



ELEARNING TOTAL

Programador Web / Nivel 1 – Unidad 3

Programador Web – Nivel 1

Unidad 3: Estilos CSS



Indice

Unidad 2: Estilos CSS

¿Qué es CSS?	4
¿Cómo incluir CSS en un documento HTML?	8
Selectores	13
Unidades de medida	22
Colores	25
Texto	29
Enlaces	32
Listas	35
Tablas	38
Modelo de cajas	41



Objetivos

Que el alumno logre:

- Conocer el lenguaje de hojas de estilo y su implementación en el código HTML.



¿Qué es CSS?

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados “documentos semánticos”).

Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

Al crear una página web, se utiliza en primer lugar el lenguaje HTML/XHTML para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc.

Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc.

BREVE HISTORIA DE CSS

Las hojas de estilos aparecieron poco después que el lenguaje de etiquetas SGML, alrededor del año 1970. Desde la creación de SGML, se observó la necesidad de definir un mecanismo que permitiera aplicar de forma consistente diferentes estilos a los documentos electrónicos.

El gran impulso de los lenguajes de hojas de estilos se produjo con el boom de Internet y el crecimiento exponencial del lenguaje HTML para la creación de documentos electrónicos. La guerra de navegadores y la falta de un estándar para la definición de los estilos dificultaban la creación de documentos con la misma apariencia en diferentes navegadores.



El organismo W3C (World Wide Web Consortium), encargado de crear todos los estándares relacionados con la web, propuso la creación de un lenguaje de hojas de estilos específico para el lenguaje HTML y se presentaron nueve propuestas. Las dos propuestas que se tuvieron en cuenta fueron la CHSS (Cascading HTML Style Sheets) y la SSP (Stream-based Style Sheet Proposal).

La propuesta CHSS fue realizada por Håkon Wium Lie y SSP fue propuesto por Bert Bos. Entre finales de 1994 y 1995 Lie y Bos se unieron para definir un nuevo lenguaje que tomaba lo mejor de cada propuesta y lo llamaron CSS (Cascading Style Sheets).

En 1995, el W3C decidió apostar por el desarrollo y estandarización de CSS y lo añadió a su grupo de trabajo de HTML. A finales de 1996, el W3C publicó la primera recomendación oficial, conocida como “CSS nivel 1”.

A principios de 1997, el W3C decide separar los trabajos del grupo de HTML en tres secciones: el grupo de trabajo de HTML, el grupo de trabajo de DOM y el grupo de trabajo de CSS.

El 12 de Mayo de 1998, el grupo de trabajo de CSS publica su segunda recomendación oficial, conocida como “CSS nivel 2”. La versión de CSS que utilizan todos los navegadores de hoy en día es CSS 2.1, una revisión de CSS 2. Al mismo tiempo, la siguiente recomendación de CSS, conocida como “CSS nivel 3”, continúa en desarrollo desde 1998 y acompaña el desarrollo de HTML 5.

ESPECIFICACIÓN OFICIAL

La especificación o norma oficial que se utiliza actualmente para diseñar páginas web con CSS se puede consultar libremente en <http://www.w3.org/>

El sitio web del organismo W3C dispone de una sección en la que se detalla el trabajo que el W3C está desarrollando actualmente en relación a CSS y también dispone de un blog en el que se publican todas las novedades relacionadas con CSS (<http://www.w3.org/blog/CSS>).



FUNCIONAMIENTO BÁSICO DE CSS

Antes de que se generalizara el uso de CSS, los diseñadores de páginas web utilizaban etiquetas HTML especiales para modificar el aspecto de los elementos de la página. El siguiente ejemplo muestra una página HTML con estilos definidos sin utilizar CSS: (ver ejemplo1.html)

```
<!DOCTYPE html >
<html>
<head>
<meta charset="utf-8" />
<title>Ejemplo de estilos sin CSS</title>
</head>
<body>
<h1><font color="red" face="Arial" size="5">Titular de la página</font></h1>
<p><font color="gray" face="Verdana" size="2">Un párrafo de texto no muy
largo.</font></p>
</body>
</html>
```

El ejemplo anterior utiliza la etiqueta con sus atributos color, face y size para definir el color, el tipo y el tamaño de letra de cada elemento de la página.

El problema de utilizar este método para definir el aspecto de los elementos se puede ver claramente con el siguiente ejemplo: si la página tuviera 50 elementos diferentes, habría que insertar 50 etiquetas . Si el sitio web entero se compone de 10.000 páginas diferentes, habría que definir 500.000 etiquetas . Como cada etiqueta tiene tres atributos, habría que definir 1.5 millones de atributos.

Como el diseño de los sitios web está en constante evolución, es habitual modificar cada cierto tiempo el aspecto de las páginas del sitio. Siguiendo con el ejemplo anterior, cambiar el aspecto del sitio requeriría modificar 500.000 etiquetas y 1.5 millones de atributos.

La solución que propone CSS es mucho mejor, como se puede ver en el siguiente ejemplo:



```
<html>
<head>
<title>Ejemplo de estilos con CSS</title>
<style type="text/css">
h1 {color:red;
font-family:Arial;
font-size:20px;
}
p {color:gray;
font-family:Verdana;
font-size:12px;
}
</style>
</head>
<body>
<h1>Titular de la página</h1>
<p>Un párrafo de texto no muy largo.</p>
</body>
</html>
```

CSS permite separar los contenidos de la página y la información sobre su aspecto. En el ejemplo anterior, dentro de la propia página HTML se crea una zona especial en la que se incluye toda la información relacionada con los estilos de la página.

Utilizando CSS, se pueden establecer los mismos estilos con menos esfuerzo y sin ensuciar el código HTML de los contenidos con etiquetas . Como se verá más adelante, la etiqueta <style> crea una zona especial donde se incluyen todas las reglas CSS que se aplican en la página.

En el ejemplo anterior, dentro de la zona de CSS se indica que todas las etiquetas <h1> de la página se deben ver de color rojo, con un tipo de letra Arial y con un tamaño de letra grande. Además, las etiquetas <p> de la página se deben ver de color gris, con un tipo de letra Verdana y con un tamaño de letra medio.

Definir los estilos de esta forma ahorra miles de etiquetas y millones de atributos respecto a la solución anterior, pero sigue sin ser una solución ideal. Como los estilos CSS sólo se aplican en la página que los incluye, si queremos que las 10.000 páginas diferentes del sitio tengan el mismo aspecto, se deberían copiar 10.000 veces esas mismas reglas CSS. Más adelante se explica la solución que propone CSS para evitar este problema.



Cómo incluir CSS en un documento HTML

Una de las principales características de CSS es su flexibilidad y las diferentes opciones que ofrece para realizar una misma tarea. De hecho, existen tres opciones para incluir CSS en un documento HTML.

INCLUIR CSS EN EL MISMO DOCUMENTO HTML

Los estilos se definen en una zona específica del propio documento HTML. Se emplea la etiqueta `<style>` de HTML y solamente se pueden incluir en la cabecera del documento (sólo dentro de la sección `<head>`).

Ejemplo:

```
<!DOCTYPE html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Ejemplo de estilos CSS en el propio documento</ title>
<style type="text/css">
p { color: black; font-family: Verdana; }
</style>
</head>
<body>
<p>Un párrafo de texto.</p>
</body>
</html>
```

Este método se emplea cuando se define un número pequeño de estilos o cuando se quieren incluir estilos específicos en una determinada página HTML que completen los estilos que se incluyen por defecto en todas las páginas del sitio web.

El principal inconveniente es que si se quiere hacer una modificación en los estilos definidos, es necesario modificar todas las páginas que incluyen el estilo que se va a modificar.

DEFINIR CSS EN UN ARCHIVO EXTERNO



En este caso, todos los estilos CSS se incluyen en un archivo de tipo CSS que las páginas HTML enlazan mediante la etiqueta <link>. Un archivo de tipo CSS no es más que un archivo simple de texto cuya extensión es .css Se pueden crear todos los archivos CSS que sean necesarios y cada página HTML puede enlazar tantos archivos CSS como necesite.

