

# Lenguajes de programación HTML y CSS

## Parte 2. Formato con CSS

### Aplicación de CSS

Después de haber terminado la primera parte, exclusivamente en HTML, ahora nos encontramos con CSS. CSS no es más complicado que HTML, es el complemento para ayudarte a dar formato a tu página web.

En esta primera parte sobre CSS, veremos la teoría CSS: ¿qué es? ¿qué aspecto tiene? ¿cuándo se escribe el código CSS? Estos aspectos teóricos no son muy complicados, pero hay que entenderlos, porque es la base de CSS. Esto es lo único que se debe recordar de memoria en CSS.

### Historia de CSS

Había informado previamente que aprenderemos dos lenguajes. Ya hemos empezado nuestro descubrimiento de HTML, aunque hay muchas más cosas que aprender (volveremos al HTML más adelante). Sin embargo, ahora es tiempo para ocuparnos de CSS.

CSS (*Cascading Style Sheets*), es otro lenguaje que complementa HTML. ¿Recordarás su papel? Consiste en gestionar el diseño de tu sitio.

### ¿Para qué sirve CSS?

- Permite elegir el color de tu texto.
- Permite seleccionar el tipo de letra utilizado en tu sitio.
- Establece el tamaño, bordes, fondo ...

Además, puede hacer el diseño de tu sitio. Se le puede decir: quiero que mi menú esté a la izquierda y ocupe una anchura tal que la cabecera de mi sitio siempre sea visible, etc.

### CSS: comienzo difícil

Debes saber que al principio de la Web, CSS no existía. De hecho, solo había HTML.

El código HTML nació en 1991 y CSS en 1996. Por lo tanto, puedes pensar sin duda: ¿cómo se daba formato desde 1991 hasta 1996? Pues sólo en HTML. En efecto, hay etiquetas para dar

formato en HTML. Por ejemplo `<font color = "#aab1c3">` permite fijar el color del texto en HTML.

Sin embargo, las páginas HTML llegaron a ser muy complejas. Había más y más etiquetas y eran una mezcla anárquica de contenido y formato, lo que hizo la actualización de las páginas web un proceso más complejo. Es por eso que se creó CSS. Sin embargo, CSS no fue inmediatamente adoptado por los diseñadores, ni mucho menos. Tenían que deshacerse de algunos malos hábitos y eso llevó tiempo. Incluso hoy en día, es frecuente encontrar sitios web con formato HTML, viejo y obsoleto, como el uso de la etiqueta `<font>`.

### **CSS: soporte de los navegadores**

Al igual que HTML, CSS ha evolucionado. Ha habido cuatro versiones principales de CSS:

- CSS 1
- CSS 2.0
- CSS 2.1
- CSS 3

Son los navegadores web los que hacen el trabajo más complejo, deben leer el código CSS y entender cómo mostrarlo en la página.

En la década del 2000, el navegador Internet Explorer (Microsoft Windows) era el más común, pero su gestión de CSS ha sido durante mucho tiempo mediocre (por no decir mala). Era el apogeo de la versión 6 (IE6), por desgracia, todavía se utiliza por una pequeña parte de los usuarios de Internet (por suerte, esta proporción tiende a disminuir).

Desde entonces, muchos navegadores se crearon y "abuchearon" a Internet Explorer: Mozilla Firefox, por supuesto, pero también Google Chrome y Opera. Y no estoy hablando del éxito de Mac y el iPhone con el navegador Safari. Esto llevó a Microsoft a responder, publicó (después de un largo período de inactividad) IE 7 e IE 8 y el IE 9. Ya se habla de IE 11.

La lección de historia, tal vez sea muy bonita, pero ¿cómo me afecta hoy en día? ¿Qué podemos aprender de esto? Que los navegadores no saben todas las propiedades CSS que existen. Cuanto el navegador es más antiguo, menos sabe de las características de CSS.

Mostraré en esta documentación una serie de características de CSS que no funcionan necesariamente en los navegadores más antiguos. No puedo evitarlo, es así: no hay ningún navegador que lo sepa todo. En el peor de los casos, si el navegador no sabe una propiedad CSS, lo ignora y no da formato, pero no se cuelga la página: siempre será legible.

Recomiendo [www.caniuse.com](http://www.caniuse.com) que ofrece tablas de características de compatibilidad de HTML y CSS en distintos navegadores (y sus diferentes versiones).

**CSS inline-block - Recommendation**

Method of displaying an element as a block while flowing it with text.

**Usage stats:** Global

Support:	93.31%
Partial support:	1.66%
Total:	94.97%

Show all versions	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	BlackBerry Browser
		3.6				3.2		2.1	
	7.0	12.0	19.0			4.0-4.1		2.2	
	8.0	13.0	20.0	5.1		4.2-4.3		2.3	
Current	9.0	14.0	21.0	6.0	12.0	5.0-5.1	5.0-7.0	4.0	7.0
Near future	10.0	15.0	22.0		12.5	6.0			10.0
Farther future		16.0	23.0						

**Notes** [Known issues](#) (0) [Resources](#) (3) [Feedback](#)

Only supported in IE6 and IE7 on elements with a display of "inline" by default. [Alternative properties](#) are available to provide complete cross-browser support.

## ¿Dónde se escribe el código CSS?

Tienes que elegir una opción porque se puede escribir código en CSS en tres lugares diferentes:

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

Voy a presentar estos tres métodos, pero ya sabemos que el primero es el mejor. En un archivo .css

Como indico, lo más habitual es escribir el código CSS en un archivo especial con la extensión css (a diferencia de los archivos HTML que tienen la extensión html). Esta es la forma más práctica y más flexible. Esto nos impide mezclar todo en un solo archivo.

Comienza a practicar ahora. Vamos a partir el archivo HTML de la siguiente manera:

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<link rel="stylesheet" href="estilo.css" />
<title>Primera prueba de CSS</title>
</head>
<body>

```

```

<h1>Mi supersitio</h1>
<p>¡Bienvenidos!</p>
<p>Por el momento está casi <em>vacío</em>.¡Ten
paciencia!</p>
</body>
</html>

```

Te darás cuenta de los contenidos de la línea 5, `<link rel="stylesheet" href="estilo.css" />`, que indica que el archivo HTML está asociado con un archivo llamado `estilo.css` y es el responsable del formato.

Guardar este archivo como se desee (por ejemplo, `pagina.html`). Por ahora, nada especial, excepto para la nueva etiqueta que hemos añadido.

A continuación, crea un nuevo archivo vacío en tu editor de textos (por ejemplo, Aptana o Bluefish) y copia ese trozo de código CSS (no te preocupes, voy a decir lo que significa):

```

p
{
color: blue;
}

```

Guarda el archivo asignándole un nombre que termine con `.css`, como por ejemplo `estilo.css`. Coloca este archivo `.css` en la misma carpeta de los `.html`.

Ahora abre tu página con el archivo `.html` en un navegador para probarlo como lo haces normalmente. Observa, los párrafos están escritos en azul, como se muestra seguidamente.

## Mi supersitio

¡Bienvenidos!  
Por el momento está casi *vacío*.¡Ten paciencia!

Otra forma de asignar formato es en el encabezado del archivo HTML `<head>`.

Hay otro método para el uso de CSS en archivos HTML: se trata de insertar el código CSS `<style>` directamente en una etiqueta en el interior de la cabecera `<head>`.

Cómo obtener exactamente el mismo resultado con un único archivo HTML que contiene el código CSS (líneas 5-10):

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
  <style>
  p
  {
  color: blue;
  }
  </style>

<title>Prueba de CSS</title>
</head>
  <body>
  <h1>Mi supersitio</h1>
  <p>¡Bienvenidos!</p>
  <p>Por el momento está casi <em>vacío</em>.¡Ten
  paciencia!</p>
  </body>
</html>

```

Y la tercera posibilidad es insertándolo directamente en las etiquetas (no recomendada).

