

Lenguajes de Marcas

Contenido:

La etiqueta <div>	2
Posicionamiento de Capas	4
Position: static	4
Position: absolute	3
Position: relative	6
Position: fixed	7



Unidad 6 - CAPAS

Las capas no son más que unos recuadros, que pueden situarse en cualquier parte de la página, en los que podemos insertar contenido HTML. Dichas capas pueden ocultarse y solaparse entre sí, lo que proporciona grandes posibilidades de diseño.

1.- La etiqueta <div>

Vamos a ver algunas de las características básicas sobre las capas, para poder insertarlas en nuestras páginas.

Podemos insertar una capa a través de las etiquetas <div> y </div>, que como ya vimos, sirven para agrupar bloques de texto.

A través del atributo `id` se le da un nombre a la capa, y a través del atributo `style` se establecen el resto de propiedades de la capa.

las propiedades **left** (izquierda) y **top** (superior) establecen la posición de la capa respecto a los márgenes izquierdo y superior de la página. Pueden tomar un número como valor, acompañado de `px` cuando haga referencia a píxeles, y acompañado de `%` cuando haga referencia a un porcentaje.

Para que la capa aparezca en la posición establecida, es necesario incluir también la propiedad `position` con el valor **absolute**. Si no se estableciera este valor, la capa se mostraría pegada al margen izquierdo, en la posición en la que hubiera sido insertada dentro del código.

A través de las propiedades **width** (anchura) y **height** (altura) se establece el tamaño de la capa. Pueden tomar un número como valor, acompañado de `px` cuando haga referencia a píxeles, y acompañado de `%` cuando haga referencia a un porcentaje.

La propiedad **z-index** establece el índice de la capa dentro de la página. Una capa podrá ser solapada por aquellas capas cuyo índice sea mayor. Siempre es un valor numérico.

A través de la propiedad **visibility** puede establecerse la visibilidad de la capa. Puede tomar los valores **inherit** (se muestra la capa mientras la capa a la que pertenece también se esté mostrando), **visible** (muestra la capa, aunque la capa a la que pertenece no se esté viendo) y **hidden** (la capa está oculta).

A través de las propiedades **layer-background-image** y **background-image** se puede establecer una imagen de fondo para la capa. La ruta y el nombre de la imagen han de aparecer entre paréntesis, después de la palabra `url`.

A través de las propiedades `layer-background-color` y `back-`

`ground-color` se puede establecer un color de fondo para la capa. Ha de ser un número hexadecimal.

A través de la propiedad **overflow** puede establecerse si se mostrará o no el contenido de la capa cuando no pueda ser visualizado en su totalidad, por ser la capa demasiado pequeña. Puede tomar los valores **visible** (se muestra todo el contenido de la capa, aunque esto implique hacer que la capa sea más grande), **hidden** (no es posible visualizar el contenido de la capa que no quepa en ella), **scroll** (se muestra la barra de desplazamiento, aunque el contenido de la capa pueda ser visualizado totalmente) y **auto** (se muestra la barra de desplazamiento cuando sea necesario).

A través de la propiedad **clip** puede establecerse el área de la capa que podrá ser visualizado. Lo que hace es recortar la capa, haciendo que partes de ella no sean visibles. Ha de especificarse la distancia de los márgenes de la capa entre paréntesis, después de la palabra `url`.

El primer valor ha de ser la distancia (se asume que está en píxeles) del margen superior, hasta la que no se visualizará el contenido de la capa.

El segundo valor ha de ser la distancia del margen izquierdo, hasta la que no se visualizará el contenido de la capa.

El tercer valor ha de ser la distancia del margen superior, hasta la que se visualizará el contenido de la capa.

El cuarto valor ha de ser la distancia del margen izquierdo, hasta la que se visualizará el contenido de la capa.

También es posible incluir `auto` como valor de alguna de estas distancias, lo que indica que la distancia se corresponderá con los bordes de la capa.

Todas estas propiedades se especifican a través del atributo `style`, y deben aparecer entre comillas dobles, con un punto y coma detrás de cada una. Para asignar los valores a las propiedades no se utiliza el símbolo `=` (igual), sino que se utiliza el símbolo `:` (dos puntos).

Por ejemplo, podríamos insertar una capa escribiendo el siguiente código:

```
<div id="capa" style="position:absolute; width:200px; height:115px; z-index:3; visibility: visible; background-color: #0099CC; layer-background-color: #0099CC; background-image: url(imagenes/fondocapa.gif); layer-background-image: url(imagenes/fondocapa.gif); clip: rect(1 auto auto 3);">
```

Este texto está; dentro de una capa.

```
</div>
```

Otra forma de insertar capas es utilizando las etiquetas `` y ``, en lugar de las etiquetas `<div>` y `</div>`.

