CSS

¿Qué son los estilos CSS?

Los estilos CSS son un conjunto de normas que indican cómo el navegador debe visualizar cada uno de los diferentes elementos que contiene una página web. Aunque estas normas habitualmente están relacionadas con aspectos visuales y de posicionamiento y colocación, con CSS3 cada vez existen más posibilidades (animaciones, transiciones, rotaciones...).

Una de las utilidades más evidentes de CSS es la posibilidad de aplicar un estilo (un conjunto de características) a múltiples páginas HTML, no teniendo que repetir los mismos estilos cada vez que creamos un HTML nuevo. En este caso se podría utilizar un único fichero **CSS** externo (con formato **.css**) que afectaría al mismo tiempo a múltiples páginas HTML, o bien incluir el código CSS dentro del mismo fichero HTML (dentro del **<head>**), o utilizar estas dos posibilidades conjuntamente.

Vamos a suponer que queremos que el fondo de nuestra página web tenga un bonito color verde oliva (nombre de color en HTML5: 'olive') y que además el color del texto sea blanco. Y para ello vamos a hacerlo de dos maneras distintas.

Utilizar un fichero CSS externo

Esta es una de las opciones más comunes, ya que posteriormente es posible utilizar el mismo fichero CSS para aplicar el mismo estilo a otras páginas HTML. De esta manera, cualquier cambio realizado en el diseño afectaría al mismo tiempo a todas las páginas HTML que utilicen dicho fichero CSS.

En este caso, tendríamos un fichero HTML y un **fichero CSS** (con extensión .css). Para conectar a ambos utilizaremos la etiqueta **<link>**, que colocaremos en el fichero HTML dentro del **<head>**.

```
<!DOCTYPE html>
<html lang="es">

   <head>
      <meta charset="utf-8"/>
      <title>Título de la web</title>
      <link rel="stylesheet" href="css/estilos.css"
type="text/css"/>
   </head>

   <body>
      Contenido de la web
   </body>
</html>
```

En el código anterior, el fichero CSS se llama **estilos.css** y estaría ubicado en una carpeta llamada **css**, que estaría en la misma carpeta que el fichero HTML.
Dentro del fichero **estilos.css** tendríamos el siguiente contenido:

```
body{
  background-color: olive;
  color: white;
}
```

La estructura siempre es la misma:

Primero se indica el **selector** (para saber a quién afecta el código), que en este caso es **body**. La línea se finaliza la línea con una llave **{**..

Posteriormente se indica la propiedad a modificar (en este caso **background-color** para modificar el color de fondo y **color** para especificar el color del texto).

Seguido a la propiedad se indica el nuevo valor (separados ambos por dos puntos **:**).

Para finalizar cada una de las sentencias CSS se utiliza obligatoriamente el símbolo del punto y coma (**;**)

Para acabar el bloque CSS utilizamos la llave de cierre **}**.

```
selector{
  propiedad: valor;
  propiedad: valor;
}
```

Incluir el código CSS dentro del HTML

Si añadimos el código CSS dentro del mismo fichero HTML (dentro del **<head>**), como aspecto positivo, podemos decir que no necesitamos tener 2 ficheros, ya que todo está en el mismo fichero HTML.

Como punto negativo (o no en algunos casos) es que dicho estilo únicamente afectará al fichero HTML en el que se coloque el código CSS. Lo cual implica que si tenemos múltiples ficheros HTML que tienen que usar los mismos estilos, dicho código se tendría que copiar y pegar en cada uno de los ficheros HTML de manera independiente.

Este código **CSS** se añade dentro del apartado **<head>**, y debe estar dentro de la etiqueta **style**:

```
<style type="text/css">
...
</style>
```

Siguiendo el ejemplo anterior, el código completo quedaría de la siguiente manera:

```
<head>
  <meta charset="utf-8"/>
  <title>Título de la web</title>
  <style type="text/css">
    body{
      background-color:olive;
      color:white;
    }
  </style>
</head>
```

Comentarios

CSS permite incluir comentarios entre sus reglas y estilos. Los comentarios son contenidos de texto que el diseñador incluye en el archivo CSS para su propia información y utilidad. Los navegadores ignoran por completo cualquier comentario de los archivos CSS, por lo que es común utilizarlos para estructurar de forma clara los archivos CSS complejos.

El comienzo de un comentario se indica mediante los caracteres **/*** y el final del comentario se indica mediante ***/**, tal y como se muestra en el siguiente ejemplo:

```
/* Este es un comentario en CSS */
```

Los comentarios pueden ocupar tantas líneas como sea necesario, pero no se puede incluir un comentario dentro de otro comentario:

```
/* Este es un
   comentario CSS de varias
   líneas */
```

Tipos de propiedades CSS

Así como en la vida real existen propiedades (como el color de ojos, estatura o bebida preferida) qué únicamente se pueden aplicar a las personas, o propiedades (como los caballos de potencia, tipo de combustible o número de puertas) que sólo se pueden aplicar a los vehículos, en el mundo de HTML y CSS pasa lo mismo, hay propiedades CSS que sólo se pueden aplicar a un tipo concreto de etiquetas HTML.

Así, existen propiedades CSS que afectan sólo...

- al texto
- a las cajas ** ** que contienen elementos
- a los enlaces
- a las listas
- a las tablas

Antes de hacer un repaso rápido a todas ellas es de obligado cumplimiento tener bien claro el funcionamiento de los selectores CSS.

Selectores

Para poder crear diseños web, es imprescindible conocer y dominar los selectores de CSS. 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 aplicar varias reglas CSS y cada regla CSS puede aplicarse a un número ilimitado 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.

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. El siguiente ejemplo establece en primer lugar las propiedades comunes de los títulos de sección (color y tipo de letra) y a continuación, establece el tamaño de letra de cada uno de ellos:

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

h1 { font-size: 2em; }
h2 { font-size: 1.5em; }
h3 { font-size: 1.2em; }
```

Selector descendente

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>
 ...
 <a href="">...<span>texto2</span></a>
 ...
</p>
...
```

El selector **p span** selecciona tanto **texto1** como **texto2**. El motivo es que en el selector descendente, un elemento no tiene que ser descendiente directo del otro. La única condición es que un elemento debe estar dentro de otro elemento, sin importar el nivel de profundidad en el 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.

Los selectores descendentes permiten aumentar la precisión del selector de tipo o etiqueta. Así, utilizando el selector descendente es posible aplicar diferentes estilos a los elementos del mismo tipo. El siguiente ejemplo amplía el anterior y muestra de color azul todo el texto de los **** contenidos dentro de un <h1>:

```
p span  { color: red;  }
h1 span { color: blue; }
```

Con las reglas CSS anteriores:

- Los elementos **** que se encuentran dentro de un elemento **<p>** se muestran de color rojo.
- Los elementos **** que se encuentran dentro de un elemento **<h1>** se muestran de color azul.
- El resto de elementos **** de la página, se muestran con el color por defecto aplicado por el navegador.

