

Desarrollo Web

Fundamentos de CSS



Índice

Introducción	3
Sintaxis de etiquetas CSS	4
¿Dónde va el código CSS?	4
Declaración externa	4
Declaración interna	4
Declaración online	5
Comentarios	5
Fuentes personalizadas	5
Modelo caja-contenedor	6
Float, margin y padding	7
Selectores	8
Selector de etiqueta	8
Selector de clase	8
Selector de ID	9
Selector de grupo	10
Selector universal	10
Selector hermano	10
Selector hijo	11
Estilos para texto	11
Colores	11
Colores RGBA	11
Degradados	12
Estilos para vínculos	12
Unidades en CSS	13
Longitudes absolutas	13
Longitudes relativas	14

Introducción

La razón de ser de **CSS** es dar a nuestras páginas una apariencia determinada.

Antes de la aparición de **CSS** lo que se hacía era escribir las etiquetas **HTML** con sus atributos. Por ejemplo, alto, ancho, color...etc.

Hoy en día separamos la estructura (**HTML**) de la apariencia (**CSS**).

Sintaxis de etiquetas CSS

Todas las etiquetas **CSS** tendrán dos partes:

- **Un selector:** Que determinará la zona de la página WEB que queremos modificar, es decir, la etiqueta a modificar.
- **Las instrucciones:** Hacen referencia al formato que vamos a aplicar a esa zona.

¿Dónde va el código CSS?

Para aplicar reglas **CSS** a nuestra página **HTML** tenemos tres opciones:

- **Declaración externa.** En un archivo independiente con extensión .CSS que estará “linkado” a nuestra página HTML. De esta forma podemos modificar todas las páginas de mi sitio web. Este es la mejor opción para mejorar el posicionamiento SEO.
- **Aplicación de estilos interna.** En la cabecera del documento HTML dentro del **head**. Así modificaremos la apariencia de una única página web en concreto.
- **Estilo in-line.** Dentro de una etiqueta. Con ello modificamos la apariencia del contenido de esa etiqueta. Opción desaconsejada, SEO no lo ve con buenos ojos. Lo ideal es usarlo solo en casos muy concretos.

Al aplicar estilos en línea, tenemos que repetir la aplicación de estilos en cada una de las líneas que queramos modificar y en cada ocasión afectamos únicamente a una línea. Con la aplicación de estilos interna nos basta con declarar una vez el estilo y el tipo de elemento al que se aplica, y automáticamente se aplicará a todos los elementos de ese tipo existentes dentro de la página web.

Declaración externa

Consiste en crear una página con extensión CSS y “linkarla” a nuestra página HTML.

Una vez creada la página CSS, nos vamos al head de nuestra página HTML y la enlazamos con una etiqueta <link>.

En nuestra cabecera, es decir, en el <head> de nuestra página, pondremos:

```
<link rel="stylesheet"
href="Hoja_Estilos.css">
```

Donde **Hoja_Estilos.css** será el archivo .css que le dará la apariencia a nuestra página web y donde pondremos las etiquetas CSS.

A partir de este momento podemos usar los selectores para hacer las modificaciones que nos interesen en las etiquetas de nuestra página HTML.

Este sería un ejemplo de <head> con etiqueta .css adjunta:

```
<head>
  <meta charset="UTF-8">
  <title>Ejemplo de HTML</title>
  <link rel="stylesheet"
href="Hoja_Estilos.css" >
</head>
```

Declaración interna

Para crear una hoja de estilos interna tendremos que usar la etiqueta <style></style> en el **head** de nuestra página HTML. Dentro crearemos nuestros estilos CSS usando los selectores disponibles en CSS.

```
<head>
  <meta charset="utf-8">
  <title>Pagina2</title>
  <meta name="Description"
content="Pagina2">
  <meta name="Keywords" content="Curso,
HTML, CSS, programacion WEB">
  <style type="text/css">
    h1 {
      text-align: center;
      font-family: "Lucida Grande";
      font-size: 32px
    }
  </style>
</head>
```

En esta ocasión le hemos dicho a **html** que las etiquetas **h1** deberán tener una alineación centrada, el tipo de letra será “*Lucida Grande*” y tendrá un tamaño de letra de 32px. De esta forma todas las etiquetas **h1** de nuestra página pasarán a tener dicho formato.

