

Modulo
HTML

Tema
CSS EN WEB

Adaptado a los cursos de la familia Educativa INFORMATICA Y
COMUNICACIONES

GUIA DEL ALUMNO

INDICE

Que es HOJAS DE ESTILO	3
CARACTERISTICAS Y	
VENTAJAS.....	3
Navegadores que lo soportan	4
USOS.....	5
Reglas de importancia en los estilos	9
SINTAXIS	11
TRUCOS	16
CAPAS	18
MAQUETACION CSS	23
FORMAS DE APLICAR ESTILOS EN MAQUETACION	25
COLORES EN CSS	27

Que es HOJAS DE ESTILO

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 imágenes 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 venían 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.

CARACTERISTICAS Y VENTAJAS

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 identado 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 mostrando 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:

- Píxeles (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

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.

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
muyfácil.
</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).

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

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>Ejemplodeestilosparatodaunaap&aacute;gina</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>P&aacute;gina con estilos</h1>
Bienvenidos...
<p>Siento ser tan hortera, pero esto es un ejemplo sin m&aacute;s 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.

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.

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 el: 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="STYLESHEET" 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>Página que lee estilos</title>
</head>

<body>
<h1>Página que lee estilos</h1>
Esta página tiene en la cabecera la etiqueta necesaria para enlazar con la hoja de estilos. Es
muy fácil.
<br>
<br>
<table width="300" cellspacing="2" cellpadding="2" border="0">
<tr>
<td>Esto está 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>
```

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)
- 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 CSS en 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

Tal como se vió en los ejemplos de los artículos anteriores del [Manual de CSS](#), la sintaxis es bastante sencilla y repetitiva.

Básicamente se trata de colocar selectores de elementos (por ahora sólo hemos visto cómo seleccionar etiquetas, para asignarles estilos, pero más adelante conoceremos otros selectores), seguidos de los valores o atributos de estilo que queramos aplicar a dichos elementos.

A lo largo del manual aprenderemos más sobre la sintaxis de CSS, así como los distintos atributos disponibles para definir el formato o forma con la que se deben mostrar los contenidos. No obstante, quiero dar en este momento unas reglas básicas que debemos saber sobre la sintaxis de CSS, que nos servirán para entender mejor nuestros ejemplos e ir avanzando en el conocimiento de las hojas de estilo en cascada.

Veamos entonces estas reglas básicas sobre la sintaxis CSS:

- 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 (Unidades CSS), por ejemplo, el valor del tamaño de un margen o el tamaño de la fuente. Las unidades de medida CSS se pueden clasificar en dos grupos, las relativas y las absolutas. Más la posibilidad de expresar valores en porcentaje.

Relativas: Se llaman así porque son unidades relativas al medio o soporte sobre el que se está viendo la página web, que dependiendo de cada usuario puede ser distinto, puesto que existen muchos dispositivos que pueden acceder a la web, como ordenadores o teléfonos móviles. En principio las unidades relativas son más aconsejables, porque se ajustarán mejor al medio con el que el usuario está accediendo a nuestra web. Son las siguientes:

Fuente actual:	em	la unidad em es relativa a la fuente actual con la que se está trabajando por defecto en el sistema del usuario. Por ejemplo si un visitante tiene configurada la fuente por defecto en 12 puntos, 1em será igual a 12 puntos y 2em será igual a 24 puntos.
Altura de la letra "x": ex	1ex	será igual a la altura de la letra x, según la fuente actual del usuario. La altura de la letra x generalmente es la mitad de la de la fuente normal.
Píxeles:px		Un pixel es un punto en la pantalla del dispositivo. Dependiendo de la resolución de la pantalla, un píxel puede ser mayor o menor.

Absolutas: Las unidades absolutas son medidas fijas, que deberían verse igual en todos los dispositivos. Como los centímetros, que son una convención de medida internacional. Pese a que en principio pueden parece más útiles, puesto que se verían en todos los sistemas igual, tienen el problema de adaptarse menos a las distintas particularidades de los dispositivos que pueden acceder a una web y restan accesibilidad a nuestro web. Puede que en tu ordenador 1 centímetro sea una medida razonable, pero en un móvil puede ser un espacio exageradamente grande,

puesto que la pantalla es mucho menor. Se aconseja utilizar, por tanto, medidas relativas.

Puntos	pt	Un punto es 1/72 pulgadas