La sintaxis formal del selector descendente se muestra a continuación:

```
selector1 selector2 selector3 ... selectorN
```

Los selectores descendentes siempre están formados por dos o más selectores separados entre sí por espacios en blanco. El último selector indica el elemento sobre el que se aplican los estilos y todos los selectores anteriores indican el lugar en el que se debe encontrar ese elemento.

En el siguiente ejemplo, el selector descendente se compone de cuatro selectores:

```
p a span em { text-decoration: underline; }
```

Los estilos de la regla anterior se aplican a los elementos de tipo **** que se encuentren dentro de elementos de tipo ****, que a su vez se encuentren dentro de elementos de tipo **<a>** que se encuentren dentro de elementos de tipo **<p>**.

No debe confundirse el selector descendente con la combinación de selectores:

```
/* El estilo se aplica a todos los elementos "p", "a", "span"
y "em" */
p, a, span, em { text-decoration: underline; }

/* El estilo se aplica sólo a los elementos "em" que se
  encuentran dentro de "p a span" */
p a span em { text-decoration: underline; }
```

Se puede restringir el alcance del selector descendente combinándolo con el selector universal. El siguiente ejemplo, muestra los dos enlaces de color rojo:

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

```
<p><a href="#">Enlace</a></p>
<p><span><a href="#">Enlace</a></span></p>
```

Sin embargo, en el siguiente ejemplo solamente el segundo enlace se muestra de color rojo:

```
p * a { color: red; }
```

```
<p><a href="#">Enlace</a></p>
<p><span><a href="#">Enlace</a></span></p>
```

La razón es que el selector **`p * a`** se interpreta como todos los elementos de tipo **`<a>`** que se encuentren dentro de cualquier elemento que, a su vez, se encuentre dentro de un elemento de tipo **`<p>`**. Como el primer elemento **`<a>`** se encuentra directamente bajo un elemento **`<p>`**, no se cumple la condición del selector **`p * a`**.

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 a todos los elementos de la página. El selector de tipo 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.

A continuación se muestra otro ejemplo de selectores de clase:

```
.aviso {
 padding: 0.5em;
 border: 1px solid #98be10;
 background: #f6feda;
}

.error {
 color: #930;
 font-weight: bold;
}
```

```
<span class="error">...</span>

<div class="aviso">...</div>
```

El elemento **** tiene un atributo **class="error"**, por lo que se le aplican las reglas CSS indicadas por el selector **.error**. Por su parte, el elemento ** ** tiene un atributo **class="aviso"**, por lo que su estilo es el que define las reglas CSS del selector **.aviso**.

En ocasiones, es necesario restringir el alcance del selector de clase. Si se considera de nuevo el ejemplo anterior:

```
<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>
```

¿Cómo es posible aplicar estilos solamente al párrafo cuyo atributo **class** sea igual a **destacado**? Combinando el selector de tipo y el selector de clase, se obtiene un selector mucho más específico:

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

El selector **p.destacado** se interpreta como *"aquellos elementos de tipo **<p>** que dispongan de un atributo **class** con valor **destacado**"*. De la misma forma, el selector **a.destacado** solamente selecciona los enlaces cuyo atributo **class** sea igual a **destacado**.

De lo anterior se deduce que el atributo **.destacado** es equivalente a ***.destacado**, por lo que todos los diseñadores obvian el símbolo ***** al escribir un selector de clase normal.

No debe confundirse el selector de clase con los selectores anteriores:

/* Todos los elementos de tipo "p" con atributo class="aviso" */

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

/* Todos los elementos con atributo class="aviso" que estén dentro

de cualquier elemento de tipo "p" */

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

/* Todos los elementos "p" de la página y todos los elementos con

atributo class="aviso" de la página */

```
p, .aviso { ... }
```

Por último, es posible aplicar los estilos de varias clases CSS sobre un mismo elemento. La sintaxis es similar, pero los diferentes valores del atributo **`class`** se separan con espacios en blanco. En el siguiente ejemplo:

```
<p class="especial destacado error">Párrafo de texto...</p>
```

Al párrafo anterior se le aplican los estilos definidos en las reglas **`.especial`**, **`.destacado`** y **`.error`**, por lo que en el siguiente ejemplo, el texto del párrafo se vería de color rojo, en negrita y con un tamaño de letra de 15 píxel:

```
.error { color: red; }
.destacado { font-size: 15px; }
.especial  { font-weight: bold; }
```

```
<p class="especial destacado error">Párrafo de texto...</p>
```

Si un elemento dispone de un atributo **`class`** con más de un valor, es posible utilizar un selector más avanzado:

```
.error { color: red; }
.error.destacado { color: blue; }
.destacado { font-size: 15px; }
.especial  { font-weight: bold; }
```

```
<p class="especial destacado error">Párrafo de texto...</p>
```

En el ejemplo anterior, el color de la letra del texto es azul y no rojo. El motivo es que se ha utilizado un selector de clase múltiple **`.error.destacado`**, que se interpreta como *"aquellos elementos de la página que dispongan de un atributo **`class`** con al menos los valores **`error`** y **`destacado`**"*.

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 la almohadilla (**#**) en vez del punto (**.**) como prefijo del nombre de la regla CSS:

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

```
<p>Primer párrafo</p>
<p id="destacado">Segundo párrafo</p>
<p>Tercer párrafo</p>
```

En el ejemplo anterior, el selector **#destacado** solamente selecciona el segundo párrafo (cuyo atributo **id** es igual a **destacado**).

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.

Al igual que los selectores de clase, en este caso también se puede restringir el alcance del selector mediante la combinación con otros selectores. El siguiente ejemplo aplica la regla CSS solamente al elemento de tipo **<p>** que tenga un atributo **id** igual al indicado:

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

A primera vista, restringir el alcance de un selector de ID puede parecer absurdo. En realidad, un selector de tipo **p#aviso** sólo tiene sentido cuando el archivo CSS se aplica sobre muchas páginas HTML diferentes.

En este caso, algunas páginas pueden disponer de elementos con un atributo **id** igual a **aviso** y que no sean párrafos, por lo que la regla anterior no se aplica sobre esos elementos.

