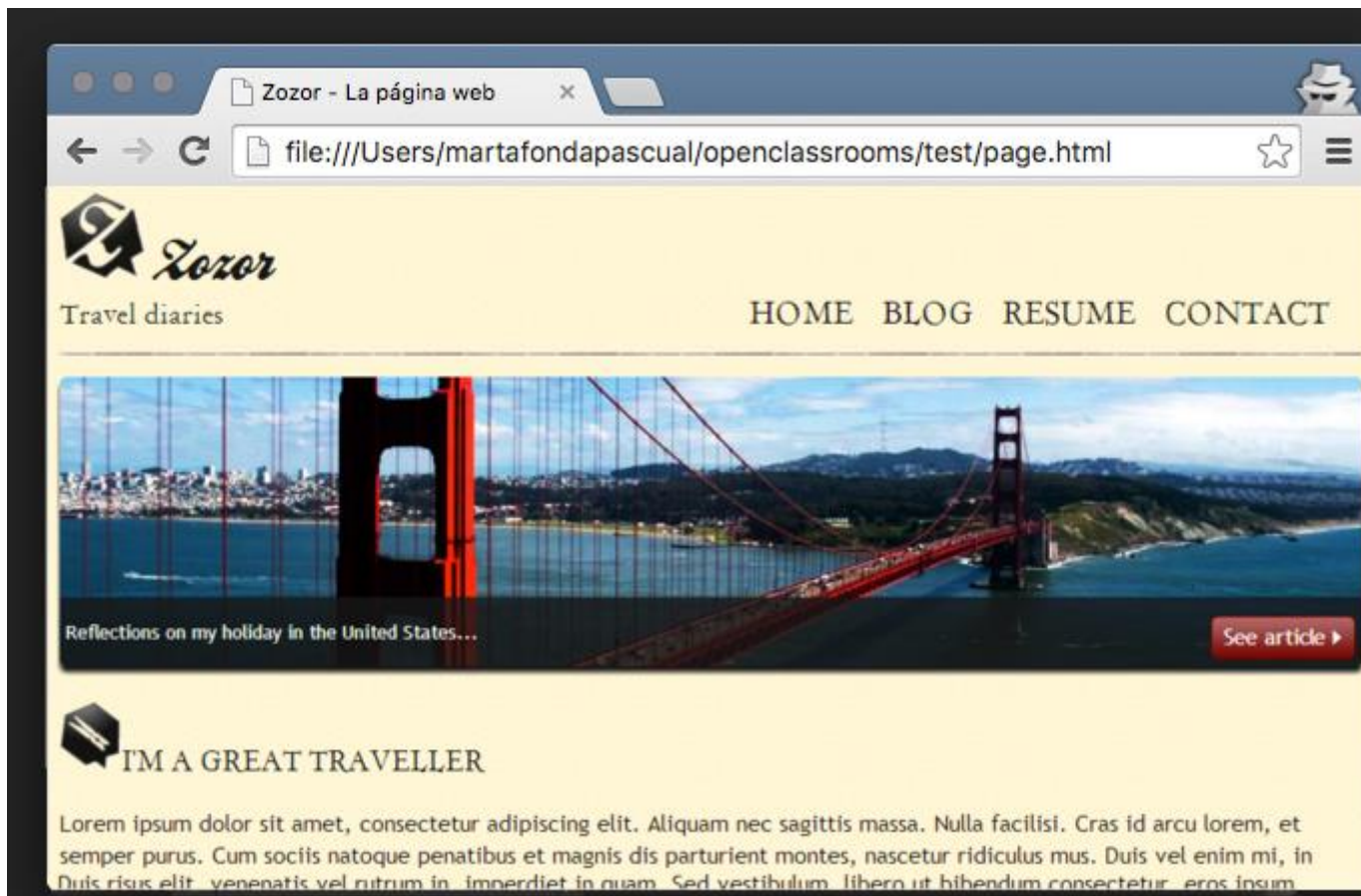
¿Y no debería estar el enlace situado en la parte inferior derecha de la página?

No, recuerda lo que ya he explicado: si un bloque se sitúa en una posición relativa dentro de otro bloque que a su vez tiene una posición absoluta, fija o relativa, el primero se posicionará *dentro* del segundo. Nuestro *banner* tiene una posición relativa (sin desplazamiento de su punto original). Como el enlace tiene una posición absoluta dentro, se colocará en la parte inferior derecha del *banner*.

Esta técnica es particularmente útil y eficiente para crear diseños, ¡así que te convendrá tenerla presente!

Un último detalle: para la leyenda del *banner* he optado por utilizar una transparencia con la notación `RGBA` en lugar de con la propiedad `opacity.opacity`, de hecho, habría hecho que todo el contenido del bloque fuera transparente, hasta el enlace «See article» de debajo. Me ha parecido mejor dejar solo el fondo transparente en vez de extenderlo a todo el bloque.

El resultado es bastante bonito (siguiente ilustración).



El banner ya está dispuesto

¿A que queda genial?

Para producir la gradación de color del botón «See article» he utilizado una imagen de fondo que presenta tal gradación y he repetido la imagen en horizontal. Deberías saber que existe una propiedad CSS3 llamada `linear-gradient` que te permite crear gradaciones sin tener que recurrir a una imagen de fondo. Como utilizarlo será un poco complicado de momento, he decidido no utilizarlo en este ejemplo. ¡Puedes buscar más información si te interesa!

El cuerpo

El cuerpo, en el centro de la página, consiste en este caso en una etiqueta `<section>` (aunque podría haber varias etiquetas, claro está).

No hay verdaderas dificultades por lo que respecta al cuerpo. El bloque «About the author» está situado como elemento *inline-block*. Jugaremos con los bordes redondos y las sombras y ajustaremos los márgenes y el tamaño del texto un poco, ¡y ya lo tenemos!

```
/* Cuerpo */
```

```
article, aside
```

```
{
```

```
    display: inline-block;
```

```
    vertical-align: top;
```

```
    text-align: justify;
```

```
}
```

```
article
```

```
{
```

```
    width: 625px;
```

```
    margin-right: 15px;
```

```
}
```

```
.cat_icon
```

```
{
```

```
    vertical-align: middle;
```

```
    margin-right: 8px;
```

```
}
```

```
article p
```

```
{
```

```
    font-size: 0.8em;
}

aside
{
    position: relative;

    width: 235px;

    background-color: #706b64;

    box-shadow: 0px 2px 5px #1c1a19;

    border-radius: 5px;

    padding: 10px;

    color: white;

    font-size: 0.9em;
}

#arrow
{
    position: absolute;

    top: 100px;

    left: -12px;
}

#zozor_picture
{
```

```

    text-align: center;
}

#zozor_picture img
{
    border: 1px solid #181818;
}

aside img
{
    margin-right: 5px;
}

```

Aquí, la parte difícil era arreglárnoslas para situar la flecha a la izquierda del bloque<aside>«About the author» para crear efecto de burbuja. Y de nuevo, nuestro mejor aliado es el posicionamiento absoluto. La técnica es la misma: He posicionado el bloque<aside>de manera relativa (sin insertar un desplazamiento del punto original), lo que me ha permitido posicionar la imagen de la flecha de manera relativa al bloque<aside>(y no de manera relativa a la página entera). Al ajustar la imagen desplazada de su punto original, puedo situarla con precisión en cualquier píxel que desee (siguiente ilustración).



El cuerpo de la página ya está dispuesto

El pie de página

Tan solo nos queda ocuparnos del pie de página. Está formado por tres sub-bloques que he creado con `<div>`, a los cuales les he asignado `id` para identificarlos mejor. Estos bloques están posicionados uno al lado del otro con `inline-block`.

```
/* Pie de página */

footer

{

    background: url('images/top.png') no-repeat top center, url('images/line.png')
    repeat-x top, url('images/shadow.png') repeat-x top;
```

```
padding-top: 25px;
}

footer p, footer ul
{
    font-size: 0.8em;
}

footer h1
{
    font-size: 1.1em;
}

#tweet, #my_pictures, #my_friends
{
    display: inline-block;

    vertical-align: top;
}

#tweet
{
    width: 28%;
}
```

```
#my_pictures
```

```
{
```

```
    width: 35%;
```

```
}
```

```
#my_friends
```

```
{
```

```
    width: 31%;
```

```
}
```

```
#my_pictures img
```

```
{
```

```
    border: 1px solid #181818;
```

```
    margin-right: 2px;
```

```
}
```

```
#my_friends ul
```

```
{
```

```
    display: inline-block;
```

```
    vertical-align: top;
```

```
    margin-top: 0;
```

```
    width: 48%;
```

```
    list-style-image: url('images/external_link.png');
```

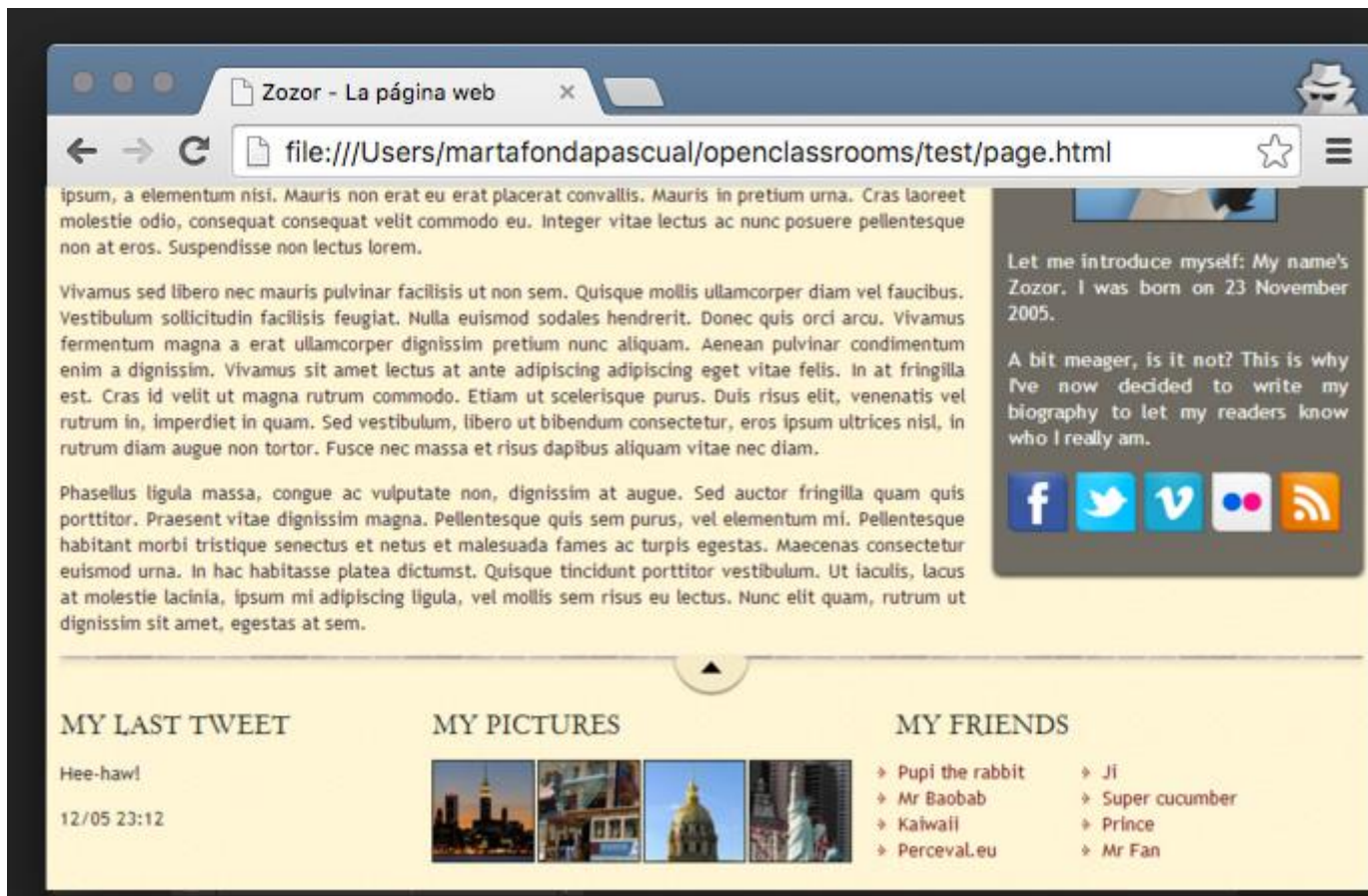
```
    padding-left: 2px;
```

```
}  
  
#my_friends a  
  
{  
  
    text-decoration: none;  
  
    color: #760001;  
  
}
```

Algunos pequeños apuntes sobre el pie de página:

- He utilizado la función de imagen de fondo múltiples de CSS3 para separar el cuerpo del pie de página. Incluye tres imágenes: el separador, la flecha pequeña ascendente y una pequeña gradación.
- He cambiado la viñeta de la lista «My Friends» de la parte inferior derecha mediante la propiedad `list-style-image`, que me ha permitido usar una imagen personalizada en lugar de las viñetas estándar. Hay muchas propiedades CSS específicas como esta y no podemos pararnos a verlas una por una en este curso, pero ahora que ya te has hecho con CSS no te costará nada aprender a usarlas leyendo [lista del anexo con las principales propiedades CSS](#).

Nuestro diseño ha quedado terminado (siguiente imagen).

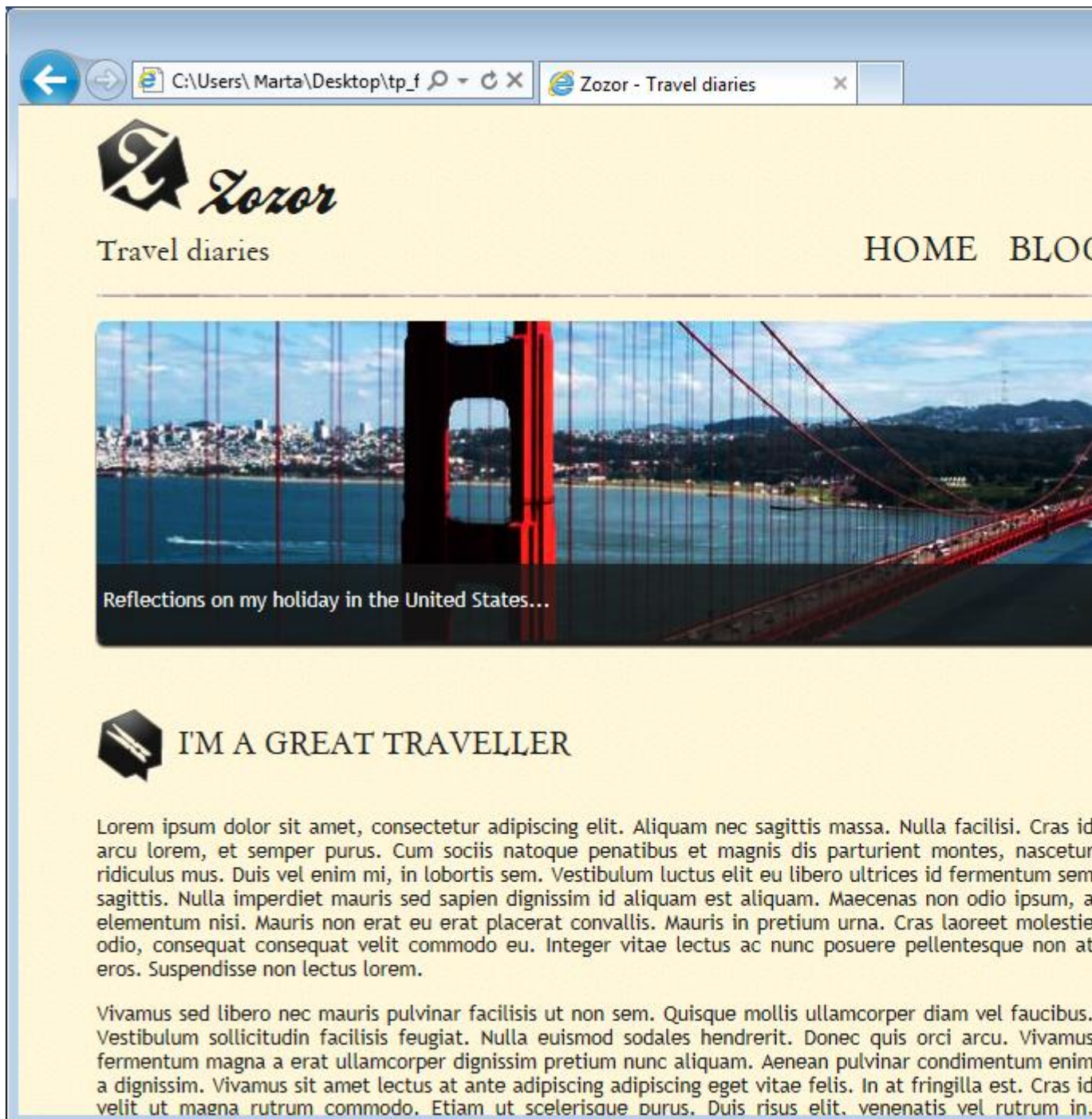


El pie de página ya está dispuesto

Ah, ¿que pensabas que habíamos acabado? Por desgracia, todavía nos queda algo de faena: hemos de probar nuestra página web en varios navegadores. Lo ideal es hacerlo en cada fase del proceso de creación del diseño. Concretamente, las versiones antiguas de Internet Explorer (IE6 a IE8) necesitan de cierta atención, ya que el resultado puede no ser el esperado...

Asegurar la compatibilidad con IE

Tras Internet Explorer 9 (IE9) no ha habido más motivos para quejarnos sobre el mítico *lag* de Internet Explorer al lidiar con CSS. Observa el resultado tú mismo (siguiente ilustración). Queda muy bien en este navegador y no necesita ningún ajuste.



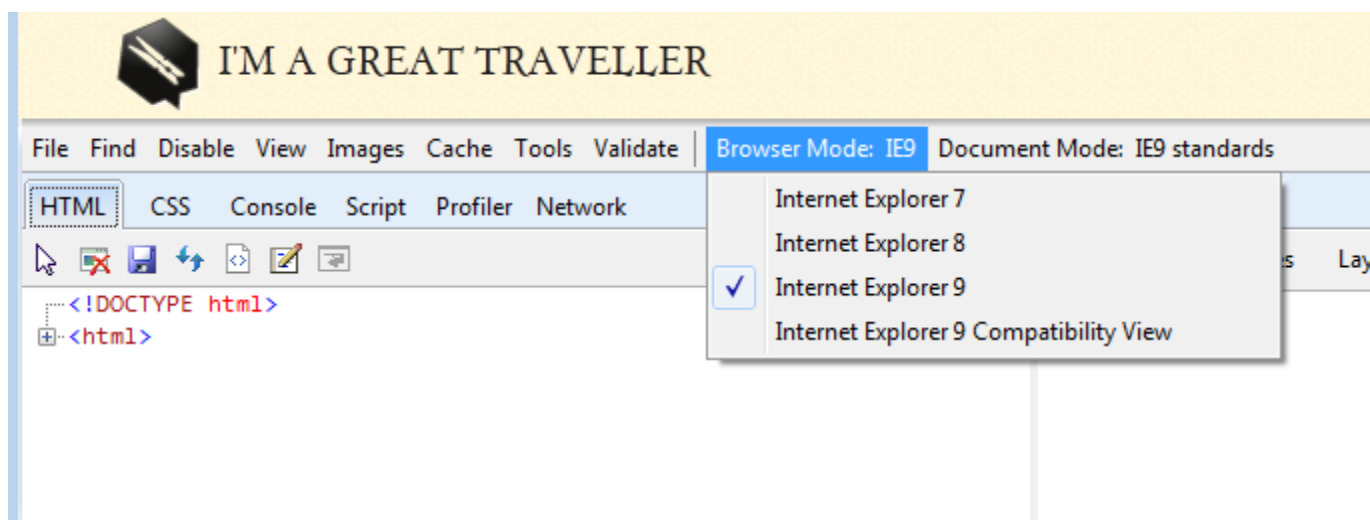
La página web en IE9: ¡ningún problema a la vista!

Pero ver el resultado en las versiones antiguas de Internet Explorer sí te puede crear un trauma severo.

¿Cómo puedo ver cuál sería el resultado en los IE 6 a 8 si tengo IE9?

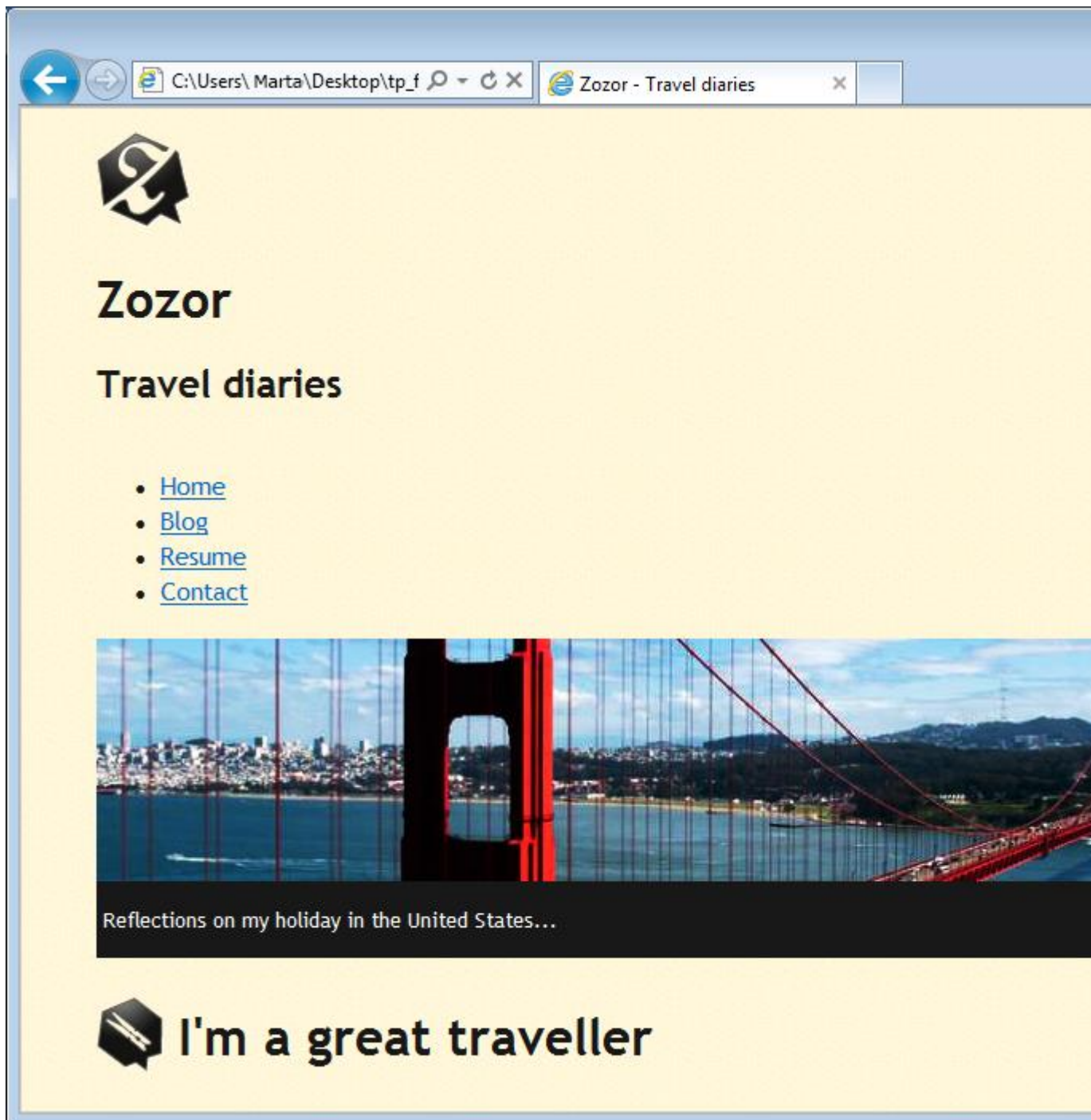
Te he hablado de IETester, una herramienta práctica aunque inestable (se cuelga a menudo). Lo puedes probar para verificar tu página en las versiones antiguas de IE.

Existe, no obstante, una versión más estable y rápida: presiona la tecla **F12** de tu teclado cuando estés en IE9 y aparecerá una barra de desarrollo web. En ella, un menú te permitirá cambiar el comportamiento de IE (siguiente ilustración) para simular las versiones antiguas (a partir de IE7).



Modificación del motor de renderizado en IE

Suele ser una experiencia un poco traumática.
La siguiente ilustración te muestra lo que verías en modo «IE7».



La página web en IE7: ¡nada está como toca!

Antes de que hagas las maletas y te escapes de este mundo cruel a Timbuktu, déjame animarte con un comentario tranquilizador: *existe una solución para cada problema* (repítetelo cuantas veces sea necesario).

De hecho, nuestra página web solo se ha topado con dos problemas en las versiones antiguas de IE:

- El posicionamiento `inline-block` no da buenos resultados en IE6 e IE7, con el resultado de que la mayoría de nuestras etiquetas de posicionamiento no funcionen por el momento... Sin embargo, ¡hemos visto una solución alternativa que corrige este defecto sin demasiadas dificultades!
- Las etiquetas estructurales HTML5 (`<header>`, `<nav>`, `<aside>`, etc.) no dan buenos resultados en IE6, IE7 e IE8, y traen serios problemas de visualización... Aunque, de nuevo, ¡este problema se puede arreglar añadiendo un pequeño *script* al principio del código HTML!

Por otro lado, tendrás que olvidarte de las funciones más recientes de CSS3 no compatibles con estas versiones antiguas:

- esquinas redondas;
- múltiples imágenes de fondo;
- transparencia;
- sombreados.

Como son funciones de aspecto, no intentaremos que funcionen en las versiones más antigua de IE. Aunque si realmente te empeñas en conseguirlo, de nuevo puedo decirte que existen *scripts* que pueden emular la mayoría de las funciones que fallan, pero hará falta un montón de trabajo extra y tu página web tal vez vaya más lenta en estos navegadores. Mientras tu página web se pueda leer en las versiones más antiguas de IE, te recomiendo que no sufras demasiado por estos problemas.