Si se quieren incluir los estilos del ejemplo anterior en un archivo CSS externo, se deben seguir los siguientes pasos:

1) Se crea un archivo de texto y se le añade solamente el siguiente contenido:

```
p { color: blue; font-family: Verdana; }
```

2) Se guarda el archivo de texto con el nombre estilos.css Se debe poner especial atención a que el archivo tenga extensión .css y no .txt

3) En la página HTML se enlaza el archivo CSS externo mediante la etiqueta <link> (ver ejemplo4b.html):

```
<html><head>
<title>Ejemplo de estilos CSS en un archivo externo</title>
<link rel="stylesheet" type="text/css" href="/css/estilos.css" media="screen" /></head>
<body>
<p>Un párrafo de texto.</p>
</body>
</html>
```

Cuando el navegador carga la página HTML anterior, antes de mostrar sus contenidos también descarga los archivos CSS externos enlazados mediante la etiqueta <link> y aplica los estilos a los contenidos de la página.

Normalmente, la etiqueta <link> incluye cuatro atributos cuando enlaza un archivo CSS:



rel: indica el tipo de relación que existe entre el recurso enlazado (en este caso, el archivo CSS) y la página HTML. Para los archivos CSS, siempre se utiliza el valor `stylesheet`

type: indica el tipo de recurso enlazado. Sus valores están estandarizados y para los archivos CSS su valor siempre es `text/css`

href: indica la URL del archivo CSS que contiene los estilos. La URL indicada puede ser relativa o absoluta y puede apuntar a un recurso interno o externo al sitio web.

media: indica el medio en el que se van a aplicar los estilos del archivo CSS. Más adelante se explican en detalle los medios CSS y su funcionamiento.

De todas las formas de incluir CSS en las páginas HTML, esta es la más utilizada con mucha diferencia. La principal ventaja es que se puede incluir un mismo archivo CSS en multitud de páginas HTML, por lo que se garantiza la aplicación homogénea de los mismos estilos a todas las páginas que forman un sitio web.

Con este método, el mantenimiento del sitio web se simplifica al máximo, ya que un solo cambio en un solo archivo CSS permite variar de forma instantánea los estilos de todas las páginas HTML que enlazan ese archivo.

INCLUIR CSS EN LOS ELEMENTOS HTML

El último método para incluir estilos CSS en documentos HTML es el peor y el menos utilizado, ya que tiene los mismos problemas que la utilización de las etiquetas ``.

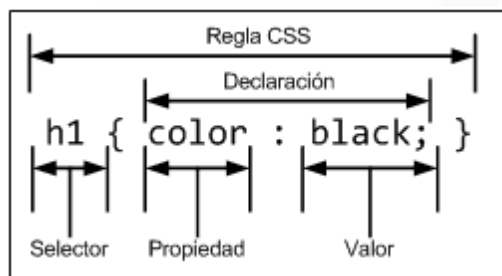
```
<!DOCTYPE html>
<head>
<meta charset="utf-8" />
<title>Ejemplo de estilos CSS en el propio documento</ title>
</head>
<body>
<p style="color: black; font-family: Verdana;">Un párrafo de texto.</p>
</body>
</html>
```



Esta forma de incluir CSS directamente en los elementos HTML solamente se utiliza en determinadas situaciones en las que se debe incluir un estilo muy específico para un solo elemento concreto.

Glosario básico

CSS define una serie de términos que permiten describir cada una de las partes que componen los estilos CSS. El siguiente esquema muestra las partes que forman un estilo CSS muy básico:



Los diferentes términos se definen a continuación:

Regla: cada uno de los estilos que componen una hoja de estilos CSS. Cada regla está compuesta de una parte de “selectores”, un símbolo de “llave de apertura” ({}), otra parte denominada “declaración” y por último, un símbolo de “llave de cierre” ({}).

Selector: indica el elemento o elementos HTML a los que se aplica la regla CSS.

Declaración: especifica los estilos que se aplican a los elementos. Está compuesta por una o más propiedades CSS.

Propiedad: característica que se modifica en el elemento seleccionado, como por ejemplo su tamaño de letra, su color de fondo, etc.

Valor: establece el nuevo valor de la característica modificada en el elemento.



Un archivo CSS puede contener infinitas reglas CSS, cada regla puede contener infinitos selectores y cada declaración puede estar formada por un número infinito de pares propiedad/valor.

El estándar CSS 2.1 define **115 propiedades**, cada una con su propia lista de valores permitidos. Por su parte, los últimos borradores del estándar CSS 3 ya incluyen **239 propiedades**.





Selectores

Para crear diseños web profesionales, es imprescindible conocer y dominar los selectores de CSS. Como se vio en el capítulo anterior, una regla de CSS está formada por una parte llamada “selector” y otra parte llamada “declaración”.

La declaración indica “qué hay que hacer” y el selector indica “a quién hay que hacérselo”. Por lo tanto, los selectores son imprescindibles para aplicar de forma correcta los estilos CSS en una página.

A un mismo elemento HTML se le pueden asignar infinitas reglas CSS y cada regla CSS puede aplicarse a un número infinito de elementos. En otras palabras, una misma regla puede aplicarse sobre varios selectores y un mismo selector se puede utilizar en varias reglas.

El estándar de CSS 2.1 incluye una docena de tipos diferentes de selectores, que permiten seleccionar de forma muy precisa elementos individuales o conjuntos de elementos dentro de una página web.

No obstante, la mayoría de páginas de los sitios web se pueden diseñar utilizando solamente los cinco selectores básicos.

SELECTORES BÁSICOS

Selector universal

Se utiliza para seleccionar todos los elementos de la página. El siguiente ejemplo elimina el margen y el relleno de todos los elementos HTML (por ahora no es importante fijarse en la parte de la declaración de la regla CSS):

```
* {  
  margin: 0;  
  padding: 0;  
}
```



El selector universal se indica mediante un asterisco (*). A pesar de su sencillez, no se utiliza habitualmente, ya que es difícil que un mismo estilo se pueda aplicar a todos los elementos de una página.

No obstante, sí que se suele combinar con otros selectores y además, forma parte de algunos hacks muy utilizados, como se verá más adelante.

Selector de tipo o etiqueta

Selecciona todos los elementos de la página cuya etiqueta HTML coincide con el valor del selector. El siguiente ejemplo selecciona todos los párrafos de la página:

```
p{  
...  
}
```

Para utilizar este selector, solamente es necesario indicar el nombre de una etiqueta HTML (sin los caracteres < y >) correspondiente a los elementos que se quieren seleccionar.

El siguiente ejemplo aplica diferentes estilos a los titulares y a los párrafos de una página HTML:

```
h1 {  
color: red;  
}  
h2 {  
color: blue;  
}  
p {  
color: black;  
}
```

Si se quiere aplicar los mismos estilos a dos etiquetas diferentes, se pueden encadenar los selectores.

En el siguiente ejemplo, los títulos de sección h1, h2 y h3 comparten los mismos estilos:



```
h1 {  
  color: #8A8E27;  
  font-weight: normal;  
  font-family: Arial, Helvetica, sans-serif;  
}
```

```
h2 {  
  color: #8A8E27;  
  font-weight: normal;  
  font-family: Arial, Helvetica, sans-serif;  
}
```

```
h3 {  
  color: #8A8E27;  
  font-weight: normal;  
  font-family: Arial, Helvetica, sans-serif;  
}
```

En este caso, CSS permite agrupar todas las reglas individuales en una sola regla con un selector múltiple. Para ello, se incluyen todos los selectores separados por una coma (,) y el resultado es que la siguiente regla CSS es equivalente a las tres reglas anteriores:

```
h1, h2, h3 {  
  color: #8A8E27;  
  font-weight: normal;  
  font-family: Arial, Helvetica, sans-serif;  
}
```

En las hojas de estilo complejas, es habitual agrupar las propiedades comunes de varios elementos en una única regla CSS y posteriormente definir las propiedades específicas de esos mismos elementos.

```
h1, h2, h3 {  
  color: #8A8E27;  
  font-weight: normal;  
  font-family: Arial, Helvetica, sans-serif;  
}
```



```
}
```

```
h1 { font-size: 18px; }
```

```
h2 { font-size: 16px; }
```

```
h3 { font-size: 14px; }
```

Selector descendiente

Selecciona los elementos que se encuentran dentro de otros elementos. Un elemento es descendiente de otro cuando se encuentra entre las etiquetas de apertura y de cierre del otro elemento.