No debe confundirse el selector de ID con los selectores anteriores:

/* Todos los elementos de tipo "p" con atributo id="aviso" */

```
p#aviso { ... }
```

/* Todos los elementos con atributo id="aviso" que estén dentro

de cualquier elemento de tipo "p" */

```
p #aviso { ... }
```

/* Todos los elementos "p" de la página y todos los elementos con

atributo id="aviso" de la página */

```
p, #aviso { ... }
```

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 **`<span>`** 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 **`<li>`** con un atributo **`class`** igual a **`destacado`**, que forma parte de una lista **`<ul>`** con un atributo **`id`** igual a **`menuPrincipal`**.

Propiedades CSS más usadas

CSS que afectan al texto

Los textos afectados por el código CSS de los siguiente ejemplos irán ubicados dentro de un ** ** o ****, excepto en los casos en los que se indique lo contrario (como por ejemplo utilizando 'text-align').

Color de Texto

Cambia el color del texto.

```
color: #1162ac;
color: chocolate;
```

Texto color:#1162ac;

Texto color:chocolate;

Se puede especificar con el código hexadecimal del color (utilizando el prefijo **#**) o con el nombre de color (olive, gold, chocolate...).

Tipografía

Utiliza una tipografía determinada.

```
font-family: helvetica;
font-family: charlemagne;
```

Helvetica

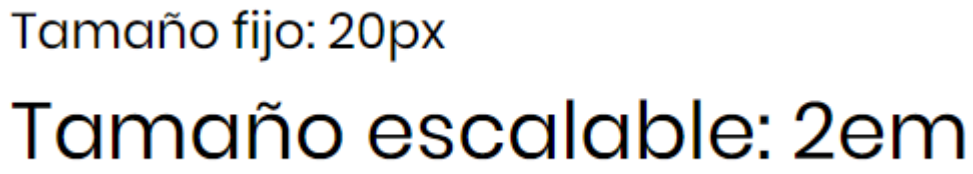
CHARLEMAGNE

Si la fuente no es de sistema se debe añadir el fichero ttf/otf y realizar una vinculación con la misma.

Tamaño del texto

Define el tamaño del texto.

```
font-size: 20px;
font-size: 2em;
```



Se puede definir con un tamaño fijo (expresado en píxeles: **px**), o con un tamaño escalable (**em**), que indica la proporción respecto al tamaño inicial de la fuente (2em equivale al doble del tamaño inicial).

Estilo 'negrita'

Define con CSS el estilo 'negrita' de un texto.

```
font-weight: bold;
```

Aunque el estilo 'negrita' se puede especificar con código HTML a través de la etiqueta **<b>**, con CSS también podemos utilizar esta propiedad de una manera mucho más eficiente.

Los valores que normalmente se utilizan son normal (el valor por defecto) y bold para los textos en negrita. El valor normal equivale al valor numérico 400 y el valor bold al valor numérico 700.

Transformación del texto

Transforma el texto original (lo transforma a mayúsculas, a minúsculas, etc.)

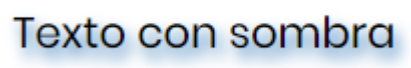
```
text-transform: uppercase;
```

La propiedad **`text-transform`** permite mostrar el texto original transformado en un texto completamente en mayúsculas (**`uppercase`**), en minúsculas **(`lowercase`)** o con la primera letra de cada palabra en mayúscula (**`capitalize`**).

Sombra de un texto

Define una sombra que afecta a un texto a través de 4 parámetros (separación horizontal, separación vertical, suavidad y color).

```
text-shadow: 2px 4px 6px #1162ac;
```

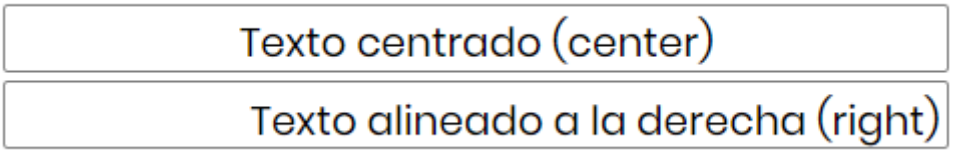


El color se puede especificar con código hexadecimal o con el nombre del color (olive, gold, silver...)

Alineación de un texto

Para definir la alineación de un texto, este debe estar contenido dentro de una caja (, <section>...)

```
text-align: center;
text-align: right;
```

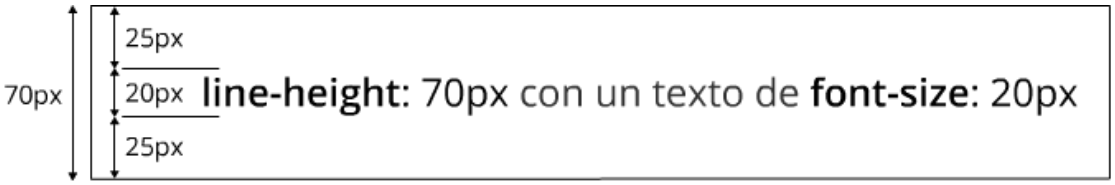


Los valores pueden ser 'left', 'right', 'center' o 'justify'.

Centrado vertical

Sirve para definir la altura de la línea donde se muestra un texto y por lo tanto para centrarlo verticalmente.

```
line-height: 70px;
```

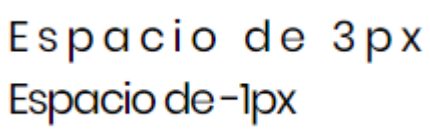


El texto sólo puede ocupar una única línea y debe estar contenido dentro de un .

Espacio entre letras

Define el espacio entre letras.

```
letter-spacing: 3px;
letter-spacing: -1px;
```

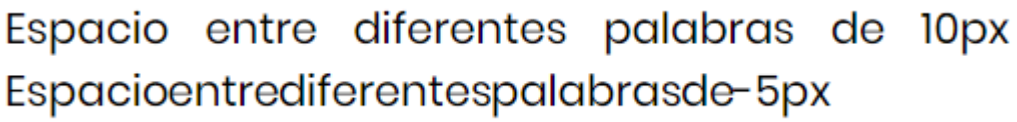


Un valor positivo las aleja entre ellas, mientras que un valor negativo las acerca.

Espacio entre palabras

Define el espacio entre palabras.

```
word-spacing: 10px;
word-spacing: -5px;
```



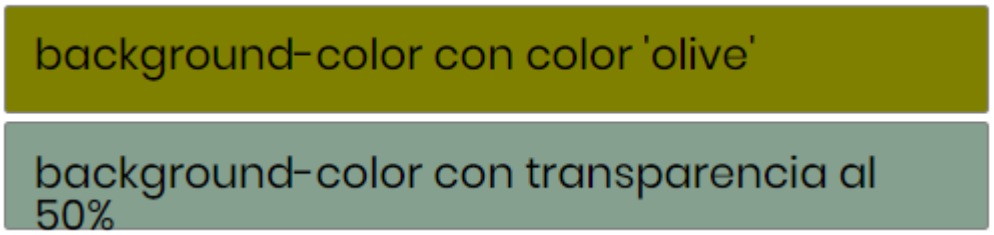
Un valor positivo las aleja entre ellas, mientras que un valor negativo las acerca.