Pulgadas	in	
Centímetros	cm	
Milímetros	mm	
Picas	pc	Una pica son 12 puntos.

Porcentaje: el porcentaje se utiliza para definir una unidad en función de la que esté definida en un momento dado. Imaginemos que estamos trabajando en 12pt y definimos una unidad como 150%. Esto sería igual al 150% de los 12pt actuales, que equivale a 18pt.

Porcentaje	%	Por ejemplo 120% es el 120 por cien de la unidad que estuviera anteriormente.

- Los colores se expresan con valores RGB, igual que los que conocemos para los [colores HTML](#). Con la salvedad que un color se puede especificar también con tres números hexadecimales, en lugar de 6, como era obligatorio en HTML. Por ejemplo #fff equivaldría a #ffffff, o #123 sería #112233. Además, los colores se pueden especificar también en valores RGB decimales, con la notación rgb(r,g,b), siendo los valores de r, g, b números entre 0 y 255, por ejemplo rgb(100,0,255). Otra notación posible es rgb(r%,g%,b%), siendo cada uno de los r%,g%, b% un valor entre 0 y 100, por ejemplo rgb(100%,50%,0%), que sería todo de rojo, la mitad de verde y cero de azul.
- Otro tipo de valores que se pueden utilizar en las hojas de estilo en cascada son las URL, que sirven para especificar rutas hacia elementos como imágenes a colocar en fondos de elementos. Las URL en CSS se especifican con la notación url(valor), siendo valor la URL a la que queremos dirigirnos, que puede ser absoluta o relativa. Si es relativa, el navegador la interpreta desde el documento CSS donde estamos, si es que es un archivo CSS, o desde el documento HTML donde estamos, si es que los estilos los estamos colocando directamente en el archivo HTML. a URL se puede indicar con comillas dobles, simples o sin comillas. Por ejemplo:

url(http://www.desarrolloweb.com/images/miimagen.gif)

url("../images/otraimagen.jpg")

Hasta aquí, he explicado todo lo que debes saber por ahora con respecto a la sintaxis CSS y las unidades de medida CSS disponibles. Ha sido todo un poco teórico, pero en el siguiente capítulo podrás encontrar una lista de los atributos de las hojas de estilo en cascada, que te ayudarán a realizar ejercicios más prácticos.

ATRIBUTOS

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	font-size: 12pt; font-size: x-large;
Sirve para indicar el tamaño de las fuentes de una manera más rígida y con mayor exactitud.		
font-family	serif sans-serif cursive fantasy monospace	font-family: arial, helvetica; font-family: fantasy;
Con este atributo indicamos la familia de tipografía del texto. Los primeros valores son genericos, 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 font-weight: bold; 300 400 500 600 700 800 900 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:] text-decoration: underline;	none;
Para establecer la decoración de un texto, es decir, si está subrayado, subrayado 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: text-transform: capitalize;	none;
-----------------------	---	--	-------

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: background-color: #000055;	green;
-------------------------	--	---	--------

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)

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

Indicamos con este atributo el tamaño del margen a la izquierda

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

Se utiliza para definir el tamaño del margen a la derecha

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

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

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

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

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

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

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

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

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

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

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

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

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

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: border-color: #ffccff;	red;
---------------------	-----------------------------	---	------

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: double;		solid;
---------------------	---	--	--------

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

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

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

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>Ejemplodelautilizaci&ocute;ndeclasses</title>
<STYLEtype="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>

<pclass=letrasverdes>
Estoesunp&aacute;rroconestilodeletrasverdes</p>
<pclass=fondonegroletrasblancas>
Esto es un p&aacute;rro con estilo de fondo negro y las letras blancas. Es todo!</p>
</body>
</html>
```

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">Enlacevisitado</a>
Pulsaresteenlaceparaverloactivo,
poner el rat&oacute;n por encima para que cambie.

