

# Repaso XHTML y CSS

## Unidades de medida

No debe dejarse un espacio en **blanco** entre el valor y la unidad.

**Correcto** → `height: 768px;`

**Incorrecto** → `height: 768 px;`

Si el valor es 0, no debe indicarse ninguna unidad.

**Correcto** → `margin: 0;`

**Incorrecto** → `margin: 0px;`

Los valores **no** se colocan entre comillas... Excepto en los nombres de fuentes compuestas:

**Correcto** → `color:white; font-family:"Times New Roman", serif;`

**Incorrecto** → `color:"white";`

## Definición de Colores

- **Hexadecimal normal:** `#rrggbb`
  - Cada dígito varía entre 0 y F, el conjunto varía entre negro `#000000` y blanco `#FFFFFF`
  - **Hexadecimal abreviado:** `#rgb`
- **Decimal:** `rgb(0-255,0-255,0-255)`
- **Decimal, indicando la transparencia:** `rgba(0-255,0-255,0-255,0-1)`
- **Porcentual:** `rgb(0-100%,0-100%,0-100%)`

## COMENTARIOS

Cuando los ficheros se hacen grandes, es difícil saber exactamente a qué corresponde cada parte del código escrito. Para hacer más legibles y comprensibles los ficheros tenemos la posibilidad de escribir comentarios **explicativos** del contenido. No afectarán a la presentación de la página en el navegador web.

- En los ficheros **.html** los comentarios se sitúan entre los símbolos `<!--` y `-->`.

*Ejemplo:*

**`<!-- Comentario en un .html: esta línea no aparece en el navegador Web -->`**

- En los ficheros **.css** los comentarios se sitúan entre los símbolos `/*` y `*/`

*Ejemplo:*

**`/* Comentario en un fichero de estilos */`**

## VALIDACIÓN

Para comprobar que **la página html** está correctamente formada haremos uso de un validador, una aplicación que identifica los errores cometidos al crear el fichero (faltan etiquetas, un parámetro es incorrecto, etc...).

[https://validator.w3.org/#validate\\_by\\_upload+with\\_options](https://validator.w3.org/#validate_by_upload+with_options)

Tanto el lenguaje XHTML, como **el CSS** deben estar correctamente escritos. Para ello existen páginas de validación que nos indican los posibles errores y nos permite así corregirlos:

[https://jigsaw.w3.org/css-validator/#validate\\_by\\_upload](https://jigsaw.w3.org/css-validator/#validate_by_upload)

[https://validator.w3.org/unicorn/#validate-by-upload+task\\_full-css](https://validator.w3.org/unicorn/#validate-by-upload+task_full-css)

## EJEMPLOS DE USO DE ETIQUETAS

### Ejemplo Lista Ordenada:

1. Elemento 1
2. Elemento 2
3. Elemento 3
4. Elemento 4

```
<ol>
  <li>Element 1</li>
  <li>Element 2</li>
  <li>Element 3</li>
  <li>Element 4</li>
</ol>
```

### Ejemplo Lista **No** Ordenada:

- Elemento 1
- Elemento 2
- Elemento 3
- Elemento 4

```
<ul>
  <li>Element 1</li>
  <li>Element 2</li>
  <li>Element 3</li>
  <li>Element 4</li>
</ul>
```

## XHTML – Ejemplos de Tablas

**TABLA 1**

```
<table>
  <tr>
    <td> A </td>
    <td> B </td>
  </tr>
  <tr>
    <td> C </td>
    <td> D </td>
  </tr>
</table>
```

A	B
C	D

**TABLA 2**

```
<table>
  <tr>
    <td colspan="2"> A </td>
  </tr>
  <tr>
    <td> B </td>
    <td> C </td>
  </tr>
</table>
```

A	
B	C

**TABLA 3**

```
<table>
  <tr>
    <td> A </td>
    <td> B </td>
  </tr>
  <tr>
    <td colspan="2"> C </td>
  </tr>
</table>
```