El asumir que tu página web «no va a quedar tan bien» recibe el nombre de **tolerancia elegante**. Significa que, aunque no busques los mismos efectos, persigues una página web que aun así se pueda leer en los navegadores más antiguos.

Ejecutar etiquetas estructurales HTML5

Tal como hemos visto, basta con añadir un simple fragmento de código JavaScript en el encabezado de la página web (líneas 6 a 8):

```
<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8" />

    <link rel="stylesheet" href="style.css" />

    <!--[if lt IE 9]>

      <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>

    <![endif]-->

    <title>Zozor - Travel diaries</title>

  </head>
```

El archivo JavaScript se descargará de los servidores de Google: es una tecnología rápida que evita que tengas que hacerte cargo del archivo tú. Si quieres, puedes hacerle una copia y guardarla junto a los archivos de tu página web.

Ajustar la posición inline-block

Hemos visto que para utilizar la posición `inline-block` había que crear una hoja de estilo especial para Internet Explorer. Se ha de utilizar CSS ligeramente diferentes para asegurarnos de que estas versiones de IE «entienden» qué hay que hacer.

Al crear una hoja de estilo especial para las versiones más antiguas de IE (que podrías llamar `style_ie.css`) y utilizar la siguiente técnica, podrás reproducir el comportamiento de las etiquetas `inline-block`:

```

<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8" />

    <link rel="stylesheet" href="style.css" />

    <!--[if lte IE 7]>

    <link rel="stylesheet" href="style_ie.css" />

    <![endif]-->

    <title>Zozor: la página web</title>

  </head>

```

La hoja de estilo `style_ie.css` contendrá instrucciones de este tipo:

```

element
{
  display: inline;

  zoom: 1;
}

```

Esta técnica se deberá aplicar a cada elemento posicionado mediante `inline-block`.

Hay más diferencias en las versiones más antiguas de IE: el texto no está siempre a la derecha, algunos bloques también tienen dimensiones incorrectas, etc. Tales diferencias han de solucionarse siguiendo una metodología individual para cada caso con la hoja de estilo `style_ie.css`.

Comprobar el funcionamiento

W3C ofrece una herramienta llamada *Validator* en su página web. El validador es un tipo de programa que analiza tu código fuente y te dice si está escrito correctamente o si contiene errores que has de arreglar.

Recuerda: W3C crea estándares que hay que respetar para garantizar que todas las páginas web hablan la misma «lengua».

Existe un validador HTML y un validador CSS (¡que pasará a ser parte de tus «Favoritos»!). El validador CSS tiene algunos errores (marca algunas hojas CSS que son válidas como inválidas) en los que no nos vamos a detener aquí. El validador HTML, sin embargo, nos serán muy útil: aquí está el enlace <http://validator.w3.org>.

Puedes validar tu página web de tres maneras distintas, razón por la cual hay tres pestañas diferentes:

- dirección (URL);
- enviar el archivo `.html`;
- copiar y pegar el código HTML.

Nuestra página web todavía no está disponible en Internet, ya que no tiene dirección URL. La mejor opción es enviar el archivo `.html` que hemos creado o copiar y pegar el código HTML directamente.

Si envías el código HTML y todo marcha como debería, el validador contestará con el mensaje que se muestra en la siguiente ilustración.

This document was successfully checked as HTML5! El

validador W3C nos informa de que nuestra página no tiene ningún error

¡En este caso, significa que todo está bien y que has creado tu página web correctamente!

Por desgracia, a menudo te encontrarás con errores, y si esto pasa, no entres en pánico en plan:

¡AYUDAAAA! ¡Mi página web no es válida, no sé qué hacer, la página está llena de errores, haced algo, ayudadmeeee!

Tienes una página web bonita, que se ve bien, es agradable a la vista, y aun así el validador envía un mensaje preocupante en rojo diciéndote que tu página web no está creada correctamente.

Lo primero de todo, quédate bien con lo siguiente: que tu página web se vea correctamente no significa que no contenga errores. Tu página web puede ser muy elegante y aun así tener un montón de errores.

¿Qué sentido tiene corregirlos entonces?

La cuestión es que los buscadores «tratan» de no mostrar los errores cuando los encuentran para no molestar al usuario. ¡Pero quizá otros navegadores sí que hagan cosas raras!

Tener una página web válida hará que te quedes tranquilo sabiendo que has hecho las cosas como toca, porque simplifica el trabajo de los programas que leen páginas web.

Además, y esto se ha demostrado, es probable que una página web creada correctamente esté mejor posicionada entre los resultados de búsqueda de Google, ¡lo que se traducirá en más visitas!

Aquí tienes una lista de consejos que te ayudarán a resolver errores que puedan surgir antes o después.

- Todos tus textos deben, por lo general, estar entre etiquetas de párrafo. No puedes poner texto directamente entre etiquetas `<body></body>` sin haber puesto antes las famosas etiquetas `<p></p>`. Esto también se aplica al salto de línea `<br />`, que deberá estar dentro de los párrafos. Es un error muy frecuente entre principiantes.

```
<p>Este texto está colocado correctamente en un párrafo.  
  
<br />  
  
No olvides que las etiquetas <br /> deben estar dentro de un párrafo</p>  
  
Este texto está dentro de un párrafo. No está permitido.  
  
<br />
```

- *Todas tus imágenes deben tener un atributo `alt` que especifique qué contiene la imagen.* Si por casualidad la imagen fuera puramente decorativa (no puedes encontrarle una descripción), se permite que no introduzcas ningún valor para el atributo `alt`.

```
<!-- La imagen contiene una descripción -->  
  

```

```
<!-- La imagen no tiene descripción pero tiene un atributo alt -->  

```

- Tus etiquetas han de estar cerradas *en el orden correcto*.

```
<!-- Las etiquetas no están cerradas en el orden en que fueron abiertas -->  
<p>Texto <em>importante</p></em>  
  
<!-- Las etiquetas están cerradas en el orden en que fueron abiertas-->  
<p>Texto <em>importante</em></p>
```

Memoriza este esquema, porque es el error de muchos principiantes.

- Si tus links tienen `&`, tendrás que sustituirlo por el código de `&` para evitarle confusiones al navegador.

```
<!-- Ejemplo de un enlace incorrecto en HTML -->  
<a href="http://www.site.com/?day=15&month=10&year=2000">  
  
<!-- Ejemplo de un enlace correcto en HTML -->  
<a href="http://www.site.com/?day=15&amp;month=10&amp;year=2000">
```

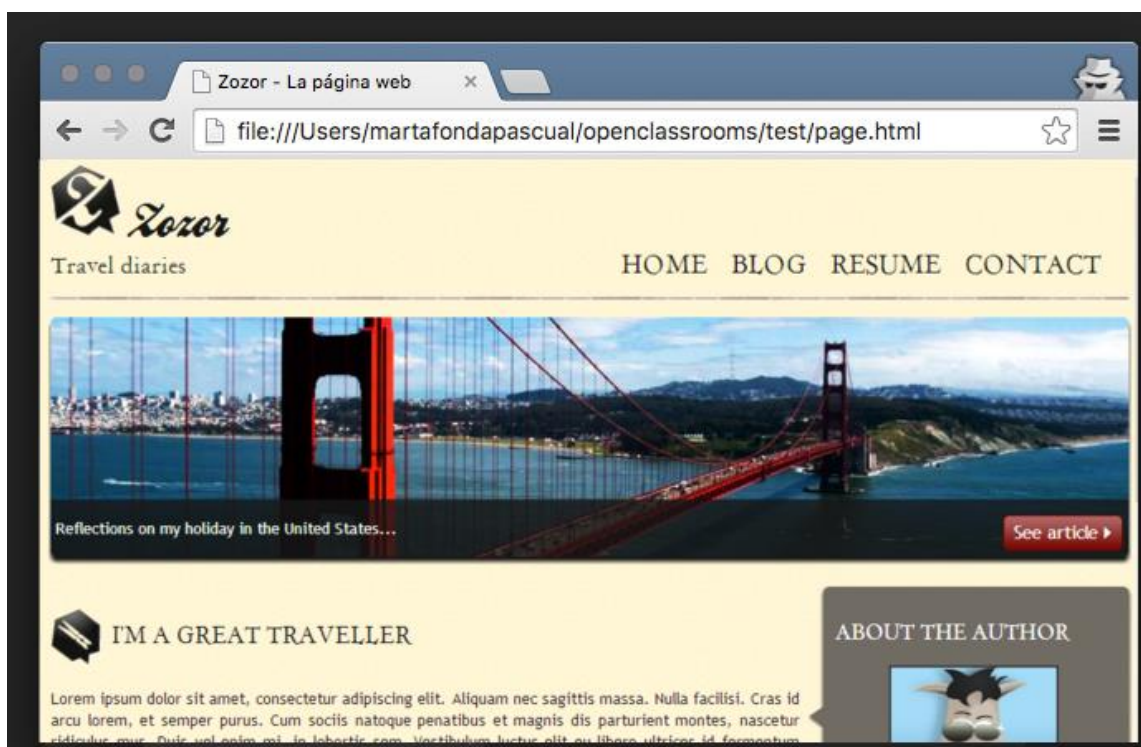
- Por último, asegúrate de no usar etiquetas antiguas y obsoletas en HTML5 (como la etiqueta antigua `<frame>`, o la etiqueta `<marquee>`, etc.).

El validador te indicará «Element XXX undefined» (etiqueta desconocida) y «There is no attribute XXX» (atributo desconocido).

Todos cometemos errores, así que no pierdas la calma. Corrige tu página web paso por paso hasta que el validador muestre un resultado correcto en verde.

El código final

He facilitado el código final de la página que hemos creado. También puedes ver el resultado (siguiente ilustración) en línea a través de un código web.



[Apariencia final de la página web ¡Pruébala!](#)

También puedes descargarte un archivo ZIP con todos los archivos de la página para probarlo en casa:

[Descargar los archivos de la página web \(500 KB\)](#)

Para asegurar que la página web funcione lo mejor posible en las versiones antiguas de Internet Explorer, verás que este código final no incluye ninguna `hojastyle_ie.css`. En su lugar he usado otra técnica que asigna una clase especial (como `.ie7`) a la etiqueta `<body>` solo para las versiones antiguas de IE, que me dice qué versión del buscador se usa en el archivo CSS (`.ie7 footer` permite, por ejemplo, cambiar el estilo del pie de página en IE7).

Quiz 3

• Pregunta 1

¿Cuál de estas etiquetas corresponde al pie de página?

- ☐ `<footer>`
- ☐ `<bottom>`
- ☐ `<nav>`
- ☐ `<div>`

• Pregunta 2

La etiqueta `<nav>` no funciona en versiones antiguas del navegador IE, a menos que se añada un script al código. ¿Cuál es este script?

- ☐ `jQuery`
- ☐ `html-ie`
- ☐ `html5shiv`
- ☐ `htmlcompatibility`

• Pregunta 3

¿Puedes incluir un `<header>` dentro de una `<section>`?

- ☐ Sí, puedes, pero no tiene sentido.
- ☐ Sí, puedes, y puede ser útil.
- ☐ No, no puedes y no deberías hacerlo

• Pregunta 4

No se pueden asignar dimensiones a una de las siguientes etiquetas... ¿Cuál de ellas?

- ☐ `<a>`
- ☐ `<div>`
- ☐ `<p>`
- ☐ `<h6>`

• Pregunta 5

¿Qué propiedad CSS te permite cambiar el margen interno derecho de un elemento?

- ☐ margin-left
- ☐ padding-right
- ☐ margin-right
- ☐ inner-top

• Pregunta 6

¿A qué valor deberías asignar la propiedad `overflow` para esconder el texto que rebosa de una caja sin añadir ningún scroll bar?

- ☐ hidden
- ☐ visible
- ☐ auto
- ☐ scroll

• Pregunta 7

¿Cuál es el rango del ancho de un elemento que tiene un `min-width` de 200 px y un `width` del 50%?

- ☐ Cualquier ancho mayor o igual a 200 px
- ☐ de 100px a 200px
- ☐ de 0px a 200px
- ☐ de 200px a 400px

• Pregunta 8

¿Cómo puedes hacer un link `<a>` comportarse como una etiqueta `<div>`?

- ☐ `transform: div;`
- ☐ `display: inline-block;`
- ☐ `display: block;`
- ☐ `visibility: full;`

• Pregunta 9

Quiero aplicar un offset a un elemento de 10 px ¿Qué valor de la propiedad `position` puedo utilizar?

- ☐ relative
- ☐ absolute
- ☐ initial
- ☐ fixed

• Pregunta 10

¿Qué website provee una herramienta para comprobar la validez de tu código HTML?

- ☐ iso.org
- ☐ html5.net
- ☐ w3schools.com
- ☐ w3.org

html5es-part3 Assessment

Por ahora nuestros CVs se muestran un tanto "verticales"... ¿Qué tal si los estructuramos para mejorar la experiencia de lectura del CV y aprovechamos todo el potencial de las funcionalidades del CSS?

Voy a pedirte que estructures el CV de la siguiente manera:

- En la izquierda, una barra lateral (Puramente decorativa, pero puedes mostrar algo de información en ella si quieres)
- En la derecha, los contenidos de tu CV incluyendo las siguientes secciones de izquierda a derecha
 - - Mi experiencia
 - Mis habilidades
 - Mi formación

Tu página debería mostrarse así:

Mathieu Nebra

Entrepreneur, author and co-founder of OpenClassrooms

My experience

- **From 1999 to 2007:** founder of the *Site du Zéro*, a website of online courses for beginners. Website development and maintenance on evenings and weekends while studying.
- **From 2007 to 2013:** creation of the startup Simple IT to manage the *Site du Zéro*. A collection of books is launched. The website reaches 2 million visitors per month and the company grows to 20 employees.
- **From 2014 to today:** Simple IT and the *Site du Zéro* merge into a unique and global label: OpenClassrooms. Internationalization, launch of video courses, eBooks on demand and online certificates.

My skills

- HTML5 and CSS3
- PHP ([OpenClassrooms certificate](#))
- SQL
- Super Smash Bros (expert level)

My education

Well it's very simple: I learned everything on OpenClassrooms!

Puedes establecer altura absoluta (En píxels) a tu barra, pero deberías utilizar valores relativos (porcentajes) para todo lo demás. Los contenidos deben de tener todo el ancho disponible de pantalla, independientemente del ancho de la ventana.

Para hacer estos cambios de diseño, deberás trabajar en tu fichero CSS, pero por supuesto puedes actualizar el fichero HTML si lo necesitas.

Tablas

Aunque las tablas son esenciales para organizar información, son algo complicadas de crear en HTML: es la razón por la que he esperado hasta ahora para hablar de ellas. Y lo son porque tendrás que incrustar etiquetas HTML nuevas en una instrucción específica.

Empezaremos creando tablas básicas y pasaremos poco a poco a aspectos más complicados: combinar celdas, división en secciones múltiples... También vamos a tratar las propiedades CSS de tabla, que usamos para personalizar su aspecto.

Una tabla simple

La primera etiqueta que te interesa es `<table>` `</table>`. Esta es la etiqueta que define el comienzo y final de una tabla.

Es una etiqueta de bloque, así que tendrá que situarse fuera de un párrafo. Ejemplo:

```
<p>Esto es un párrafo anterior a la tabla.</p>

<table>

  <!-- Inserta aquí el contenido de la tabla -->

</table>

<p>Esto es un párrafo posterior a la tabla.</p>
```

Vale, ¿qué escribes dentro de la tabla?

Llegados aquí, prepárate para una nueva avalancha de etiquetas.

Comenzaremos con calma. Aquí tienes dos etiquetas nuevas muy importantes:

- `<tr>` `</tr>`: define el comienzo y el final de una fila de la tabla.
- `<td>` `</td>`: define el comienzo y el final del contenido de una celda.

En HTML, las tablas se crean fila por fila. En cada fila (<tr>) se especifican los contenidos de las celdas (<td>).

Una tabla se construye básicamente tal y como se muestra en la siguiente ilustración.

Nombre	Edad	País
Anne	27 years old	France
Carmen	33 years old	Spain
Michelle	26 years old	United States
Ogasaku Nyagatosoka	18 years old	Japan

Una tabla, con celdas en

línea

Contiene una etiqueta de fila (<tr>) que incluye un grupo de celdas (<td>).

Por ejemplo, si quiero crear una tabla con dos filas y tres celdas por fila (tres columnas, por tanto) tendré que escribir lo siguiente:

```
<table>

  <tr>

    <td>Carmen</td>

    <td>33 years old</td>

    <td>Spain</td>

  </tr>

  <tr>

    <td>Michelle</td>

    <td>26 years old</td>

    <td>United States</td>

  </tr>

</table>
```

El resultado es un poco lúgubre (siguiente ilustración).

Carmen 33 years old Spain
Michelle 26 years old United States

Una tabla sin bordes

¿Y a eso llamas una «tabla»?
¡El texto sigue y sigue y ni siquiera hay bordes!

Sí, una tabla sin CSS tiene una pinta bastante vacía. Y con toda la razón, porque añadir bordes es muy sencillo. ¡Ya conoces el código CSS correspondiente!

```
td /* Todas las celdas de la tabla... */  
  
{  
  
    border: 1px solid black; /* tendrá un borde de 1 px */  
  
}
```

La siguiente imagen muestra el resultado.

Carmen	33 years old	Spain
Michelle	26 years old	United States

Cada celda tiene su propio borde

Hum, aún no está tan perfecto como querías. De hecho, en realidad nos gustaría tan solo un borde entre dos celdas y aquí no es el caso.

Afortunadamente, hay una propiedad específica para tablas en CSS, `border-collapse`, que implica «pegar los bordes».

Esta propiedad puede tener dos valores:

- `collapse`: los bordes se pegan, es el efecto que estamos buscando;
- `separate`: los bordes se separarán (valor por defecto).

```
table  
  
{  
  
    border-collapse: collapse; /* Los bordes de la tabla se pegarán (mejor aspecto) */  
  
}  
  
td  
  
{
```

```
border: 1px solid black;
}
```

La siguiente imagen muestra el resultado obtenido.

Carmen	33 years old	Spain
Michelle	26 years old	United States

Los bordes están pegados

¡Ahora está mucho mejor!

La fila de encabezado

Ahora que tenemos lo que buscábamos, vamos a añadir la fila de encabezado a la tabla. En el siguiente ejemplo, los encabezados son «Nombre», «Edad» y «País». La fila de encabezado se ha creado con una etiqueta `<tr>` como hemos visto hasta ahora, pero las celdas que contiene están esta vez enmarcadas por etiquetas `<th>` y no `<td>`.

```
<table>

  <tr>

    <th>Nombre</th>

    <th>Edad</th>

    <th>País</th>

  </tr>

  <tr>

    <td>Carmen</td>

    <td>33 años</td>

    <td>España</td>

  </tr>

  <tr>

    <td>Michelle</td>
```

```
<td>26 años</td>

<td>Estados Unidos</td>

</tr>

</table>
```

La fila de encabezado es fácil de reconocer por dos razones:

- las celdas contienen etiquetas `<th>`, en lugar de las habituales etiquetas `<td>`;
- es la primera fila de la tabla (hay que remarcarlo, aunque pueda parecer muy obvio).

Ahora que tenemos lo que buscábamos, vamos a añadir la fila de encabezado a la tabla. En el siguiente ejemplo, los encabezados son «Nombre», «Edad» y «País». La fila de encabezado se ha creado con una etiqueta `<tr>` como hemos visto hasta ahora, pero las celdas que contiene están esta vez enmarcadas por etiquetas `<th>` y no `<td>`.

```
<table>

  <tr>

    <th>Name</th>

    <th>Age</th>

    <th>Country</th>

  </tr>

  <tr>

    <td>Carmen</td>

    <td>33 years old</td>

    <td>Spain</td>

  </tr>

  <tr>
```

```

        <td>Michelle</td>

        <td>26 years old</td>

        <td>United States</td>

    </tr>

</table>

```

La fila de encabezado es fácil de reconocer por dos razones:

- las celdas contienen etiquetas `<th>`, en lugar de las habituales etiquetas `<td>`;
- es la primera fila de la tabla (hay que remarcarlo, aunque pueda parecer muy obvio).

Puesto que el nombre de la celda es ligeramente diferente en el encabezado, recuerda actualizar el código CSS para indicarle que aplique un borde en celdas normales y en el encabezado (siguiente imagen).

```

table
{
    border-collapse: collapse;
}

td, th /* Asigna un borde a las etiquetas td Y th */
{
    border: 1px solid black;
}

```

Name	Age	Country
Carmen	33 years old	Spain
Michelle	26 years old	United States

Una tabla con encabezado

Como puedes comprobar, el navegador ha puesto el texto de las celdas de encabezado en negrita. Aunque es lo que hacen los navegadores normalmente, puedes, si quieres, cambiarlo mediante CSS: cambiar el color de fondo, la fuente, los bordes, etc. de las celdas de encabezado.

Título de tabla

Normalmente, todas las tablas deben tener título. El título informa rápidamente al visitante sobre el contenido de la tabla.

Nuestro ejemplo es tan solo una lista de gente.... pero ¿y qué? ¿Qué representa? Sin un título para la tabla, estamos un poco perdidos, como puedes ver.

¡Afortunadamente, existe `<caption>`!

Esta etiqueta se coloca al comienzo de la tabla, justo antes del encabezado. Es la etiqueta que contiene el título de la tabla (imagen siguiente)

```
<table>

  <caption>Passengers of flight 377</caption>


  <tr>

    <th>Name</th>

    <th>Age</th>

    <th>Country</th>

  </tr>


  <tr>

    <td>Carmen</td>

    <td>33 years old</td>

    <td>Spain</td>

  </tr>


  <tr>

    <td>Michelle</td>

    <td>26 years old</td>

    <td>United States</td>
```

```
</tr>
</table>
```

Passengers of flight 377

Name	Age	Country
Carmen	33 years old	Spain
Michelle	26 years old	United States

Una tabla con título

¡Así está un poco más claro!

Ten en cuenta que puedes cambiar la posición del título mediante la propiedad de CSS `caption-side`, que puede tener hasta cuatro valores:

- `top`: el título se situará encima de la tabla (por defecto);
- `bottom`: el título se situará debajo de la tabla.

Una tabla estructurada

Hemos aprendido a hacer tablas pequeñas y simples. Aunque en la mayoría de los casos es suficiente con estas pequeñas tablas, puede que algunas veces necesites tablas más completas.

Vamos a descubrir dos técnicas especiales:

- Las tablas grandes pueden **dividirse** en tres partes:
 - Encabezado;
 - Cuerpo de la tabla;
 - Pie de tabla.
- En algunas tablas, puede que necesites **combinar** celdas.

Dividir una tabla grande

Si la tabla es bastante grande, lo mejor es dividirla en varias partes. Para esto, las etiquetas HTML se utilizan para definir los tres «campos» de la tabla:

- **el encabezado (al principio)**: se define con las etiquetas `<thead></thead>`;
- **el cuerpo (en el centro)**: se define con las etiquetas `<tbody></tbody>`;
- **el pie de tabla (al final)**: se define con las etiquetas `<tfoot></tfoot>`.

¿Qué deberías poner en el pie de tabla? Si es una tabla larga, normalmente copiarás los encabezados. Esto te permite ver a lo que se refiere cada columna, incluso al final de la tabla. Fundamentalmente, una tabla se divide en tres partes como en el ejemplo siguiente.

table header	Name	Age	Country	<thead>
	Carmen	33 years old	Spain	
table body	Michelle	26 years old	United States	
	Francois	43 years old	France	<tbody>
	Martine	34 years old	France	
	Jonathan	13 years old	Australia	
	Xu	19 years old	China	
table footer	Name	Age	Country	<tfoot>

Una tabla dividida

en varias partes

Es un poco confuso, pero es recomendable escribir las etiquetas en el orden siguiente:

1. <thead>
2. <tfoot>
3. <tbody>

Por lo tanto, en el código, primero defines la parte superior, luego la parte final y, por último, la parte principal (<tbody>). No te preocupes, el navegador se ocupará de mostrar cada elemento en el lugar adecuado.

Por lo tanto, este es el código que tienes que introducir para hacer una tabla en tres partes:

```
<table>

  <caption>Pasajeros del vuelo 377</caption>

  <thead> <!-- Pasajeros del vuelo 377 -->

    <tr>

      <th>Nombre</th>

      <th>Edad</th>

      <th>País</th>

    </tr>
```

```
</thead>

<tfoot> <!-- Pie de tabla -->

    <tr>

        <th>Nombre</th>

        <th>Edad</th>

        <th>País</th>

    </tr>

</tfoot>

<tbody> <!-- Cuerpo de la tabla -->

    <tr>

        <td>Carmen</td>

        <td>33 años</td>

        <td>España</td>

    </tr>

    <tr>

        <td>Michelle</td>

        <td>26 años</td>

        <td>Estados Unidos</td>

    </tr>

    <tr>

        <td>François</td>

        <td>43 años</td>
```

```

        <td>Francia</td>

    </tr>

    <tr>

        <td>Martine</td>

        <td>34 años</td>

        <td>Francia</td>

    </tr>

    <tr>

        <td>Jonathan</td>

        <td>13 años</td>

        <td>Australia</td>

    </tr>

    <tr>

        <td>Xu</td>

        <td>19 años</td>

        <td>China</td>

    </tr>

</tbody>

</table>

```

No tienes que utilizar necesariamente estas tres etiquetas (<thead>, <tbody>, <tfoot>) en todas las tablas. De hecho, las utilizarás principalmente si la tabla es muy grande y necesitas organizarla de forma más clara. En tablas «pequeñas», puedes mantener fácilmente la organización más simple que vimos al principio.

3, 2, 1... ¡Combina!

En algunas tablas complejas, necesitarás «combinar» celdas. ¿Un ejemplo de combinar? El ejemplo de la tabla a continuación proporciona una lista de películas y su audiencia prevista.

Movie title	For children?	For teenagers?
Chainsaw massacre	No, too violent	Yes
The care bears go skiing	Yes, suitable	Not violent enough...
Lucky Luke, goes it alone	For the whole family!	

