

Hojas de estilo CSS

[Curso de Diseño de Páginas Web]

4

1. INTRODUCCIÓN A LAS HOJAS DE ESTILO CSS.	1
CONCEPTOS PREVIOS	1
CARACTERÍSTICAS Y VENTAJAS DE LAS CSS	1
NAVEGADORES QUE LO SOPORTAN	2
2. COMO APLICAR ESTILOS CSS.	3
PEQUEÑAS PARTES DE LA PÁGINA	3
ESTILO DEFINIDO PARA UNA ETIQUETA	3
ESTILO DEFINIDO EN UNA PARTE DE LA PÁGINA	4
ESTILO DEFINIDO PARA TODA UNA PÁGINA	4
ESTILO DEFINIDO PARA TODO UN SITIO WEB	5
1- CREAMOS EL FICHERO CON LA DECLARACIÓN DE ESTILOS.	5
2- ENLAZAMOS LA PÁGINA WEB CON LA HOJA DE ESTILOS	6
REGLAS DE IMPORTANCIA EN LOS ESTILOS	7
3. SINTAXIS CSS	8
URL COMO VALOR DE UN ATRIBUTO	8
4. ATRIBUTOS DE LAS HOJAS DE ESTILO	9
5. TRUCOS AVANZADOS CON CSS	13
DEFINIR ESTILOS UTILIZANDO CLASES	13
EJEMPLO DE LA UTILIZACIÓN DE CLASES	13
ESTILO EN LOS ENLACES	13
ENLACES NORMALES	14
ENLACES VISITADOS	14
ENLACES ACTIVOS	14

ENLACES HOVER	14
OCULTAR ESTILOS EN NAVEGADORES ANTIGUOS	15
CREACIÓN DE UNA BARRA DE NAVEGACIÓN CON ESTILOS CSS	15
CÓMO PONER ESTILO A LOS ENLACES	15
CÓMO DAR ESTILO A UN ENLACE EN CONCRETO	15
CÓMO UTILIZAR LAS CLASES AL APLICAR ESTILO A LOS ENLACES	16
EJEMPLOS DE ESTILOS PARA LOS FORMULARIOS	18
6. CAPAS	19
ATRIBUTOS PARA CAPAS	19
ATRIBUTO POSITION	19
ATRIBUTO TOP	20
ATRIBUTO LEFT	20
ATRIBUTO HEIGHT	20
ATRIBUTO WIDTH	20
ATRIBUTO VISIBILITY	20
ATRIBUTO Z-INDEX	20
ATRIBUTO CLIP	21

1. Introducción a las hojas de estilo CSS.

Conceptos previos

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.

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 sitio 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...

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.

Sin embargo, todos los navegadores en sus últimas versiones soportan esta tecnología, ya que está incluida dentro del nuevo estándar XHTML.

2. Como aplicar estilos CSS.

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.

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 estilos

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 p&acute;aquina</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&acute;aquina con estilos</h1>
Bienvenidos...
<p>Esto es otro ejemplo m&acute;as</p>
</body>
</html>
```

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

- Subrellado
- 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.

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="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>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>

```

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)

3. Sintaxis CSS

Tal como se vio 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

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)`

4. 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.

FUENTES - FONT		
Nombre del atributo	Posibles valores	Ejemplos
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 estándar, 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.

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=ningún borde, dotted=punteado (no parece funcionar), solid=sólido, 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.

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.

5. 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 utilizaci&ocute;n 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 p&aacute;rrafo con estilo de letras verdes</p>
<p class=fondonegroletrasblancas>
Esto es un p&aacute;rrafo 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">Enlace visitado</a>
Pulsar este enlace para verlo activo,
poner el ratón por encima para que cambie.
</body>
</html>
```

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:

Creación de una barra de navegación con estilos CSS

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.

Cómo poner estilo a los enlaces

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.

Enlace 1
Otro enlace
Enlace 3

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 declaración.

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>
```

Ejemplos de estilos para los formularios

Crear un formulario y aplicarle los estilos siguientes

```
INPUT {FONT-SIZE: 11px;
        FONT-FAMILY: Verdana,Helvetica;
        color: #FFFFFFf;
        background-color : #C9CCCF;
        border: #FFFFFFf;
        border-style: solid;
        border-top-width : 1px;
        border-right-width : 1px;
        border-bottom-width : 1px;
        border-left-width : 1px}

input.button {border-top-width : 1px;
               border-right-width : 1px;
               border-bottom-width : 1px;
               border-left-width : 1px}

textarea {font-size: 11px;
           font-family: Verdana,Helvetica;
           color: #ffffff;
           background-color : #768089;
           border: #ffffff;
           border-style: solid;
           border-top-width : 1px;
           border-right-width : 1px;
           border-bottom-width : 1px;
           border-left-width : 1px}

select {font-size: 11px;
        background-color : #C9CCCF;
        color : #ffffff;
        font-family: Verdana, Geneva, Arial, Helvetica;
        border: #ffffff;
        border-style: solid;
        border-top-width : 1px;
        border-right-width : 1px;
        border-bottom-width : 1px;
        border-left-width : 1px}
```

6. Capas

Como ya hemos visto, y <DIV> sirven para aplicarle estilo a partes de una página HTML.