A	B
C	

**TABLA 4**

```

<table>
  <tr>
    <td rowspan="2"> A </td>
    <td> B </td>
  </tr>
  <tr>
    <td> C </td>
  </tr>
</table>

```

A	B
	C

**TABLA 5**

```

<table>
  <tr>
    <td> A </td>
    <td rowspan="2"> B </td>
  </tr>
  <tr>
    <td> C </td>
  </tr>
</table>

```

A	B
C	

**TABLA 6**

```

<table>
  <tr>
    <td rowspan="2"> A </td>
    <td colspan="2"> B </td>
  </tr>
  <tr>
    <td> C </td>
    <td> D </td>
  </tr>
</table>

```

A	B	
	C	D

**TABLA 7**

```

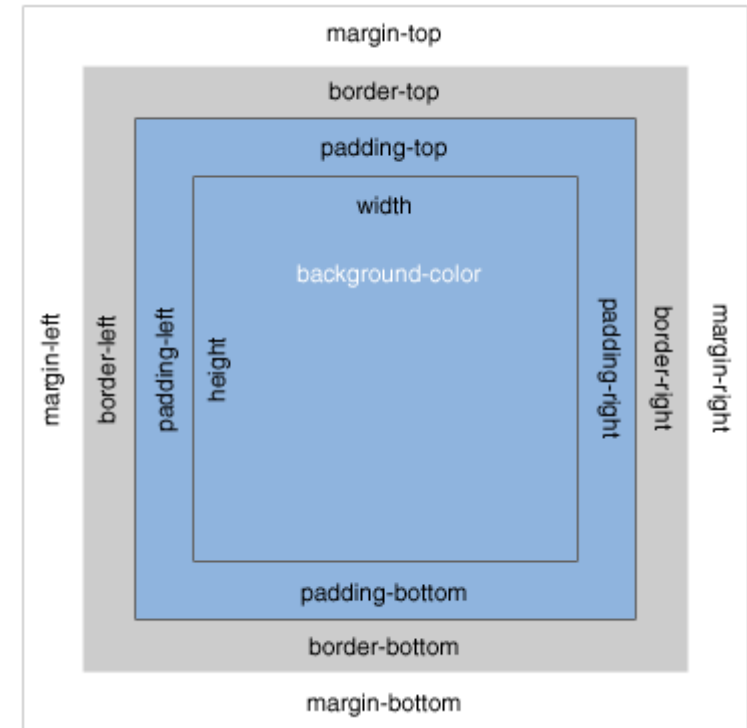
<table>
  <tr>
    <td> A </td>
    <td> B </td>
    <td> C </td>
    <td> D </td>
  </tr>
  <tr>
    <td> E </td>
    <td> F </td>
  </tr>
</table>

```

A	B	C	D
E	F		



## CSS (Cascade Style Sheet)



Cajas			
<b>margin</b> margin-top margin-right margin-bottom margin-left	<b>padding</b> padding-top padding-right padding-bottom padding-left	<b>border</b> border-top border-right border-bottom border-left	<b>border-width</b> border-width-top border-width-right border-width-bottom border-width-left border-radius
<b>border-color</b> border-color-top border-color-right border-color-bottom border-color-left	<b>border-style</b> border-style-top border-style-right border-style-bottom border-style-left	<u>Tipo de bordes:</u> dotted dashed solid double	groove ridge inset outset

## **Ejemplos**

```
/* para una etiqueta que tenga como parámetro id="cabecera" */
#cabecera {
    width: 50%; /* anchura en la página del elemento cabecera */
    border-width: 1px; /* parámetros de las bordes */
    border-color: #FF0000;
    border-style: solid;
    margin: 20px; /* margen con respecto al elemento que contiene cabecera */
    padding: 20px; /* espacio del contenido cerca de la caja del elemento cabecera */
    font-family: verdana; /* tipo de letra */
    color: #FFFFFF; /* color del texto */
    background-color: #6699CC; /* color del fondo del elemento cabecera */
}
```