La etiqueta `` se utiliza del mismo modo que la etiqueta `<div>`, y es compatible con un mayor número de navegadores.

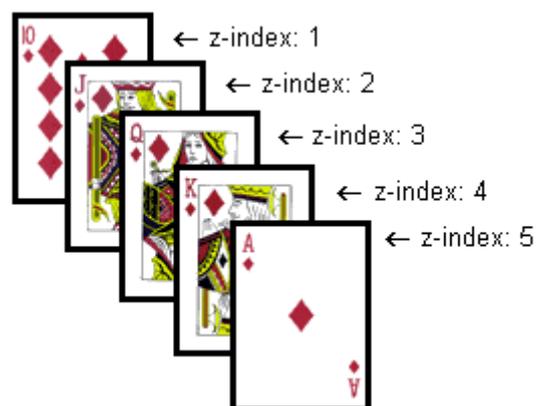
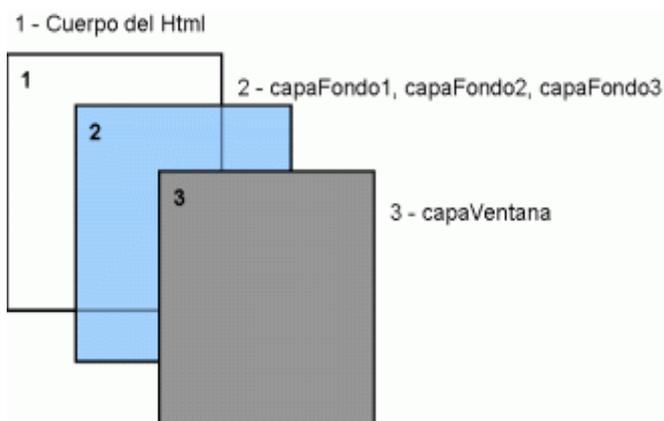
También existen las etiquetas `<layer>` y `</layer>`, pero solamente son reconocidas por el navegador Netscape. La principal diferencia de la etiqueta `<layer>`, en lo que se refiere a las otras dos, es que las propiedades de la capa se especifican como atributos independientes, y no como valores dentro del atributo `style`.

Por ejemplo, podríamos insertar una capa escribiendo el siguiente código:

```
<layer name="capa" width="200px" height="115px" z-index="3" visibility="show" bgcolor="#0099CC" background="imagenes/fondocapa.gif">
```

Este texto est´ dentro de una capa.

```
</layer>
```



La primera capa (llamamos así a los elementos DIV que tienen posicionamiento CSS), tiene como todas las del ejemplo, posicionamiento absoluto. Los atributos top: 100px y left: 30px quieren decir que se posiciona a 100 píxeles de la parte superior de la página y a 30 píxeles de la izquierda. En este caso las distancias top y left para ubicar la capa con position absolute son relativas a la esquina superior izquierda del área disponible del navegador, pues esta capa no está dentro de ninguna otra con posicionamiento distinto de static. Cabe llamar la atención en esta primera capa también sobre el atributo z-index: 2, que servirá para indicarle al navegador la posición de la capa, en la tercera dimensión, con respecto a otras que se puedan superponer, para que sepa cuál tiene que estar debajo y cuál arriba.

La segunda capa podemos ver que tiene un z-index:1. Eso quiere decir, que en caso se posicione en el mismo lugar se ocultará por la capa primera, que tiene un z-index mayor.

En la tercera capa hemos probado el posicionamiento utilizando los atributos bottom y right, así que la estamos posicionando con respecto a la esquina inferior derecha.

Veamos un segundo ejemplo donde vamos a colocar una capa con posicionamiento absoluto y dentro varias capas también posicionadas con absolute.

```
<html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="en" lang="en">
<body bgcolor="#FFFFFF" text="#000000"
link="#0000FF" vlink="#800080" alink="#FF0000">
<div style="position: absolute; top: 100px; left: 200px;
background-color: #ff9966; width: 400px; height:
100px;">
<div style="position: absolute; top: 10px; left:10px;
width: 200px; height: 30px; background-color: #00AA00;
z-index: 1"> Uno </div>
<div style="position: absolute; top: 10px; left:100px;
background-color: #AA0000; width:200px; height: 30px;
z-index: 2"> Dos </div>
<div style="position: absolute; top: 10px; left:200px;
background-color: #0000ff; width:200px; height: 30px; z
-index: 3"> Tres</div>
</div><h1>Posicionamiento CSS</h1>
</body></html>
```

Posicionamiento CSS



En este caso la primera capa, que no está dentro de ninguna otra, se posiciona con top y left con respecto a la esquina superior izquierda del espacio disponible en el navegador para el cuerpo de la página. Las capas anidadas están también con position: absolute, pero al estar dentro de otra capa que tiene posicionamiento distinto de static, sus valores top y left son relativos a la esquina superior izquierda de la capa que las contiene.

