# 2.6.6. Posicionamiento

CSS tiene 5 formas de establecer la posición en pantalla de una caja: Estático, Relativo, Absoluto, Fijo y Flotante.

Para establecer la posición disponemos de las propiedades: position, top, bottom, right, left y float.

**La propiedad position**

* **Establece la posición de la caja en la pantalla.**
* **Valores**: static | relative | absolute | fixed |inherit
* **Aplicación: a todos los elementos.**
* **El valor por defecto es static**

**- static: la caja ignora las propiedades** top, bottom, right, left. Es el valor por defecto e indica la posición normal del elemento.

**- relative: la caja se desplaza según los valores top, bottom, right, left** . Las coordenadas se calculan de forma relativa a la posición que ocuparía el elemento normalmente.

-**absolute: la caja se desplazará según los valores top, bottom, right, left** partiendo de su elemento contenedor. Las coordenadas se indican de forma absoluta respecto al resto de los elementos del documento.

-**fixed**: **Las coordenadas se indican de forma absoluta respecto al medio de visualización, por ejemplo en la pantalla**. Este posicionamiento es como el absoluto pero la caja permanece inmóvil independientemente del resto de elementos y del movimiento de la página por parte del usuario.

**-inherit:** heredado.

**Propiedades: top, bottom, right,left**

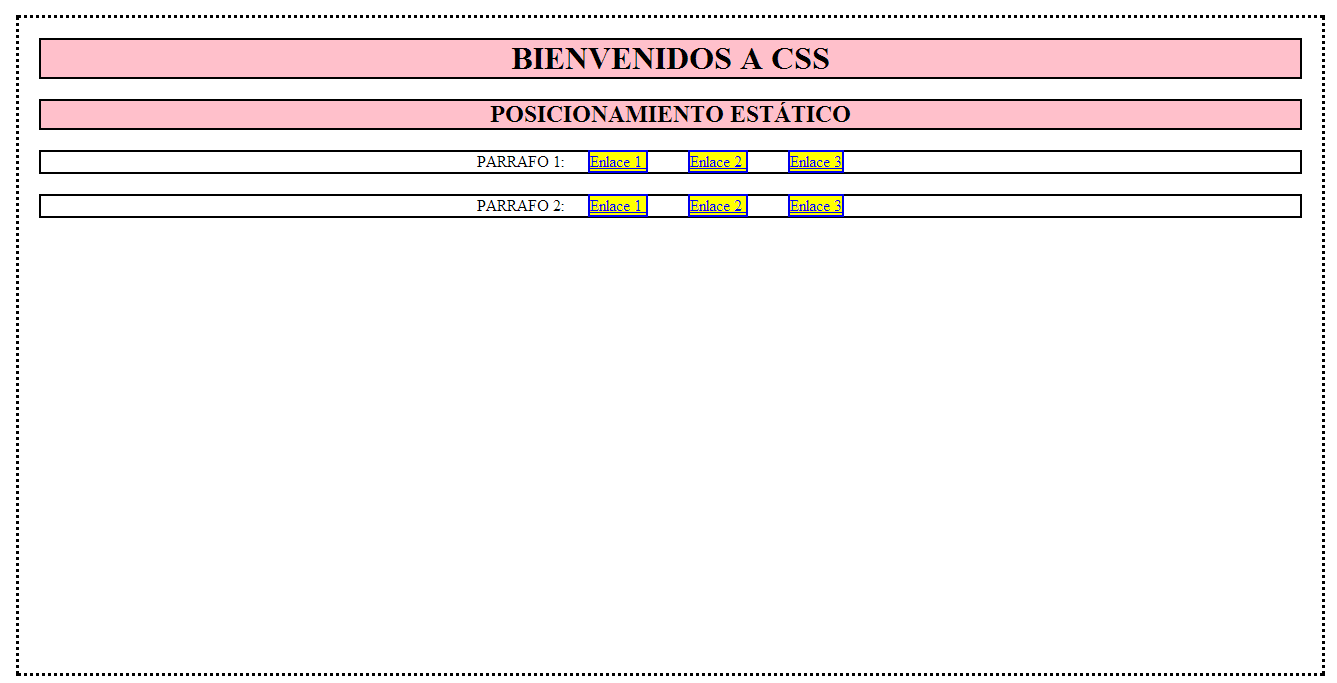
* **Establece desplazamiento vertical y horizontal de la caja.**
* **Valores <nº px>|<nº%>|auto|inherit.**
* **Se aplica a todos los elementos**
* **El valor por defecto es auto**

|  |  |
| --- | --- |
| - Posicionamiento estático |  |

El posicionamiento por defecto que utilizan todos los navegadores.

* Si los elementos a posicionar son de bloque, como un párrafo, las cajas se colocarán verticalmente una debajo de otra, dentro del elemento contenedor.
* Si los elementos a posicionar son de línea, como un hipervínculo, las cajas se colocarán una junto a otra horizontalmente dentro del elemento bloque.
* Si las cajas no caben en una línea, entonces ocuparan la siguiente, etc y si caben en una línea se pueden alinear horizontalmente.

{position:static}

EJEMPLO

<html>

<head><title>posicionamiento estatico</title></head>

<style>

\*{border: 2px solid; text-align: center; margin:20px;}

html{border:4px solid;}

body{border:dotted;}

h1, h2{background-color:pink;}

a{background-color:yellow;}

</style>

<body>

<h1>BIENVENIDOS A CSS</h1>

<h2> POSICIONAMIENTO ESTÁTICO</h2>

<p> PARRAFO 1:

<a href="#"> Enlace 1 </a>

<a href="#"> Enlace 2 </a>

<a href="#"> Enlace 3 </a>

</p>

<p> PARRAFO 2:

<a href="#"> Enlace 1 </a>

<a href="#"> Enlace 2 </a>

<a href="#"> Enlace 3 </a>

</p>

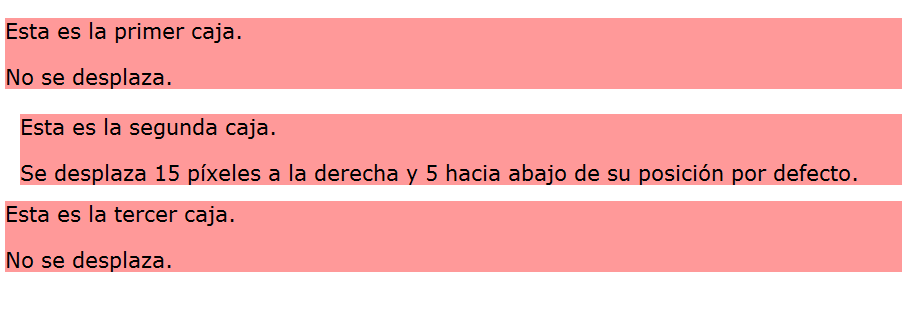
</body>

</html>

|  |  |
| --- | --- |
| - Posicionamiento relativo. |  |

Consiste en desplazar una cantidad con respecto al posicionamiento estático.{position:static}

**La propiedad position** determina el punto de referencia donde se debe localizar cada elemento HTML. Por defecto esta propiedad se inicializa con el **valor static.**  
Con el valor por defecto static, cada elemento HTML se localiza de izquierda a derecha y de arriba hacia abajo.  
El segundo valor posible para esta **propiedad es relative**. En caso de fijar la propiedad position con el valor relative, podemos modificar la posición por defecto del elemento HTML modificando los valores left y top (con valores positivos o inclusive negativos).  
Veamos un ejemplo con tres div, de los cuales el segundo lo desplazamos 20 pixeles a nivel de columna y 5 pixeles a nivel de fila:



<html>

<head>

<title>Problema</title>

<link rel="StyleSheet" href="estilos.css" type="text/css">

</head>

<body>

<div id="caja1">

<p>Esta es la primer caja.</p>

<p>No se desplaza.</p>

</div>

<div id="caja2">

<p>Esta es la segunda caja.</p>

<p>Se desplaza 15 píxeles a la derecha y 5 hacia abajo de su posición

por defecto.</p>

</div>

<div id="caja3">

<p>Esta es la tercer caja.</p>

<p>No se desplaza.</p>

</div>

</body>

</html>

La hoja de estilo asociada es:

#caja1,#caja2,#caja3 {

background-color:#f99;

font-family:verdana;

font-size:1.3em;

}

#caja2 {

position:relative;

left:15px;

top:5px;

}

Recordemos que cuando un conjunto de elementos tienen los mismos valores para una serie de propiedades, los podemos agrupar separándolos por coma, esto sucede para los tres Id #caja1,#caja2 y #caja3 que tienen los mismos valores para las propiedades background-color, font-family y font-size:

#caja1,#caja2,#caja3 {

background-color:#f99;

font-family:verdana;

font-size:1.3em;

}

Luego como debemos inicializar la propiedad position sólo para el Id #caja3 lo hacemos en forma separada:

#caja2 {

position:relative;