El selector del siguiente ejemplo selecciona todos los elementos `` de la página que se encuentren dentro de un elemento `<p>`:

```
p span { color: red; }
```

Si el código HTML de la página es el siguiente:

```
<p>  
<span>texto1</span>  
</p>  
<h1><span>texto2</span></h1>
```

El selector `p span` selecciona tanto `texto1` como `texto2`. El motivo es que en el selector descendiente, un elemento no tiene que ser “hijo directo” de otro. La única condición es que un elemento debe estar dentro de otro elemento, sin importar lo profundo que se encuentre.

Al resto de elementos `` de la página que no están dentro de un elemento `<p>`, no se les aplica la regla CSS anterior.



Selector de clase

Si se considera el siguiente código HTML de ejemplo:

```
<body>
<p>Lorem ipsum dolor sit amet...</p>
<p>Nunc sed lacus et est adipiscing accumsan...</p>
<p>Class aptent taciti sociosqu ad litora...</p>
</body>
```

¿Cómo se pueden aplicar estilos CSS sólo al primer párrafo? El selector universal (*) no se puede utilizar porque selecciona todos los elementos de la página. El selector de tipo o etiqueta (p) tampoco se puede utilizar porque seleccionaría todos los párrafos. Por último, el selector descendente (body p) tampoco se puede utilizar porque todos los párrafos se encuentran en el mismo sitio.

Una de las soluciones más sencillas para aplicar estilos a un solo elemento de la página consiste en utilizar el atributo class de HTML sobre ese elemento para indicar directamente la regla CSS que se le debe aplicar:

```
<body>
<p class="destacado">Lorem ipsum dolor sit amet...</p>
<p>Nunc sed lacus et est adipiscing accumsan...</p>
<p>Class aptent taciti sociosqu ad litora...</p>
</body>
```

A continuación, se crea en el archivo CSS una nueva regla llamada destacado con todos los estilos que se van a aplicar al elemento. Para que el navegador no confunda este selector con los otros tipos de selectores, se prefija el valor del atributo class con un punto (.) tal y como muestra el siguiente ejemplo:

```
.destacado { color: red; }
```

El selector .destacado se interpreta como “cualquier elemento de la página cuyo atributo class sea igual a destacado”, por lo que solamente el primer párrafo cumple esa condición.



Este tipo de selectores se llaman selectores de clase y son los más utilizados junto con los selectores de ID que se verán a continuación. La principal característica de este selector es que en una misma página HTML varios elementos diferentes pueden utilizar el mismo valor en el atributo class:

```
<body>
<p class="destacado">Lorem ipsum dolor sit amet...</p>
<p>Nunc sed lacus et <a href="#" class="destacado">est adipiscing</a> accumsan...</p>
<p>Class aptent taciti <em class="destacado">sociosqu ad</em> litora...</p>
</body>
```

Los selectores de clase son imprescindibles para diseñar páginas web complejas, ya que permiten disponer de una precisión total al seleccionar los elementos. Además, estos selectores permiten reutilizar los mismos estilos para varios elementos diferentes.

Selectores de ID

En ocasiones, es necesario aplicar estilos CSS a un único elemento de la página. Aunque puede utilizarse un selector de clase para aplicar estilos a un único elemento, existe otro selector más eficiente en este caso.

El selector de ID permite seleccionar un elemento de la página a través del valor de su atributo id. Este tipo de selectores sólo seleccionan un elemento de la página porque el valor del atributo id no se puede repetir en dos elementos diferentes de una misma página.

La sintaxis de los selectores de ID es muy parecida a la de los selectores de clase, salvo que se utiliza el símbolo de numeral (#) en vez del punto (.) como prefijo del nombre de la regla CSS:

```
#contenido { width:200px; height:200px;
border:2px dotted; background-color: red; }
<div id="contenido">
<p>Primer párrafo</p>
</div>
```



En el ejemplo anterior, el selector #contenido da estructura a la etiqueta div (cuyo atributo id es igual a contenido).

La principal diferencia entre este tipo de selector y el selector de clase tiene que ver con HTML y no con CSS. Como se sabe, en una misma página, el valor del atributo id debe ser único, de forma que dos elementos diferentes no pueden tener el mismo valor de id. Sin embargo, el atributo class no es obligatorio que sea único, de forma que muchos elementos HTML diferentes pueden compartir el mismo valor para su atributo class.

De esta forma, la recomendación general es la de utilizar el selector de ID cuando se quiere aplicar un estilo a un solo elemento específico de la página y utilizar el selector de clase cuando se quiere aplicar un estilo a varios elementos diferentes de la página HTML.

Combinación de selectores básicos

CSS permite la combinación de uno o más tipos de selectores para restringir el alcance de las reglas CSS. A continuación se muestran algunos ejemplos habituales de combinación de selectores.

```
.aviso .especial { ... }
```

El anterior selector solamente selecciona aquellos elementos con un class="especial" que se encuentren dentro de cualquier elemento con un class="aviso".

Si se modifica el anterior selector:

```
div.aviso span.especial { ... }
```

Ahora, el selector solamente selecciona aquellos elementos de tipo con un atributo class="especial" que estén dentro de cualquier elemento de tipo <div> que tenga un atributo class="aviso".

La combinación de selectores puede llegar a ser todo lo compleja que sea necesario:



`ul#menuPrincipal li.destacado a#inicio { ... }`

El anterior selector hace referencia al enlace con un atributo id igual a inicio que se encuentra dentro de un elemento de tipo con un atributo class igual a destacado, que forma parte de una lista con un atributo id igual a menuPrincipal.





Unidades de medida

Muchas de las propiedades de CSS que se ven en los próximos capítulos permiten indicar medidas y colores en sus valores. Además, CSS es tan flexible que permite indicar las medidas y colores de muchas formas diferentes.

Por este motivo, se presentan a continuación todas las alternativas disponibles en CSS para indicar las medidas y los colores.

UNIDADES DE MEDIDA

Las medidas en CSS se emplean, entre otras, para definir la altura, anchura y márgenes de los elementos y para establecer el tamaño de letra del texto. Todas las medidas se indican como un valor numérico entero o decimal seguido de una unidad de medida (sin ningún espacio en blanco entre el número y la unidad de medida).

CSS divide las unidades de medida en dos grupos: absolutas y relativas. Las medidas relativas definen su valor en relación con otra medida, por lo que para obtener su valor real, se debe realizar alguna operación con el valor indicado. Las unidades absolutas establecen de forma completa el valor de una medida, por lo que su valor real es directamente el valor indicado.

Si el valor es 0, la unidad de medida es opcional. Si el valor es distinto a 0 y no se indica ninguna unidad, la medida se ignora completamente, lo que suele ser una fuente habitual de errores para los diseñadores que empiezan con CSS. Algunas propiedades permiten indicar medidas negativas, aunque habitualmente sus valores son positivos.

Unidades relativas

Las unidades relativas son más flexibles que las unidades absolutas porque se adaptan más fácilmente a los diferentes medios. A continuación se muestra la lista de unidades de medida relativas y la referencia que se toma para determinar su valor real:



∴ **em**, (no confundir con la etiqueta de HTML) relativa respecto del tamaño de letra empleado. Aunque no es una definición exacta, el valor de 1em se puede aproximar por la anchura de la letra M (“eme mayúscula”) del tipo y tamaño de letra que se esté utilizando

∴ **ex**, relativa respecto de la altura de la letra x (“equis minúscula”) del tipo y tamaño de letra que se esté utilizando

∴ **px**, (píxel) relativa respecto de la resolución de la pantalla del usuario

Unidades absolutas

Las unidades absolutas definen las medidas de forma completa, ya que sus valores reales no se calculan a partir de otro valor de referencia, sino que son directamente los valores indicados. A continuación se muestra la lista completa de unidades absolutas definidas por CSS y su significado:

∴ **in**, del inglés “inches”, pulgadas (1 pulgada son 2.54 centímetros)

∴ **cm**, centímetros

∴ **mm**, milímetros

∴ **pt**, puntos (1 punto equivale a 1 pulgada/72, es decir, unos 0.35 milímetros)

∴ **pc**, picas (1 pica equivale a 12 puntos, es decir, unos 4.23 milímetros)

Porcentajes

CSS define otra unidad de medida relativa basada en los porcentajes. Un porcentaje está formado por un valor numérico seguido del símbolo % y siempre está referenciado a otra medida. Cada una de las propiedades de CSS que permiten indicar como valor un porcentaje, define el valor al que hace referencia ese porcentaje.