CSS que afecta a las cajas ( )

Color de fondo

Define el color de fondo de una caja (o).

```
background-color: olive;
background-color: rgba(12,65,32,0.5);
```

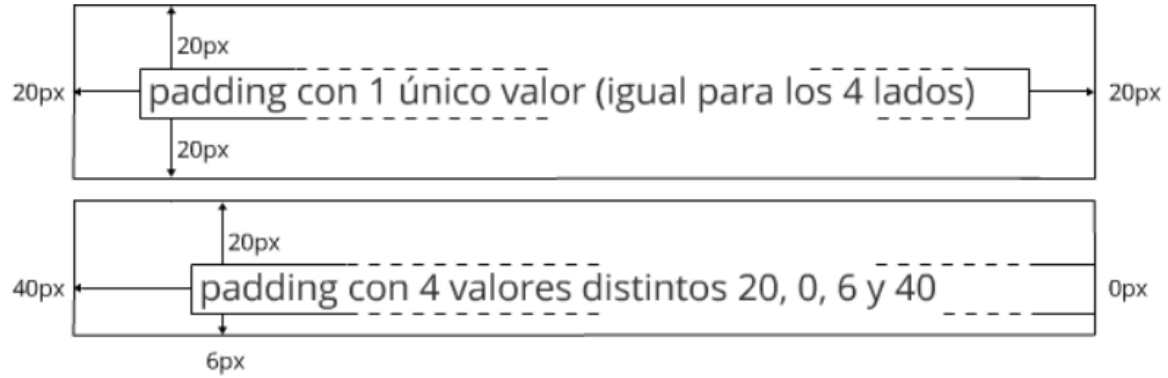


Se puede indicar un color hexadecimal, nombre de color o rgb+a (alpha), donde el 4º valor es la cantidad de transparencia (del 0 al 1).

Separación interna

Define el espacio existente entre el contenido y el borde de la caja.

```
padding: 20px;
padding: 20px 0px 6px 40px;
```

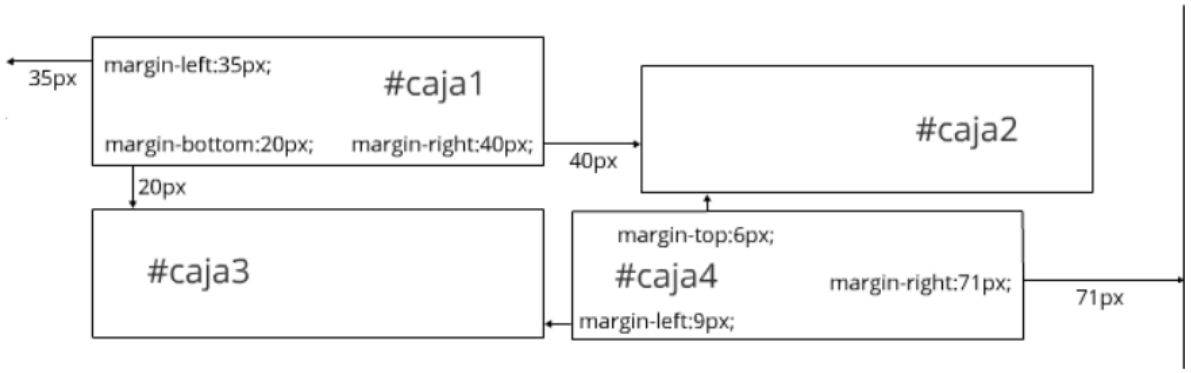


Si se especifica un solo valor será utilizado para la separación por los 4 lados. Si se especifican 4 valores serán utilizados en orden (arriba, derecha, abajo e izquierda) y se especifican 2 serán utilizados (arriba/abajo y derecha/izquierda).

Márgenes y separación

Define el espacio existente entre cajas ( ) o bien entre cajas y los márgenes de la página

```
#caja1{
  margin-left:35px;
  margin-bottom:20px;
}
#caja2{
  margin-left:40px;
}
#caja3{
  margin-left:35px;
  margin-bottom:20px;
}
```



Si se especifica un solo valor será utilizado para la separación por 4 lados. Si se especifican 4 valores serán utilizados en orden (arriba, derecha, abajo e izquierda) y se especifican 2 serán utilizados (arriba/abajo y derecha/izquierda).

Bordes

CSS permite modificar el aspecto de cada uno de los cuatro bordes de la caja de un elemento. Para cada borde se puede establecer su anchura o grosor, su color y su estilo, por lo que en total CSS define 20 propiedades relacionadas con los bordes.

Anchura

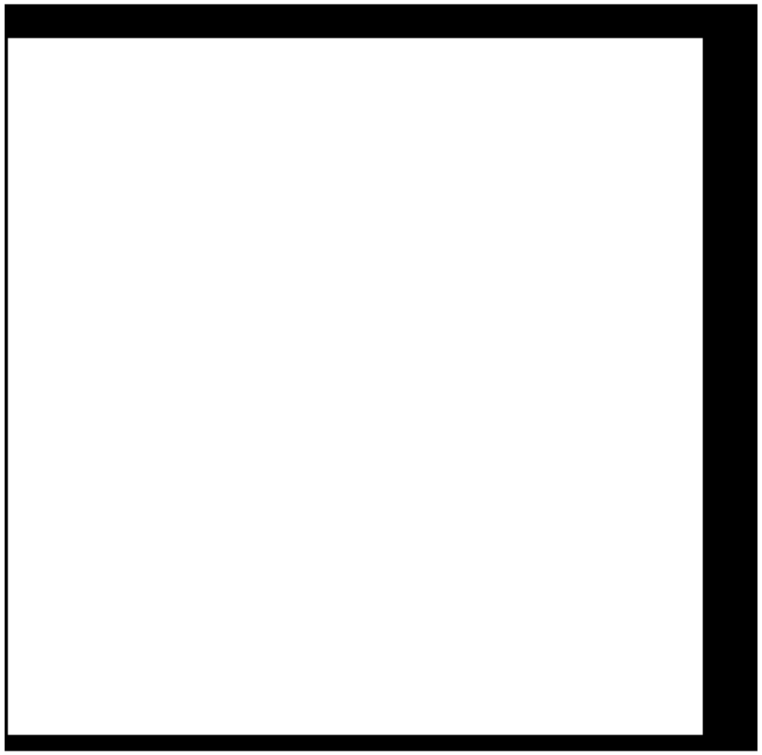
La anchura de los bordes se controla con las cuatro propiedades siguientes:

- border-top-width
- border-right-width
- border-bottom-width
- border-left-width

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

La unidad de medida más habitual para establecer el grosor de los bordes es el píxel, ya que es la que permite un control más preciso sobre el grosor. Las palabras clave apenas se utilizan, ya que el estándar CSS no indica explícitamente el grosor al que equivale cada palabra clave, por lo que pueden producirse diferencias visuales entre navegadores. Así por ejemplo, el grosor **`medium`** equivale a **`4px`** en algunas versiones de Internet Explorer y a **`3px`** en el resto de navegadores.

