



Docs\_

Trucos y consejos



## 20 Trucos y Consejos CSS básicos que deberían conocer los diseñadores web

Una de las mejores formas de aprender y mejorar tus técnicas CSS es a través de pequeños y prácticos trucos y consejos. Si eres principiante o ya llevas un tiempo realizando trabajos de diseño, quizás estés esperando aprender algo que los mejore o que les de algo más de 'chispa'. De esa forma puedes conseguir que tus sitios parezcan un poco más profesionales y diferentes a la media.

Empecemos con algunos trucos que serán útiles para diseñadores principiantes y no tan principiantes pero que quieren mejorar sus técnicas:

Trucos y consejos CSS para diseñadores

### Usa reset.css

Cada navegador puede interpretar a su manera las etiquetas HTML, aunque sea a niveles mínimos. Además, también algunos usuarios asigna sus propias preferencias de navegación asignando colores o tipografías propias.

El 'reset CSS' lo que hace básicamente es poner a cero o 'resetear' todos los valores antes de que apliques tus propios estilos.

Un ejemplo de reset.css sería:

```
*{  
  
    margin: 0;  
  
    padding: 0;  
  
    list-style: none;  
  
    text-decoration: none;  
  
    border: none;  
  
    outline: none;  
  
}
```

De esta manera puedes resetear los estilos básicos y asegurarte de que comienzas con hojas de estilo realmente en blanco.

## Posición absoluta

Si quieres controlar dónde se sitúa exactamente cualquier elemento en el sitio web, necesitas posicionamiento absoluto.

Este código CSS configura la posición de un elemento a 40px de la parte superior y 50 de la parte derecha:

```
position: absolute;  
  
top: 40px;  
  
right: 50px;
```

## El asterisco (\*) junto a un selector

El asterisco (\*) te permite seleccionar todos los elementos de un selector concreto.

Por ejemplo, si utilizas \*h2 para añadir estilos CSS, lo estarás haciendo a todos los elementos en tu documento con una etiqueta <h2>.

## Utiliza atajos CSS

Los atajos CSS te da una forma más sencilla y corta de escribir tu código CSS, reduce el tamaño de los archivos CSS y hace que el código sea más sencillo y limpio.

Por ejemplo:

```
.p {background-color: #ffffff;
background-image: url(background.jpg);
background-repeat: repeat;
background-position: topleft;
}
```

Quedaría convertido en:

```
.p {background-color: #fff url(background.jpg) repeat top left}
```

## Consigue formas geométricas con CSS

No todas las formas en CSS tienen que ser cuadradas o rectangulares. Puedes aprovechar la propiedad CSS `shape-outside` para conseguir otro tipo de formas geométricas. Por ejemplo, puedes conseguir formas circulares o elipses para que el texto de un DIV se ajuste de esa manera alrededor de una imagen:

```
shape-outside: circle(50%);
shape-outside: ellipse(130px 140px at 20% 20%);
```

Incluso puedes conseguir otro tipo de formas basadas en imágenes externas:

```
shape-outside: url(/images/pelota.jpg);
```

## Entiende la diferencia entre Class e ID

Los dos selectores funcionan de forma diferente y es algo que suele confundir a los principiantes.

Básicamente, una clase (class) se representa con un punto (.) y puede reutilizarse varias veces.

Un ID se representa con una almohadilla (#) y es único, no puede repetirse ni reutilizarse.

Evita selectores CSS cuando sea posible

Intenta simplificar tu CSS evitando selectores cuando sea posible. Por ejemplo:

```
ul li { }  
table tr td { }
```

Puede simplificarse a:

```
li { }  
td { }
```

## Aprende el uso de !important

Un estilo marcado con !important será tenido en cuenta incluso si hay otra regla que lo machaque. Por ejemplo:

```
.p {background-color: white !important!;}
```

El código anterior hará que el color de fondo sea blanco, incluso si existen otras reglas que indiquen otros colores de fondo diferentes.

```
.page {color: black !important; color: blue;}
```

Aquí se tomará el color negro, aunque el azul se haya indicado después, ya que está marcado con **!important**.

Por otro lado, no es bueno abusar de !important, úsalo solo para sobrescribir en casos concretos, pero utilizarlo de forma general va a crearte un gran lío en tu código CSS.

## Cómo centrar horizontalmente con CSS

En ocasiones, centrar con CSS necesita de algún truco, porque depende de qué es lo que quieres que se centre.

### Centrar texto

Para centrar texto, utiliza **text-align: center**:

```
h1 { text-align: center; }
```

Este código CSS centraría horizontalmente las cabeceras. Igualmente puedes cambiar 'center' por 'right' o 'left' si quieres alinear a derecha o izquierda.

## Centrar contenido

Para centrar contenido, por ejemplo, para centrar un DIV que ocupa un 50% del ancho de la pantalla puedes utilizar márgenes automáticos:

```
#div {  
display: block;  
margin: auto;  
width: 50%;  
}
```

Este código centraría un DIV dentro de otro DIV:

```
<div id="fuera">  
  <div id="dentro">Foo foo</div>  
</div>  
  
#dentro {  
  width: 50%;  
  margin: 0 auto;  
}
```

## Alinear texto verticalmente en un menú

Alinear texto verticalmente con CSS puede ser tan sencillo como hacer coincidir la altura con el line-height:

```
.nav li{  
  line-height:75px;  
  height: 75px;  
}
```

## Aplica el mismo CSS a diferentes clases o selectores

Es posible que quieras añadir el mismo atributo CSS a diferentes elementos del código, aunque se trate de elementos que no están relacionados entre sí.

