



Curso de CSS

Índice

Prefacio	3
Introducción a CSS	4
Capítulos de contenidos del Curso	6
1. Primeros pasos en CSS	6
2. Sintaxis CSS	8
3. Atributos y valores	9
4. Los selectores CSS	10
5. Agrupación de los selectores CSS	13
6. Clases, pseudoclases y pseudoelementos	13
7. Bordes	15
8. Fondos.....	17
9. El texto	19
10.Márgenes.....	23
11.Cursores.....	24
12.Transparencias y CSS	26
13.Las reglas AT	27
14.La regla @import.....	28
15.Tipos de medio	29
Más documentación sobre CSS	31

Prefacio

En este curso te expondremos las características y funcionamiento de las **Hojas de Estilo en Cascada (CSS, *Cascading Style Sheets*)**.

Para lograr que comprendas correctamente este lenguaje comenzaremos con una explicación sobre las CSS, haciendo énfasis en su completo funcionamiento e importancia.

Hemos decidido, como es costumbre en nuestros cursos y guías, separar los contenidos del curso en capítulos. Además este curso posee un glosario de términos para proporcionarte el significado de palabras o términos, a consideración nuestra, de difícil entendimiento.

Aclaremos que para poder entender correctamente este curso deben poseer solo conocimientos muy básicos sobre los lenguajes de programación y en especial el lenguaje HTML.

Igualmente para este curso no es necesaria la utilización de un software en particular. Si desea puede utilizar algún editor de texto como el Bloc de notas de Windows, pero en lo personal recomendamos el editor de texto Notepad++ pues es mucho más cómodo para todo lo relacionado con lenguajes de programación. También si desea tener un mejor control y visualización del código y diseño, recomendamos **Dreamweaver**. Esta aplicación **Adobe** está destinada a la construcción, diseño y edición de sitios y aplicaciones Web.

Introducción a CSS

Comenzaremos tu familiarización con este lenguaje por una breve introducción, para presentar las CSS y posteriormente analizar y comprender su utilización y ventajas.

¿Qué es CSS?

Las CSS son un conjunto de instrucciones que permiten definir la apariencia de un sitio web, compuesto por diferentes elementos HTML. CSS posee una forma de representación de los estilos que permite un control total sobre la apariencia y posición de todos los elementos en una página web.

Algo historia sobre CSS

CSS ha sido creado en varios niveles y perfiles. Cada nivel de CSS se construye sobre el anterior, generalmente incorporando funciones al nivel previo.

Los perfiles son, generalmente, parte de uno o varios niveles de CSS definidos para una determinada interfaz.

- **CSS1**

La primera especificación oficial de CSS fue publicada en diciembre 1996 y fue abandonada en abril de 2008.

- **CSS2**

La especificación CSS2 fue publicada como recomendación en mayo de 1998, y abandonada en abril de 2008.

- **CSS 2.1**

CSS 2.1 fue la primera revisión de CSS2, usualmente conocida como "CSS 2.1". Esta versión corrige algunos errores encontrados en CSS2, elimina funcionalidades poco soportadas o inoperables en los navegadores y añade alguna nueva especificación.

De acuerdo al sistema de estandarización técnica de las especificaciones, CSS2.1 tuvo el estatus de "candidato" durante varios años, pero la propuesta fue rechazada en junio de 2005.

En junio de 2007 fue propuesta una nueva versión candidata, actualizada en 2009, pero en diciembre de 2010 fue nuevamente rechazada.

En abril de 2011, CSS 2.1 volvió a ser propuesta como candidata, y después de ser revisada, fue finalmente publicada como recomendación oficial el 7 de junio de 2011.

- **CSS3**

En diferencia a CSS2, que fue una gran especificación que definía varias funcionalidades, CSS3 está dividida en varios documentos separados, llamados "módulos".

Cada módulo añade nuevas funcionalidades a las definidas en CSS2, de manera que se preservan las anteriores para mantener la compatibilidad.

Los trabajos en el CSS3, comenzaron a la vez que se publicó la recomendación oficial de CSS2, y los primeros borradores de CSS3 fueron liberados en junio de 1999.

Capítulos de contenidos del Curso

Continuamos con tu familiarizarnos con las CSS, de manera sencilla para comprender correctamente las Hojas de Estilo en Cascada.

