



Introducción a las CSS

El lenguaje HTML está limitado a la hora de aplicarle forma a un documento. Esto es así porque fué concebido para otros usos (científicos sobretodo), distinto a los actuales, mucho más amplios.

Para solucionar estos problemas los diseñadores han utilizado técnicas tales como la utilización de tablas imagenes transparentes para ajustarlas, utilización de etiquetas que no son estándares del HTML y otras. Estas "trampas" han causado a menudo problemas en las páginas a la hora de su visualización en distintas plataformas.

Además, los diseñadores se han visto frustrados por la dificultad con la que, aun utilizando estos trucos, se encontraban a la hora de maquetar las páginas, ya que muchos de ellos venian maquetando páginas sobre el papel, donde el control sobre la forma del documento es absoluto.

Finalmente, otro antecedente que ha hecho necesario el desarrollo de esta tecnología consiste en que las páginas web tienen mezclado en su código HTML el contenido del documento con las etiquetas necesarias para darle forma. Esto tiene sus inconvenientes ya que la lectura del código HTML se hace pesada y difícil a la hora de buscar errores o depurar las páginas. Aunque, desde el punto de vista de la riqueza de la información y la utilidad de las páginas a la hora de almacenar su contenido, es un gran problema que estos textos estén mezclados con etiquetas incrustadas para dar forma a estos: se degrada su utilidad.

En estas páginas de CSS pretendemos dar a conocer la tecnología con un enfoque práctico para que en pocos capítulos podáis usar las CSS de una manera depurada, reflejando toda nuestra experiencia en su uso. No pretendemos explorar todos los aspectos de la tecnología ya que para realizar esto necesitaríamos un la extensión de un libro entero.

Características y ventajas de las CSS

El modo de funcionamiento de las CSS consiste en definir, mediante una sintaxis especial, la forma de presentación que le aplicaremos a:

- Un web entero, de modo que se puede definir la forma de todo el web de una sola vez.
- Un documento HTML o página, se puede definir la forma, en un pequeño trozo de código en la cabecera, a toda la página.
- Una porción del documento, aplicando estilos visibles en un trozo de la página.
- Una etiqueta en concreto, llegando incluso a poder definir varios estilos diferentes para una sola etiqueta. Esto es muy importante ya que ofrece potencia en nuestra programación. Podemos definir, por ejemplo, varios tipos de párrafos: en rojo, en azul, con márgenes, sin ellos...

La potencia de la tecnología salta a la vista. Pero no solo se queda aquí, ya que además esta sintaxis CSS permite aplicar al documento formato de modo mucho más exacto. Si antes el HTML se nos quedaba corto para maquetar las páginas y teníamos que utilizar trucos para conseguir nuestros efectos, ahora tenemos muchas más herramientas que nos permiten definir esta forma:

- Podemos definir la distancia entre líneas del documento.
- Se puede aplicar indentado a las primeras líneas del párrafo.
- Podemos colocar elementos en la página con mayor precisión, y sin lugar a errores.
- Y mucho más, como definir la visibilidad de los elementos, márgenes, subrayados, tachados...

Y seguimos mostrandoos ventajas, ya que si con el HTML tan sólo podíamos definir atributos en las páginas con píxeles y porcentajes, ahora podemos definir utilizando muchas más unidades como:

- Pixels (px) y porcentaje (%), como antes.
- Pulgadas (in)
- Puntos (pt)
- Centímetros (cm)

Navegadores que lo soportan

Esta tecnología es bastante nueva, por lo que no todos los navegadores la soportan. En concreto, sólo los navegadores de Netscape versiones de la 4 en adelante y de Microsoft a partir de la versión 3 son capaces de comprender los estilos en sintaxis CSS. Además cabe destacar que no todos los navegadores implementan las mismas funciones de hojas de estilos, por ejemplo, Microsoft Internet Explorer 3 no soporta todo lo relativo a capas.

Esto quiere decir que debemos de usar esta tecnología con cuidado, ya que muchos usuarios no podrán ver los formatos que apliquemos a las páginas con CSS. Así pues, utilizad las hojas de estilos cuando estas no vayan a suponer un problema.

Usos de las CSS I

Vamos ahora a describir los diferentes usos de las CSS introducidos en el anterior capítulo. Vamos por orden, describiendo los puntos según su dificultad e importancia.

CSS tiene una sintaxis propia, la veremos a través de ejemplos.

Luego se verá con detalle

Hemos partido este capítulo en dos partes por su extensión y por haber varias formas distintas de aplicar estilos, aquí veremos las más sencillas y en el capítulo siguiente otras más complicadas pero más potentes.

Pequeñas partes de la página

Para definir estilos en secciones reducidas de una página se utiliza la etiqueta ****. Con su atributo **style** indicamos en sintaxis CSS las características de estilos. Lo vemos con un ejemplo, pondremos un párrafo en el que determinadas palabras las vamos a visualizar en color verde.

```
<p>
```

```
Esto es un párrafo en varias palabras <SPAN style="color:green">en color verde</SPAN>. resulta muy fácil.
```

```
</p>
```

Que tiene como resultado:

Esto es un párrafo con varias palabras **en color verde**. resulta muy fácil.

Estilo definido para una etiqueta

De este modo podemos hacer que toda una etiqueta muestre un estilo determinado. Por ejemplo, podemos definir un párrafo entero en color rojo y otro en color azul. Para ello utilizamos el atributo **style**, que es admitido por todas las etiquetas del HTML (siempre y cuando dispongamos de un navegador compatible con CSS).

```
<p style="color:#990000">
Esto es un párrafo de color rojo.
</p>
<p style="color:#000099">
Esto es un párrafo de color azul.
</p>
```

Que tiene como resultado:

Esto es un párrafo de color rojo.

Esto es un párrafo de color azul.

Estilo definido en una parte de la página

Con la etiqueta **<DIV>** podemos definir secciones de una página y aplicarle estilos con el atributo **style**, es decir, podemos definir estilos de una vez a todo un bloque de la página.

```
<div style="color:#000099; font-weight:bold">  
<h3>Estas etiquetas van en <i>azul y negrita</i></h3>  
<p>  
Seguimos dentro del DIV, luego permanecen los etilos  
</p>  
</div>
```

Que tiene como resultado:

Estas etiquetas van en azul y negrita

Seguimos dentro del DIV, luego permanecen los etilos

Hasta aquí este capítulo, en el siguiente seguiremos viendo formas más avanzadas de usar las CSS.