La definición de cada lado, tanto de **padding** como de **margin** se pueden hacer por separado, como por ejemplo:

```
#navegacion {
    margin-top: 10px;
    margin-bottom: 10px;
    margin-left: 10px;
    margin-right: 10px;
}
```

O bien todo junto, ahorrando líneas de código, con el siguiente patrón **top right bottom left**;

```
#navegacion {  
    padding 10px 10px 10px 10px;  
}  
  
#navegacion {  
    /* agrupados top y bottom primero, left y right después */  
    padding 10px 10px;  
}  
  
/* para una etiqueta que tenga como parámetro class="importante" */  
.importante {  
    font-family: "verdana"; /* tipo de letra */  
    color: #FF0000; /* color del texto */  
    font-weight: bold; /* negrita */  
    font-size: 30px; /* medida de la fuente */  
}  
  
/* para cualquier etiqueta p que no tenga asignado un parámetro class o id */  
p {  
    background-color: #FF3300; /* color del fondo de todos los elementos p */  
}
```

**Sintaxis fichero CSS**

```
/* Comentarios */

selector {
  propiedad_1: valor;
  propiedad_2: valor;
  ...
}
```

**Pseudo-selectores**

Se escriben detrás de un selector para indicar alguna particularidad

:first-line	Primera línea del elemento
:first-letter	Primera letra del elemento
:hover	Elemento cuando se sitúa encima el ratón
:link	Links no activos ni visitados, sin ratón encima
:visited	Links visitados

## Enlaces

```
a {
  text-decoration:none;
}
a:link {
  color:#d90000;
}
a:visited {
  color:#cc6666;
}
a:hover, a:focus{
  text-decoration:underline;
}
a:active {
  background-color:#d90000;
}
```

## Selectores

Los selectores se pueden **agrupar con una coma**, para aplicar las mismas reglas a diferentes elementos:

```
h1, h2 {  
    font-family: arial, verdana;  
}
```

Los selectores se pueden **anidar con un espacio en blanco** para limitar el ámbito de aplicación:

```
#navegacion a {  
    text-decoration:none;  
}  
  
#navegacion li {  
    list-style-type:none;  
    display:inline;  
}
```

## LISTA NO ORDENADA

### XHTML

```
<ul class="nombre_clase">
  <li> A </li>
  <li> B </li>
  <li> C </li>
  <li> D </li>
  <li> E </li>
</ul>
```

### CSS

```
.nombre_clase {
  list-style-type: disc;
}
```

- A
- B
- C
- D
- E

```
.nombre_clase {
  list-style-type: square;
}
```

- A
- B
- C
- D
- E

```
.nombre_clase {
  list-style-type: circle;
}
```

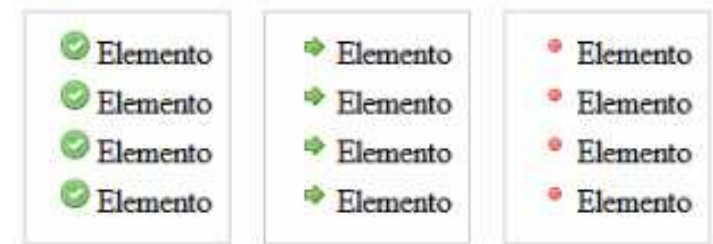
- A
- B
- C
- D
- E

```
.nombre_clase {
  list-style-type: none;
  display: inline;
}
```

A B C D E

### Ejemplo de lista no ordenada con imágenes

```
ul.ok { list-style-image: url("imagenes/ok.png"); }  
ul.flecha { list-style-image: url("imagenes/flecha.png"); }  
ul.circulo { list-style-image: url("imagenes/circulo_rojo.png"); }
```