1. Primeros pasos en CSS

Primeramente mostraremos como aplicar estilos a elementos HTML, lo cual es algo básico y muy importante para comenzar a familiarizarse con el funcionamiento de las CSS.

- Aplicar estilo CSS a un elemento HTML

Para aplicar un estilo CSS directamente a un elemento, utilizamos el parámetro **style**.

Ejemplo de utilización del parámetro **style**:

```
1 <input style="border: 1px solid black;" type="text">
2
```

- Agrupar estilos entre etiquetas **style** asignando cada estilo a un elemento.

También podemos agrupar los estilos entre las etiquetas **style** y asignar cada estilo a un elemento usando [selectores](#).

Ejemplo de documento HTML con etiquetas **style**:

```
1 <html>
2   <head>
3     <title>Ejemplo 1</title>
4     <style>
5       input {
6         border: 2px solid blue;
7       }
8     </style>
9   </head>
10  <body>
11    <input type="text">
12  </body>
13 </html>
```

style.html

- Otra forma para aplicar estilos a una página HTML.

Otra forma para aplicar estilos a páginas HTML es crear un archivo **.css** donde guardaremos todos los estilos del documento para luego cargar o llamar a este fichero en las páginas HTML que se le quiera aplicar un estilo determinado.

Esta manera de incorporar estilos es la recomendada por nosotros pues así tendríamos todos los estilos, de nuestra página, en un solo archivo asegurando un estilo uniforme para nuestra aplicación web.

Ejemplo de un archivo CSS:

```
1
2 input {
3   border: 1px solid black;
4 }
```

style.css

- A continuación veremos como incluir un archivo CSS (**style.css**) en una página HTML.

```
1 <html>
2   <head>
3     <title>Ejemplo 2</title>
4     <link rel="stylesheet" type="text/css" href="style.css">
5   </head>
6   <body>
7     <input type="text">
8   </body>
9 </html>
```

style.html

2. Sintaxis CSS

La sintaxis de un archivo CSS está compuesta por 3 elementos:

- **Atributos** - Palabras que identifican un estilo (van seguido de dos puntos y el valor que se le asigna y separados por coma).
- **Valores** - Definen a los atributos.
- **Selectores** - Indican los elementos a los que se les va a aplicar el estilo.

- **Nota:**

Cuando se está trabajando fuera de un objeto, es decir, no se está usando el parámetro **style=""**, se tiene que indicar sobre que elemento de la página se quiere aplicar el estilo, esto se realiza a través de **selectores**.

Los selectores van seguidos de los atributos encerrados en llaves.

Ejemplo de un selector:

```
1 input {
2   color: #000000;
3   background: #FFFFFF
4 }
```


En el ejemplo anterior se le está aplicando el estilo a todos los elementos input. Asignando color negro (`#000000`) y fondo blanco (`#FFFFFF`).

▪ Nota:

Los valores aplicados al atributo, dependerán de este. Estos valores serán abordados más adelante.

3. Atributos y valores

Atributos

Como observamos en la parte de la sintaxis de CSS, los atributos y valores en son fáciles de utilizar. Solo indicamos el nombre del atributo seguido de dos puntos y el valor que tomará el atributo.

Valores

Los valores en CSS pueden ser numéricos, booleanos o cadena de caracteres.

Ejemplo de valores de diferentes formatos:

```
3 border: 0;  
4 border: 1px solid black;  
5 border: false;
```

En el ejemplo anterior se muestran valores CSS, de formatos diferentes.

En el segundo valor "**1px solid black**", los valores numéricos están acompañados por una unidad de medida, que puede ser una unidad relativa (`%`, `em`, `ex`) o una unidad métrica (`px`, `pt`, `pc`, `in`, `mm`, `cm`).

4. Los selectores CSS

Un selector es la parte de un estilo CSS donde se indica sobre que parte del documento (X) HTML se debe aplicar el estilo, resumiendo, se podría decir que es el nombre del estilo

Ejemplo:

```
1 div {  
2     font-size: 18px;  
3 }
```

En la anterior regla el selector sería **div** y para ser un poco más técnicos, **font-size** sería el atributo o propiedad y **18px** el valor, a su vez estos dos últimos junto con las llaves de inicio y cierre serían la declaración.

