

Propiedades Básicas CSS3

CSS



Introducción a CSS 3

CSS 3 trae grandes novedades para el diseño de webs y algunos navegadores comienzan a implementar CSS 3.

Desde que CSS comenzó han pasado muchos años y ya vamos por la especificación de CSS3, que incorpora una serie de novedades que vamos a tratar de resumir en este artículo.

Qué es CSS

Si no sabes lo que es CSS probablemente te interesaría comenzar leyendo nuestro manual de CSS o la sección de CSS a fondo. No obstante, cabría decir que CSS es un lenguaje para definir el estilo o la apariencia de las páginas web, escritas con HTML o de los documentos XML. CSS se creó para separar el contenido de la forma, a la vez que permite a los diseñadores mantener un control mucho más preciso sobre la apariencia de las páginas.

Con CSS 3, más control sobre la forma

El objetivo inicial de CSS, separar el contenido de la forma, se cumplió ya con las primeras especificaciones del lenguaje. Sin embargo, el objetivo de ofrecer un control total a los diseñadores sobre los elementos de la página ha sido más difícil de cubrir. Las especificaciones anteriores del lenguaje tenían muchas utilidades para aplicar estilos a las webs, pero los desarrolladores aun continúan usando trucos diversos para conseguir efectos tan comunes o tan deseados como los bordes redondeados o el sombreado de elementos en la página.

CSS 1 ya significó un avance considerable a la hora de diseñar páginas web, aportando mucho mayor control de los elementos de la página. Pero como todavía quedaron muchas otras cosas que los diseñadores deseaban hacer, pero que CSS no permitía especificar, éstos debían hacer uso de trucos para el diseño. Lo peor de esos trucos es que muchas veces implica alterar el contenido de la página para incorporar nuevas etiquetas HTML que permitan aplicar estilos de una manera más elaborada. Dada la necesidad de cambiar el contenido, para alterar al diseño y hacer cosas que CSS no permitía, se estaba dando al traste con alguno de los objetivos para los que CSS fue creado, que era el separar por completo el contenido de la forma.

CSS 2 incorporó algunas novedades interesantes, que hoy ya utilizamos habitualmente, pero CSS 3 todavía avanza un poco más en la dirección, de aportar más control sobre los elementos de la página.

Así pues, la novedad más importante que aporta CSS 3, de cara a los desarrolladores de webs, consiste en la incorporación de nuevos mecanismos para mantener un mayor control sobre el estilo con el que se muestran los elementos de las páginas, sin tener que recurrir a trucos o hacks, que a menudo complicaban el código de las web.

Propiedades nuevas en CSS 3

He aquí una lista de las principales propiedades que son novedad en CSS3.

El color de primer plano

La propiedad color hace referencia al **foreground** color, es decir, al color que está por encima del fondo. Hablando en plata, viene a ser el color del texto. Si queremos que nuestra página tenga las letras de gris oscuro, lo conseguimos con esto:

```
body { color : #666; }
```

El fondo

Podemos modificar el fondo de un elemento con la propiedad background, que tiene la siguiente sintaxis:

```
background : color | image | repeat | attachment | position ;
```

El primer parámetro corresponde al color de fondo, los siguientes son relativos a la imagen de fondo:

image: aquí indicamos la ruta a la imagen que pondremos. Por ejemplo, url(fondo.gif).

repeat: con esto establecemos si queremos que la imagen se repita o no, tanto horizontal como verticalmente. Con repeat se repite siempre en ambos sentidos (valor por defecto), mientras que con no-repeat no se repite nunca. Con repeat-x se repite sólo en horizontal, y con repeat-y sólo en vertical.

attachment: sirve para indicar si el fondo se queda fijo en el sitio o se desplaza con scroll. Es un poco difícil de explicar, así que lo mejor es que lo probéis vosotros mismos: con scroll (valor por defecto) el fondo se desplaza, y con fixed se queda siempre en el mismo sitio.

position: indica la posición del fondo. Indicamos tanto la posición desde la izquierda como desde arriba (ya sea en píxeles, porcentajes, o incluso palabras).