Declaración online

En esta ocasión usaremos las etiquetas **<style></style>** en cualquier etiqueta de nuestro documento HTML. El estilo que elijamos solo alterará dicha etiqueta.

```
<body>
  <h1 style="color: aqua">Titulo de
la página</h1>
  <h2>Subtítulo de la página</h2>
  <h2>Subtítulo2 de la página</h2>
</body>
```

En este ejemplo hemos modificado el estilo solo de una etiqueta **h1** en particular cambiando su color.

En caso de que una etiqueta se vea afectada por varios estilos CSS, el navegador aplicará el siguiente criterio:

Declaración en línea > Declaración interna > Declaración externa

Comentarios

En CSS, la sintaxis de los comentarios es;

```
/*Aquí va el comentario*/
```

Como en cualquier lenguaje de programación los comentarios serán ignorados a la hora de ejecutar el código. Son simplemente anotaciones que HTML no va a tener en cuenta.

Fuentes personalizadas

Si queremos usar un tipo de fuente específico que hay que descargar, tendremos que usar la propiedad **@font-face** y deberemos especificar la ruta donde se encuentra el archivo true-type de dicha fuente:

```
@font-face {
  font-family: 'miFuente';
  src: url('Bauhaus.ttf');
}
```

Si la fuente se encuentra en local, es decir, en nuestro disco duro, tiene que estar en la raíz. Si la ponemos en una carpeta no suele funcionar.

Y luego usaremos el tipo de fuente que hemos creado:

```
p{  
  font:bold 32px 'miFuente';  
}
```

En esta ocasión le hemos dicho a html que para las etiquetas `<p>` use esta fuente en concreto.

Modelo caja-contenedor

Existen dos tipos de etiquetas en HTML:

- **Etiquetas block.** Provocan un salto de línea por arriba y por debajo de la etiqueta y ocupan el 100% del ancho de la página. La mayoría de etiquetas de estructura son de este tipo.
- **Etiquetas in-line.** No generan salto de línea.

MODELO CAJA-CONTENEDOR

- Etiquetas BLOCK

Provocan un salto de página por arriba y por debajo de la etiqueta y ocupan el 100% del ancho de la página. La mayoría de etiquetas de estructura son de este tipo.

```
<h1>Titulo de mi página WEB</h1>  
<h2>Subtitulo de mi página WEB</h2>  
<p>Primer párrafo</p>  
<br>  
<p>Segundo párrafo</p>  
<div></div>  
<br>  
<p>Tercer párrafo</p>
```

- Etiquetas IN-LINE

No generan salto de línea. No se encuadran en un contenedor.

```
<strong> <a> <image>
```



Para que un elemento “**block**” pase a ser “**in-line**” utilizamos la propiedad:

`display: inline`

Float, margin y padding

A la hora de trabajar con la posición de los elementos de nuestra página HTML hay varias propiedades que hay que conocer:

Float: Altera el modo en el que se colocan los elementos en una página web. Al aplicar esta propiedad podremos poner un elemento en el lugar de la página que deseemos, ya no ocupará todo el ancho de página y el resto de elementos se distribuirán alrededor suyo en lugar de encima o debajo. Para hacer que el elemento vuelva a tener propiedades de caja-contenedor usamos la propiedad `clear`.