Este último método, se ha de manejar con cuidado, puedes agregar un atributo de estilo a cualquier etiqueta. Inserta el código CSS directamente en este atributo:

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>Prueba de CSS</title>
</head>
  <body>
  <h1>Mi supersitio</h1>
  </p>
  <p style="color: blue;">¡Bienvenidos!
  <p>Por el momento está casi <em>vacío</em>.¡Ten
  paciencia!</p>
  </body>
</html>

```

Esta vez, sólo el texto del párrafo (línea 10), cuya etiqueta contiene el CSS, será de color azul.

¿Qué método elegir?

El primer método que se recomienda parece ser más complicado que los otros dos. ¿Por qué? Por qué me aconsejas crear dos archivos, en lugar de un solo archivo .html. Te recomiendo que adquieras el hábito de trabajar con el primer método, ya que es el utilizado por la mayoría de los diseñadores.

¿Por qué? Por el momento, realizas las pruebas en un único archivo HTML. Sin embargo, el sitio más tarde se compone de varias páginas HTML, estamos de acuerdo. Imagina si pones el código CSS directamente en el archivo HTML, has de copiar el código en todos los archivos HTML del sitio. Y si cambias de opinión en el futuro, por ejemplo, si quieres que los párrafos, en vez de azul aparezcan en rojo, hay que modificar cada fichero HTML.

### Aplicación de un estilo, mediante una etiqueta

Código CSS:

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

En el código CSS de esta manera, hay tres elementos diferentes:

- **Nombres de las etiquetas:** escribe los nombres de etiquetas a las que deseas cambiar la apariencia. Por ejemplo, si quiero cambiar la aparición de todos los párrafos <p>, tengo que escribir p (sin los galones de HTML).
- **Propiedades de CSS:** "efectos de estilo" en la página se almacenan las propiedades. Existe por ejemplo la propiedad color que se utiliza para especificar el color del texto, tamaño de fuente que te permite especificar el tamaño del texto, etc. Hay muchas propiedades CSS y, como te he dicho, no es necesario conocerlas todas de memoria.
- **Valores:** para cada propiedad CSS, debes especificar un valor. Por ejemplo, la propiedad de color, indica el nombre del color. Por tamaño de fuente, indica el tamaño que se desea, etc.

Esquemáticamente, una hoja de estilo CSS, se vería así:

```
etiqueta1
{
propiedad1: valor1;
propiedad2: valor2;
propiedad3: valor3;
}
```

```

etiqueta2
{
propiedad1: valor1;
propiedad2: valor2;
propiedad3: valor3;
propiedad4: valor4;
}
etiqueta3
{
propiedad1: valor1;
}

```

En este fragmento de etiquetas se ven, los atributos y valores que acabo de mencionar. Como puedes ver, se escribe el nombre de la etiqueta (por ejemplo, h1) y llaves de apertura, en el interior, pones las propiedades y los valores que desees. Se pueden indicar tantas propiedades como se desee dentro de las llaves. Cada propiedad está seguida del símbolo de "dos puntos" (:) y el valor correspondiente. Por último, cada línea termina con un punto y coma (;).

Verás muchas propiedades en los capítulos siguientes. Por el momento, en los ejemplos, solo tendremos que cambiar el color.

El código CSS que hemos utilizado hasta ahora, es:

```

p
{
color: navy;
}

```

esto significa: *"Quiero que todos mis párrafos estén escritos en azul marino"*.

Prueba a cambiar el nombre de la etiqueta asignada por CSS. Por ejemplo, si escribo h1 en el título que se escribe en azul. Edita tu archivo estilo.css de esta manera:

```

h1
{
color: navy;
}

```

Ahora, vuelve a abrir la página HTML (recuerda, esta es la página HTML que se abre en el navegador, no el archivo con extensión css): debes ver el título, que aparece en azul como se muestra seguidamente.

## Mi supersitio

¡Bienvenidos!  
Por el momento está casi *vacío*. ¡Ten paciencia!

### Aplicar un estilo a varias etiquetas

Tomemos el siguiente código CSS

```
h1
{
color: navy;
}
em
{
color: navy;
}
```

Significa que nuestros títulos y nuestros textos con las etiquetas <h1> y <em> deben mostrarse de color azul marino. Por contra, es un poco repetitiva, ¿no?. Afortunadamente, hay una manera en CSS para ir más rápido si ambas etiquetas deben tener la misma presentación. Es suficiente con combinar las declaraciones separándolas con una coma,

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

El resultado es el siguiente

## Mi supersitio

¡Bienvenidos!  
Por el momento está casi *vacío*. ¡Ten paciencia!

### Comentarios en CSS

Al igual que HTML, es posible poner comentarios. Los comentarios no se mostrarán, sólo sirven para especificar la información para ti, por ejemplo, en un archivo CSS muy largo.

Por otra parte, te darás cuenta, que en general, el archivo HTML es muy corto y el CSS



suficientemente extenso (contiene todos los elementos de estilo del sitio, es normal). Ten en cuenta que es posible crear varios archivos de CSS para tu sitio si sientes la necesidad de separar el CSS un poco (de acuerdo a las diferentes secciones de tu sitio, por ejemplo).

Así que para hacer un comentario, es muy fácil escribe `/*` seguido por tu comentario, y luego `*/` para poner fin al comentario. Los comentarios se pueden escribir en una o más líneas. Por ejemplo:

```
/*
estilo.css
-----
Por Escéptico antimagufos
*/
p
{
color: red; /* Los párrafos están en rojo */
}
```

### Aplicar un estilo: class e id

Lo que se ha mostrado hasta el momento todavía tiene un defecto: esto implica, por ejemplo, que todos los párrafos tienen la misma presentación (en este caso, por lo tanto se escribirá en azul). ¿Cómo conseguir que sólo algunos párrafos aparezcan escritos de una manera diferente? Podríamos poner el código CSS en un atributo estilo en la etiqueta que estás diseñado (que es la técnica que he descrito antes), pero como he dicho, esto no es recomendable (es mejor usar un archivo css externo).

Para resolver el problema, podemos utilizar estos atributos especiales que trabajan en todas las etiquetas:

el atributo de clase;  
el atributo id.

Los atributos clase e id son casi idénticos. Hay sólo una pequeña diferencia que voy a revelar a continuación. Por el momento, y en pocas palabras, no vamos a mirar el atributo de clase.

Es un atributo que se puede poner en cualquier etiqueta, así como párrafo, imagen, etc.

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

Sí, ¿pero cómo se da valor al atributo de la clase?

De hecho, debes escribir un nombre que identifica a la etiqueta. Lo que quieras, siempre y cuando el nombre empiece con una letra.

Por ejemplo, asociar la clase introducción a mi primer párrafo (línea 11):

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<link rel="stylesheet" href="estilo.css" />
<title>Prueba de CSS</title>
</head>

<body>
  <h1>¡Mi supersitio!</h1>
  <p>
    <p class="introduccion">¡Bienvenidos!
    <p>Por el momento está casi <em>vacío</em>.¡Ten
    paciencia!</p>
  </body>
</html>
```

Ahora lo que se ha hecho, es que el párrafo esté identificado. Tiene un nombre: introducción. Serás capaz de volver a utilizar el nombre en el archivo CSS para decir: "quiero que las etiquetas que tienen nombres como" introducción "se muestren en azul."

Para hacer esto en CSS, especifica el nombre de la clase, a partir de un punto (.), de la siguiente manera:

```
.introduccion
{
color: navy;
}
```

Apareciendo al ejecutarlo

## Mi supersitio

¡Bienvenidos!

Por el momento está casi *vacío*.¡Ten paciencia!

## ¿Y entonces el atributo id?

Funciona exactamente de la misma manera que la clase, excepto por una cosa: que su denominación sólo puede utilizarse una vez en el código.

¿Qué interés? Hay poco para ser honesto, será útil si luego se usa JavaScript para reconocer ciertas etiquetas. Por otra parte, ya hemos visto el atributo id en el capítulo sobre los enlaces (para hacer los anclajes). En la práctica, vamos a poner esa Identificación en los elementos que son únicos dentro de la página, tales como el logotipo.

```

```

Si utilizas la Identificación, cuando se definen las propiedades en el archivo css, se precederá el nombre de la Identificación por el signo sostenido (#):

```
#logo
{
/* Indica las propiedades CSS aquí */
}
```

## Etiquetas universales

Llegará el momento en que se necesite aplicar una clase (o id) a ciertas palabras que, originalmente, no están rodeadas por etiquetas.

En efecto, el problema de la clase es que es un atributo. Entonces no se puede poner en una etiqueta. Si, por ejemplo, quiero cambiar sólo "bienvenido" en el párrafo: <p>¡Buenos días y bienvenido a mi sitio!</p>.

Sería fácil de hacer si hay una etiqueta alrededor de "bienvenido", pero por desgracia no hay ninguna. Por suerte, existe la etiqueta que no sirve para nada.

De hecho, hemos inventado dos etiquetas de las llamadas universales, sin un significado especial (no indican que la palabra es importante, por ejemplo). Hay una diferencia mínima (pero importante) entre estas dos etiquetas:

- **<span> </span>**: esta etiqueta es de línea, es decir que es una etiqueta se coloca dentro de un párrafo de texto, para seleccionar sólo algunas palabras. Las etiquetas <em> y <strong> son de la misma familia. Esta etiqueta, por lo tanto, se utiliza en medio de un párrafo y así la vamos a utilizar para dar color a "bienvenida".