Los porcentajes se pueden utilizar por ejemplo para establecer el valor del tamaño de letra de los elementos:



ELEARNING TOTAL

Programador Web / Nivel 1 – Unidad 3

```
body { font-size: 1em; }
```

```
h1 { font-size: 200%; }
```





Colores

Los colores en CSS se pueden indicar de cinco formas diferentes: palabras clave, colores del sistema, RGB hexadecimal, RGB numérico y RGB porcentual.

Los métodos más habituales son las Palabras claves y el RGB hexadecimal, por eso desarrollaremos estas alternativas.

PALABRAS CLAVE

CSS define 17 palabras clave para referirse a los colores básicos. Las palabras se corresponden con el nombre en inglés de cada color:

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

maroon #800000	red #ff0000	orange #ffa500	yellow #ffff00	olive #808000
purple #800080	fuchsia #ff00ff	white #ffffff	lime #00ff00	green #008000
navy #000080	blue #0000ff	aqua #00ffff	teal #008080	
black #000000	silver #c0c0c0	gray #808080		

La imagen anterior ha sido extraída de la sección sobre colores de la especificación oficial de CSS.



Aunque es una forma muy sencilla de referirse a los colores básicos, este método prácticamente no se utiliza en las hojas de estilos de los sitios web reales, ya que se trata de una gama de colores muy limitada.

Además de la lista básica, los navegadores modernos soportan muchos otros nombres de colores. La lista completa se puede ver en http://en.wikipedia.org/wiki/Web_safe.

RGB hexadecimal

Aunque es el método más complicado para indicar los colores, se trata del método más utilizado con mucha diferencia. De hecho, prácticamente todos los sitios web reales utilizan exclusivamente este método.

Para entender el modelo RGB hexadecimal, en primer lugar es preciso introducir un concepto matemático llamado sistema numérico hexadecimal. Cuando realizamos operaciones matemáticas, siempre utilizamos 10 símbolos para representar los números (del 0 al 9). Por este motivo, se dice que utilizamos un sistema numérico decimal.

No obstante, el sistema decimal es solamente uno de los muchos sistemas numéricos que se han definido. Entre los sistemas numéricos alternativos más utilizados se encuentra el sistema hexadecimal, que utiliza 16 símbolos para representar sus números.

Como sólo conocemos 10 símbolos numéricos, el sistema hexadecimal utiliza también seis letras (de la A a la F) para representar los números. De esta forma, en el sistema hexadecimal, después del 9 no va el 10, sino la A. La letra B equivale al número 11, la C al 12, la D al 13, la E al 14 y la F al número 15.

Definir un color en CSS con el método RGB hexadecimal requiere realizar los siguientes pasos:

1. Determinar las componentes RGB decimales del color original, por ejemplo: R = 71, G = 98, B = 176
2. Transformar el valor decimal de cada componente al sistema numérico hexadecimal. Se trata de una operación exclusivamente matemática, por lo que puedes utilizar una calculadora. En el ejemplo anterior, el valor hexadecimal de cada componente es: R = 47, G = 62, B = B0



3. Para obtener el color completo en formato RGB hexadecimal, se concatenan los valores hexadecimales de las componentes RGB en ese orden y se les añade el prefijo #. De esta forma, el color del ejemplo anterior es #4762B0 en formato RGB hexadecimal.

Siguiendo el mismo ejemplo de las secciones anteriores, el color del párrafo se indica de la siguiente forma utilizando el formato RGB hexadecimal:

```
p { color: #4762B0; }
```

Recuerda que aunque es el método más complicado para definir un color, se trata del método que utilizan la inmensa mayoría de sitios web, por lo que es imprescindible dominarlo. Afortunadamente, todos los programas de diseño gráfico convierten de forma automática los valores RGB decimales a sus valores RGB hexadecimales, por lo que no tienes que hacer ninguna operación matemática.

Colores web safe

Como cada componente RGB de los colores puede tomar un valor entre 0 y 255, el número total de colores que se pueden representar con este formato es de $256 \times 256 \times 256 = 16.777.216$ colores. Sin embargo, en la década de los 90 los monitores de los usuarios no eran capaces de mostrar más de 256 colores diferentes.

A partir de todos los colores disponibles, se eligieron 216 colores que formaron la paleta de colores “web safe”. Esta paleta de colores podía ser utilizada por los diseñadores con la seguridad de que se verían correctamente en cualquier navegador de cualquier sistema operativo de cualquier usuario.

Hoy en día, su importancia ha descendido notablemente, ya que prácticamente todos los usuarios utilizan dispositivos con una profundidad de color de 16 y 32 bits. No obstante, el auge en el uso de los dispositivos móviles hace que siga siendo un tema a considerar, ya que las pantallas de muchos móviles sólo pueden representar un número reducido de colores.



La lista completa de colores web safe y sus valores hexadecimales se pueden consultar en http://en.wikipedia.org/wiki/Web_colors#Web-safe_colors.





Texto

CSS define numerosas propiedades para modificar la apariencia del texto. A pesar de que no dispone de tantas posibilidades como los lenguajes y programas específicos para crear documentos impresos, CSS permite aplicar estilos complejos y muy variados al texto de las páginas web.

La propiedad básica que define CSS relacionada con la tipografía se denomina color y se utiliza para establecer el color de la letra.

Atributos:

.: color: Establece el color de letra utilizado para el texto Ej: `h1 { color: #B1251E; }`

.: font-family: Establece el tipo de letra utilizado para el texto

El tipo de letra del texto se puede indicar de dos formas diferentes:

.: Mediante el nombre de una familia tipográfica: en otras palabras, mediante el nombre del tipo de letra, como por ejemplo “Arial”, “Verdana”, “Garamond”, etc.

.: Mediante el nombre genérico de una familia tipográfica: los nombres genéricos no se refieren a ninguna fuente en concreto, sino que hacen referencia al estilo del tipo de letra. Las familias genéricas definidas son serif (tipo de letra similar a Times New Roman), sans-serif (tipo Arial), cursive (tipo Comic Sans), fantasy (tipo Impact) y monospace (tipo Courier New).

Ej: `p {font-family: Arial, Helvetica, sans-serif};`

.: font-size: Establece el tamaño de letra utilizado para el texto

Además de todas las unidades de medida relativas y absolutas y el uso de porcentajes, CSS permite utilizar una serie de palabras clave para indicar el tamaño de letra del texto:



.: tamaño_absoluto: indica el tamaño de letra de forma absoluta mediante alguna de las siguientes palabras clave: xx-small, x-small, small, medium, large, x-large, xx-large.

.: tamaño_relativo: indica de forma relativa el tamaño de letra del texto mediante dos palabras clave (larger, smaller) que toman como referencia el tamaño de letra del elemento padre.

Ej: p {font-size: medium}

.: font-style: Establece el estilo de la letra utilizada para el texto

Valores: normal | italic | oblique | inherit

Ej: h1 {font-style: italic}

Apariencia

Además de las propiedades relativas a la tipografía del texto, CSS define numerosas propiedades que determinan la apariencia del texto en su conjunto. Estas propiedades adicionales permiten controlar al alineación del texto, el interlineado, la separación entre palabras, etc.

Atributos:

.: text-align: Establece la alineación del contenido del elemento

Valores: left | right | center | justify | inherit

Ej: p {text-align: center}

.: line-height: Permite establecer la altura de línea de los elementos

Valores: normal | <numero> | <medida> | <porcentaje> | inherit

Ej: p { line-height: 120%}



.: text-decoration: Establece la decoración del texto (subrayado, tachado, parpadeante, etc.)

Valores: none | (underline || overline || line-through || blink) | inherit

.: text-transform: Transforma el texto original (lo transforma a mayúsculas, a minúsculas, etc.)