Position: relative

El valor relative en el atributo position indica que la capa sí forma parte del flujo normal de elementos de la página, por lo que su posición dependerá del lugar donde esté en el código y el flujo HTML. Además, las capas con posicionamiento relative, admiten los valores top y left para definir la distancia a la que se colocan con respecto al punto donde esté en ese momento el flujo normal del HTML. Como afectan al mencionado flujo del HTML, los elementos colocados

después de las capas relative, tendrán en cuenta sus dimensiones para continuar el flujo y saber dónde colocarse. Sin embargo, no se tendrá en cuenta los top y left configurados.

Veamos un ejemplo que quizás aclare las cosas.

```
<h1>Hola</h1>
```

```
<div style="background-color: #606; color:#ffc; padding:10px; text-align: center; width: 300px;">Hola esto es una prueba</div>
```

```
<div style="position: relative; width: 300px; padding: 10px; background-color: #066; color:#ffc; top:100px; left: 30px;">capa de posicionamiento relative<br>Se tiene en cuenta esta capa para posicionar las siguientes.</div>
```

```
<h2>hola de nuevo!</h2>
```

Hola

Hola esto es una prueba

hol
capa de posicionamiento relative
Se tiene en cuenta esta capa para posicionar las siguientes.

Las etiquetas H1 y H2 respetan el flujo HTML y también tenemos un elemento DIV donde no se ha especificado nada en position, luego es static y por tanto también es afectada por el flujo. Hay una capa relative, en el segundo elemento DIV, que también se posiciona con respecto al flujo normal. Como tiene un top y left, aparece un poco desplazada del lugar que le tocaría con respecto al flujo.

El último H2 que aparece se coloca teniendo en cuenta al flujo y tiene en cuenta la capa relative, por eso deja un espacio en blanco arriba, pero no atiende a la posición real de ésta, que se marcó con los atributos top y left.

Position: fixed

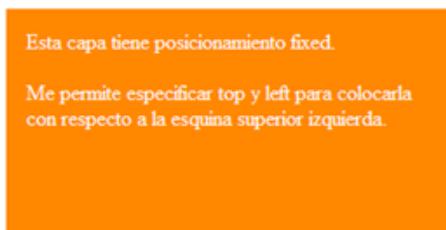
Este atributo sirve para posicionar una capa con posicionamiento absoluto, pero su posición final será siempre fija, es decir, aunque se desplace el documento con las barras de desplazamiento del navegador, siempre aparecerá en la misma posición.

El lugar donde se "anclará" la capa siempre es relativo al cuerpo (el espacio disponible del navegador para la página). Si utilizamos top y left, estaremos marcando su posición con respecto a la esquina superior izquierda y si utilizamos bottom y right su posición será relativa a la esquina inferior derecha.

Veamos un ejemplo.

```
<html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="en" lang="en">
<head></head>
<body bgcolor="#FFFFFF" text="#000000" link="#0000FF"
vlink="#800080" alink="#FF0000">
<div style="position: fixed; width: 300px; height: 140px; top:
100px; left: 30px; background-color: #ff8800; color: #fff;
padding: 15px;z-index: 1;">
```

Hola



Esta capa tiene posicionamiento fixed.

```
<br>
<br>
Me permite especificar top y left para colocarla con res-
pecto a la esquina superior izquierda.
</div>
<div style="position: fixed; width: 700px; height: 30px;
padding: 10px; background-color: #0df; top: 450px; left:
10px; z-index: 2;">Posicionamiento fixed</div>
<h1>Hola</h1>
<div style="position: fixed; width: 100px; height: 30px;
padding: 10px; background-color: #0df; bottom: 10px;
right: 10px; z-index: 4;">Posicionamiento fixed</div>
<br><br><br><br><br><br><br><br><br><br><br><br><br><br>
</body>
</html>
```

Se puede ver la página resultante con este código.



