



## Introducción a CSS3

### Que 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.

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.

### Enlazar Hojas de Estilo

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.

#### Incluirlo en el mismo documento

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>`).

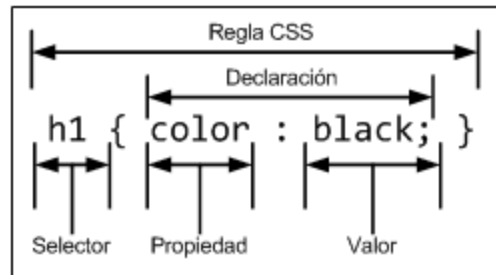
#### Desde un archivo externo

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.

```
<link rel="stylesheet" type="text/css" href="/css/estilos.css" media="screen" />
```

### Glosario Basico

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:

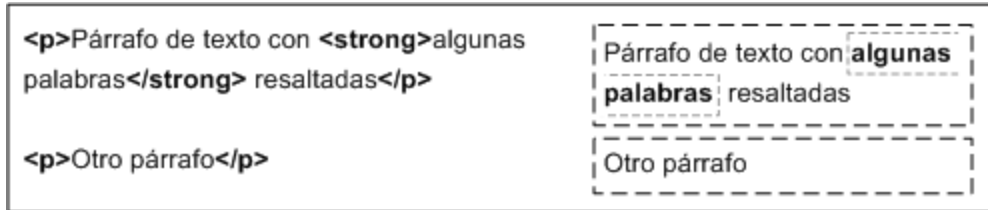
- **Regla:** cada uno de los estilos que componen una hoja de estilos CSS. Cada regla está compuesta de una parte de "selectores", un símbolo de "llave de apertura" (`{`), otra parte denominada "declaración" y por último, un símbolo de "llave de cierre" (`}`).
- **Selector:** indica el elemento o elementos 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.

## 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:



Las cajas de las páginas no son visibles a simple vista porque inicialmente no muestran ningún color de fondo ni ningún borde. Los navegadores crean y colocan las cajas de forma automática, pero CSS permite modificar todas sus características.

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 se muestra el color o imagen de fondo (si están definidos) y en el espacio ocupado por el margen se muestra el color o imagen de fondo de su elemento padre (si están definidos). Si ningún elemento padre tiene definido un color o imagen de fondo, se muestra el color o imagen de fondo de la propia página (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.

## Selectores Basicos

Visitar este link: [http://www.librosweb.es/css/capitulo2/selectores\\_basicos.html](http://www.librosweb.es/css/capitulo2/selectores_basicos.html)

## Posicionamiento

Los navegadores crean y posicionan de forma automática todas las cajas que forman cada página HTML. No obstante, CSS permite al diseñador modificar la posición en la que se muestra cada caja.

Utilizando las propiedades que proporciona CSS para alterar la posición de las cajas es posible realizar efectos muy avanzados y diseñar estructuras de páginas que de otra forma no serían posibles.

El estándar de CSS define cinco modelos diferentes para posicionar una caja:

- Posicionamiento normal o estático: se trata del posicionamiento que utilizan los navegadores si no se indica lo contrario.
- Posicionamiento relativo: variante del posicionamiento normal que consiste en posicionar una caja según el posicionamiento normal y después desplazarla respecto de su posición original.
- Posicionamiento absoluto: la posición de una caja se establece de forma absoluta respecto de su elemento contenedor y el resto de elementos de la página ignoran la nueva posición del elemento.
- Posicionamiento fijo: variante del posicionamiento absoluto que convierte una caja en un elemento inamovible, de forma que su posición en la pantalla siempre es la misma independientemente del resto de elementos e independientemente de si el usuario sube o baja la página en la ventana del navegador.
- Posicionamiento flotante: se trata del modelo más especial de posicionamiento, ya que desplaza las cajas todo lo posible hacia la izquierda o hacia la derecha de la línea en la que se encuentran.

El significado de cada uno de los posibles valores de la propiedad `position` es el siguiente:

- `static`: corresponde al posicionamiento normal o estático. Si se utiliza este valor, se ignoran los valores de las propiedades `top`, `right`, `bottom` y `left` que se verán a continuación.
- `relative`: corresponde al posicionamiento relativo. El desplazamiento de la caja se controla con las propiedades `top`, `right`, `bottom` y `left`.
- `absolute`: corresponde al posicionamiento absoluto. El desplazamiento de la caja también se controla con las propiedades `top`, `right`, `bottom` y `left`, pero su interpretación es mucho más compleja, ya que el origen de coordenadas del desplazamiento depende del posicionamiento de su elemento contenedor.
- `fixed`: corresponde al posicionamiento fijo. El desplazamiento se establece de la misma forma que en el posicionamiento absoluto, pero en este caso el elemento permanece inamovible en la pantalla.

La propiedad `position` no permite controlar el posicionamiento flotante, que se establece con otra propiedad llamada `float` y que se explica más adelante. Además, la propiedad `position` sólo indica cómo se posiciona una caja, pero no la desplaza.