Valores: capitalize | uppercase | lowercase | none | inherit

Ej: `<div style="text-transform: none"><h1>Original</h1>Lorem ipsum dolor
sit amet...</div>`

.: vertical-align: Determina la alineación vertical de los contenidos de un elemento

Valores: baseline | sub | super | top | text-top | middle | bottom | text-bottom | inherit |
<porcentaje> | <medida>

.: letter-spacing: Permite establecer el espacio entre las letras que forman las palabras del texto

Valores: normal | <medida> | inherit

.: word-spacing: Permite establecer el espacio entre las palabras que forman el texto

Valores: normal | <medida> | inherit



Enlaces

ESTILOS BÁSICOS

Los estilos más sencillos que se pueden aplicar a los enlaces son los que modifican su tamaño de letra, su color y la decoración del texto del enlace. Utilizando las propiedades text-decoration y font-weight se pueden conseguir estilos como los que se muestran en la siguiente imagen:



A continuación se muestran las reglas CSS del ejemplo anterior

```
a { margin: 1em 0; }  
.alternativo {color: #CC0000;}  
.simple {text-decoration: none;}  
.importante {font-weight: bold; font-size: 1.3em;}  
.raro {text-decoration:overline;}  
<a href="#">Enlace con el estilo por defecto</a>  
<a class="alternativo" href="#">Enlace de color rojo</ a>
```



`Enlace sin subrayado`
`Enlace muy importante`
`Enlace con un estilo raro`

PSEUDO-CLASES

CSS también permite aplicar diferentes estilos a un mismo enlace en función de su estado. De esta forma, es posible cambiar el aspecto de un enlace cuando por ejemplo el usuario pasa el ratón por encima o cuando el usuario pincha sobre ese enlace.

Como con los atributos id o class no es posible aplicar diferentes estilos a un mismo elemento en función de su estado, CSS introduce un nuevo concepto llamado pseudo-clases. En concreto, CSS define las siguientes cuatro pseudo-clases:

:link, aplica estilos a los enlaces que apuntan a páginas o recursos que aún no han sido visitados por el usuario.

:visited, aplica estilos a los enlaces que apuntan a recursos que han sido visitados anteriormente por el usuario. El historial de enlaces visitados se borra automáticamente cada cierto tiempo y el usuario también puede borrarlo manualmente.

:hover, aplica estilos al enlace sobre el que el usuario ha posicionado el puntero del ratón.

:active, aplica estilos al enlace que está pinchando el usuario. Los estilos sólo se aplican desde que el usuario pincha el botón del ratón hasta que lo suelta, por lo que suelen ser unas pocas décimas de segundo.

Como se sabe, por defecto los navegadores muestran los enlaces no visitados de color azul y subrayados y los enlaces visitados de color morado.

Las pseudo-clases anteriores permiten modificar completamente ese aspecto por defecto y por eso casi todas las páginas las utilizan.

El siguiente ejemplo muestra cómo ocultar el subrayado cuando el usuario pasa el ratón por encima de cualquier enlace de la página:



```
a:hover { text-decoration: none; }
```

Aplicando las reglas anteriores, los navegadores ocultan el subrayado del enlace sobre el que se posiciona el ratón:





LISTAS

Viñetas personalizadas

Por defecto, los navegadores muestran los elementos de las listas no ordenadas con una viñeta formada por un pequeño círculo de color negro. Los elementos de las listas ordenadas se muestran por defecto con la numeración decimal utilizada en la mayoría de países.

No obstante, CSS define varias propiedades para controlar el tipo de viñeta que muestran las listas, además de poder controlar la posición de la propia viñeta. La propiedad básica es la que controla el tipo de viñeta que se muestra y que se denomina `list-style-type`.

.: **list-style-type**: Permite establecer el tipo de viñeta mostrada para una lista

Valores: `disc` | `circle` | `square` | `decimal` | `decimal-leading-zero` | `lower-roman` | `upper-roman` | `lower-greek` | `lower-latin` | `upper-latin` | `armenian` | `georgian` | `lower-alpha` | `upper-alpha` | `none` | `inherit`

En primer lugar, el valor `none` permite mostrar una lista en la que sus elementos no contienen viñetas, números o letras. Se trata de un valor muy utilizado, ya que es imprescindible para los menús de navegación creados con listas, como se verá más adelante.

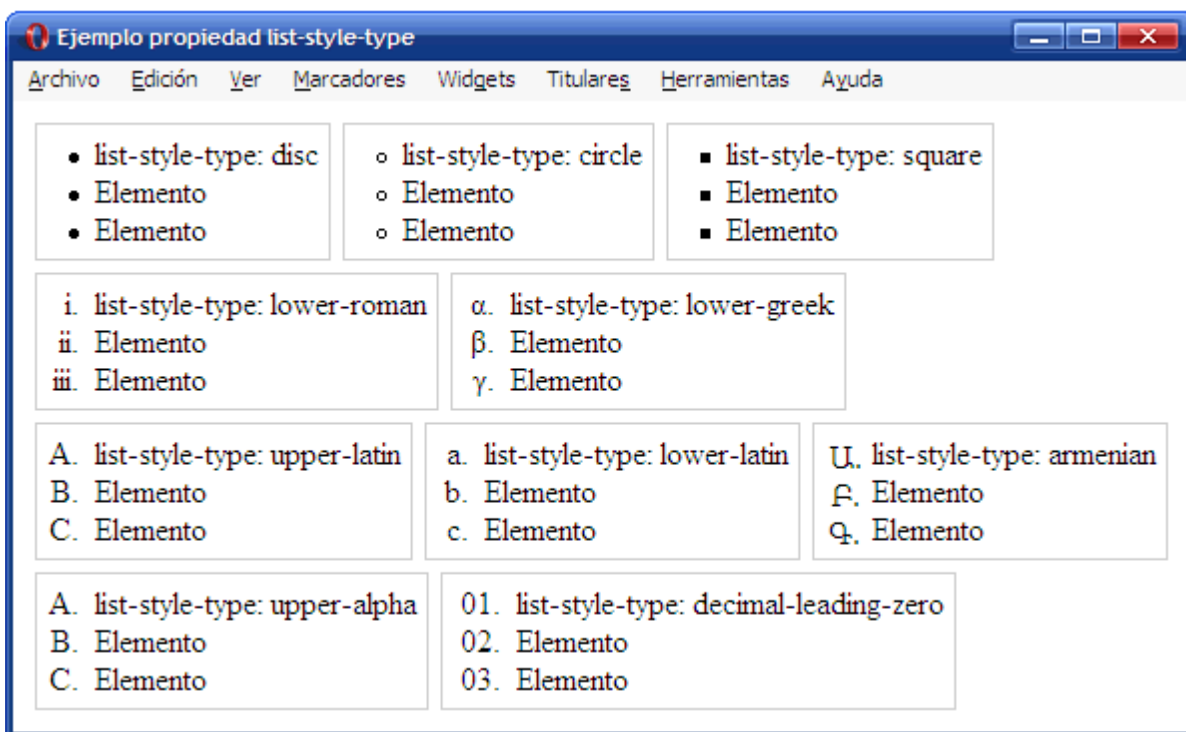
El resto de valores de la propiedad `list-style-type` se dividen en tres tipos: gráficos, numéricos y alfabéticos.

.: **Los valores gráficos** son `disc`, `circle` y `square` y muestran como viñeta un círculo relleno, un círculo vacío y un cuadrado relleno respectivamente.

.: **Los valores numéricos** están formados por `decimal`, `decimal-leading-zero`, `lower-roman`, `upper-roman`, `armenian` y `georgian`.

.: Por último, los **valores alfanuméricos** se controlan mediante `lower-latin`, `lower-alpha`, `upper-latin`, `upper-alpha` y `lower-greek`.

La siguiente imagen muestra algunos de los valores definidos por la propiedad `list-style-type`:



El código CSS de algunas de las listas del ejemplo anterior se muestra a continuación:

```
<ul style="list-style-type: square">
<li>list-style-type: square</li>
<li>Elemento</li>
<li>Elemento</li>
</ul>
<ol style="list-style-type: lower-roman">
<li>list-style-type: lower-roman</li>
<li>Elemento</li>
<li>Elemento</li>
</ol>
<ol style="list-style-type: decimal-leading-zero; padding-left: 2em;">
<li>list-style-type: decimal-leading-zero</li>
<li>Elemento</li>
<li>Elemento</li>
</ol>
```



Cuando se requiere personalizar el aspecto de las viñetas, se debe emplear la propiedad `list-style-image`, que permite mostrar una imagen propia en vez de una viñeta automática.

