

Trucos CSS para no enloquecer

Todavía no tires el monitor contra la pared. Aquí los principales trucos CSS para hacer frente a los típicos problemas que se enfrentan los diseñadores web cuando maquetan con CSS. Podrán existir discrepancias entre los lectores, pero aclaro que estas son técnicas que a mi personalmente me han dado resultado, después de muchas pruebas e intentos aprendí esto...

Usa un contenedor global para todas las cajas (cuando las cosas se disparan)

De esta forma estas prefijando globalmente el orden de todas las demás cajas. En referencia a este contenedor ordena el resto de las cosas interiores. Es como si haces una cerca o valla para que nada es escape. Obviamente estamos hablando de sitios "fijos" no elásticos.

A veces es bueno usar un contenedor hasta el cuerpo del sitio, luego dejar el pié afuera.

Ejemplo para un contenedor de 900px centrado:

```
#contenedor {
margin-top: 0px;
margin-right: auto;
margin-bottom: 0px;
margin-left: auto;
width: 900px;
}
```

Que flote a la izquierda (cuando las cajas se superponen)

Esta es una muy buena forma de evitar incompatibilidades entre navegadores. El uso de hacks de CSS se debía en gran parte porque se trabajaba "centrando" las cajas. Si por ejemplo precisas poner tres cajas de 300px en un contenedor de 900px puedes hacer lo siguiente.

Ejemplo:

```
#caja {
float: left;
width: 300px;
}
```

Calcular bien los paddings o rellenos (cuando las cajas se van abajo)

Casi todos los dolores de cabeza y maldiciones echadas sobre el CSS se deben al mal uso o a la mala interpretación que se hace del padding. Pero es más simple de lo que parece.

¿Para que sirven los paddings o rellenos? Bueno, lo que hace es generar un "relleno" de determinada medida para dar por ejemplo como un margen a los elementos, pero lo hace sobre el ancho en píxeles que esté prefijado. Por ejemplo: si tenemos una caja de 300px y le aplicamos un padding de 10px en la izquierda, ahora tendremos una caja de 310px. Esto hará desbordar al resto de las cajas y las desplazarán para abajo. Ahí es cuando el diseñador principiante se vuelve loco. El tema es que si hay una diferencia de hasta un 1px se producirán estos desbordes, sino fíjate cuando le incluyes bordes a tu caja, se producirán diferencias.

Lo que se debe hacer es simple, calcular bien y recordar cada ajuste que se haga de los rellenos. Ahora tendremos que hacer una caja de 290px con paddings de 10px a la izquierda.

Ejemplo:

```
#caja {
float: left;
width: 290px;
padding-left: 10px;
background-color: #FFE6DD;
}
```

El pié de página con ancho fijo (cuando el pié de página enloquece):

Para entender mejor como funciona el uso de cajas en CSS se puede pensar en un grupo de objetos de diferentes formas que luchan por adaptarse y ocupar el espacio que se ha prefijado. Sucede muchas veces que los pié de página son los más problemas traen cuando se maqueta un sitio. O se va para arriba, se alinea a la izquierda, o se desborda, etc. Muchos resolvíamos este tema prefijando valores fijos a las alturas de cajas, pero no tiene sentido. Lo que se debe hacer es de nuevo establecer un valor de ancho fijo. De esta forma el pié se va a hacer su lugar del resto e irá a parar donde tiene que ir.

Ejemplo:

```
#pie {
width: 900px;
background-color: #666666;
}
```

No todo es 1+1=2 en CSS (cuando los anchos no cierran)

Un problema común en css es pensar que todos los anchos entre cajas cierran perfectamente. A veces es necesario jugar con los valores de los contenedores, a veces contrario a la lógica hay que añadir algunos px a los contenedores.

OTROS TRUCOS RÁPIDOS:

Trucos sencillos, de los que no hace falta explicar mucho pero que son muy prácticos y te harán más fácil el trabajo y evitarán posibles errores.

- Usa colores diferentes para distinguir las cajas
- Pon una palabra descriptiva en cada caja
- Comenta el código fuente y señala los finales de los contenedores grandes
- No mezquines espacios entre los divs
- No seas un fundamentalista y no quieras escribir tu CSS con dos o tres líneas. Si no quieres errores escribe lo necesario.
- Cuidado con el tamaño de las imágenes que insertas, estas cambian el ancho de los contenedores.
- Elige bien los nombres de cada div y trata de ser ordenado en el código.
- Si vas a trabajar con varias cajas, trata de agruparlas de a grupo, esto es muy importante. Por ejemplo un contenedor que agrupe tres o cuatro cajas.

CONCLUSIÓN:

Todos estos párrafos son simplemente algunas sugerencias o comentarios de lo que me ha dado resultado a mí. Existen otras muchas "ataaduras" de este tipo, si tienes alguna no dudes en comentarlas en este mismo artículo.

Que pasa cuando no puedes resolver un problema con [CSS](#) o similar? A mi me ha dado resultado levantarme un rato, hacer cualquier otra cosa y luego volver e intentar de nuevo.

Dejar de renegar y no enloquecer con [CSS](#) dependerá de la cantidad de tiempo, trabajo y esfuerzo que le metas a tu trabajo. No lo dudes.

Redactor: Leonardo A. Correa

Sitio Web: [Webnova - Recursos Webmaster](#)