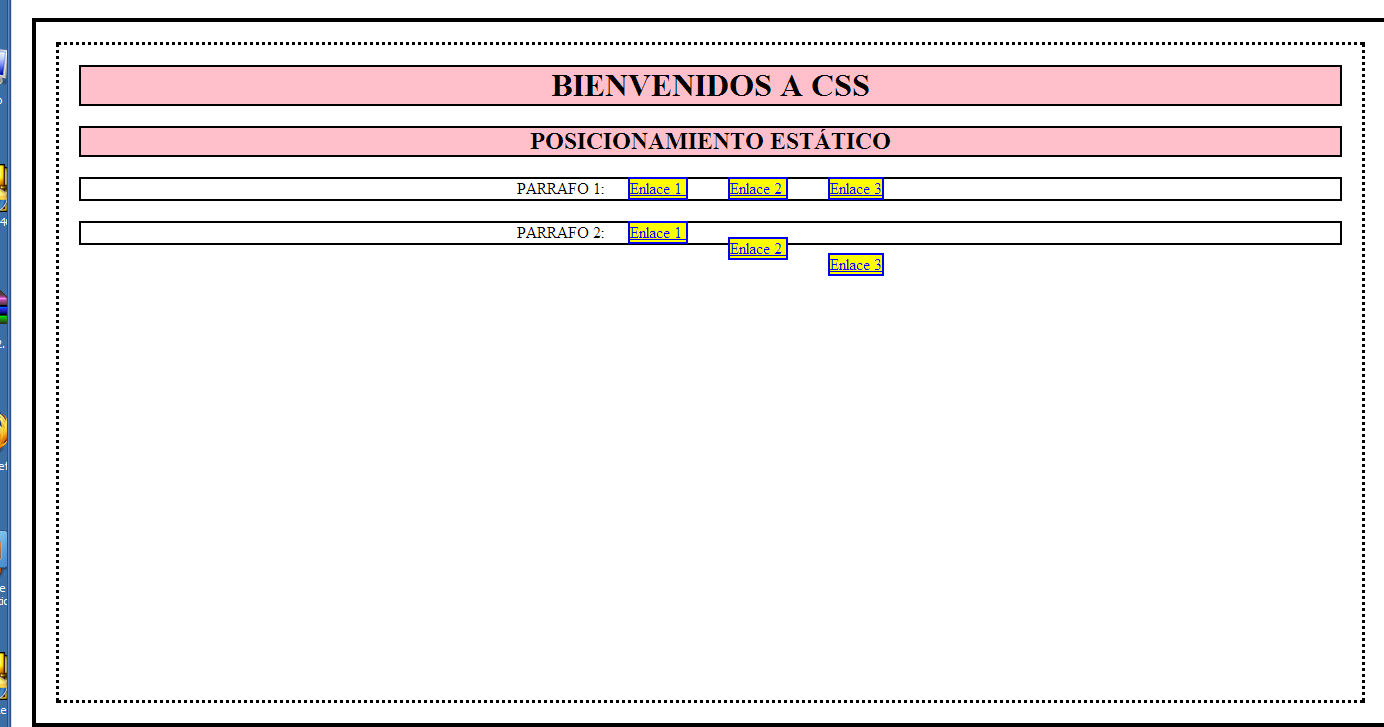
left:15px;

top:5px;

}

Aquí es donde inicializamos la propiedad position con el valor relative y desplazamos el elemento 15 píxeles a la derecha y 5 píxeles hacia abajo. Tengamos en cuenta que si asigno un valor muy grande a la propiedad top se superpone este elemento con el contenido del tercer div.

**EJEMPLO**



<html>

<head><title>posicionamiento RELATIVO</title></head>

<style>

\*{border: 2px solid; text-align: center; margin:20px;}

html{border:4px solid;}

body{border:dotted;}

h1, h2{background-color:pink;}

a{background-color:yellow;}

a.abajo{position:relative; top: 1em;}

a.mas\_abajo{position:relative; top: 2em;}

</style>

<body>

<h1>BIENVENIDOS A CSS</h1>

<h2> POSICIONAMIENTO ESTÁTICO</h2>

<p> PARRAFO 1:

<a href="#"> Enlace 1 </a>

<a href="#"> Enlace 2 </a>

<a href="#"> Enlace 3 </a>

</p>

<p> PARRAFO 2:

<a href="#"> Enlace 1 </a>

<a href="#" class="abajo"> Enlace 2 </a>

<a href="#" class="mas\_abajo"> Enlace 3 </a>

</p>

</body>

</html>

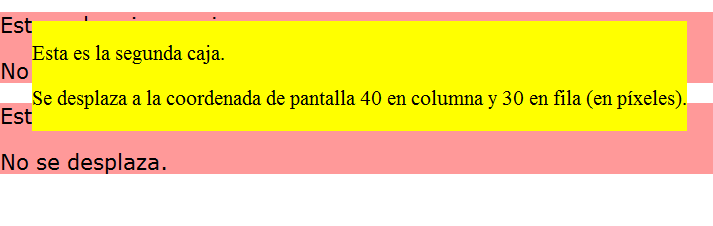
Como se puede observar, la caja “Enlace2” se ha desplazado 1em hacia abajo desde su posición normal (static) y la caja “Enlace3” se ha desplazado 2em.

|  |  |
| --- | --- |
| - Posicionamiento absoluto. |  |

El posicionamiento absoluto dispone un elemento HTML completamente fuera del flujo de la página. El valor que debemos asignar **a la propiedad position es absolute.**Hay que tener en cuenta que no se reserva espacio en el flujo del documento como pasaba con el posicionamiento relativo (recordemos que con este posicionamiento podemos desplazar el elemento a cualquier parte de la página, pero el espacio por defecto para dicho elemento queda vacío). El posicionamiento es siempre con respecto a la página.

* + **El posicionamiento absoluto consiste en desplazar la caja una cantidad respecto de un elemento contenedor.**
  + **La caja sale del flujo normal de la página y el resto de cajas ocupan su lugar.**
  + **Para el desplazamiento se usan top, bottom, left, right, y se toma como origen de coordenadas la esquina superior izquierda del primer contenedor que no sea static**.

**Position:absolute**  
Veamos un ejemplo para ver el funcionamiento del posicionamiento absoluto:

****

<html>

<head>

<title>Problema</title>

<link rel="StyleSheet" href="estilos.css" type="text/css">

</head>

<body>

<div id="caja1">

<p>Esta es la primer caja.</p>

<p>No se desplaza.</p>

</div>

<div id="caja2">

<p>Esta es la segunda caja.</p>

<p>Se desplaza a la coordenada de pantalla 40 en columna y 30 en fila (en

píxeles).</p>

</div>

<div id="caja3">

<p>Esta es la tercer caja.</p>

<p>No se desplaza.</p>

</div>

</body>

</html>

La hoja de estilo definida:

#caja1,#caja3 {

background-color:#f99;

font-family:verdana;

font-size:1.3em;

}

#caja2 {

background-color:#ff0;

font-size:1.3em;

position:absolute;

left:40px;

top:30px;

}

Como vemos inicializamos la propiedad position con el valor absolute y fijamos como coordenada para la caja la columna 40 y la fila 30 (en píxeles).

**EJEMPLO\_1: POSICIONAMIENTO ABSOLUTO**



En este caso “Enlace1” se desplaza 30px desde arriba y desde la izquierda, partiendo desde la esquina superior izquierda de la página, porque <div> tiene posición normal y el siguiente es <body> (observar que además hay 20 px de margen).

Las cajas “Enlaces2” y “Enlaces3” ocupan el hueco dejado por la caja “”Enlace1”.

<html>

<head><title>posicionamiento ABSOLUTO</title></head>

<style>

\*{ text-align: center; margin:20px;padding:5px;}

html{border:4px solid;}

body{border:dotted;}

h1, h2{background-color:pink; border:2px solid ;}

div{border: 2 px solid;}

a{background-color:yellow;border:2px solid ;}

.absoluto{background-color:orange; position:absolute; top: 30px;left:30 px ;}

</style>

<body>

<h1>BIENVENIDOS A CSS</h1>

<h2> POSICIONAMIENTO ABSOLUTO</h2>

<div>

<p>Esta caja tienen posición normal o estática</p>

<a href="#" class="absoluto"> Enlace 1 </a>

<a href="#"> Enlace 2 </a>

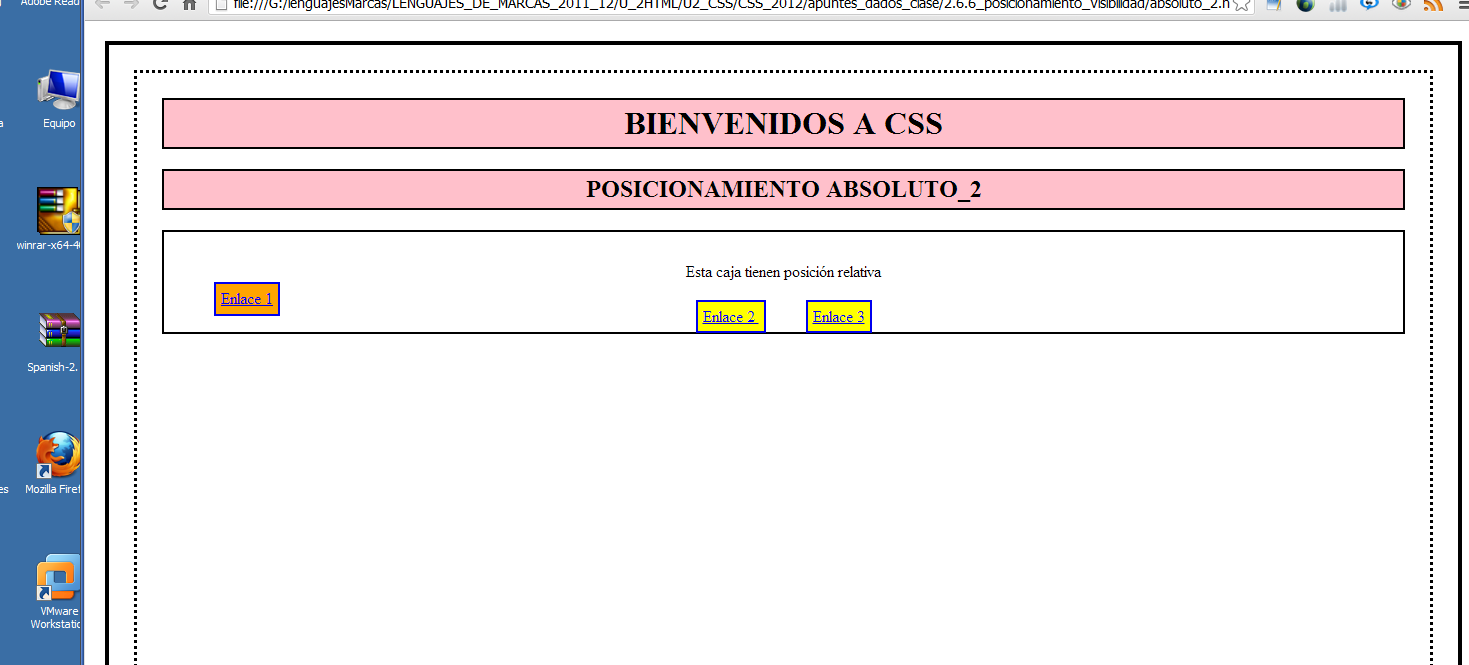
<a href="#"> Enlace 3 </a>

</div>

</body>

</html>

**EJEMPLO\_2 OTRO EJEMPLO DE POSICIONAMIENTO ABSOLUTO**



En este caso el “Enlace1” se desplaza 30 px desde arriba y desde la izquierda, pero partiendo de su caja contenedora, que es la caja <div> porque es la primera caja con posicionamiento no estático( tiene posición relativa).