El siguiente ejemplo muestra un elemento con cuatro anchuras diferentes de borde:



Las reglas CSS utilizadas se muestran a continuación:

```
div {
 width: 200px;  /* anchura de la caja */
 height: 200px; /* altura  de la caja */
 border: solid; /* color sólido para el borde de la caja */
 border-top-width: 10px;
 border-right-width: 1em;
 border-bottom-width: thick;
 border-left-width: thin;
}
```

Si se quiere establecer de forma simultánea la anchura de todos los bordes de una caja, es necesario utilizar una propiedad "shorthand" llamada **`border-width`**:

La propiedad **`border-width`** permite indicar entre uno y cuatro valores. El significado de cada caso es el habitual de las propiedades *"shorthand"*:

```
p { border-width: thin }          /* thin thin thin thin */
p { border-width: thin thick }    /* thin thick thin thick */
p { border-width: thin thick medium }     /* thin thick
medium thick */
p { border-width: thin thick medium thin } /* thin thick
medium thin */
```

Si se indica un solo valor, se aplica a los cuatro bordes. Si se indican dos valores, el primero se aplica al borde superior e inferior y el segundo valor se aplica al borde izquierdo y derecho.

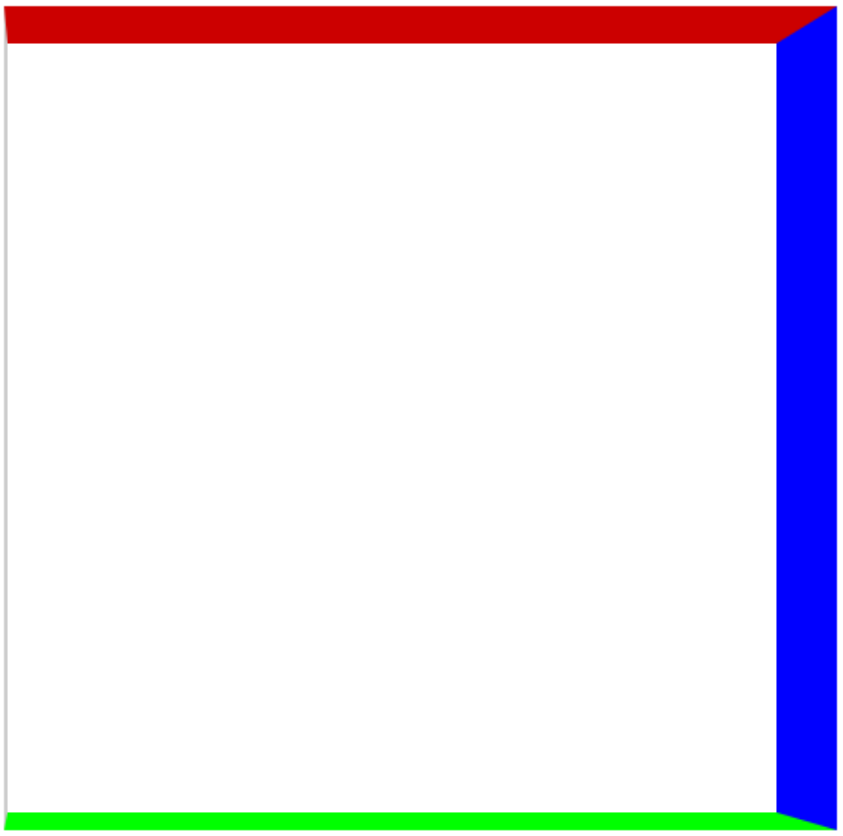
Si se indican tres valores, el primero se aplica al borde superior, el segundo se aplica al borde izquierdo y derecho y el tercer valor se aplica al borde inferior. Si se indican los cuatro valores, el orden de aplicación es superior, derecho, inferior e izquierdo.

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

El siguiente ejemplo muestra el ejemplo anterior en el cual el elemento div tiene cuatro colores para las diferentes anchuras de borde:



Las reglas CSS que se han añadido a las anteriores se muestran a continuación:

```
div {
 border-top-color: #CC0000;
 border-right-color: blue;
 border-bottom-color: #00FF00;
 border-left-color: #CCC;
}
```

CSS incluye una propiedad *"shorthand"* llamada **`border-color`** para establecer de forma simultánea el color de todos los bordes de una caja:

En este caso, al igual que sucede con la propiedad **`border-width`**, es posible indicar de uno a cuatro valores y las reglas de aplicación son idénticas a las de la propiedad **`border-width`**.

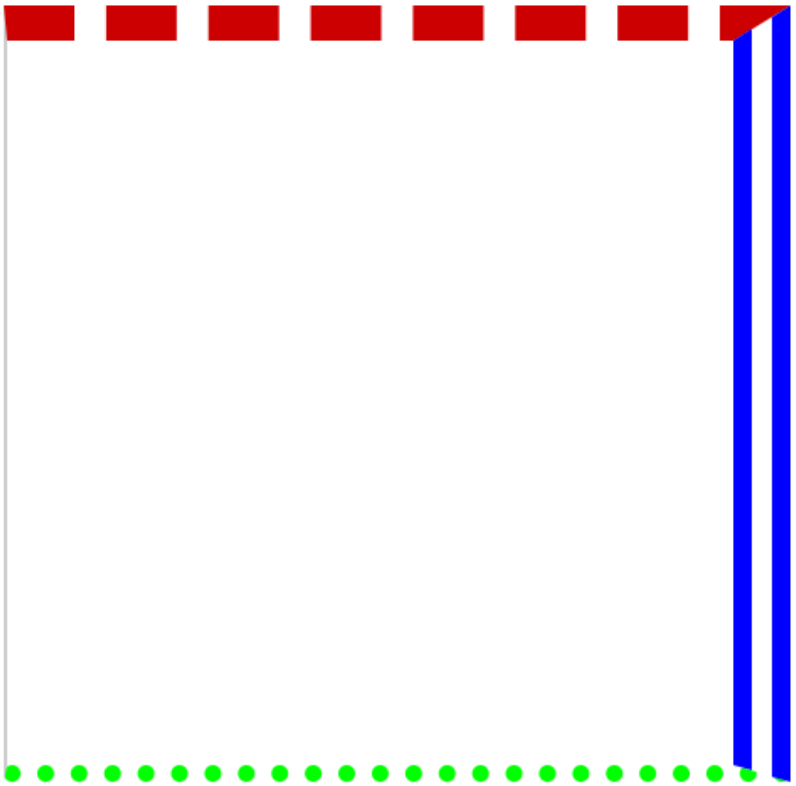
Estilo

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.