La diferencia entre y <DIV> es que con esta última sí 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 el uso más destacado de <DIV> 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.

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.

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.

Atributos para capas

Hemos visto en el capítulo anterior qué son las capas y algunas pequeñas muestras sobre cómo crearlas y darles algún estilo.

Ahora vamos a ver 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 cuatro valores, static, absolute, relative o fixed:

- relative indica que las distancias que se indiquen serán relativas al 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.
- Fixed indica que la capa no se moverá aunque se muevan las barras de scroll.
- Static es el valor por defecto, indica que la capa será colocada justo donde estamos escribiendo la etiqueta div.

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

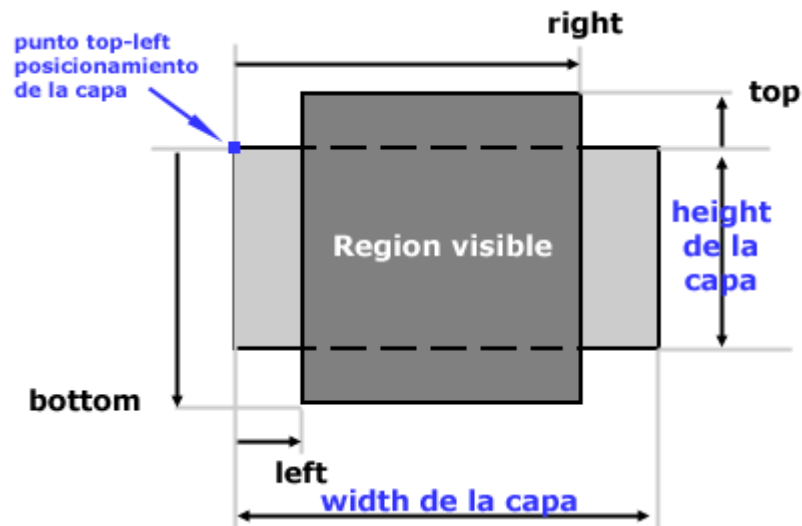
Sirve para indicar la posición sobre el eje z que tendrán las distintas capas de la página. Dicho de otra forma, con z-index podemos decir qué capas se verán encima o debajo de otras, en caso de que estén superpuestas. El atributo z-index toma valores numéricos y a mayor z-index, más arriba se verá la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


7. Posicionamiento CSS

Otra forma de colocar los elementos en las páginas es mediante lo que se conoce con el nombre de CSS-P o Posicionamiento CSS.

Consiste en crear definiciones de estilo para las diferentes colocaciones de elementos en la página y darles un nombre a cada una de ellas. Por ejemplo:

```
#contenedor{  
position: absolute;  
left: 50px;  
top: 100px;  
width: 100px;  
height:100px;  
}
```

El elemento al que le asociemos este estilo estará contenido en un rectángulo rojo de 100px de ancho por 100 pixels de alto a 50 pixels del borde izquierdo de la página y a 100 pixels del superior.

Para asociar el estilo a un elemento de la página se utilizaría la etiqueta div con el parámetro id de la siguiente forma:

```
<div id="contenedor">Este es un texto de prueba</div>
```

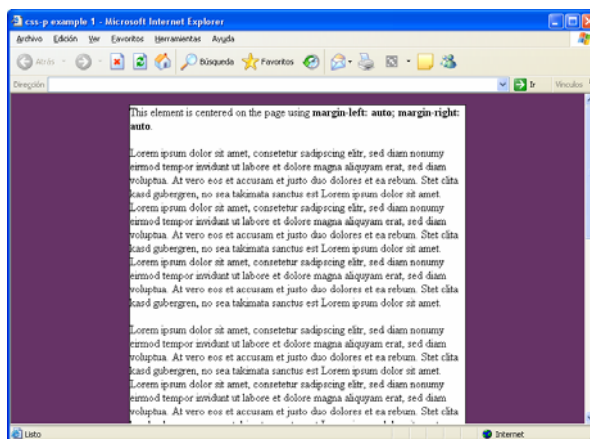
Esto hará que el texto de prueba se coloque dentro del rectángulo rojo.

Actualmente el W3C recomienda la utilización del CSS-P para la maquetación de las páginas en lugar de la utilización de tablas, ya que se apuesta por una separación entre el contenido y la forma de colocarlo, de ahí que hayan aparecido las hojas de estilo.