## LISTA ORDENADA

### XHTML

```
<ol class="nombre_clase">  
  <li> Silla </li>  
  <li> Mesa </li>  
  <li> Sofá </li>  
  <li> Escritorio </li>  
  <li> Armario </li>  
</ol>
```

CSS

```
.nombre_clase {  
  list-style-type: lower-latin;  
}
```

a. Silla  
b. Mesa  
c. Sofá  
d. Escritorio  
e. Armario

```
.nombre_clase {  
  list-style-type: upper-latin;  
}
```

A. Silla  
B. Mesa  
C. Sofá  
D. Escritorio  
E. Armario

```
.nombre_clase {  
  list-style-type: decimal;  
}
```

1. Silla  
2. Mesa  
3. Sofá  
4. Escritorio  
5. Armario

```
.nombre_clase {  
  list-style-type: none;  
}
```

Silla  
Mesa  
Sofá  
Escritorio  
Armario

Puedes encontrar más en... [http://www.w3schools.com/css/tryit.asp?filename=trycss\\_list-style-type\\_all](http://www.w3schools.com/css/tryit.asp?filename=trycss_list-style-type_all)



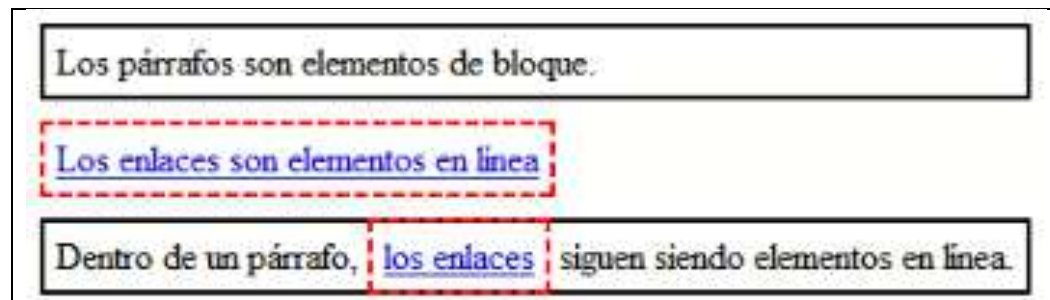
## CSS – Posicionamiento

### Tipos de elementos

El estándar HTML clasifica a todos sus elementos en dos grandes grupos: elementos **en línea** y **de bloque**.

Los elementos **de bloque** (*"block elements"* en inglés) siempre empiezan en una nueva línea y ocupan todo el espacio disponible hasta el final de la línea.

Por su parte, los elementos **en línea** (*"inline elements"* en inglés) no empiezan necesariamente en nueva línea y sólo ocupan el espacio necesario para mostrar sus contenidos.



Algunos elementos **en línea** definidos por HTML son:

a, abbr, b, big, br, cite, code, em, font, i, img, input, label, select, small, span, strike, strong, sub, sup, textarea, u.

Algunos elementos **de bloque** definidos por HTML son:

address, blockquote, center, div, dl, fieldset, form, h1, h2, h3, h4, h5, h6, hr, menu, ol, p, pre, table, ul.

## Flujo del documento

Un elemento se coloca tan **arriba** y tan a la **izquierda** como sea posible. Cuando se añaden otros elementos, se colocan a su derecha hasta que no hay más espacio. Cuando no hay más espacio a la derecha, saltan una línea hacia abajo y vuelven a empezar por la izquierda.