Puedes hacerlo de manera sencilla, ahorrando espacio y código. Por ejemplo, para añadir el mismo atributo de color a cabecera, barra lateral y pie de página:

```
.header, sidebar, footer {  
  color: #eee;  
}
```

## Comprueba que el navegador soporta tu CSS

No todos los navegadores admiten las mismas propiedades, y no todos lo hacen de la misma manera.

Una buena forma de asegurarte de que un navegador acepta el CSS que estás escribiendo es utilizando **@supports**. Por ejemplo:

```
@supports (display: grid) {  
  .main {  
    display: grid;  
  }  
}
```

Este código solo afectará a los navegadores que soporten display: grid.

## Cómo reemplazar texto con una imagen

Esta es una técnica que se utiliza sobre todo para substituir texto en cabeceras por imágenes. Por ejemplo, para substituir este título en la cabecera por una imagen:

```
<h1>Título de cabecera</h1>
```

Podríamos utilizar:

```
h1 {  
background:url("titulo-cabecera.jpg") no-repeat;  
text-indent:-9999px;  
width:400px;  
height:150px;  
}
```

## Efectos hover

Los efectos hover son los que ocurren cuando alguien pasa el puntero del ratón sobre un elemento en pantalla, y se consiguen tan fácil como añadir **:hover** de esta manera:

```
.h1 {  
color: red;  
font-size: 80px;  
}  
  
.h1:hover {  
color: black;  
}
```

El ejemplo anterior convierte el texto de cabecera h1 en rojo, pero al pasar el ratón por encima el texto se volvería negro.

Si quieres darle un toque de efecto extra al asunto, puedes utilizar algún efecto:

```
.h1:hover {  
color: black;  
transition: all 0.5s;  
}
```

Al añadir 'transition' lo que haces es que el efecto se produzca con una transición que dura los segundos que hayas indicado.

Añadir espacio de borde (padding) dentro de la caja – div

Este truco soluciona un problema con el que te puedes encontrar muy a menudo. Al añadir un borde (padding) a un div, este borde se va a sumar al exterior del div, quedando un espacio extra alrededor.

Es decir, si tienes un div con tamaño de 200px de ancho y 200px de alto y le añades un padding de 10px alrededor, al final el div va a ocuparte 210×200 px

La solución es utilizar **box-sizing: border-box;**

```
.div {  
width: 200px;  
height: 200px;  
padding: 10px;  
box-sizing: border-box;  
}
```

De esta manera, los 10px del borde quedarían incluidos dentro del tamaño del div, que no superaría los 200px de ancho y alto.



## Capitalizar la primera letra de un párrafo

Este truco CSS permite aplicar el efecto de capitalizar la primera letra de un párrafo. De esta manera, este primer carácter aparece desproporcionadamente grande, pero da un efecto elegante que llama la atención, similar a cómo comenzaban los textos de libros antiguos.

Para conseguir este efecto puedes aplicar este truco CSS:

```
p:first-letter{  
    display:block;  
    float:left;  
    margin:4px;  
    color:blue;  
    font-size:500%;  
}
```

Utilizando `:first-leter`, puedes hacer referencia al primer carácter de cualquier texto.

## Colores para los enlaces activos y visitados

Algo que suelen olvidar los diseñadores a la hora de generar su CSS son los colores de los enlaces. Por ejemplo, puedes indicar colores diferentes para los enlaces que el usuario haya visitado y los que no haya visitado. Un ejemplo claro lo tienes en las páginas de resultado de Google, donde los enlaces no visitados y ya visitados tienen colores indicativos diferentes:

```
a:link { color: blue; }  
  
a:visited { color: purple; }
```

## Usar :before

Este selector CSS te permite elegir un elemento CSS y añadirle un contenido delante cada vez que aparezca, también puede insertar un contenido.

Por ejemplo:

```
a:before {  
  content: "♥";  
  color: red;  
}
```

Este código añadiría un pequeño corazón de color rojo delante de cada enlace.

## Usar :after

De la misma manera que utilizas :before, también puedes utilizar :after para insertar contenido detrás de elementos concretos.

```
p:after {  
  content: "Seguir leyendo";  
  color: #red;  
}
```

De esta forma añadirías automáticamente el texto 'Seguir leyendo' detrás de cada párrafo, y este texto aparece en color rojo.

## Redimensiona las imágenes para que entren en su sitio

En ocasiones puedes encontrarte con que necesitas que una imagen se adapte a una anchura concreta mientras que la altura se ajusta proporcionalmente. Lo puedes hacer fácilmente así:

```
img {  
  max-width:100%;  
  height:auto;  
}
```

## Selectores de hijos

Los selectores de hijos te permiten seleccionar directamente todos los hijos descendientes de otro. Por ejemplo:

```
#header > a
```

Este código te permite seleccionar todos los enlaces situados dentro de la cabecera de una vez.

## Convertir todas las letras a mayúsculas o a minúsculas

En ocasiones vas a necesitar asegurarte de que todas las letras en un texto aparezcan como mayúsculas o como minúsculas. Esto puede ser particularmente útil en encabezados, para mostrar nombres de usuario, etc:

```
h1 {  
  text-transform: uppercase;  
}  
  
Convierte todas las letras a mayúsculas.  
  
h1 {  
  text-transform: lowercase;  
}
```

## Convierte todas las letras a minúsculas.

```
h1 {  
  text-transform: capitalize;  
}
```

Convierte a mayúsculas la primera letra de cada palabra.