Siguiendo el ejemplo anterior, se puede modificar el estilo de cada uno de los bordes:



Las reglas CSS que se han añadido a las anteriores se muestran a continuación:

```
div {
  border-top-style: dashed;
  border-right-style: double;
  border-bottom-style: dotted;
  border-left-style: solid;
}
```

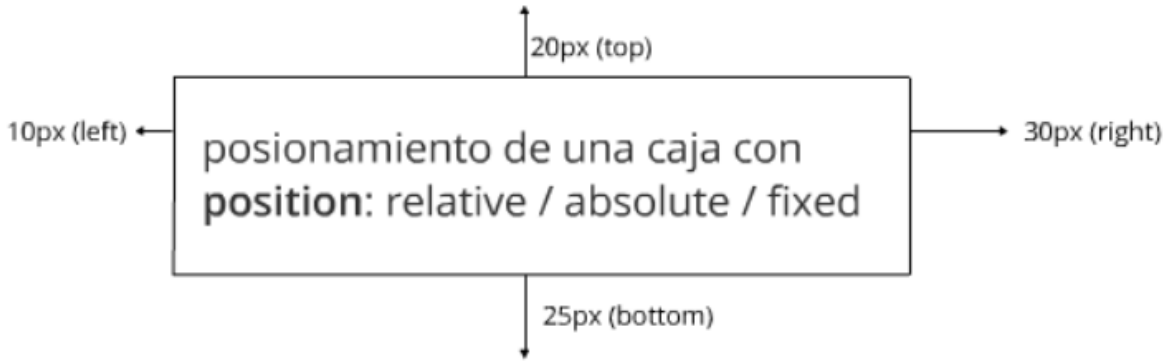
Los bordes más utilizados son **`solid`** y **`dashed`**, seguidos de **`double`** y **`dotted`**. Los estilos **`none`** y **`hidden`** son idénticos visualmente, pero se diferencian en la forma en la que los navegadores resuelven los conflictos entre los bordes de las celdas adyacentes en las tablas.

Para establecer de forma simultánea los estilos de todos los bordes de una caja, es necesario utilizar la propiedad *"shorthand"* llamada **`border-style`**:

La propiedad permite indicar de uno a cuatro valores diferentes y las reglas de aplicación son las habituales de las propiedades *"shorthand"*.

Posicionamiento

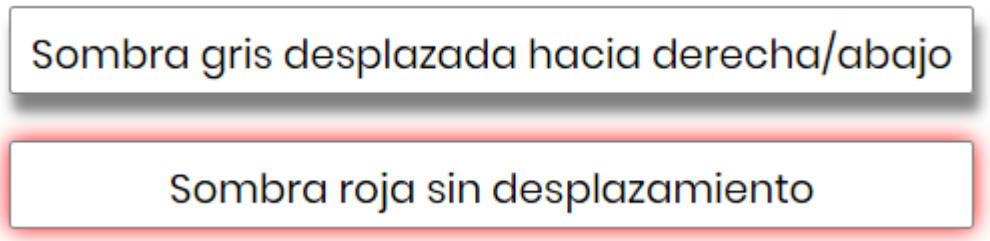
```
...
position: relative / absolute / fixed;
left:20px;
right:20px;
top:40px;
bottom:40px;
}
```



Para ubicar un ( ) en una coordenada concreta antes se tiene que definir su posicionamiento (**position**). Posteriormente, para indicar sus coordenadas se puede utilizar **left** o **right** para alinearlo horizontalmente desde la izquierda o derecha y/o **top** o **bottom** para alinearlo desde la parte superior o inferior.

Sombras

```
box-shadow: 2px 7px 5px gray;
box-shadow: 0 0 9px red;
```



Para definir una sombra a un ( ) se deben indicar 4 valores. El primero y segundo es la separación horizontal y vertical de la sombra (con un número positivo se desplaza hacia derecha/abajo y con uno negativo hacia la izquierda/arriba, mientras que con "0" se sitúa justo en la posición del texto). El tercer valor hace referencia a la suavidad del borde de la sombra, mientras que el último valor pertenece al color que tendrá la sombra.

Alinear (a la derecha)

Para poder colocar diferentes alineados en la misma línea podemos utilizar **display:inline-block** o **float**:left o right;.

En este último caso, deberemos también definir la anchura de cada caja.

```
width:230px;
float: left / right;
clear: both;
```

Para evitar que el que sigue en el HTML al último se coloque en el hueco vacío dejado en la línea anterior utilizamos **clear**:both;..

CSS que afecta a los enlaces

Eliminar subrayado

Elimina el subrayado que aparece por defecto en cualquier enlace.

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

Enlace no subrayado

Es necesario utilizar el selector **a{.** Utilizado sin indicar ningún elemento afectará a todos los enlaces. Utilizando **#div a{** afectará únicamente a los enlaces que se encuentran dentro de la caja con id="div"

Colores

Para definir el color que debe tener el enlace **cuando el cursor pase por encima** es necesario utilizar el evento **:hover**, detrás del selector **a**.

```
a{
  color: chocolate;
}
a:hover{
  color: red;
}
```

Enlace que cambia de color al pasar encima con el cursor

Enlace que cambia de color al pasar encima con el cursor

Cursor

Es posible definir un tipo de cursor (dentro de los existentes). Aunque los más habituales son "pointer" (el que aparece por defecto al crear un enlace) o "default" (el cursor habitual de la flecha), que en este caso aparece cuando el cursor pasa por encima del siguiente enlace de ejemplo.

```
a:hover{
  cursor: default;
}
```

Enlace en el que el cursor no cambia al pasar el cursor por encima

CSS que afecta a las listas

Viñetas personalizadas (list-style-type)

Permite establecer el tipo de viñeta mostrada para una lista.

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.

```
list-style-type: none;
```

Capitales de Europa

Berlín
Londres
Madrid
París

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.