### Posicionamiento por defecto

```
position: static; /* corresponde al posicionamiento normal o estático. */
```

### Posicionamiento relativo: flujo desplazado

Desplaza la caja fuera de su posición normal en el flujo (si se indica en desplazamiento con **top**, **right**, **bottom**, **left**). Marca la **posición original** del elemento como **protegida** (ningún otro elemento ocupa su lugar). Los demás elementos del documento se comportan como si el elemento aún se encontrara a su lugar original dentro del flujo normal.

```
position: relative;  
top: 25px;  
left: 60%;
```



## Posicionamiento absoluto: fuera de flujo

Se saca el elemento del flujo. Los demás elementos se comportan como si ya no estuviera allí. El origen de coordenadas del desplazamiento depende del posicionamiento de su elemento contenedor.

```
position: absolute;  
top: 25px;  
left: 60%;
```



## Posicionamiento fijo

Se comporta como el posicionamiento absoluto, pero en este caso el elemento **no se desplaza junto al resto de contenido cuando se visualiza la página**, hacia arriba o hacia abajo. El elemento permanece inamovible en la pantalla. Las cajas no modifican su posición ni aunque el usuario suba o baje la página en la ventana de su navegador.

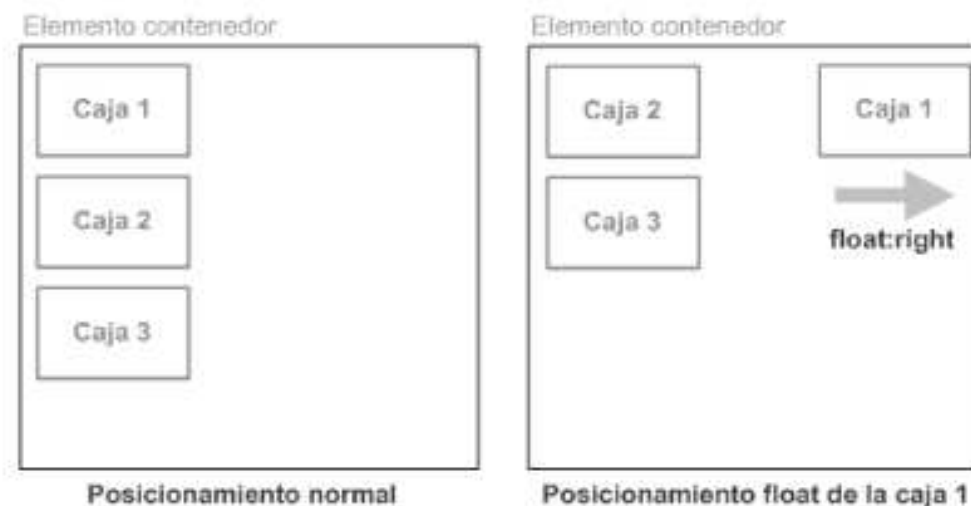
```
position: fixed;  
top: 10px;  
left: 10px;
```

## Float: cuadros flotantes

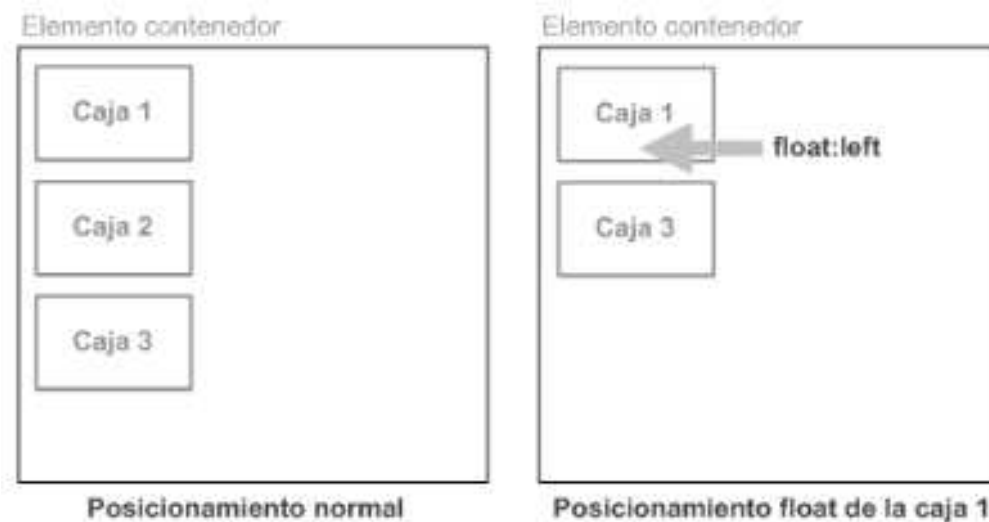
*El posicionamiento flotante es el más difícil de comprender pero al mismo tiempo es el más utilizado. La mayoría de estructuras de las páginas web complejas están diseñadas con el posicionamiento flotante*