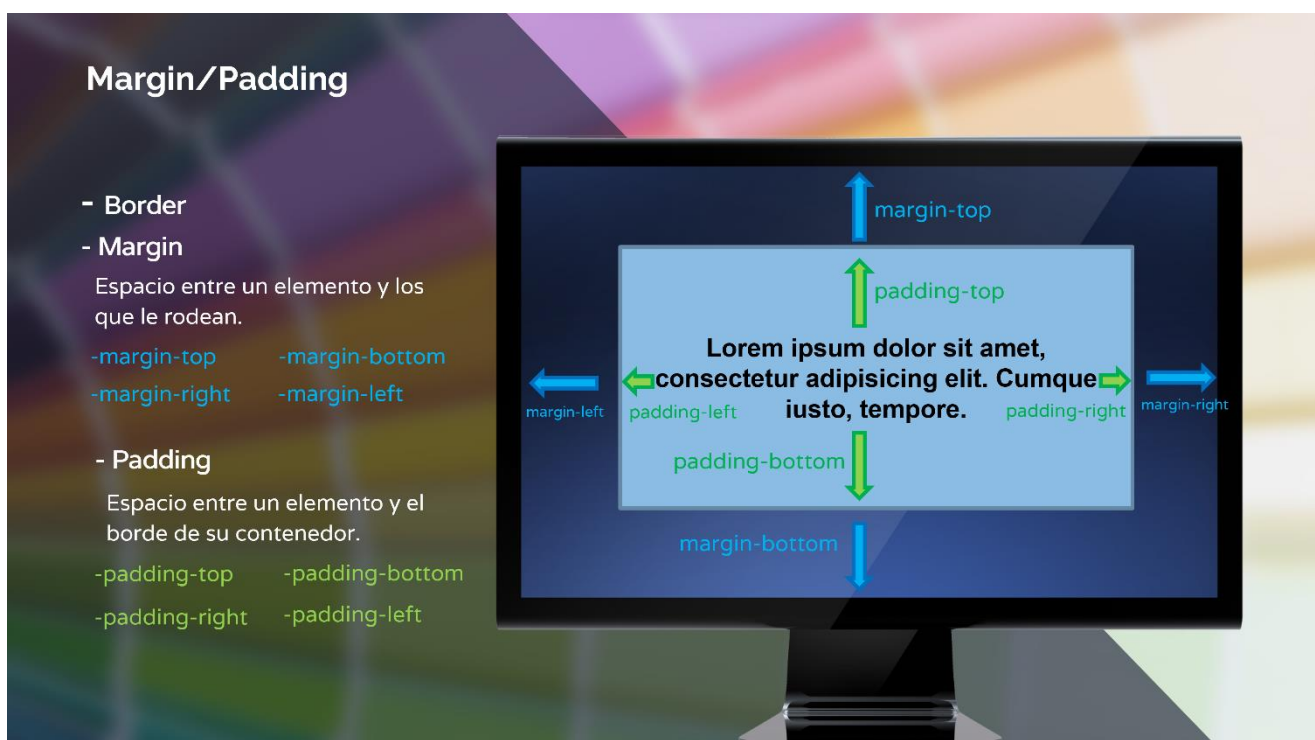
Margin: Establece el espacio entre el borde de la caja-contenedor y lo que pueda haber alrededor. (Los elementos con propiedad `float` no cuentan para él).

Todo elemento de nuestra página web tendrá un margen por arriba (`margin top`), por la derecha (`margin right`), por abajo (`margin bottom`) y por la izquierda (`margin left`).

Padding: Establece el espacio entre el contenido de la caja-contenedor y su borde.

De forma análoga a como ocurre con la propiedad `margin`, todo elemento de nuestra página web tendrá un padding por arriba (`padding top`), por la derecha (`padding right`), por abajo (`padding bottom`) y por la izquierda (`padding left`).

Border: Modifica el borde de un elemento, por ejemplo, una imagen, un cuadro de texto, etc.



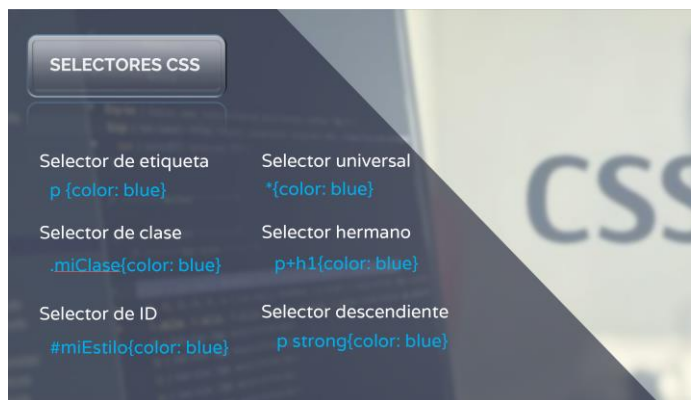
Selectores

En CSS, los selectores son patrones que se utilizan para seleccionar los elementos que desea modificar.