Usos de las CSS y II

Estilo definido para toda una página

Podemos definir, en la cabecera del documento, estilos para que sean aplicados a toda la página. Es una manera muy cómoda de darle forma al documento y muy potente, ya que estos estilos serán seguidos en toda la página y nos ahorraremos así muchas etiquetas HTML que apliquen forma al documento. Además, si deseamos cambiar los estilos de la página lo haremos de una sola vez.

Este ejemplo es más complicado, puesto que se utiliza la sintaxis CSS de manera más avanzada. Pero no te preocupes puesto que con los ejemplos irás aprendiendo su uso y más tarde comentaremos la sintaxis en profundidad.

En el ejemplo vemos que se utiliza la etiqueta **<STYLE>** colocada en la cabecera de la página para definir los distintos estilos del documento.

A grandes rasgos, entre de **<STYLE>** y **</STYLE>**, se coloca el nombre de la etiqueta que queremos definir los estilos y entre llaves **-{}-** colocamos en sintaxis CSS las características de estilos.

```

<html>
<head>
<title>Ejemplo de estilos para toda una página</title>
<STYLE type="text/css">
<!--
H1 {text-decoration: underline; text-align:center}
P {font-family:arial,verdana; color: white; background-color: black}
BODY {color:black;background-color: #cccccc; text-indent:1cm}
// -->
</STYLE>
</head>

<body>
<h1>Página con estilos</h1>
Bienvenidos...
<p>Siento ser tan hortera, pero esto es un ejemplo sin importancia</p>
</body>
</html>

```

Como se puede apreciar en el código, hemos definido que la etiqueta <H1> se presentará

- Subrayado
- Centrada

También, por ejemplo, hemos definido que el cuerpo entero de la página (etiqueta <BODY>) se le apliquen los estilos siguientes:

- Color del texto negro
- Color del fondo grisáceo
- Margen lateral de 1 centímetro

Caber destacar que si aplicamos estilos a la etiqueta <BODY>, estos serán heredados por el resto de las etiquetas del documento. Esto es así siempre y cuando no se vuelvan a definir esos estilos en las siguientes etiquetas, en cuyo caso el estilo de la etiqueta más concreta será el que mande. Puede verse este detalle en la etiqueta <P>, que tiene definidos estilos que ya fueron definidos para <BODY>. Los estilos que se tienen en cuenta son los de la etiqueta <P>, que es más concreta.

Por último, ha de apreciarse los comentarios HTML que engloban toda la declaración de estilos: <!--Declaración de estilos-->. Estos comentarios se utilizan para que los navegadores antiguos, que no comprenden la sintaxis CSS, no incluyan ese texto en el cuerpo de la página. Si no se pusiera, los navegadores antiguos (por ejemplo Netscape 3) escribirían ese "feo código" en la página.

[Pulsa para ver el ejemplo anterior.](#)

Hemos preparado la misma página, pero con declaraciones de estilos distintas, para que comprobéis las diferencias en la forma del documento con sólo unos cambios en sus estilos. [Puedes verla pinchando aquí.](#)

Estilo definido para todo un sitio web

Una de las características más potentes de la programación con hojas de estilos consiste en que, de una vez, podemos definir los estilos de todo un sitio web. Esto se consigue creando un archivo donde tan sólo colocamos las declaraciones de estilos de la página y enlazando todas las páginas del sitio con ese archivo. De este modo, todas las páginas comparten una misma declaración de estilos y, por tanto, si la cambiamos, cambiarán todas las páginas. Con las ventajas añadidas de que se ahorra en líneas de código HTML (lo que reduce el peso del documento) y se evita la molestia de definir una y otra vez los estilos con el HTML, tal como se comentó anteriormente.

Veamos ahora cómo el proceso para incluir estilos con un fichero externo.

1- Creamos el fichero con la declaración de estilos

Es un fichero de texto normal, que puede tener cualquier extensión, aunque le podemos asignar la extensión .css para aclararnos qué tipo de archivo es. El texto que debemos incluir debe ser escrito exclusivamente en sintaxis CSS, es decir, sería erróneo incluir código HTML en él: etiquetas y demás. Podemos ver un ejemplo a continuación.

```
P {
  font-size : 12pt;
  font-family : arial,Helvetica;
  font-weight : normal;
}

H1 {
  font-size : 36pt;
  font-family : verdana,arial;
  text-decoration : underline;
  text-align : center;
  background-color : Teal;
}

TD {
  font-size : 10pt;
  font-family : verdana,arial;
  text-align : center;
  background-color : 666666;
}

BODY {
  background-color : #006600;
  font-family : arial;
  color : White;
}
```

2- Enlazamos la página web con la hoja de estilos

Para ello, vamos a colocar la etiqueta <LINK> con los atributos

- **rel="STYLE SHEET"** indicando que el enlace es con una hoja de estilos
- **type="text/css"** porque el archivo es de texto, en sintaxis CSS
- **href="estilos.css"** indica el nombre del fichero fuente de los estilos

Veamos una página web entera que enlaza con la declaración de estilos anterior:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
```

```
<html>
<head>
  <link rel="stylesheet" type="text/css" href="estilos.css">
  <title>P&acute;gina que lee estilos</title>
</head>

<body>
<h1>P&acute;gina que lee estilos</h1>
Esta p&acute;gina tiene en la cabecera la etiqueta necesaria para enlazar con la hoja de
estilos. Es muy f&acute;cil.
<br>
<br>
<table width="300" cellspacing="2" cellpadding="2" border="0">
<tr>
  <td>Esto est&acute; dentro de un TD, luego tiene estilo propio, declarado en el fichero
externo</td>
</tr>
<tr>
  <td>La segunda fila del TD</td>
</tr>
</table>