A diferencia del resto, **float** no es un valor de la propiedad **position** de xhtml, sino una propiedad CSS diferente. Con **float** el elemento se eleva y sale del flujo normal, flota como un globo dentro del elemento contenedor y se desplaza hacia la izquierda o a la derecha con: **float: left;** o con: **float: right;**

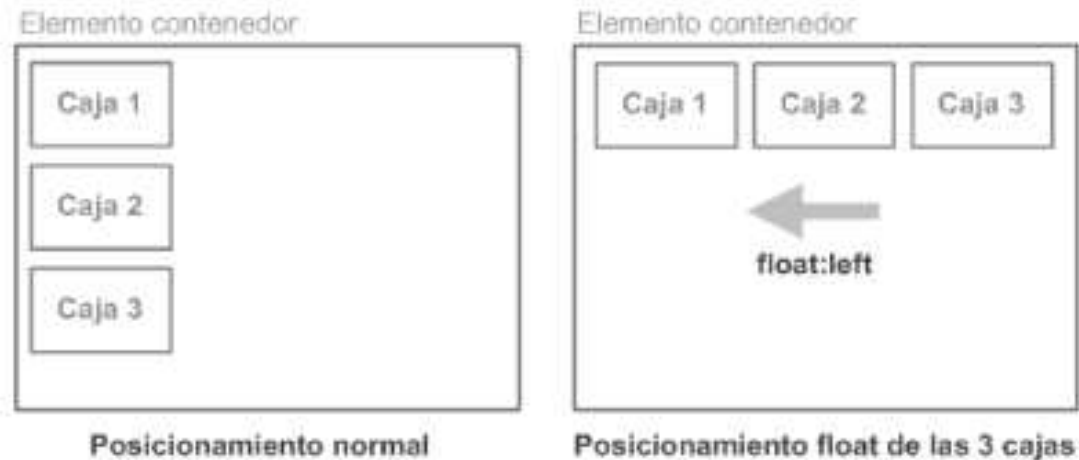
Cuando se posiciona una caja de forma flotante, la caja deja de pertenecer al flujo normal de la página, lo que significa que el resto de cajas ocupan el lugar dejado por la caja flotante. La caja flotante se posiciona lo más a la izquierda o lo más a la derecha posible de la posición en la que se encontraba originalmente.



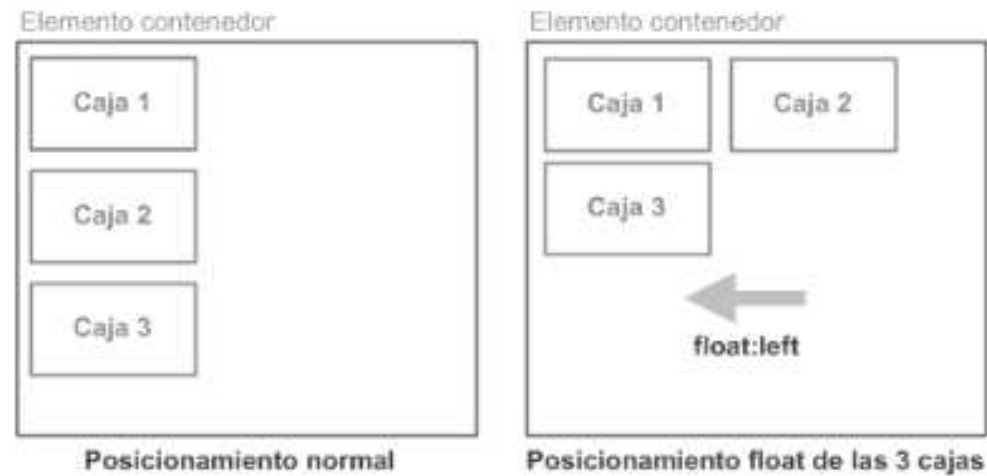
Si en el anterior ejemplo la caja 1 se posiciona de forma flotante hacia la izquierda, el resultado es el que muestra la siguiente imagen (la caja 2 ahora se muestra donde estaba la caja 1 y la caja 3 se muestra donde estaba la caja 2).