- **<div> </div>**: esta es una etiqueta del tipo bloque, en torno a un bloque de texto. Las

etiquetas <p>, <h1>, etc. son de la misma familia. Estas etiquetas tienen algo en común: crear un nuevo "bloque" en la página y generar por lo tanto, necesariamente un salto de línea. <div> es una etiqueta que es utilizada con frecuencia en la construcción de un diseño, como veremos más adelante.

Así que por ahora, vamos a utilizar la etiqueta <span>. Se pone alrededor de "bienvenida", añadiendo una clase (con el nombre que quieras).

Código HTML:

```
<p>¡Buenos días y <span class="saludos">bienvenido a</span> mi
sitio!¿</p>
```

y el código CSS:

```
.saludos
{
color: blue;
}
```

### Aplicar un estilo: selectores avanzados

En CSS, la parte más difícil es conocer el texto de destino al que se desea cambiar la forma. Para hacer referencia (o seleccionar) los elementos de la página a modificar, usamos lo que se llaman selectores. Has usado alguno anteriormente en este capítulo, un resumen de ellos para empezar.

Selectores que ya conoces:

Estos selectores, como hemos visto anteriormente, son en gran medida los más comúnmente utilizados. Se deben saber de memoria.

Comienza con la base de la base:

```
p
{
}
```

quiere decir "quiero afectar a cada párrafo." Después de eso, te toca decir lo que hacer en esos párrafos (escribir en rojo, por ejemplo).

También hemos visto:

```
h1, em
{ }
```

que significa "Todos los títulos y todos los textos importantes". Hemos seleccionado dos etiquetas a la vez.

Y, por último, vimos cómo seleccionar etiquetas específicas a las que hemos dado un nombre con los atributos clase e id:

```
.class
{
}
#id
{
}
```

Hay otras formas de poner etiquetas en CSS. No vamos a ver todas porque hay muchas y algunas son complejas, pero estas son las que puede ser más eficaces en CSS!

### Selectores avanzados

\*: selector universal

```
*
{
}
```

Selecciona todas las etiquetas sin excepción. Se le llama el selector universal.

A B: una etiqueta dentro de otra

```
h3 em
{
}
```

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

Código HTML equivalente:

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

A + B : una etiqueta que sigue a otra

```
h3 + p
{
}
```

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

Ejemplo de código HTML

```
<h3>Título</h3>
<p>Párrafo</p>
```

A [atributo]: una etiqueta que posee un atributo

```
a[titulo]
{
}
```

Selecciona todos los enlaces <a> que poseen un atributo título.

Ejemplo de código HTML

```
<a href="http://www.esceptiismo.es" title="Escepticismo">
```

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

```
a[titulo="Pulse aquí"]
{
}
```

Lo mismo, pero el atributo también debe tener exactamente el valor "Pulse aquí".

Ejemplo de código HTML

```
<a href="http://www.escepticismo.es" title="Pulse aquí">
```

A[atributo\* = "valor"]: una etiqueta, un atributo y un valor.

```
a[titulo*="aquí"]
{
}
```

Lo mismo ocurre, pero el atributo debe contener en su valor la palabra "aquí" (cualquier posición).

Ejemplo de código HTML

```
<a href="http://www.escepticismo.es" title="Cualquier parte por aquí">
```

### Existen otros selectores

Aquí se ha presentado una parte de los selectores CSS, pero sabemos que hay muchos otros. Si deseas una lista completa, puedes acceder directamente a la fuente: el sitio del [W3C](#). Es muy completo. Otros selectores se considerarán más adelante.

### En resumen

- CSS es otro lenguaje que complementa a HTML. Su función consiste en dar formato a las páginas web.
- Se ha de tener cuidado con la compatibilidad del navegador con algunas características recientes de CSS3.
- Cuando un navegador no tiene conocimiento de un formato de instrucción, simplemente lo ignora.
- Podemos escribir el código CSS en varios lugares diferentes, lo más aconsejable es crear un archivo separado con extensión .css (ejemplo: estilo.css).
- En CSS, seleccionamos qué partes de la página HTML se desean cambiar y modificar su presentación con propiedades de CSS:
- Hay muchas maneras de seleccionar la parte de la página a la que se desea dar formato. Por ejemplo:
  - todas las etiquetas del mismo tipo, escribiendo su nombre (por ejemplo, h1);
  - algunas etiquetas específicas, que se le han dado nombres mediante el uso de los atributos class o id (.nombreclase o #nomid);
  - únicamente etiquetas que están dentro de otras etiquetas (h3 em).
  - etc.

## Formato de texto

Llegamos ahora a un apartado que te interesará mucho. Simplemente significa que vamos a cambiar la apariencia del texto.

No es una sorpresa especial: todavía estamos en CSS y usamos lo que hemos aprendido en el capítulo anterior. Por lo tanto, vas a trabajar directamente en el archivo .css que hemos creado previamente.

Este capítulo será una oportunidad para descubrir muchas de las propiedades CSS: vamos a ver cómo cambiar el tamaño del texto, cambiar la fuente, alinear el texto ...

### Tamaño del texto

Para cambiar el tamaño del texto, utiliza la propiedad CSS font-size. Pero, ¿cómo especificar el tamaño de la fuente? Aquí es donde las cosas se ponen difíciles, porque hay muchas técnicas disponibles:

Indicar un tamaño absoluto: en píxeles, centímetros o milímetros. Este método es muy preciso, pero es aconsejable utilizarlo sólo si es absolutamente necesario, porque puede indicar un tamaño demasiado pequeño para algunos lectores.

Indicar un tamaño relativo en tanto por ciento, <em> o <ex>, esta técnica tiene la ventaja de ser más flexible. lo que resulta más fácil a las preferencias de tamaño de los visitantes.

**Tamaño absoluto.** Para especificar un tamaño absoluto, se utilizan generalmente los píxeles. Para un texto de 16 píxeles de alto, escribe:

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

He aquí un ejemplo de uso, puedes copiar el código en el fichero estilo.css:

```
p
{
font-size: 14px; /* Párrafo de 14 pixeles */
}
h1
{
font-size: 40px; /* Títulos de 40 pixeles */
}
```



Si lo deseas, los tamaños se pueden definir en cm o mm (Sistema Internacional de Unidades, SI), reemplazando px por cm o mm. Sin embargo aunque son obligatorias legalmente, estas unidades se adaptan peor a las pantallas.

## Valor relativo

Es el método recomendado pues el texto se adapta más fácilmente a las preferencias de todos los visitantes. Hay varias formas de indicar un valor relativo, por ejemplo se puede indicar el tamaño con palabras en inglés como:

xx-small : minúscula ;  
x-small : muy pequeña;  
small : pequeña ;  
medium : media ;  
large : grande ;  
x-large : muy grande;  
xx-large : gigantesca;

Es fácil probar el código

```
p
{
font-size: small;
}
h1
{
font-size: large;
}
```

Esta técnica tiene un inconveniente: hay siete tamaños disponibles (porque solo hay siete nombres). Afortunadamente, hay otros, es habitual especificar el tamaño en "em". Si escribes 1em, el texto tiene un tamaño normal. Si deseas ampliar el texto, puede introducir un valor mayor que 1, como 1.3em. Si deseas reducir el texto, introduce un valor inferior a 1, como 0.8em.

Ten cuidado: para los números decimales, debes poner un punto y no una coma. Tendrás que escribir "1.4em" y no "1,4em"

Ejemplo de código CSS:

```
p
```

```

{
font-size: 0.8em;
}
h1
{
font-size: 1.3em;
}

```

Otras unidades están disponibles. Puedes probar el "ex" (que funciona con el mismo principio que em pero significa base más pequeña) y el porcentaje (80%, 130% ...).

## Fuentes

De hecho, hay un problema: para que un tipo de letra se visualice correctamente, se requiere que todos los usuarios que lo visiten tengan la fuente instalada en su sistema operativo. Si un usuario no tiene la misma fuente que supones como fuente predeterminada del navegador (tipo de letra estándar) lo que se muestra puede que no tenga nada que ver con lo esperado.

La buena noticia es que desde CSS3, es posible descargar automáticamente en un navegador la fuente. Explicaré en un segundo paso cómo hacer esto.