En principio puede parecer difícil maquetar páginas utilizando las propiedades CSS ya que estamos acostumbrados a utilizar tablas. Vamos a ver algunos ejemplos que muestran cómo maquetar las páginas utilizando CSS, pero con la apariencia de tablas.

Centrar el contenido horizontalmente

Es una forma muy común de colocar el contenido en la página, todo centrado en un único bloque independientemente del tamaño de la ventana del navegador. Hay varias formas de hacer esto con CSS. El resultado que pretendemos es el que se muestra en la imagen:



Utilizando margin:auto

Este es el método más sencillo para centrar páginas:

```
#contenido {
width: 60%;
margin-left: auto;
margin-right: auto;
margin-top: auto;
margin-bottom: auto;
border: 1px solid black;
background-color: white;
}
```

Sin embargo el valor auto no funciona bien para margin-top o margin-bottom, y no es reconocido por los navegadores de versión 4.0 e IE5.0

Utilizando porcentajes para width, margin-left y margin-right

En este ejemplo, la anchura del elemento **contenido** es de 60%, con un margen izquierdo y derecho de 20%. Esto funciona incluso con Netscape 4. El código es el siguiente:

```
#contenido {
border-right: black 1px solid;
border-top: black 1px solid;
margin-left: 20%;
border-left: black 1px solid;
width: 60%;
margin-right: 20%;
border-bottom: black 1px solid;
background-color: white
}
```

Utilizando márgenes negativos y posición absoluta

El código es el siguiente:

```
#contenido {  
  position: absolute;  
  width: 500px;  
  left: -250px;  
  margin-left: 50%;  
}
```

La anchura del elemento es fija y le damos un margen izquierdo de 50%, lo cual indicaría que el borde izquierdo del recuadro aparecería justo en el centro de la página. Para corregirlo, indicamos que la posición izquierda sea justo la mitad de la anchura en negativo, con lo que el cuadro aparecerá centrado.

Utilizando porcentajes que completen el 100%

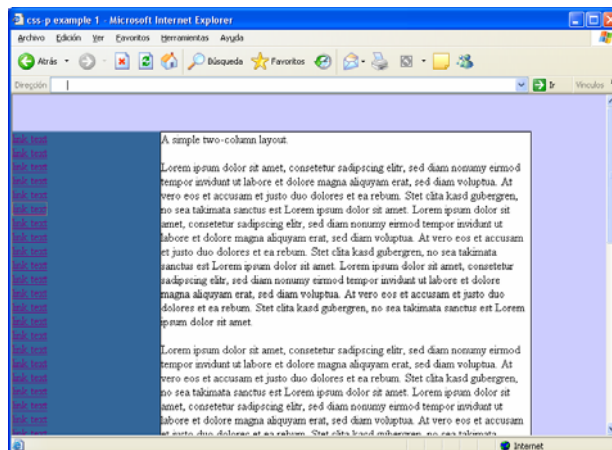
El código es el siguiente:

```
select {  
  width: 70%; height: 70%;  
  margin: 15% 15% 15% 15%;  
}
```

Esto no funciona bien en todos los navegadores y el centrado horizontal sale mayor que el vertical.

Distribuir el contenido en dos columnas

Otra distribución muy utilizada en las páginas web es la de dos columnas. Normalmente el menu de navegación está a la izquierda y el contenido en la derecha tal como se muestra en la imagen:



Dos columnas utilizando valores absolutos

Es el método más simple. Utilizaremos el id #navegación para la parte del menú y #contenidos para la principal. El código es el siguiente:

```
#navegacion {  
  position: absolute;  
  left: 0px; top: 50px;  
  width: 200px;  
}
```

```
#contenido {  
  position: absolute;  
  top: 50px;  
  width: 500px;  
  left: 200px;  
}
```

Dos columnas movibles utilizando valores relativos

El código es el siguiente:

```
#navegación {  
  position: absolute;  
  top: 0;  
  left: 0px;  
  width: 20%;  
  background-color: #336699;  
}
```

```
#contenido {  
  position: absolute;  
  top: 0;  
  left: 20%;  
  width: 80%;  
  margin-left: 20%;  
  border: 1px solid black;  
  background-color: white;  
}
```

En algunos navegadores puede ocurrir que aparezca la barra de scroll horizontal aunque no sea necesaria, pero se puede corregir si hacemos que el elemento del contenido ocupe un porcentaje un poco menor de forma que no llegue al 100% de la página.

Dos columnas utilizando la propiedad float

La propiedad float hace que el contenido del elemento “fluya” alrededor del elemento que le precede, en este caso el menú. Con esta propiedad le podemos indicar por qué lado del menú queremos que “fluya” el contenido.

```
#navegación {  
  position: absolute;  
  top: 0;  
  left: 0;  
  margin-top: 0; margin-bottom: 0;  
  width: 20%;  
  background-color: #336699;  
}
```

```
#contenido {  
  position: relative;  
  top: 0;  
  width: 80%;  
  float: right;  
  border: 1px solid black;  
  background-color: white;  
}
```

Para que esto funcione el elemento que no “fluye” debe tener la propiedad **position: absolute**.