En CSS disponemos de los siguientes selectores.

- Selectores de etiqueta.
- Selectores de clase.
- Selectores ID.
- Selectores de grupo.
- Selector universal.

Existen más tipos de selectores, pero estos son los más usados.



A la hora de crear nuestros estilos, las instrucciones se dividen en dos grupos:

- **Selectores:** Sirven para especificar dentro de nuestra página *web*, a qué elementos de la misma vamos a aplicar el estilo.
- **Bloque de declaración:**

Selector ➡ `p{color:#3333}` ◀ Bloque de declaraciones.

Selector de etiqueta

Cambia el estilo de **todos** los elementos del tipo de etiqueta que le especifiquemos.

Puede ir en una hoja de estilos externa o interna, en la interna iría en el [head](#).

Su sintaxis en una hoja interna sería la siguiente:

```
<head>
  <meta charset="utf-8">
  <title>Pagina2</title>
  <meta name="Description"
content="Pagina2">
  <meta name="Keywords" content="Curso,
HTML, CSS, programacion WEB">
  <style type="text/css">
    p{color: aqua}
  </style>
</head>
```

En esta ocasión hemos cambiado el color todas las etiquetas `<p>` de nuestra página.

En una hoja de estilos externa (un archivo **CSS**):

```
p{color: aqua}
```

Esto igualmente nos cambiaría el estilo de todas las etiquetas `<p>` de la página.

Selector de clase

Con ellos aplicamos un formato de estilo a uno o varios elementos determinados. Puede ir en una hoja de estilos externa o en el [head](#) de una hoja de estilos interna, luego lo aplicamos en el elemento que lo necesite.

Comienza siempre por un punto ".", seguido del nombre que deseemos.

Nota: El nombre tiene que empezar por una letra y es case sensitive, es decir, distingue entre mayúsculas y minúsculas.

Hoja de estilos interna:

```
<head>
  <meta charset="utf-8">
  <title>Pagina2</title>
  <meta name="Description"
content="Pagina2">
  <meta name="Keywords" content="Curso,
HTML, CSS, programacion WEB">
  <style type="text/css">
    .miEstilo {color: aqua}
  </style>
</head>
```

Hoja de estilos externa:

```
.miEstilo {color: aqua}
```

Luego, en la etiqueta (o etiquetas) en cuestión ponemos;

```
<p class= "miEstilo ">Párrafo</p>
```

En este caso le hemos aplicado el estilo a una etiqueta `<p>`, si quisiésemos aplicarlo a una palabra suelta o cualquier otro elemento de la página que no lleva etiqueta lo haríamos poniendo la etiqueta ``. Esta etiqueta no hace nada, pero me permite poner un identificador y aplicarle estilo. Por ejemplo:

```
Vamos a resaltar una <spam class=
"miEstilo">palabra</spam> de nuestro
texto
```

Podemos asignar varias clases a un elemento:

```
<p class="clase1 clase2"></p>
```

Selector de ID

Es parecido al selector de etiqueta, pero este se aplica únicamente a un elemento en concreto para diferenciarlo de los demás. En lugar de comenzar por un punto, comienza por una `#`.

Podemos darle nombre, no es imprescindible.

Puede ir en una hoja de estilos externa o en el [head](#) de una hoja de estilos interna:

Hoja interna:

```
<head>
  <meta charset="utf-8">
  <title>Pagina2</title>
  <meta name="Description"
content="Pagina2">
  <meta name="Keywords" content="Curso,
HTML, CSS, programacion WEB">
  <style type="text/css">
    #miEstilo {color: aqua}
  </style>
</head>
```

En una hoja externa:

```
#miEstilo {color: aqua}
```

Luego, en la etiqueta (o etiquetas) en cuestión ponemos;

```
<p id= "miEstilo ">Párrafo</p>
```

Recordemos que **un elemento puede tener solo un ID y que cada ID ha de ser único.**

Selector de grupo

Con él, aplicamos unas características a todas las etiquetas que elijamos.