</body>
</html>
```

[El resultado conseguido se puede ver aquí](#)

Reglas de importancia en los estilos

Los estilos se heredan de una etiqueta a otra, como se indicó anteriormente. Por ejemplo, si tenemos declarado en el <BODY> unos estilos, por lo general, estas declaraciones también afectarán a etiquetas que estén dentro de esta etiqueta, o lo que es lo mismo, dentro de todo el cuerpo.

En muchas ocasiones más de una declaración de estilos afecta a la misma porción de la página. Siempre se tiene en cuenta la declaración más particular. Pero las declaraciones de estilos se pueden realizar de múltiples modos y con varias etiquetas, también entre estos modos hay una jerarquía de importancia para resolver conflictos entre varias declaraciones de estilos distintas para una misma porción de página. Se puede ver a continuación esta jerarquía, primero ponemos las formas de declaración más generales, y por tanto menos respetadas en caso de conflicto:

- Declaración de estilos con fichero externo. (Para todo un sitio web)
- Declaración de estilos para toda la página. (Con la etiqueta <STYLE> en la cabecera de la página)
- Estilos definidos en una parte de la página. (Con la etiqueta <DIV>)
- Definidos en una etiqueta en concreto. (Utilizando el atributo style en la etiqueta en cuestión)
- Declaración de estilo para una porción pequeña del documento. (Con la etiqueta)

Ya vimos cómo incluir estilos en la página, de todas las maneras posibles e hicimos un repaso con la lista anterior. Ahora estás en condiciones de empezar a usar las hojas de estilo en cascada para mejorar tus páginas y aumentar la productividad de tu trabajo. Pero estate atento a los siguientes capítulos donde aprenderás las lecciones que te faltan para dominar bien la materia: conocer la sintaxis, los distintos atributos de estilos y otras cosas que mejorarán tus páginas.

Otra manera de definir estilos en un archivo externo

Veamos ahora otra manera de importar una declaración externa de estilos CSS: `@import url("archivo_a_importar.css")`, que se utiliza para importar unas declaraciones de estilos guardadas en la ruta que se indica entre paréntesis. (las comillas son opcionales, pero los paréntesis son obligatorios, por lo menos, en Explorer).

Se debe incluir en la declaración de estilos global a una página, es decir entre las etiquetas `<style type="text/css">` y `</style>`, que se colocan en la cabecera del documento.

Es importante señalar que la sentencia de importación del archivo CSS se debe escribir en la primera línea de la declaración de estilos, algo parecido al código siguiente.

```
<style type="text/css">
@import url ("estilo.css");
body{
  background-color: #ffffcc;
}
</style>
```

El funcionamiento es el mismo que si escribiésemos todo el fichero a importar dentro de las etiquetas de los estilos, con la salvedad de que, si redefinimos dentro del código HTML (entre las etiquetas `</style>`) estilos que habían quedado definidos en el archivo externo, los que se aplicarán serán los que hayamos redefinido.

Así, en el ejemplo anterior, aunque hubiésemos definido en `estilo.css` un color de fondo para la página, el color que prevalecería sería el definido a continuación de la importación: `#ffffcc`

La diferencia entre este tipo de importación del tipo y la que hemos visto anteriormente:

```
<link rel="stylesheet" type="text/css" href="hoja.css">
```

Es que `@import url ("estilo.css")` se suele utilizar cuando hay unas pautas básicas en el trabajo con los estilos (que se definen en un archivo a importar) y unos estilos específicos para cada página, que se definen a continuación, dentro del código HTML entre las

etiquetas `</style>`, como es el caso del ejemplo visto anteriormente.

Sintaxis CSS

Tal como se vió en los ejemplos la sintaxis es bastante sencilla y repetitiva. Vamos a verla:

- Para definir un estilo se utilizan atributos como `font-size`, `text-decoration`... seguidos de dos puntos y el valor que le deseemos asignar. Podemos definir un estilo a base de definir muchos atributos separados por punto y coma.

Ejemplo:

font-size: 10pt; text-decoration: underline; color: black; (el último punto y coma de la lista de atributos es opcional)

- Para definir el estilo de una etiqueta se escribe la etiqueta seguida de la lista de atributos encerrados entre llaves.

Ejemplo:

H1{ text-align: center; color:black }

- Los valores que se pueden asignar a los atributos de estilo se pueden ver en una tabla en el siguiente capítulo. Muchos estos valores son unidades de medida, por ejemplo, el valor del tamaño de un margen o el tamaño de la fuente. Las unidades de medida son las siguientes:

Puntos	pt
Pulgadas	in
Centímetros	cm
pixels	px

Hasta aquí, queda dicho todo lo relativo a la sintaxis. En el siguiente capítulo podrás encontrar una lista de los atributos de las hojas de estilo en cascada.

Atributos de las hojas de estilo

Tanto para practicar en tu aprendizaje como para trabajar con las CSS lo mejor es disponer de una tabla donde se vean los distintos atributos y valores de estilos que podemos aplicarle a las páginas web.

Aquí puedes ver la tabla de los atributos CSS, tenla a mano cuando utilices las CSS.

Nombre del atributo	Posibles valores	Ejemplos
---------------------	------------------	----------

FUENTES - FONT

color	valor RGB o nombre de color	color: #009900; color: red;
--------------	-----------------------------	--------------------------------

Sirve para indicar el color del texto. Lo admiten casi todas las etiquetas de HTML. No todos los nombres de colores son admitidos en el estandar, es aconsejable entonces utilizar el valor RGB.

font-size	xx-small x-small small medium large x-large xx-large Unidades de CSS	font-size: 12pt; font-size: x-large;
------------------	---	---

Sirve para indicar el tamaño de las fuentes de manera más rígida y con mayor exactitud.

font-family	serif sans-serif cursive fantasy monospace Todas las fuentes habituales	font-family: arial, helvetica; font-size: fantasy;
--------------------	--	---

Con este atributo indicamos la familia de tipografía del texto. Los primeros valores son genéricos, es decir, los exploradores las comprenden y utilizan las fuentes que el usuario tenga en su sistema.

También se pueden definir con tipografías normales, como ocurría en html. Si el nombre de una fuente tiene espacios se utilizan comillas para que se entienda bien.

font-weight	normal bold bolder lighter 100 200 300 400 500 600 700 800 900	font-weight: bold; font-weight: 200;
--------------------	--	---

Sirve para definir la anchura de los caracteres, o dicho de otra manera, para poner negrillas con total libertad.

Normal y 400 son el mismo valor, así como bold y 700.

font-style	normal italic oblique	font-style: normal; font-style: italic;
-------------------	---------------------------	--

Es el estilo de la fuente, que puede ser normal, itálica u oblicua. El estilo oblique es similar al italic.

PÁRRAFOS - TEXT

line-height	normal y unidades CSS	line-height: 12px; line-height: normal;
--------------------	-----------------------	--

El alto de una línea, y por tanto, el espaciado entre líneas. Es una de esas características que no se podían modificar utilizando HTML.

text-decoration	none [underline overline line-through]	text-decoration: none; text-decoration: underline;
------------------------	--	---

Para establecer la decoración de un texto, es decir, si está subrayado, sobrerayado o tachado.

text-align	left right center justify	text-align: right; text-align: center;
-------------------	---------------------------------	---

Sirve para indicar la alineación del texto. Es interesante destacar que las hojas de estilo permiten el justificado de texto, aunque recuerda que no tiene por que funcionar en todos los sistemas.

text-indent	Unidades CSS	text-indent: 10px; text-indent: 2in;
--------------------	--------------	---

Un atributo que sirve para hacer sangrado o márgenes en las páginas. Muy útil y novedosa.

text-transform	capitalize uppercase lowercase none	text-transform: none; text-transform: capitalize;
-----------------------	---	--

Nos permite transformar el texto, haciendo que tenga la primera letra en mayúsculas de todas las palabras, todo en mayúsculas o minúsculas.

FONDO - BACKGROUND

Background-color	Un color, con su nombre o su valor RGB	background-color: green; background-color: #000055;
-------------------------	--	--

Sirve para indicar el color de fondo de un elemento de la página.

Background-image	El nombre de la imagen con su camino relativo o absoluto	background-image: url (mármol.gif) ; background-image: url(http://www.x.com/fondo.gif)
-------------------------	--	---

Colocamos con este atributo una imagen de fondo en cualquier elemento de la página, se puede ver una [página de ejemplo](#)

BOX - CAJA

Margin-left	Unidades CSS	margin-left: 1cm; margin-left: 0,5in;
--------------------	--------------	--

Indicamos con este atributo el tamaño del margen a la izquierda

Margin-right	Unidades CSS	margin-right: 5%; margin-right: 1in;
---------------------	--------------	---

Se utiliza para definir el tamaño del margen a la derecha

Margin-top	Unidades CSS	margin-top: 0px; margin-top: 10px;
-------------------	--------------	---------------------------------------

Indicamos con este atributo el tamaño del margen arriba de la página

Margin-bottom	Unidades CSS	margin-bottom: 0pt; margin-top: 1px;
----------------------	--------------	---

Con el se indica el tamaño del margen en la parte de abajo de la página

Padding-left	Unidades CSS	padding-left: 0.5in; padding-left: 1px;
---------------------	--------------	--

Indica el espacio insertado, por la izquierda, entre el borde del elemento-continente y el contenido de este. Es parecido a el atributo cellpadding de las tablas.

El espacio insertado tiene el mismo fondo que el fondo del elemento-continente.

Padding-right	Unidades CSS	padding-right: 0.5cm; padding-right: 1pt;
----------------------	--------------	--

Indica el espacio insertado, en este caso por la derecha, entre el borde del elemento-continente y el contenido de este. Es parecido a el atributo cellpadding de las tablas.

El espacio insertado tiene el mismo fondo que el fondo del elemento-continente.

Padding-top	Unidades CSS	padding-top: 10pt; padding-top: 5px;
--------------------	--------------	---

Indica el espacio insertado, por arriba, entre el borde del elemento-continente y el contenido de este.

Padding-bottom	Unidades CSS	padding-right: 0.5cm; padding-right: 1pt;
-----------------------	--------------	--

Indica el espacio insertado, en este caso por abajo, entre el borde del elemento-continente y el contenido de este.

Border-color	color RGB y nombre de color	border-color: red; border-color: #ffccff;
---------------------	-----------------------------	--

Para indicar el color del borde del elemento de la página al que se lo aplicamos. Se puede poner colores por separado con los atributos border-top-color, border-right-color, border-bottom-color, border-left-color.

Border-style	none dotted solid double groove ridge inset outset	border-style: solid; border-style: double;
---------------------	--	---

El estilo del borde, los valores significan: none=ningun borde, dotted=punteado (no parece funcionar), solid=solido, double=doble borde, y desde groove hasta outset son bordes con varios efectos 3D.

border-width	Unidades CSS	border-width: 10px; border-width: 0.5in;
---------------------	--------------	---

El tamaño del borde del elemento al que lo aplicamos. Para ver otros ejemplos de Box [pulsar aquí](#)

float	none left right	float: right;
--------------	---------------------	---------------

Sirve para alinear un elemento a la izquierda o la derecha haciendo que el texto se agrupe alrededor de dicho elemento. Igual que el atributo align en imágenes en sus valores right y left.

clear	none right left	clear: right;
--------------	---------------------	---------------

Si este elemento tiene a su altura imágenes u otros elementos alineados a la derecha o la izquierda, con el atributo clear hacemos que se coloque en un lugar donde ya no tenga esos elementos a el lado que indiquemos. Para ver una página que utiliza float y clear [pulsar aquí](#)

La especificación de estilos CSS es muy amplia, seguro que se queda en el tintero algún atributo de estilo, pero creo que la inmensa mayoría están, y por supuesto la selección de los más importantes.

Trucos avanzados con CSS

Las hojas de estilos son un tema largo del que se han escrito libros enteros. Nosotros nos centramos en los temas prácticos y por ello vamos a acabar ya con este capítulo, en unos cuantos puntos

Definir estilos utilizando clases

Las clases nos sirven para crear definiciones de estilos que se pueden utilizar repetidas veces.

Una clase se puede definir entre las etiquetas <STYLE> (en la cabecera del documento), o en un archivo externo a la página. Para definir las utilizamos la siguiente sintaxis, un punto seguido del nombre de la clase y entre llaves los atributos de estilos deseados. De esta manera:

```
.nombredelaclase { atributo: valor; atributo2: valor2; ... }
```

Una vez tenemos una clase, podemos utilizarla en cualquier etiqueta HTML. Para ello utilizaremos el atributo class, poniéndole como valor el nombre de la clase, de esta forma:

```
<ETIQUETA class="nombredelaclase">
```

Ejemplo de la utilización de clases

```
<html>
<head>
  <title>Ejemplo de la utilización de clases</title>
  <STYLE type="text/css">
    .fondonegroletrasblancas { background-color: black; color: white; font-size: 12; font-family: arial }
    .letrasverdes { color: #009900 }
  </STYLE>
</head>

<body>
<h1 class=letrasverdes>Titulo 1</h1>
<h1 class=fondonegroletrasblancas>Titulo 2</h1>

<p class=letrasverdes>
Esto es un párrafo con estilo de letras verdes</p>
<p class=fondonegroletrasblancas>
Esto es un párrafo con estilo de fondo negro y las letras blancas. Es todo!</p>
</body>
</html>
```

[Ver el ejemplo anterior](#)

Estilo en los enlaces

Una técnica muy habitual, que se puede realizar utilizando las hojas de estilo en cascada y no se podía en HTML, es la definición de estilos en los enlaces, quitándoles el subrayado o hacer enlaces en la misma página con distintos colores.

Para aplicar estilo a los enlaces debemos definirlos para los distintos tipos de enlaces que existen (visitados, activos...). Utilizaremos la siguiente sintaxis, en la declaración de estilos global de la página (<STYLE>) o del sitio (Definición en un archivo externo):

Enlaces normales

