Writing Projects RSS Micro Blog Twitter CodePen GitHub

A Modern CSS Reset

1st October 2019 — 9 minute read

JP *If you would prefer to read this article in Japanese, head over here, where コリス has very kindly translated it for me.*

RU *If you would prefer to read this article in Russian, head over here, where Stas has very kindly translated it for me.*

ES *If you would prefer to read this article in Spanish, head over here, where superSimple has very kindly translated it for me.*

I think about and enjoy very boring CSS stuff—probably much more than I should do, to be honest. One thing that I've probably spent too much time thinking about over the years, is CSS resets.

In this modern era of web development, we don't really need a heavy-handed reset, or even a reset at all, because CSS browser compatibility issues are much less likely than they were in the old IE 6 days. That era was when resets such as normalize.css came about and saved us all heaps of hell. Those days are gone now and we can trust our browsers to behave more, so I think resets like that are probably mostly redundant.

A reset of sensible defaults

I still like to reset stuff, so I've been slowly and continually tinkering with a reset myself over the years in an obsessive code golf manner. I'll explain what's in there and why, but before I do that, here it is in its entirety:

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

Breaking it down

We start with box-sizing. I just flat out reset all elements and pseudo-elements to use `box-sizing: border-box`.

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

Some people think that pseudo-elements should `inherit` box sizing, but I think that's silly. If you want to use a different box-sizing value, set it explicitly—at least that's what I do, anyway. I wrote about box-sizing more over on CSS From Scratch.

```
/* 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;
}
```

After box-sizing, I do a blanket reset of `margin` and `padding`, where it gets set by the browser styles. This is all pretty self-explanatory, so I won’t get into it too much.

I *will* mention the situation with lists, though. I select only lists that **do** have a `class` attribute because if a plain ol’ `<ul>` or `<ol>` gets used, I want it to look like a list. A lot of resets, including my previous ones, aggressively remove that.

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

Next up: body styles. I keep this really simple. It’s useful for the `<body>` to fill the viewport, even when empty, so I do that by setting the `min-height` to `100vh`. I also like smooth anchor scrolling, so I set `scroll-behavior: smooth`, too.

I only set two text styles. I set the `line-height` to be `1.5` because the default `1.2` just isn’t big enough to have accessible, readable text. I also set `text-rendering` to `optimizeSpeed`. Using `optimizeLegibility` makes your text look nicer, but can have serious performance issues such as 30 second loading delays, so I try to avoid that now. I do sometimes add it to sections of microcopy though.

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

Just like the margin and padding rules, I only reset `list-style` where a list element has a `class` attribute.

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

For links without a class attribute, I set `text-decoration-skip-ink: auto` so that the underline renders in a much more readable fashion. This could be set on links globally, but it’s caused one or two conflicts in the past for me, so I keep it like this.

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

Good ol’ fluid image styles come next. I set images to be a block element because frankly, life is too short for that weird gap you get at the bottom, and realistically, images—especially with work I do—tend to behave like blocks.

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

I *really* like this CSS trick and I’ve finally been brave enough to add it to the reset. The lobotomised owl selector targets direct descendants of an article and adds `1em` of top margin to them. This gives a solid rhythm to flow content. I actually use a `.flow` utility in every project now. You can read more about it on 24 Ways. In fact, I reckon it’s my most used CSS these days.

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

Another thing I’ve finally been brave enough to set as default is `font: inherit` on input elements, which as a shorthand, does exactly what it says on the tin. No more tiny (mono, in some cases) text!

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

Last, and by no means least, is a single `@media` query that resets animations, transitions and scroll behaviour if the user prefers reduced motion. I like this in the reset, with specificity trumping `!important` selectors, because most likely now, if a user doesn’t want motion, they won’t get it, regardless of the CSS that follows this reset.

🛈 **Update**: *Thanks to @atomiks, this has been updated so it doesn’t break JavaScript events watching for* `animationend` *and* `transitionend`*.*

Wrapping up

That’s it, a very tiny reset that makes my life a lot easier. If you like it, you can use it yourself, too! You can find it on GitHub or NPM.

Hi 👋, I’m Andy — an educator and web designer

I produce front-end development tutorials over at Front-End Challenges Club and Piccalilli. You can sign up for updates on Piccalilli to stay up to date with its progress.

Sign up for updates

Filed under writing

Writing Projects RSS Micro Blog Twitter CodePen GitHub

Dark theme

Powered by my Eleventy Starter Kit, Hylia — Go back to the top.