Una tabla que contiene

títulos de películas y su audiencia

En la última película, puedes ver que las celdas se han combinado en una. Es exactamente el efecto que estamos intentando conseguir.

Para combinar, se añade un atributo a la etiqueta `<td>`. Hay dos tipos de combinar:

- **Combinar columnas:** es lo que acabamos de hacer en este ejemplo. Las celdas se combinan horizontalmente.

Se utiliza el atributo `colspan`.

- **Combinar filas:** en este caso, dos filas se combinan en una. Las celdas se combinan verticalmente.

Se utiliza el atributo `rowspan`.

Como sabes, tienes que darle un valor al atributo (ya sea `colspan` o `rowspan`). Tienes que especificar el número de celdas que quieres combinar. En nuestro ejemplo, hemos combinado dos celdas: la celda de la columna «¿Para niños?» y la celda de la columna «¿Para adolescentes?». De esta manera, tenemos que introducir:

```
<td colspan="2">
```

...que significa: «Esta celda es la combinación de dos celdas». Puedes combinar varias celdas a la vez (tres, cuatro, cinco, etc. tantas como quieras).

Aquí está el código HTML que combina las celdas correspondientes a la tabla de arriba:

```
<table>

  <tr>

    <th>Título de la película</th>

    <th>¿Para niños?</th>

    <th>¿Para adolescentes?</th>

  </tr>
```

```

<tr>

  <td>La matanza de Texas</td>

  <td>No, demasiado violenta</td>

  <td>Sí</td>

</tr>

<tr>

  <td>Los osos amorosos se van a esquiar</td>

  <td>Sí, apropiada</td>

  <td>No hay violencia suficiente...</td>

</tr>

<tr>

  <td>Lucky Luke el intrépido</td>

  <td colspan="2">¡Para toda la familia!</td>

</tr>

</table>

```

Nota importante: puedes ver que la línea 19 solo contiene dos celdas en lugar de tres (solo hay dos etiquetas `<td>`). Es bastante normal, ya que he combinado las últimas dos celdas. `<td colspan="2">` especifica que esta celda ocupa el lugar de dos celdas a la vez.

¿Y cómo se procede para combinar verticalmente con `rowspan`?

Es un poco más complicado. En nuestro ejemplo, vamos a «revertir» el orden de nuestra tabla: en lugar de poner los títulos de las películas a la izquierda, los pondremos arriba.

Es otra forma de ver la tabla: en lugar de hacerla en altura, la hacemos en anchura.

En este caso `colspan` deja de ser adecuado y se debería usar `rowspan`:

```

<table>

  <tr>

```

```

    <th>Título de la película</th>

    <td>La matanza de Texas</td>

    <td>Los osos amorosos se van a esquiar</td>

    <td>Lucky Luke el intrépido</td>

</tr>

<tr>

    <th>¿Para niños?</th>

    <td>No, demasiado violenta</td>

    <td>Sí, apropiada</td>

    <td rowspan="2">¿Para toda la familia!</td>

</tr>

<tr>

    <th>¿Para adolescentes?</th>

    <td>Sí</td>

    <td>No hay violencia suficiente...</td>

</tr>

</table>

```

Resultado: ¡las celdas se combinan verticalmente (ejemplo debajo)!

Film title	Chainsaw massacre	The care bears go skiing	Lucky Luke, goes it alone
For children?	No, too violent	Yes, suitable	For the whole family!
For adolescents?	Yes	Not violent enough...	

Las celdas se han combinado verticalmente

Ten en cuenta que se puede cambiar la alineación vertical del texto en las celdas de la tabla usando la opción `vertical-align` que descubrimos en el capítulo de diseño de página.

En resumen

- Una tabla se inserta con la etiqueta `<table>` y se define fila por fila utilizando la etiqueta `<tr>`.
- Cada fila contiene celdas `<td>` (celdas normales) o `<th>` (celdas de encabezado).
- El título de la tabla se define mediante `<caption>`.
- Puedes añadir un borde a las celdas de la tabla mediante `border`. Para combinar bordes, se usa la propiedad de CSS `border-collapse`.
- Una tabla puede dividirse en tres secciones: `<thead>` (encabezado), `<tbody>` (cuerpo) y `<tfoot>` (pie de la tabla). El uso de estas etiquetas no es obligatorio.
- Puedes combinar celdas horizontalmente mediante el atributo `colspan` o verticalmente con `rowspan`. Tienes que especificar cuántas celdas tienen que combinarse.

Formularios

Cualquier página HTML puede mejorarse con formularios interactivos que piden al visitante que rellene información: introducir texto, seleccionar opciones, confirmar con un botón... ¡cualquier cosa es posible!

Sin embargo, estamos alcanzando los límites del lenguaje HTML, ya que después tienes que analizar la información que el visitante ha introducido... y eso no se puede hacer en HTML. Como vamos a ver, los resultados tienen que procesarse en otro lenguaje, como PHP, por ejemplo.

Mientras tanto, tenemos un gran número de etiquetas HTML por descubrir. Bienvenido al maravilloso mundo de los formularios. Un mundo en el que los botones, las casillas de verificación y las listas desplegables coexisten en armonía (bueno, casi).

Crear un formulario

Cuando tienes la repentina necesidad de insertar un formulario en tu página HTML, necesitas empezar escribiendo una etiqueta `<form>` `</form>`. Esta es la etiqueta principal de los formularios, y especifica su comienzo y su fin.

```
<p>Texto antes del formulario</p>

<form>

<p>Texto dentro del formulario</p>
```

```
</form>
```

```
<p>Texto después del formulario</p>
```

Ten en cuenta que tienes que introducir etiquetas de bloque (como `<p>` `</p>`) dentro del formulario si quieres incluir texto en él.

Esta es la estructura básica. Ahora, presta atención: lo que voy a decirte no es fácil, puesto que hemos alcanzado los límites del HTML.

Veamos un ejemplo para clarificar las cosas. Supón que un visitante acaba de incluir un comentario en el formulario, como, por ejemplo, un mensaje que les gustaría publicar en el foro. Este mensaje tiene que *enviarse* para que puedas recibirlo (lógico, ¿verdad?) y mostrarlo a otros visitantes.

Bien, ese es el problema o, más bien, esos son los problemas, que vamos a abordar:

- **Problema n.º 1:** ¿cómo enviar el texto introducido por el visitante? ¿De qué manera?
- **Problema n.º 2:** una vez que los datos se han enviado, ¿cómo los procesas? ¿quieres recibir el mensaje automáticamente mediante correo electrónico, o prefieres un programa que lo guarde en algún sitio y después lo muestre en una página que todo el mundo lo pueda ver?

Para aportar las soluciones a estos dos problemas, tienes que añadir dos atributos a la etiqueta `<form>`:

- **method:** este atributo especifica cómo se van a enviar los datos (solución al **problema n.º 1**). Hay dos maneras de enviar datos a la Web:
 - `method="get"`: este método normalmente es menos adecuado, ya que se limita a 255 caracteres. Esta particularidad proviene del hecho de que la información se enviará a la dirección de la página (`http://...`, aunque este detalle en realidad no tiene importancia por el momento. Te recomiendo que utilices la mayoría de las veces el otro método: `post`.
 - `method="post"`: es el método más usado en formularios, puesto que permite que se envíe un montón de información. Los datos introducidos en el formulario no van a través de la barra de direcciones.
- **action:** es la dirección de la página o programa que va a *procesar* la información (solución al **problema n.º 2**). Esta página te enviará un correo electrónico con el mensaje si eso es lo que quieres o, por otro lado, guardará el mensaje con todos los demás mensajes en una base de datos. Esto no se puede hacer en HTML y CSS, así que normalmente utilizamos otro lenguaje sobre el que puede que hayas escuchado: PHP.

Por lo tanto, ahora vamos a completar la etiqueta `<form>` con los dos atributos que acabamos de ver.

Para `method`, lo has adivinado, voy a introducir el valor `post`.

Para `action`, voy a introducir el nombre de una página ficticia en PHP (`tratamiento.php`). Esta es la página que será solicitada cuando el visitante haga clic en el botón de envío en el formulario.

```
<p>Texto antes del formulario</p>

<form method="post" action="tratamiento.php">

    <p>Texto dentro del formulario</p>

</form>

<p>Texto después del formulario</p>
```

Por el momento, no sabemos lo que pasa dentro de la página `tratamiento.php`: asumamos que esta página está lista y funcionando.

Nuestra prioridad, por ahora, es descubrir en HTML/CSS cómo insertar campos de texto, botones y casillas de verificación en nuestra página web. Y eso es lo que vamos a ver ahora.

Campos de entrada básicos

Por lo tanto, regresemos a algo tangible.

Vamos a revisar varias etiquetas HTML que necesitamos para introducir texto en un formulario.

Deberías saber que hay dos campos de texto diferentes.

- **El campo de texto de una línea:** como su nombre sugiere, solo puedes introducir una única línea en él. Se utiliza para textos cortos, por ejemplo, un nombre de usuario.
- **El campo de texto de líneas múltiples:** este campo de texto permite introducir un gran cantidad de texto en varias líneas, por ejemplo, una tesis sobre el uso de HTML en el desarrollo de países del Sudeste de Asia (tan solo una sugerencia...).

Campo de texto de una línea

Esta imagen a continuación muestra el aspecto de un campo de texto de una línea.



Para insertar un campo de texto en una línea, vamos a utilizar la etiqueta `<input />`.

Encontraremos esta etiqueta varias veces más adelante del capítulo. En cada caso, el que cambia es el valor de `type`.

Para crear un campo de texto de una línea, tienes que introducir:

```
<input type="text" ></code>
```

Aún no es suficiente: tienes que dar un nombre a tu campo de texto. Aunque este nombre no aparece en la página, lo necesitarás más adelante. De hecho, te dirá (en PHP, por ejemplo) de dónde viene la información: sabrás que tal y cual cadena de texto es el nombre de usuario del visitante, otra cadena de texto es la contraseña que han seleccionado, etc.

Para dar un nombre a un elemento de formulario, se utiliza el atributo `name`. En este caso, vamos a asumir que al visitante se le ha pedido que vuelva a introducir su nombre de usuario:

```
<input type="text" name="nombre" ></code>
```

Así que vamos a intentar crear un formulario muy básico con nuestro campo de texto:

```
<form method="post" action="tratamiento.php">

  <p><input type="text" name="username" /></p>

</form>
```

Como de costumbre, te sugiero encarecidamente que intentes este código por ti mismo y veas el resultado.

Etiquetas

Este campo de texto está muy bien, pero si tu visitante se lo encuentra, no sabrá lo que poner. Esta es exactamente la labor de la etiqueta `<label>`:

```
<form method="post" action="tratamiento.php">

  <p>
```

```
<label>Nombre usuario</label> : <input type="text" name="username" />

</p>

</form>
```

Este código proporciona exactamente el mismo resultado que has visto en la imagen anterior.

Pero no es suficiente. Hay que enlazar la etiqueta con el campo de texto. Para hacer esto, tienes que darle un nombre al campo de texto, no con el atributo `name`, sino con el atributo `id` (que puede usarse con todas las etiquetas).

¿Un `name` y un `id` en el campo? ¿No sería redundante?

Sí, de cierta manera. Personalmente, doy el mismo valor a `name` y a `id`; no supone ningún problema.

Para enlazar la etiqueta con el campo, se le tiene un que dar un atributo `for`, que tiene el mismo valor que el campo `id`... Lo mejor es verlo con un ejemplo:

```
<form method="post" action="tratamiento.php">

  <p>

    <label for="username">Nombre usuario</label> : <input type="text"
name="username" id="username" />

  </p>

</form>
```

Intenta hacer clic en el texto «Tu nombre de usuario»: verás que el cursor se sitúa automáticamente en el campo de texto correspondiente.

Algunos atributos adicionales

Puedes añadir varios atributos adicionales a la etiqueta `<input/>` para personalizar su funcionamiento:

- Puedes aumentar el tamaño del campo con `size`.
- Puedes limitar el número de caracteres que pueden introducirse mediante `maxlength`.
- Puedes prerrellenar el campo con un valor por defecto con `value`.
- Puedes dar algún indicio de los contenidos del campo mediante `placeholder`. Esta indicación desaparecerá tan pronto como el visitante haga clic dentro del campo.

En el siguiente ejemplo, el campo de texto contiene una indicación para entender lo que debería escribirse; aunque el campo mide 30 caracteres, no puedes escribir mas de 10 dentro:

```
<form method="post" action="tratamiento.php">

  <p>

    <label for="username">Nombre usuario:</label>

    <input type="text" name="username" id="username" placeholder="Zozor"
size="30" maxlength="10" />

  </p>

</form>
```

Prueba el resultado por ti mismo para ver cómo se comporta el campo. Mientras tanto, la siguiente imagen muestra el aspecto del campo en su estado inicial.

Nombre usuario

Un campo de texto con un indicador (marcador de posición)

Campo de contraseña

Puedes hacer fácilmente que el campo de texto se comporte como un «campo de contraseña», es decir, un campo donde no puedes ver en la pantalla los caracteres que se escriban. Para crear este tipo de campo de entrada, utiliza el atributo `type="password"`.

Completaré mi formulario. Ahora le pide al visitante que escriba su nombre de usuario y su contraseña:

```
<form method="post" action="tratamiento.php">

  <p>

    <label for="username">Tu nombre de usuario:</label>

    <input type="text" name="username" id="username" />

    <br />

    <label for="pass">Tu contraseña:</label>
```

```

<input type="password" name="pass" id="pass" />

</p>

</form>

```

Verás que los caracteres de la contraseña no se visualizan en pantalla, como se muestra en la siguiente imagen.

Nombre usuario

Contraseña

Un campo de entrada de contraseña

Campo de texto de líneas múltiples

Para crear un campo de texto de líneas múltiples, cambia la etiqueta: vamos a utilizar `<textarea>` `</textarea>`.

Como cualquier otro elemento de formulario, tienes que proporcionarle un nombre mediante `name` y utilizar `label` para explicar qué es.

```

<form method="post" action="tratamiento.php">

  <p>

    <label for="mejora">¿Cómo crees que podría mejorar mi página web?</label><br
  />

    <textarea name="mejora" id="mejora"></textarea>

  </p>

</form>

```

¡Y aquí está la imagen con el resultado (siguiente captura)!

¿Cómo crees que podría mejorar mi página web?

Un campo de entrada pequeño de líneas

múltiples

Como puedes ver, ¡es un poco pequeño! Afortunadamente, puedes cambiar el tamaño de `<textarea>` de dos maneras distintas.

- **En CSS:** tan solo aplica las propiedades CSS `width` y `height` a `<textarea>`.
- **Con atributos:** puedes añadir los atributos `rows` y `cols` a la etiqueta `<textarea>`. El primero especifica el número de líneas de texto que pueden mostrarse simultáneamente y el segundo, el número de columnas.

¿Por qué abres la etiqueta `<textarea>` y la cierras de nuevo inmediatamente después?

Puedes prerrellenar `<textarea>` con un valor por defecto. En este caso, no utilizamos el atributo `value`: simplemente escribimos el texto por defecto entre la etiqueta de apertura y la de cierre.

```
<form method="post" action="tratamiento.php">

  <p>

    <label for="mejora">

      ¿Cómo crees que podría mejorar mi página web?

    </label>

    <br />

    <textarea name="mejora" id="mejora" rows="10" cols="50">

      ¡¿Mejorar tu página web?!

      ¡Venga! ¡Es tan perfecta que no necesita ninguna mejora!

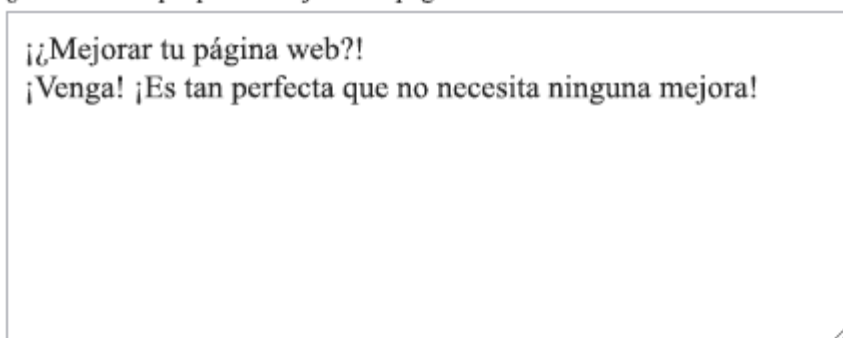
    </textarea>

  </p>

</form>
```

La siguiente imagen muestra el resultado.

¿Cómo crees que podría mejorar mi página web?

A screenshot of a web form with a text area containing two lines of feedback text: "¿Mejorar tu página web?!" and "¡Venga! ¡Es tan perfecta que no necesita ninguna mejora!". The text area is a simple rectangular box with a thin border.

Un campo de

entrada de líneas múltiples completado previamente

Campos de entrada mejorados

HTML5 incluye muchas características nuevas en los formularios. De hecho, han afluado nuevos tipos de campos con esta versión. Tan solo dale al atributo `type` de la etiqueta `<input/>` uno de los nuevos valores disponibles. ¡Echemos un vistazo rápido!

Todos los navegadores todavía no reconocen estos campos entrada mejorados. En su lugar, los navegadores antiguos mostrarán un campo de texto simple de una sola línea (como si hubieras escrito `type="text"`). Que quede entre nosotros, no pasa nada: los nuevos navegadores pueden disfrutar de las últimas funciones, mientras que los navegadores antiguos muestran un campo de texto de sustitución que encaja igual de bien.

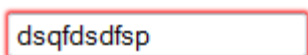
¡Así que te recomendamos usar estos nuevos campos de entrada desde ahora! En el mejor de los casos, tus visitantes se beneficiarán de estas nuevas funciones y, en el peor, no verán ningún problema.

[Correo electrónico](#)

Puedes pedir a los visitantes que introduzcan una dirección de correo electrónico:

```
<input type="email" ></code>
```

El campo *normalmente* te parecerá el mismo, pero el navegador ahora sabe que el usuario tiene que escribir una dirección de correo electrónico. Puede dar una indicación si la dirección no es un correo electrónico, que es lo que hace Firefox (ejemplo a continuación).

A screenshot of a web form showing an email address 'dsqfdfsdfsp' with a red border. The text is inside a rectangular box.

Un correo electrónico escrito incorrectamente aparece enmarcado en rojo en Firefox

Ten en cuenta que algunos navegadores, como, por ejemplo, los navegadores móviles de iPhone y Android, muestran un teclado diseñado para escribir direcciones de correo electrónico (debajo).



Teclado para correo electrónico de

iPhone

Una URL

Con el tipo `url`, puedes pedirle al visitante que escriba una dirección absoluta (normalmente comienza con `http://`):

```
<input type="url" ></code>
```

Mismo principio: si el campo no parece diferente en tu ordenador, ten en cuenta que ha entendido eficazmente que se supone que el visitante debe escribir una dirección. Por ejemplo, los navegadores de móviles muestran un teclado destinado a escribir una URL (imagen siguiente).



Teclado para URL de iPhone

Número de teléfono

Este campo se usa para escribir números de teléfono.

```
<input type="tel" ></code>
```

En el iPhone, por ejemplo, se visualiza un teclado para esta función cuando el visitante tiene que completar este campo (imagen a continuación).



Teclado para números de teléfono de

iPhone

Número

Este campo se utiliza para escribir un número entero:

```
<input type="number" ></code>
```

El campo normalmente se muestra con unas pequeñas flechas para cambiar el valor (imagen siguiente).



Campo de entrada de números

Puedes personalizar la manera en la que funciona el campo con los siguientes atributos:

- `min`: valor mínimo permitido.
- `max`: valor máximo permitido.
- `step`: es el desplazamiento. Si especificas un «step» de 2, el campo solo aceptará valores de 2 en 2 (por ejemplo: 0, 2, 4, 6, etc.).

Un cursor

El tipo `range` se utiliza para seleccionar un número con un cursor (también llamado un *slider*), como se puede ver en el siguiente ejemplo:

```
<input type="range" ></code>
```



Un cursor del tipo range

Puedes utilizar de nuevo los atributos `min`, `max` y `step` para limitar los valores disponibles.

Color

Este campo te permite introducir un color:

```
<input type="color" ></code>
```

En la práctica, los navegadores no lo utilizan demasiado en la actualidad. No te sorprendas si solo ves un campo de texto normal.

Fecha

Hay varios tipos de campos de selección de fecha:

- `date`: para la fecha (05/08/1985, por ejemplo);
- `time`: para la hora (13:37, por ejemplo);
- `week`: para la semana;
- `month`: para el mes;
- `datetime`: para la fecha y la hora (con administración de zona horaria);
- `datetime-local` para la fecha y la hora (sin administración de zona horaria).

Ejemplo:

```
<input type="date" ></code>
```

Como puedes ver, ¡hay muchas opciones!

Por el momento, pocos navegadores soportan este tipo de campo, excepto Opera.

Buscar

Así puedes crear un campo de búsqueda como este:

```
<input type="search" ></code>
```

Entonces, el navegador decide cómo muestra el campo de búsqueda. Por consiguiente, puede añadir una pequeña lupa que indica que es un campo de búsqueda y posiblemente para almacenar las últimas búsquedas realizadas por el visitante.

Opciones

HTML incluye un montón de opciones que pueden usarse en el formulario. Son elementos que requieren que el visitante elija a partir de una lista de posibilidades. Vamos a repasar:

- casillas de verificación;
- campos de opciones;
- listas desplegables.

Casillas de verificación

¿Cómo se crea una casilla de verificación? ¡No podría ser más fácil! Vamos a utilizar la etiqueta `<input />`, pero esta vez especificando el tipo `checkbox`:

```
<input type="checkbox" name="opciones" ></code>
```

¡Añade `<label>` en la posición correcta, ¡y ya lo tienes!

```
<form method="post" action="tratamiento.php">

  <p>

    Marca las comidas que te gustan:<br />

    <input type="checkbox" name="patatas fritas" id="patatas fritas" /> <label
for="patatas fritas">Patatas fritas</label><br />

    <input type="checkbox" name="hamburguesa" id="hamburguesa" /> <label
for="hamburguesa">Hamburguesa</label><br />

    <input type="checkbox" name="espinacas" id="espinacas" /> <label
for="espinacas">Espinacas</label><br />

    <input type="checkbox" name="ostras" id="ostras" /> <label
for="ostras">Ostras</label>

  </p>

</form>
```

La siguiente imagen muestra el resultado.

Marca las comidas que te gustan:

- ☐ Patatas Fritas
- ☐ Hamburguesa
- ☐ Espinacas
- ☐ Ostras

Casillas de verificación

No olvides dar un nombre diferente a cada casilla de verificación. Más tarde, te permitirá identificar las que ha seleccionado el visitante.

Por último, recuerda que puedes tener una casilla marcada por defecto con el atributo `checked`:

```
<input type="checkbox" name="opciones" checked ></code>
```

Todos los atributos normalmente tienen un valor. Sin embargo, en ese caso, añadir un valor no es necesario: la presencia del atributo es suficiente para hacer que el ordenador entienda que la casilla tiene que estar marcada.

Si te molesta, recuerda que puedes asignar cualquier valor al atributo (algunos administradores de páginas algunas veces escriben `checked="checked"`, pero es un poco redundante). La casilla se marcará sin orden.

[Campos de opciones.](#)

Los campos de opciones te permiten elegir una (y solamente una) opción de una lista de posibilidades. De alguna manera, parecen casillas de verificación pero hay una ligera dificultad adicional: tienen que estar organizadas como grupos. Aunque las opciones en el mismo grupo tienen el mismo (`nombre`), cada opción tienen que tener un (`valor`) diferente.

La etiqueta que se utiliza siempre es `<input />`, esta vez con el valor `radio` en el atributo `type`.

El siguiente ejemplo lo aclarará:

```
<form method="post" action="tratamiento.php">

  <p>

    Indica el grupo de edad al que perteneces:<br />

    <input type="radio" name="edad" value="menos15" id="menos15" /> <label
for="menos15">Menos de 15 años</label><br />

    <input type="radio" name="edad" value="entre15-25" id="entre15-25" /> <label
for="entre15-25">Entre 15 y 25 años</label><br />

    <input type="radio" name="edad" value="entre25-40" id="entre25-40" /> <label
for="entre25-40">Entre 25 y 40 años</label><br />

    <input type="radio" name="edad" value="mas40" id="mas40" /> <label
for="mas40">Incluso mayor que eso?</label>

  </p>

</form>
```

Lo que nos da el resultado de la siguiente imagen.

Indica el grupo de edad al que perteneces:

- ☐ Menos de 15 años
- ☐ Entre 15 y 25 años
- ☐ Entre 25 y 40 años
- ☐ Incluso mayor que eso?

Botones de radio

¿Por qué pusiste el mismo nombre para cada opción? Solo para que el navegador sepa a qué «grupo» pertenece el botón.

Intenta eliminar los atributos `name` y verás que entonces puedes seleccionar todas las opciones. Sin embargo, no es lo que queremos, por esa razón los «enlazamos» al darles el mismo nombre.

Te darás cuenta de que esta vez hemos elegido una `id` diferente al `name`. De hecho, dado que los valores de `name` son los mismos, no podríamos haber diferenciado las `ids` (¡y sabes muy bien que una `id` tiene que ser única!). Por esta razón hemos elegido dar a `id` el mismo valor que a `value`.

Si tienes dos campos de opciones diferentes, tienes que dar un `name` único a cada grupo, de esta manera:

```
<form method="post" action="tratamiento.php">

  <p>

    Indica el grupo de edad al que perteneces:<br />

    <input type="radio" name="edad" value="menos15" id="menos15" /> <label
for="menos15">Menos de 15 años</label><br />

    <input type="radio" name="edad" value="entre15-25" id="entre15-25" /> <label
for="entre15-25">Entre 15 y 25 años</label><br />

    <input type="radio" name="edad" value="entre25-40" id="entre25-40" /> <label
for="entre25-40">Entre 25 y 40 años</label><br />

    <input type="radio" name="edad" value="mas40" id="mas40" /> <label
for="mas40">¿Incluso mayor?</label>

  </p>

  <p>

    ¿En qué continente vives?<br />

    <input type="radio" name="continente" value="europa" id="europa" /> <label
for="europa">Europa</label><br />
```

```

        <input type="radio" name="continente" value="africa" id="africa" /> <label
for="africa">África</label><br />

        <input type="radio" name="continente" value="asia" id="asia" /> <label
for="asia">Asia</label><br />

        <input type="radio" name="continente" value="america" id="america" /> <label
for="america">América</label><br />

        <input type="radio" name="continente" value="australia" id="australia" />
<label for="australia">Australia</label>

    </p>

</form>

```

El atributo `checked` está de nuevo disponible para seleccionar un valor por defecto.