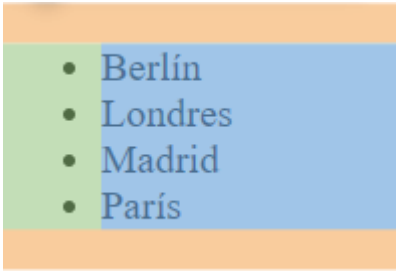
- Los valores gráficos son **`disc`**, **`circle`** y **`square`** y muestran como viñeta un círculo relleno, un círculo vacío y un cuadrado relleno respectivamente.
- Los valores numéricos están formados por **`decimal`**, **`decimal-leading-zero`**, **`lower-roman`**, **`upper-roman`**, **`armenian`** y **`georgian`**.
- Por último, los valores alfanuméricos se controlan mediante **`lower-latin`**, **`lower-alpha`**, **`upper-latin`**, **`upper-alpha`** y **`lower-greek`**.

list-style-position

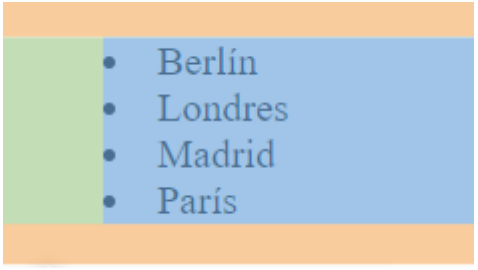
Permite establecer la posición de la viñeta de cada elemento de una lista.

El valor por defecto que aplican los navegadores es **`outside`**, que hace que el marcador se muestre fuera de la caja invisible que encierra al contenido del elemento. El otro valor disponible es **`inside`**, que hace que el marcador se muestre dentro de la caja invisible que encierra al contenido del elemento.

```
list-style-position: outside;
```




```
list-style-position: inside;
```



Si el contenido de los elementos de la lista ocupa menos de una línea, no se aprecian las diferencias visuales entre inside y outside. Sin embargo, cuando el contenido de un

elemento de la lista ocupa más de una línea, se aprecian claramente las diferencias visuales.

- **list-style-position: outside**
- Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nullam ultrices nibh a neque.
- Integer ac est quis turpis placerat varius. Sed tempor viverra quam. Praesent in lacus ac lorem scelerisque consectetuer.
- Vestibulum pellentesque pretium ligula. Pellentesque tincidunt. Sed sit amet dui.

- **list-style-position: inside**
- Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nullam ultrices nibh a neque.
- Integer ac est quis turpis placerat varius. Sed tempor viverra quam. Praesent in lacus ac lorem scelerisque consectetuer.
- Vestibulum pellentesque pretium ligula. Pellentesque tincidunt. Sed sit amet dui.

list-style-image

Permite reemplazar las viñetas automáticas por una imagen personalizada

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`**).

```
list-style-image: url("images/check.png");
```

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

Capitales de Europa

- Berlín
- Londres
- Madrid
- París

Cuando se utiliza una viñeta personalizada, es conveniente indicar la viñeta automática que se mostrará cuando no se pueda cargar la imagen:

```
list-style-image: url("images/check.png") square;
```

NOTA: CSS define una propiedad de tipo *"shorthand"* que permite establecer todas las propiedades de una lista de forma directa. La propiedad se denomina **`list-style`**.

list-style

Propiedad que permite establecer de forma simultánea todas las opciones de una lista.

```
list-style: (list-style-type || list-style-position ||
list-style-image );
```

En la definición anterior, la notación **||** significa que el orden en el que se indican los valores de la propiedad es indiferente. El siguiente ejemplo indica que no se debe mostrar ni viñetas automáticas ni viñetas personalizadas:

```
list-style: none
```

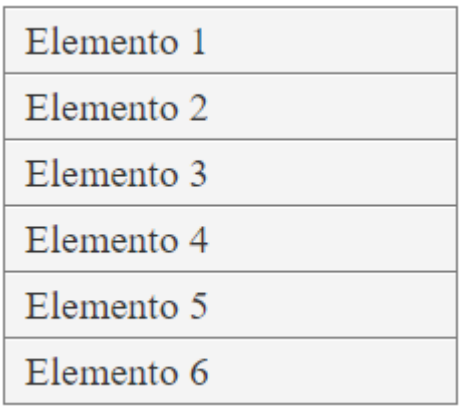
Menú vertical

Los sitios web correctamente diseñados emplean las listas de elementos para crear todos sus menús de navegación. Utilizando la etiqueta **<ul>** de HTML se agrupan todas las opciones del menú y haciendo uso de CSS se modifica su aspecto para mostrar un menú horizontal o vertical.

A continuación se muestra la transformación de una lista sencilla de enlaces en un menú vertical de navegación. Lista de enlaces original:

```
<ul>
 <li><a href="#">Elemento 1</a></li>
 <li><a href="#">Elemento 2</a></li>
 <li><a href="#">Elemento 3</a></li>
 <li><a href="#">Elemento 4</a></li>
 <li><a href="#">Elemento 5</a></li>
 <li><a href="#">Elemento 6</a></li>
</ul>
```

Aspecto final del menú vertical:



El proceso de transformación de la lista en un menú requiere de los siguientes pasos:

1. Definir la anchura del menú:

```
ul.menu { width: 180px; }
```

2. Eliminar las viñetas automáticas y todos los márgenes y espaciados aplicados por defecto:

```
ul.menu {
 list-style: none;
 margin: 0;
 padding: 0;
 width: 180px;
}
```

3. Añadir un borde al menú de navegación y establecer el color de fondo y los bordes de cada elemento del menú:

```
ul.menu {
 border: 1px solid #7C7C7C;
 border-bottom: none;
 list-style: none;
 margin: 0;
 padding: 0;
 width: 180px;
}
ul.menu li {
 background: #F4F4F4;
 border-bottom: 1px solid #7C7C7C;
 border-top: 1px solid #FFF;
}
```

4. Aplicar estilos a los enlaces: mostrarlos como un elemento de bloque para que ocupen todo el espacio de cada <li> del menú, añadir un espacio de relleno y modificar los colores y la decoración por defecto:

```
ul.menu li a {
 color: #333;
 display: block;
 padding: .2em 0 .2em .5em;
 text-decoration: none;
}
```

Menú horizontal básico

Partiendo de la misma lista de elementos del menú vertical, resulta muy sencillo crear un menú de navegación horizontal. La clave reside en modificar el posicionamiento original de los elementos de la lista.

Código HTML del menú horizontal:

```
<ul>
 <li><a href="#">Elemento 1</a></li>
 <li><a href="#">Elemento 2</a></li>
 <li><a href="#">Elemento 3</a></li>
 <li><a href="#">Elemento 4</a></li>
 <li><a href="#">Elemento 5</a></li>
 <li><a href="#" class="ultimo">Elemento 6</a></li>
</ul>
```

Aspecto final del menú horizontal:

Elemento 1	Elemento 2	Elemento 3	Elemento 4	Elemento 5

1. Aplicar los estilos CSS básicos para establecer el estilo del menú (similares a los del menú vertical anterior):

   ```
   ul.menu {
    background: #F4F4F4;
    border: 1px solid #7C7C7C;
    list-style: none;
    margin: 0;
    padding: 0;
   }
   ```

2. Establecer la anchura de los elementos del menú. Como el menú es de anchura variable y contiene cinco elementos, se asigna una anchura del 20% a cada elemento. Si se quiere tener un control más preciso sobre el aspecto de cada elemento, es necesario asignar una anchura fija al menú.