En nuestra hoja de estilos pondremos:

```
P[name="miEstilo"] {
  color: aqua
}
```

Luego, en la etiqueta (o etiquetas) en cuestión ponemos;

```
<p name="miEstilo">Párrafo</p>
```

Aquí hay tres modificadores disponibles para especificar a qué atributos se les aplicará el estilo. Irían inmediatamente detrás del **name**:

^ : Se aplica a todos los elementos que empiecen por...

```
<p name^= "mi">Párrafo</p>
```

\$: Se aplica a todos los elementos que terminen por...

```
<p name$= "ilo">Párrafo</p>
```

***** : Se aplica a todos los elementos que contengan...

```
<p name*= "Est">Párrafo</p>
```

Selector universal

Aplica las características que desees a todos los elementos a la página web.

```
<head>
  <meta charset="utf-8">
  <title>Pagina2</title>
  <meta name="Description"
content="Pagina2">
  <meta name="Keywords" content="Curso,
HTML, CSS, programacion WEB">
  <style type="text/css">
    * {color: aqua}
  </style>
</head>
```

En este caso hemos aplicado el color "aqua" a todos los elementos de nuestra página web.

Selector hermano

Modifica los elementos que estén a continuación del que yo le diga.

```
<head>
  <meta charset="utf-8">
  <title>Pagina2</title>
  <meta name="Description"
content="Pagina2">
  <meta name="Keywords" content="Curso,
HTML, CSS, programacion WEB">
  <style type="text/css">
    h1+p {color: aqua}
  </style>
</head>
```

Esto cambiaría el color de los párrafos **<p>** que estén a continuación de una etiqueta **<h1>**.

Selector hijo

Aplica un estilo a etiquetas que están a su vez dentro de otras etiquetas. Tomando como padre a la etiqueta contenedora y como hija a la etiqueta contenida dentro.

```
<head>
  <meta charset="utf-8">
  <title>Pagina2</title>
  <meta name="Description"
content="Pagina2">
  <meta name="Keywords" content="Curso,
HTML, CSS, programacion WEB">
  <style>
    Li a {color: aqua}
  </style>
</head>
```

Lo que conseguimos aquí es poner aplicar estilo solo a los vínculos `<a>` que estén dentro de una etiqueta ``, excluyendo el resto de los vínculos que pudiese haber en la página.

Nota: Si quiero darle estilo solo a una palabra o a un grupo de palabras dentro de un texto recordemos que tenemos que usar la etiqueta ``

Estilos para texto

Colores

Para cambiar los colores de un elemento de nuestra página web tenemos varias opciones:

- **Colores preestablecidos:** Accederemos directamente poniendo su nombre en inglés:
`p {color:blue}`
- **Rgb (rojo, verde, azul):** Donde rojo, verde y azul son números enteros desde 0 a 255 o porcentajes de 0% a 100%:
`p {color:rgb(42,44,156)}`
- **#RGB:** Donde R, G y B son números hexadecimales desde 0 hasta F:
`p {color: #353E9A}`

Colores RGBA

Los valores de color RGBA son una extensión de los valores de color RGB con un canal alfa, que especifica la opacidad del objeto.

Un color RGBA se especifica con la función `rgba()`, que tiene la siguiente sintaxis:

`rgba(red, green, blue, alpha).`

El parámetro alfa es un número entre 0,0 (totalmente transparente) y 1,0 (totalmente opaco).

```
#p1 {background-color: rgba(255, 0, 0,
0.3);} /*Rojo con transparencia*/
#p2 {background-color: rgba(0, 255, 0,
0.3) /*Verde con transparencia*/ #p3
{background-color: rgba(0, 0, 255, 0.3);}
/*Azul con transparencia*/
```