Disponemos de diferentes tipos de selectores que como es obvio actúan de forma diferente:

- Selectores.
- Selectores Descendientes.
- Selectores Hermano Adyacentes.
- Selectores de Atributos.
- Selectores de clases.
- Selectores de Id.
- Selectores Hijos.

A continuación explicaremos cada uno con su respectivo ejemplo.

Pero antes de empezar y para los usuarios más novatos, sería conveniente entender el concepto de **Elemento Padre** y **Elemento Hijo** si ya lo conoces sáltate este párrafo.

Cuando hablamos de elemento nos referimos a una etiqueta (X)HTML como pueda ser **span** o **div** por lo que si le buscas la lógica el padre es la etiqueta principal que dentro de

ella tiene otras etiquetas, por ejemplo **body**, siempre será un padre y en un código como el siguiente:

```
1 <div><strong>Padre e hijo...</strong></div>
2
```

El padre sería **div** y **strong** el hijo puesto que **strong** está dentro de **div**. A su vez un hijo puede tener otros hijos y así de forma jerárquica.

Selectores

Simplemente selectores, es lo más básico y que ya se ha insinuado al inicio del artículo. Los selectores hacen referencia a etiquetas de nuestro documento (X) HTML tal y como se ha mostrado en el primero ejemplo de selector:

```
1 div {
2     font-size: 18px;
3 }
```

Esto indicaría que todos los elementos **div** de nuestro documento (X) HTML utilizarían el estilo marcado por este código (tamaño de letra a 18px).

Selectores Descendientes

Este selector afectaría a una etiqueta hijo de un padre de forma descendiente:

```
1 div span{
2     font-size: 18px;
3 }
```

Por lo que todos los elementos **span** de nuestro documento (X) HTML que estén dentro de un elemento **div** se verán afectados por esta regla.

Ejemplo:

```
1 <head>
2   <style type="text/css">
3     div span { font-size: 18px; }
4   </style>
5 </head>
6 <body>
7   <span>En un lugar de la Mancha</span>
8   <div>
9     <span>De cuyo nombre no puedo acordarme</span>
10  </div>
11 </body>
```

En el ejemplo anterior vemos como en el resultado se ven dos frases, una con tamaño normal y otra (la segunda) con un tamaño de 18px.

Selectores hermano Adyacentes

Quizá sea el tipo de selector menos utilizado pero vamos a analizarlo. A diferencia de otros selectores este no se aplica ni a padres ni a hijos sino a hermanos.

```
1 h5 + h6 {
2   font-size: 18px;
3 }
```

Esta regla afectaría únicamente a **h6** pero solo cuando este se encontrara inmediatamente después de un **h5** o cuando **h5** y **h6** sean hijos del mismo padre.

Ejemplo:

```
1 <head>
2   <style type="text/css">h5 + h6 {font-size:18px}</style>
3 </head>
4
5 <body>
6   <h5>En un lugar de la mancha</h5>
7   <h6>De cuyo nombre no puedo acordarme</h6>
8 </body>
```

Efectivamente **h6** aparecerá con letra de 18 pixeles pero si entre **h5** y **h6** colocamos algún otro elemento como un párrafo **p** el selector hermano "adyacente" dejará de aplicarse a **h6**.

5. Agrupación de los selectores CSS

En un documento podemos encontrar elementos con un mismo estilo, por ejemplo el color. En estos casos podemos utilizar la coma (,) para agrupar selectores para cada elemento y definir el estilo una sola vez.

Ejemplo de selectores agrupados (*En este caso se agrupan para que los títulos <h1>, <h2> y <h3> sean de color azul (blue)*):

```
1 h1, h2, h3 {
2   color: blue;
3 }
```

6. Clases, pseudoclases y pseudoelementos

Clases

Una clase es una serie de propiedades y métodos que se agrupan en una estructura para poder aplicarlos conjuntamente a una serie de objetos, como CSS no tiene métodos, en

este caso la clase solo contendrá propiedades (atributos) y decidiremos que objetos pertenecen a esta clase con el atributo HTML **class**.

La clase se definirá como expliqué en [Sintaxis CSS](#), utilizando un selector de clase y poniendo todos los atributos y valores entre llaves.