<html>

<head><title>posicionamiento ABSOLUTO</title></head>

<style>

\*{ text-align: center; margin:20px;padding:5px;}

html{border:4px solid;}

body{ border:dotted;}

h1, h2{background-color:pink; border:2px solid ;}

div{position:relative;border: 2 px solid;}

a{background-color:yellow;border:2px solid ;}

a.absoluto{background-color:orange; position:absolute; top: 30px;left:30 px ;}

</style>

<body>

<h1>BIENVENIDOS A CSS</h1>

<h2> POSICIONAMIENTO ABSOLUTO</h2>

<div>

<p>Esta caja tienen posición normal o estática</p>

<a href="#" class="absoluto"> Enlace 1 </a>

<a href="#"> Enlace 2 </a>

<a href="#"> Enlace 3 </a>

</div>

</body>

</html>

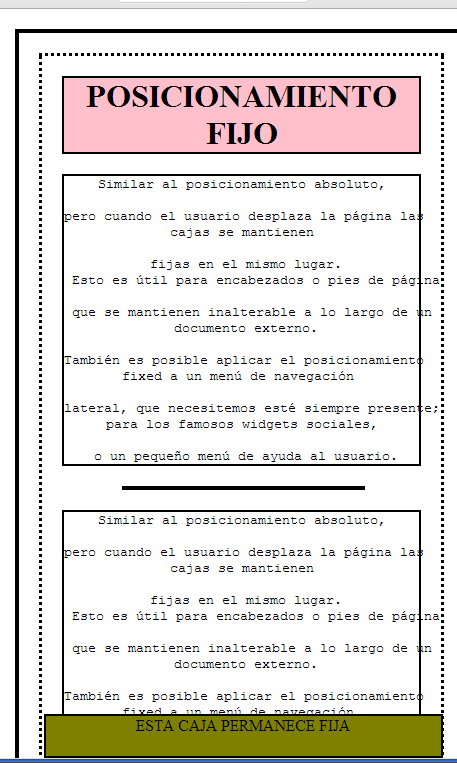
# -Posicionamiento fixed

Similar al posicionamiento absoluto, pero cuando el usuario desplaza la página, las cajas se mantienen fijas en el mismo lugar.

Esto es útil para encabezados o pies de página que se mantienen inalterable a lo largo de un documento externo.

También es posible aplicar el posicionamiento fixed a un menú de navegación lateral, que necesitemos esté siempre presente; para los famosos widgets sociales, o un pequeño menú de ayuda al usuario.

**EJEMPLO: Se puede copiar varias veces el mismo párrafo para comprobar que la caja inferior permanece fija.**



<html>

<head><title>posicionamiento fixed</title></head>

<style>

\*{ border: 2px solid; text-align: center; margin:20px;}

html{border:4px solid;}

body{ border:dotted;}

h1{background-color:pink;}

.fijo{position:fixed; bottom:5%; width: 80%; height:40px;

margin:0; left:10%; background-color:olive;}

</style>

<body>

<h1> POSICIONAMIENTO FIJO</h1>

<p >

<pre >Similar al posicionamiento absoluto,<br>

pero cuando el usuario desplaza la página las

cajas se mantienen<br>

fijas en el mismo lugar.

Esto es útil para encabezados o pies de página<br>

que se mantienen inalterable a lo largo de un

documento externo.<br>

También es posible aplicar el posicionamiento

fixed a un menú de navegación <br>

lateral, que necesitemos esté siempre presente;

para los famosos widgets sociales,<br>

o un pequeño menú de ayuda al usuario.

</pre>

</p>

<p class="fijo">ESTA CAJA PERMANECE FIJA</p>

</body>

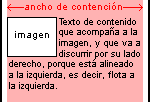
</html>

# -Posicionamiento flotante.

* Desplaza las cajas todo lo posible hacia la izquierda o hacia la derecha de la línea en la que se encuentran.
* El resto de las cajas ocupan el lugar dejado por la caja flotante.
* No se produce solapamiento de cajas porque la propiedad float tiene en cuenta el resto de las cajas flotantes.
* Los elementos situados alrededor de la caja se adaptan a la posición de la caja, lo que hace que sea idónea para la combinación de texto e imágenes.

Con esta propiedad especificaremos la **disposición de un elemento alrededor de un texto**.

Un elemento flotante consiste en una caja que se encuentra alineada a la derecha o a la izquierda respecto a la línea en que se encuentre definida. Esta característica permite que el contenido que rodea a dicho elemento flotante discurra por el costado que queda libre, es decir, aquel que no esté alineado a un lateral del bloque de contención. Así, por ejemplo, la caja de un elemento que flota a la izquierda tiene su lado izquierdo junto al borde de contención, y el lado por el que fluye el contenido es el derecho.



Los valores que permite son:

* **Left**

El elemento va a generar una caja flotante que se encontrará alineada a la izquierda, de forma que su límite externo izquierdo tocará el borde izquierdo del bloque de contención, y el flujo normal discurrirá por el lado derecho de la caja flotante.

* **Right**

La caja generada se encontrará alineada a la derecha, de manera que el límite exterior derecho toca el borde derecho del ancho de contención, y el flujo normal fluye por el lado izquierdo de la caja.

* **None**

El elemento no genera una caja flotante, es decir, se incluye dentro del flujo normal, como un elemento cualquiera.

**Ejemplo: dar imágenes**

/\* Con esta regla definimos un bloque de texto que flotará

a la derecha dentro del bloque principal. Este bloque va

a contener datos personales de la ficha del actor. \*/

div#peliculas { width: 500px; font: 10pt Arial;

text-align: justify; padding: 5px; border: navy 1px solid;

background: paleturquoise; }

/\* Con esta otra regla definimos las características del

bloque principal de la ficha. Contiene una introducción y

un listado de algunas de las películas protagonizadas por

el actor. \*/

**div#peliculas img.adulto { float: left; padding: 2px 5px; }**

/\* Esta regla se refiere a la fotografía que alinearemos a

la izquierda del texto introductorio. \*/

</style>

...

<div id="peliculas">

**<div id="datos">**

**<img src="images/bw-joven.jpg" class="joven">**

**<b>Nombre:</b> Bruce Willis<br>**

**<b>Fecha de nacimiento:</b> 19 de Marzo de 1955<br>**

**<b>Nacido en:</b> Idar-Oberstein, Alemania<br>**

**<b>Altura:</b> 6.1 ft**

**</div>**

**<img src="images/bw-adulto.jpg" class="adulto">**

Ha demostrado una versatilidad increíble en una carrera que

presenta caracterizaciones tan diversas como el boxeador

profesional de Pulp Fiction (1994); el heroico viajero del

tiempo en Doce Monos (1995); el psicólogo infantil compasivo

en El Sexto Sentido (1999), y el significativo papel del

detective John McClane en la trilogía de La Jungla de

Cristal, personaje que tiene previsto recuperar con el

rodaje de la cuarta entrega.

<ul>

<li>La jungla de cristal (1988)</li>

<li>La jungla 2 (1990)</li>

<li>El último boy scout (1991)</li>

<li>Hudson Hawk (1991)</li>

<li>Pulp Fiction (1994)</li>

<li>12 monos (1995)</li>

<li>Four rooms (1995)</li>

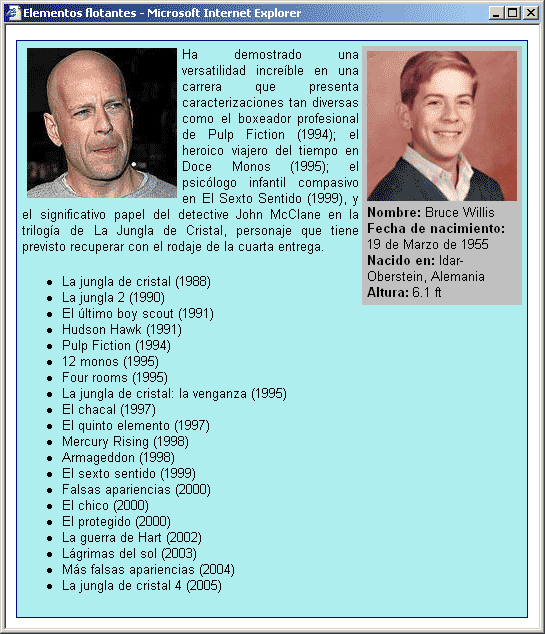
<li>La jungla de cristal: la venganza (1995)</li>

...

</ul>

</div>

Los resultados del ejemplo:



Comentarios

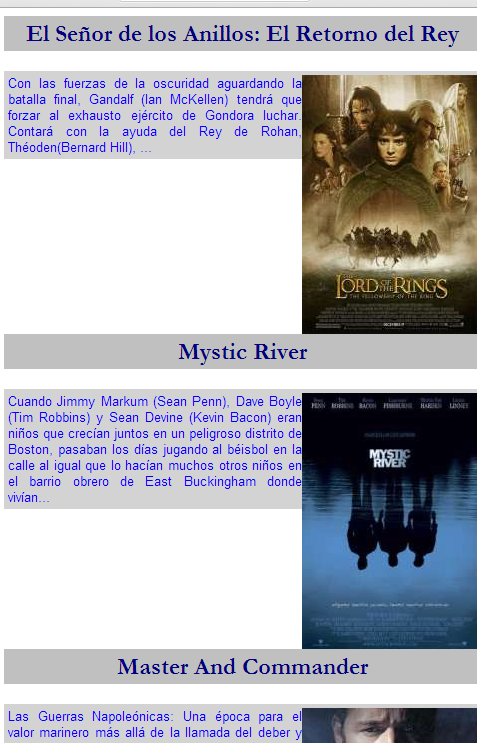
El ancho de la caja de un elemento flotante debe ser un valor explícito, ya sea mediante la propiedad **width** o por el valor intrínseco que nos indique el elemento flotante, por ejemplo el ancho de la imagen.

Los elementos alineados generan una caja de bloque, que es la caja que flota a la izquierda o a la derecha hasta que su límite exterior toca el borde lateral del bloque de contención.

REALIZAR EL EJERCICIO QUE MUESTRA LA IMAGEN.



REALIZAR EL EJERCICIO QUE MUESTRA LA IMAGEN.