## Tipografía

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.

---

```
h1 { color: #369; }  
p { color: black; }  
a, span { color: #B1251E; }  
div { color: rgb(71, 98, 176); }
```

---

La otra propiedad básica que define CSS relacionada con la tipografía se denomina `font-family` y se utiliza para indicar el tipo de letra con el que se muestra 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).

Los navegadores muestran el texto de las páginas web utilizando los tipos de letra instalados en el ordenador o dispositivo del propio usuario. De esta forma, si el diseñador indica en la propiedad `font-family` que el texto debe mostrarse con un tipo de letra especialmente raro o rebuscado, casi ningún usuario dispondrá de ese tipo de letra.

---

```
font-family: Arial, Helvetica, sans-serif;  
font-family: "Times New Roman", Times, serif;  
font-family: "Courier New", Courier, monospace;  
font-family: Georgia, "Times New Roman", Times, serif;
```

---

## Propiedad Font

El orden en el que se deben indicar las propiedades del texto es el siguiente:

- En primer lugar y de forma opcional se indican el `font-style`, `font-variant` y `font-weight` en cualquier orden.
- A continuación, se indica obligatoriamente el valor de `font-size` seguido opcionalmente por el valor de `line-height`.
- Por último, se indica obligatoriamente el tipo de letra a utilizar.

---

```
font: 76%/140% Verdana,Arial,Helvetica,sans-serif;  
font: normal 24px/26px "Century Gothic","Trebuchet MS",Arial,Helvetica,sans-serif;  
font: normal .94em "Trebuchet MS",Arial,Helvetica,sans-serif;  
font: bold 1em "Trebuchet MS",Arial,Sans-Serif;  
font: normal 0.9em "Lucida Grande", Verdana, Arial, Helvetica, sans-serif;  
font: normal 1.2em/1em helvetica, arial, sans-serif;
```

---

font: 11px verdana, sans-serif;  
font: normal 1.4em/1.6em "helvetica", arial, sans-serif;  
font: bold 14px georgia, times, serif;

---

## Listas

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`.

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`.

---

```
<ul style="list-style-type: square">  
  <li>list-style-type: square</li>  
  <li>Elemento</li>  
  <li>Elemento</li>  
</ul>
```

---

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.

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`).

---

```
ul.ok { list-style-image: url(imagenes/ok.png); }  
ul.go { list-style-image: url(imagenes/bullet_go.png); }
```

---

```
ul.redondo { list-style-image: url(imagenes/bullet_red.png); }
```

---

## 960 Grid System y Reset.css

960 Grid System es una librería de CSS enfocada a la maquetación de una página web. La librería utiliza un contenedor (div) principal de un tamaño fijo de 960 pixels, de ahí su nombre. Se eligió este tamaño por ser fácilmente divisible entre 2, 3, 4, 5, 6, 8, 10, 12, 15, 16, 20, 24, 30, 32, 40, 48, 60, 64, 80, 96, 120, 160, 192, 240, 320 y 480.

Una página se estructura usando la etiqueta HTML div. Primero se crea un contenedor principal que contendrá el resto de capas y estará dividido en 12 o 16 columnas. El número de columnas y cuántas ocuparán cada capa se define mediante el atributo class.

Para el div contenedor se pueden utilizar las clases container\_12 o container\_16 para especificar si contendrá 12 o 16 columnas.

Para las capas interiores se usa la clase grid\_X, donde X es el valor de columnas que ocupan. Con el siguiente código se crearía una página con una cabecera para un título que ocuparía 12 columnas. En la siguiente fila se crearían dos capas, una para un menú con una anchura de 4 columnas y otra para el contenido de 8 columnas. Al final habría una capa como pie de página ocupando las 12 columnas.

```
<div class="container_12">
  <div class="grid_12">Cabecera</div>
  <div class="grid_4">Menú</div>
  <div class="grid_8">Contenido</div>
  <div class="grid_12">Pie de página</div>
</div>
```

---

También existen las siguientes clases complementarias:

- alpha y omega: Se utilizan para crear capas dentro de capas.
- prefix\_XX y suffix\_XX: Para crear capas en blanco antes o después de la capa actual.

La biblioteca viene con el archivo reset.css, desarrollado por Eric Meyer, que sirve para eliminar algunos valores de **estilo preestablecido** que los navegadores web asignan a algunos elementos, evitando así ciertas inconsistencias de los mismos en la visualización del documento. El archivo (reset.css), si se enlaza como externo, debe añadirse antes de cualquier otro archivo CSS externo

```
<link href="ruta/reset.css" rel="stylesheet" type="text/css"/>
<link href="ruta/960.css" rel="stylesheet" type="text/css"/>
```

---

## **Referencias**

<http://www.librosweb.es/css/>

<http://www.mclibre.org/consultar/amaya/index.html>