Ejemplo:

```
1 <html>
2   <head>
3     <title>Mi CSS</title>
4     <style>
5       .miClase {
6         border: 1px solid red;
7         padding: 4 4 4 4;
8       }
9     </style>
10  </head>
11  <body>
12    <b>Menú</b>
13    <div class="miClase">Indice</div>
14    <div class="miClase">Descargas</div>
15    <div class="miClase">Links</div>
16  </body>
17 </html>
```

En el ejemplo anterior vemos que todos los objetos **div** están en la clase **miClase**, por lo tanto todos estos objetos tendrán las mismas propiedades.

Pseudoclases

Las pseudoclases son unas clases especiales de HTML, que se refieren a algunos estados especiales del elemento, las pseudoclases, igual que los pseudoelementos, se escriben como se muestra en el siguiente ejemplo:

```
1 objeto:pseudoclase {
2   propiedad: valor;
3 }
```

Las principales pseudoclasas son para el objeto A (vínculo), y son las siguientes:

- **link** - Enlace sin explorar.
- **visited** - Enlace que ha sido seguido por el usuario.
- **active** - Enlace seleccionado por el usuario.
- **hover** - Enlace que tiene el ratón encima

Pseudoelementos

Los pseudoelementos son partes determinadas de un objeto, y tenemos las siguientes:

- **first-letter** - Se refiere a la primera letra del objeto.
- **first-line** - Se refiere a la primera línea del objeto.

De esta manera, con las pseudoclasas y pseudoelementos, podemos definir una letra capital para la primera fila de cada párrafo, un enlace que se recuadra si le ponemos el mouse encima.

7. Bordes

El borde de un objeto, es el espacio que lo limita, y puede ser o no ser visible, CSS nos permite una gran cantidad de bordes que podemos escoger entre diferentes tipos, grosores y colores de la siguiente manera:


border: grosor tipo color;

Y además también podemos definir un borde determinado:


- Superior: - **border-top: grosor tipo color;**
- Inferior: - **border-bottom: grosor tipo color;**
- Izquierdo: - **border-left: grosor tipo color;**
- Derecho: - **border-right: grosor tipo color;**

Grososres

El grosor de un borde se define en píxeles, si definimos un borde de dos píxeles haremos:

```
 2 <div style="border: 2px solid black;">Ejemplo</div>
```

Por otro lado, si no queremos que el borde sea visible, le daremos grosor 0:

```
 2 <div style="border: 0;">Ejemplo</div>
```

Tipos

El tipo de borde define el aspecto del mismo, ya puede ser a rayas, continuo, punteado... a elegir entre la siguiente lista:

- **none** - Objeto sin borde
- **hidden** - Borde del objeto oculto
- **dotted** - Borde punteado
- **dashed** - Borde discontinuo
- **solid** - Borde continuo
- **double** - Borde doble
- **groove** - Borde con sombra oscura
- **ridge** - Borde con sombra clara
- **inset** - Borde 3D hacia afuera
- **outset** - Borde 3D hacia adentro

Por ejemplo, podemos hacer un borde de puntos de 4 pixeles:

```
<div style="border: 4px dotted black;">Ejemplo</div>
```

Colores

Es el color del borde, puede ser cualquier color HTML:

```
<div style="border: 4px dotted #ff0000;">Ejemplo</div>
```

Y este es el resultado del ejemplo:

Ejemplo

8. Fondos

Con CSS se pueden personalizar los fondos de cada objeto de nuestra página, permitiéndonos escoger el color de fondo, una imagen, la posición de la imagen, el comportamiento, y las repeticiones de la misma.

El color de fondo

El color de fondo se define usando el atributo **background-color** y un color RGB como valor.

```
<div style="background-color: #ff0000;">Ejemplo</div>
```

La imagen de fondo

Podemos definir un objeto con una imagen de fondo usando el atributo **background-image** y cómo valor una forma del estilo url (imagen.jpg), por ejemplo:

```
<div style="background-image: url(imagen.gif);">Ejemplo</div>
```

La posición del fondo

La imagen de fondo se puede alinear a la parte superior o inferior, para esto usaremos el atributo **background-position**, indicando uno o más de los siguientes valores:

- **top** - Alinear en la parte superior
- **bottom** - Alinear en la parte inferior
- **center** - Alinear en el medio
- **left** - Alinear a la izquierda
- **right** - Alinear a la derecha