```
A:link { atributos }
```

Enlaces visitados

```
A:visited { atributos }
```

Enlaces activos (Los enlaces están activos en el preciso momento en que se pincha sobre ellos)

A:active {atributos}

Enlaces hover (Cuando el ratón está encima de ellos, solo funciona en ieplorer)

A:hover {atributos}

El atributo para definir enlaces sin subrayado es **text-decoration:none**, y para darles color es color: tu_color.

También podemos definir el estilo de cada enlace en la propia etiqueta <A>, con el atributo style. De esta manera podemos hacer que determinados enlaces de la página se vean de manera distinta

Ejemplo de estilos en enlaces

```
<html>
<head>
<title>Ejemplos de estilo en enlaces</title>
<STYLE type="text/css">
A:link {text-decoration:none;color:#0000cc;}
A:visited {text-decoration:none;color:#ffcc33;}
A:active {text-decoration:none;color:#ff0000;}
A:hover {text-decoration:underline;color:#999999;font-weight:bold}
</STYLE>
</head>

<body>

<a href="http://dominioinexistente.nofunciona.com">Enlace normal</a>
<br>
<br>
<a href="enlaces.html">Enlace visitado</a>
Pulsar este enlace para verlo activo,
poner el ratón por encima para que cambie.

</body>
</html>
```

[Ver el ejemplo anterior](#)

URL como valor de un atributo:

Determinados atributos de estilos, como **background-image** necesitan una URL como valor, para indicarlas podemos definir tanto caminos relativos como absolutos. Así pues, podemos indicar la URL de la imagen de fondo de estas dos maneras:

background-image: url(fondo.gif) En caso de que la imagen esté en el mismo directorio que la página. **background-image: url(http://www.desarrolloweb.com/images/fondo.gif)** ç

Ocultar estilos en navegadores antiguos

En caso de definir dentro de la etiqueta <STYLE> unas declaraciones de estilos debemos asegurarnos que estas no se imprimirán en la página web con navegadores antiguos. Pensar en un navegador que no reconozca la etiqueta <STYLE>, pensará que

corresponde con algo que no entiende y se olvidará de la etiqueta. Lo siguiente que encuentra es texto normal y hará que este se vea en la página, como con cualquier otro texto.

Para evitarlo debemos ocultar con comentarios HTML (`<!-- esto es un comentario -->`) todo lo que hay dentro de la etiqueta `<STYLE>`. Se puede ver un ejemplo de esto a continuación:

De este modo hemos terminado nuestro manual de CSS, que espero pueda ayudar a hacer páginas mejores y más rápidamente.

Quiero recordaros que siempre es útil ver como han hecho sus páginas otros programadores de Internet. Para ver una página definida enteramente con hojas de estilos visitar la dirección www.guiarte.com

Para ver otras páginas que enseñan a utilizar las hojas de estilos visitar la [sección correspondiente a CSS de nuestro buscador](#):

Aplicación de estilo avanzada a los enlaces

En este artículo vamos a ver cómo podríamos crear una barra de navegación bastante dinámica utilizando únicamente las Hojas de Estilo en Cascada. En el ejemplo vamos a construir una barra de navegación que contiene enlaces de varios colores que cambian de tonalidad al pasar el ratón por encima.

Podemos [ver el resultado de nuestra barra pulsando este enlace](#), para tener una idea más exacta de lo que queremos conseguir.

Cómo poner estilo a los enlaces

Ya lo vimos en capítulos anteriores de nuestro [manual de CSS](#), pero lo repetimos aquí. Se define el estilo de los enlaces asignando su apariencia en sus distintos estados:

- Enlace no visitado. Se define con el atributo `link`.
- Enlace visitado. Se define con el atributo `visited`.
- Enlace activo (cuando se está pulsando). Se define con `active`.
- Enlace con el ratón encima. Se define con `hover`.

Esta definición se realiza en la cabecera de la página, entre las etiquetas `<STYLE>` Y `</STYLE>`, y es global a toda la página.

Un ejemplo de esto se puede ver en esta declaración de estilos:

```
<STYLE type="text/css">
A:link { text-decoration: none; color: #0000cc; }
```

```
A:visited {text-decoration:none;color:#ffcc33;}
A:active {text-decoration:none;color:#ff0000;}
A:hover {text-decoration:underline;color:#999999;}
```

</STYLE>

Cómo dar estilo a un enlace en concreto

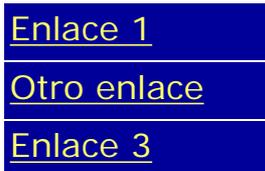
Si queremos resaltar nuestra barra de navegación probablemente querramos colocarla en una tabla de nuestra página web, con un color que contraste un poco con el fondo. En un caso como este, será necesario que los enlaces de la barra de navegación y los enlaces normales de la página tengan colores distintos, por estar situados sobre dos tipos de fondos distintos.

Es por esto que los enlaces de la barra van a tener un color distinto de los definidos en la cabecera de la página, con los estilos. Esto lo podemos conseguir de esta manera.

```
<a href="#" style="color:#ff0000">Mi enlace</a>
```

Hemos definido el color de un enlace de una manera específica, utilizando el atributo `style`, de modo que este enlace siempre tendrá el color indicado, independientemente de su estado.

Es un enlace amarillo, que quedaría muy bien resaltado sobre un fondo oscuro, como se puede ver en el ejemplo de barra de navegación siguiente.



En la tabla anterior tenemos enlaces amarillos en una web donde los enlaces son azules por defecto.

Cómo utilizar las clases al aplicar estilo a los enlaces

También vimos en anteriores capítulos que el uso de clases puede ser muy útil a la hora de definir estilos especiales que podemos aplicar a las etiquetas que queramos. A la hora de trabajar con los enlaces también podemos aplicar las clases para definir distintos tipos de enlaces, que tienen distintos tipos de estilos.