Listas desplegables

Las listas desplegables son otra manera elegante de seleccionar una opción a partir de varias posibilidades. Funcionan de una manera ligeramente distinta. Vamos a utilizar la etiqueta `<select> </select>`, que especifica el comienzo y el final de la lista desplegable. Añadimos el atributo `name` a la etiqueta para darle un nombre a la lista.

Después, dentro de la etiqueta `<select> </select>`, vamos a insertar varias etiquetas `<option> </option>` (una para cada opción posible). Añadimos un atributo `value` a cada uno de ellas para identificar lo que el visitante ha elegido.

Aquí podéis ver un ejemplo de uso:

```

<form method="post" action="tratamiento.php">

    <p>

        <label for="pais">¿En qué país vives?</label><br />

        <select name="pais" id="pais">

            <option value="estadosunidos">Estados Unidos</option>

            <option value="canada">Canadá</option>

            <option value="reinounido">Reino Unido</option>

            <option value="francia">Francia</option>

            <option value="espana">España</option>

            <option value="italia">Italia</option>

```

```

        <option value="china">China</option>

        <option value="japon">Japón</option>

    </select>

</p>

</form>

```

El resultado se muestra en la siguiente imagen.



Una lista desplegable

Si quieres que una opción esté seleccionada por defecto, esta vez utiliza el atributo `selected`:

```

<option value="canada" selected>Canadá</option>

```

También puedes agrupar las opciones con la etiqueta `<optgroup>` `</optgroup>`. En nuestro ejemplo, ¿por qué no separas países según su continente?

```

<form method="post" action="tratamiento.php">

    <p>

        <label for="pais">¿En qué país vives?</label><br />

        <select name="pais" id="pais">

            <optgroup label="Europa">

                <option value="reinounido">Reino Unido</option>

                <option value="francia">Francia</option>

                <option value="espana">España</option>

```

```

        <option value="italia">Italia</option>

    </optgroup>

    <optgroup label="América">

        <option value="estadosunidos">Estados Unidos</option>

        <option value="canada">Canadá</option>

    </optgroup>

    <optgroup label="Asia">

        <option value="china">China</option>

        <option value="japon">Japón</option>

    </optgroup>

</select>

</p>

</form>

```

El resultado se muestra en la siguiente imagen.

¿En qué país vives?

Europe
 United Kingdom
 France
 Spain
 Italy
America
 United States
 Canada
Asia
 China
 Japan

Opciones agrupadas por continente

Los grupos no pueden seleccionarse. Así que no podemos elegir «Europa» en nuestro ejemplo: solo los nombres de países pueden seleccionarse.

Finalizar y enviar el formulario

Casi hemos terminado. Solo nos queda decorar nuestro formulario con algunas funciones finales (por ejemplo, confirmación), y entonces añadir el botón de envío del formulario.

Agrupar campos

Si el formulario ha crecido y tiene un montón de campos, puede que sea útil agruparlos entre varias etiquetas `<fieldset>`. Cada `<fieldset>` puede contener un título con la etiqueta `<legend>`. Veamos este ejemplo:

```
<form method="post" action="tratamiento.php">

  <fieldset>

    <legend>Tus datos de contacto</legend> <!-- Título del conjunto de campos -->

    <label for="apellidos">¿Cuáles son tus apellidos?</label></br>

    <input type="text" name="apellido" id="apellido" /></br>

    <label for="nombre">¿Cuál es tu nombre?</label></br>

    <input type="text" name="nombre" id="nombre" /></br>

    <label for="email">¿Cuál es tu dirección de correo electrónico?</label></br>

    <input type="email" name="email" id="email" /></br>

  </fieldset>

  <fieldset>

    <legend>Tu deseo</legend> <!-- Título del conjunto de campos -->
```

```

<p>

    Pide un deseo que quieres que se haga realidad:</br>

    <input type="radio" name="deseo" value="rico" id="rico" /> <label
for="rico">Ser rico</label></br>

    <input type="radio" name="deseo" value="famoso" id="famoso" /> <label
for="famoso">Ser famoso</label></br>

    <input type="radio" name="deseo" value="inteligente" id="inteligente" />
<label for="inteligente">Ser <strong>aún más</strong> inteligente</label></br>

    <input type="radio" name="deseo" value="otro" id="otro" /> <label
for="otro">Otro...</label></br>

</p>

<p>

    <label for="specs">Si has elegido «Otro», especifica cuál:</label></br>

    <textarea name="specs" id="specs" cols="40" rows="4"></textarea>

</p>

</fieldset>

</form>

```

El resultado se muestra en la siguiente imagen.

Tus datos de contacto

¿Cuales son tus apellidos?

¿Cual es tu nombre?

¿Cual es tu correo electrónico?

Tu deseo

Pide un deseo que quieres que se haga realidad:

☐ Ser rico

☐ Ser famoso

Los campos están agrupados

Seleccionar un campo automáticamente

Puedes situar el cursor automáticamente en uno de los campos del formulario con el atributo `autofocus`. Una vez que el visitante cargue la página, el cursor aparece en este campo.

Por ejemplo, para situar el cursor por defecto en el campo `name`:

```
<input type="text" name="nombre" id="nombre" autofocus ></code>
```

Hacer un campo obligatorio

Puedes hacer un campo obligatorio al asignarle el atributo `required`.

```
<input type="text" name="nombre" id="nombre" required ></code>
```

Entonces el navegador le indicará al usuario, si el campo está vacío cuando el formulario sea enviado, que necesita completarlo.

Los navegadores antiguos que no reconocen este atributo envían el contenido del formulario sin revisar. En estos navegadores, necesitarás añadir pruebas, por ejemplo, con scripts de JavaScript.

CSS incluye seudoformatos para cambiar el estilo de los elementos que son `(:required)` y `(:invalid)`. No olvides que tienes el seudoformato `:focus` para cambiar la apariencia de un control cuando el cursor está en él.

```
:required
```

```
{
```

```
background-color: red;
}
```

El botón de envío

Así que ahora lo que nos queda por hacer es crear el botón de envío. De nuevo, la etiqueta `<input />` viene a nuestro rescate. Hay cuatro versiones:

- `type="submit"`: el botón de envío principal del formulario. Este es el que usarás más a menudo. El usuario será redireccionado a la página especificada en el atributo `action` del formulario.
- `type="reset"`: restablece el formulario
- `type="image"`: equivalente al botón `submit`, pero esta vez mostrado como una imagen. Añade el atributo `src` para especificar la URL de la imagen.
- `type="button"`: botón genérico, que no tendrá efecto alguno (por defecto). En general, este botón se maneja en JavaScript para desarrollar acciones en la página. No lo utilizaremos aquí.

Puedes cambiar el texto mostrado dentro de los botones con el atributo `value`.

Para crear un botón de envío, entonces escribimos, por ejemplo:

```
<input type="submit" value="Enviar" ></code>
```

Lo que nos da el resultado de la siguiente imagen.



Un botón de envío

Cuando haces clic en el botón «Enviar», entonces el formulario te lleva a la página especificada en el atributo `action`. Recuerda, imaginamos una página ficticia: `tratamiento.php`.

El problema es que no puedes crear esta página exclusivamente en HTML. Tienes que aprender un nuevo lenguaje, como PHP, para ser capaz de «recuperar» la información introducida y decidir lo que hacer con ella. Sin ir más lejos, ¡también estoy escribiendo un curso sobre lenguaje PHP para cualquiera que esté interesado! El curso se publicará dentro de poco.

Algunas páginas web proporcionan servicios llamados «Mailform», que te envían un correo electrónico cuando uno de los visitantes ha completado el formulario. Aunque esto elimina la necesidad de aprender un nuevo lenguaje, no es muy conveniente: o bien tienes que pagar, o bien contribuir con publicidad... y no tienes muchas opciones para personalizar el procesamiento de datos.

En resumen

- Un formulario es una zona interactiva en la página, donde los visitantes pueden introducir información.
- Un formulario está encasillado con la etiqueta `<form>` a la que deben añadirse dos atributos: `method` (método de envío de datos) y `action` (página a la que el visitante se redirige después de enviar el formulario y que procesa la información).
- Muchos de los elementos del formulario pueden insertarse con la etiqueta `<input />`. El valor de su atributo `type` especifica el tipo de campo que va a insertarse:
 - `text`: campo de texto;
 - `password`: campo de texto para contraseña;
 - `tel`: número de teléfono;
 - `checkbox`: casilla de verificación;
 - ...
- La etiqueta `<label>` se utiliza para introducir una etiqueta. Está asociada con un campo de formulario mediante el atributo `for`, que debe tener el mismo valor que el campo de formulario `id`.
- Puedes hacer un campo obligatorio con el atributo `required`, asegúrate de que está seleccionado por defecto usando `autofocus`, y especifica un indicador en el campo mediante `placeholder`...
- Para recuperar lo que los visitantes han escrito, no es suficiente con el lenguaje HTML. Tienes que utilizar un lenguaje de «servidor», como el PHP... Si quieres seguir, ¡vas a tener que aprender un nuevo lenguaje!

Vídeo y audio

Desde la llegada de YouTube y Dailymotion, ahora ver vídeos en páginas web se ha convertido en algo común. Hay que decir que la llegada de la banda ancha ha ayudado a democratizar los vídeos en Internet.

Hasta ahora, sin embargo, no se proporcionaba ninguna etiqueta HTML para gestionar vídeos. En su lugar, había que utilizar un **complemento** como, por ejemplo, Flash. Incluso a día de hoy, Flash es de lejos la manera más usada para ver vídeos en YouTube, Dailymotion, Vimeo y en otras páginas. Pero usar un complemento tiene varios inconvenientes: te hace dependiente de los que administran el complemento (en este caso, de la empresa Adobe, que es el dueño de Flash), no siempre puedes controlar su funcionamiento, hay fallos de seguridad ocasionales, etc. Al final, es bastante engorroso.

Por esta razón, se crearon dos nuevas etiquetas estándar en HTML5: `<video>` y `<audio>`.

Formatos de vídeo y audio

Cuando hablamos de imágenes y de la etiqueta `<img>`, comenzamos rápidamente a repasar varios formatos de imagen (JPEG, PNG, GIF, etc.) Para vídeo y audio, haremos lo mismo, pero es más complicado.

De hecho, la manera en la que funcionan los vídeos es tan compleja que podría completar un curso entero. Pero aquí estamos hablando sobre HTML, no vamos a pasar las próximas noches explorando la complejidad de la codificación de vídeo. Así que voy a simplificar las cosas y solo voy a decirte lo que necesitas saber.

Formatos de audio

Para reproducir música o cualquier sonido, se pueden utilizar muchos formatos. La mayoría de ellos son comprimidos (como lo son las imágenes en JPEG, PNG y GIF), lo que reduce su tamaño de almacenamiento:

- **MP3**: ¡es imposible que *no* hayas oído hablar de él! Es uno de los más antiguos y también uno de los formatos más compatibles (todos los dispositivos pueden leer MP3), así que aún se utiliza mucho hoy en día.
- **AAC**: lo utiliza principalmente Apple con iTunes, y es un formato de buena calidad. Los iPod, iPhone y iPad pueden leerlo sin problemas.
- **OGG**: el formato Ogg Vorbis se utiliza ampliamente en el mundo del software libre y, en particular, en Linux. El formato tiene la ventaja de ser libre, en otras palabras, de no estar protegido por ninguna patente.
- **WAV (formato sin comprimir)**: este formato debería evitarse cuando sea posible, puesto que los archivos que utiliza son muy pesados. Es básicamente el equivalente a Bitmap (BMP) en audio.

Ningún navegador maneja todos estos formatos al mismo tiempo. La compatibilidad con MP3 y OGG es lo principal a tener en cuenta.

Navegador	MP3	OGG
Internet Explorer	Sí	-
Chrome	Sí	Sí
Firefox	-	Sí
Safari	Sí	-
Opera	-	Sí

¿No hay un formato «ideal» que reconocen todos los navegadores?

¡Pues no! Afortunadamente, podemos proporcionar a los navegadores diferentes formatos y dejarles que seleccionen el que puedan leer.

Formatos de vídeo

El almacenamiento de vídeo es mucho más complejo. Necesitamos tres elementos:

- **Un formato contenedor:** de alguna manera es como una caja que se utiliza para contener los dos elementos siguientes. El tipo de contenedor normalmente se puede reconocer por la extensión del archivo. AVI, MP4, MKV, etc.
- **Un códec de audio:** es el formato de sonido del vídeo, que normalmente está comprimido. Los hemos visto y vamos a utilizar los mismos: MP3, AAC, OGG, etc.
- **Un códec de vídeo:** es el formato que comprime las imágenes. Aquí es donde las cosas se complican, dado que estos formatos son complejos y su uso no siempre es libre. Los principales que hay que conocer para Internet son:
 - **H.264:** ahora es uno de los formatos más potentes y más utilizados... pero no es 100 % libre. De hecho, puede usarse de forma libre en algunos casos (como en streaming de vídeo en una página web personal), pero la incertidumbre legal lo hace arriesgado de utilizar a gran escala.
 - **Ogg Theora:** un códec que es libre y no está protegido por copyright, pero menos potente que H.264. Aunque es bastante conocido en Linux, tienes que instalar software para reproducirlo en Windows.
 - **WebM:** otro códec más reciente y libre, que no está protegido por copyright. Provisto por Google, ahora es el principal competidor de H.264.

¿Qué códec de vídeo son compatibles con qué navegadores? Una vez más, verás que es un poco lío.

Navegador	H.264	Ogg Theora	WebM
Internet Explorer	Sí	-	Sí
Chrome	-	Sí	Sí
Firefox	-	Sí	Sí
Safari	Sí	-	-
Opera	Sí	Sí	Sí

De nuevo, ningún formato realmente destaca. Es recomendable proporcionar cada vídeo en varios formatos para que se pueda reproducir en tantos navegadores como sea posible.

Para convertir vídeo en estos formatos, recomiendo este excelente programa gratuito, [Miro Video Converter](#), que puedes descargar a través de la dirección web haciendo clic en el nombre.

Tan solo arrastra y suelta tu vídeo en la ventana del programa y selecciona el formato de salida deseado. ¡Te permitirá crear varias versiones del vídeo!

Insertar un elemento de audio

La etiqueta `<audio>` que vamos a descubrir la reconocen todos los navegadores nuevos, incluido Internet Explorer desde la versión 9 (IE9) en adelante.

En teoría, solo necesitas una etiqueta para reproducir un sonido en tu página:

```
<audio src="cancion.mp3"></audio>
```

En la práctica, es un poco más complicado.

Si pruebas este código... ¡no verás nada! De hecho, el navegador solo descargará información general del archivo (llamados **metadatos**), pero no ocurrirá nada en particular.

Puedes añadir los siguientes atributos a la etiqueta:

- **controls**: para añadir los botones de «Reproducir» y «Pausar» y la barra de desplazamiento. Puede parecer fundamental, y puede que te preguntes por qué no lo incluyen por defecto, pero algunas páginas web prefieren crear sus propios botones y controlar la reproducción con JavaScript.
- **width**: para cambiar el ancho de la herramienta de reproducción de audio.
- **loop**: la música se reproducirá en bucle.
- **autoplay**: la música se reproducirá cuando la página esté cargada. No te excedas con esto. ¡Una página web que reproduce música por sí misma cuando está cargada normalmente es irritante!
- **preload**: especifica si la música puede precargarse cuando la página esté cargada o no. Este atributo puede llevar los siguientes valores:
 - **auto** (por defecto): el navegador decide si cargar previamente toda la música, solo los metadatos o nada.
 - **metadata**: solo carga los metadatos (duración, etc.).
 - **none**: no hay carga previa. Útil si no quieres malgastar ancho de banda de tu página web.

No puedes forzar la carga previa de música, siempre es el navegador el que decide. Los navegadores para móviles, por ejemplo, nunca cargan música previamente para ahorrar ancho de banda (dado que el tiempo de carga es largo en un teléfono móvil).

Estará mejor con controles, así que ¡vamos a añadirlos!

```
<audio src="hype_home.mp3" controls></audio>
```

La apariencia del reproductor de audio varía de un navegador a otro. La siguiente imagen muestra, por ejemplo, el reproductor de audio de Google Chrome



El reproductor de audio de Google

Chrome

¿Por qué abrir la etiqueta si vas a cerrarla inmediatamente después?

Te permite visualizar un mensaje o proporcionar una solución alternativa para los navegadores que no soportan esta nueva etiqueta. Por ejemplo:

```
<audio src="hype_home.mp3" controls>¡Actualiza tu navegador!</audio>
```

Los que tengan un navegador reciente no verán este mensaje. Sin embargo, los navegadores antiguos que no entienden la etiqueta mostrarán el texto.

Recomiendo que incluyas un solución alternativa a Flash, por ejemplo, [Drewplayer](#). Sitúas el código para Flash entre las etiquetas `<audio>` y `</audio>`: por tanto, los navegadores antiguos mostrarán el reproductor de Flash mientras que los nuevos mostrarán el reproductor nativo.

Hemos visto que algunos navegadores no soportan MP3. ¿Cuál es la solución?

Tienes que incluir varias versiones del archivo de audio. En este caso, vamos a hacer nuestra etiqueta de esta manera:

```
<audio controls>

  <source src="hype_home.mp3"></source>

  <source src="hype_home.ogg"></source>

</audio>
```

El navegador tomará automáticamente el formato que reconozca.

Insertar un vídeo

La etiqueta `<video>` que vamos a descubrir la reconocen todos los navegadores, incluyendo Internet Explorer desde la versión 9 (IE9).

Solamente necesitas una única etiqueta `<video>` para insertar un vídeo en la página:

```
<video src="sintel.webm"></video>
```

Pero puede que te sientas de nuevo decepcionado si solo utilizas este código. ¡No hay controles para empezar el vídeo!

Vamos a añadir algunos atributos (la mayoría de ellos son los mismos que para la etiqueta `<audio>`):

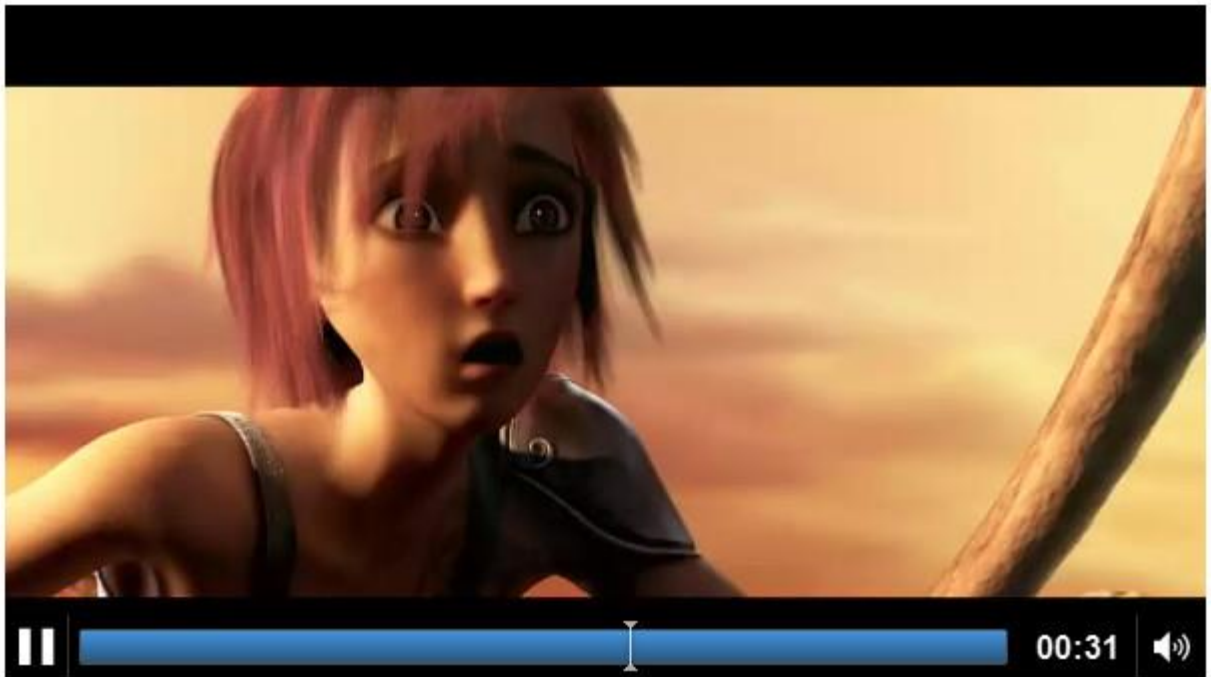
- **poster:** se muestra otra imagen en lugar de la del vídeo hasta que se reproduzca. Por defecto, el navegador toma el primer fotograma del vídeo, pero como a menudo es un fotograma negro o un fotograma que no representa el vídeo, te recomiendo crear uno. Puedes hacer simplemente una captura de imagen de cualquier momento del vídeo.
- **controls:** para añadir los botones de «Reproducir» y «Pausar» y la barra de desplazamiento. Puede parecer fundamental, pero algunas páginas web prefieren crear sus propios botones y controlar la reproducción con JavaScript. En nuestro caso, ¡será más que suficiente!
- **width:** para cambiar el ancho del vídeo.
- **height:** para cambiar la altura del vídeo.
- **loop:** el vídeo se reproducirá en bucle.
- **autoplay:** el vídeo se reproducirá cuando cargue la página. Una vez más, no te excedas con esto. ¡Una página web que ejecuta algo por su cuenta cuando carga es, por lo general, irritante!
- **preload:** especifica si el vídeo puede precargarse cuando la página cargue o no. Este parámetro puede tener los siguientes valores:
 - **auto** (valor por defecto): el navegador decide si precargar el vídeo entero, solo los metadatos o nada.
 - **metadata:** solo carga los metadatos (duración, tamaño, etc.).
 - **none:** no hay carga previa. Útil si quieres evitar malgastar ancho de banda de la página web.

No podemos forzar la carga previa del vídeo, siempre es el navegador el que decide. Las proporciones del vídeo siempre se preservan. Si fijas un ancho y un alto, el navegador se asegurará de que las dimensiones especificadas no se sobrepasen, pero mantendrá las proporciones.

Aquí podéis ver algo de código que está un poco más completo:

```
<video src="sintel.webm" controls poster="sintel.jpg" width="600"></video>
```

La siguiente imagen muestra el resultado.



Un vídeo con las opciones de reproducción y un tamaño definido

¿Por qué abrir y cerrar inmediatamente después de la etiqueta?

La respuesta es la misma que para la etiqueta `<audio>`. Te permite visualizar un mensaje o incluir una solución alternativa (en Flash) si el navegador no reconoce la etiqueta:

```
<video src="sintel.webm" controls poster="sintel.jpg" width="600">
```

¡Es hora de actualizar el navegador!

```
</video>
```

¿Cómo complaces a todos los navegadores si todos ellos reconocen diferentes formatos de vídeos?

Utilizas la etiqueta `<source>` dentro de la etiqueta `<video>` para incluir diferentes formatos. El navegador elegirá el que reconozca.

```
<video controls poster="sintel.jpg" width="600">
```

```
<source src="sintel.mp4" ></source>
```

```
<source src="sintel.webm" ></source>
```

```
<source src="sintel.ogv" ></source>
```

```
</video>
```

Actualmente, los iPhone, iPad y iPod solo reconocen el formato H.264 (archivo .mp4)... ¡y solo si aparece el primero en la lista! Por consiguiente,

recomiendo que especifiques el formato H.264 el primero para garantizar una compatibilidad máxima.

¿Cómo se visualiza el vídeo en pantalla completa?

Por el momento no es posible. De hecho, sí que hay una manera en Firefox, pero está un poco escondida: tienes que hacer clic con el botón derecho en el vídeo y seleccionar «Pantalla completa».

No hay ninguna forma de forzar la pantalla completa, ni siquiera en JavaScript. Es comprensible, dado que algunas páginas web podrían perturbar considerablemente la navegación al mostrar vídeos en pantalla completa sin preguntar antes al usuario si realmente lo quieren.

¿Cómo puedo proteger mi vídeo? ¡No quiero que se pueda copiar fácilmente!

No es posible. Las etiquetas no se diseñaron para limitar o impedir la descarga. Es bastante lógico cuando lo piensas: para que los usuarios puedan ver el vídeo, tendrán que descargarlo de todas formas, de una u otra manera.

Así que no creas que puedes impedir que el vídeo sea descargado mediante esta técnica.

Los reproductores de vídeo Flash te permiten «proteger» el contenido de los vídeos, pero de nuevo, existen soluciones. Muchos *complementos* te permiten descargar vídeos de YouTube, por ejemplo.

En resumen

- Antes no solía ser posible insertar música o vídeo en HTML. Tenías que utilizar un complemento como Flash.
- Desde HTML5, se han introducido las etiquetas `<audio>` y `<video>` y permiten reproducir música y vídeo sin necesidad de complementos.
- Hay varios formatos de audio y vídeo. En particular, deberías conocer:
 - para audio: MP3 y Ogg Vorbis;
 - para vídeo: H.264, Ogg Theora y WebM.
- Ningún formato es reconocido por todos los navegadores: tienes que incluir música o vídeo en diferentes formatos para satisfacer a todos los navegadores.
- Tienes que añadir el atributo `controls` a las etiquetas `<audio>` y `<video>` para permitir que los usuarios empiecen o paren un medio audiovisual.
- Estas etiquetas no se han diseñado para impedir la descarga de música y vídeo. No puedes proteger tus contenidos audiovisuales de ser copiados.

Diseño de página adaptable con consultas multimedia

¿Sabes sobre lo que están más preocupados los administradores que implementan el diseño de su página? La resolución de pantalla del visitante. Por supuesto: el espacio o número de píxeles varía ampliamente de una pantalla a otra.