Degradados

Podemos aplicarle a un elemento un degradado de color. En este caso hay que especificar por orden, tipo de degradado, dirección del degradado, color inicial y color final.

```
div {
  background: linear-gradient(to
bottom, #33ccff 0%, #ff99cc 100%);
}

div {
  background: linear-gradient(to right,
#33ccff 0%, #ff99cc 100%);
}

div {
  background: linear-gradient(to top
left, #33ccff 0%, #ff99cc 100%);
}

div {
  background: linear-gradient(to top
right, #33ccff 0%, #ff99cc 100%);
}

div {
  background: linear-gradient(to top,
#33ccff 0%, #ff99cc 100%);
}

div {
  background: linear-gradient(to left,
#33ccff 0%, #ff99cc 100%);
}

div {
  background: linear-gradient(to bottom
left, #33ccff 0%, #ff99cc 100%);
}

div {
  background: linear-gradient(to bottom
left, #33ccff 0%, #ff99cc 100%);
}
```

Estilos para vínculos

Una etiqueta de vínculo `<a>`;

```
<a href="http://www.google.es">Ir a
google</a>
```

Puede tener diferentes estados:

- **Link**; En reposo, solo se están mostrando en la página, no estamos pasando el ratón por encima, etc.
- **Visited**; Cuando el vínculo ya ha sido visitado y se encuentra en la caché del navegador.
- **Hover**; Cuando se pasa el ratón por encima. Este se puede usar también en otras partes de la página, por ejemplo, para resaltar un párrafo cuando se pase el botón por encima.
- **Active**; Cuando se está haciendo "click".

Si se utilizan los cuatro estados a la vez hay que hacerlo en este orden, de lo contrario no funcionarán. (regla **LoVe HAte**). Es decir, **Link** => **Visited** => **Hover** => **Active**

La sintaxis para darle un estado al estilo de nuestro link:

```
a:estado{....}
```

```
a:visited{color:red}
```

Cambia a rojo el color de todos los vínculos ya visitados.

Unidades en CSS

CSS tiene varias unidades diferentes para expresar una longitud, píxeles, porcentajes y escala em. Revisa el vídeo de este tema para saber más sobre ellos.

Muchas propiedades CSS toman valores de "longitud", como [width](#), [margin](#), [padding](#), [font-size](#), etc.

La longitud es un número seguido de una unidad de longitud, como [10px](#), [2em](#), etc.

Se establecen diferentes valores de longitud, usando px (píxeles):

```
h1 {
  font-size: 60px;
}

p {
  font-size: 25px;
  line-height: 50px;
}
```

Nota: No puede aparecer un espacio en blanco entre el número y la unidad. Sin embargo, si el valor es [0](#), se puede omitir la unidad.

Para algunas propiedades de CSS, se permiten longitudes negativas.

Hay dos tipos de unidades de longitud: **absolutas** y **relativas**.

Longitudes absolutas

Las unidades de longitud absoluta son fijas y una longitud expresada en cualquiera de ellas aparecerá exactamente como ese tamaño.

Las unidades de longitud absoluta no se recomiendan para usar en pantalla, porque los tamaños de pantalla varían mucho. Sin embargo, se pueden utilizar si se conoce el medio de salida, por ejemplo, para el diseño de impresión.

Unidad	Descripción
cm	centímetros
mm	milímetros
in	Pulgadas (1in = 96px = 2.54cm)
px *	píxeles (1px = 1/96th of 1in)
pt	Puntos (1pt = 1/72 of 1in)
pc	picas (1pc = 12 pt)

Longitudes relativas

Las unidades de longitud relativa especifican una longitud relativa a otra propiedad de longitud. Las unidades de longitud relativa escalan mejor entre diferentes medios de renderizado.

Unidades	Descripción
em	Relativo al tamaño de fuente del elemento (2em significa 2 veces el tamaño de la fuente actual)
ex	En relación con la altura x de la fuente actual (rara vez se usa)
ch	Relativo al ancho del "0" (cero)
rem	Relativo al tamaño de fuente del elemento raíz
vw	Relativo al 1% del ancho de la ventana gráfica *
vh	Relativo al 1% de la altura de la ventana gráfica *
vmin	Relativo al 1% de la dimensión más pequeña de la ventana gráfica
vmax	Relativo al 1% de la dimensión más grande de la ventana gráfica
%	Relativo al elemento padre