Vamos a ver un ejemplo de imagen de fondo alineada en la parte superior derecha:

```
<div style="background-position: top right;">Ejemplo</div>
```

El comportamiento

El comportamiento de la imagen de fondo cuando usamos las barras de desplazamiento también se puede definir con el atributo **background-attachment**, que puede ser:

- **scroll** - La imagen se mueve con el interior del objeto
- **fixed** - La imagen no se mueve

Podemos hacer que el fondo no se desplace al desplazar la página:

```
2 <body style="background-image: url(bg_BG.gif); background-attachment: fixed"></body>
```

Las repeticiones

Podemos hacer que la imagen de fondo se repita usando el atributo **background-repeat**:

- **repeat** - La imagen se repite vertical y horizontalmente
- **repeat-x** - La imagen se repite horizontalmente
- **repeat-y** - La imagen se repite verticalmente
- **no-repeat** - La imagen no se repite

Podemos hacer que el fondo no repita en la página:

```
2 <body style="background-image: url(bg_BG.gif); background-attachment: fixed"></body>
```

El atributo background

El atributo **background** es una mezcla de los cinco anteriores y nos permite usar todos sus posibles valores para definir los fondos de los elementos rápidamente:

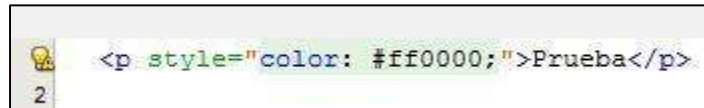
```
2 <body style="background: #ff0000 url(imagen.gif) repeat-x top fixed;">Ejemplo</body>
```

9. El texto

CSS nos permite definir muchos parámetros relativos al texto que pertenece a un determinado objeto, entre ellos, podemos definir el color del texto, el color del resaltado, su espaciado.

El color del texto

El color del texto, lo definimos dando de valor un color (por ejemplo RGB) al atributo **color**.

A code editor snippet showing the CSS attribute 'color' with the hex value '#ff0000' applied to a paragraph tag. The text 'Prueba' is shown as the content of the paragraph.

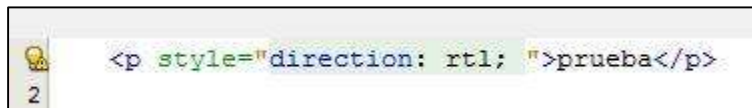
```
<p style="color: #ff0000;">Prueba</p>
```

La dirección del texto

El atributo **direction**, define si el texto se escribe de derecha a izquierda o de izquierda a derecha.

- **rtl** - De derecha a izquierda.
- **ltr** - De izquierda a derecha.

Veamos un ejemplo de texto de derecha a izquierda:

A code editor snippet showing the CSS attribute 'direction' with the value 'rtl' applied to a paragraph tag. The text 'prueba' is shown as the content of the paragraph.

```
<p style="direction: rtl;">prueba</p>
```

El espacio entre letras

El atributo **letter-spacing**, indica el tamaño que debe separar las letras entre ellas, su valor puede ser *normal* o una unidad de medida:

A code editor snippet showing the CSS attribute 'letter-spacing' with the value '2px' applied to a paragraph tag. The text 'prueba' is shown as the content of the paragraph.

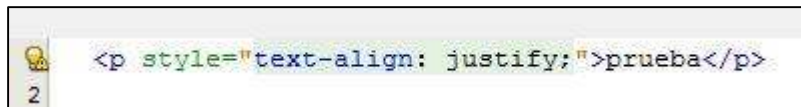
```
<p style="letter-spacing: 2px;">prueba</p>
```

La alineación del texto

El atributo ***text-align*** nos permite alinear el texto en de cualquiera de las siguientes maneras:

- **left** - Alinea el texto a la izquierda.
- **right** - Alinea el texto a la derecha.
- **center** - Alinea el texto en el centro.
- **justify** - Alinea el texto a ambos lados.