Esta información es importante cuando creas un diseño: ¿cómo debería mostrarse tu página en resoluciones de pantalla diferentes? Si tienes una pantalla panorámica, puede que te olvides de que algunas personas navegan con pantallas pequeñas. Sin mencionar los navegadores de los teléfonos inteligentes, que son todavía menos anchos.

Aquí es donde aparecen las *consultas multimedia*. Estas son las reglas que deben aplicarse para cambiar el diseño de una página web según las características de la pantalla. Mediante esta técnica, ¡puedes crear un diseño que se ajuste automáticamente a la pantalla de cada usuario!

Implementación de consultas multimedia

Las consultas multimedia son parte de las nuevas características de CSS3. No son nuevas propiedades, sino más bien *reglas* que pueden aplicarse en determinadas condiciones. En concreto, serás capaz de decir: «Si la resolución de pantalla del visitante es menor que tal valor, entonces aplica las siguientes propiedades de CSS». Esto te permite cambiar el aspecto de la página web en determinadas condiciones: Puedes incrementar el tamaño del texto, cambiar el color de fondo, cambiar la posición del menú en determinadas resoluciones, etc.

Al contrario de lo que se podría pensar, las consultas multimedia no se limitan solo a las resoluciones de pantalla. Puedes cambiar el aspecto de la página según otros criterios, como el tipo de pantalla (teléfono inteligente, TV, proyector, etc.), el número de colores, la orientación de la pantalla (vertical u horizontal), etc. ¡Hay un montón de posibilidades!

Las consultas multimedia funcionan en todos los navegadores, incluido Internet Explorer desde la versión 9 (IE9) en adelante.

Aplicar una consulta multimedia

Por lo tanto, las consultas multimedia son reglas que especifican cuándo tienen que aplicarse las propiedades CSS. Hay dos formas de usarlas:

- cargando una hoja de estilo `.css` según la regla (por ejemplo: «Si la resolución es menor que 1280 px de ancho, carga el archivo `resoluciones_peques.css`»);

- escribiendo la regla directamente en el archivo `.css` habitual (por ejemplo: «Si la resolución es menor que 1280 px de ancho, carga las siguientes propiedades CSS»).

Cargar una hoja de estilo diferente

¿Recuerdas la etiqueta `<link />` que, en nuestro código HTML, cargaba un archivo `.css`?

```
<link rel="stylesheet" href="estilo.css" ></code>
```

Puedes añadir un atributo `media`, en el que vamos a escribir la regla que debe aplicarse al archivo que se tiene que cargar. Esto se conoce como hacer una *consulta multimedia*. Por ejemplo:

```
<link rel="stylesheet" media="screen and (max-width: 1280px)"
href="resoluciones_peques.css" ></code>
```

Al final, tu código HTML puede incluir varios archivos CSS: uno por defecto (que se carga en todos los casos) y uno o dos más que solo se cargan si se aplican las reglas correspondientes.

```
<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8" />

    <link rel="stylesheet" href="estilo.css" /> <!-- Para todos -->

    <link rel="stylesheet" media="screen and (max-width: 1280px)"
href="resoluciones_peques.css" /> <!-- Para los que tengan una resolución menor a
1280 px -->

    <title>Consultas multimedia</title>

  </head>
```

Cargar reglas directamente en la hoja de estilo

Otra técnica, que personalmente prefiero por razones prácticas, es escribir estas reglas en el mismo archivo CSS como habitualmente.

En este caso, escribimos la regla en el archivo `css` de la siguiente manera:

```
@media screen and (max-width: 1280px)
{
```

```
/* Escribe aquí tus propiedades en CSS */
```

```
}
```

Las reglas disponibles

Hay muchas reglas para crear consultas multimedia. Tan solo mencionaré aquí las principales:

- `color`: gestión de color (en bits/píxel).
- `height`: muestra la altura del campo (ventana).
- `width`: muestra el ancho del campo (ventana).
- `device-height`: altura del dispositivo.
- `device-width`: ancho del dispositivo.
- `orientation`: orientación del dispositivo (vertical u horizontal).
- `media`: tipo de pantalla de salida. Algunos posibles valores:
 - `screen`: pantalla «normal»;
 - `handheld`: dispositivo móvil;
 - `print`: imprimir;
 - `tv`: televisión;
 - `projection`: proyector;
 - `all`: todo tipo de pantallas.

Se puede añadir el prefijo `min-` o `max-` delante de la mayoría de estas reglas. Por consiguiente, `min-width` quiere decir «ancho mínimo» y `max-height`, «altura máxima», etc.

La diferencia entre `width` y `device-width` se puede observar fundamentalmente en navegadores para teléfonos móviles inteligentes, como veremos más adelante.

Las reglas se pueden combinar con estas palabras:

- `only`: «solo»;
- `and`: «y»;
- `not`: «no»;

Aquí van algunos ejemplos de consultas multimedia para ayudarte a comprender el principio.

```
/* En pantallas con un ancho de ventana máximo de 1280 px */
```

```
@media screen and (max-width: 1280px)
```

```
/* En todo tipo de pantallas con un ancho de ventana entre 1024 px y 1280 px */
```

```
@media all and (min-width: 1024px) and (max-width: 1280px)
```

```
/* En televisiones */
```

```
@media tv
```

```
/* En todas los tipos de pantalla orientadas verticalmente */
```

```
@media all and (orientation: portrait)
```

Los navegadores antiguos, incluidos IE6, IE7 e IE8, no conocen las consultas multimedia pero son capaces de interpretar el comienzo de la regla (pueden leer @media screen, por ejemplo). Por lo tanto, ¡leerán las siguientes propiedades CSS aunque no les afecte la regla! Para evitarlo, un truco es usar la palabra clave `only` que estas antiguas versiones no conocen:

«@media only screen» no provoca un error en los navegadores antiguos.

Probando consultas multimedia

Las consultas multimedia se utilizan principalmente para adaptar el diseño de la página web a diferentes anchos de pantalla.

Hagamos una prueba muy simple: vamos a cambiar el tamaño del texto y el color si la ventana es mayor o menor que 1024 píxeles de ancho. Para esta prueba, voy a usar el segundo método: escribir la regla directamente en el mismo archivo `.css` como de costumbre:

```
/* Párrafos azul por defecto */
```

```
p
```

```
{
```

```
    color: blue;
```

```
}
```

```
/* Nuevas reglas si la ventana es más ancha de 1024px */
```

```
@media screen and (max-width: 1024px)
```

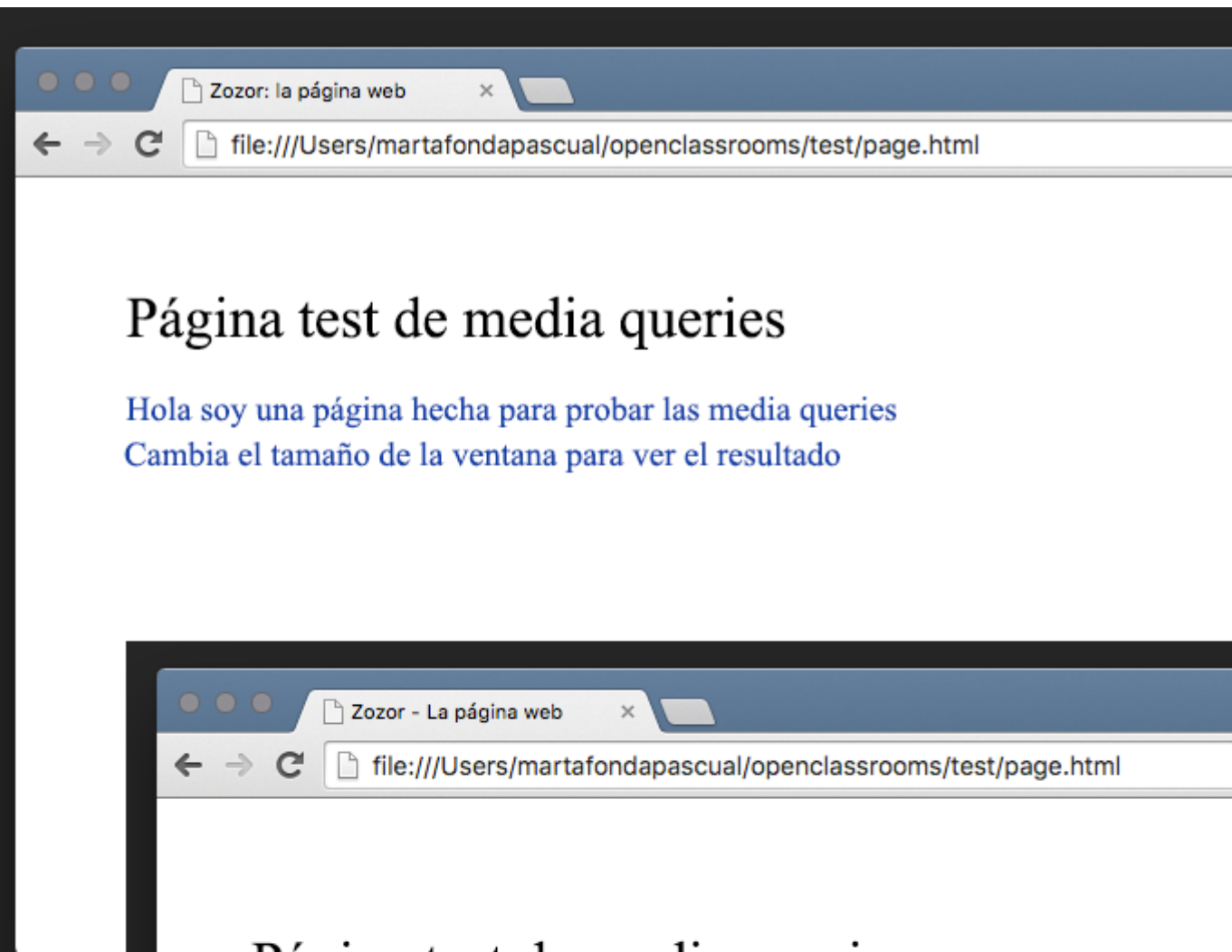
```
{
```

```
    p
```

```
{  
  
  color: red;  
  
  background-color: black;  
  
  font-size: 1.2em;  
  
}  
}
```

En nuestro CSS, primero pedimos que el texto del párrafo se escriba en azul. Hasta aquí nada nuevo. Sin embargo, hemos añadido una media query que se aplica a todas las pantallas cuya anchura no supere los 1024px. En el interior, hemos aplicado las reglas CSS a los párrafos para escribirlos más grandes y en rojo.

Resultado: ¡el aspecto de la página varía según el tamaño de la ventana (figura de abajo)!



El aspecto de la página varía según el tamaño de la ventana

Uso práctico de las media queries en el diseño

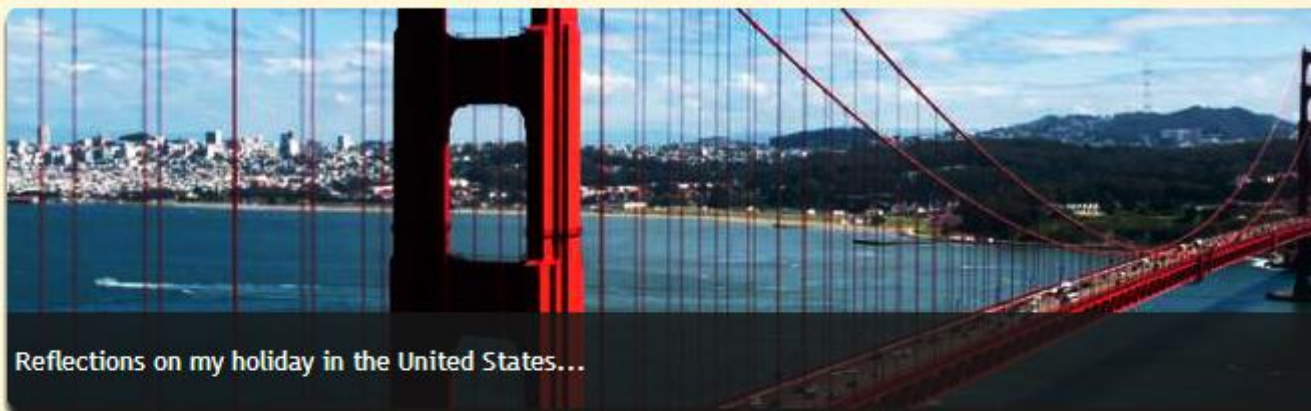
Cambiar el color del texto es agradable, pero en realidad no añade gran cosa. Sin embargo, utilizar media queries para cambiar el aspecto de tu sitio web dependiendo de la resolución de la pantalla resulta en seguida mucho más útil. ¡Verás que puedes hacer lo que quieras!

En este ejemplo, sugiero que vuelvas a utilizar el diseño que creamos para el sitio web de Zozor (figura siguiente).



Travel diaries

HOME BLOG



Reflections on my holiday in the United States...



I'M A GREAT TRAVELLER

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam nec sagittis massa. Nulla facilisi. Cras id arcu lorem, et semper purus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Duis vel enim mi, in lobortis sem. Vestibulum luctus elit eu libero ultrices id fermentum sem sagittis. Nulla imperdiet mauris sed sapien dignissim id aliquam est aliquam. Maecenas non odio ipsum, a elementum nisi. Mauris non erat eu erat placerat convallis. Mauris in pretium urna. Cras laoreet molestie odio, consequat consequat velit commodo eu. Integer vitae lectus ac nunc posuere pellentesque non at eros. Suspendisse non lectus lorem.

Vivamus sed libero nec mauris pulvinar facilisis ut non sem. Quisque mollis ullamcorper diam vel faucibus. Vestibulum sollicitudin facilisis feugiat. Nulla euismod sodales hendrerit. Donec quis orci arcu. Vivamus fermentum magna a erat ullamcorper dignissim pretium nunc aliquam. Aenean pulvinar condimentum enim a dignissim. Vivamus sit amet lectus at ante adipiscing adipiscing eget vitae felis. In at fringilla est. Cras id velit ut magna rutrum commodo. Etiam ut scelerisque purus. Duis risus elit, venenatis vel rutrum in, imperdiet in quam. Sed vestibulum, libero ut bibendum consectetur, eros ipsum ultrices nisl, in rutrum diam augue non tortor. Fusce nec massa et risus dapibus aliquam vitae nec diam.

Phasellus ligula massa, congue ac vulputate non, dignissim at augue. Sed auctor fringilla quam quis porttitor. Praesent vitae dignissim magna. Pellentesque quis sem purus, vel elementum mi. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Maecenas consectetur euismod urna. In hac habitasse platea dictumst. Quisque tincidunt porttitor vestibulum. Ut iaculis, lacus at molestie lacinia, ipsum mi adipiscing ligula, vel mollis sem risus eu lectus. Nunc elit quam, rutrum ut dignissim sit amet, egestas at sem.

MY LAST TWEET

Hee-haw!

12/05 23:12

MY PICTURES



MY FRIENDS

- » Pupi the rab
- » Mr Baobab
- » Kaiwail
- » Perceval.eu

El sitio web creado durante el ejemplo práctico

Aunque el sitio web es adecuado para la mayoría de las resoluciones de pantalla, tienes que "desplazarlo" hacia la derecha para ver la página completa cuando la pantalla sea de menos de 1024 px. El sitio web no es muy cómodo de ver en una pantalla pequeña.

Sugiero que utilices media queries para cambiar el aspecto del sitio web en el caso de resoluciones de menos de 1024 px de ancho. Vamos a hacer los siguientes cambios:

- el menú de navegación de la parte superior derecha se organizará a lo alto en lugar de a lo ancho y los enlaces se escribirán en caracteres más pequeños;
- el banner con el puente de San Francisco (el *Golden Gate*) se borrará porque ocupa mucho espacio y no proporciona demasiada información;
- el bloque "Acerca del autor" `<aside>` se colocará debajo del artículo (en lugar de al lado) y se reorganizará su contenido, (la foto de Zozor estará en posición flotante).

Sin duda podrías hacer muchos otros cambios: cambiar el color, la disposición del pie de página, etc. Pero esto ya es más que suficiente para tener un poco de práctica con las media queries.

Vamos a trabajar directamente en el archivo `style.css` que creamos durante el ejemplo práctico y añadir unas cuantas instrucciones media queries para ajustar el diseño. Te sugiero que descargues los archivos del ejemplo práctico si todavía no lo has hecho.

[Descargar el ejemplo práctico](#)

La página

El ancho de la página está actualmente establecido en 900 px y el contenido está centrado:

```
#bloc_page
{
    width: 900px;
    margin: auto;
}
```

Te sugiero que añadas la siguiente regla de media query tras estas líneas:

```
@media all and (max-width: 1024px)
```

```
{
    #bloc_page
    {
        width: auto;
    }
}
```

Esta regla significa: "si el ancho de la ventana no supera los 1024 px, ejecuta las siguientes reglas CSS para todos los tipos de pantalla".

Las reglas CSS en cuestión son muy sencillas; en realidad, solo hay una regla: el ancho de la página se ajusta automáticamente (en lugar de dar un ancho fijo de 900 píxeles). La página ocupará entonces todo el espacio disponible de la ventana. Esto evita la aparición de barras de desplazamiento horizontal con resoluciones bajas.

`auto` es el valor predeterminado de la propiedad `anchura`. Por defecto, los bloques tienen una anchura automática (ocupan todo el espacio disponible). Este valor "sobrescribe" el valor de 900px que establecimos unas líneas antes, por lo que esto restaura el comportamiento predeterminado del bloque.

El menú de navegación

Queremos que el menú de navegación ocupe menos espacio en pantallas de menos resolución. En lugar de darle un tamaño fijo, vamos a restaurar su tamaño original flexible y automático. Cada elemento del menú se escribe debajo del anterior; por ello, vamos a convertir las viñetas en etiquetas de `block` en lugar de `inline-block`.

Por último, el texto se escribirá más pequeño y se eliminará el borde de la parte inferior de los enlaces cuando se sitúe encima el puntero, ya que es menos adecuado con una disposición de pantalla como esta.

```
@media all and (max-width: 1024px)
{
    nav
    {
        width: auto;
        text-align: right;
    }
}
```

```

}

nav li

{

    display: block;

    padding-left: 4px;

}

nav a

{

    font-size: 1.1em;

}

nav a:hover

{

    border-bottom: 0;

}

}

```

El banner

Para eliminar el banner no hay nada más sencillo: utilizamos la propiedad `display` asignándola con el valor `none`. Si la ventana es demasiado pequeña, puede ser preferible ocultar el banner completamente:

```

@media all and (max-width: 1024px)

{

    #banner_image

```

```
{  
  
    display: none;  
  
}
```

El bloque "Acerca del autor"

En lugar de colocar este bloque a la derecha del artículo, vamos a colocarlo debajo. Esta disposición "de arriba hacia abajo" es más adecuada para pantallas pequeñas.

Reajustaremos ligeramente la posición de los elementos dentro del bloque, en particular, la foto de Zozor se colocará en posición flotante a la derecha.

```
@media all and (max-width: 1024px)
```

```
{  
  
    article, aside  
  
    {  
  
        width: auto;  
  
        display: block;  
  
        margin-bottom: 15px;  
  
    }
```

```
#arrow
```

```
{  
  
    display: none;  
  
}
```

```
#zozor_picture img
```

```
{
```

```
width: 110px;

float: right;

margin-left: 15px;

}

aside p:last-child

{

    text-align: center;

}

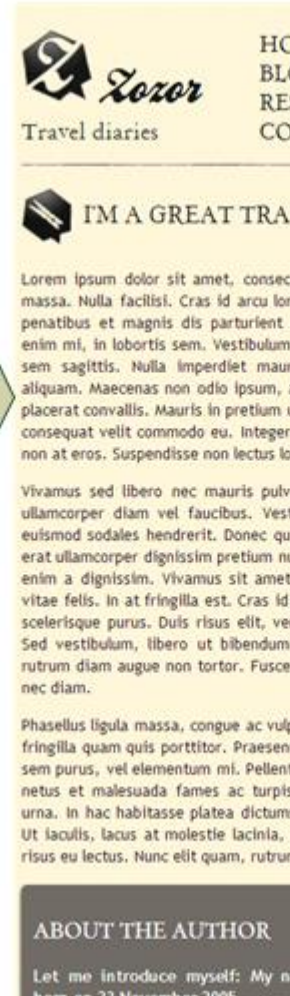
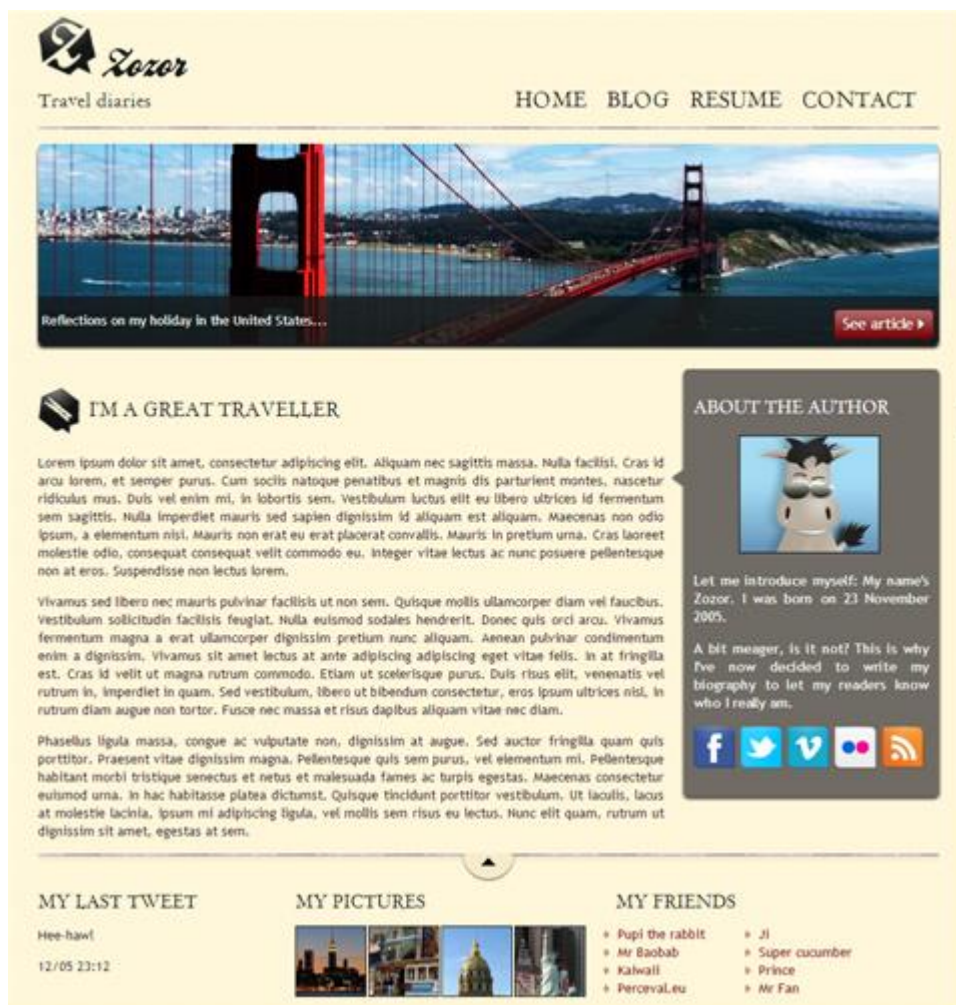
}
```

¿Qué significa `aside p:last-child`?

Es un selector avanzado que aún no hemos usado. `aside p` quiere decir "Todos los párrafos dentro de la etiqueta `<aside>`". `:last-child` afecta solo al último párrafo del bloque `aside` (que contiene los enlaces a Facebook y Twitter), para centrar las imágenes. Evidentemente, podrías haber asignado un `class` o un `id` a este párrafo para abordarlo directamente, pero no quería cambiar el código HTML.

El resultado

Ahora la página se reorganizará totalmente cuando la ventana sea de 1024 px de ancho o menos. Puedes ver los resultados en la siguiente figura, ¡que habla por sí misma!



El mismo sitio web, mostrado de manera diferente según la anchura de la pantalla

Media queries y navegadores móviles

Como probablemente sabes, las pantallas de los smartphone son mucho más estrechas que las pantallas de nuestros ordenadores habituales (solo tienen unos cientos de píxeles de ancho). Para adaptarse a esto, los navegadores móviles muestran la página web "ampliándola" para poder ver toda la página. El área de visualización simulada se llama **viewport**, que es el ancho de la ventana del navegador del teléfono móvil.

Con media queries, si en CSS aplicas a la pantalla de un móvil `max-width`, comparará el ancho que especifiques con el de su viewport. ¡El problema es que el viewport cambia según el navegador móvil utilizado!

Navegador	Ancho predeterminado del viewport
Opera Mobile	850 píxeles
iPhone Safari	980 píxeles

Navegador	Ancho predeterminado del <i>viewport</i>
Android	800 píxeles

Windows Phone 7 1024 píxeles

Un iPhone se comporta como si la ventana tuviera 980 px de ancho, mientras que un Android se comporta como si la ventana fuera de 800 px de ancho.

Para los smartphones, puede ser más conveniente, en lugar de usar `max-width`, utilizar `max-device-width`: este es el ancho del dispositivo. Los dispositivos móviles no tienen más de 480 px de ancho, por lo que podemos dirigirnos solo a los navegadores móviles con esta media query:

```
@media all and (max-device-width: 480px)

{

    /* Tus reglas CSS para tus teléfonos móviles aquí */

}
```

¿Por qué no aplicar en los teléfonos móviles la norma `handheld`?

Veo que me sigues, ¡muy bien! Efectivamente, en teoría podemos aplicar en los teléfonos móviles la regla media `handheld`... Por desgracia, ningún navegador móvil excepto Opera reconoce `handheld`. Todos se comportan como si tuvieran pantallas normales (`screen`). Así que no puedes utilizar `handheld` en los teléfonos móviles.