### Cambiar la fuente

La propiedad CSS que permite especificar el tipo de letra a utilizar es font-family. Debes escribir el nombre de la fuente así:

```

etiqueta
{
font-family: fuente;
}

```

Sólo para evitar problemas si el usuario no tiene la misma fuente que se precisa, en general se indican los nombres de diferentes fuentes, separados por comas:

```

etiqueta
{
font-family: fuente1, fuente2, fuente3, fuente4;
}

```

El navegador intentará utilizar la fuente1. Si no es así, intentará la fuente2. Si no lo hace, se irá a la fuente3, y así sucesivamente.

En general, se indica la última como serif, que corresponde a una fuente predeterminada (que

sólo se aplica si no se ha encontrado otra fuente). También hay otro tipo de letra por defecto llamado sans-serif. La diferencia entre los dos es la presencia de pequeñas lengüetas de conexión en la parte inferior de las letras, la fuente sans serif no tiene. Sí, es sutil.

Sí, pero ¿cuáles son las fuentes más comunes que tienes el "derecho" a utilizar

Arial ;  
 Arial Black ;  
 Comic Sans MS ; (no recomendada)  
 Courier New ;  
 Georgia ;  
 Impact ;  
 Times New Roman ;  
 Trebuchet MS ;  
 Verdana.

Así, si escribo el siguiente código CSS

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

significa: "pon la fuente impact si no es así, "Arial Black", o si no Arial, o si no Verdana o si nada funciona, poner un tipo de letra estándar (sans-serif).

En general, se indica una selección de tres o cuatro fuentes (+ serif o sans-serif) para asegurarse de que al menos una de ellas ha sido encontrada en el ordenador del visitante de nuestra web.

Si el nombre de la fuente lleva espacios en blanco, entonces se ha de poner entre comillas, como por ejemplo "Arial Black".

### Uso de una fuente personalizada con @font-face

Si te parece la elección de fuentes demasiado pequeña. ¿Cómo puedo usar mi fuente favorita de mi sitio web? Durante mucho tiempo esto no era posible. Actualmente, con CSS3, hay afortunadamente una manera de utilizar cualquier fuente en tu web. Esto funciona bien con la mayoría de los navegadores. Pero cuidado, hay defectos (de lo contrario sería demasiado bueno):

- Será necesario que el navegador de los visitantes descargue automáticamente el archivo de la fuente, que puede "pesar" o exceder de 1MB.

- La mayoría de las fuentes están sujetas a derechos de autor, no es legal utilizarlas en tu sitio. Afortunadamente, hay sitios como [dafont.com](http://dafont.com) y [fontquirrel.com](http://fontquirrel.com) que ofrecen la posibilidad de descargar fuentes libres de derechos de autor. Recomendamos especialmente fontquirrel.com, ya que permite que descargues los paquetes listos para CSS3.

- Hay varios formatos de archivos de fuentes y no funcionan en todos los navegadores, necesitas saber:

- . ttf: *Font TrueType*. Funciona en IE9 y los demás navegadores.
- . eot: *Embedded OpenType*. Sólo funciona en todas las versiones de Internet Explorer. Este formato es propietario, producido por Microsoft.
- . tfo: *Font OpenType*. No funciona en Internet Explorer.
- . svg: *SVG Font*. El único formato compatible con iPhones y iPads por ahora.
- . woff: *Web Open Font Format*. Nuevo formato diseñado para la web, que funciona en IE9 y las demás navegadores.

En CSS para definir un nuevo tipo de letra o fuente, debes declararlo así:

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

El archivo de fuente (aquí MiFuenteEsceptica.eot) debe estar ubicado en la misma carpeta que el archivo CSS (o en una subcarpeta, si se utiliza una ruta relativa).

Sí, además EOT sólo funciona en Internet Explorer. Lo ideal es ofrecer múltiples formatos para las fuentes que el navegador descarga y que pueda leer. Se indican como varios formatos:

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

Para probar, sugiero descargar una fuente de Squirrel Font, por ejemplo, Learning Curve Pro. Haz clic en "@ font-face Kit" que permitirá descargar una lista para su uso con todos los formatos de la fuente.

El archivo CSS se verá al final como:

```

@font-face { /* Definición de una fuente 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 /* Utilización de la fuente definida en los títulos*/
{
font-family: 'LearningCurveProRegular', Arial, serif;
}

```

La primera sección `@font-face` sirve para definir un nuevo nombre de tipo de letra que se puede utilizar en el archivo CSS. A continuación, utilizamos este nombre de la fuente con la propiedad `font-family`, que ya sabemos, para cambiar la apariencia de los títulos `<h1>`. Se puede ver una muestra de esta fuente a continuación.

abcdefghijklm

Te darás cuenta de algunas peculiaridades en el CSS generado por el tipo de letra Squirrel. El objetivo es superar algunos errores de Internet Explorer porque las versiones anteriores no entienden si definimos varios formatos. Eso explica la presencia de un `?#iefix` en el código.

### **Cursiva, negrita, subrayado ...**

Hay un número de propiedades CSS de formato de texto de forma convencional. Vamos a descubrir aquí cómo mostrar el texto en negrita, cursiva y subrayado ... y de paso veremos que incluso es posible hasta el parpadeo.

#### **Cursiva**

Se piensa que la etiqueta `<em>` permite poner texto en cursiva. Nunca dije eso. Vuelve al capítulo anterior, si tienes dudas, pero nunca he dicho que la etiqueta `<em>` se hizo para poner texto en cursiva (como yo nunca he dicho que `<strong>` se hizo para negrita).

`<em>`, sirve para hacer hincapié en las palabras. Esto significa que rodea a las palabras que son bastante importantes. Para representar esta importancia, la mayoría de los navegadores optan por mostrar el texto en cursiva, pero esto no es obligatorio. CSS puede decir: "Quiero que este texto esté en cursiva." Nada lo impide, por ejemplo, para decidir que todos tus títulos irán en cursiva.

En concreto, en CSS, para poner en cursiva, se utiliza `font-style`, que puede tomar tres valores:

- `italic`: el texto en cursiva.
- `oblique`: el texto aparece oblicuamente (si las letras son examinadas, el resultado es ligeramente diferente de la cursiva).
- `normal`: el texto será normal (por defecto). Esto permite cancelar la cursiva. Por ejemplo, si deseas que los textos entre `<em>` ya no estén en cursiva, se debe escribir:

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

Por lo tanto, en el siguiente ejemplo, uso `font-style` poner en cursiva todos mis títulos `<h2>`:

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

## Negrita

Así que de nuevo, recuerda que `<strong>` no significa negrita (su función es indicar que el texto es importante, por lo general, el navegador lo mostrará en negrita). La negrita en CSS se puede aplicar a diversos valores, algunos párrafos enteros, etc. Todo depende de ti.

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

- `bold`: el texto aparecerá en negrita;
- `normal`: el texto será escrito normalmente (por defecto).

Por ejemplo, la forma de escribir los títulos en negrita:

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

## Subrayado y otras decoraciones

La propiedad CSS asocia un nombre apropiado a: text-decoration. Permite, entre otros, resaltar el texto, pero no sólo. Diferentes valores de que puede tomar:

underline: subrayado.  
line-through: cruzado.  
overline: línea encima.  
blink: Intermitente. No funciona en todos los navegadores.  
none: normal (por defecto).

Este código en CSS permitirá probar los efectos de text-decoration:

```
h1
{
text-decoration: blink;
}
.subrayado
{
text-decoration: underline;
}
.cruzado
{
text-decoration: line-through;
}
.linea_superior
{
text-decoration: overline;
}
```

Y por ejemplo, con el siguiente código en HTML,

```
<!DOCTYPE HTML>
<html>

<head>
<link rel="stylesheet" href="estilo.css" />
<title> Ejemplo css</title>
<meta name="Generator" content="Quanta + 3.5">
<meta name="Description" content="Ejemplo de atributos a caracteres
mediante text-decoration">
<meta name="Keywords" content="css, ejemplo">
```

```

</head>

<body>
Prueba de CSS<br/><br/>

<h1> La propiedad css text-decoration permite<br/> decorar un poco el
texto.</h1><br/><br/>

<span class="subrayado">subray&aacute;ndolo</span><br/>
<span class="cruzado">cruz&aacute;ndolo</span><br/>
<span class="linea_superior">... o poniendo una l&aacute;nea por
encima</span>

</body>
</html>

```

se tendría:

## La propiedad css text-decoration permite decorar un poco el texto.

subrayándolo  
cruzándolo  
... o poniendo una línea por encima

### Alineación

CSS nos permite todas las alineaciones: izquierda, centro, derecha o justificada.

