

[¿Olvidaste tu usuario o clave?](#)

Usuario

Password

[entra](#) o [regístrate](#)

[Blog](#) [Foros](#) [Tutoriales](#) [Cursos](#) [Videotutoriales](#) [Comics](#) [Tags](#)

[Publica un tutorial](#)

[¿Qué es Cristalab?](#)

[Ejemplos](#)

[Eventos](#)

[Contacto](#)

» [Tutoriales](#) / Tutorial de posicionamiento y layout en CSS

## Tutorial de posicionamiento y layout en CSS



Por Ramm el 17 de Julio de 2005 con 417,733 visitas.

[HTML, CSS y Javascript](#) [Otros tutoriales por Ramm.](#)

### ¿A qué le llamamos posicionamiento?

---

Obviamente no me refiero a los buscadores como Google o Yahoo, posicionamiento en CSS consiste en ubicar los elementos HTML en la pagina, el "layout" o distribución de la pagina. Basicamente esto se logra ordenándole desde CSS a los elementos que tengan cierto tamaño y posición, además de obtener una buena organización del código HTML o XHTML.

El posicionamiento con CSS nos permite control total sobre los elementos de la pagina, además con documentos muy livianos y completamente válidos ante [W3C](#).

### Los Divs

---

Antes de comenzar con el CSS quiero exponer aqui la importancia de usar los divs, un div es basicamente un contenedor, en el podemos meter cualquier clase de contenido, con la gran ventaja de que los podemos manipular a nuestro antojo con CSS, colocarlo en cualquier parte de la pagina, de cualquier tamaño, de cualquier color, con bordes o sin ellos, con imagenes de fondo o sin ellas, a diferencia de las tablas, los divs no se dividen por dentro, pero podemos anidarlos y organizarlas casi como las tablas (en caso de ser necesario).

Para lograr que el posicionamiento no se vuelva complicado, debemos tener en cuenta que la organización en el código HTML es muy importante, tener bien estructurados los elementos nos facilitará el trabajo con CSS, con esto quiero decir que debemos contar con un código semántico.

#### **Semántica**

*Proviene del griego "semantikos", que quería decir "significado relevante", derivada de "sema", lo que significaba "signo") Se dedica al estudio del significado de los signos lingüísticos y de sus combinaciones, desde un punto de vista sincrónico o diacrónico. (definición de wikipedia)*

Esto significa que a la hora de hacer el código HTML, este bien organizado jerárquicamente, o sea anidar los divs según importancia, agrupados según sección, utilizar las etiquetas h1, h2, h3, etc para los títulos, las etiquetas <ul> y <li> para las listas, las etiquetas <p> en los párrafos, ayuda a organizar mejor el contenido, además de que están ahí para eso. La mayoría se ha olvidado del uso de estas etiquetas por trabajar en entornos gráficos para diseñar, o porque solo le dan importancia a lo que se ve. Otras ventajas son que los buscadores entienden el código, por ejemplo Google sabe que <h1>Titulo</h1> es un título y le da la importancia necesaria, si alguien usa Lynx para navegar, o cualquier navegador de texto, podrá ver un contenido ordenado y lógico y es mucho más fácil de mantener/modificar. Un ejemplo de una estructura sencilla:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=iso-  
<title>Documento sin título</title>  
</head>  
  
<body>  
<div id="contenedor">  
<div id="encabezado">  
<h1>Titulo de la Pagina</h1>  
</div>  
<div id="menu">  
<ul>  
<li><a href="#">Link</a></li>  
<li><a href="#">Link</a></li>  
<li><a href="#">Link</a></li>  
<li><a href="#">Link</a></li>  
</ul>  
</div>  
<div id="contenido">  
<h2>Subtitulo 1 </h2>  
<p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. C  
<p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Se  
<h3>Sub-subtitulo</h3>  
<p> Praesent tincidunt. Nam quis quam at nunc pretium eleifend.  
<div id="footer">Copyright2005 - Derechos reservados</div>  
</div></div>  
</body>  
</html>
```

[Así se ve el ejemplo sin estilos](#)

De aquí en adelante vamos con lo que nos interesa, lo primero que haremos será centrar nuestro div "contenedor" en la ventana del navegador, y darle un tamaño específico



```

/*empezamos por esto, para evitar resultados extraños en distintos navegadores
body {
    font-family:Georgia, "Times New Roman", Times, serif;
    font-size:12px;
    color:#666666;
    text-align:center; /*aquí mandamos a centrar todo el contenido de la página
}

```

y luego al div:

```

#contenedor {
    width:600px; /*ancho total de la pagina*/
    border:1px solid #333333; /*encadeno los 3 atributos claves de la pagina
    background-color:#FFF8F0;
    margin:10px auto; /*10px arriba y abajo, y auto a los lados,
    text-align:left; /*aquí alineamos todo de nuevo a la izquierda
}

```

Bien, ya tenemos el div contenedor centrado en la pagina, ahora vamos con el "encabezado":

Aquí, veremos como funciona float. Así como suena, float nos permite "flotar" elementos hacia la izquierda o la derecha de la pagina, haciendo que los demas elementos fluyan alrededor

```
#encabezado {
  margin:10px; /*para que no se pegue al borde*/
  padding:10px; /*algo de relleno*/
  width:224px; /*este ancho es para que cuadre con el texto*/
  height:50px; /*idem*/
  border:1px solid #333333; /*decoracion*/
  background-color:#CC6600; /*more*/
  float:left; /*lo flotamos a la izquierda*/
  display:inline; /*(*)*/
}
```

*(\*)display:inline; se usa en este caso para corregir un error en IE, en la interpretación del margin y el padding, si flotamos a la izquierda, y hay 10px de margin a la izquierda, IE sumara el doble de ese margen. Con esta línea corregimos eso.*

Ahora algo de estilo para el título:

```
#encabezado h1 { /*encadenamos el identificador del div, con la
  font-size:18px;
  font-weight:bold;
  width:224px;
  border-bottom:16px solid #FFFFFF;
  color:#FFFFFF;
  margin:0; /*los h1 vienen con margen, con esto se lo quito*/
  text-transform:uppercase; /*convierte todas en mayusculas*/
  padding-bottom:10px;
}
```

Seguimos con el menú:

```
#menu {
  float:left;
  margin:10px 10px auto 0;
  border:1px solid #333333;
  width:320px;
  height:24px;
}
```

Y luego con la lista. las listas traen varios pre-formatos desde HTML, como las viñetas, margen, saltos de línea a cada ítem, padding, etc... Aquí vamos a cambiar completamente el aspecto de una lista, con sus vínculos dentro.

```

ul {
  list-style-type:none; /*con esto quitamos las viñetas*/
  margin:0; /*le quitamos el margen a la lista*/
  padding:0; /*y el relleno*/
}
li {
  margin:0; /*le quito el margen al elemento de la lista*/
  float:left; /*y los floto a la izquierda para eliminar el sa
}
li a {
  display:block; /*con esto el vinculo tendra un area rectangu
background-color:#CC3300; /*algo de color*/
color:#FFFFFF; /*mas*/
width:58px; /*defino el ancho del rectangulo del vinculo*/
padding:4px 0 5px 22px; /*sumamos el relleno al ancho y alto
text-decoration:none; /*elimino el subrayado del vínculo*/
font-weight:bold; /*decoración*/
text-transform:uppercase; /*decorare los vinculos en mayuscu
}
li a:hover {
  background-color:#999900; /*decoración para el evento hover
}

```

Esto es suficiente para cambiar el aspecto de la lista, y los vinculos en el menu. Así podemos jugar con los anchos y rellenos para lograr menús mas complejos, poniendole bordes, etc ...

Ahora el contenido:

```

#contenido {
  width:580px; /*el ancho mas el padding me da 600px de ancho
clear:both; /*con esto nos aseguramos de que no se ponga nad
padding:10px;
}

```

Algo de estilo para los titulos:

Te puede



Tutor



Cómo  
CSS



Cómo  
de ur



Conv



PSD

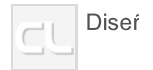


Camt  
abrir



Cómo  
Del M

```
#contenido h2 {
  font-size:16px;
  color:#CC3300;
  width:246px;
  text-transform:uppercase;
  border-bottom:1px dotted #CC3300;
}
#contenido h3 {
  width:246px;
  font-size:14px;
  color:#999900;
  border-bottom:1px dotted #999900;
}
```



[« Regresar a los](#)

Y terminamos con el footer:

```
#footer {
  width:560px; /*menos porque debemos sumar el padding del div
  text-align:right;
  padding:10px;
  font-size:80%; /*mas pequeño el texto*/
  font-style:italic;
  color:#999999;
}
```

Los comentarios en el código explican lo que hace cada uno de los atributos clave usados en el ejemplo.

Y con esto terminamos nuestra página, obteniendo [este ejemplo terminado](#), una página semántica, fácilmente entendible por los navegadores, los buscadores y por nosotros mismos.

1

Me gusta

A 24 personas les gusta esto.

[css](#) [diseño](#) [html](#) [xhtml](#)

[Envía un comentario \(0\)](#)



Tutorial de AJAX



Componer un diseño de 3 columnas en XHTML y CSS



Crear n usando