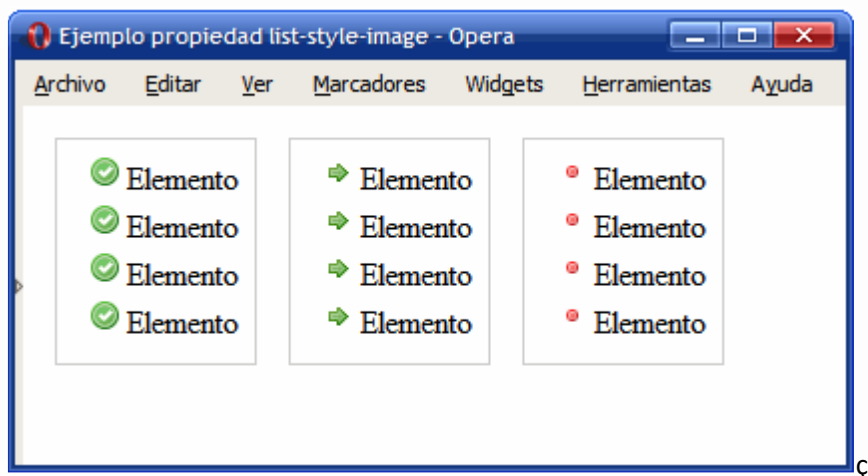
.: `list-style-image`: Permite reemplazar las viñetas automáticas por una imagen personalizada

Valores: `<url>` | `none` | `inherit`

Las imágenes personalizadas se indican mediante la **URL de la imagen**.

Si no se encuentra la imagen o no se puede cargar, se muestra la viñeta automática correspondiente (salvo que explícitamente se haya eliminado mediante la propiedad `list-style-type`).

La siguiente imagen muestra el uso de la propiedad `list-style-image` mediante tres ejemplos sencillos de listas con viñetas personalizadas:

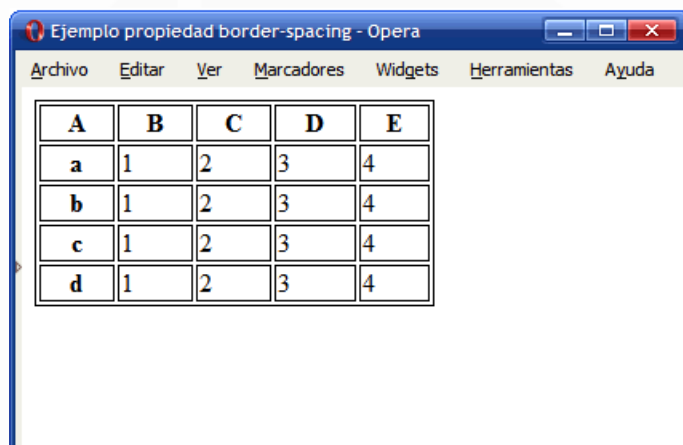




Tablas

ESTILOS BÁSICOS

Cuando se aplican bordes a las celdas de una tabla, el aspecto por defecto con el que se muestra en un navegador es el siguiente:



A	B	C	D	E
a	1	2	3	4
b	1	2	3	4
c	1	2	3	4
d	1	2	3	4

El código HTML y CSS del ejemplo anterior se muestra a continuación (ver ejemplo16.html):

```
.normal {  
width: 250px;  
border: 1px solid #ff0000;  
}  
.normal th, .normal td {  
border: 1px solid #00ff00;  
}  
<table class="normal">  
<tr>  
<th>A</th>  
<th>B</th>  
<th>C</th>  
<th>D</th>
```



```
<th>E</th>  
</tr>  
</table>
```

Bordes

El estándar CSS 2.1 define dos modelos diferentes para el tratamiento de los bordes de las celdas. La propiedad que permite seleccionar el modelo de bordes es border-collapse:

A	B	C	D	E
a	1	2	3	4
b	1	2	3	4
c	1	2	3	4
d	1	2	3	4

El modelo collapse fusiona de forma automática los bordes de las celdas adyacentes, mientras que el modelo separate fuerza a que cada celda muestre sus cuatro bordes. Por defecto, los navegadores utilizan el modelo separate, tal y como se puede comprobar en el ejemplo anterior.

En general, los diseñadores prefieren el modelo collapse porque estéticamente resulta más atractivo y más parecido a las tablas de datos tradicionales. El ejemplo anterior se puede rehacer para mostrar la tabla con bordes sencillos y sin separación entre celdas.

Pseudo-clase

La pseudo-clase :hover permite añadir fácilmente esta característica:

La regla CSS necesaria se muestra a continuación:



```
table tr:hover {  
  background: #FFFF66;  
}
```



Cambio	Compra	Venta	Máximo	Mínimo
€ Euro/Dolar	1.2524	1.2527	1.2539	1.2488
\$ Dolar/Yen	119.01	119.05	119.82	119.82
£ Libra/Dolar	1.8606	1.8611	1.8651	1.8522
¥ Yen/Euro	0.6711	0.6705	0.6676	0.6713



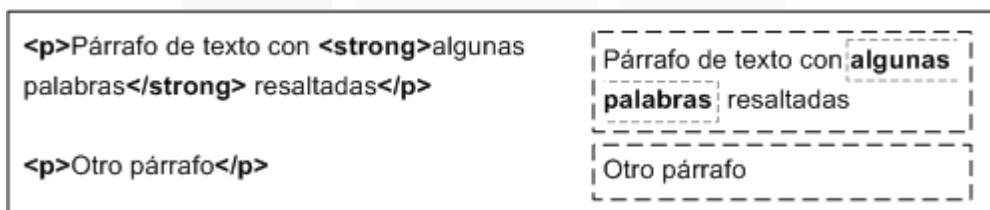
Modelo de cajas

El modelo de cajas o “box model” es seguramente la característica más importante del lenguaje de hojas de estilos CSS, ya que condiciona el diseño de todas las páginas web.

El modelo de cajas es el comportamiento de CSS que hace que todos los elementos de las páginas se representen mediante cajas rectangulares.

Las cajas de una página se crean automáticamente. Cada vez que se inserta una etiqueta HTML, se crea una nueva caja rectangular que encierra los contenidos de ese elemento.

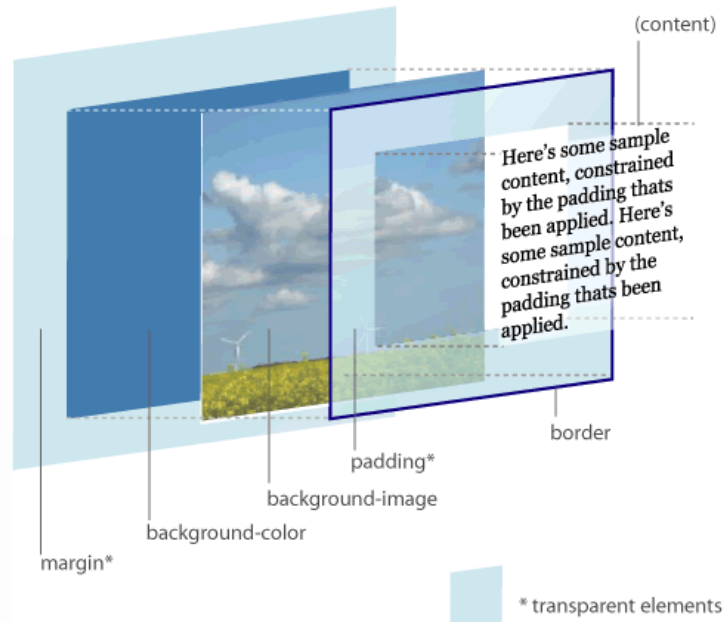
La siguiente imagen muestra las tres cajas rectangulares que crean las tres etiquetas HTML que incluye la página:



Los navegadores crean y colocan las cajas de forma automática, pero CSS permite modificar todas sus características. Cada una de las cajas está formada por seis partes, tal y como muestra la siguiente imagen:



THE CSS BOX MODEL HIERARCHY



(Esquema utilizado con permiso de <http://www.hicksdesign.co.uk/boxmodel/>)

Las partes que componen cada caja y su orden de visualización desde el punto de vista del usuario son las siguientes:

- .: Contenido (content):** se trata del contenido HTML del elemento (las palabras de un párrafo, una imagen, el texto de una lista de elementos, etc.)
- .: Relleno (padding):** espacio libre opcional existente entre el contenido y el borde.
- .: Borde (border):** línea que encierra completamente el contenido y su relleno.
- .: Imagen de fondo (background image):** imagen que se muestra por detrás del contenido y el espacio de relleno.
- .: Color de fondo (background color):** color que se muestra por detrás del contenido y el espacio de relleno.
- .: Margen (margin):** separación opcional existente entre la caja y el resto de cajas adyacentes.



