sitepoint premium

Handout

Principals of CSS Architecture
Lesson 1 - Step 2
Zsolt Nagy

In this section we're going to examine the goals of **CSS architecture**. Before that, though, we should figure out what good CSS looks like.

As you know, CSS serves the purpose of replicating visual mockup that stakeholders, designers, UX interaction designers and so on imagined. A CSS developer should not only use the latest technologies but should also be aware of best practices in the CSS world and preferably should also be able to interact with front end developers.

The question is, though, what makes our code maintainable in general? The latest technologies, the latest tricks, knowing the ins and outs of software engineering? Let's figure it out. If you have ever inherited bad CSS, for example a WordPress site done by a freelancer who was not up-to-date with the latest technologies, especially in the CSS world, then you know what maintainable CSS doesn't look like.

Creating CSS and maintaining it in general are two completely separate things. Therefore whenever you think about CSS architecture, or writing maintainable CSS code in general, always think about all the people that will maintain your CSS code in the future. Creating maintainable CSS in general is an engineering activity.

Quoting the definition of **engineering** of the Carnegie-Mellon University, is the following, engineering is about creating cost effective solutions to practical problems by applying scientific knowledge, building things in the service of mankind. And engineering also has this bulletproof nature to it, that we should not just create something once but whatever we create an as engineering activity should also last for the future. Software engineering is a profession. And it's very important to execute this profession in a manner that we create things that last.

According to Nicholas C. Zakas maintainable code is intuitive, understandable, extendable, debuggable, and testable.

Before continuing let me introduce two abbreviations:

- DRY - don't repeat yourself
- WET - we enjoy typing. Typing doesn't really have to be typing: it can also be copy pasting. It's very important that whenever you recognize two or more occurrences of the same concept in your CSS code It is worth abstracting the concept to something more abstract, basically, a super-class, or even a mix-in.

Another important aspect of writing maintainable CSS is scalability. Today's applications are larger than ever before. I have personally maintained an application that the JavaScript code was two and half megabytes long and it's not the biggest application that I have seen personally. Same with the CSS. We have component libraries, CSS frameworks, resets and normalizers. And if you want to maintain all these bits and pieces together, it requires team effort. In order to provide the good, scalable structure, what we need is architecture.

It is also very important that you should not only consider system goals, but you should also consider the user and the business. Without user experience, it doesn't really make sense to write an application, and without solving the needs of a business, what is the point in coding?

Therefore, whenever you deal with CSS architecture, there is a user aspect and there is a business aspect as well.

As you have just learned, creating good CSS is an engineering activity. When thinking about engineering activities, think about your local dentist, think about the flights that you can enjoy because of all the engineers creating good quality airplanes. Or think about any service that you use online or offline.

Whenever it come to relatively big projects, maintaining CSS is a team effort. It is not necessarily an individual effort. Although it is possible on some level to write some smaller applications maintained by just one person. But even then, there is going to be the front end developer, the back end developer, the API team, the stakeholders, and so on.

Even then, we're not talking about one person, but a corporation of multiple people creating a product. CSS should not only satisfy our system requirements in general, but it should also consider the needs of user. From our perspective, good CSS architecture, from the system requirements point of view, addresses maintainability, reusability, predictability, and scalability.