Puedes cambiar el ancho del viewport del navegador móvil con una etiqueta `meta` que ha de insertarse en el encabezado del documento (`<head>`):

```
<meta name="viewport" content="width=320" ></code>
```

Puedes usar esta etiqueta para cambiar la manera en que se organiza el contenido de tu página en los teléfonos móviles. Para obtener una imagen legible sin zoom, puedes ajustar el viewport a la misma anchura que la pantalla:

```
<meta name="viewport" content="width=device-width" ></code>
```

En resumen

- Las media queries permiten cargar varios estilos CSS basados en ciertos ajustes.
- Hay un gran número de ajustes que permiten las media queries: número de colores, resolución de pantalla, orientación, etc. En la práctica, se utilizan sobre todo para cambiar el aspecto de la página web en diferentes resoluciones de pantalla.

- Creas una media query con la directiva `@media` seguida del tipo de pantalla y una o más condiciones (como el ancho máximo de pantalla). El siguiente estilo CSS se activará solo si se cumplen las condiciones.
- Los navegadores móviles simulan un ancho de pantalla que se llama el *viewport*.
- Con los smartphones puede aplicarse una regla basada en el número real de píxeles que se muestran en la pantalla: `max-device-width`.

Hay que seguir

Conforme este curso va llegando a su fin, podría ser tentador pensar que lo hemos visto todo. ¿Todo? En serio no pensarás eso, ¿verdad? Venga, tienes todavía cientos de cosas que descubrir, ya sea en HTML, CSS o en tecnologías relacionadas (PHP, JavaScript, etc.).

Este capítulo tiene como objetivo darte algunos consejos para completar tu aprendizaje. Así que no estés triste, ¡aún hay un montón de cosas que descubrir!

Desde los sitios web a las aplicaciones web (JavaScript, AJAX, etc.)

JavaScript es un lenguaje que existe desde hace algunos años y se utiliza con frecuencia en la creación web, además de HTML y CSS. Es probablemente uno de los primeros lenguajes que hay que aprender ahora que sabes algo acerca de HTML y CSS.

¿Para qué se usa JavaScript? ¿No podemos hacer todo con HTML y CSS?

Ya puedes hacer muchas cosas en HTML y CSS, pero si quieres hacer tu página más interactiva, un lenguaje como JavaScript es esencial.

Estos son algunos ejemplos de para qué se puede utilizar JavaScript:

- A menudo la utilizaremos para cambiar las propiedades CSS sin tener que recargar la página. Por ejemplo, señalas una imagen y el fondo de tu sitio web cambia de color (no puedes hacer esto con `:hover` puesto que implica dos etiquetas distintas. Es una limitación de CSS).
- También se puede utilizar para editar el código fuente HTML sin tener que recargar la página *mientras* el visitante está viendo la página.
- También te permite mostrar cuadros de diálogo en la pantalla del visitante ...
- ... o cambiar el tamaño de la ventana.

JavaScript es un lenguaje que está más cerca de lenguajes de programación como C, C ++, Python, Ruby, etc. HTML y CSS, sin embargo, son más lenguajes de descripción:

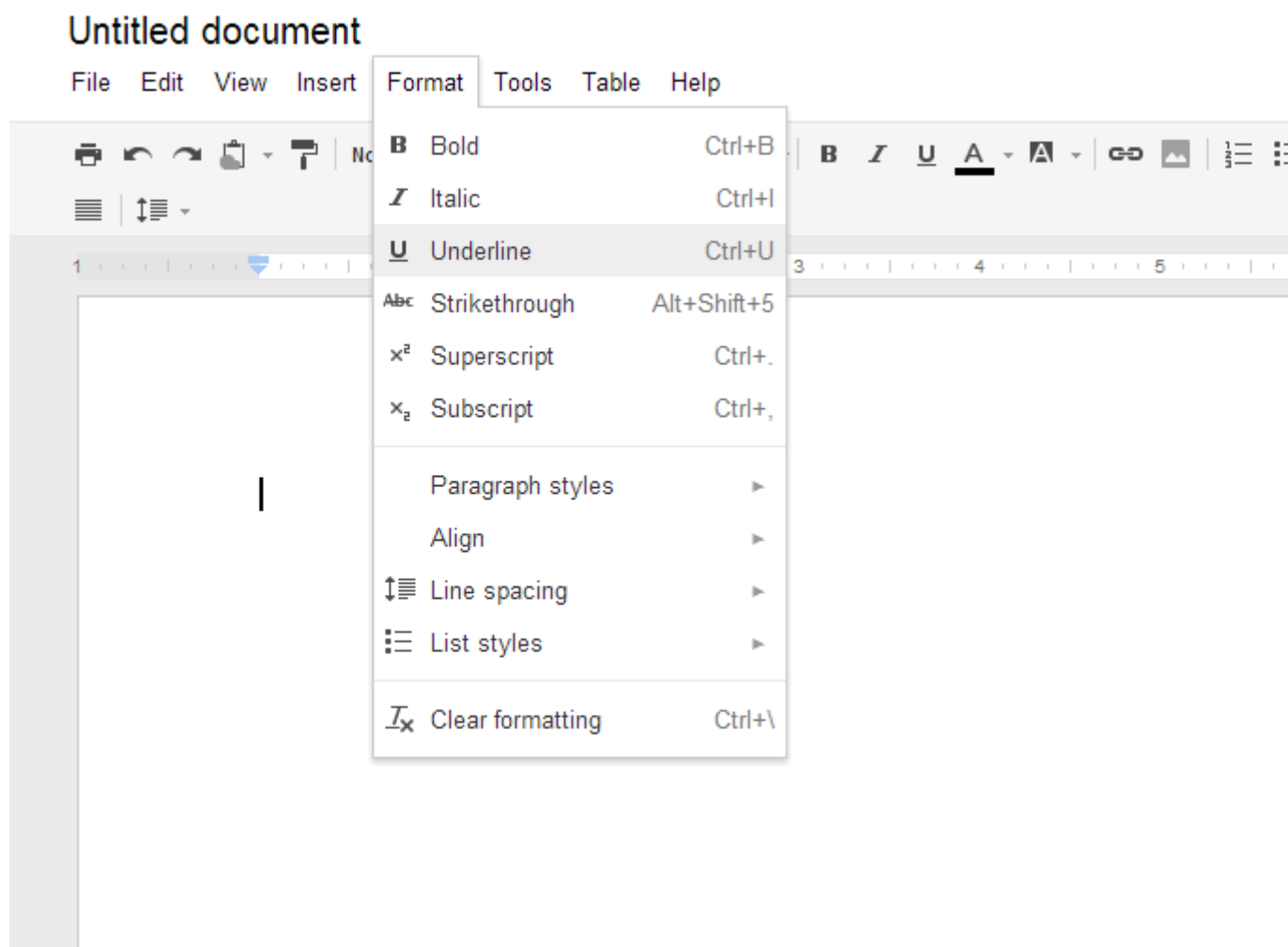
describen cómo debería aparecer la página, pero no dan instrucciones directas al ordenador ("haz esto, haz aquello, etc.") como hace JavaScript.

JavaScript no tiene *nada* que ver con el lenguaje Java. Solo se parecen los nombres.

JavaScript se utiliza normalmente hoy para AJAX (*JavaScript asíncrono y XML*). Esta técnica se utiliza para cambiar cualquier parte de la página web vista por el visitante a través del intercambio de datos con el servidor. Esto da la impresión de que las páginas son más dinámicas y sensibles. El visitante no ha de recargar sistemáticamente toda la página.

Como los navegadores son cada vez más eficaces procesando JavaScript, las páginas que utilizan JavaScript son más sensibles. Algunos sitios web que se crean ahora se convierten en aplicaciones web literalmente, el equivalente de software pero disponibles como sitios web.

Un ejemplo destacado: Google Docs, la suite de oficina de Google disponible en internet (figura siguiente).



La hoja de cálculo de Google Docs

¡Aprenderás más sobre Javascript en otros cursos!

Tecnologías relacionadas con HTML5 (Canvas, SVG, Web Sockets, etc.)

El W3C no solo funciona en los lenguajes HTML y CSS. Si bien son sin duda los más conocidos, el W3C quiere también identificar otras tecnologías que complementen a HTML y CSS. Hay un gran número y también se confunden a menudo con HTML5.

De hecho, HTML5 se ha convertido en un término ampliamente utilizado que se refiere a las tecnologías que no sean HTML. Cuando hoy en día alguien habla de "HTML5", también puede estar refiriéndose a otros elementos más allá del marco estricto de HTML.

Esta es una lista de las nuevas tecnologías introducidas en paralelo con HTML5 (ten en cuenta que algunas no son realmente "nuevas" sino que han vuelto a la escena):

- **Canvas:** te permite dibujar en la página web dentro de la etiqueta HTML `<canvas>`. Puedes dibujar figuras (triángulos, círculos, etc.) y añadir y hacer las cosas con imágenes, aplicar filtros gráficos, etc. Al final, todo esto nos permite desarrollar juegos reales y aplicaciones gráficas directamente en las páginas web.
- **SVG:** te permite crear gráficos vectoriales en páginas web. A diferencia de canvas, estos gráficos se pueden ampliar hasta el infinito (el principio del vector). El [programa Inkscape](#) es conocido por ser capaz de crear dibujos SVG.
- **Drag & Drop:** permite "arrastrar y soltar" objetos en la página web de la misma forma que puedes arrastrar y soltar archivos en tu escritorio. Gmail lo utiliza para permitirte añadir fácilmente archivos adjuntos a un correo electrónico.
- **File API:** proporciona acceso a los archivos almacenados en el ordenador del visitante (con su permiso). Concretamente lo usarás en combinación con Drag & Drop.
- **Geolocalización:** para localizar a los visitantes y ofrecer servicios relacionados con el lugar donde se encuentren (por ejemplo, los horarios de cines cercanos). Aunque la ubicación no siempre es muy precisa, se puede identificar a un visitante dentro de unos pocos kilómetros (con su permiso).
- **Almacenamiento web:** se utiliza para almacenar una gran cantidad de información en el ordenador del visitante. Es una alternativa más potente a las cookies de siempre. La información se organiza de forma jerárquica, como en una base de datos.
- **Appcache:** dice al navegador que ponga en caché ciertos archivos que ya no va a descargar sistemáticamente. Muy útil para crear aplicaciones web que pueden incluso trabajar "offline" (desconectadas).
- **Web Sockets:** permite intercambios rápidos, en tiempo real, entre el navegador del usuario y el servidor que gestiona el sitio web (es una especie de AJAX mejorado). Esto puede ser el futuro de las aplicaciones web, que acabarán siendo tan sensibles como los programas reales.

- **WebGL:** permite introducir el 3D en páginas web, utilizando el estándar OpenGL 3D (figura siguiente). Las escenas 3D las maneja directamente la tarjeta gráfica.

La mayor parte de estas tecnologías se utilizan con JavaScript. Se trata por tanto de nuevas funciones que se pueden utilizar en JavaScript.



Una aplicación web 3D usando WebGL

Como puedes ver, ¡hay nuevos mundos por descubrir! Cuando sepas lo suficiente de JavaScript, podrás avanzar en la gestión de tu sitio web ... ¡e incluso convertirlo en una aplicación real!

Sitios web dinámicos (PHP, JEE, ASP. NET, etc.)

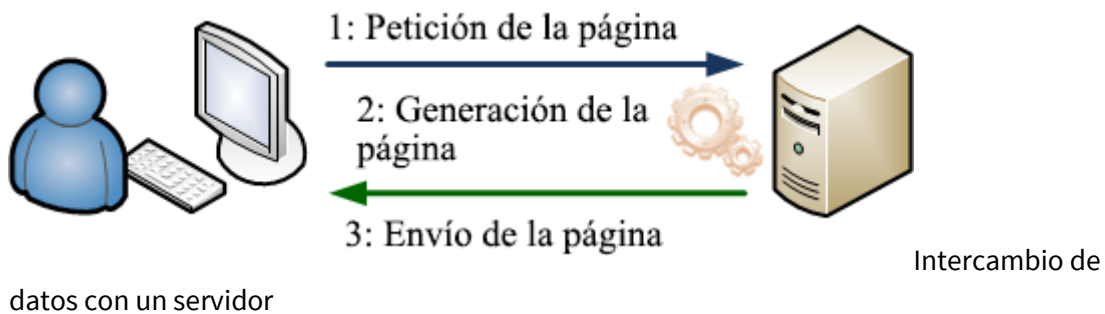
Los lenguajes de los que vamos a hablar aquí también son lenguajes de programación. ¿Como JavaScript? Sí, pero con una diferencia importante: JavaScript se ejecuta en el ordenador de tu visitante, mientras que los lenguajes que vamos a ver se ejecutan en el servidor que contiene tu sitio web.

¿Qué diferencia implica que el programa se ejecute en el ordenador del visitante o en el servidor?

Las diferencias son considerables. En primer lugar, en términos de potencia, un servidor suele ser más rápido que el ordenador del visitante, y por lo tanto puede realizar cálculos más complejos. También tienes un mayor control en el lado del servidor que en JavaScript ...

Pero JavaScript sigue siendo insustituible, ya que hay algunas cosas que solo se pueden hacer en el lado de "visitante".

Los lenguajes del servidor generan la página web cuando el visitante llega a tu sitio web (figura siguiente). ¡Cada visitante puede por tanto tener una página web personalizada de acuerdo a sus necesidades!



Así, vemos que los lenguajes no se utilizan para las mismas cosas sino que se complementan entre sí. Si combinas HTML + CSS + Javascript + PHP, por ejemplo, puedes manejar AJAX (intercambio de datos entre la página y servidor), realizar cálculos, almacenar información en bases de datos ... en definitiva, ¡crear sitios web dinámicos reales!

Hay muchos lenguajes "del lado del servidor". Mencionemos algunos:

- **PHP**: uno de los más conocidos. Este lenguaje fácil de utilizar y potente lo utiliza en particular Facebook. ¡También estoy escribiendo un tutorial de PHP!
- **JEE** (Java): esta extensión de Java se usa ampliamente en el mundo profesional para construir sitios web dinámicos, potentes y sólidos. Es un poco más complejo que PHP para empezar.
- **ASP .NET** (C#): es el lenguaje de Microsoft y es parecido a JEE. Se utiliza en combinación con otras tecnologías de Microsoft (Windows Server, etc.). Utiliza el potente marco .NET, que es la auténtica navaja suiza de los desarrolladores, y que ofrece muchas funciones.
- **Django** (Python): se trata de una extensión del lenguaje Python que permite construir sitios web dinámicos con rapidez y facilidad. Es conocido por generar listas para usar interfaces de administración.
- **Ruby on Rails** (Ruby): una extensión del lenguaje Ruby, bastante similar a Django, la cual permite construir sitios web dinámicos con facilidad y con mayor flexibilidad.

Conocer uno de estos lenguajes es esencial si quieres procesar los resultados de formularios HTML. Recuerda la etiqueta `<form>`: Te he explicado cómo crear formularios, pero no cómo *recuperar* la información introducida por tus visitantes. Has de tener un lenguaje de servidor, como PHP, para recuperar y procesar estos datos.

En última instancia, estos lenguajes permiten hacer realidad hacer tus mejores sueños en tu sitio web:

- foros;
- boletín de noticias;
- contador de visitantes;
- sistema de noticias automático;
- gestión de socios;
- juegos por internet (juegos de estrategia, de cría de animales virtuales, etc.)
- ...

Quiz 4

• Pregunta 1

En una tabla ¿Qué etiqueta de las siguientes se refiere a una fila?

- ☐ `<tr>`
- ☐ `<table>`
- ☐ `<td>`
- ☐ `<line>`

• Pregunta 2

Si quiero fusionar dos celdas de una misma linea en una tabla, ¿qué atributo neceisto utilizar?

- ☐ `fusion`
- ☐ `colspan`
- ☐ `rowspan`
- ☐ `groupline`

• Pregunta 3

La etiqueta `<tbody>` es obligatoria en una tabla HTML5. ¿Verdadero o falso?

- ☐ Verdadero
- ☐ Falso

• Pregunta 4

¿Cuál es el método más recomendado para enviar la máxima cantidad de datos en un formulario?

- ☐ get
- ☐ action
- ☐ post
- ☐ submit

• **Pregunta 5**

¿Cómo puedo añadir un valor por defecto en un campo `<textarea>` ?

- ☐ con el atributo `value`
- ☐ con el atributo `default`
- ☐ Incluyendo el texto entre las etiquetas de apertura y cierre (`<textarea>` et `</textarea>`)
- ☐ No es posible

• **Pregunta 6**

¿Qué etiqueta te permite incluir check boxes, campos de texto de una línea, y selectores de fecha?

- ☐ `<input />`
- ☐ `<form>`
- ☐ `<textarea>`
- ☐ `<select />`

• **Pregunta 7**

Si creo dos grupos diferentes de botones radio, ¿Cómo puedo permitir sólo seleccionar un valor en cada grupo?

- ☐ No necesitas hacer nada, es automático.
- ☐ Debes de situarlos al lado en el código
- ☐ Debes de colocarlos con el mismo `<fieldset>`
- ☐ Debes darle el mismo `name` a los botones de cada grupo

• **Pregunta 8**

¿Por qué es mejor introducir archivos en distintos formatos en una etiqueta `<video>` ?

- ☐ Para cargar el video más rápido
- ☐ Para que funcione con navegadores que no disponen de la etiqueta `<video>`
- ☐ Para permitir a los navegadores elegir el formato de vídeo compatible con ellos

- ☐ Para que funcione en navegadores que no soportan JavaScript

• Pregunta 9

¿Qué instrucción CSS se usa para crear un website responsive?

- ☐ @media
- ☐ screensize
- ☐ handheld
- ☐ [responsive]

• Pregunta 10

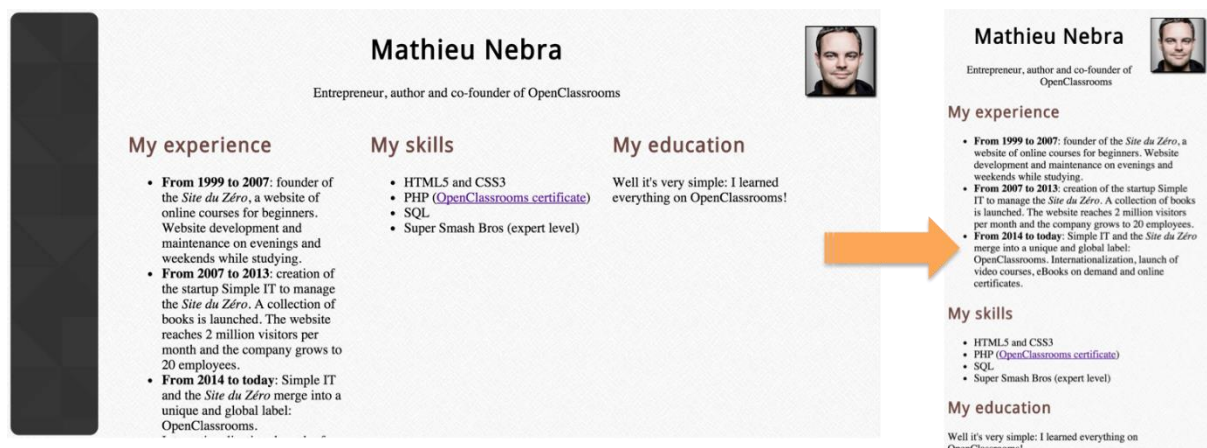
¿Qué ocurre si cambio el código HTML y CSS directamente utilizando las herramientas de debugging de mi navegador?

- ☐ Que se actualizan los archivos fuente
- ☐ Que la visualización de la página se actualiza pero los archivos fuente continúan sin cambiar
- ☐ Que el navegador deja de funcionar
- ☐ Que los cambios en el código se mandan al servidor web

html5es-part4 Assessment

En este ejercicio, adaptarás el CV que has hecho para que se pueda mostrar de forma vertical en pequeñas pantallas.

La imagen siguiente muestra cómo la página web debería mostrarse. La versión de la izquierda es la versión "normal" para pantallas grandes (Digamos, mayores de 1024 px de ancho), la versión de la derecha es una versión reducida más adaptada a las pantallas móviles.



Esto es lo que necesitas hacer en la versión móvil:

- Quita la barra lateral
- Muestra las secciones *experiencia/habilidades/educación* verticalmente en lugar de horizontalmente.

Envío de tu sitio web en internet

Tu sitio web está arreglado, limpio y listo... pero, como está en tu disco duro, ¡nadie más va a poder disfrutarlo!

Quieres enviarlo a internet, pero no sabes cómo.

En este anexo, vamos a averiguar todo lo que necesitas saber para enviar tu página web a internet:

1. ¿Cómo reservas un **nombre de dominio**?
2. ¿Qué es un **proveedor de alojamiento web** y cómo funciona?
3. Finalmente, ¿cómo usas un **cliente FTP** para transferir archivos en la red?

Nombre del dominio

¿Cómo sabes lo que es un **nombre de dominio** ?

En realidad, es una dirección en internet: `google.com` es un nombre de dominio, por ejemplo.

El nombre de un dominio se compone de dos partes: `google.com`.

- En rojo, el nombre del dominio en sí. Es un nombre que normalmente se puede elegir libremente, siempre y cuando nadie ya lo haya reservado antes. Puede contener letras y números.
- En azul, la extensión (también llamada "TLD", del inglés *Top-Level Domain*). Hay más o menos una extensión para cada país (`.es` para España, `.fr` para Francia, `.be` para Bélgica, `.ca` para Canadá, etc.). Sin embargo, también hay extensiones utilizadas internacionalmente como `.com`, `.net` y `.org`. Estaban reservadas originalmente para sitios web comerciales, organizaciones, etc., pero nadie ha sido capaz de reservarlos durante mucho tiempo. Y, de todos modos, `.com` es probablemente la extensión más utilizada en internet.

Una dirección de internet suele ir precedida de `www`, como `www.google.com`, por ejemplo. Esto no forma parte del nombre del dominio: de hecho, `www` es lo que se llama un sub-dominio y en teoría puedes crear tantos como quieras, siempre que seas el propietario del nombre del dominio.

`www` ha sido adoptado por todos los webmasters y es una especie de convención, a pesar de que no es en absoluto obligatorio.

Reservar el nombre de un dominio

¡Yo también quiero un nombre de dominio para mi sitio web! ¿Qué he de hacer?

Veamos, tengo buenas y malas noticias. Como de costumbre, vamos a comenzar con las malas noticias:

- **la mala noticia es que:** no es gratis...
- **la buena noticia es que:** ... en realidad no es muy caro.

De hecho, un nombre de dominio cuesta entre 7 y 12 euros al año.

El precio puede variar en función de la extensión. Por ejemplo, la extensión `.info` se ofrece generalmente a un precio más bajo y puede llegar a ser una alternativa atractiva. Pero si quieres una dirección más "común", mejor una extensión del tipo `.com`.

Para reservar un nombre de dominio, hay dos soluciones:

- Ir a través de un **registrador** especializado. Es una organización que actúa como puente entre la ICANN (la organización que gestiona todos los nombres de dominio a nivel internacional) y tú mismo. GoDaddy, Tucows y Register son empresas de registro muy conocidas.
- Mejor aún: puedes pedir el nombre de dominio junto con tu proveedor de alojamiento (esto es lo que yo recomiendo). De esta manera, matas dos pájaros de un tiro, puesto que de todos modos vas a necesitar un proveedor de alojamiento web y un nombre de dominio.

En este capítulo, vamos a ver cómo pedir un nombre de dominio junto con el proveedor de alojamiento web, ya que esto resulta de lejos lo más simple y menos costoso.

Proveedor de alojamiento web

Pasemos ahora al proveedor de alojamiento web.

¿Qué es un proveedor de alojamiento web y por qué necesito uno?

Todos los sitios web de internet se almacenan en equipos individuales llamados **servidores** (figura siguiente). Se trata generalmente de ordenadores muy potentes que no dejan de funcionar. Contienen las páginas del sitio web y se las proporcionan a los usuarios de internet según las solicitan, a cualquier hora del día o de la noche.

Un servidor

Un servidor no tiene pantalla puesto que funciona la mayor parte del tiempo por sí mismo sin necesidad de hacer nada en él. Como puedes ver, los servidores son muy planos: se

colocan en un formato de servidor especial (llamado "1U"). Esto permite apilarlos en **racks**, es decir, una especie de armario con aire acondicionado para servidores (figura siguiente).



Un rack de servidores

Como se ve, solo hay una pantalla para todo el rack. Es suficiente ya que la pantalla solo se conecta a un servidor cuando hay algún problema. La mayor parte del tiempo el servidor funciona afortunadamente sin problemas.

La función del proveedor de alojamiento web

El proveedor de alojamiento web es una empresa que se encarga de gestionar los racks del servidor. Se aseguran de que los servidores funcionen correctamente durante las veinticuatro horas del día, los siete días de la semana. De hecho, si un servidor falla, todos los sitios web que funcionen en él se vuelven inaccesibles (y esto crea clientes insatisfechos).

Estos racks están ubicados en lugares específicos llamados **centros de datos** (figura siguiente). Los centros de datos pueden por tanto considerarse "almacenes" de servidores y su acceso está altamente protegido.



Un centro de datos que contiene varios racks de servidores

En teoría, también puedes alojar un sitio web en tu propio ordenador. Esto es complejo; sin embargo, necesitarías algunos conocimientos de Linux, el equipo tendría que ser bastante potente y funcionar día y noche y, sobre todo, requeriría una conexión de muy alta velocidad (especialmente para **cargar**, ya que la velocidad a la que se envían archivos es crucial). Las personas no suelen tener una conexión lo suficientemente potente como para alojar sitios web, a diferencia de los centros de datos que cuentan con fibra óptica por cable (¡lo cual permite alcanzar velocidades de varios Gbps!)

En resumen, gestionar un servidor tú mismo es complejo; las personas y empresas en su mayoría confían en un proveedor de alojamiento web profesional.