El **relleno** y el **margen** son transparentes, por lo que en el espacio ocupado por el relleno o el margen se muestra el color o imagen de fondo (si están definidos).

Si una caja define tanto un color como una imagen de fondo, la imagen tiene más prioridad y es la que se visualiza. No obstante, si la imagen de fondo no cubre totalmente la caja del elemento o si la imagen tiene zonas transparentes, también se visualiza el color de fondo. Combinando imágenes transparentes y colores de fondo se pueden lograr efectos gráficos muy interesantes.

Anchura

La propiedad CSS que controla la anchura de los elementos se denomina width.

La propiedad width no admite valores negativos y los valores en porcentaje se calculan a partir de la anchura de su elemento padre, es decir, del elemento que lo contiene.

El valor inherit indica que la anchura del elemento se hereda de su elemento padre.

El valor auto, que es el que se utiliza si no se establece de forma explícita un valor a esta propiedad, indica que el navegador debe calcular automáticamente la anchura del elemento, teniendo en cuenta sus contenidos y el sitio disponible en la página.

Altura

La propiedad CSS que controla la altura de los elementos se denomina height.

Al igual que sucede con width, la propiedad height no admite valores negativos.

Si se indica un porcentaje, se toma como referencia la altura del elemento padre.

Si el elemento padre no tiene una altura definida explícitamente, se asigna el valor auto a la altura.

El valor inherit indica que la altura del elemento se hereda de su elemento padre.



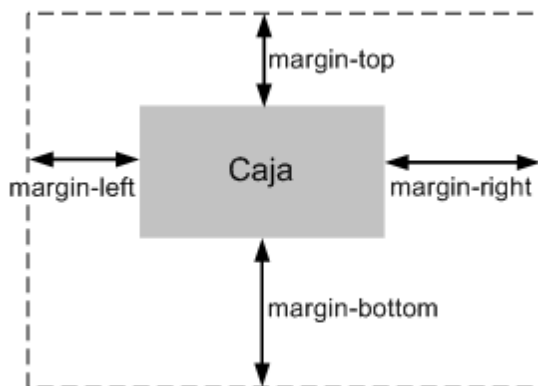
El valor auto, que es el que se utiliza si no se establece de forma explícita un valor a esta propiedad, indica que el navegador debe calcular automáticamente la altura del elemento, teniendo en cuenta sus contenidos y el sitio disponible en la página.

Margen

CSS define cuatro propiedades para controlar cada uno de los márgenes horizontales y verticales de un elemento.

```
.: margin-top  
.: margin-right  
.: margin-bottom  
.: margin-left
```

Cada una de las propiedades establece la separación entre el borde lateral de la caja y el resto de cajas adyacentes:



Las unidades más utilizadas para indicar los márgenes de un elemento son los **píxeles** (cuando se requiere una precisión total), los **em** (para hacer diseños que mantengan las proporciones) y los **porcentajes** (para hacer diseños líquidos o fluidos).

Los márgenes verticales (**margin-top** y **margin-bottom**) sólo se pueden aplicar a los elementos de bloque y las imágenes, mientras que los márgenes laterales (**margin-left** y **margin-right**) se pueden aplicar a cualquier elemento.



Además de las cuatro propiedades que controlan cada uno de los márgenes del elemento, CSS define una propiedad que permite establecer los cuatro márgenes de forma directa empleando una única propiedad.

La propiedad que permite definir de forma simultánea los cuatro márgenes se denomina **margin**.

La propiedad **margin** admite entre uno y cuatro valores, con el siguiente significado:

.: Si solo se indica **un valor**, todos los márgenes tienen ese valor.

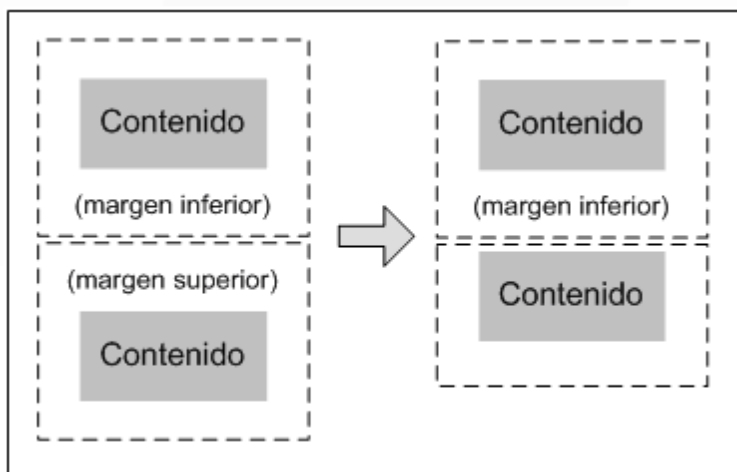
.: Si se indican **dos valores**, el primero se asigna al margen superior e inferior y el segundo se asigna a los márgenes izquierdo y derecho.

.: Si se indican **tres valores**, el primero se asigna al margen superior, el tercero se asigna al margen inferior y el segundo valor se asigna los márgenes izquierdo y derecho.

.: Si se indican los **cuatro valores**, el orden de asignación es: margen superior, margen derecho, margen inferior y margen izquierdo.

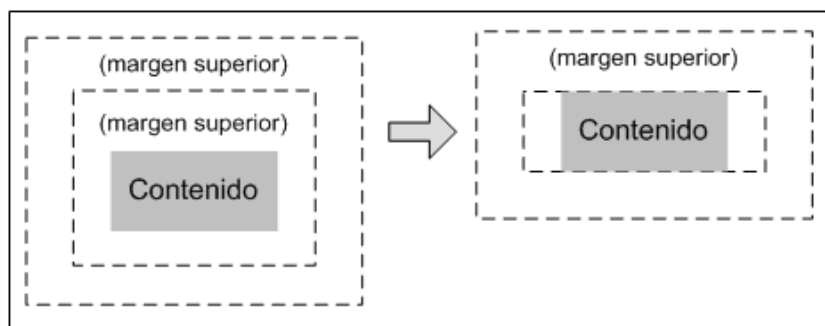
El comportamiento de los márgenes verticales es más complejo que el de los márgenes laterales

Cuando se juntan dos o más márgenes verticales, se fusionan de forma automática y la altura del nuevo margen será igual a la altura del margen más alto de los que se han fusionado.

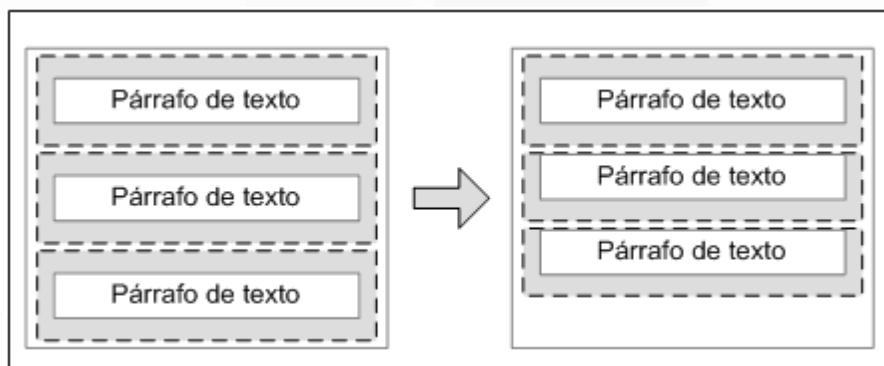




De la misma forma, si un elemento está contenido dentro de otro elemento, sus márgenes verticales se fusionan y resultan en un nuevo margen de la misma altura que el mayor margen de los que se han fusionado:



Aunque en principio puede parecer un comportamiento extraño, la razón por la que se propuso este mecanismo de fusión automática de márgenes verticales es el de dar uniformidad a las páginas web habituales. En una página con varios párrafos, si no se diera este comportamiento y se estableciera un determinado margen a todos los párrafos, el primer párrafo no mostraría un aspecto homogéneo respecto de los demás.



En el caso de un elemento que se encuentra en el interior de otro y sus márgenes se fusionan de forma automática, se puede evitar este comportamiento añadiendo un pequeño relleno (padding: 1px) o un borde (border: 1px solid transparent) al elemento contenedor.