En este caso, si podemos usar palabras, es preferible que lo hagamos. Disponemos de top (arriba), bottom (abajo), left (izquierda) y right (derecha).

Los valores por defecto son 0 % 0 %, que sitúan al fondo en la esquina superior izquierda. Si lo quisiéramos en las coordenadas 20,30 (tomando como el origen a la esquina superior izquierda), escribiríamos 20px 30px. Si queremos el fondo centrado, pues 50 % 50 %.

Podemos omitir alguna propiedad si queremos. Además, podemos establecer los valores de forma individual, usando las propiedades **background-color**, **background-repeat**, etc.

Veamos algunos ejemplos para poner fondo a nuestra página:

```
/* sólo color de fondo */
```

```
body { background : #fff; }
```

```
/* color de fondo e imagen en mosaico */
```

```
body { background : #fff url(fondo.gif); }
```

```
/* imagen fija , centrada y sin repetir */
```

```
/* a modo de marca de agua */
```

```
body {
```

```
    background-color: #fff;
```

```
    background-image: url(fondo.gif);
```

```
    background-attachment: fixed;
```

```
    background-repeat: no-repeat;
```

```
    background-position: 50% 50%;
```

```
}
```

Fuente

Hay varias propiedades que nos permiten jugar con el aspecto del texto. Podemos englobar todas bajo font, pero primero veremos algunas subpropiedades.

Antes de nada, debemos considerar que no todos los ordenadores tienen las mismas fuentes instaladas. Es decir, que mi maravillosa fuente llamada chachi.ttf no tiene por qué tenerla mi vecino. Es más, lo normal es que no la tenga. ¿Entonces qué hacemos? Usar sólo fuentes "estándar", que tengan la mayoría de ordenadores. Además, podemos especificar varias, de forma que si no se tiene la primera, se muestre la segunda, si no se tiene la segunda, pues la tercera, etc.

A lo que íbamos. La propiedad que sirve para cambiar la fuente es font-family. Como acabamos de ver, podemos indicar varias fuentes, por orden de preferencia. Si el nombre de una fuente tiene espacios en blanco, hay que ponerla entre comillas. Por ejemplo:

```
font-family: "Comic Sans MS", Arial, sans-serif;
```

EL tamaño de la fuente lo controlamos con font-size. Podemos indicar medidas en píxeles (px) o en puntos (pt).

Para alinear el texto tenemos a text-align, que puede tomar los valores right (derecha), left (izquierda), center (centrado) o justify (justificado).

Es muy normal tener una clase así:

```
.centrar { text-align: center; }
```

Si queremos indentar los párrafos, pues usamos text-indent:

```
p { text-indent : 2em; }
```

¿Qué significa em? Es una unidad relativa. Lo más fácil es trabajar con píxeles o porcentajes, pero en algunos casos conviene utilizar otras unidades. 1 em equivale al tamaño de la fuente. Bueno, exactamente equivale a la anchura de la letra "M" mayúscula.

Así, si por herencia o por cualquier otra cosa el texto se muestra en un tamaño de fuente diferente, la proporción de indentado sería siempre la misma. Si pusiéramos 20px, siempre sería esa, aunque el tamaño de la fuente fuera de 20 o de 32 píxeles.

También podemos cambiar la decoración del texto mediante `text-decoration`. Puede tomar diversos valores, como `none` (sin adornos), `underline` (subrayado), `overline` (subrayado superior) o `line-through` (tachado).

Si queremos que nuestros links no tengan subrayado, pondremos:

```
a { text-decoration : none; }
```

Podemos transformar las mayúsculas y minúsculas con la propiedad `text-transform`. Si ponemos `lowercase`, todo se mostrará en minúsculas; con `uppercase`, en mayúsculas; y con `capitalize` pondremos una letra capital al principio de cada palabra.

En cuanto al espaciado, para la distancia entre palabras usamos `word-spacing`; para el de las letras, `letter-spacing`; y para el interlineado, `line-height`.