Es muy sencillo. Usamos la propiedad text-align y se indica la alineación deseada:

- left, a la izquierda: el texto está alineado a la izquierda (por defecto).
- center, el texto se centrará.
- right, el texto está alineado a la derecha.
- justify, el texto está "justificado". Justificar el texto puede usarse para asegurarse de que se ocupa todo el ancho posible sin dejar espacios en blanco al final de las líneas. Los artículos de periódicos, por ejemplo, están habitualmente justificados.

En el siguiente ejemplo se muestran las distintas posibilidades:



```

h1
{
text-align: center;
}
p
{
text-align: justify;
}

.signature
{
text-align: right;
}

```

Y por ejemplo con el siguiente código html,

```

<head>
<link rel="stylesheet" href="estilo3.css" />
<title> Ejemplo css</title>
<meta name="Generator" content="Quanta + 3.5">
<meta name="Description" content="Ejemplo de alineación">
<meta name="Keywords" content="css, ejemplo, alineación">
</head>

<body>

<h1>La Educación Científica No es
Garantía de Escepticismo</h1><br/>

<p>Muchos científicos asumen cierta medida de placer
en el tipo de tareas a menudo puestas delante de ellos:
evaluación de fotografías borrosas,
realización de experimentos de laboratorio que reducen o
eliminan el engaño, crítica de la ciencia imperfecta y
la pseudociencia, y contrarrestando las afirmaciones de evidentes
charlatanes. Por supuesto, los científicos esperan que su
esfuerzo ayude al avance de la educación
científica [1]. A pesar de estos esfuerzos, los datos de
encuestas de diversas fuentes sugieren que la creencia en lo
paranormal y el pensamiento pseudocientífico siguen
siendo comunes [2].<br/><br/>

```

Los escépticos suelen utilizar estos hallazgos para reforzar los argumentos a favor de más educación científica. Su argumento se basa en la suposición no comprobada de que un mayor conocimiento científico reduce el número de creencias paranormales que un individuo posee. Sin embargo, esta suposición puede no ser válida. Andrew Ede sostuvo recientemente que la educación científica puede hacer poco para elevar el nivel de pensamiento racional y, de hecho, en realidad lo desalienta! [3]. Recientes debates sobre la inclusión de la "ciencia de la creación" y/o eliminación de la evolución del currículo de biología en la enseñanza secundaria [4], son un ejemplo de ello, lo que indica que muchos legisladores, miembros del público, y algunos educadores están confundidos acerca de cómo criticar y comparar teorías con el fin de separar los hechos de las creencias. Ede ha identificado tres razones de porque esto puede ser cierto [5]:

Chile Skeptic

Se tendría

### La Educación Científica No es Garantía de Escepticismo


Muchos escépticos sienten cierta medida de placer en el tipo de tareas a menudo puestas delante de ellos: evaluación de fotografías borrosas, realización de experimentos de laboratorio que reducen o eliminan el engaño, crítica de la ciencia imperfecta y la pseudociencia, y contrarrestando las afirmaciones de evocados charlatanes. Por supuesto, los escépticos esperan que su esfuerzo ayude al avance de la educación científica [1]. A pesar de estos esfuerzos, los datos de encuestas de diversos países sugieren que la creencia en lo paranormal y el pensamiento pseudocientífico siguen siendo comunes [2].

Los escépticos suelen utilizar estos hallazgos para reforzar los argumentos a favor de más educación científica. Su argumento se basa en la suposición no comprobada de que un mayor conocimiento científico reduce el número de creencias paranormales que un individuo posee. Sin embargo, esta suposición puede no ser válida. Andrew Ede sostuvo recientemente que la educación científica puede hacer poco para elevar el nivel de pensamiento racional y, de hecho, en realidad lo desalienta! [3]. Recientes debates sobre la inclusión de la "ciencia de la creación" y/o eliminación de la evolución del currículo de biología en la enseñanza secundaria [4], son un ejemplo de ello, lo que indica que muchos legisladores, miembros del público, y algunos educadores están confundidos acerca de cómo criticar y comparar teorías con el fin de separar los hechos de las creencias. Ede ha identificado tres razones de porque esto puede ser cierto [5].

Chile Skeptic

No se puede cambiar la alineación del texto de una etiqueta *inline* (como `<span>`, `<a>`, `<em>`, `<strong>` ...). La alineación sólo funciona en las etiquetas de tipo bloque (`<p>`, `<div>`, `<h1>`, `<h2>`, ...) y tiene un poco de lógica, cuando piensas: no se puede cambiar la alineación de unas pocas palabras en medio de un párrafo. Por lo general es todo el párrafo el que se debe alinear.

Seguidamente se indica cómo lograr el ejemplo mostrado a continuación



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.

Supongo que ahora la pregunta que te haces es: "¿pero cuál es la propiedad mágica que hace que flote?". La respuesta es ... float ("flotar" en Inglés). Esta propiedad puede tener dos valores simples:

left: el elemento flotará a la izquierda.  
right: el elemento flotará a la derecha.

El uso de float es muy sencillo:

1. Aplica float a una etiqueta.
2. A continuación, sigue escribiendo texto de forma normal.

Puedes utilizar la propiedad float en etiquetas block e inline. Es común hacer flotar una imagen para ser rodeada por el texto, como en el ejemplo anterior.

### Hacer flotar una imagen

Aquí vamos a aprender a flotar una imagen. El código HTML que debemos escribir:

```
<p> Este es un texto normal de párrafo, escrito a continuación de una imagen y en el que aparecerá la imagen flotando.</p>
```

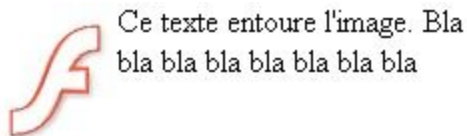
El código CSS es:

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

## Detener una imagen flotante

Cuando se implementa un texto flotante, fluye. Pero si lo que quieres es que después de un tiempo, el texto continúe por debajo de la flotación, se podrían encadenar varios `<br />` pero no es ni elegante ni muy propio.

Básicamente, a uno le gustaría obtener el mismo resultado que en la siguiente imagen.



Texte écrit en-dessous

En realidad, hay una propiedad CSS que nos permite decir: alto, este texto debe estar por debajo del float y no al lado, es la propiedad `clear`, que puede tomar tres valores:

- `left`: el texto continúa más abajo después de un float: `left`;
- `right`: el texto continúa más abajo después de un float: `right`;
- `both`: el texto sigue más adelante, ya sea después de un float: `left`; o después de un float: `right`;

Para simplificar, vamos a utilizar todo el tiempo `clear: both`, que trabaja después de un float a la izquierda justo después de un float a la derecha. (Así funciona todo el tiempo). Para ilustrar cómo funciona, vamos a ver el código HTML:

```
<p></p>
<p>Este texto está escrito a la derecha de la imagen flotante.</p>
<p class="debajo">Este texto está escrito bajo la Imagen flotante.</p>
```

Y en código CCS

```
.imagenflotante
{
float: left;
}
.debajo
{
clear: both;
}
```

Y eso es todo. Aplicamos `clear: both`; al párrafo que queremos ver bajo la imagen flotante.

**Resumen:**

- Cambiamos el tamaño del texto con la propiedad CSS font-size. Se puede especificar el tamaño en píxeles (16px) en 'em', (1.3em), por ciento (110%), etc.
- Cambiamos el tipo de letra con font-family. Atención, sólo algunas fuentes son conocidas por todos los equipos. Sin embargo, se puede utilizar una fuente personalizada con @font-face: esto hará que el navegador descargue la fuente de tu elección.
- Existen muchas propiedades de formateo de texto: font-style para cursiva, font-weight para negrita, text-decoration para subrayado, etc.
- El texto se puede alinear con text-align.
- Puedes hacer que una imagen se vista ("rodee") de texto con float.

## Color y fondo

Continuamos con nuestra visión general de las propiedades CSS. Nos centraremos aquí en las propiedades relacionadas directamente con:

- cómo cambiar el color del texto;
- cómo poner un color o imagen de fondo;
- cómo añadir sombras;
- cómo jugar con los niveles de transparencia.

### Color del texto

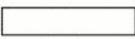




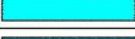
Ahora, pasamos al amplio tema del color. Ya conoces la propiedad que te permite cambiar el color del texto: es color. Veremos distintas maneras de especificar el color, ya que hay varias.

### Indicar el nombre del color

La forma más sencilla y cómoda de elegir un color, basta con escribir su nombre (en Inglés, por supuesto).

El único inconveniente de este método es que sólo hay dieciséis colores llamados "estándar". Otros colores son informales pero, no necesariamente funcionan de la misma manera en todos los navegadores, voy a evitar mostrarlos.