```
A.clase1:visited {color:#ff0000;}
A.clase1:active {color:#ff0000;}
A.clase1:link {color:#ff0000}
A.clase1:hover {color:#00ff00}
```

La ventaja al utilizar las clases con los estilos de los enlaces es que podemos especificar un formato distinto al enlace dependiendo de su estado: visitado o no, activo o con el ratón sobre él.

Por si no quedó claro, al especificar el estilo con el atributo `style` del enlace sólo podíamos decir que el enlace lo queremos en amarillo, y siempre lo tendremos en amarillo (sea visitado o no, activo, o estemos o no con el ratón encima). Con las clases definimos un nuevo tipo de enlace al que podemos dar distintos formatos dependiendo su estado.

Otras ventajas de utilizar las clases consisten en que escribimos una única vez los estilos y que podemos cambiar el color de todos los enlaces de la clase con cambiar la

delaración.

A partir de lo que acabamos de aprender podemos crear el ejemplo de barra de navegación dinámica utilizando CSS que habíamos visto al principio del capítulo. El código sería el siguiente.

```
<html>
<head>
  <title>Ejemplo CSS para enlaces</title>
<style type="text/css">
  A:link {color:#0000cc;}
  A:visited {color:#0000cc;}
  A:active {color:#0000cc;}
  A:hover {color:#0000ff;}

  A.clase1:visited {color:#ffff00;}
  A.clase1:active {color:#ffff00;}
  A.clase1:link {color:#ffff00}
  A.clase1:hover {color:#00ff00}

  A.clase2:visited {font-size:12;color:#ffffff;}
  A.clase2:active {font-size:12;color:#ffffff;}
  A.clase2:link {font-size:12;color:#ffffff;}
  A.clase2:hover {font-size:12;color:#ffff33;}

  body {font-family:arial;font-size:11;font-weight:bold}
  td {font-family:arial;font-size:11;font-weight:bold}
</style>
</head>

<body>
<a href="#">Este enlace es normal</a>
<br>
<br>
<br>
<span style="font-weight:normal;font-size:10">
Los enlaces de esta barra son especiales,
<br>
están definidos por clases
</span>
<br>
<table width="110" cellspacing="1" cellpadding="2" border="0">
<tr>
  <td bgcolor="#aa0000"><a href="#" class="clase2">Opciones 1</a></td>
</tr>
<tr>
  <td bgcolor="red"><a href="#" class="clase1">Enlace clase1</a></td>
</tr>
<tr>
  <td bgcolor="red"><a href="#" class="clase1">Otro de clase1</a></td>
</tr>
<tr>
  <td bgcolor="red"><a href="#" class="clase1">Más enlaces</a></td>
</tr>
<tr>
  <td bgcolor="red"><a href="#" class="clase1">Todavía más</a></td>
</tr>
</table>

</body>
</html>
```

Se puede [ver la página en funcionamiento en este enlace](#). Esperamos que sirva de provecho este pequeño taller de CSS para vuestras creaciones.

Quiero agradecer la ayuda prestada para la elaboración de este artículo a Rafael Chacón, que nos mandó un generoso mail con el truquillo que hemos expuesto aquí.

Qué son las capas

Veamos una pequeña introducción a lo que son las capas, la etiqueta HTML <DIV> utilizada para construirla y los atributos CSS con los que podemos aplicar estilos.

Como ya hemos visto en nuestro [manual de CSS](#), sirve para aplicarle estilo a una pequeña parte de una página HTML. Por ejemplo, con ella podríamos hacer que una parte de un párrafo se coloree en rojo. Con no es habitual englobar un trozo muy grande de texto, por ejemplo el que comprenda a varios párrafos.

Con <DIV> también podemos aplicar estilo a partes de la página HTML.

La diferencia entre y <DIV> es que con esta última si que podemos aplicar estilo a una parte más amplia de la página, por ejemplo a tres párrafos. **Además que la etiqueta <DIV> tiene un uso adicional que es el de crear divisiones en la página a las que podremos aplicar una cantidad adicional de atributos para modificar sus comportamientos.** Por ejemplo, con el atributo align de HTML podemos alinear la división al centro, izquierda, derecha o justificado. Pero **su uso más destacado es el de convertir esa división en una capa.**

Una capa es una división, una parte de la página, **que tiene un comportamiento muy independiente** dentro de la ventana del navegador, ya que la podemos colocar en cualquier parte de la misma y la podremos mover por ella independientemente, por poner dos ejemplos. En el uso de capas se basan muchos de los efectos más comunes del DHTML.

Las etiquetas <LAYER> e <ILAYER> tienen como objetivo construir capas, pero estas no son compatibles más que con Netscape, por lo que es recomendable utilizar la etiqueta <DIV> para hacer capas preferentemente a las otras dos.

Los atributos que podemos aplicar a estas etiquetas, pero en concreto a las dos recomendadas y <DIV>, son principalmente de estilos CSS. Estos atributos nos permiten, como hemos podido ver en el manual de hojas de estilo en cascada de desarrolloweb, modificar de una manera muy exhaustiva la presentación de los contenidos en la página. Para aplicar estilos a estas etiquetas se utiliza el atributo de HTML style, de esta manera:

```
<SPAN style="text-decoration: underline; font-weight: bold">...</SPAN>
```

```
<DIV style="color: red; font-size: 10px">...</DIV>
```

Como ya pudimos ver muchos ejemplos en el [manual de CSS](#), nos referimos a él para ampliar esta información. Pero no habíamos visto todavía una serie de atributos que nos sirven para posicionar la división en la página como una capa. Estos atributos se pueden aplicar a la etiqueta <DIV> que es la que servía para crear capas compatibles con todos los navegadores.

Los atributos para que la división sea una capa son varios y se pueden ver a continuación.