</body>
</html>

```

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.

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:

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.

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.

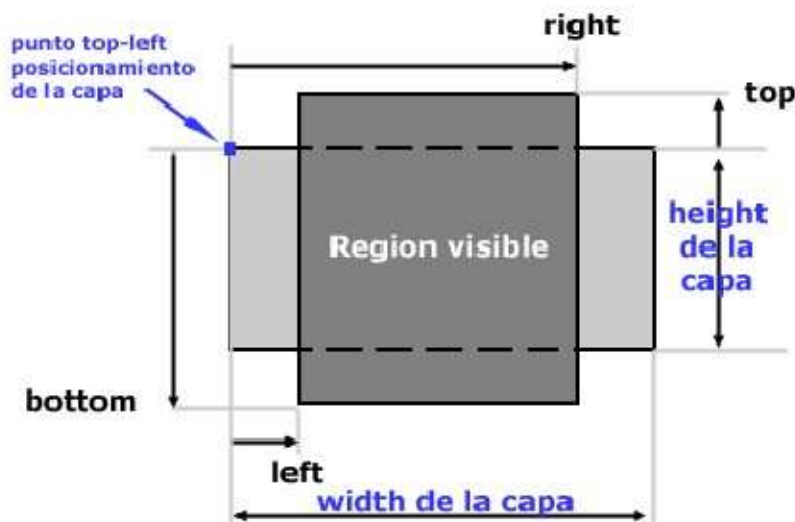
Dicho de otra forma, con z-index podemos decir qué capas se verán encima o debajo de otras, en caso de que estén superpuestas. El atributo z-index toma valores numéricos y a mayor z-index, más arriba se verá la página.

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.

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 cliping.

MAQUETACION CSS

vamos a conocer la maquetación de páginas utilizando Hojas de estilos en cascada (CSS). Veremos cómo realizar este tipo de maquetación, junto con algunas ventajas e inconvenientes. Para muchos será todavía un campo por explorar. Aunque no vamos a entrar en grandes detalles, vamos a intentar dar a conocer la maquetación con CSS para que cubrir la posible laguna por parte del lector. En capítulos sucesivos ampliaremos la información y ofreceremos tutoriales más prácticos.

Como se ha podido aprender en el [Manual de CSS](#), las hojas de estilo en cascada ayudan a separar el contenido de la forma, es decir, los elementos que componen una página de la forma con la que se muestran. Además, CSS ayuda en gran medida a la definición de estilos en la página, ya que permite ajustar de una manera mucho más precisa cualquier aspecto de cualquier elemento de la página.

La maquetación con CSS lleva la utilización de las hojas de estilo a su grado máximo, de manera que cualquier definición del aspecto de la página se realiza en la declaración CSS que enlazamos con el documento HTML. Para definir la situación de los elementos en la página se utilizan las capas, a las que se aplica un posicionamiento a través también de las hojas de estilo.

Para crear las capas se utilizan etiquetas <DIV>, en las que se introducen los elementos que queramos que aparezcan en la página. Los elementos dentro de los <DIV> también se pueden anidar, para heredar las propiedades y posicionamiento de las capas padre.

En la maquetación por capas se definen únicamente etiquetas <DIV> y las tablas sólo se utilizan para mostrar información tabulada, es decir, mostrada en filas y columnas. Cabe señalar que en la maquetación tradicional se utilizan las tablas para ajustar la posición de los elementos en la página. Seguro que muchos de los lectores están muy familiarizados con el uso de tablas para maquetar una web, pues se trata de una técnica bastante sencilla, aunque bastante poco práctica y que complica un tanto el código de las páginas web

resultantes.

La maquetación por tablas es una forma de crear webs más evolucionada y que mejora en mucho a la maquetación tradicional, aunque también tiene sus inconvenientes.

Veamos primero algunas de las ventajas de la maquetación CSS:

- La separación del contenido de la página y del estilo o aspecto con el que se deben mostrar. Tener en cuenta que, cuanto más separemos estos dos elementos, más sencillo será el mantenimiento de las páginas y el procesamiento de la información. Con ello también podremos obtener páginas más limpias y claras.
- Ahorro en la transferencia. Si todos los estilos y posiciones de los elementos se introducen en un documento externo, liberaremos el código de la página y ocupará mucho menos. Como la declaración de estilos se almacena en la caché del navegador, sólo se transfiere en la primera página que se visita del sitio, con lo que la segunda y posteriores páginas que se soliciten se cargarán mucho más rápido.
- Facilidad para alterar el aspecto de la página sin tocar el código HTML. Como toda la información de los estilos y el posicionamiento de las capas se encuentra en un mismo archivo, si deseamos cambiar cualquier elemento de la página -ya sea su posición o su aspecto-, sólo tenemos que actualizar la hoja de estilos y los cambios se verán automáticamente en todo el web.

Como decimos, también hay algunas desventajas:

- Compatibilidad con navegadores antiguos. Se necesita que el visitante disponga de un navegador bastante avanzado. La mayoría de los visitantes disponen de navegadores que soportan características avanzadas de las CSS, pero todavía hay mucha gente que no ha actualizado sus equipos o que navega en sistemas de sólo texto. Los navegadores que no soportan hojas de estilos, por lo menos leerán el código de la página y lo mostrarán sin ningún posicionamiento. Ello puede resultar fastidioso, pero por lo menos visualizarán todos los datos de la página, aunque descolocados y sin estilo.
- Diferencias entre navegadores. Dependiendo del navegador también cambian las etiquetas de estilos soportadas, por lo que las páginas pueden que no se vean exactamente igual en unos clientes que en otros. También, al igual que ocurre con HTML, hay atributos no estándar o que tienen valores por defecto diferentes. Cuando se empieza con la maquetación en CSS, puede resultar un tema bastante complicado y crearnos bastantes dolores de cabeza, no obstante, se trata de, poco a poco, ir aprendiendo todos los atributos y los navegadores donde se visualizan o no.

- Dificultad. Sin duda, si estamos acostumbrados al HTML, pasar a CSS resulta bastante más complicado y requiere un estudio más profundo. Sin embargo, este paso nos brindará un mayor control de los elementos de la página y ampliará nuestras fronteras a la hora de maquetar.

FORMAS DE APLICAR ESTILOS EN MAQUETACION

Aplicación de estilo a etiquetas

Se puede asignar el estilo a una etiqueta concreta de HTML. Para ello, en la declaración de estilos escribimos la etiqueta y entre llaves, los atributos de estilo que deseemos.