La siguiente figura muestra los dieciséis colores que se pueden utilizar simplemente escribiendo su nombre.

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

Puedes aprenderlos de memoria, si quieres. Para poner todos los títulos en rojo, podemos escribir:

En CSS:

```
h1
{
color: red;
}
```

## Notación hexadecimal

Dieciséis colores, es todavía un poco limitado si se considera que la mayoría de los monitores actuales pueden mostrar dieciséis millones.

Por otra parte, si tuviéramos que dar un nombre a cada uno de los dieciséis millones de colores. Afortunadamente, hay varias maneras de elegir un color CSS de todos los existentes. La primera vez que se va a mostrar es la notación hexadecimal. Se utiliza comúnmente en la web pero también hay otro método que veremos más adelante.

Un nombre de color en hexadecimal, es así: #FF5A28. En pocas palabras, se trata de una

combinación de letras y números indicando un color. Siempre hay que empezar por escribir un sostenido (#) seguido de seis letras o dígitos del 0 al 9 y de A a F. Estas letras o números operan en parejas. Los dos primeros dígitos indican la cantidad de rojo, los dos siguientes la cantidad de verde y los dos últimos la cantidad de azul. Mediante la mezcla de estas cantidades (que son componentes rojo, verde y azul) se puede obtener el color deseado.

Así, #000000 es el color negro #FFFFFF el blanco. Pero ahora, no me preguntes por qué hay una combinación que produce el color naranja "puesta de sol".

Por ejemplo, ¿cómo se aplicará color blanco a los párrafos en hexadecimal?:

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

### Método RGB

¿Qué significa RGB? En inglés, *Red-Green-Blue* esto es Rojo-Verde-Azul, que se abrevia como "RGB". Al igual que con la notación hexadecimal, si eliges un color, debes definir una cantidad de rojo, verde y azul.

Se verá que es mucho más cómodo y con un sencillo programa de dibujo, encontrar el color que desees, e incluso hay utilidades en línea, muy potentes para gestión del color.

Introduce los valores en este orden en el archivo CSS, como en el siguiente código:

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

### Color de fondo

Para especificar un color de fondo, utiliza el código CSS background-color. Se emplea de la misma manera que la propiedad color, es decir, puedes escribir el nombre de un color, en notación hexadecimal o utilizar el método RGB.

Para especificar el color de fondo de la página web, es necesario trabajar en <body>, que corresponde a toda la página web, así que esto es cambiando el color de fondo se cambia el color de fondo de la página.

Código en CSS:



```

/* Opera sobre la etiqueta <body>, sobre toda la página*/
body
{
background-color: black; /* el fondo de la página será negro*/
color: white; /* el texto de la página será blanco*/
}

```

## CSS y herencia

En CSS, si aplicas un estilo a una etiqueta, todas las etiquetas que están en el interior deben adoptar el mismo estilo.

De hecho, es fácil de entender e intuitiva. La etiqueta <body> contiene entre otras las etiquetas de párrafo <p> y <h1>.

Si aplico un color de fondo negro y color de texto blanco en la etiqueta <body> todos mis títulos y párrafos también tendrán un fondo negro y texto en blanco. Este fenómeno se denomina herencia se dice de las etiquetas que se encuentran dentro de otra etiqueta "heredan" sus propiedades.

Aquí también es donde obtenemos el nombre de "CSS", que significa "*Cascading Style Sheets*", es decir "Hojas de estilo en cascada". Las propiedades CSS se heredan en cascada: si se le da un estilo a un elemento, todos los subelementos tienen el mismo estilo.

¿Esto significa que todo el texto de mi página web será necesariamente escrito en blanco? No, no necesariamente. Si dices más adelante que quieres que los títulos aparezcan en rojo, este estilo y sus acciones tendrán prioridad, por lo tanto estará en rojo. Sin embargo, si no se especifica nada en particular (como se hizo anteriormente), entonces sus acciones heredan el color blanco.

Esto funciona no sólo para el color, que quede claro. Todas las propiedades CSS se heredan. Por ejemplo, puedes poner negrita en la etiqueta <body> y todos los títulos y los párrafos estarán en negrita.

### Ejemplo de herencia con la etiqueta <mark>

Tendemos a creer que sólo podemos cambiar el color de fondo de la página. No es cierto: se puede cambiar el fondo de cualquier elemento: títulos, párrafos, algunas palabras ... En este caso, aparecerá resaltado (como si usáramos un marcador por encima).

¿Recuerdas que por ejemplo, la etiqueta <mark> puede resaltar algunas palabras? Utilízala de nuevo aquí:

Código en HTML:

```
<h1> ¿Quién apagó las luces? </ h1>
<p> Todo es negro en este sitio, que es un poco
<mark> inquietante </ mark> como la atmósfera ¿no te parece? </ p>
```

Código en CCS:

```
body
{
background-color: black;
color: white;
}
mark
{
/*El color de fondo tiene prioridad sobre el de toda la página*/
background-color: red;
}
```

En el texto de la etiqueta <mark> se aplica el color de fondo rojo. Incluso si el fondo de la página es negro, es la propiedad del CSS del elemento más preciso el que tiene prioridad, por lo que la palabra a la que se aplica mark aparecería con fondo rojo.

El mismo principio se aplica a todas las etiquetas HTML y propiedades CSS. Si dices: mis párrafos tienen un tamaño de 1.2em; mis documentos importantes (<strong>) tienen un tamaño de 1.4em; uno podría pensar que hay un conflicto.

Si el texto importante es parte de un párrafo, ¿qué tamaño le da? 1.2em o 1.4em? CSS determina que es la declaración más precisa prevalece: como <strong> es más específico que párrafos, el texto será escrito en 1.4 em.

## Imágenes de fondo

En los ejemplos que siguen, voy a cambiar la imagen de fondo de la página. Sin embargo, como con el color de fondo, recordar que la imagen de fondo no se aplica necesariamente a toda la página. También puedes poner una imagen de fondo detrás de títulos, párrafos, etc.

Aplicar una imagen de fondo

La propiedad para especificar una imagen de fondo es background-image. Como valor, debemos indicar url ("nombre\_de\_la\_imagen.png"). Por ejemplo:

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

Opciones disponibles para la imagen de fondo

Podemos completar la imagen de fondo, hemos visto otras varias otras propiedades que permiten cambiar el comportamiento de la imagen de fondo.

- background-attachment: fija el fondo

El código CSS background-attachment permite fijar el fondo. El efecto es interesante porque vemos a continuación, el texto "deslizarse" sobre el fondo. Dos parámetros están disponibles:

fixed : la imagen de fondo se mantiene fija;

scroll : la imagen de fondo desfila con el texto (por defecto).

Código CSS

```
body
{
background-image: url("nube.png");
background-attachment: fixed; /* El fondo quedará fijo*/
}
```

- background-repeat : repetición del fondo

De forma predeterminada, la imagen de fondo se repite en el mosaico. Puedes cambiar esto con el background-repeat:

no-repeat: el fondo no se repite. La imagen será única en la página.

repeat-x: el fondo se repetirá sólo en la primera línea, horizontalmente.

repeat-y: el fondo se repetirá sólo en la primera columna, en posición vertical.

repeat: el fondo se repetirá en mosaico (por defecto).

Ejemplo de uso:

Código: CSS

```
body
{
background-image: url("nubes.png");
background-repeat: no-repeat;
}
```

- background-position : posición de fondo

Se puede indicar dónde se encuentra la imagen de fondo con background-position. Esta propiedad es interesante sólo si se combina con background-repeat: no-repeat; (un fondo que no se repite).

Debes dar a background-position dos valores de los píxeles para indicar la posición relativa a la esquina inferior y parte superior izquierda de la página (o en el párrafo, si se aplica el fondo de un párrafo). Así que si escribes:

Código: CSS

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

tu fondo se colocará 30 píxeles desde la izquierda y 50 píxeles desde la parte superior. También es posible utilizar estos valores en Inglés:

```
top: parte superior;
bottom: parte inferior;
left: izquierda;
center: centro;
right: derecha.
```

Es posible combinar estas palabras. Por ejemplo, para alinear una imagen en la parte superior derecha, tendrías que escribir:

Código: CSS

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

Así que si voy a poner un sol en el fondo (figura siguiente), en una sola muestra (no-repeat), siempre visible (fixed) y se coloca en la parte superior derecha (top right), voy a escribir esto:

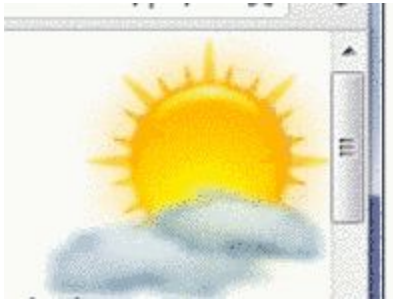
Código: CSS

```

body
{
background-image: url("sol.png");
background-attachment: fixed; /* El fondo quedará fijo*/
background-repeat: no-repeat; /* El fondo no se repetirá */
background-position: top right; /* El fondo se colocará encima a la
derecha */
}

```

Apareciendo el sol de la siguiente forma:



### Combinar propiedades

Si utilizas una gran cantidad de propiedades relacionadas con el fondo (como es el caso de este último ejemplo), puedes utilizar una especie de "super-propiedad" de fondo, cuyo valor puede combinar varias propiedades vistas previamente: `background-image`, `background-repeat`, `background-attachment` y `background-position`.