   Además, se posicionan de forma flotante los elementos de la lista mediante la propiedad **`float`**. Esta es la clave de la transformación de una lista en un menú horizontal::

   ```
   ul.menu li {
    float: left;
    width: 20%;
   }
   ```

Después de posicionar de forma flotante a todos los elementos de la lista, el elemento **<ul>** es un elemento vacío ya que en su interior no existe ningún elemento posicionado de forma normal. La solución de este problema consiste en aplicar la propiedad **overflow:hidden;** al elemento **<ul>,** de forma que encierre a todos los elementos posicionados de forma flotante:

```
ul.menu {
 overflow: hidden;
}
```

3. Establecer los bordes de los elementos que forman el menú:

```
ul.menu li a {
 border-left: 1px solid #FFF;
 border-right: 1px solid #7C7C7C;
 color: #333;
 display: block;
 padding: .3em;
 text-decoration: none;
}
```

4. Por último, se elimina el borde derecho del último elemento de la lista, para evitar el borde duplicado:

```
ul.menu li a.ultimo {
 border-right: none;
}
```

CSS que afecta a las tablas

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:


```
.normal {
 width: 250px;
 border: 1px solid #000;
}
.normal th, .normal td {
 border: 1px solid #000;
}
```

```
<table class="normal">
  <tr>
    <th>A</th>
    <th>B</th>
    <th>C</th>
    <th>D</th>
    <th>E</th>
  </tr>
  ...
</table>
```

El estándar CSS 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`**:

border-collapse

Define el mecanismo de fusión de los bordes de las celdas adyacentes de una tabla.

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:

A	B	C	D	E

```
.normal {
 width: 250px;
 border: 1px solid #000;
 border-collapse: collapse;
}
.normal th, .normal td {
 border: 1px solid #000;
}
```

Aunque parece sencillo, el mecanismo que utiliza el modelo **`collapse`** es muy complejo, ya que cuando los bordes que se fusionan no son exactamente iguales, debe tener en cuenta la anchura de cada borde, su estilo y el tipo de celda que contiene el borde (columna, fila, grupo de filas, grupo de columnas) para determinar la prioridad de cada borde.

Si se opta por el modelo **separate** (qué es el que se aplica si no se indica lo contrario) se puede utilizar la propiedad **border-spacing** para controlar la separación entre los bordes de cada celda.

border-spacing

Establece la separación entre los bordes de las celdas adyacentes de una tabla.

Si solamente se indica como valor una medida, se asigna ese valor como separación horizontal y vertical. Si se indican dos medidas, la primera es la separación horizontal y la segunda es la separación vertical entre celdas.

La propiedad **border-spacing** sólo controla la separación entre celdas y por tanto, no se puede utilizar para modificar el tipo de modelo de bordes que se utiliza. En concreto, si se establece un valor igual a **0** para la separación entre los bordes de las celdas, el resultado es muy diferente al modelo **collapse**:


```
.normal {
 width: 250px;
 border: 1px solid #000;
 border-spacing: 0;
}
.normal th, .normal td {
 border: 1px solid #000;
}
```

En la tabla del ejemplo anterior, se ha establecido la propiedad **border-spacing: 0**, por lo que el navegador no introduce ninguna separación entre los bordes de las celdas. Además, como no se ha establecido de forma explícita ningún modelo de bordes, el navegador utiliza el modelo **separate**.

El resultado es que **border-spacing: 0** muestra los bordes con una anchura doble, ya que en realidad se trata de dos bordes iguales sin separación entre ellos. Por tanto, las diferencias visuales con el modelo **border-collapse: collapse** son muy evidentes.

empty-cells

Define el mecanismo utilizado para el tratamiento de las celdas vacías de una tabla.

CSS define otras propiedades específicas para el control del aspecto de las tablas. Una de ellas es el tratamiento que reciben las celdas vacías de una tabla, que se controla mediante la propiedad **empty-cells**. Esta propiedad sólo se aplica cuando el modelo de bordes de la tabla es de tipo **separate**.

El valor **hide** indica que las celdas vacías no se deben mostrar. Una celda vacía es aquella que no tiene ningún contenido, ni siquiera un espacio en blanco o un ** **.

La siguiente imagen muestra las diferencias entre una tabla normal y una tabla con la propiedad **empty-cells: hide**.

Tabla normal

A	B	C	D	E
X	X		X	X

Tabla con **empty-cells: hide;**

A	B	C	D	E
X	X		X	X

```
.normal {
 width: 250px;
 border: 1px solid #000;
 border-spacing: 0;
}
.normal th, .normal td {
 border: 1px solid #000;
}
```

caption-side

Por otra parte, el título de las tablas se establece mediante la etiqueta **<caption>**, que por defecto se muestra encima de los contenidos de la tabla. La propiedad **caption-side** permite controlar la posición del título de la tabla.

El valor **bottom** indica que el título de la tabla se debe mostrar debajo de los contenidos de la tabla. Si también se quiere modificar la alineación horizontal del título, debe utilizarse la propiedad **text-align**.

A continuación se muestra el código HTML y CSS de un ejemplo sencillo de uso de la propiedad **caption-side:**

```
.especial {
  width: 250px;
  border: 1px solid #000;
  caption-side: bottom;
}
```

```
<table class="especial">
  <caption>Tabla 2.- Título especial</caption>
  <tr>
    <td>1</td>
    <td>2</td>
    <td>3</td>
  </tr>
</table>
```

La siguiente imagen muestra el resultado de visualizar el ejemplo anterior en cualquier navegador:

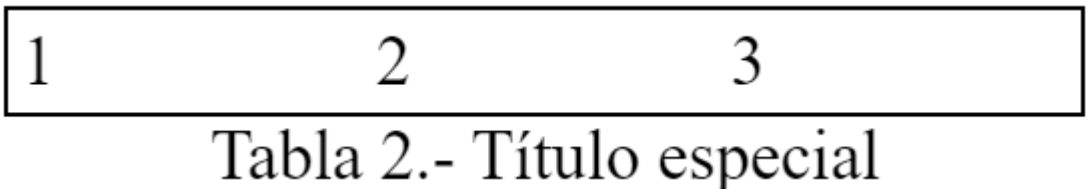


Tabla 2.- Título especial