Encontrar un proveedor de alojamiento web

Al contrario que registradores, hay muchos proveedores de alojamiento web. Hay muchos tipos y en todos los rangos de precios. Para escoger alguna de las soluciones que ofrecen, necesitas conocer la terminología utilizada:

- Alojamiento compartido : si optas por una solución de alojamiento compartido, tu sitio web se colocará en un servidor que administra varios sitios web a la vez (tal vez un centenar o más). Es la solución

más barata y la que yo recomiendo si estás empezando con tu sitio web.: si optas por una solución de alojamiento compartido, tu sitio web se colocará en un servidor que administra varios sitios web a la vez (tal vez un centenar o más). *Es la solución más barata y la que yo recomiendo si estás empezando con tu sitio web.*

- Alojamiento virtual dedicado: en este caso el servidor no gestiona muchos sitios web (normalmente menos de diez). Esta solución suele ser adecuada para sitios web que, por un lado, ya no son adecuados para el alojamiento compartido, ya que generan mucho tráfico (demasiados visitantes), y por otro lado, su propietario no puede permitirse una solución de alojamiento dedicado (ver a continuación).: en este caso el servidor no gestiona muchos sitios web (normalmente menos de diez). Esta solución suele ser adecuada para sitios web que, por un lado, ya no son adecuados para el alojamiento compartido, ya que generan mucho tráfico (demasiados visitantes), y por otro lado, su propietario no puede permitirse una solución de alojamiento dedicado (ver a continuación).
- **Alojamiento dedicado** (también conocido como "servidor dedicado"): esta es la solución para todo. El servidor solo gestiona tu sitio web y ningún otro. Ten en cuenta que es bastante caro y que es recomendable tener conocimientos de Linux para administrar el servidor de forma remota.

Por ejemplo, la propia Wikipedia utiliza un proveedor de alojamiento dedicado, ya que tiene una gran cantidad de tráfico.

¿Dónde encuentro un proveedor de alojamiento web?

Es muy sencillo.

Una búsqueda en Google de "alojamiento web", te dará millones de resultados. Las opciones te abrumarán.

Si puedo darte un consejo, te recomiendo que eches un vistazo al [alojamiento web de PlanetHoster](#), que ofrece servicios de alojamiento de calidad. ¡

Si las soluciones PlanetHoster no te convienen, echa un vistazo a los alojamientos web de [1&1](#) o [MavenHosting](#), que proporcionan otras soluciones para particulares y empresas.

Aunque el resto de este capítulo describe cómo alojar tu sitio web en *PlanetHoster*, funciona prácticamente de la misma manera con *1&1* y *Mavenhosting*.

Volvamos a *PlanetHoster*. Este proveedor de alojamiento web ofrece varias soluciones de alojamiento compartido, como se muestra en la siguiente figura.

Shared Web Hosting

PlanetHoster Shared packages include powerful features and tools to help you build and define your website, including file transfer protocol (FTP) accounts, popular programming languages, a blog, and redundant infrastructure.

Billing Options:
 Monthly
 Annually
 Biennially

Plan	Price /month	Features
Essential Plan	\$4.79	✓ 10GB Storage ✓ 250GB Bandwidth ✓ No Setup Fees ✓ Instant Activation ✓ FREE Domain Name
Performance Plan	\$9.59	✓ 50GB Storage ✓ 500GB Bandwidth ✓ No Setup Fees ✓ Instant Activation ✓ FREE Domain Name

Buttons: ADD TO CART, EXPAND PLANS

Alojamiento PlanetHoster

PlanetHoster ofrece descuentos especiales en todo su alojamiento web compartido para los lectores de este libro:

- 5% de descuento en el plan "Essential";
- 15% de descuento en los planes "Performance" y "Unlimited".

Estos descuentos son válidos si se introduce un código de promoción (entraré en más detalles sobre esto un poco más adelante) para un pedido anual de uno de estos planes:

- **Plan Essential:** Espacio de 10 GB de disco, tráfico de 250 GB, un nombre de dominio y número ilimitado de cuentas de correo electrónico;
- **Plan Performance:** Espacio de 50 GB de disco, tráfico ilimitado, un nombre de dominio y número ilimitado de cuentas de correo electrónico;
- **Plan ilimitado :** espacio en disco, tráfico y un número ilimitado de cuentas de correo electrónico, así como un nombre de dominio.: espacio en disco, tráfico y un número ilimitado de cuentas de correo electrónico, así como un nombre de dominio.

Estas soluciones son en realidad muy similares y solo difieren en su espacio de almacenamiento y el tráfico.

Pero ¿qué se entiende por "tráfico"?

El tráfico es la cantidad de datos enviados por mes a los visitantes de tu sitio web. Por ejemplo, si tienes una imagen de 1 MB en tu página web y tus visitantes la cargan

500 veces a lo largo del mes, creas 500 MB de tráfico.

En la práctica, hay que señalar que los navegadores de los visitantes suelen colocar las imágenes en la memoria caché, evitando así tener que recargar la misma imagen varias veces. Esto también reduce el tráfico necesario.

Si tienes una gran cantidad de visitantes (y por tanto de tráfico), debes elegir un plan que permita más tráfico.

Pedir una solución de alojamiento para tu sitio web

Al hacer clic en cualquier botón de "Añadir al carrito" vas a la página que se muestra en la siguiente figura.

The screenshot shows a checkout interface for Planet Hoster. On the left, under the heading "Domain Name" with a globe icon, there is a text prompt "Please enter the domain you would like to use below." and a search bar with a magnifying glass icon and the text "www." followed by an empty input field. On the right, the "ORDER SUMMARY" section is highlighted with a green header. It lists the selected plan as "Shared/Mutualisé Essential/Essentiel Plan" with a price of "\$57.60CAD". Below this, a yellow box displays the "Subtotal: \$57.60CAD" and "Total Due Today: \$57.60CAD". At the bottom of the summary, there is a field for "Enter Promotion Code" and a blue "GO" button. Below the button, it states "Recurring Charges: \$57.60CAD Annually".

Formulario de pedido de Planet Hoster

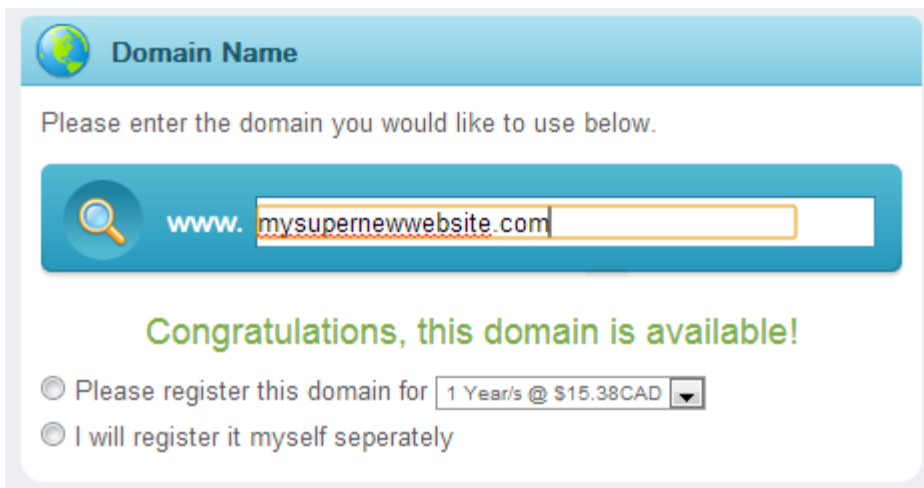
Hay tres elementos de información:

- Seleccionar un producto : indicar el plan elegido. Puedes cambiar esto directamente a través de este suplemento; indicar el plan elegido. Puedes cambiar esto directamente a través de este suplemento;
- **Nombre de dominio:** esta sección permite elegir el nombre de dominio de tu sitio web. Volveremos a esto.
- **Datos del pedido:** como su nombre indica, este cuadro resume tu pedido con el plan elegido y el precio a pagar.

El campo de texto abajo de **Nombre de dominio** solicita que introduzcas ... tu nombre de dominio. El sitio web de Planet Hoster comprobará a continuación si el dominio está disponible. Si lo está, puedes pasar al siguiente paso.

Si no, has de elegir otro nombre de dominio ya que ese ya lo han elegido y no se puede hacer mucho al respecto.

El sitio web te pregunta si deseas guardar este dominio o si quieres guardarlo por separado (figura de abajo).



PlanetHoster te indica si el nombre de dominio ya está registrado

Los alojamientos web suelen cobrar por un "verdadero" nombre de dominio (.com, .net, .org ...). Sin embargo, si compras servicios de alojamiento por un año en PlanetHoster, proporcionan el nombre de dominio de forma gratuita (está incluido en el servicio de alojamiento).

Todo lo que te queda por hacer es introducir tus datos personales y completar la compra con tarjeta de crédito o Paypal.

Cuando hayas completado los trámites y el pago, te redireccionarán a *PlanetHoster*, lo cual confirma que se ha registrado tu pedido. Un poco más adelante, has de recibir un correo electrónico con toda la información necesaria para configurar tu sitio web. Guarda esta información en un lugar seguro, ya que la necesitarás más adelante.

Cuando hayas recibido por correo electrónico la información de inicio de sesión para conectarte a tu servidor de alojamiento, puedes ir al siguiente paso: ¡ *enviar tu sitio web a tu servidor de alojamiento!*

Usar un cliente FTP

Instalar un cliente FTP

FTP significa *File Transfer Protocol* y, en pocas palabras, es la forma que usamos para enviar nuestros archivos.

Hay software disponible para usar FTP para transferir archivos a través de internet.

Hay, por supuesto, cientos de programas FTP gratuitos y de pago.

Para asegurar que coincidamos, te sugiero el que yo uso, que es gratuito: **FileZilla** (figura de abajo).



El icono del conocido cliente FTP FileZilla

Este programa no tiene nada que ver con Mozilla, más que también termina en "zilla". No creas que te estoy obligando a usar programas de la mismo publicador

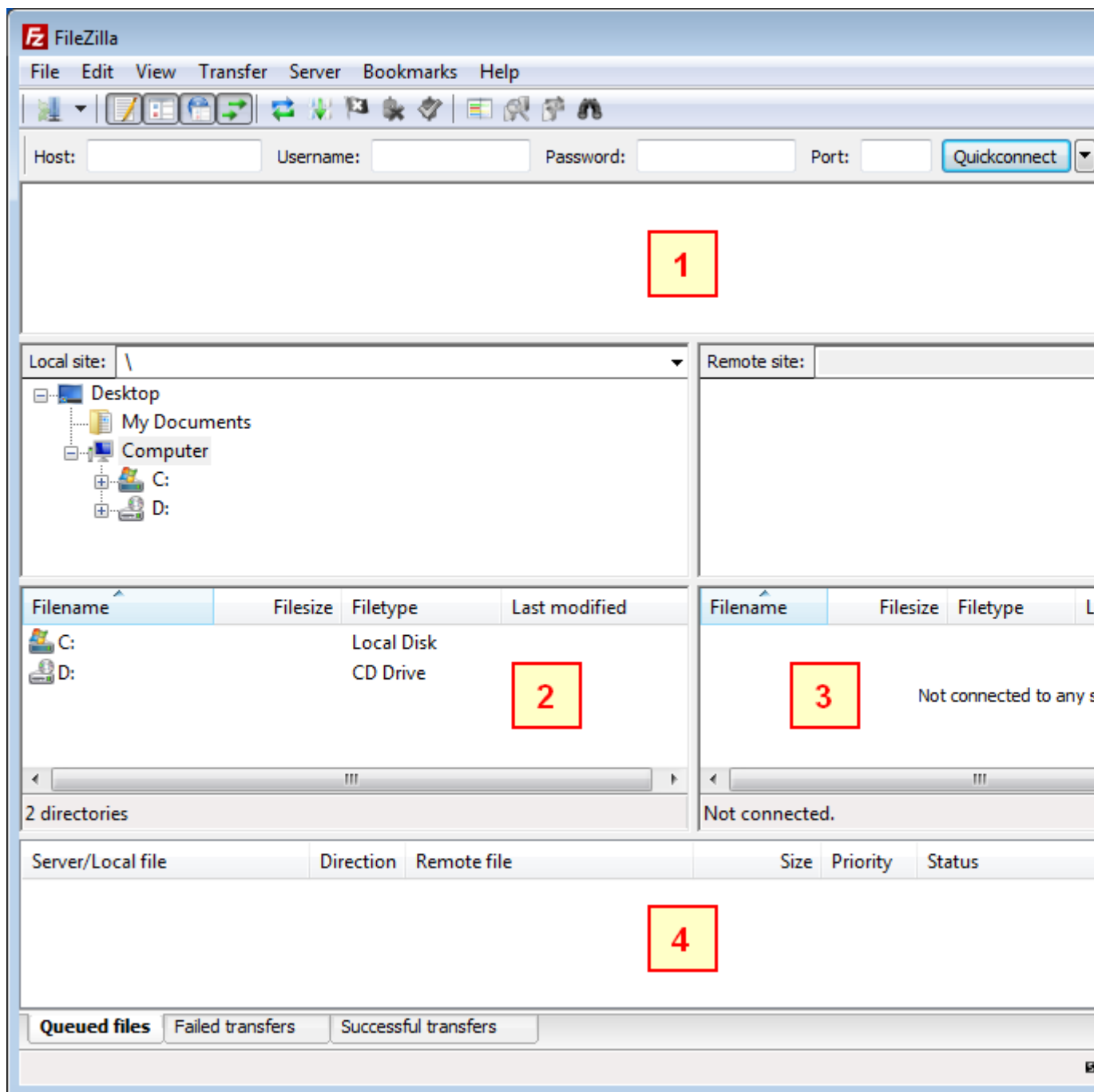
web. No es cierto. Además, puedes utilizar cualquier otro programa FTP, si quieres. No me importa.

De todos modos, voy a mostrar el procedimiento con FileZilla. El primer paso es, por supuesto, descargarlo.

Descargar FileZilla

Escoge la versión correspondiente a tu sistema operativo (Windows, Mac OS X o Linux)

Confío en ti para instalarlo. Es muy sencillo y no deberías tener ningún problema. Ejecuta el software y deberías ver algo parecido a la figura de abajo.



FileZilla

A primera vista, parece un poco complicado (solo a primera vista). De hecho, su principio es muy sencillo.

Hay cuatro campos principales a considerar en la ventana:

1. En la parte superior, verás los mensajes que el programa envía y recibe. Si tienes suerte, el servidor incluso te saludará (en serio, no es broma). En general, este campo no importa, a menos que haya mensajes de error en rojo...

2. Tu disco duro aparece a la izquierda. Las carpetas se muestran en la parte superior y la lista de los archivos de la carpeta actual aparece en la parte inferior.
3. La lista de los archivos subidos al sitio remoto se muestra a la derecha. De momento no aparece nada, ya que no estamos conectados, pero aparecerá no te preocupes.
4. Por último, en la parte inferior, verás los archivos que se envían (y el porcentaje enviados).

El primer paso será conectar con tu servidor de alojamiento.

Configurar el cliente FTP

Independientemente del servicio de alojamiento servicio que elijas, siempre funciona de la misma manera. Vamos a ofrecerte *tres elementos de información* esenciales para que FileZilla pueda conectar con el servidor:

- **IP:** es la "dirección" del servidor. Normalmente te dará información del tipo `ftp.myhost.com`, pero también puede ser una serie de números como `122.65.203.27`.
- **Usuario:** es tu nombre de usuario, que probablemente te habrán pedido elegir. Es posible que hayas introducido tu alias, o el nombre de tu sitio web. Mi usuario podría ser, por ejemplo, `mateo21`.
- **Contraseña:** o te han pedido que elijas una contraseña, o (lo que es más probable) te han asignado automáticamente una (algo impronunciable como `crf45u7h`).

Si tienes estos tres elementos de información, podrás continuar.

Si no, tendrás que buscarlos, ya que son esenciales. Es probable que te los enviaran por correo electrónico. Si no es así, no dudes en pedirselos a tu proveedor de alojamiento web (IP, nombre de usuario y contraseña).

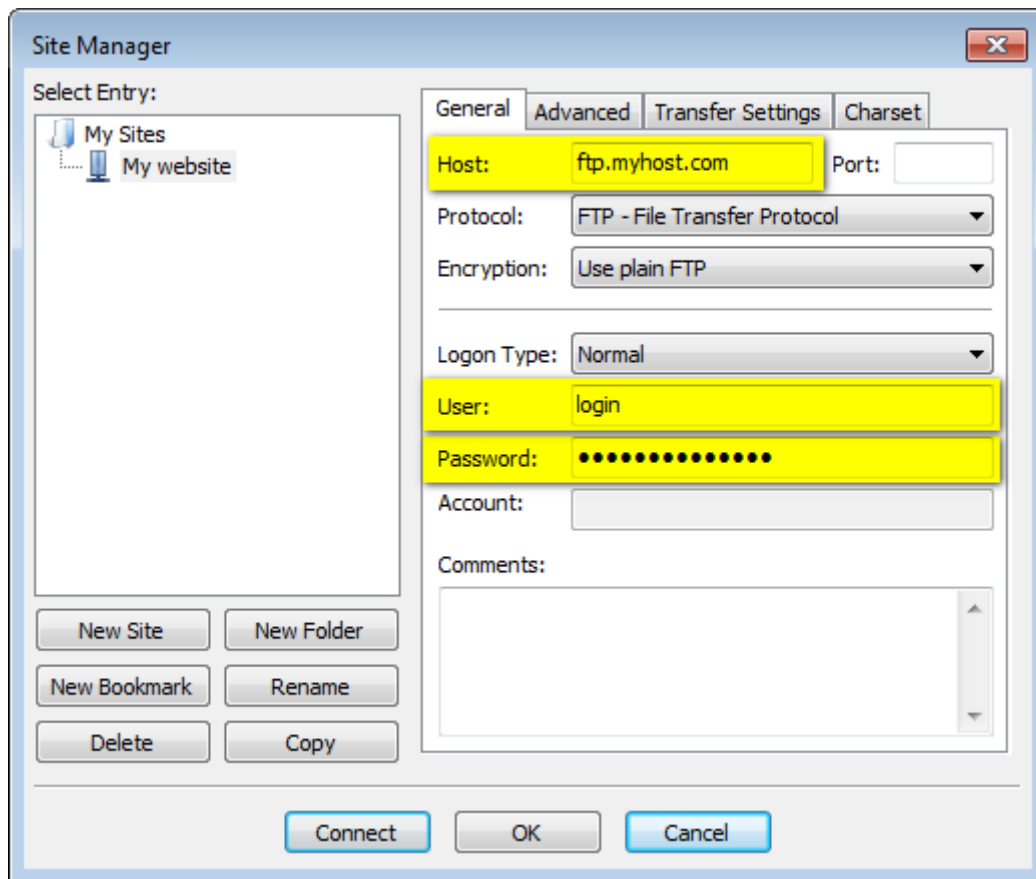
Ahora que tenemos esta información, vamos a dársela a FileZilla, que la necesita para conectarse al servidor.

Haz clic en el pequeño icono de la parte superior izquierda (no la flecha pequeña de la derecha, sino la imagen real), como muestra la siguiente figura.



El icono de conexión de FileZilla

Se abre una ventana. Haz clic en `Nuevo sitio` y dale el nombre que quieras ("Mi súper sitio web", por ejemplo). A la derecha, tendrás que especificar los tres elementos de información que acabo de mencionar, como en la siguiente figura.



Añadir

un nuevo sitio web en FileZilla

El alojamiento web se muestra en la parte superior (aquí es donde has de especificar `ftp.myhost.com`, por ejemplo). Comprueba Tipo de autenticación: `normal` para permitirte introducir tu nombre de usuario y contraseña.

Haz clic en `Conexión` y (casi) has terminado.

Transferir archivos

En este punto tenemos dos posibilidades:

- La conexión se ha realizado correctamente: verás entonces un mensaje verde como "Conectado" en la parte superior. En este caso, el campo de la derecha de la ventana de FileZilla debe activarse y verás los archivos que ya están en el servidor (puede haber unos pocos ya).
- O se ha colgado, en cuyo caso tendrás un montón de mensajes en rojo y no hay vuelta de hoja: has cometido un error al introducir la dirección IP, el nombre de usuario o la contraseña. Uno de estos elementos es incorrecto, así que asegúrate de pedírselos a tu proveedor de alojamiento web de nuevo si es necesario, ya que si son correctos *debe* funcionar.

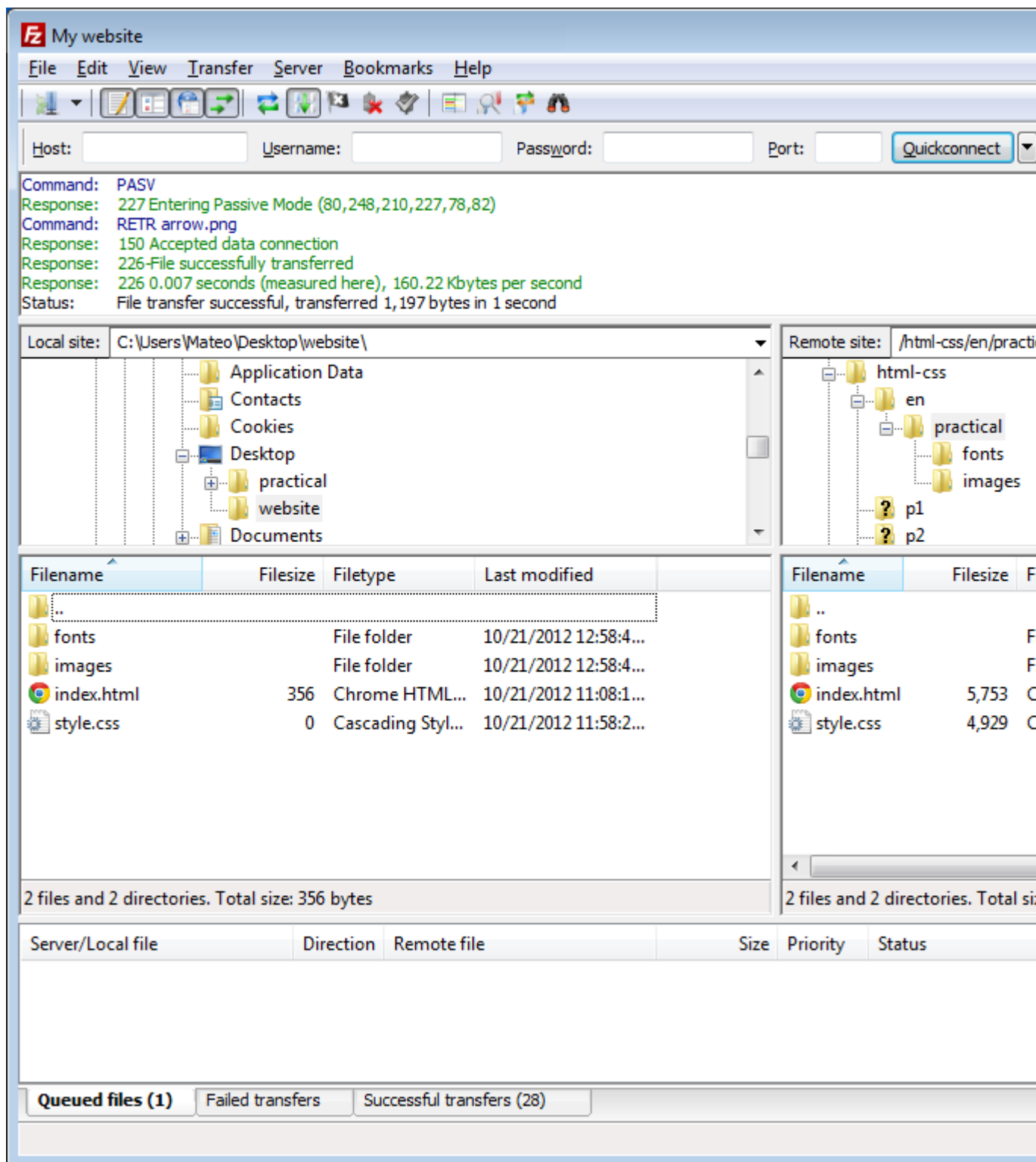
Si la conexión es correcta, lo que tienes que hacer es muy fácil: en la parte izquierda, mira dónde están ubicados tus archivos `.html` y `.css` (así como tus

imágenes .jpg, .png y .gif, etc.) en el disco duro.

A la izquierda, haz doble clic en el archivo que deseas transferir. Tras unos segundos, aparecerá a la derecha, lo que significa que se ha cargado correctamente en el servidor, ¡y por lo tanto es accesible en internet!

Puedes enviar cualquier tipo de archivo. Por supuesto, enviamos generalmente archivos e imágenes .php, .html, .css, pero puedes también enviar archivos .pdf, programas, archivos .zip, etc.

La siguiente figura, por ejemplo, muestra el resultado obtenido después de transferir un archivo `index.html` y algunos otros archivos.



Los archivos están alojados en el servidor FTP

Esto aparece a la derecha, lo que significa que ahora está disponible en el servidor.

Ten en cuenta que tu página de inicio *debe* llamarse `index.html`. Esta es la página que se carga cuando un nuevo visitante llega a tu sitio web.

También puedes transferir carpetas enteras de una sola vez: solo tienes que arrastrar y soltar la carpeta desde el lado izquierdo (o directamente desde la ventana de tu sistema operativo) al lado derecho de la ventana de FileZilla.

Una vez configurado, se puede ver que el envío de archivos es muy sencillo.

En resumen

- Por el momento, tu sitio web solo puedes verlo tú mismo en tu ordenador. Tienes que subirlo a internet para que puedan verlo todos.
- Necesitas dos cosas:
 - Un nombre de dominio: se trata de la dirección de tu sitio web. Puedes reservar una dirección como un dominio `.com`, `.fr` o `.net`. Por ejemplo: `google.com`.
 - Un proveedor de alojamiento web: almacenará tu sitio web en un equipo llamado "servidor". Su trabajo consistirá en enviar tu sitio web a tus visitantes en cualquier momento del día o de la noche.
- Para transferir archivos desde tu sitio web a tu servidor de alojamiento web, tienes que utilizar un cliente FTP como FileZilla.
- Para conectar con el servidor, necesitas tres cosas: la dirección IP del servidor (o su nombre de host), tu nombre de usuario y tu contraseña. Esto te lo proporciona tu proveedor de alojamiento web.