Si existen otras cajas flotantes, al posicionar de forma flotante otra caja, se tiene en cuenta el sitio disponible. En el siguiente ejemplo se posicionan de forma flotante hacia la izquierda las tres cajas:



Si no existe sitio en la línea actual, la caja flotante baja a la línea inferior hasta que encuentra el sitio necesario para mostrarse lo más a la izquierda o lo más a la derecha posible en esa nueva línea:



## Columnas

En CSS casi siempre hay varias formas de alcanzar el objetivo deseado, especialmente en el caso de los **diseños de varias columnas**. Para que los diseños sean más **flexibles**, es recomendable indicar las anchuras en **porcentaje**. La ventaja en este caso es que en resoluciones de pantalla grandes se aprovecha mejor la superficie de pantalla disponible.

```
#navegacion{  
position:absolute;  
top:0;  
left:0;  
width:25%;  
}
```

```
#principal{  
width:50%;  
margin-left:25%;  
margin-right:25%;  
}
```

```
#navegacio2{  
position:absolute;  
top:0;  
right:0;  
width:25%;  
}
```

Cuando se utilice float, siempre se debe declarar la anchura con ***width***, en caso contrario el comportamiento del navegador se vuelve demasiado complejo.

```
#navegacion{  
float:left;  
width:30%;  
}
```

```
#principal{  
margin-left:30%;  
}
```



Si se desea que los siguientes elementos empiecen debajo de él (al finalizar la dimensión vertical del elemento flotante), se utiliza la propiedad **clear**:

```
clear: left; /* acaba un float: left */
clear: right; /* acaba un float: right */
clear: both; /* acaba un float: left y un float: right */
```

```
#navegacion{
float:left;
width:30%;
}
```

```
#principal{
margin-left:30%;
}
```

```
#pie{
clear:both;
}
```

Si tenemos un pie **y la columna #navegacion es más larga que #principal**, se solapará. Hemos de usar **float**.

```
#navegacion{  
float:left;  
width:30%;  
}
```

```
#principal{  
float:right;  
width:70%;  
}
```

```
#pie{  
clear:both;  
}
```

En el fichero html, se podría definir antes **#principal** que **#navegacion** ya que después float se encarga de poner cada uno a su lugar.

Para hacer **3 columnas**, se pueden crear dos nuevos div dentro de principal, con la siguiente estructura:

```
<div id="contenedor">
  <div id="cabecera">...</div>
  <div id="principal">
    <div id="contenido">...</div>
    <div id="columna">...</div>
  </div>
  <div id="navegacion">...</div>
  <div id="pie">...</div>
</div>
```

#cabecera		
#navegacion{ float:left; width:25%; }	#contenido{ float:left; width:60%; }	#columna{ float:right; width:40%; }
#pie{ clear:both; }		

Se vuelven a definir **#contenido** y **#columna** con floats opuestos y los anchos que nos interesen. Los valores de anchura vuelven a ser relativos al 100% de la medida de **#principal**, no se suman al ancho de **#navegacion**.