```
body {  
    background-color: #f0f0f0;  
    color: #333366;  
}
```

Podemos aplicar el mismo estilo en un conjunto de etiquetas. Para ello, indicamos las etiquetas seguidas por comas y luego, entre llaves, los atributos que queramos definir.

```
h1, p{  
    color: red;  
}
```

En este caso se define que los encabezados de nivel 1 y los párrafos, tengan letra roja.

Definición de clases

Podemos utilizar una clase si deseamos crear un estilo específico, para luego aplicarlo a distintos elementos de la página. Las clases en la declaración de estilos se declaran con un punto antes del nombre de la clase.

```
.miclase{  
    color: blue;  
}
```

Para asignar el estilo definido por una clase en un elemento HTML, simplemente se añade el atributo class a la etiqueta que queremos aplicar dicha clase. El atributo class se asigna al

nombre de la clase a aplicar. Por ejemplo:

```
<p class="miclase">este párrafo tiene el estilo definido en la clase "miclase".</p>
```

El párrafo anterior se presentaría con color azul. La definición de clases y su utilización es sencilla, pero veamos un ejemplo más detallado:

Para la siguiente declaración de estilos:

```
body, td, p{  
    background-color: #000000;  
    color: #ffffff;  
}  
  
.inverso{  
    background-color: #ffffff;  
    color: #000000;  
}
```

Se ha definido un fondo negro y color del texto blanco para el cuerpo de la página, así como las celdas y los párrafos. Luego se ha declarado una clase, de nombre "inverso", con los colores al revés, es decir, fondo blanco y texto negro.