Por tanto, podemos escribir simplemente:

Código CSS

```

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

```

Esta es la primera "super-propiedad" tal como indiqué, habrá otras. Debes saber que:

El orden de los valores no importa.

Se pueden combinar los valores en cualquier orden.

No son obligatorios.

Así que si no deseas escribir `fixed`, se puede quitar sin ningún problema.

## Varias imágenes de fondo

Desde CSS3, es posible dar varias imágenes de fondo a un elemento. Para hacer esto, simplemente usa declaraciones separadas por una coma, así:

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

La primera imagen de esta lista se coloca encima de la otra. Así que ten cuidado, el orden de declaración. La imagen es importante: si se invierte el sol y la nieve en el código CSS anteriormente, no verás el sol.



Ten en cuenta que varias imágenes de fondo funcionan en todos los navegadores excepto en las versiones anteriores de Internet Explorer, que no reconoce esta característica hasta la versión 9 (IE9).

## Transparencias

CSS permite jugar fácilmente con los niveles de transparencia de los elementos. Para ello utilizamos características CSS3: la propiedad de opacity y notación RGBA.

Propiedad de opacidad

La propiedad opacity, es muy simple para especificar el nivel de opacidad (es decir, la inversa de la transparencia).

Con un valor de 1, el elemento estará completamente opaco: este es el comportamiento por defecto. Con un valor 0, el elemento será completamente transparente. Así que selecciona un valor entre 0 y 1. Así, con un valor de 0.6, será el 60% opaco.

¿Cómo podemos utilizarla?

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

Un ejemplo que nos permitirá apreciar la transparencia.

```
body
{
  background: url('nubes.png');
}
p
{
  background-color: black;
  color: white;
  opacity: 0.3;
}
```

Notación RGBA

CSS3 ofrece otra manera de jugar con la transparencia: la notación RGBA. Esto es en realidad la notación RGB que hemos visto antes, pero con un cuarto parámetro: el nivel de transparencia (llamado "alfa"). Como anteriormente, con un valor de 1, el fondo es completamente opaco. Con un valor menor que 1, es transparente.

Código: CSS

```
p
{
  background-color: rgba(255, 0, 0, 0.5); /* Fondo rojo medio transparente */
}
```

Es tan simple como eso. Puedes conseguir el mismo efecto que con opacity jugando con la notación opacidad RGBA.

Esta notación es conocida por todos los navegadores, incluido Internet Explorer (IE9). Para los navegadores antiguos se recomienda la notación más clásica RGB. Para estos navegadores, el fondo no será transparente pero no lo menos hay un color de fondo.

## Resumen

- Cambiamos el color del texto con la propiedad color, el color de fondo con background-color.
- Se puede especificar un color escribiendo su nombre en Inglés (black, por ejemplo), como hexadecimal (# FFC8D3) o notación RGB (rgb (250,25,118)).
- Puedes agregar una imagen de fondo con background-image. Puedes optar por configurar la imagen de fondo, en la pantalla en mosaico o no, e incluso posicionarla en cualquier lugar de la página.
- Puedes hacer una parte de la página transparente con la propiedad de opacidad o la notación RGBA (igual notación RGB, con un valor que indica el cuarto nivel de transparencia).



## Bordes y sombreados

Nuevo capítulo, nuevo lote de propiedades CSS. Aquí vamos a ver los bordes y efectos de sombra que se pueden aplicar tanto en el texto como en los bloques que componen nuestra página.

Vamos a volver a utilizar, en particular, nuestro conocimiento de los colores para elegir el color de nuestros bordes y sombreados.

### Bordes estándar

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

Para llegar al punto, propongo aquí utilizar directamente la propiedad border que incluye todas estas propiedades. ¿Recuerdas la propiedad background? Esto funciona con el mismo principio: vamos a ser capaces de combinar varios valores.

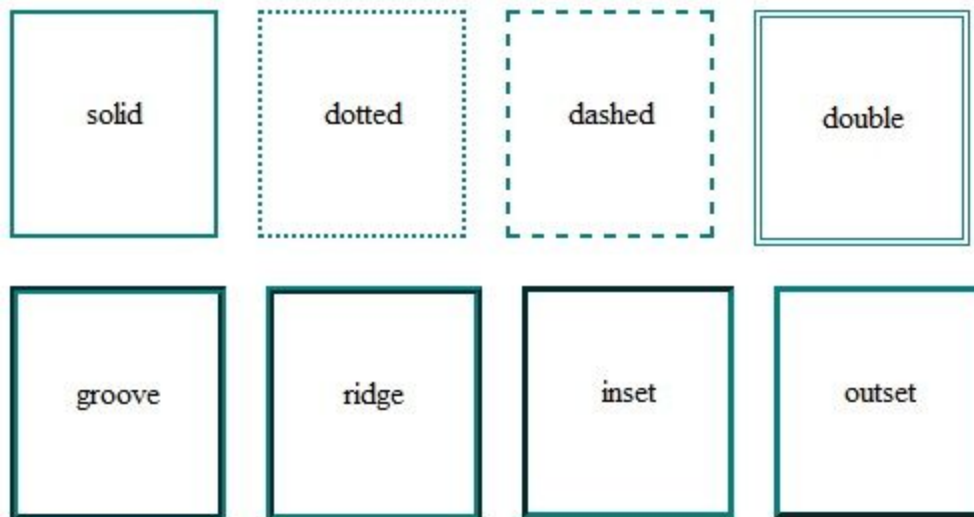
Para border se pueden utilizar un máximo de tres valores para cambiar la apariencia de los bordes:

- ancho: introduce la anchura del borde. Poner un valor en píxeles (por ejemplo 2px).
- color: el color del borde. El uso, es como hemos aprendido, un nombre de color (negro, rojo, ...) o en hexadecimal (# FF0000) o un valor RGB (rgb (198, 212, 37)).
- tipo de borde: aquí tienes la opción del borde que puede ser una línea o puntos, o guiones, etc. Aquí los diferentes valores disponibles:

- none: sin bordes (por defecto);
- solid: una sola línea;
- dotted: puntos;
- dashed: trazos;
- double: doble contorno;
- groove: en relieve;
- ridge: otro efecto en relieve;
- inset: efecto 3D global deprimido;
- outset: efecto 3D global resaltado.

Por lo tanto, para tener un borde azul, tipo guiones, de tres píxeles alrededor de mis títulos, escribo:

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



### Arriba, derecha, izquierda, abajo ...

¿Quién dice que tienes que aplicar el mismo margen en los cuatro lados de un elemento? Si quieres poner bordes diferentes según el lado (arriba, abajo, izquierda o derecha), puedes, no hay problema. En este caso, debes utilizar estas cuatro propiedades:

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

También hay equivalentes para configurar cada detalle del borde, caso de ser necesario: `border-top-width` para cambiar el grosor del borde superior, `border-top-color` para el color superior, etc.

También son super-propiedades, pero que funcionan como `border`, pero se aplican sólo a un lado. Para añadir un borde solamente a las secciones izquierda y derecha, escribir:

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

```
}
```

Puedes cambiar los bordes de cualquier tipo de elemento de la página. Lo hemos hecho aquí con los párrafos pero también se pueden cambiar bordes de imágenes, textos destacados con `<strong>`, etc.