```
<div id="c1" style="position:absolute; left: 200px; top: 100px;">
Hola!
</div>
```

El primero, **position**, indica que se posicione de manera absoluta en la página y los segundos, **left** y **top**, son la distancia desde el borde izquierdo de la página y el borde superior.

Hay otros atributos especiales para capas como **width** y **height** para indicar la anchura y altura de la capa, **Z-index** que sirve para indicar qué capas se ven encima de qué otras, **clip** que sirve para recortar una capa y hacer que partes de ella no sean visibles, o **visibility** para definir si la capa es visible o no. Estos y otros atributos los veremos en el [siguiente capítulo, donde hablaremos del posicionamiento de capas](#).

Atributos para capas

Hemos visto en el capítulo anterior [qué son las capas y algunas pequeñas muestras sobre cómo crearlas y darle algún estilo](#). Ahora vamos a ver en detenimiento los atributos específicos para aplicar posicionamiento a una capa y otros estilos.

Antes que nada cabe decir que una capa puede tener cualquier atributo de estilos de los que hemos visto en el [manual de CSS](#). Así, el atributo color indica el color del texto de la capa, el atributo font-size indica el tamaño del texto y así con todos los atributos ya vistos.

Ahora bien, existen una serie de atributos que sirven para indicar la forma, el tamaño de las capas, la visibilidad, etc, que no hemos visto en capítulos anteriores y que veremos a continuación.

Atributo position

Indica el tipo de posicionamiento de la capa. Puede tener dos valores, relative o absolute.

- relative indica que la posición de la capa es relativa a el lugar donde se estaba escribiendo en la página en el momento de escribir la capa con su etiqueta

- absolute indica que la posición de la capa se calcula con respecto al punto superior izquierdo de la página.

Atributo top

Indica la distancia en vertical donde se colocará la capa. Si el atributo position es absolute, top indica la distancia del borde superior de la capa con respecto al borde superior de la página. Si el atributo position era relative, top indica la distancia desde donde se estaba escribiendo en ese momento en la página hasta el borde superior de la capa.

Atributo left

Básicamente funciona igual que el atributo top, con la diferencia que el atributo left indica la distancia en horizontal a la que estará situada la capa.

Atributo height

Sirve para indicar el tamaño de la capa en vertical, es decir, su altura.

Atributo width

Indica la anchura de la capa

Atributo visibility

Sirve para indicar si la capa se puede ver en la página o permanece oculta al usuario. Este atributo puede tener tres valores.

- visible sirve para indicar que la capa se podrá ver.
- hidden indicará que la capa está oculta.
- inherit es el valor por defecto, que quiere decir que hereda la visibilidad de la capa donde está metida la capa en cuestión. Si la capa no está metida dentro de ninguna otra se supone que está metida en la capa documento, que es toda la página y que siempre está visible.

Atributo z-index

Sirve para indicar la posición sobre el eje z que tendrán las distintas capas de la página. Dicho de otra forma, con z-index podemos decir qué capas se verán encima o debajo de otras, en caso de que estén superpuestas. El atributo z-index toma valores numéricos y a mayor z-index, más arriba se verá la página.

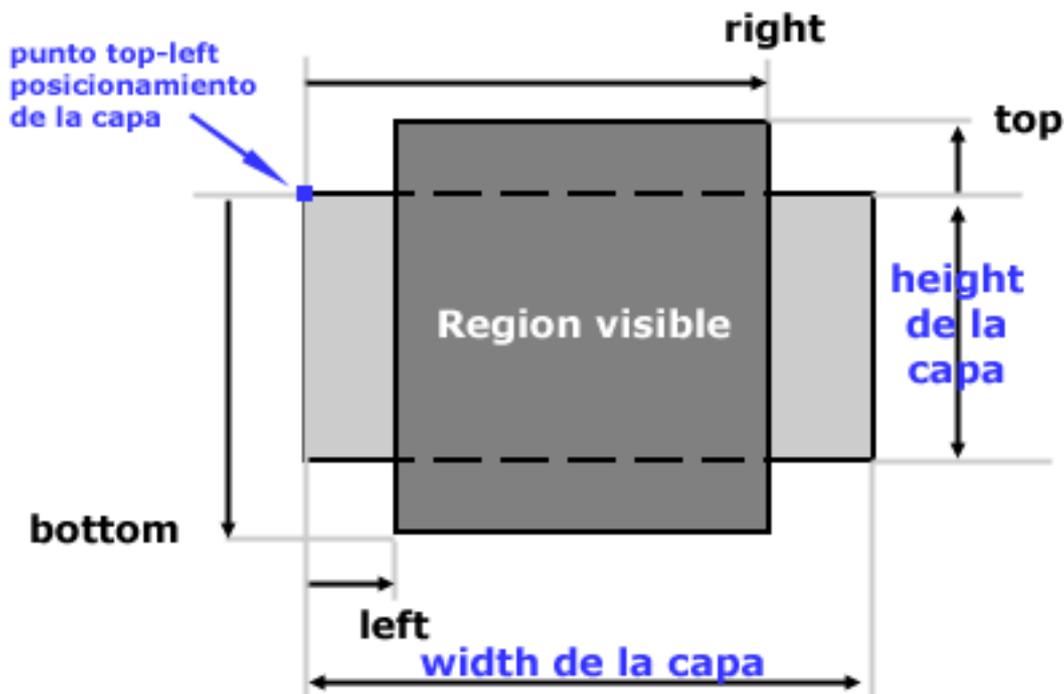
Atributo clip

Es un atributo un poco difícil de explicar. En concreto sirve para recortar determinadas áreas de la capa y que no se puedan ver. Una capa que está visible se puede recortar para que se vea, pero que no se vea algún trozo de esta. El clipping se indica por medio de 4 valores, con esta sintaxis.

rect (<top>, <right>, <bottom>, <left>)

Los valores <top>, <right>, <bottom> y <left> indican distancias que se pueden

apreciar en este esquema.



Este es un ejemplo de capa que utiliza todos los atributos que hemos visto en este artículo y alguno más para aplicar estilo a la capa.

```
<div style="clip: rect(0,158,148,15); height: 250px; width: 170px; left: 10px; top: 220px; position: absolute; visibility: visible; z-index: 10; font-size: 14pt; font-family: verdana; text-align: center; background-color: #bbbbbb">
```

Esta capa tiene un clipping, por eso se ve entrecortada.