Veamos un ejemplo de texto alineado a ambos lados:



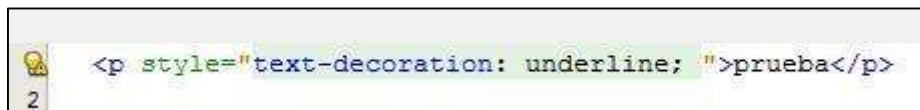
```
<p style="text-align: justify;">prueba</p>
```

La decoración del texto

Usando el atributo ***text-decoration*** podemos indicar las siguientes decoraciones:

- **none** - Sin decoración
- **underline** - Texto subrayado
- **overline** - Texto sobre rallado
- **line-through** - Texto tachado
- **blink** - Texto intermitente

Veamos un ejemplo que nos subrayará el texto:



```
<p style="text-decoration: underline;">prueba</p>
```

Grosor del texto

Usando el atributo **font-weight** podemos poner un texto en negrita:

```
<p style="font-weight: bold;">prueba</p>
```

El desplazado de la primera línea

Usando el atributo **text-indent** podemos hacer que la primera línea del texto aparezca ligeramente desplazada:

```
<p style="text-indent: 10px;">prueba</p>
```

Mayúsculas y minúsculas

Usando el atributo **text-transform** podemos hacer que el texto se vea en los siguientes modos:

- **none** - Como está escrito.
- **capitalize** - Todas las palabras empiezan por mayúsculas.
- **uppercase** - Todo en mayúsculas.
- **lowercase** - Todo en minúsculas.

Ejemplo de texto en mayúsculas:

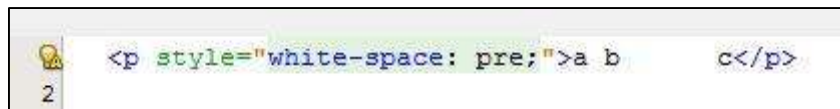
```
<p style="text-transform: capitalize;">prueba</p>
```

Los espacios

Con ***white-space*** podemos indicar cómo se interpretarán los espacios:

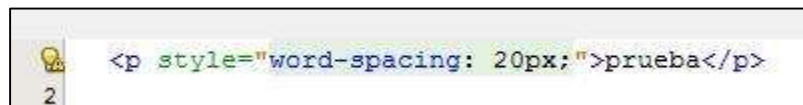
- **normal** - Los espacios en blanco són ignorados.
- **pre** - Se respetan los espacios en blanco.
- **nowrap** - No se puede cortar la línea por los espacios.

Ejemplo respetando espacios:

A screenshot of a code editor with a light blue background. It shows a single line of CSS code: `<p style="white-space: pre;">a b c</p>`. The code is color-coded: `<p` is blue, `style="white-space: pre;"` is green, `>a b c</p>` is black. On the left side of the editor, there is a yellow lightbulb icon and the number '2'.

El espacio entre palabras

El atributo ***word-spacing***, indica el tamaño que debe separar las palabras entre ellas, su valor puede ser *normal* o una unidad de medida:

A screenshot of a code editor with a light blue background. It shows a single line of CSS code: `<p style="word-spacing: 20px;">prueba</p>`. The code is color-coded: `<p` is blue, `style="word-spacing: 20px;"` is green, `>prueba</p>` is black. On the left side of the editor, there is a yellow lightbulb icon and the number '2'.

10. Márgenes

El margen de un objeto es el espacio que le separa de sus objetos adyacentes tanto por la parte superior e inferior como por ambos lados.

- **Definir el margen**

Para definir el margen entre un objeto y sus adyacentes utilizamos el atributo ***margin***, que tendrá como valor entre 1 y 4 unidades de medida.

Ejemplo:

```
1 <div style="margin: 4px;">ejemplo</div>
2
```

En este ejemplo, hemos usado sólo una unidad de medida (4px) pero podemos usar hasta 4 separadas por espacios:

```
1 <div style="border: 4px 4px 4px 4px;">ejemplo</div>
2
```

En este caso, serán respectivamente:

- margen superior: 4px 4px 4px 4px;
- margen derecho: 4px 4px 4px 4px;
- margen inferior: 4px 4px 4px 4px;
- margen izquierdo: 4px 4px 4px 4px;

El margen interior

De la misma manera que definimos el margen (por el exterior del [borde](#)) usando **padding** del mismo modo que usamos **margin**.