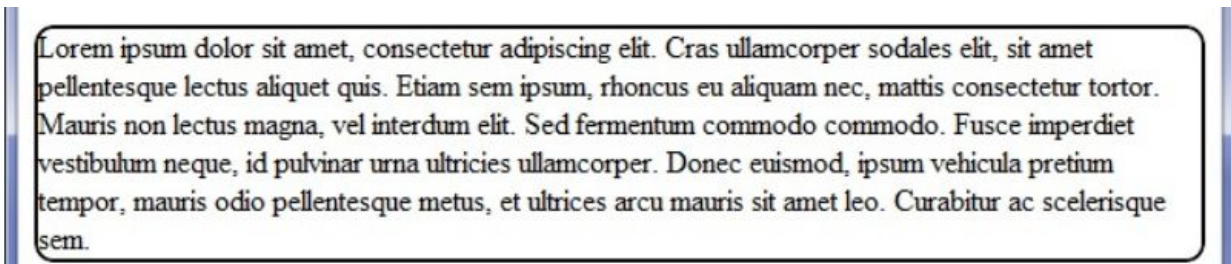
### Bordes redondeados

Los bordes redondeados, que es una propiedad esperada por los desarrolladores de páginas web desde hace mucho tiempo, con CSS3 han llegado, ahora es posible crearlos fácilmente.

La propiedad `border-radius` nos permitirá completar fácilmente las esquinas de cualquier elemento. Basta indicar el tamaño ("importancia") del redondeo en píxeles:

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

El redondeo se verá como se muestra seguidamente



Se puede precisar la forma del redondeo en cada esquina. En este caso se indican los cuatro valores:

```
p
{
border-radius: 10px 5px 10px 5px;
}
```

Los valores corresponden a los ángulos en este orden:

1. parte superior izquierda;
2. parte superior derecha;
3. parte inferior derecha;

#### 4. parte inferior izquierda.

Por último, es posible refinar nuestras esquinas redondeadas para crear curvas elípticas. En este caso, se han de especificar dos valores separados por una barra (/). Probarlo es sin duda la mejor manera de ver el efecto:

```
p
{
border-radius: 20px / 10px;
}
```

Los bordes redondeados trabajan con todos los navegadores, incluido Internet Explorer desde la versión 9 (IE9). Para versiones anteriores de Mozilla Firefox, Chrome y Safari, era necesario el uso de prefijos, es decir, tenías que escribir el código CSS en diferentes versiones para la propiedad (-moz-border-radius para Firefox, -webkit-border-radius para Safari, etc). Esto, afortunadamente, ya no es necesario hoy en día, a menos que desees administrar las versiones anteriores de estos navegadores.

## Sombras

Las sombras son parte de las últimas innovaciones que ofrece CSS3. Hoy en día, apenas una sola línea de CSS basta para agregar sombras en una página.

Aquí vamos a describir dos tipos de sombras:

sombras de cajas;  
sombras de texto.

- box-shadow: sombras de cajas

La propiedad box-shadow se aplica a todo el bloque y toma cuatro valores en el orden siguiente:

1. desplazamiento horizontal de la sombra;
2. desplazamiento vertical de la sombra;
3. difuminado del gradiente;
4. color del sombreado.

Por ejemplo, para un sombreado negro de seis píxeles, sin suavizar, escribimos:

```
p
{
box-shadow: 6px 6px 0px black;
```

```
}
```

Añadir difuminado a través de un tercer parámetro. El difuminado puede ser bajo (menos desplazamiento), normal (igual al cambio) o elevado (por encima de desplazamiento). Prueba con un desplazamiento normal:

```
p
{
  box-shadow: 6px 6px 6px black;
}
```

También puedes agregar un cuarto valor opcional en el recuadro. En este caso, la sombra se coloca dentro de la caja, para dar un efecto enfatizado.

```
p
{
  box-shadow: 6px 6px 6px black inset;
}
```

- text-shadow: sombra texto

Con text-shadow, puedes añadir directamente una sombra en las letras del texto. Los valores de la función son exactamente de la misma forma box-shadow: desplazamiento, difuminado y color.

```
p
{
  text-shadow: 2px 2px 4px black;
}
```

El resultado se muestra a continuación:

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.

## Resumen

- Se puede aplicar un borde a un elemento con la propiedad border. Debes indicar el ancho del

borde, su color y tipo (línea sólida, línea de puntos ...).

- Puedes redondear los bordes con `border-radius`.
- Puedes agregar una sombra a bloques de texto con `box-shadow`. Se debe indicar el desplazamiento horizontal y vertical sombra, nivel de reblandecimiento y color.
- El texto también puede tener una sombra con `text-shadow`.

## Creación de apariencias dinámicas

Este es uno de sus puntos fuertes: CSS también permite modificar la apariencia de los elementos de forma dinámica, es decir que los elementos pueden cambiar de forma, una vez cargada la página. Vamos a utilizar una potente función de CSS, pseudoformatos.

Vemos en este capítulo cómo cambiar la apariencia:

- al vuelo;
- cuando se hace clic;
- en el foco (punto seleccionado);
- cuando se accede a un enlace.

Verás que CSS no para de sorprendernos.

- Al vuelo

Vamos a descubrir en este capítulo varios pseudo-formatos. El primero que voy a mostrar se llama: hover. Al igual que todos los otros pseudo-formatos que veremos, esta es la información que se añade después del nombre de la etiqueta (o la clase) en el CSS, así:

```
a:hover
{
}
```

:hover significa "sobrevolar". a:hover se puede traducir como: "cuando el ratón está sobre el enlace".

A partir de ahí, te toca definir el aspecto que debe tener al hacer clic en los enlaces. Da rienda suelta a tu imaginación, no hay límite.

Aquí se indica el código de un ejemplo de representación de los enlaces, pero no dudes en inventar los tuyos propios

```
a /* Enlaces por defecto (no al vuelo) */
{
text-decoration: none;
color: red;
font-style: italic;
}
```

```
a:hover /* Aspecto al sobrevuelo de los enlaces */
{
text-decoration: underline;
color: green;
}
```

Se han definido dos versiones de los estilos para los enlaces:

- para los enlaces por defecto (no al vuelo);
- para los enlaces al vuelo.

Si bien a menudo se utiliza en los enlaces, se puede cambiar el aspecto de cualquier elemento. Por ejemplo, puedes cambiar la apariencia de los párrafos al hacer clic en:

```
p:hover /* Cuando se apunta sobre un párrafo */
{
}
```

### Para hacer clic, en la selección

Puedes interactuar aún más finamente en el CSS. Vamos a ver que podemos hacer para cambiar la apariencia de las características cuando se hace clic y cuando se seleccionan

:active: en el momento del clic

El pseudoformato :active permite aplicar un estilo en particular cuando se hace clic. En la práctica, sólo se utiliza en los enlaces.

El enlace mantendrá este aspecto un corto período de tiempo. de hecho, el cambio se produce cuando se pulsa el botón del ratón. Claramente, esto no es necesariamente siempre visible. Por ejemplo, puedes cambiar el color de fondo del enlace al hacer clic:

```
a:active /* Cuando el usuario pulsa el enlace */
{
background-color: #FFCC66;
}
```

:focus: cuando el elemento está seleccionado

Aquí es un poco diferente. El pseudoformato :focus se aplica cuando el elemento está seleccionado.

Una vez que haces clic en el enlace, es "seleccionado" (hay un pequeño borde de trazos a su



alrededor). Es decir, la selección.

Este pseudoformato puede ser aplicado a otras etiquetas HTML que no hemos visto todavía, como los formularios.

```
a:focus /* Cuando el visitante selecciona el enlace */
{
background-color: #FFCC66;
}
```

### Cuando el enlace ya ha sido consultado

Es posible aplicar un estilo a un enlace a una página que ya se ha visto. De forma predeterminada, los colores de tu navegador muestran el enlace en un color púrpura más bien feo (desde mi perspectiva por lo menos).

Puedes cambiar esta apariencia con: `visited` (significa "visitado"). En la práctica, en los enlaces consultados, no se puede cambiar mucho más que el color.

```
a:visited /* Cuando el visitante ya ha visto la página */
{
color: #AAA; /* Aplicar un color gris*/
}
```

Si no quieres que los enlaces ya visitados sean de color diferente, tendrás que aplicar el mismo color que los enlaces normales. Muchos sitios web hacen esto. Una excepción notable: Google. Esto es muy útil, ya que podemos ver en los resultados de búsqueda si se han consultado o no los sitios que presenta.

### Resumen

- En CSS, puedes cambiar la apariencia de ciertas secciones de forma dinámica después de la carga de la página, cuando se producen determinados eventos. Esto se hace usando los pseudoformatos.
- El pseudoformato `:hover` puede cambiar el aspecto al pasar el ratón (por ejemplo: `a:hover` para cambiar la apariencia que se vincula cuando el ratón pasa por encima).
- El pseudoformato `:active` modifica la apariencia de los enlaces cuando se hace clic, `:visited` - cuando un enlace ha sido visitado.
- El pseudoformato `:focus` permite cambiar la apariencia de un elemento seleccionado.