Un ejemplo con todo esto, para la etiqueta ``:

```
strong {  
  color: #000;  
  letter-spacing: 0.25 em;  
  text-trasform: uppercase;  
}
```

¿Demasiada información? Sí, pero ten en cuenta que no hay que memorizar todo esto, lo importante es saber que existe una propiedad que hace tal cosa, no cómo se llama ni qué parámetros tiene. Para eso ya están las guías de referencia rápida y los editores de CSS.

El modelo de caja

los fundamentos más importantes del CSS: el modelo de caja (box model). Es muy fácil, pero entenderlo bien es vital para poder realizar una buena maquetación de la web.

¿Cómo es el modelo de caja?

todos los elementos de una web (párrafos, enlaces, imágenes, tablas, etc.) son cajas rectangulares. Los navegadores sitúan estas cajas de la forma que nosotros les hayamos indicado para maquetar la página.

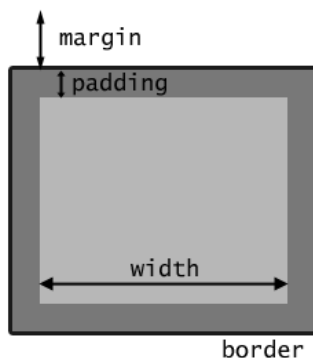
Hay dos tipos de cajas: **block** e **inline**. Los elementos **block** rompen el flujo de maquetación. Esto es, aparecen solos, insertando “saltos de línea”. Los elementos **inline** siguen el flujo, y están contenidos dentro de elementos de bloque.

Por ejemplo, un párrafo sería un elemento **block** (no podemos tener un párrafo al lado del otro, sino que dos párrafos seguidos aparecerán uno abajo del otro. En cambio, un enlace es un elemento **inline**, ya que no “corta” el texto donde está metido.

Estas dos diferencias son importantes, pero hay que tener en mente que ambos tipos comparten el modelo de caja, que es el que aparece en la figura 1.

Modelo de Caja

Figura 1 - Modelo de Caja



Las propiedades más importantes de una caja son:

width (ancho),

height (alto),

padding (relleno),

border (borde)

margin (margen).

Ancho y alto

`width` representa el ancho de la caja. Pero es el ancho interior, es decir, si bordes, márgenes, ni padding. Podemos indicar este ancho en medidas absolutas (normalmente píxeles) o relativas (normalmente %).

Aunque los elementos `inline` tienen `width`, si la modificamos con CSS no veremos ningún resultado visual. Esto es porque el ancho de estos elementos se establece automáticamente para que se ajuste a las dimensiones del elemento inline. Por ejemplo, si tenemos un enlace que consiste en un texto de cinco caracteres, el ancho `width` de este elemento será lo que ocupen esos cinco caracteres.

Sobre el alto de la caja, se controla con la propiedad `height`, y todo lo que hemos dicho antes sobre el ancho, también se aplica aquí.

Padding

Con padding establecemos la distancia de “relleno” entre el límite interior de la caja y el exterior (borde). Es una definición muy mala, pero se entiende a la perfección en el dibujito.

Si queremos poner un padding de 20 píxeles para toda la caja, lo haríamos así:

```
padding : 20 px;
```

Podemos establecer un padding distinto para cada lado, usando los sufijos `-top (superior)`, `-bottom (inferior)`, `left (izquierda)` y `right (derecha)`:

```
padding-top: 10px;
```

```
padding-bottom: 5px;
```

```
padding-left: 30px;
```

```
padding-right: 20px;
```

Podemos abreviar lo anterior en una sola línea, indicando primero el `padding` superior y luego siguiendo el orden de las agujas del reloj. Es decir, nos quedaría: arriba, derecha, abajo, izquierda. El ejemplo anterior se acortaría así:

```
padding: 10px 20px 5px 30px;
```


Otro atajo útil es especificar sólo dos medidas: una corresponderían al padding superior e inferior, y la otra al lateral. Si queremos que los paddings superior e inferior sean de 10 píxeles, y los laterales (izquierdo y derecho) de 20 píxeles, escribimos:

```
padding: 10px 20px;
```

Bordes

Si queremos que nuestra caja tenga bordes, lo conseguimos con la propiedad border. Tiene la siguiente sintaxis:

```
border: width | style | color
```

Como habrás supuesto, width especifica el grosor del borde. Normalmente es una medida en píxeles, pero también podemos utilizar las palabras thin (fino), medium (normal) y thick (grosso). Por supuesto, cómo de gordo es thick queda a interpretación del navegador.