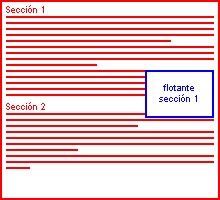
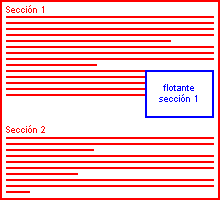


**Propiedad para deshabilitar la flotación de los elementos “clear”**

Con esta propiedad especificamos el comportamiento del flujo normal de un documento a partir de la posición de un elemento flotante.

En determinadas circunstancias puede que no nos interese que el texto fluya junto a un elemento flotante. Para ello CSS dispone de la propiedad **clear**, mediante la cual podremos deshabilitar el efecto de la propiedad **float**, de forma que el flujo no discurrirá por el costado no alineado de la caja flotante.

Un ejemplo de esas situaciones puede consistir en un documento dividido en varias secciones, dentro de cada cual podremos habilitar cajas flotantes, por ejemplo imágenes. Puede darse el caso de que una imagen tenga la suficiente altura para flotar sobre la siguiente sección. Aplicando esta propiedad al inicio de la nueva sección el elemento dejará de flotar, y la nueva sección dispondrá de todo el espacio horizontal del bloque de contención para presentar su contenido.

El funcionamiento de esta propiedad consiste en incrementar el margen superior del elemento hasta que su parte superior se encuentra por debajo de la caja flotante. Por tanto, el valor de anchura que apliquemos al margen superior del elemento será ignorado.

Los valores que permite son:

* **Left**

Anula el efecto de los flotantes a la izquierda.

* **Right**

Anula el efecto de los flotantes a la derecha.

* **Both**

Anula tanto el efecto de los flotantes a la izquierda como los flotantes a la derecha.

* **None**

Indica que no se ningún efecto de elementos flotantes.

Es el valor por defecto.

La propiedad clear permite evitar el comportamiento de posicionamiento flotante

EJEMPLO



<html>

<head><title>posicionamiento flotante</title></head>

<style>

#p{border:2px; solid; width :400px; font-size:large;}

img{clear:left;}

p{margin:5px;}

</style>

<body>

<h1> PROPIEDAD CLEAR </h1>

<div id "p">

<img src="bart.jpg">

<p >La propiedad clear evita posicionamiento flotante y por tanto,

el texto se coloca inmediatamente debajo de la imagen, en lugar de rodearla

</p>

</div>

</body>

</html>

|  |  |
| --- | --- |
| - Disposición de 2 columnas. |  |

Empezaremos a ver como componer una página sin utilizar las tablas (un medio muy utilizado hasta hace muy poco, ya que es ampliamente soportado por navegadores antiguos).  
Una solución para crear una página con dos columnas es utilizar el posicionamiento absoluto:

<html>

<head>

<title>Problema</title>

<link rel="StyleSheet" href="estilos.css" type="text/css">

</head>

<body>

<div id="columna1">

Este es el contenido de la columna 1.

</div>

<div id="columna2">

Este es el contenido de la columna 2.

</div>

</body>

</html>

La hoja de estilo para esta página es: estilos.css

\* {

margin:0px;

padding:0px;

}

#columna1 {

position:absolute;

top:0px;

left:0px;

width:200px;

margin-top:10px;

background-color:#ffff55;

}

#columna2 {

margin-left:220px;

margin-right:20px;

margin-top:10px;

background-color:#ffffbb;

}

La primera regla el selector universal, afecta a todas las marcas HTML, es sacar el margen y padding (generalmente dispondremos esta regla):

\* {

margin:0px;

padding:0px;

}

Ahora la regla definida para la primera columna es:

#columna1 {

position:absolute;

top:0px;

left:0px;

width:200px;

margin-top:10px;

background-color:#ffff55;

}

Inicializamos la propiedad position con el valor absolute, con lo que debemos indicar la posición del div en la página por medio de las propiedades top y left, en esta caso lo posicionamos en la columna 0 y fila 0 y además inicializamos la propiedad width, con lo cual le estamos indicando que esta columna tendrá siempre 200 píxeles de ancho.  
Además inicializamos la propiedad margin-top con 10 píxeles, recordemos que todos los elementos tienen margin y padding cero.  
Ahora veamos cómo inicializamos la segunda columna:

#columna2 {

margin-left:220px;

margin-right:20px;

margin-top:10px;

background-color:#ffffbb;

}

Esta regla no inicializa la propiedad position, por lo que el div ocupa la posición que le corresponde por defecto, es decir, empieza en la coordenada 0,0 de la página. El truco está en inicializar la propiedad margin-left con un valor mayor a 200, que es el ancho de la columna1.  
El resto de propiedades que inicializamos son el margin-top, para que sea igual que la primera columna y el margin-right, para que no quede el texto pegado a la derecha.

|  |  |
| --- | --- |
| - Disposición de 2 columnas (propiedad float) |  |

Una segunda forma de implementar una página con dos columnas es utilizar la propiedad float. Disponemos dos div. Al primero lo flotamos hacia la izquierda con un width fijo y el segundo se acomoda inmediatamente a la derecha.

<html>

<head>

<title>Problema</title>

<link rel="StyleSheet" href="estilos.css" type="text/css">

</head>

<body>

<div id="columna1">

Aquí el contenido de la columna 1.

</div>

<div id="columna2">

Aquí el contenido de la columna 2.

</div>

</body>

</html>

La hoja de estilo:

\* {

margin:0;

padding:0;

}

#columna1 {

float:left;

width:200px;

background-color:#ff5;

border:1px solid #555;

}

#columna2 {

margin-left:210px;

background-color:#ffb;

border:1px solid #555;

}

La columna1 tiene un ancho de 200 pixeles. Luego para evitar que la columna dos envuelva a la columna uno en caso de ser más larga inicializamos margin-left con 210 pixeles.

|  |  |
| --- | --- |
| - Disposición de 2 columnas, cabecera y pie. |  |

Una estructura muy común en la web es la disposición de una cabecera de página, seguida de dos columnas y un pie de página. La implementación de esta estructura de página es la siguiente:

<html>

<head>

<title>Problema</title>

<link rel="StyleSheet" href="estilos.css" type="text/css">

</head>

<body>

<div id="contenedor">

<div id="cabecera">

<h1>Aca el título de la página</h1>

</div>

<div id="columna1">

<p>columna1. columna1. columna1.</p>

</div>

<div id="columna2">

<h2>Título de la columna</h2>

<p>Contenido de la columna2. </p>

</div>

<div id="pie">

Pié de página.

</div>

</div>

</body>

</html>

La hoja de estilo definida para esta página es la siguiente:

\* {

margin:0px;

padding:0px;

}

#contenedor

{

width:100%;

margin:0px;

border:1px solid #000;

line-height:130%;

background-color:#f2f2f2;

}

#cabecera

{

padding:10px;

color:#fff;

background-color:#becdfe;

clear:left;

}

#columna1

{

float:left;

width:200px;

margin:0;

padding:1em;

}

#columna2

{

margin-left:210px;

border-left:1px solid #aaa;

padding:1em;

}

#pie {

padding:10px;

color:#fff;

background-color:#becdfe;

clear:left;

}

Hay algunas propiedades claves que debemos analizar en la regla #contenedor:

width:100%;

Con esto estamos indicando que siempre ocupe todo el espacio en ancho del navegador, indistintamente de la resolución de pantalla o el tamaño de ventana del navegador.  
Luego, tanto para la cabecera como para el pie, tenemos: **clear:left;**

La propiedad clear hace que un elemento no tenga elementos flotantes a su lado. Eso es lo que queremos para la cabecera y el pie.

|  |  |
| --- | --- |
| - Disposición de 3 columnas, cabecera y pie. |  |

Una modificación al concepto anterior nos permite agregar una tercera columna flotando a derecha, lo único que hay que tener cuidado es que dentro del HTML debemos disponer los div de la columna 1 y 3 en primer lugar, ya que son los que se flotan, y por último, la columna 2, que es la central:

<html>

<head>

<title>Problema</title>

<link rel="StyleSheet" href="estilos.css" type="text/css">

</head>

<body>

<div id="contenedor">

<div id="cabecera">

<h1>Aca el título de la página</h1>

</div>

<div id="columna1">

<p>columna1. </p>

</div>

<div id="columna3">

<p>columna3.</p>

</div>

<div id="columna2">

<h2>Título de la columna</h2>

<p>Contenido de la columna2.</p>

</div>

<div id="pie">

Pié de página.

</div>

</div>

</body>

</html>

La hoja de estilo es:

\* {

margin:0px;

padding:0px;

}

#contenedor

{

width:100%;

margin:0px;

border:1px solid #000;

line-height:130%;

background-color:#f2f2f2;

}

#cabecera

{

padding:10px;

color:#fff;

background-color:#becdfe;

clear:left;

}

#columna1

{

float:left;

width:200px;

margin:0;

padding:1em;

}

#columna2

{

margin-left:210px;

margin-right:230px;

border-left:1px solid #aaa;

border-right:1px solid #aaa;

padding:1em;

}

#columna3

{

float:right;

width:200px;

margin:0;

padding:1em;

}

#pie {

padding:10px;

color:#fff;

background-color:#becdfe;

clear:left;

}

# 2.7 Visualización

# Las propiedades que controlan la visualización de las cajas son: display, visibility, z-index y overflow.

# Propiedad display

Con esta propiedad especificaremos del **modo en que van a visualizarse los elementos**.

Podremos especificar la visualización de un elemento, de forma que podrá ser un elemento a nivel de bloque, a nivel de línea, a nivel de lista, a nivel de tabla o un elemento que no se permita su visualización. Veamos algunos ejemplos:

* **Vista de elementos de bloque**



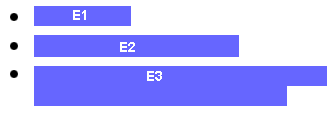
Los elementos van a formar un bloque individual para cada uno, de forma que presentan un salto de línea después de cada elemento.

* **Vista de elementos de línea**



Los elementos van a aparecer uno a continuación de otro, hasta completar el ancho de la caja de línea y saltar a la siguiente cuando el contenido rebase dicha anchura.

* **Vista de elementos en lista**



Cada elemento va a formar parte de una lista. Individualmente son bloques, pero en conjunto conforman una lista que tiene cada elemento precedido de un símbolo marcador.

**Los valores que permite son:**

* **inline**

Mediante este valor el elemento va a generar las cajas en línea que sean necesarias para su visualización.