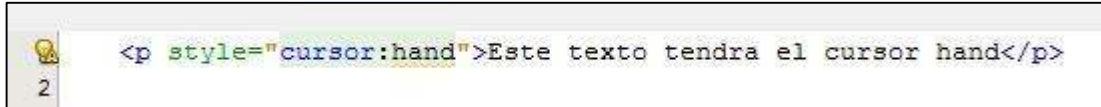
Ejemplo:

```
1 <div style="margin: 4px; padding: 10px; border: 1px solid red">ejemplo</div>
2
```

11. Cursores

Los estilos CSS nos permiten la configuración del cursor que se mostrará cuando el puntero se sitúe encima de un determinado objeto, que puede ir desde un vínculo al mismo objeto **body**.

Para configurar un cursor podemos usar distintos métodos, el más sencillo de ellos es usar el atributo **style**:

A screenshot of a code editor window. On the left, there is a lightbulb icon and the number '2'. The main area shows an HTML snippet: `<p style="cursor:hand">Este texto tendra el cursor hand</p>`. The text 'Este texto tendra el cursor hand' is highlighted in a light blue color.

El valor que daremos al estilo cursor (hand en el ejemplo) será uno de los nombres de la siguiente lista:

- **all-scroll** - Cursor con un punto en el centro y 4 flechas cardinales
- **auto** - Cursor por defecto del objeto
- **crosshair** - Puntero en forma de cruz
- **col-resize** - Flechas al este y oeste separadas por una barra vertical
- **default** - El cursor por defecto del objeto **body**
- **hand** - Puntero con forma de mano
- **help** - Cursor interrogante, típico de ayuda
- **move** - Mouse con forma de cruz aspada
- **no-resize** - Mano con una señal pequeña de prohibido
- **not-allowed** - Cursor circular con una barra diagonal
- **n-resize** - Forma de flecha, cambiaremos **n** por:
 - **e** - Flecha hacia el este
 - **ne** - Flecha hacia el noreste
 - **nw** - Flecha hacia el noroeste
 - **n** - Flecha hacia el norte
 - **sw** - Flecha hacia el suroeste
 - **s** - Flecha hacia el sur
 - **w** - Flecha hacia el oeste
- **pointer** - Puntero con forma de mano
- **progress** - Cursor default junto un reloj de arena

- **row-resize** - Cursor *col-resize* pero en vertical
- **text** - Cursor con forma de escritura
- **vertical-text** - Puntero en forma de escritura vertical
- **wait** - Cursor de espera (Reloj de arena)

▪ Nota:

Los cursores que aparecen a continuación solo funcionan en IE6:

- | | | |
|---------------------|------------------------|---------------------|
| ▪ all-scroll | ▪ hand | ▪ Progress |
| ▪ no-resize | ▪ vertical-text | ▪ row-resize |
| ▪ col-resize | ▪ not-allowed | |

Además, existe la posibilidad de definir un puntero propio para IE6, para ello usaremos:

```
<a href="index.php" style="cursor:url(http://www.google.com/favicon.ico)">Este texto tendra el cursor de google</a>
```

Para modificar el cursor de un objeto desde JavaScript, usaremos:

```
<script>
2   document.body.style.cursor = 'crosshair';
3 </script>
```

Este código cambia el cursor de la página por una cruz

12. Transparencias y CSS

La propiedad **opacity** introducida en CSS3 permite definir la transparencia de un elemento con valores entre 0 (transparente) y 1 (visible), de esta manera deberíamos poder definir un elemento como se muestra en el siguiente ejemplo.

Ejemplo:

```
1 elemento {  
2     opacity: 0.5;  
3 }
```

Pero esto no deja de ser "en teoría" ya que si bien hasta CSS3 no existía una propiedad estándar para las transparencias, los diferentes navegadores habían desarrollado sus propios métodos para conseguirlas.

Las versiones de Mozilla anteriores a la 1.7 soportan únicamente la propiedad **-moz-opacity** idéntica a **opacity** mientras que Internet Explorer dispone de un filtro para definir la opacidad en tanto por ciento.

```
1 elemento {  
2     opacity: 0.5;  
3     -moz-opacity: 0.5;  
4     filter: alpha(opacity=50);  
5 }
```

13. Las reglas AT

Las reglas AT, son agrupaciones de selectores que nos permiten aplicarlas en bloque a un objeto determinado, su sintaxis es parecida a la de un selector.

```
1 <style>  
2     @media print {  
3         .publicidad {  
4             visibility: hidden;  
5         }  
6     }  
7 </style>
```