```
<br>
```

```
<br>
```

Esto es una capa de prueba

```
</div>
```

Puede [verse el ejemplo en una página web](#), donde también podrá apreciarse el efecto conseguido al realizar el clipping.

Problema con el posicionamiento absoluto de capas

He recibido una consulta en mi correo sobre colocación de capas de manera absoluta, pero en la que no nos importe la definición de la pantalla del usuario y otros ir y venir de los elementos HTML. Nuestro compañero expresó su duda de la siguiente manera:

Si trabajamos con position: absolute dando un left y un top funciona si tienes tu página alineada a la izquierda. Mi página está alineada en el centro, entonces lo que sucede es que dependiendo de la resolución de pantalla que tengas (ancho de 800px, 1024px, etc) me baila toda la página y no cuadra nada.

Primero que todo, debemos saber que si trabajamos con el position relative las capas se

colocan en el lugar donde aparecen dentro del código HTML. De este modo, si colocamos una capa con position relative dentro de una celda de una tabla, dicha capa aparecería dentro de la celda donde la estamos colocando, independientemente del lugar donde se sitúe la celda al cambiar la definición de la pantalla.

El problema de esta solución es que la capa haría crecer la celda de la tabla donde queremos colocarla (al igual que cualquier otro elemento HTML que colocásemos dentro de la tabla) y es muy probable que nuestro diseño no nos permita este hecho. Seguramente ya habrías notado este problema y si no es así te invito a que crees la capa que intentas colocar con el atributo position a relative para ver si con eso tu problema ya está resuelto.

En casi todos los casos, la capa que intentamos colocar va a tener que tener el position absolute, porque con relative no arreglamos totalmente el problema. Entonces volvemos a el problema inicial, que era situar la capa con position absolute en el lugar exacto, independientemente de la definición de pantalla.

La solución final que propongo pasa por aplicar algún truquillo. De hecho, estuve hace unos días preguntándome sobre esa cuestión y al final encontré la solución, aunque no se me ocurrió a mi, sino que la extraje de www.cross-browser.com.

La idea es un poco compleja y para su puesta en marcha debemos realizar una serie de acciones que, sinceramente, considero excesivas para un problema inicialmente sencillo. Así pues, que no asuste lo que voy a soltar a continuación, que luego trataré de explicarlo un poco mejor.

Nuestro esquema de trabajo consistirá en una capa con posición relativa, que nos servirá de "ancla" y otra con la posición absoluta, donde colocaremos el contenido final a mostrar en la capa.

La capa relativa la colocaremos en el lugar aproximado donde queramos que aparezca la capa absoluta. La capa absoluta la posicionaremos, una vez cargada la página, en un lugar próximo a la capa relativa. Por supuesto, estas acciones las vamos a tener que realizar con Javascript, que es el lenguaje que nos permite actualizar las posiciones de las capas dinámicamente.

Detenidamente

Decíamos que habría que colocar una capa relativa cercana al lugar donde tiene que aparecer la capa con position absolute. Insisto en que las capas relativas se colocan en el lugar donde las metemos dentro del código HTML, por lo que será fácil colocar la capa relativa en el lugar exacto y que este lugar sea válido para cualquier definición.

La segunda capa, la que tiene el contenido final, la pondremos inicialmente en una posición cualquiera y escondida, de manera que no se vea que está mal colocada. Una vez terminada de cargar la página, podremos acceder a la posición de la capa relativa, extrayendo sus valores top y left y colocándolos en los correspondientes top y left de la capa con posición absoluta. Una vez marcada la posición de la capa absoluta podemos volverla visible.

A la vista de la imagen siguiente, la capa con posición relativa la hemos colocado en el enlace. En realidad habría tres capas con posición relativa para poder posicionar otras tantas capas con posición absoluta. La parte que vemos sombreada de verde corresponde al espacio que abarcaría la capa relativa.

Su posición sería la que está marcada por el aspa roja que aparece en su esquina superior izquierda. Dicha posición depende del lugar donde aparezcan los enlaces en la página.

Luego, con Javascript deberíamos asignar la posición de la capa absoluta de una manera parecida a esta.

left de la capa absoluta = left de la capa relativa
top de la capa absoluta = top de la capa relativa + altura de la capa relativa

Podemos sumarle algún píxel más a la posición de la capa, si es que queremos moverla un poco abajo y a la derecha, tal como hemos visto en la imagen.

No pretendo en este artículo, muy a mi pesar y por falta de espacio y tiempo, explicar cómo se hacen esas operaciones de Javascript. Advierto que si no se conoce nada de Javascript va a ser imposible ponerse con una tarea tan tediosa como el manejo de capas. Si por el contrario, ya hemos tenido contacto con Javascript y DHTML anteriormente, no debería ser un problema realizar esas acciones.

Referencias Javascript

En DesarrolloWeb tenemos un par de manuales de Javascript, que sería necesario estudiar para empezar a introducirse en el lenguaje.

- [Programación en Javascript I](#)
- [Programación en Javascript II](#)

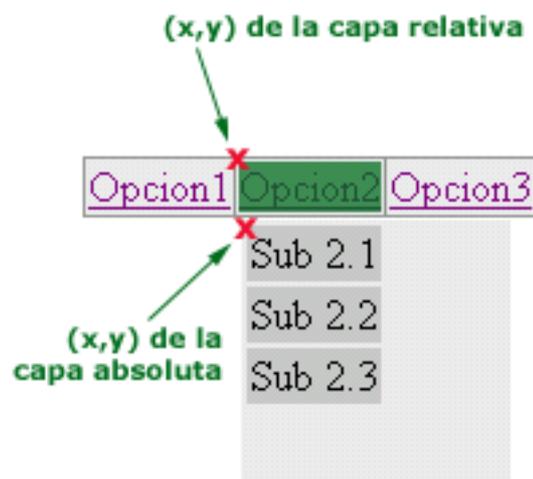
En el [Taller de Javascript](#) tenemos algún artículo sobre tratamiento dinámico de capas.

En el momento en el que escribo estas líneas no tenemos un manual de DHTML y tratamiento de capas completo, por lo que sería recomendable estudiarlo en alguno de los enlaces recomendados de [nuestra sección de DHTML del buscador](#)

Cómo evitar que una página se imprima

Para ello, hay que echar mano de las hojas de estilo. Tanto si el documento tiene una hoja ya asociada como sino, lo que vamos a hacer es asociarle un nueva hoja de estilos. Dicha hoja contendrá un único estilo, con el código necesario para ocultar un elemento:

```
.nover{  
visibility:hidden  
}
```



A la hora de asociar la hoja de estilos, se le añade un modificador a la etiqueta HTML que enlaza con el fichero .css que permite especificar para qué tipo de medio se aplicará esta hoja. En nuestro caso, se aplica en el ámbito de la impresión, por lo que se utiliza el atributo `media="print"`.

```
<link href="nombre_hoja" rel="stylesheet" type="text/css" media="print">
```

Una vez hecho esto, basta que toda nuestra página este dentro de un elemento `div`, que pertenezca a la clase `nover`.

```
<body>  
<div class="nover">
```

```
-- Contenido --
```

```
</div>  
</body>
```

Al hacer esto se provoca que en pantalla se visualice la página, pero que si por el contrario se decide imprimir, se le aplicará la hoja de estilos de impresión, en la que el elemento esta puesto como no visible, por lo que no se imprimirá.

Codigo Completo:

Veamos el código íntegro de la página web y la hoja de estilos asociada.

Página HTML

```
<html>  
<head>  
<link href="estilos.css" rel="stylesheet" type="text/css" media="print">  
</head>  
<body>  
<div class="nover">  
  
... contenido de la pagina  
  
</div>  
</body>  
</html>
```

Hoja estilos: estilos.css

```
.nover{  
visibility:hidden  
}
```

Nota: Esta característica de las hojas de estilos funciona con éxito en navegadores IExplorer 6, Netscape 7 y Opera 7. No la hemos probado en otras versiones.