**inline**: el elemento se muestra en el mismo bloque o párrafo que el elemento anterior, siempre que el elemento anterior se le haya aplicado también el valor inline a la propiedad display. Su caja se convierte en un elemento en línea y por tanto sólo ocupa el espacio necesario para mostrar sus contenidos.

* **block**

Mediante este valor el elemento va a generar una caja de bloque principal, dentro de la cual se encontrarán aquellos elementos que se encuentren en su ámbito de actuación según la estructura del documento.

* **inline-block**

Con este valor el elemento genera una caja de bloque, pero fluye con si se tratase de una caja de línea, es decir, su comportamiento es similar al de un elemento reemplazado.

* **list-item**

Al utilizar este valor el elemento va a generar una caja de bloque principal y una caja de línea de elemento de lista.

* **none**

Este valor nos sirve para especificar que el elemento no será visible, ni genera la caja que lo contendría. Del mismo modo sus elementos descendientes en la estructura del documento tampoco generarán caja alguna.

No sólo se vuelve invisible sino que también se omite el hueco que debería quedar, puesto que no existe ninguna caja.

* **table, inline-table, table-row-group, table-header-group, table-footer-group, table-row, table-column-group, table-column, table-cell, table-caption**

Estos valores son tratamientos específicos para tablas, filas, columnas, celdas y grupos de celdas.

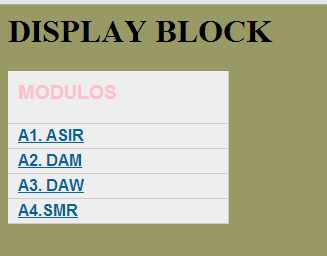
Ejemplo:

|  |
| --- |
| <body>  <p>Por las mañanas <img src="images/sol.jpg"> me levanto de  muy mal humor <img src="images/humor.jpg">.</p>  <p>Pero después de una ducha y un café  <img src="images/cafe.jpg"> me encuentro mucho mejor.</p>  <p>Por la noche me voy con mis compañeros a tomar una cerveza  <img src="images/cerveza.jpg"> y ver el partido de fútbol  <img src="images/balon.jpg"> de mi equipo favorito.</p>  </body> |

Vamos a ir mostrando como quedaría visualmente la página modificando los valores que puede tomar la propiedad “display”, asociandola a los elmentos <p> e <img>.

|  |  |
| --- | --- |
| sin display |  |
| p {display:inline} |  |
| img {display:block} |  |
| p {display:inline}  img {display:block} |  |
| img {display:none} |  |
| p {display:list-item} |  |

REALIZAR EL EJERCICIO QUE MUESTRA LA IMAGEN.



<html>

<head><title>display block</title></head>

<style>

body{background-color:#996;}

a:link{color: #006699; padding:2px 0px 2px 10px; border-top: 1px solid #cccccc; display:block;}

a:hover{color: #996699; padding:2px 0px 2px 10px; border-top: 1px solid #cccccc;bacground-color: #dddddd;}

h3{color:pink; padding:10px 0px 0px 10px;}

#pep{ width :220px;height:auto; margin-top:10px;padding:0px; background-color:#eeeeee;

font-family:arial;border-bottom:1px solid #ccc; border-right:1px solid #ccc;

font-weight:bold;}

</style>

<body>

<h1> DISPLAY BLOCK </h1>

<div id ="pep">

<h3>MODULOS</H3>

<a href="asir.htm"> A1. ASIR </a>

<a href="dam.htm"> A2. DAM </a>

<a href="daw.htm"> A3. DAW </a>

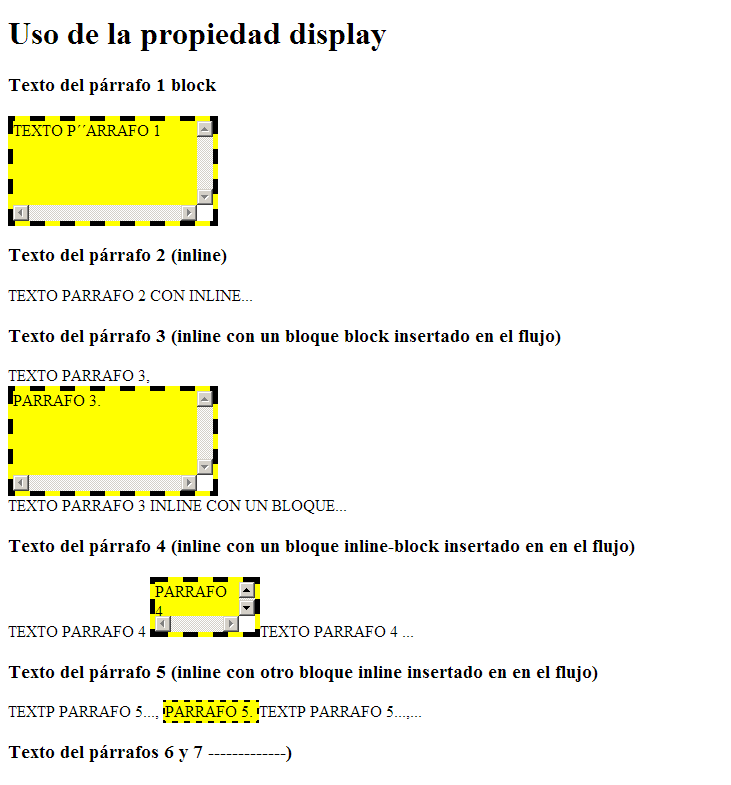
<a href="smr.htm"> A4.SMR </a>

</div>

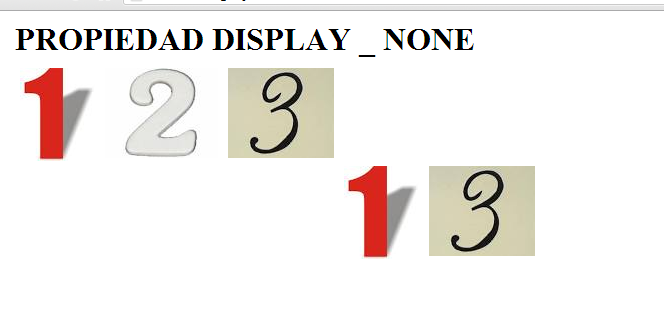
</body>

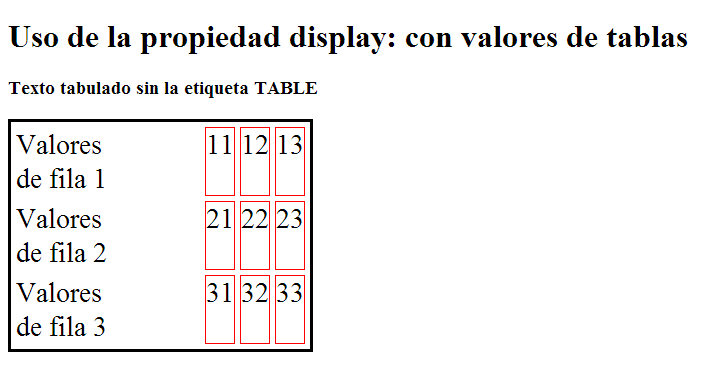
</html>

REALIZAR EL EJERCICIO QUE MUESTRA LA IMAGEN.



**EJERCICIO.**





<!DOCTYPE html>

<html>

<!-- archivo Parte 2/code/cap1013.html -->

<head>

<title>Prácticas con la propiedad display y las tablas</title>

<style>

div {font-size: 1.2em;}

#tabla {

display: table;

border-spacing: 5px;

border: 3px black solid;

}

#fila {

display: table-row;

}

#celda, celdacol1 {

display: table-cell;

border: 1px red solid;

}

#celdacol1 {

width: 50%;

}

</style>

</head>

<h1>Uso de la propiedad display: con valores de tablas</h1>

<body>

<h3>Texto tabulado sin la etiqueta TABLE</h3>

<div id="tabla">

<div id="fila">

<div id="celdacol1">Valores de fila 1</div>

<div id="celda">11</div>

<div id="celda">12</div>

<div id="celda">13</div>

</div>

<div id="fila">

<div id="celdacol1">Valores de fila 2</div>

<div id="celda">21</div>

<div id="celda">22</div>

<div id="celda">23</div>

</div>

<div id="fila">

<div id="celdacol1">Valores de fila 3</div>

<div id="celda">31</div>

<div id="celda">32</div>

<div id="celda">33</div>

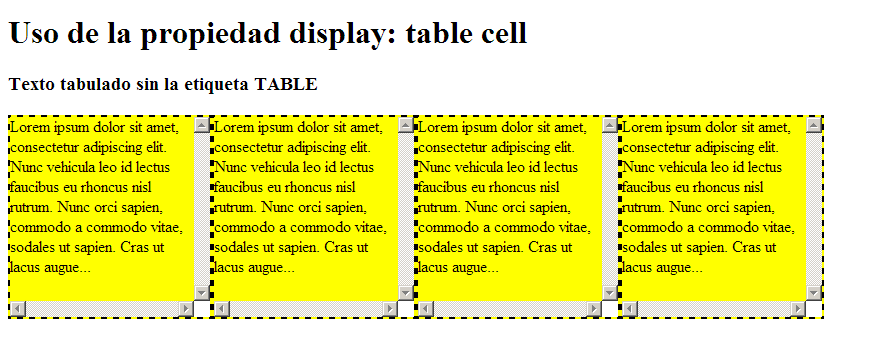
</div>

</div>

</body>

</html>

REALIZAR EL EJERCICIO QUE MUESTRA LA IMAGEN.



# Propiedad para indicar la visibilidad de los elementos “visibility”

Con esta propiedad Especificación de la visibilidad de las cajas generadas por los elementos.

Esta propiedad nos va a servir para indicar si la caja del elemento debe ser representada en nuestra aplicación de usuario. En combinación con lenguajes de script, por ejemplo Javascript, se podrán crear efectos dinámicos.

Los valores que permite son:

* **Visible**

La caja generada por el elemento es visible, así como su contenido.

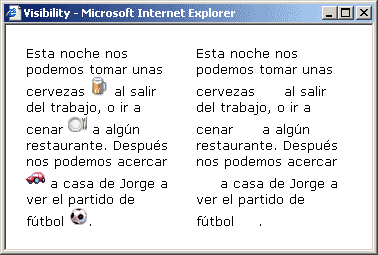
Es el valor por defecto.

* **Hidden**

La caja se vuelve totalmente transparente, pero se genera un hueco en el lugar donde debería estar.