Las reglas AT que podemos encontrar frecuentemente y que se explicarán en futuros artículos de este curso son:

- **font-face** - Define una fuente personalizada a partir de un conjunto de atributos, como pueden ser su tipo de fuente, su tamaño, un archivo ttf.
- **import** - Importa los estilos de una hoja CSS externa
- **media** - Especifica un conjunto de selectores para diferentes medios
- **page** - Especifica los márgenes y formato de hoja para medios paginados como puede ser la impresora

Debemos tener cuidado, porque algunas de estas reglas no están soportadas todavía por todos los navegadores, esto lo veremos cuando profundicemos en cada una de ellas.

14. La regla @import

La regla **import** es una regla que nos permite importar una hoja de estilos CSS externa en el interior de nuestras etiquetas **style**, de esta manera podemos aplicar los archivos de esa hoja en la página actual.

La sintaxis correcta de la regla **import** es la siguiente:


```
1 <style>
2   @import url("estilos.css");
3 </style>
```

El nombre entre dobles comillas el archivo que hay que cargar, hay una sintaxis abreviada y también aceptada que suprime la clave url, veamos el mismo ejemplo con esta otra sintaxis:

```
1 <style>
2   @import "estilos.css";
3 </style>
```

Con cualquiera de los dos casos, el navegador cargaría la hoja de estilos 'estilos.css' y aplicaría su contenido a la página actual.

Opcionalmente, podemos importar un archivo para un tipo de medio determinado, por ejemplo, si queremos una hoja de estilos para la impresora y otra para la página, podemos hacer:



```
<style>
2   @import url("impresora.css") print;
3   @import url("normal.css") screen;
4 </style>
```

Esto nos cargará un estilo diferente para la versión imprimida de la página.

15. Tipos de medio

Un medio describe el soporte en el que se muestra la página a la que se aplica el CSS en cuestión, podemos desear un estilo diferente según si vemos la página en la pantalla de un ordenador, en una hoja impresa.

Los usaremos para reglas cómo import o media para definir estilos alternativos para nuestras páginas.

Los medios existentes son:

- **all** - Para todos los medios.
- **aural** - Para sintetizadores de voz.
- **braille** - Para dispositivos táctiles en braille.
- **embossed** - Para impresoras braille.
- **handheld** - Para dispositivos de mano como pocket pc's, palms.
- **print** - Para la versión imprimible de las páginas.
- **projection** - Para presentaciones en proyector.
- **screen** - Para pantallas de ordenador.
- **tty** - Para teletipos, consola y otros medios de texto

- **tv** - Para pantallas de televisión

Internet Explorer, soporta los siguientes tipos de medio: ***all***, ***screen*** y ***print***.

Además existen grupos de medios que incluyen diversos medios, y son:

- **continous** - Incluye aural, braille, handheld, screen, tty y tv
- **paged** - Incluye emboss, handheld, print, projection y tv
- **visual** - Incluye handheld, print, projection, screen, tty y tv
- **aural** - Incluye aural y tv
- **tactile** - Incluye braille y emboss
- **grid** - Incluye braille, emboss, handheld y tty
- **bitmap** - Incluye handheld, print, projection, screen y tv
- **interactive** - Incluye todos menos print y projection
- **static** - Incluye todos

De esta manera, si usamos el tipo de medio bitmap, el estilo se aplicará para handheld, print, projection, screen y tv.

Más documentación sobre CSS

A continuación te muestro algo de documentación, a través de la cual podrás aumentar y consolidar tus conocimientos sobre las CSS, por supuesto, luego de haber estudiado completamente nuestra guía.

1. [CSS. Técnicas profesionales para el diseño moderno](#)
2. [Curso de CSS](#)
3. [CSS3 \(Guías Prácticas Usuarios\)](#)
4. [CSS para diseño Web \(Anaya Multimedia-Wrox\)](#)
5. [CSS \(Manual Imprescindible\)](#)
6. [CSS para diseño Web \(Anaya Multimedia-Wrox\)](#)
7. [Diseño Web con CSS](#)
8. [CSS \(Manual Avanzado\)](#)
9. [CSS - The View From The Inside](#)
10. [The CSS3 Anthology: Take Your Sites To New Heights, 4th Edition \(The CSS Anthology\)](#)