



We will conclude this lesson with the introduction of an example CSS architecture called **ITCSS**. This introduction was included here so that you can get an overview of how a CSS architecture can encourage you to write CSS according to the principles of software engineering. ITCSS is backed by CSS Wizardry, Harry Roberts.

ITCSS is referred to as scalable, maintainable CSS architecture. There is little documentation available on it to date but the sources are straightforward enough to put the principles into action. This is exactly what we will do in lesson five, where we are going to use the ITCSS architecture to create a small form.

IT in ITCSS stands for "inverted triangle". This triangle represents layers of your CSS code. These layers build on top of each other and make you organize your files from generic to specific. The triangle contains seven layers:

- The Settings layer contains the processor setup, fonts, your corporate identity colors, and definitions. You will define all your global variables in Settings. Furthermore, some configuration setup will also be placed here.
- Tools contain functions and mixins that you use in the lower layers. In these two layers, no CSS is generated.
- Generic includes reset styles and normalizers. This is our first CSS layer. We will cover resets, normalizers in depth in lesson four.
- Base elements describe generic behavior of HTML elements. This layer may contain tag names, this is not a violation of separation of concerns. We include these ties so that we come up with the default base tie for our elements.
- In Objects, we place all classes that we define. It doesn't matter if you use the block element modifier notation, or you define class names in the way how you read it in your SMACSS book, or in your OOCSS reference description.
- Components contains styles for UI components. The components layer may use object definitions.
- Trumps or utilities define helper classes to describe specific states. It is worth mentioning that ITCSS doesn't depend on the used methodology or naming convention. Feel free to use block element modifier, feel free to use it with SMACSS or OOCSS. As I mentioned before, in ITCSS, we go from generic to specific.

In the next section, we will discover the different layers of the hierarchy a bit more in depth.

Let's have a look at InuitCSS. Inuitcss is an OOCSS framework based on ITCSS. You can recognize the ITCSS hierarchy in it, you can build your own stylesheet hierarchy based on the framework. You can easily install InuitCSS via NPM and Yarn and get started building your style hierarchy. Note, though, that even the GitHub readme file encourages InuitCSS to be used in case of large, long-lasting projects, and not small projects.

This wraps up lesson three. Notice that we are going from low level towards higher level of abstraction. In lesson two, we discovered some low-level elements of CSS architecture such as class naming, the block element modifier syntax using units for better maintainability and selector efficiency.