Ejemplo

|  |  |
| --- | --- |
| <style type="text/css">  img.invisible { visibility: hidden; }  /\* Las imágenes de clase "invisible" no aparecerán, pero el contenido se representará como si estuvieran presentes, es decir, ocupan posición. \*/  </style> | <div>  Esta noche nos podemos tomar unas cervezas  <img src="images/cerveza.gif">  al salir del trabajo, o ir a cenar  <img src="images/cena2.gif">  a algún restaurante. Después nos podemos acercar  <img src="images/coche.gif">  a casa de Jorge a ver el partido de fútbol  <img src="images/futbol.gif">.  </div>  <div>  Esta noche nos podemos tomar unas cervezas  <img src="images/cerveza.gif" class="invisible">  al salir del trabajo, o ir a cenar  <img src="images/cena2.gif" class="invisible">  a algún restaurante. Después nos podemos acercar  <img src="images/coche.gif" class="invisible">  a casa de Jorge a ver el partido de fútbol  <img src="images/futbol.gif" class="invisible">.  </div> |



Podemos apreciar en la figura que el texto de la derecha es idéntico al del lado izquierdo, incluso en las alturas de las cajas de línea que producen los tamaños verticales de las imágenes, por tanto, aunque sean invisibles, ocupan posición dentro del contenido.

PROBAD EJEMPLO CON EL EJERCICIO ANTERIOR APLICANDO ESTA PROPIEDAD

# Propiedad “overflow”

Le indica al navegador lo que tiene que hacer cuando un elemento no cabe completamente dentro de la caja o elemento que lo contiene. Los valores que permite:

**Auto:** su valor por defecto, dejando la gestión al navegador.

**Hidden:** corta y oculta lo que no cabe.

**Scroll:** coloca unas barras de desplazamiento para ver todo el contenido que no cabe en la caja.

**Visible:** lo que no cabe desborda a un contenedor y se muestra en la página.

Existen las propiedades overflow-x y overflow-y por si queremos controlar individualmente las dos posibles barras de scroll.

# Propiedad “clip”

Nos permite fijar la parte visible de un elemento definiendo una zona rectangular y sólo es aplicable en elementos cuya posición se ha fijado como absolute. La sintaxis es:

Clip: rect(0 200px 100px 0);

El ejemplo anterior se podría describir como un rectángulo que, respecto al elemento en el que se aplique, empieza en 0px de la parte superior, llega hasta los 200px hacia la derecha, 100px hacia abajo y 0px desde la izquierda( cuatro coordenadas en el sentido de la agujas del reloj , empezando desde arriba)

**EJEMPLO**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

**<html xml:lang="es" xmlns="http://www.w3.org/1999/xhtml">**

**<head>**

**<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />**

**<title>.:: Introducción al modelo de cajas en CSS ::.</title>**

**<link rel="stylesheet" type="text/css" href="estilos.css" media="all" />**

**</head>**

**<body>**

**<h1>Diferentes ejemplos de la propiedad overflow y clip</h1>**

**<div class="caja1">**

**<img src="http://cort.as/1MA4" alt="Green Lantern" />**

**</div>**

**<div class="caja2">**

**<img src="http://cort.as/1MA4" alt="Green Lantern" />**

**</div>**

**<div class="caja3">**

**<img src="http://cort.as/1MA4" alt="Green Lantern" />**

**</div>**

**<div class="caja4">**

**<img src="http://cort.as/1MA4" alt="Green Lantern" />**

**</div>**

**</body>**

**</html>**

# ESTILO.CSS

# div { width: auto;

# height: 200px;

# margin: 20px;

# padding: 10px;

# border: 1px dashed black;

# float: left;

# }

# /\* Jugando con la propiedad overflow \*/

# div.caja1 {

# overflow: visible;

# 

# }

# 

# div.caja2 {

# overflow: hidden;

# }

# 

# div.caja3 {

# overflow: scroll;

# }

# 

# /\* Jugando con la propiedad clip \*/

# div.caja4 {

# position: absolute;

# top: 10;

# left: 750px;

# clip: rect(0 150px 150px 0);

# }

# Propiedad Z-INDEX

Con esta propiedad especificamos la **profundidad de un elemento posicionado**.

Hemos visto que los elementos pueden situarse en cualquier posición dentro de un bloque de contención. Por tanto, cabe esperar que dos elementos posicionados puedan superponerse uno sobre otro, de forma que uno de ellos quede parcial o totalmente oculto por aquel que se encuentre en un plano visual superior. El orden en que aparece uno sobre otro vendrá determinado por el orden en que son definidos en el documento.

|  |  |
| --- | --- |
| <div style="position: absolute; top: 20px;  left: 30px; height: 100px; width: 100px;  background: red;">  </div>  <div style="position: absolute; top: 30px;  left: 70px; height: 80px; width: 80px;  background: blue;">  </div>  <div style="position: absolute; top: 40px;  left: 110px; height: 60px; width: 60px;  background: limegreen;">  </div>  <div style="position: absolute; top:50px;  left: 140px; height: 40px; width: 40px;  background:violet;">  </div> | Descripción: Profundidad de elementos |

En este ejemplo podemos ver cuatro elementos **DIV**, con distintos colores de fondo que se van superponiendo según el orden de definición, de manera que el que aparece en primer lugar estará por debajo del resto, y así sucesivamente.

La propiedad **z-index** nos permitirá situar un elemento posicionado en un plano superior o inferior a otros, sin importar el orden en que aparezcan en el código original.

Los valores que permite son:

* **Auto**

Se establece el mismo valor que para el elemento padre.

Es el valor por defecto.

* **Valor entero**

Mediante un valor entero, que puede ser negativo, se determina el nivel de pila que ocupará el elemento posicionado, para el cuál se establece que los mayores valores estarán por encima de los de menor valor.

Ejemplo

En el siguiente ejemplo vamos a manejar cuatro elementos distintos, todos ellos posicionados, y vamos a mostrar distintos órdenes de nivel de pila para así poder apreciar cómo se disponen unos sobre otros, en función de los valores de la propiedad **z-index** que especifiquemos para cada uno.

<style type="text/css">

div { color: white; font: bold 12pt Verdana; width: 100px; height: 100px; position: absolute; border: 1px solid white; }

/\* Para todas las cajas del ejemplo definimos un tamaño (100x100px) con borde blanco de 1px y un posicionamiento absoluto, además de un color de fuente blanco en negrita con fuente Verdana de 12pt de tamaño. \*/

div#una { left: 25px; top: 25px; background: dodgerblue; **z-index: 1;** }

/\* La primera caja que está definida va a tener un color de fondo azul claro. Su posición va a ser de 25px a la izquierda y arriba. La colocaremos en el plano 1, el más bajo. \*/

div#dos { left: 50px; top: 50px; background: blue; **z-index: 2;** }

/\* Colocamos esta segunda capa un poco más abajo y a la derecha. El color de fondo será algo más oscuro, el color azul, y el nivel de pila es superior al

anterior. \*/

div#tres { left: 75px; top: 75px; background: darkblue; **z-index: 3;** }

div#cuatro { left: 100px; top: 100px; **z-index: 4;** background: midnightblue; }

/\* Estas dos capas van aumentando gradualmente las distancias y los niveles de pila, al tiempo que el color de fondo se va siendo tonalidades de azul más oscuro. \*/

</style>

...

<div id="una">capa 1</div>

<div id="dos">capa 2</div>

<div id="tres">capa 3</div>

<div id="cuatro">capa 4</div>

Esta es la situación inicial en la cual hemos asignado a cada elemento **div** el valor que aparece en la tabla y podemos apreciar el resultado que obtendremos:

|  |  |  |
| --- | --- | --- |
| Nombre capa: | Valor **z-index**: | Descripción: Paso 1 |
| Una | 1 |
| Dos | 2 |
| Tres | 3 |
| Cuatro | 4 |

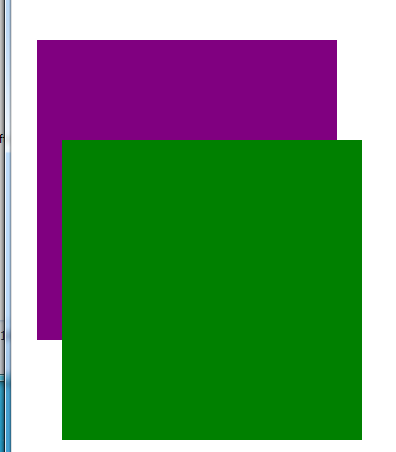
Ahora invertimos los valores, de forma que la capa 1 aparece en primer lugar, y la capa 4 al fondo, según los valores que podemos observar de la siguiente tabla:

|  |  |  |
| --- | --- | --- |
| Nombre capa: | Valor **z-index**: | Descripción: Paso 2 |
| Una | 4 |
| Dos | 3 |
| Tres | 2 |
| Cuatro | 1 |

En este último ejemplo modificaremos el orden de aparición de los elementos, según los valores especificados en la tabla, y obtendremos el resultado que vemos en la figura:

|  |  |  |
| --- | --- | --- |
| Nombre capa: | Valor **z-index**: | Descripción: Paso 3 |
| Una | 1 |
| Dos | 3 |
| Tres | 4 |
| Cuatro | 2 |

**EJERCICIO**



<html>

<head>

<title> Documento de ejemplo </title>

</head>

<body>

<div style="background-color:green; width:300; height:300 ; position:absolute; left:50; top:200; z-index:4">

</div>

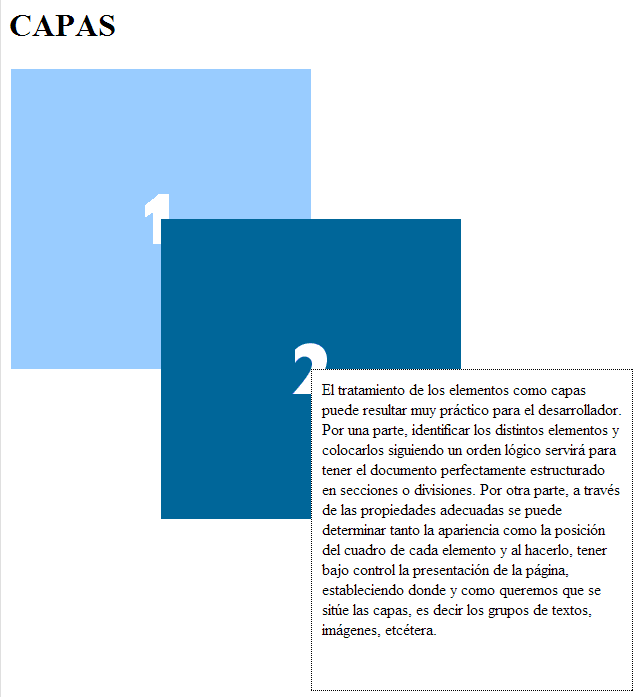
<div style="background-color:purple; width:300; height:300; position:absolute; left:25; top:100; z-index:3">