Se puede ver que hay varias capas con position: fixed y un montón de BR para que la página pueda tener un desplazamiento. Si vemos la página en marcha y hacemos scroll hacia abajo con la barra de desplazamiento, veremos que las capas fixed siempre mantienen la misma posición.

Nota: El valor fixed en el atributo position funciona en todos los navegadores, pero en el caso de Internet Explorer sólo funciona en la versión 7 y superiores. Además, para que funcione en Explorer tiene que declararse un DOCTYPE!. Servirían varios tipos de DOCTYPE!, sin embargo debería declararse con el formato completo. Algo así como:

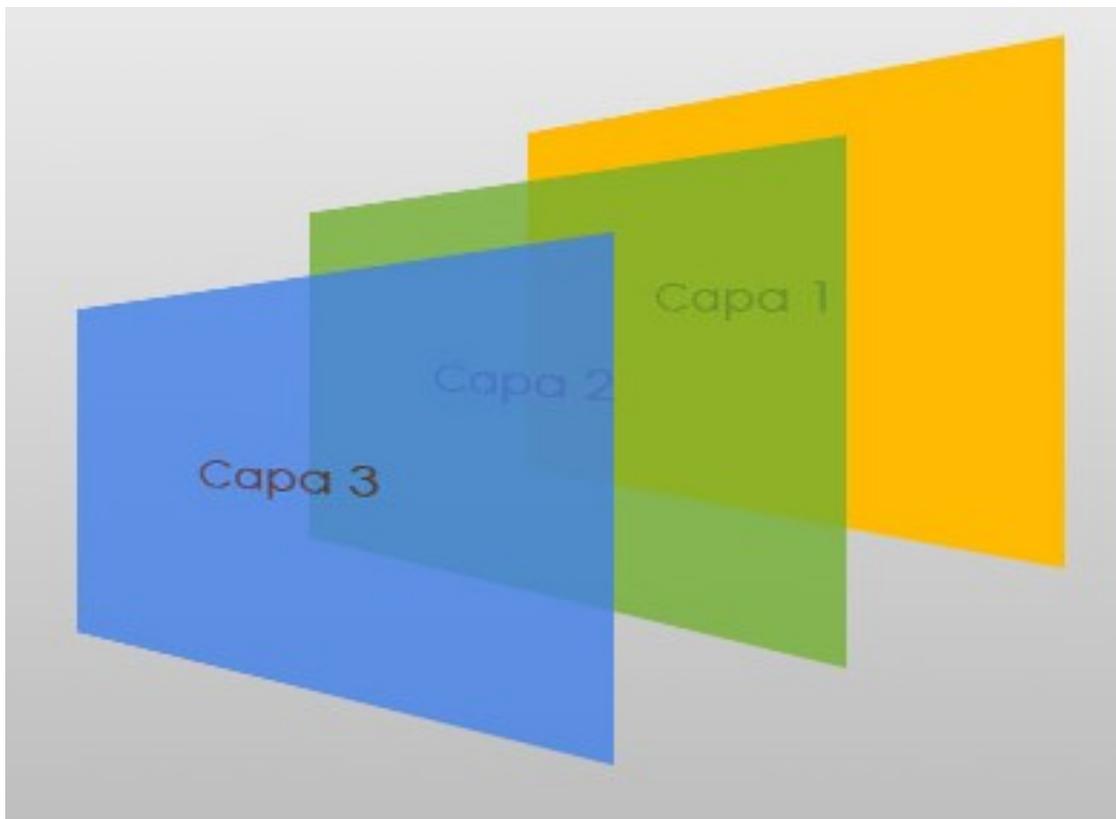
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

O por poner otro ejemplo:

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

Position: inherit

El valor inherit indica que el valor de position tiene que heredarse del elemento padre. No funciona en Explorer, al menos hasta la versión 8. Tiene en verdad poca utilidad y además, como no funciona en el navegador más utilizado en la actualidad, tiene aun menos sentido usarlo. Por ello, no ponemos ejemplos.



IES MIGUEL ROMERO ESTEO
MARTÍN CARRIÓN S/N
29006
MÁLAGA

Teléfono: 951298668
Fax: 951298670

Estos apuntes forman parte del curso “Lenguajes de Marcas y Sistemas de Gestión de la Información” que forma parte del currículum del ciclo superior de “Administración de Sistemas Informáticos en Red”.

El temario completo se encuentra en la web del instituto Miguel Romero Esteo.



LENGUAJES DE MARCAS

Estamos en
www.romeroesteo.es