Recuerdo de las etiquetas HTML

Esta página ofrece una lista *no exhaustiva* de las etiquetas HTML existentes. Sí que contiene, sin embargo, un gran número de etiquetas HTML. Aunque ya hemos visto alguna durante el curso, hay otras que no hemos tenido la oportunidad de estudiar todavía. Las etiquetas que no hemos analizado todavía no se utilizan muy a menudo. No obstante, es muy posible que encuentres lo que necesitas en este nuevo conjunto de etiquetas.

Puedes utilizar este anexo como lista de verificación cuando desarrolles tu sitio web.

Quisiera, sin embargo, subrayar que: *este anexo no incluye todo intencionadamente*. Prefiero incluir *menos* etiquetas y solo mantener aquellas que parecen más útiles en la práctica.

Recuerdo

Etiquetas de primer nivel

Las etiquetas de primer nivel son las principales variables utilizadas para estructurar una página HTML. Son esenciales para crear el "código mínimo" de una página web.

Etiqueta	Descripción
<code><html></code>	Etiqueta principal
<code><head></code>	Encabezado de la página
<code><body></code>	Cuerpo de la página

Código mínimo de una página HTML:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Título</title>

</head>


<body>


</body>

</html>
```

Etiquetas de encabezado

Estas etiquetas se colocan todas en el encabezado de la página web, es decir entre `<head>` y `</ head>`:

Etiqueta	Descripción
<code><link /></code>	Enlace con una hoja de estilo
<code><meta /></code>	Metadatos de una página web (charset, palabras clave, etc.).
<code><script></code>	Código JavaScript
<code><style></code>	Código CSS

Etiqueta	Descripción
<title>	Título de la página

Etiquetas de estructuración del texto

Etiqueta	Descripción
<abbr>	Abreviatura
<blockquote>	Cita (larga)
<cite>	Citar el título de un trabajo o un evento
<q>	Cita (corta)
<sup>	Exponente
<sub>	Índice
	Resaltar fuerte
	Resaltar suave
<mark>	Resaltar visualmente
<h1>	Título nivel 1
<h2>	Título nivel 2
<h3>	Título nivel 3
<h4>	Título nivel 4
<h5>	Título nivel 5
<h6>	Título nivel 6
	Imagen
<figure>	Figura (imagen, código, etc.)
<figcaption>	Descripción de la figura
<audio>	Sonido
<video>	Vídeo
<source>	Formato de fuente para las etiquetas <audio> y <video>

Etiqueta	Descripción
<a>	Enlace de hipertexto
 	Salto de línea
<p>	Párrafo
<hr />	Línea de separación horizontal
<address>	Dirección de contacto
	Texto borrado
<ins>	Texto insertado
<dfn>	Definición
<kbd>	Entrada de teclado
<pre>	Visualización con formato (para códigos fuente)
<progress>	Barra de progreso
<time>	Fecha u hora

Etiquetas de lista

En esta sección se enumeran todas las etiquetas HTML utilizadas para crear listas (listas con viñetas, listas numeradas, listas de definiciones, etc.)

Etiqueta	Descripción
	Listas de viñetas desordenadas
	Lista numerada
	Elemento de lista con viñetas
<dl>	Lista de definiciones
<dt>	Término a definir
<dd>	Definición del término

Etiquetas de tabla

Etiqueta	Descripción
<table>	Tabla

Etiqueta	Descripción
<caption>	Título de tabla
<tr>	Fila de tabla
<th>	Celda de encabezado
<td>	Celda
<thead>	Sección del encabezado de la tabla
<tbody>	Sección del cuerpo de la tabla
<tfoot>	Sección del pie de la tabla

Etiquetas de formulario

Etiqueta	Descripción
<form>	Formulario
<fieldset>	Grupo de campo
<legend>	Título del grupo de campo
<label>	Descripción del campo
<input />	Campo de formulario (texto, contraseña, casilla, botón, etc.).
<textarea>	Campo de entrada de líneas múltiples
<select>	Lista desplegable
<option>	Elemento de lista desplegable
<optgroup>	Grupo de elementos de una casilla desplegable

Etiquetas de sección

Estas etiquetas se utilizan para esbozar nuestra página web.

Etiqueta	Descripción
<header>	Encabezado
<nav>	Enlaces de navegación principales
<footer>	Pie de página

Etiqueta	Descripción
<code><section></code>	Sección de página
<code><article></code>	Artículo (contenido independiente)
<code><aside></code>	Información adicional

Etiquetas genéricas

Las etiquetas genéricas son etiquetas que no tienen ningún significado semántico.

De hecho, todas las otras etiquetas HTML tienen un *significado*: `<p>` significa "Párrafo", `<h2>` significa "Subtítulo", etc.

A veces se necesita utilizar etiquetas genéricas (también llamadas **etiquetas de finalidad general**) ya que ninguna de las otras etiquetas es apropiada. Lo que más a menudo utilizas para construir tu diseño son las etiquetas genéricas.

Hay dos etiquetas genéricas: inline y block.

Etiqueta	Descripción
<code></code>	Etiqueta genérica inline
<code><div></code>	Etiqueta genérica block

Estas etiquetas solo son útiles si se las asocia con un atributo `class`, `id` o `style`:

- **class**: especifica el nombre de la clase de CSS que se utilizará.
- **id**: da un nombre a la etiqueta. Este nombre debe ser único en la página, ya que identifica a la etiqueta. Puedes utilizar el ID para muchas cosas, por ejemplo para crear un enlace a un ancla, para un estilo CSS tipo ID, para operaciones en JavaScript, etc.
- **style**: este atributo permite especificar directamente el código CSS que se aplicará. No estás obligado a tener una hoja de estilos separada, puedes incluir los atributos CSS directamente. Ten en cuenta que es preferible utilizar una hoja de estilos externa en lugar de este atributo, ya que hace que tu página web sea más fácil de actualizar más adelante.

Estos tres atributos no se limitan a las etiquetas genéricas sino que puedes utilizarlos sin ningún problema en la mayoría de las demás etiquetas.

Recuerdo de las propiedades CSS

Esta página ofrece una lista *no exhaustiva* de las propiedades CSS disponibles en CSS3. Aunque en su mayor parte hemos visto estas propiedades durante el curso, hay algunas nuevas que no hemos abordado.

La lista no es exhaustiva, ya que mi objetivo no es elaborar una lista de todas las propiedades CSS que puedan existir; en realidad, hay demasiadas (¡más de doscientas!) y algunas se utilizan muy raramente.

Recuerdo

Propiedades para dar formato al texto

Voy a resumir aquí la mayor parte de las propiedades para dar **formato al texto**.

¿Qué es dar formato al texto? Es todo lo que se refiere a la presentación del texto en sí: negrita, cursiva, subrayado, tipo de letra, alineación, etc.

Propiedad	Valores (ejemplos)	Descripción
font-family	<i>police1, police2, police3, serif, sans-serif, monospace</i>	Nombre de la fuente
@font-face	<i>Origen y nombre de la fuente</i>	Fuentes personalizadas
font-size	1.3em, 16px, 120%...	Tamaño de la fuente
font-weight	bold, normal	Negrita
font-style	italic, oblique, normal	Cursiva
text-decoration	underline, overline, line-through, blink, none	subrayado, línea por encima, tachado, parpadeo
font-variant	small-caps, normal	Mayúsculas pequeñas
text-transform	capitalize, lowercase, uppercase	Mayúsculas
font	-	La súper propiedad fuente combina: font-weight, font-

Propiedad	Valores (ejemplos)	Descripción
		style, font-size, font-variant, font-family.
text-align	left, center, right, justify	Alineación horizontal
vertical-align	baseline, middle, sub, super, top, bottom	Alineación vertical (celdas de la tabla, o solo elementos inline-block)
line-height	18px, 120%, normal...	Altura de la línea
text-indent	25px	Subpárrafo
white-space	pre, nowrap, normal	Unión con guión
word-wrap	break-word, normal	Unión con guión forzada
text-shadow	5px 5px 2px azul (horizontal, vertical, fade, color)	Sombreado del texto

Propiedades de color y fondo

Propiedad	Valores (ejemplos)	Descripción
color	nombre, rgb (red, green, blue), rgba (red, green, blue, transparency), #CF1A20, etc.	Color del texto
background-color	<i>Igual que el color</i>	Color de fondo
background-image	url('image.png')	Imagen de fondo
background-attachment	fixed, scroll	Fondo fijo
background-repeat	repeat-x, repeat-y, no-repeat, repeat	Repetición del fondo
background-position	(x y), top, center, bottom, left, right	Posición del fondo
background	-	Súper propiedad background. Combina: background-image, background-

Propiedad	Valores (ejemplos)	Descripción
		repeat, background-attachment, background-position

opacity 0.5 Transparencia

Propiedades box

Propiedad	Valores (ejemplos)	Descripción
width	150px, 80%...	Width
height	150px, 80%...	Altura
min-width	150px, 80%...	Ancho mínimo
max-width	150px, 80%...	Ancho máximo
min-height	150px, 80%...	Altura mínima
max-height	150px, 80%...	Altura máxima
margin-top	23px	Margen superior
margin-left	23px	Margen izquierdo
margin-right	23px	Margen derecho
margin-bottom	23px	Margen inferior
margin	23px 5px 23px 5px (top, right, bottom, left)	Súper propiedad margin Combina: margin-top, margin-right, margin-bottom, margin-left.
padding-left	23px	Margen interno a la izquierda
padding-right	23px	Margen interno a la derecha
padding-bottom	23px	Margen interno inferior
padding-top	23px	Margen interno superior

Propiedad	Valores (ejemplos)	Descripción
padding	23px 5px 23px 5px (<i>top, right, bottom, left</i>)	Súper propiedad inner margin. Combina: padding-top, padding-right, padding-bottom, padding-left.
border-width	3px	Ancho del borde
border-color	<i>nombre</i> , rgb (red, green, blue), rgba (red, green, blue, transparency), #CF1A20, etc.	Color del borde
border-style	solid, dotted, dashed, double, groove, ridge, inset, outset	Tipo de borde
border	3px solid black	Súper propiedad border. Combina border-width, border-color, border-style. También disponible para estas versiones border-top, border-right, border-bottom, border-left.
border-radius	5px	Bordes redondeados
box-shadow	6px 6px 0px black (<i>horizontal, vertical, fade, color</i>)	Box shadow

Propiedades del posicionamiento en pantalla

Propiedad	Valores (ejemplos)	Descripción
display	block, inline, inline-block, table, table-cell, none...	Tipo de elemento (block, inline, inline-block, none...)
visibility	visible, hidden	Visibility
clip	rect (0px, 60px, 30px, 0px) <i>rect (top, right, bottom, left)</i>	Mostrar una porción del elemento
overflow	auto, scroll, visible, hidden	Comportamiento en caso de exceder

Propiedad	Valores (ejemplos)	Descripción
<code>float</code>	<code>left</code> , <code>right</code> , <code>none</code>	Flotante
<code>clear</code>	<code>left</code> , <code>right</code> , <code>both</code> , <code>none</code>	Parar algo flotante
<code>position</code>	<code>relative</code> , <code>absolute</code> , <code>static</code>	Posicionamiento
<code>top</code>	<code>20px</code>	Posición relativa a la parte superior
<code>bottom</code>	<code>20px</code>	Posición relativa a la parte inferior
<code>left</code>	<code>20px</code>	Posición relativa a la parte izquierda
<code>right</code>	<code>20px</code>	Posición relativa a la parte derecha
<code>z-index</code>	<code>10</code>	Mostrar el orden en caso de solapamiento. El valor más alto se muestra por encima de los demás.

Propiedades list

Propiedad	Valores (ejemplos)	Descripción
<code>list-style-type</code>	<code>disc</code> , <code>circle</code> , <code>square</code> , <code>decimal</code> , <code>lower-roman</code> , <code>upper-roman</code> , <code>lower-alpha</code> , <code>upper-alpha</code> , <code>none</code>	Tipo de lista
<code>list-style-position</code>	<code>inside</code> , <code>outside</code>	Posición de la sangría
<code>list-style-image</code>	<code>url('puce.png')</code>	Viñeta personalizada
<code>list-style</code>	-	Súper propiedad list. Combina <code>list-style-type</code> , <code>list-style-position</code> , <code>list-style-image</code> .

Propiedades de tabla

Propiedad	Valores (ejemplos)	Descripción
<code>border-collapse</code>	<code>collapse</code> , <code>separate</code>	Fusionar bordes
<code>empty-cells</code>	<code>hide</code> , <code>show</code>	Mostrar las celdas en blanco
<code>caption-side</code>	<code>bottom</code> , <code>top</code>	Posición del título de tabla

Otras propiedades

Propiedad	Valores (ejemplos)	Descripción
cursor	crosshair, default, help, move, pointer, progress, text, wait, e-resize, ne-resize, auto...	Cursor del ratón

Manejar la compatibilidad entre navegadores

Esta es la situación: estás trabajando duro en el diseño de tu sitio web. Has invertido tiempo en él, teniendo cuidado y al final estás bastante satisfecho contigo mismo. Haces que un amigo pruebe el sitio web y responde: "He de decir que no está tan bien".

O bien tu amigo y tú tenéis gustos completamente diferentes, o el sitio web está bien *desde tu lado*, pero no *desde el suyo*. De hecho, los navegadores no generan las imágenes exactamente de la misma manera, aunque intenten seguir las mismas reglas de HTML y CSS. Cuanto más antiguo sea el navegador, más probable será que no entienda las funciones que estés utilizando.

En lugar de entrar en pánico, puedes encontrar en este capítulo algunas maneras de aprender a tratar estas diferencias entre navegadores y obtener los mejores resultados posibles, independientemente del navegador.

Comentarios condicionales

A menudo las versiones antiguas de Internet Explorer ocasionarán problemas. Como algunas de ellas todavía se usan, tienes que aprender a lidiar con ellas.

Sintaxis de los comentarios condicionales

Afortunadamente, hay una manera muy cómoda de adaptar el código a versiones antiguas de Internet Explorer. Mediante comentarios condicionales, puedes asegurarte de que una parte de tu código HTML sea de solo lectura para ciertas versiones de IE. Estos son comentarios HTML que tienen una forma bastante especial y solo los leerá IE.

```
<!--[if IE]>

Código HTML reservado para IE

<![endif]>
```

El conjunto forma un comentario HTML, empieza con `<!--` y termina en `-->`. Por lo tanto, los navegadores ignoran este código situado en el interior, excepto IE, que lo interpreta como una condición `[if IE]` que significa "Si se trata de Internet Explorer".

La ventaja es que esta técnica te permite abordar diferentes versiones de Internet Explorer. Por ejemplo:

```
<!--[if IE8]>
```

... solo abordará Internet Explorer 8.

Los siguientes símbolos se pueden utilizar para abordar a un conjunto de versiones:

- lt: menor que;
- lte: menor que o igual a;
- gt: mayor que;
- gte: mayor que o igual a.

Por tanto:

```
<!--[if lte IE8]>
```

... significa "Todas las versiones de Internet Explorer menores o iguales a IE8" (en otras palabras, IE8, IE7, IE6 ...).

Por otro lado, los navegadores que no sean Internet Explorer pueden abordarse con la siguiente sintaxis:

```
<!--[if !IE]><!-->
```

Código HTML para algunos de los otros navegadores

```
<!--<![endif]-->
```

Aunque la sintaxis es un poco compleja, la solución está bien diseñada. Si te fijas bien, hay dos etiquetas de comentario. Como el código HTML se encuentra fuera de las etiquetas de comentario, lo leerán todos los navegadores excepto IE, que lee los comentarios que le piden que no lea el código HTML que sigue.

[Utilidad de los comentarios condicionales](#)

¿Para qué se usan los comentarios condicionales?

Puede utilizar comentarios condicionales para mostrar un mensaje destinado a versiones anteriores de Internet Explorer: "Precaución, estás navegando con un explorador muy antiguo, por favor, actualízalo...".

No obstante, se utiliza sobre todo para cargar hojas de estilos especiales para IE:

```
<!--[if lte IE8]>
```

```
<link rel="stylesheet" href="style_ie.css" />
```

```
<![endif]-->
```

La hoja de estilos `style_ie.css` la leerán, en este caso, solamente versiones de IE *no superiores a* IE8. Puedes "adaptar" el código CSS del interior para asegurarte de que se muestre el sitio web lo mejor que se pueda en estos navegadores. Depende de ti cambiar el tamaño de los elementos, si es necesario, o utilizar otros métodos de posicionamiento más adecuados, etc.

Sin embargo, ya que hay varias versiones de Internet Explorer en circulación (IE6, IE7, IE8, IE9, etc.), una hoja de estilos no es suficiente. A veces querrás hacer un cambio solo para IE6 y IE7, otras veces solo para IE6 y en ocasiones para IE8 y IE9... La única solución es crear una hoja de estilos para cada versión. Pero esto hará que tengas que manejar una gran cantidad de archivos `.css`.

La solución que ahora suelen usar los webmasters experimentados es asignar un nombre de clase a la etiqueta `<body>` que abarca el cuerpo de la página. El nombre de la clase indicará el nombre de la versión de IE correspondiente al navegador, en el archivo `.css`. En este caso, un solo archivo CSS es suficiente. Por ejemplo, si escribes:

```
<body class="ie8">
```

... IE8 se especifica como navegador en el archivo CSS. Podemos entrar en lo siguiente en el archivo CSS:

```
.ie8 nav
{
    /* Cambiar el estilo de la etiqueta de navegación para IE8 */
}
```

Con comentarios condicionales, puedes escribir varias versiones de la etiqueta `<body>` según el navegador:

```
<!--[if IE6]><body class="ie6 old_ie"><![endif]-->

<!--[if IE7]><body class="ie7 old_ie"><![endif]-->

<!--[if IE8]><body class="ie8"><![endif]-->

<!--[if IE9]><body class="ie9"><![endif]-->

<!--[if !IE]><!--><body><!--<![endif]-->
```

Podemos dar varios nombres de clase a una etiqueta separados por un espacio, como se muestra aquí. Así, si el navegador es IE6, la etiqueta `<body>` tendrá dos

clases: `ie6` y `code data-claire-semantic="html">old_ie`. La clase `old_ie` se aplica por tanto a IE6 y IE7 simultáneamente.

Hacks CSS

Los "*Hacks*" son soluciones temporales para permitir que solo algunas versiones de Internet Explorer lean propiedades CSS. El principio es el mismo que en los comentarios condicionales, pero es menos "transparente"; los hacks se basan en errores de estos navegadores causados al leer propiedades CSS cuando no deberían.

En general, no se recomienda utilizar hacks, ya que se basan en errores del navegador. No hay garantía de que funcionen en el futuro.

Este es uno de los hacks más conocidos:

```
balise
{
    margin: 15px; /* Para otros navegadores */

    ma\rigin: 20px; /* Para IE6 y IE7 */
}
```

Un error en IE6 y IE7 hace que lean la segunda propiedad. Normalmente, la barra invertida (\) en medio del nombre de la propiedad hace que no sea válida, pero estos navegadores ignoran la barra invertida y leen la propiedad como si todo fuera normal.

Existen otros hacks, por ejemplo:

```
balise
{
    margin: 15px; /* Para otros navegadores */

    *margin: 20px; /* Para IE6 y IE7 */

    _margin: 20px; /* Para IE6 */
}
```

Un asterisco antes del nombre de la propiedad hace que todos los navegadores, excepto IE6 e IE7, ignoren la línea. Del mismo modo, incluir un guión bajo (_) hace que la línea sea ignorada por todos los navegadores, excepto IE6.

Herramientas de depuración

Tu página a veces no se visualiza como quieres, incluso en un navegador más reciente. En este caso, tienes que buscar el error que has cometido. Aunque hay varias técnicas, solo voy a mostrar las que utilizo más a menudo.

Utilizar el color de fondo para marcar elementos

La técnica más sencilla para depurar es simplemente dar un color de fondo a un elemento que no esté comportándose como quisieras (por ejemplo, un menú mal colocado). Es tan simple como eso:

```
nav
{
    background-color: blue;
}
```

Este color de fondo muy llamativo es temporal, lo usas solo para ver dónde está "colocado" tu elemento. A menudo, esto te permite entender lo que está pasando en tu página.

Si esta técnica no es suficiente, tenemos que sacar la artillería pesada: una herramienta de depuración.

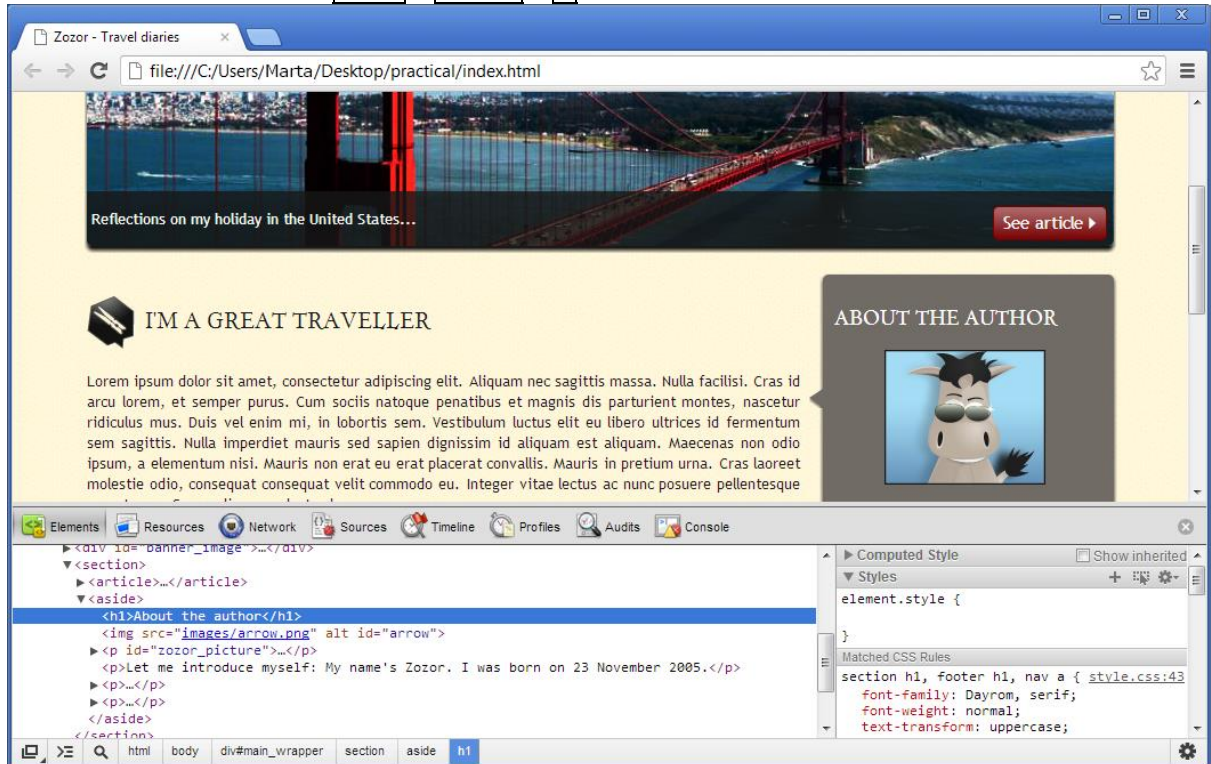
Herramientas de depuración

Son herramientas muy potentes (y bastante impresionantes) que te permiten interactuar con la página web *directamente* en el navegador. La mayoría de los navegadores actuales cuentan con estas herramientas, especialmente diseñadas para los desarrolladores web que somos:

- **Google Chrome:** ve al menú `Llave inglesa > Herramientas > Herramientas de depuración` (figura de abajo). También puedes hacer clic derecho en un elemento de tu página y a continuación hacer clic en `Inspeccionar el elemento`.
- **Mozilla Firefox:** instala la extensión [Firebug](#). Aparecerá un botón en la parte superior o inferior derecha de la ventana (dependiendo de la versión del navegador) para abrir el panel de depuración.
- **Internet Explorer:** ve al menú `Rueda de engranaje > Herramientas de desarrollo` (disponibles para IE8 y superiores).
- **Opera:** puedes ejecutar su potente herramienta de depuración *Dragonfly* pulsando `Ctrl + Mayús + I`. También puedes hacer

clic derecho en un elemento de tu página y a continuación hacer clic en Inspeccionar el elemento.

En Google Chrome, Mozilla Firefox e Internet Explorer, puedes abrir el panel de depuración pulsando la tecla **F12** del teclado. En Google Chrome y Firefox, también puedes pulsar las teclas **Ctrl** + **Mayús** + **I** para abrir el panel de depuración.



Herramienta de depuración de Google Chrome (en la parte inferior del navegador)

Hay varias maneras de utilizar estas herramientas. Puedes examinar la jerarquía de tus etiquetas visualizando el código fuente (en la parte inferior izquierda de la figura a continuación) y editar el código de tu página en tiempo real: haz doble clic sobre un texto o una etiqueta que quieras editar ¡y verás el cambio del resultado en la ventana del navegador a medida que avanzas! Una advertencia: los cambios no se guardan, ya que solo estás haciendo una prueba sencilla para entender el comportamiento del navegador.

La lista de las propiedades CSS que se aplican al elemento seleccionado actualmente en el código fuente se muestran en la parte inferior derecha. Puedes añadir, quitar o cambiar sus propiedades, etc. Todos los cambios se aplican inmediatamente. Ten en cuenta que las propiedades CSS que sobrescriban otras propiedades se tachan o se difuminan; esto te permite entender cómo se comporta tu hoja de estilos.

Por último, para abordar directamente un elemento de tu página, puedes hacer clic derecho sobre el elemento a analizar, y luego hacer clic en Inspeccionar elemento. La barra de herramientas de depuración también ofrece un botón para abordar un elemento (una lupa en Google Chrome, un cursor en IE, etc.).

Estas herramientas proporcionan muchas otras funciones más avanzadas que, en lugar de entrar a estudiar aquí, te dejo descubrir por ti mismo: la página de análisis del tiempo de carga de la página, depuración de JavaScript... Has de saber, sin embargo, que se usan principalmente para depurar código HTML y CSS.

¡Aquí termina este curso! Considera consultar todos los apéndices si no lo has hecho todavía: ¡descubrirás nuevas pistas para continuar tu aprendizaje de HTML y CSS!