</div>

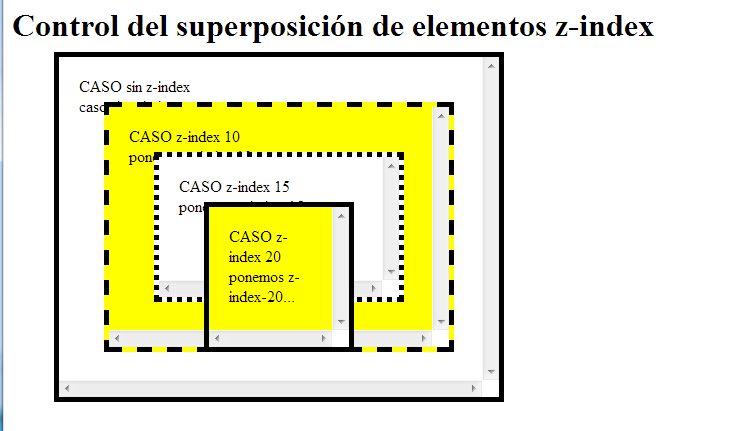
</body>

</html>

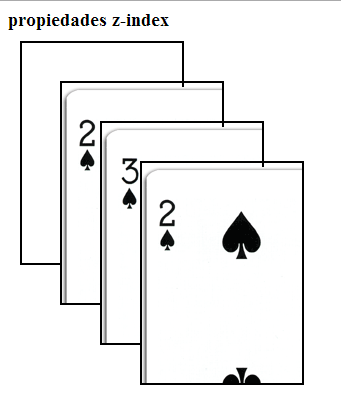
REALIZAR EL EJERCICIO QUE MUESTRA LA IMAGEN.

****

REALIZAR EL EJERCICIO QUE MUESTRA LA IMAGEN.



**EJEMPLO Buscar las imágenes, de forma que muetre la siguiente imagen, de forma que si se pasa el ratón por encima de una carta pase a ocupar el primer lugar. Inicialmente las cartas de la baraja tienen un z-index:1,2,3,4. Al pasar por encima el ratón asignamos a z-index:10, por tanto la carta seleccionada pasa a primer lugar.**

****

**2.8. Reglas de estilo aplicado a Tablas**

**Propiedades CSS en tablas HTML**

Aunque la tendencia es el uso de CSS, aún existe mucho código de presentación en tablas. A continuación se muestra la equivalencia entre ambos enfoques:

| **ASPECTO** | **ATRIBUTO HTML** | **PROPIEDAD CSS** |
| --- | --- | --- |
| Alineación | align | text-align |
| valign | vertical-align |
| Ancho | width | width y table-layout |
| Bordes | border | border |
| Espaciado en celdas | cellpadding | padding |
| Espaciado entre celdas | cellspacing | border-spacing |

Para definir los bordes de una tabla y sus elementos dependientes se pueden utilizarlas propiedades border. Por ejemplo:

table {border: 3px solid black;}

La propiedad border-spacing permite especificar la distancia entre dos celdas consecutivas, especificando el espaciado horizontal y el vertical. Por ejemplo:

table {border-spacing: 3px 3px;}

La propiedad padding permite ajustar el espaciado entre el contenido y el borde.

El modelo collapse fusiona de forma automática los bordes de las celdas adyacentes, mientras que el modelo separate fuerza a que cada celda muestre sus cuatro bordes. Por defecto, los navegadores utilizan el modelo separate. Por ejemplo:

table, table \* {border-collapse: collapse;}

La propiedad empty-cells permite controlar el comportamiento de las celdas vacías. Admite los valores:

* show.- muestra la celda aunque esté vacía.
* hide.- oculta la celda si está vacía.

La propiedad table-layout determina cómo hace el agente de usuario para dimensionar la tabla. Puede tener los valores:

* auto.- valor predeterminado. El ancho de la columna se ajusta al necesario para mostrar el elemento más ancho (teniendo en cuenta si se puede separar en varias líneas).
* fixed.- fijo. El diseño horizontal solo depende del ancho de la tabla y del ancho de las columnas, no del contenido de las celdas.
* inherit.- aceptado solo en algunos navegadores. El valor de la propiedad se hereda del elemento padre.

Las tablas pueden tener títulos definidos por la etiqueta< caption> y CSS tiene la capacidad para establecer la alineación y el posicionamiento de la leyenda del título mediante la propiedad caption-side, que tiene como valores posibles: top (arriba), bottom (abajo) o inherit (heredado). La alineación de la etiqueta caption se controla mediante la propiedad text-align o vertical-align

EJEMPLO

<html>

<head>

<title>Problema</title>

<link rel="StyleSheet" href="estilos.css" type="text/css">

</head>

<body>

<table>

<caption>

cantidad de lluvia caida en mm.

</caption>

<thead>

<tr>

<th>Provincia</th><th>Enero</th><th>Febrero</th><th>Marzo</th>

</tr>

</thead>

<tbody>

<tr>

<th>Córdoba</th>

<td>210</td><td>170</td><td>120</td>

</tr>

<tr>

<th>Buenos Aires</th>

<td>250</td><td>190</td><td>140</td>

</tr>

<tr>

<th>Santa Fe</th>

<td>175</td><td>140</td><td>120</td>

</tr>

</tbody>

</table>

</body>

</html>

La hoja de estilo definida a esta tabla es: estilos.css

caption

{

font-family:arial;

font-size:15px;

text-align: center;

margin: 0px;

font-weight: bold;

padding:10px;

}

table

{

border-collapse: collapse;

}

th

{

border-right: 1px solid #fff;

border-bottom: 1px solid #fff;

padding: 0.5em;

background-color:#6495ed;;

}

thead th

{

background-color: #6495ed;

color: #fff;

}

tbody th

{

font-family:arial;

font-weight: normal;

background-color: #6495ed;

color:#ff0;

}

td {

border: 1px solid #000;

padding: .5em;

background-color:#ed8f63;

width:100px;

text-align:center;

}

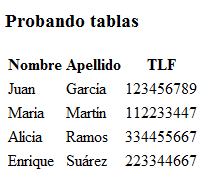
La marca caption dentro de una tabla es el título que debe aparecer arriba.  
**La propiedad border-collapse puede tomar dos valores: collapse o separate. Separate deja las celdas con unos pixeles de separación, no así collapse**.  
El resto es la definición de una serie de reglas para las marcas th, th dentro de la marca tbody, th dentro de la marca thead y por último td.

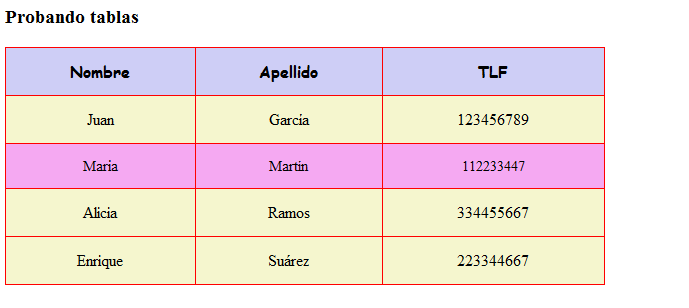
Realizar el siguiente ejercicio

Dada la tabla “tabla.html”, aplica estilos para obtener el aspecto mostrado en la imagen adjunta.

Estilos:

1. Para la tabla borde de una línea u ancho de 600 px
2. Para las etiquetas th y td a la vez sólido de color rojo. Padding arriba/debajo de 0, y derecha e izquierda de 0.3em. El texto alineado al centro y la altura de las celdas de 3em.
3. Sólo para th: relleno de color #CECEF6 y tipo de letra "comic sans ms"
4. Para las filas tr: color de fondo #F5F6CE
5. Y finalmente al pasar el ratón por las filas, el fondo cambie a color #F5A9F2 y el tamaño de fuente tenga 15 píxeles.

****

****

REALIZAR EL EJERCICIO QUE MUESTRA LA IMAGEN.

## A partir del fichero tablas.html debes conseguir el aspecto mostrado en la imagen con el navegador Firefox.

## Cuando pasamos con el ratón por las celdas de filas de puestos impares el resultado es como este:

## 

## Cuando pasamos con el ratón por las celdas de filas de puestos pares el resultado es como este:

## 

## Utiliza las imágenes proporcionadas como recursos.

**2.9 Reglas de estilo aplicado a Menús**

|  |  |
| --- | --- |
| Creación de un menú vertical configurando las pseudo-clases. |  |

Un recurso muy útil es disponer un menú en una página, si no requerimos uno muy elaborado podemos resolverlo utilizando sólo CSS y HTML (en otros casos se requiere además de JavaScript).  
Vamos a implementar uno muy sencillo que requiere agrupar en un div una serie de párrafos que contienen un hipervínculo cada uno. Cuando la flecha del mouse se encuentra sobre el hipervínculo cambiamos el color del fondo y la letra del hipervínculo.  
EJEMPLO:

<html>

<head>

<title>Problema</title>

<link rel="StyleSheet" href="estilos.css" type="text/css">

</head>

<body>

<div id="menu">

<p><a href="http://www.google.com">Google</a></p>

<p><a href="http://www.yahoo.com">Yahoo</a></p>

<p><a href="http://www.msn.com">Msn</a></p>

<p><a href="http://www.altavista.com">Altavista</a></p>

</div>

</body>

</html>

La hoja de estilo asociada a esta página es: estilos.css

#menu {font-family: Arial;}

#menu p {margin:0px; padding:0px;}

#menu a {

display: block;

padding: 3px;

width: 160px;

background-color: #f7f8e8;

border-bottom: 1px solid #eeeeee;

text-align:center;

}

#menu a:link, #menu a:visited {color: #ff0000;text-decoration:none;}

#menu a:hover {

background-color: #336699;

color: #ffffff;

}

Podemos decir que definimos un estilo por medio de un Id llamado menú. Definimos una regla para este Id:

#menu {

font-family: Arial;

}

La segunda regla:

#menu p {

margin:0px;

padding:0px;

}

Estamos indicando que todos los párrafos contenidos en el estilo #menu deben tener cero en margin y padding.  
Luego las anclas definidas dentro del estilo #menu definen las siguientes características:

#menu a {

display: block;

padding: 3px;

width: 160px;

background-color: #f7f8e8;

border-bottom: 1px solid #eeeeee;

text-align:center;

}

El valor block para la propiedad display permite que el ancla ocupe todo el espacio del párrafo, indistintamente del largo del hipervínculo.  
Otra propiedad nueva es width, esta fija el tamaño máximo que puede tener el hipervínculo antes de provocar un salto de línea.  
Por último inicializamos las pseudoclases:

#menu a:link, #menu a:visited {

color: #ff0000;

text-decoration: none;

}

#menu a:hover {

background-color: #336699;

color: #ffffff;

}

Estamos definiendo el mismo color de texto para los vínculos seleccionados como aquellos que no han sido seleccionados:

#menu a:link, #menu a:visited {

color: #ff0000;

Por último cambiamos el color de fondo de la opción que tiene la flecha del mouse encima:

#menu a:hover {

background-color: #336699;}

**Práctica menús a partir de una lista**

Crear un menú a partir de una lista y se usará una imagen de fondo que se activará al pasar el ratón. Por defecto estará desactivada. La imagen cuando pasemos el ratón se llama **flecha\_activa.png** 

Y cuando no pasa **flecha\_inactiva.png** 

Se parte del siguiente código html, los enlaces son ficticios.

**<html>**

**<head>**

**<title>Ejercicio menu con imágenes</title>**

**<body>**

**<h2>Menú que contiene imágenes de fondo </h2>**

**<ul class="menu">**

**<li><a href="#" title="Enlace1">Elemento 1</a></li>**

**<li><a href="#" title="Enlace2">Elemento 2</a></li>**

**<li><a href="#" title="Enlace3”>Elemento 3</a></li>**

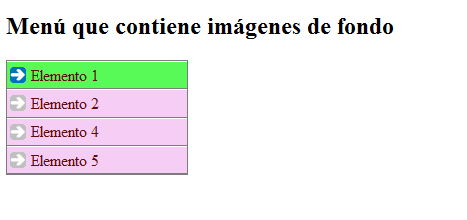
**<li><a href="#" title="Enlace4">Elemento 4</a></li>**

**<li><a href="#" title="Enlace5">Elemento 5</a></li>**

**</ul>**

**</body>**

**</html>**



SOLUCIÓN

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<title>Ejercicio menu con imágenes</title>

<style type="text/css">

.menu {

width: 180px;

list-style: none;

margin: 0;

padding: 0;

border: 1px solid #7C7C7C;}

.menu li {

border-bottom: 1px solid #7C7C7C;

border-top: 1px solid #FFFFFF;

}

.menu li a {

padding: .2em 0 .2em 1.5em;

display: block;

text-decoration: none;

color: #610B0B;

background: #F6CEF5 url("flecha\_inactiva.png") no-repeat 3px;

}

.menu li a:hover, ul.menu li a:active {

background: #58FA58 url("flecha\_activa.png") no-repeat 3px;

}

</style>

</head>

<body>

<h2>Menú que contiene imágenes de fondo </h2>

<ul class="menu">

<li><a href="#" title="Enlace1">Elemento 1</a></li>

<li><a href="#" title="Enlace2">Elemento 2</a></li>

<li><a href="#" title="Enlace3”>Elemento 3</a></li>

<li><a href="#" title="Enlace4">Elemento 4</a></li>

<li><a href="#" title="Enlace5">Elemento 5</a></li>

</ul>

</body>

</html>

|  |  |
| --- | --- |
| Creación de un menú horizontal con una lista. |  |

Otro estilo de menú muy común es donde las opciones se encuentran una al lado de otra. Veamos el código para la implementación de un menú horizontal:

<html>

<head>

<title>Problema</title>

<link rel="StyleSheet" href="estilos.css" type="text/css">

</head>

<body>

<ul id="menuhorizontal">

<li><a href="http://www.google.com">Google</a></li>

<li><a href="http://www.yahoo.com">Yahoo</a></li>

<li><a href="http://www.msn.com">Msn</a></li>

<li><a href="http://www.altavista.com">Altavista</a></li>

</ul>

</body>

</html>

El archivo de estilos es: estilos.css

#menuhorizontal {

float:left;

width:100%;

padding:0px;

margin:0px;

list-style-type:none;

}

#menuhorizontal a {

float:left;

width:100px;

text-decoration:none;

text-align:center;

color:#ff0000;

background-color:#f7f8e8;

padding:3px 5px;

border-right:1px solid white;

}

#menuhorizontal a:hover {

background-color:#336699;

}

#menuhorizontal li {

display:inline;

}

Como queremos que se localice a la izquierda inicializamos la propiedad float con el valor left, tanto para el id menuhorizontal como para la marca a que se encuentra en el id menuhorizontal. Fijamos la propiedad width del id menuhorizontal con el valor de 100% para que ocupe toda la línea. El ancho de cada opción es de 100px que lo indicamos en la propiedad width de la marca a.  
Cuando inicializamos la propiedad padding con: padding:3px 5px; estamos fijando una separación de 3 pixeles en la parte superior e inferior y 5 pixeles a izquierda y derecha.  
Separamos cada opción por una línea blanca de un pixel: border-right:1px solid white;

Otra forma de expresar los colores es utilizar una serie de nombres predefinidos (white, red, green, blue, etc.) de todos modos, lo más seguro para que entiendan todos los navegadores, es indicando los valores de rojo, verde y azul como hemos venido trabajando.  
Luego con:

**#menuhorizontal li {**

**display:inline;**

**}**

Indicamos que todos los list item (li) se muestren en la misma línea, ya que por defecto este tipo de marca indica que luego al mostrar un item de la lista el siguiente es abajo. Fácilmente esta propiedad la podemos entender si inicializamos la marca p (párrafo) con el siguiente valor:

p {

display:inline;

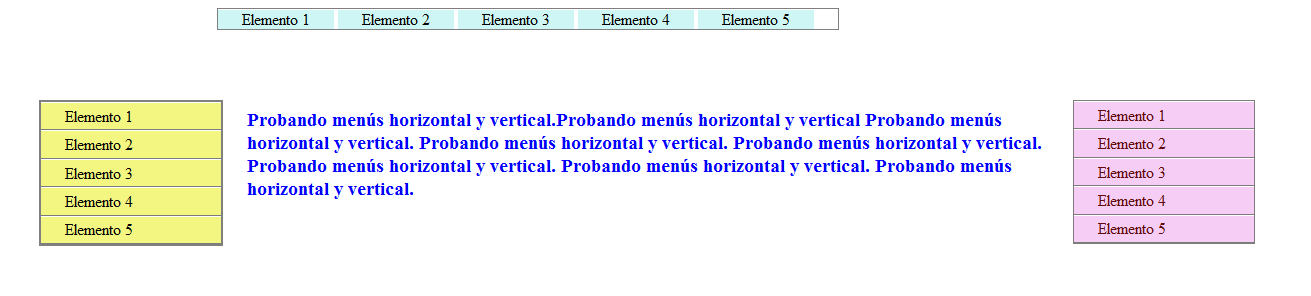
}

Luego si en la página HTML imprimimos dos párrafos, los mismos deberían aparecer en la misma línea.

**REALIZA EL SIGUIENTE EJERCICIO**

**PÁGINA CON TRES MENÚS** MENU/**MENU\_3\_MENU**

Con los menús realizados anteriormente, realiza una página con tres menús. Utiliza enlaces ficticios. La página debe tener un aspecto similar al de la imagen siguiente. Utiliza la propiedad position para colocar los menús verticales en la página. Pon margen a la izquierda y a la derecha de texto para que no sea tapado por los menús.

****

|  |  |
| --- | --- |
| 3. Aplicación de hojas de estilo a un formulario. |  |

Un formulario es el elemento esencial para el envío de datos al servidor por parte del visitante del sitio.  
Veamos un ejemplo donde implementamos un formulario y le aplicamos una serie de reglas de estilo a las diferentes marcas HTML que intervienen:

<html>

<head>

<title>Problema</title>

<link rel="StyleSheet" href="estilos.css" type="text/css">

</head>

<body>

<div id="contenedorform">

<form method="post" action="#">

<label>Introduzca nombre:</label>

<input type="text" name="nombre" size="30">

<br>

<label>Introduzca mail:</label>

<input type="text" name="mail" size="45">

<br>

<label>Comentarios:</label>

<textarea name="comentarios" cols="30" rows="5"></textarea>

<br>

<input class="botonsubmit" type="submit" value="confirmar">

</form>

</div>

</body>

</html>

La hoja de estilo que se aplica es: estilos.css

#contenedorform {

width:500px;

margin-left:20px;

margin-top:10px;

background-color:#ffe;

border:1px solid #CCC;

padding:10px 0 10px 0;

}

#contenedorform form label {

width:120px;

float:left;

font-family:verdana;

font-size:14px;

}

.botonsubmit {

color:#f00;

background-color:#bbb;

border: 1px solid #fff;

}

Podemos observar que definimos un div contenedor y dentro de este el formulario. Para que los textos aparezcan a la izquierda, definimos una serie de label que las flotamos a izquierda, por lo que los controles del formulario aparecerán a derecha todos en columnas.

**Otras propiedades**

* + **Barras de desplazamiento**: Sólo funciona con Explorer. Se aplica a <BODY> para que afecte a la página completa. Sus propiedades una a una:
    - *SCROLLBAR-FACE-COLOR: Parte que se ve de la barra. (red)*
    - *SCROLLBAR-TRACK-COLOR: Parte de abajo, donde se desliza la barra. (green)*
    - *SCROLLBAR-HIGHLIGHT-COLOR: El vivo de la barra y la flecha (blue)*
    - *SCROLLBAR-SHADOW-COLOR: Sombra abajo y derecha (aqua)*
    - *SCROLLBAR-ARROW-COLOR: Color de la fleca (olive)*
    - *SCROLLBAR-3DLIGHT-COLOR: Borde externo de barra y flechas (orange);*
    - *SCROLLBAR-DARKSHADOW-COLOR: borde de toda la barra derecha y abajo (black);*