Distribución en tres columnas

Vamos a seguir los pasos para crear una distribución en tres columnas con una separación entre ellas de 9 pixels.

1. Primero indicamos el margen de la página y el color de fondo. Indicamos también padding porque el navegador Opera no lo tiene a 0 por defecto. También indicamos el color de fondo a blanco porque Netscape y Macintosh tiene el fondo de color gris por defecto.

```
body {  
    margin:9px 9px 0 9px;  
    padding:0;  
    background:#FFF;  
}
```

2. Ahora definimos nuestro primer elemento, un DIV al que llamaremos level0. El único parametro para este elemento sera el color de fondo que es el que queremos para la columna izquierda.

```
#level0 {  
    background:#FC0;  
}
```

3. Añadimos una segunda definición llamada level1

```
#level1 {  
    margin-left:143px;  
    padding-left:9px;  
    background:#FFF;  
}
```

Esto hace que haya un espacio entre la primera y la segunda columna si colocamos el elemento level1 dentro del level0, tal como se muestra

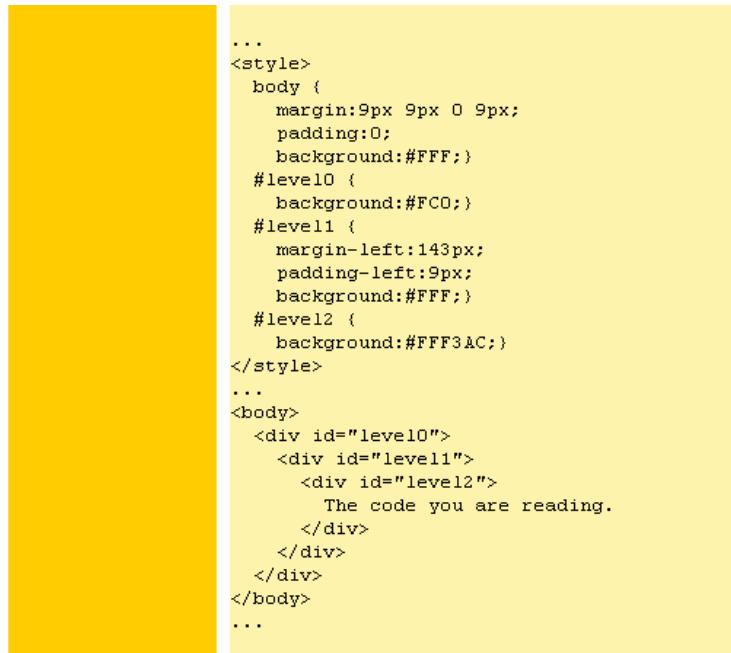


```
...  
<style>  
    body {  
        margin:9px 9px 0 9px;  
        padding:0;  
        background:#FFF;  
    }  
    #level0 {  
        background:#FC0;  
    }  
    #level1 {  
        margin-left:143px;  
        padding-left:9px;  
        background:#FFF;  
    }  
</style>  
...  
<body>  
    <div id="level0">  
        <div id="level1">  
            The code you are reading.  
        </div>  
    </div>  
</body>  
...
```

4. Ahora añadimos dentro de level1 un nuevo elemento level2

```
#level2 {  
    background:#FFF3AC;  
}
```

Sólo hacemos que cambia el color de fondo, para que se note la columna central y la separación.



5. Ahora usamos la misma técnica para la segunda columna, añadiendo un Nuevo elemento level3 dentro de level2. Este elemento dibujará un fondo blanco manteniendo un margen derecho que coincide con el ancho de la columna de la derecha.

```
#level3 {  
    margin-right:143px;  
    padding-right:9px;  
    background:#FFF;  
}  
#main {  
    background:#CCC;  
}
```

Por ultimo añadimos el elemento main solo para indicar el color de fondo y hacer que se muestren las separaciones entre las columnas.

El aspecto final es el que se muestra a continuación.

```

...
<style>
  body {
    margin:9px 9px 0 9px;
    padding:0;
    background:#FFF;}
  #level0 {
    background:#FC0;}
  #level1 {
    margin-left:143px;
    padding-left:9px;
    background:#FFF;}
  #level2 {
    background:#FFF3AC;}
  #level3 {
    margin-right:143px;
    padding-right:9px;
    background:#FFF;}
  #main {
    background:#CCC;}
</style>
...
<body>
  <div id="level0">
    <div id="level1">
      <div id="level2">
        <div id="level3">
          <div id="main">
            The code you are reading.
          </div>
        </div>
      </div>
    </div>
  </div>
</body>
...

```