En cuanto a **style**, es el tipo de borde. Hay bastantes, pero los más comunes son:

```
solid (línea continua),
```

```
dashed (línea discontinua),
```

```
dotted (línea de puntos) y
```

```
double (línea continua doble).
```

Podemos escoger un tipo de borde diferente para cada lado con los sufijos -top, -bottom, -left y -right. Por ejemplo, para poner que algo tenga el borde inferior de 1 píxel a puntitos rojos:

```
border-bottom: 1px dotted #f00;
```

Para eliminar el borde, simplemente ponemos que tiene de grosor 0 píxeles o que el estilo del borde es none. Esto es muy importante con las imágenes, ya que si tenemos una imagen enlazando a algo, los navegadores la ponen con un reborde muy feo. Así que esto se ha convertido ya en un fijo de las hojas de estilos:

```
img { border: none; }
```

Márgenes

Los márgenes se controlan con la propiedad `margin`, y es la distancia entre el borde de la caja y los elementos que la rodean.

En cuanto a la forma de usarla, es igual que con la propiedad `padding`, así que la forma de escribir y los atajos es exactamente la misma. Por ejemplo, si queremos márgenes superior e inferior de 20 píxeles, y laterales de 5 píxeles:

```
margin: 20px 5px;
```

¡Truco! Para centrar un elemento de bloque, podemos hacer uso de auto:

```
margin: 0px auto;
```

Capas

Vamos a hablar de una etiqueta XHTML que nos quedó por ver y está estrechamente ligada con CSS: `<div>`. Esta etiqueta se encarga de crear una capa, que es un elemento de bloque que sirve de contenedor a otros elementos de bloque e `inline`.

¿Para qué nos sirven? Primero, para organizar semánticamente nuestra página. El atributo `id` tiene carga semántica, así que si queremos poner en la cabecera3 de nuestra web el título y el menú, haríamos esto:

¡Atencion! Cabecera en cuanto a maquetación. ¡No tiene nada que ver con la etiqueta `head`!

```
<div id="header">
```

```
<h1>Mi blog</h1>
```

```
<ul id="menu">
```

```
<li><a href="..." title="...">Principal</a></li>
```

```
<li><a href="..." title="...">Acerca de</a></li>
```

```
<li><a href="..." title="...">Enlaces</a></li>
```

```
</ul>
```

```
</div>
```

El otro uso de las capas es el de maquetar. Por ejemplo, el layout típico de un blog tiene cuatro capas: la cabecera, la del contenido principal, la de la barra lateral y la del pie de página. Mediante CSS, podemos controlar la disposición de estas capas, y así conseguir el diseño que queramos.

Floats

Los floats son probablemente una de las cosas que más cuesta dominar. En inglés los califican como “tricky”. Yo los califico como “pequeños cabroncetes”.

Lo que hacen los floats es alterar el flujo de la página, “incrustando” en él un elemento de bloque. El ejemplo más típico es el de si queremos poner una imagen acompañando a un texto, y que el texto “envuelva” a la imagen. Esto lo conseguimos creando una clase como la siguiente:

```
.izquierda {  
    float: left;  
}
```

Podemos indicar tanto `left (izquierda)` como `right (derecha)`. La mayoría de ellos suelen desaparecer mediante la propiedad `clear`, que se encarga de “anular” los floats. Los valores que admiten son `left`, `right` y `both` (que significa “ambos”).

Volviendo al ejemplo de layouts de blogs, lo normal es poner el contenido y la barra lateral mediante floats. Lo que suele pasar es que una de estas dos columnas pasa por encima del pie de página, en lugar de quedar el pie al final de todo. Esto se suele arreglar así:

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

Bibliografia

<http://www.cristalab.com/curso-html/modelo-caja/>