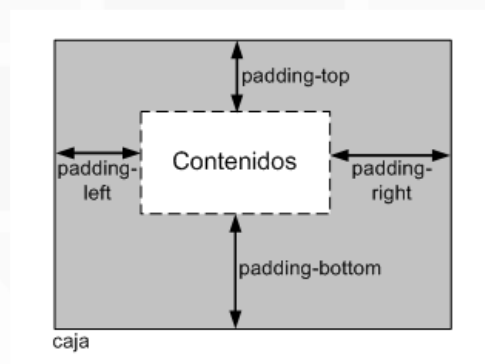


Relleno

CSS definen **cuatro propiedades** para controlar cada uno de los espacios de relleno horizontal y vertical de un elemento.

- .: padding-top
- .: padding-right
- .: padding-bottom
- .: padding-left

Cada una de las propiedades establece la separación entre el lateral de los contenidos y el borde lateral de la caja:



La propiedad que permite definir de forma simultánea los cuatro rellenos se denomina padding.

La propiedad padding admite entre uno y cuatro valores, con el mismo significado que el de la propiedad margin.

Bordes

CSS permite definir el aspecto de cada uno de los cuatro bordes horizontales y verticales de los elementos. Para cada borde se puede establecer su anchura, su color y su estilo.



Anchura

La anchura de los bordes se controla con las **cuatro propiedades** siguientes:

- .: border-top-width
- .: border-right-width
- .: border-bottom-width
- .: border-left-width

La anchura de los bordes se puede indicar mediante una medida (absoluta o relativa y en cualquier unidad de medida de las definidas) o mediante las palabras clave thin (borde delgado), medium (borde normal) y thick (borde ancho).

Color

El color de los bordes se controla con las cuatro propiedades siguientes:

- .: border-top-color
- .: border-right-color
- .: border-bottom-color
- .: border-left-color

Estilo

Por último, CSS permite establecer el estilo de cada uno de los bordes mediante las siguientes propiedades:

- .: border-top-style
- .: border-right-style
- .: border-bottom-style
- .: border-left-style



El estilo de los bordes sólo se puede indicar mediante alguna de las palabras reservadas definidas por CSS. Como el valor por defecto de esta propiedad es none, los elementos no muestran ningún borde visible a menos que se establezca explícitamente un estilo de borde.

Los bordes más utilizados en los diseños habituales son solid y dashed, seguidos de double y dotted.

La propiedad borde admite los 3 valores en la misma regla:

```
td { border: 5px dotted #660000; }
```

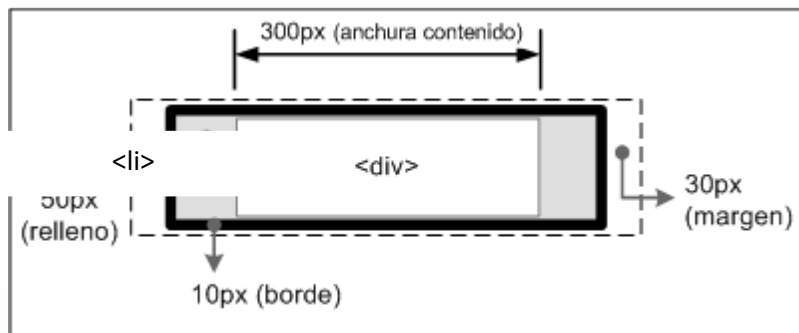
Margen, relleno, bordes y modelo de cajas

La **anchura** y **altura** de un elemento no solamente se calculan teniendo en cuenta sus propiedades width y height. El margen, el relleno y los bordes establecidos a un elemento determinan la anchura y altura final del elemento.

En el siguiente ejemplo se muestran los estilos CSS de un elemento:

```
li {  
  width: 300px;  
  padding-left: 50px;  
  padding-right: 50px;  
  margin-left: 30px;  
  margin-right: 30px;  
  border: 10px solid black;  
}
```

La **anchura total** con la que se muestra el elemento no son los 300 píxel indicados en la propiedad width, sino que se tienen en cuenta todos sus márgenes, rellenos y bordes:



De esta forma, la anchura del elemento en pantalla sería igual a la suma de la anchura original, los márgenes, los bordes y los rellenos:

$$30px + 10px + 50px + 300px + 50px + 10px + 30px = 480 \text{ píxel}$$

Así, la anchura/altura establecida con CSS siempre hace referencia a la anchura/altura del contenido. **La anchura/altura total del elemento debe tener en cuenta además los valores del resto de partes que componen la caja del box model.**

Fondos

El último elemento que forma el box model es el fondo de la caja del elemento. El fondo puede ser un color simple o una imagen. El fondo solamente se visualiza en el área ocupada por el contenido y su relleno, ya que el color de los bordes se controla directamente desde los bordes y las zonas de los márgenes siempre son transparentes.

Para establecer un color o imagen de fondo en la página entera, se debe establecer un fondo al elemento `<body>`. Si se establece un fondo a la página, como el valor inicial del fondo de los elementos es transparente, todos los elementos de la página se visualizan con el mismo fondo a menos que algún elemento especifique su propio fondo.

CSS define cinco propiedades para establecer el fondo de cada elemento (`background-color`, `background-image`, `background-repeat`, `background-attachment`, `background-position`)



En ocasiones, es necesario crear un fondo más complejo que un simple color. CSS permite mostrar una imagen como fondo de cualquier elemento:

.: background-image: Establece una imagen como fondo para los elementos

Valores: <url> | none | inherit

CSS permite establecer de forma simultánea un color y una imagen de fondo. En este caso, la imagen se muestra delante del color, por lo que solamente si la imagen contiene zonas transparentes es posible ver el color de fondo.

Las imágenes de fondo se indican a través de su URL, que puede ser absoluta o relativa. Suele ser recomendable crear una carpeta de imágenes que se encuentre en el mismo directorio que los archivos CSS y que almacene todas las imágenes utilizadas en el diseño de las páginas.

Así, las imágenes correspondientes al diseño de la página se mantienen separadas del resto de imágenes del sitio y el código CSS es más sencillo (por utilizar URL relativas) y más fácil de mantener (por no tener que actualizar URL absolutas en caso de que se cambie la estructura del sitio web).

Por otra parte, suele ser habitual indicar un color de fondo siempre que se muestra una imagen de fondo. En caso de que la imagen no se pueda mostrar o contenga errores, el navegador mostrará el color indicado (que debería ser, en lo posible, similar a la imagen) y la página no parecerá que contiene errores.

Si la imagen que se quiere mostrar es demasiado grande para el fondo del elemento, solamente se muestra la parte de imagen comprendida en el tamaño del elemento. Si la imagen es más pequeña que el elemento, CSS la repite horizontal y verticalmente hasta llenar el fondo del elemento.

Este comportamiento es útil para establecer un fondo complejo a una página web entera.

El siguiente ejemplo utiliza una imagen muy pequeña para establecer un fondo complejo a toda una página:

Imagen original





Reglas CSS

```
body {  
  background-image:url(imagenes/fondo.gif);  
}
```

Con una imagen muy pequeña (y que por tanto, se puede descargar en muy poco tiempo) se consigue cubrir completamente el fondo de la página, con lo que se consigue un gran ahorro de ancho de banda.

En ocasiones, no es conveniente que la imagen de fondo se repita horizontal y verticalmente. Para ello, CSS introduce la propiedad **background-repeat** que permite controlar la forma de repetición de las imágenes de fondo.

El valor **repeat** indica que la imagen se debe repetir en todas direcciones y por tanto, es el comportamiento por defecto.

El valor **no-repeat** muestra una sola vez la imagen y no se repite en ninguna dirección.

El valor **repeat-x** repite la imagen sólo horizontalmente y el valor **repeat-y** repite la imagen solamente de forma vertical.

```
body {  
  background-image:url(imagenes/fondo.gif);  
  background-repeat: repeat-x;  
}
```



Resumen

En esta Unidad...

En la presente unidad desarrollamos los conceptos necesarios para incorporar los atributos gráficos a nuestras estructuras de HTML utilizando el lenguaje CSS

Con las propiedades propuestas podemos comenzar a plantear la estructura gráfica de una página web.

En la próxima Unidad...

En la próxima unidad vamos a comenzar a trabajar con los nuevos elementos incorporados en la versión HTML5 para maquetar sitios web.