

# 10 Buenas Prácticas Para Mejorar Su Código CSS

A veces el resultado de la codificación descuidada desde el principio del diseño, es causa de múltiples cambios de código y de consumo excesivo de tiempo. Escribir [código CSS](#) limpio, es simple cuando se empieza con el pie derecho, como consecuencia hacemos que el código sea más fácil de mantener y editar más tarde.

En este artículo te damos unos cuantos consejos para que afiances el proceso en el desarrollo del [código CSS](#), con esto podrás evitar problemas, y al contrario, crearas un archivo bastante robusto y entendible.

## 1. Sea Organizado

```

/* resets and overrides
----- */
* {
margin: 0;
padding: 0;
}
img { border: none; }

/* links & type
----- */
body {
font-family: "Lucida Sans", Verdana, sans-serif;
font-size: 100%;
}

a:link, a:active, a:visited { text-decoration: none; }
a:hover { color: #999; text-decoration: underline; }

h1 { }
h1 a { }
h1 a:hover { }

h2 { }
h2 a { }
h2 a:hover { }

/* layout
----- */
#page { }
#header { }
#col-alpha { }
#col-beta { }
#footer { }

/* callout boxes
----- */
#callout-alpha { }
#callout-beta { }

/* form elements
----- */
label { }
input { }
input.button { }

/* misc
----- */
.left { float: left; }
.right { float: right; }
.clear { clear: both; }

```

Regla de oro en el desarrollo, vale la pena mantener el código organizado. En lugar de escribir aleatoriamente las clases y los id's, es bueno tener un orden con esto. Esto le ayudará a mantener el concepto de cascada de CSS y establece su hoja de estilo para tomar ventaja de la herencia en estos archivos.

Declare sus artículos más genéricos primero, entonces siga con el que no es tan generico y así sucesivamente. Esto permite que su CSS herede correctamente los atributos y lo hace mucho más fácil para que usted elimine un estilo específico cuando sea necesario. Usted será más rápido en la edición de su CSS

más tarde porque va a seguir un formato fácil de leer, contara con una estructura lógica.

Utilice una estructura que funciona mejor para usted, manteniendo ediciones futuras y otros cambios futuros.

- Resets y overrides
- Enlaces y type
- Diseño principal
- Estructuras de diseño secundarias
- Elementos de formulario
- Demás diseño

---

## 2. Título, Fecha Y Signos

Deja que otros sepan quien escribió el código CSS, cuando se escribió y a quién contactar si tienen preguntas al respecto. Esto es especialmente útil en el diseño de plantillas o themes.

```
/* =====  
Subtle Shark  
Copyright 2009 WDD  
written for - www.WebDesignerDepot.com  
  
swatch colors  
-----  
13437a - dark blue  
1d74be - mid blue  
e1e1e1 - light gray  
a3a3a4 - mid gray  
8a8a8a - gray  
===== */
```

Espera un momento ... ¿qué es ese resumen acerca de los colores de muestras? La adición de una simple lista de colores comunes utilizados en as hojas de estilo es extremadamente útil durante el desarrollo inicial y a la hora de hacer cambios más adelante. Cuando necesite el código HTML para que el azul específico, simplemente puede ir hacia arriba y copiarlo.

---

## 3. Mantener Una Biblioteca De Plantillas

Una vez que se decide por una estructura



para usar, quite todo lo que no es genérico y guarde el archivo como una plantilla CSS para su uso posterior.

Puede guardar varias versiones para múltiples usos: un diseño de dos columnas, un diseño de blog, de impresión, móviles y así sucesivamente. Es una locura tener que volver a escribir cada una de sus hojas de estilo desde el principio, sobre todo cuando se está utilizando las mismas convenciones y metodologías en cada uno.

## 4 . Utilice Convenciones de nomenclatura de interés

Se dará cuenta de por encima de donde me declaré un par de columnas de id y les llamé **col-alpha** y **col-beta**. ¿Por qué no llamarlos **col-left** y **col-right**? Piense en futuras modificaciones, siempre.