```
<body>  
<p>Hola esto es un parrafo normal</p>  
<p class="inverso">Párrafo con los colores invertidos</p>  
<table>  
<tr>  
    <td class="inverso">INVERSO</td>  
    <td>NORMAL</td>  
</tr>  
</table>  
</body>
```

Esta página tiene, generalmente, el fondo negro y el texto blanco. El primer párrafo, que es un párrafo normal, sigue esa definición general de estilos, pero el segundo párrafo, al que se ha aplicado la clase "inverso", tiene el fondo blanco y el texto en negro. Por lo que respecta a la tabla, en su primera celda se ha asignado la clase "inverso", por lo que se verá con fondo

blanco y color de texto en negro. Mientras que la segunda celda, que no tiene asignada ninguna clase, se presentará como se definió en la regla general.

Estilos que sólo se utilizan una vez

También podemos tener un estilo específico para un único elemento, que no va a repetirse en ningún otro caso. Para ello tenemos los estilos asignados por identificador. Los identificadores se definen en HTML utilizando el atributo id en la etiqueta que deseamos identificar. El valor del atributo id será el que definamos nosotros.

```
<div id="capa1">
```

En la hoja de estilos, para definir el aspecto de ese elemento con id único, se escribe el carácter almohadilla, seguido del identificador indicado en la etiqueta y entre llaves los atributos css que deseemos.

```
#capa1{  
    font-size: 12pt;  
    font-family: arial;  
}
```

En este caso se ha asignado fuente de tamaño 12 puntos y cuerpo arial.

Como se puede concluir en la lectura de estas líneas, generalmente se prefiere utilizar estilos definidos en clases a los definidos con identificadores, a no ser que estemos seguros que ese estilo no se va a repetir en todo el documento.

COLORES EN CSS

Con CSS se puede especificar colores para cada elemento HTML de la página, incluso hay elementos que podrían admitir varios colores, como el color de fondo o el color del borde. Pero bueno, vamos a ver ahora es las distintas maneras de escribir un color en una declaración CSS.

Porque lo más habitual es que especifiquemos un color con su valor RGB, de una manera similar a como aprendimos a [definir colores en HTML](#). Pero en CSS tenemos otras maneras de declarar colores que pueden interesarnos, como mínimo para poder entender el código CSS cuando lo veamos escrito.

Notación hexadecimal RGB

Esta notación es la que ya conocemos. Se especifican los tres valores de color (rojo, verde y azul) con valores en hexadecimal entre 00 y FF.

```
background-color: #ff8800;
```

Notación hexadecimal abreviada

Esta notación es muy parecida a la anterior, pero permite abreviar un poco la declaración del color, indicando sólo un número para cada valor rojo, verde y azul. Por ejemplo, para especificar el color de antes (#ff8800) podríamos haber escrito:

```
background-color: #f80;
```

Nombre del color

También podemos definir un color por su nombre. Los nombres de colores son en inglés, los mismos que sirven para especificar colores con HTML.

```
color: red;
border-color: Lime;
```

Notación de color con porcentajes de RGB

Se puede definir un color por los distintos porcentajes de valores RGB. Si todos los valores están al 100% el color es blanco. Si todos están al 0% obtendríamos el negro y con combinaciones de distintos porcentajes de RGB obtendríamos cualquier matiz de color.

```
color: rgb(33%, 0%, 0%);
```

Notación por valores decimales de RGB, de 0 a 255

De una manera similar a la notación por porcentajes de RGB se puede definir un color directamente con valores decimales en un rango desde 0 a 255.

```
color: rgb(200,255,0);
```

De entre todas estas notaciones podemos utilizar la que más nos interese o con la que nos sintamos más a gusto. Nosotros en nuestros ejemplos venimos utilizando la notación hexadecimal RGB por habernos acostumbrado a ella en HTML.

Color transparente

Para finalizar, podemos comentar que también existe el color transparente, que no es ningún color, sino que especifica que el elemento debe tener el mismo color que el fondo donde está. Este valor, `transparent`, sustituye al color. Podemos indicarlo en principio sólo para fondos de elementos, es decir, para el atributo `background-color`.

```
background-color: transparent;
```