



2020/04/13

Un CSS Reset moderno

Traducción de [A Modern CSS Reset](#), por [Andy Bell](#).

Pienso y disfruto sobre asuntos de CSS muy aburridos, probablemente mucho más de lo que debería, para ser honesto. Una de las cosas en la que probablemente he pasado demasiado tiempo pensando a lo largo de los años, es en los [CSS resets](#).

En esta época moderna del desarrollo web, realmente no necesitamos un reset tan fuerte, o incluso ningún reset en absoluto, ya que las incompatibilidades CSS de los navegadores son mucho menos comunes de lo que lo era en los viejos tiempos de IE6. Aquella época, en la que resets como [normalize.css](#) surgieron y nos salvaron de aquel infierno. Esos días quedaron atrás y podemos confiar en que nuestros navegadores tengan un mejor comportamiento, por lo que creo que resets como aquellos son, probablemente, redundantes en su mayoría.

Un reset de valores predeterminados razonables

Me sigue gustando aplicar un reset, así que he estado retocando un CSS reset propio, a lo largo de los años, de manera obsesiva. Voy a explicar qué contiene y por qué, pero antes e hacerlo, aquí está en su totalidad:

```
/* Box sizing rules */
*,
*::before,
*::after {
  box-sizing: border-box;
}

/* Remove default padding */
ul[class],
ol[class] {
  padding: 0;
}

/* Remove default margin */
body,
h1,
h2,
h3,
```

```
h4,
p,
ul[class],
ol[class],
li,
figure,
figcaption,
blockquote,
dl,
dd {
  margin: 0;
}

/* Set core body defaults */
body {
  min-height: 100vh;
  scroll-behavior: smooth;
  text-rendering: optimizeSpeed;
  line-height: 1.5;
}

/* Remove list styles on ul, ol elements with a class attribute */
ul[class],
ol[class] {
  list-style: none;
}

/* A elements that don't have a class get default styles */
a:not([class]) {
  text-decoration-skip-ink: auto;
}

/* Make images easier to work with */
img {
  max-width: 100%;
  display: block;
}

/* Natural flow and rhythm in articles by default */
article > * + * {
  margin-top: 1em;
}

/* Inherit fonts for inputs and buttons */
input,
button,
textarea,
select {
  font: inherit;
}
```

```
}

/* Remove all animations and transitions for people that prefer not to see
@media (prefers-reduced-motion: reduce) {
  * {
    animation-duration: 0.01ms !important;
    animation-iteration-count: 1 !important;
    transition-duration: 0.01ms !important;
    scroll-behavior: auto !important;
  }
}
```

Descomponiéndolo por partes

Empezamos con box-sizing. Simplemente reinicio todos los elementos y pseudo-elementos para que usen `box-sizing: border-box`.

```
*,
*::before,
*::after {
  box-sizing: border-box;
}
```

Algunas personas piensan que los pseudo-elementos deberían heredar el box-sizing, pero creo que es absurdo. Si quieres utilizar un valor diferente de box-sizing, configúralo explícitamente—al menos es lo que yo hago. Escribí más acerca de box-sizing en [CSS From Scratch](https://super-simple.net/blog/un-css-reset-moderno/).

```
/* Remove default padding */
ul[class],
ol[class] {
  padding: 0;
}

/* Remove default margin */
body,
h1,
h2,
h3,
h4,
p,
ul[class],
ol[class],
li,
```

```
figure,  
figcaption,  
blockquote,  
dl,  
dd {  
    margin: 0;  
}
```

Tras box-sizing, hago un reinicio general de `margin` y `padding`, de los estilos por defecto del navegador. Esto es bastante auto-explicativo, así que no profundizaré en ello demasiado.

Sin embargo, mencionaré un caso particular con los listados. Selecciono sólo los listados que tengan un atributo `class` porque si se usa un simple `` o ``, quiero que tengan estilos vean como de listado. Muchos resets, incluyendo previos míos, eliminan estos estilos de manera agresiva.

```
body {  
    min-height: 100vh;  
    scroll-behavior: smooth;  
    text-rendering: optimizeSpeed;  
    line-height: 1.5;  
}
```

A continuación: los estilos de `<body>`. Mantengo esto realmente simple. Es útil que el `<body>` ocupe todo el alto de la ventana, aunque esté vacío, lo que hago estableciendo `min-height` a `100vh`. También me gusta que el scroll sea fluido, así también que utilizo `scroll-behaviour: smooth`.

Sólo quiero establecer dos estilos de texto. Establezco `line-height` a `1.5` porque el valor por defecto `1.2` no es suficiente para obtener un texto legible y accesible. También establezco `text-rendering` a `optimizeSpeed`. Usar `optimizeLegibility` hace que el texto se vea más bonito, pero tiene serios problemas de rendimiento, como retrasos de carga de 30 segundos, por lo que trato de evitarlo. Sin embargo, a veces lo añado a secciones con poco texto.

```
ul[class],  
ol[class] {  
    list-style: none;  
}
```

Tal como en las reglas para `margin` y `padding`, tan sólo reinicio `list-style` si el listado tiene un atributo `class`.

```
a:not([class]) {  
  text-decoration-skip-ink: auto;  
}
```

Para los enlaces que tengan atributo `class`, establezco `text-decoration-skip-ink: auto` para que el subrayado se renderize de un modo mucho más legible. Esto podría establecerse en los enlaces a nivel global, pero me ha causado varios conflictos en el pasado, por lo que lo mantengo así.

```
img {  
  max-width: 100%;  
  display: block;  
}
```

A continuación viene las imágenes fluidas. Establezco las imágenes para que sean un elemento en bloque porque, francamente, la vida es demasiado corta para ese margen extraño que aparece debajo. Además, las imágenes—especialmente en mi trabajo—suelen comportarse como bloques.

```
article > * + * {  
  margin-top: 1em;  
}
```

Me encanta este truco de CSS y por fin he sido tan valiente como para incluirlo en el reset. El selector del buho lobotomizado apunta a los descendientes directos de un artículo y les añade `1em` de margen superior. Esto le da un ritmo sólido al flujo de contenido. En realidad, ahora uso una utilidad `.flow` en cada proyecto. Puedes [leer más al respecto en 24 ways](#). De hecho, reconozco que es el CSS que más uso hoy en día.

```
input,  
button,  
textarea,  
select {  
  font: inherit;  
}
```

Otra cosa que por fin he sido tan valiente como para incluirla por defecto es `font: inherit` en campos de formulario, que hace exactamente lo que explica [este artículo](#). ¡No más texto minúsculo!

```
@media (prefers-reduced-motion: reduce) {  
  * {  
    animation-duration: 0.01ms !important;  
    animation-iteration-count: 1 !important;  
    transition-duration: 0.01ms !important;  
    scroll-behavior: auto !important;  
  }  
}
```

Por último, pero no menos importante, un `@media` query que reinicia animaciones, transiciones y comportamiento de scroll si [el usuario prefiere un movimiento reducido](#). Me gusta tener esto en el reset, con `!important`, porque si un usuario no quiere movimiento, no lo querrá sin importar el CSS que pueda haber tras este reset.

Actualización: Gracias a [@atomiks](#), esto [ha sido actualizado](#) de manera que no rompe los eventos de JavaScript que observan `animationend` y `transitionend`.

Para terminar

En conclusión, un reset minúsculo que hace mi vida mucho más fácil. Si te gusta, ¡también puedes usarlo! Puedes encontrarlo en [GitHub](#) o [NPM](#).

[+ VER MÁS ARTÍCULOS](#)

Créditos: [Font Awesome](#)