Digamos, manejaos columnas en nuestro diseño, queremos mover la columna de la izquierda a la derecha, solo bastaría con cambiar la ubicación, pero tengamos en cuenta que el id, nos daría a entender que debería de estar en la izquierda.

Confuso, verdad?

Una de las principales ventajas del CSS es la capacidad de separar estilos de contenido. Puede rediseñar completamente su sitio con sólo modificar el CSS sin tener que tocar el código HTML . Así que no arruinar definitivamente el CSS mediante el uso de nombres de limitantes. Utilice convenciones de nomenclatura más versátiles y coherentes.

Deja posición o estilo palabras específicas de sus estilos y de identificación. Digamos, para una etiqueta A, manejar un clase `.link-blue`, con esto podemos saber que el color del vinculo será azul, y seguramente tendrá otras propiedades. Esta misma estrategia la puedes aplicar a otros tipos de elementos.

## 5. Guiones En Lugar De Under Lines

Para una mejor compatibilidad con versiones anteriores de los navegadores, es mejor entrar en el hábito de usar guiones en su lugar de los conocidos sub rayados ó under lines. Use `#col-alpha` en lugar de `#col_alpha`.

## 6. No Repitas Código

La re utilización de atributos es útil, siempre que sea posible mediante la agrupación de elementos en lugar de declarar los estilos de nuevo. Si su `h1` y `h2` utilizan el mismo tamaño de fuente, color y márgenes, agruparlos mediante una coma, justo de la siguiente manera:

```
h1, h2 {  
margin: 1em 0 2em 0;  
font-size: 1em;  
color: #222;  
}
```

much more efficient than this:

```
h1 {  
margin: 1em 0 2em 0;  
font-size: 1em;  
color: #222;  
}
```

```
h2 {  
margin: 1em 0 2em 0;  
font-size: 1em;  
color: #222;  
}
```

En esta imagen observamos dos bloques, uno en el cual tenemos agrupados dos elementos, h1 y h2. Estos elementos comparten los mismos atributos. Esto es mucho mas limpio que el bloque de abajo, el cual separa los dos elementos a pesar que comparten las mismas características.

---

Usted también debe hacer uso de los atributos en linea siempre que sea posible. Siempre estar en la búsqueda de oportunidades para los elementos del grupo y utilizar los métodos abreviados de declaración. En la siguiente imagen veremos algo de esto:

```
h1 {  
margin-top: 1em,  
margin-bottom: 2em,  
margin-left: 0,  
margin-right: 0,  
}
```

```
h1 {  
margin: 1em 0 2em 0,  
}
```

Como vemos, la imagen del lado izquierdo tiene una serie de configuración que se pueden acortar mediante uso de una linea, como el ejemplo que vemos en el lado derecho.

## 7. Optimizar Para Archivos CSS Ligeros

El uso de los consejos anteriores, hará que realmente pueda reducir el tamaño de sus hojas de estilo. Cargas más pequeñas más rápido y más pequeño es más fácil de mantener y actualizar. Elimine lo que no es necesario y consolide en lo posible mediante la agrupación. Tenga cuidado también al usar Frameworks CSS. Es probable que herede una gran cantidad de código que no se utilizará.

## 8. Escriba Su Base Para Gecko, Luego Ajustelo Para Webkit E IE

Sálvate a ti mismo, previene problemas de dolores de cabeza y escribe el código CSS primero para los navegadores **Gecko** (Firefox, Mozilla, Netscape, Flock, Camino). Si su CSS funciona correctamente con **Gecko**, es mucho más probable que no presente inconvenientes en Webkit (Safari, Chrome) e Internet Explorer.

## 9. Validar El Código

Haga uso del [validador gratuito de CSS de W3C](#). Si usted está atascado y su diseño no está haciendo lo que quiere que haga, el validador de CSS será de gran ayuda en señalar los errores.

## 10. Mantenga Una Casa Ordenada

Separe el código CSS específico para cada explorador web a su propia hoja de estilo individual, y uselas, según sea necesario con Javascript, código del lado del servidor o [comentarios condicionales](#). Utilice este método para evitar **hacks CSS** en tus hojas de estilo principales. Esto hará que tu CSS base sea limpio y manejable.