Chapter 3. A gentle introduction to ECMASCRIPT 6

If you're reading this, we can be pretty sure you have heard of JavaScript. What we call JS is one implementation of a standard specification, called ECMAScript. The spec version you know the most about is the version 5, that has been used these last years.

But recently, a new version of the spec has been in the works, called ECMASCRIPT 6, ES6, or ECMASCRIPT 2015. From now on, I'll mainly say ES6, as it is the most popular way to reference it. It adds A LOT of things to JavaScript, like classes, constants, arrow functions, generators... It has so much stuff that we can't go through all of it, as it would take the whole book. But Angular has been designed to take advantage of the brand new version of JavaScript. And, even if you can still use your old JavaScript, things will be more awesome if you use ES6. So we're going to spend some time in this chapter to get a grip on what ES6 is, and what will be useful to us when building an Angular app.

That means we're going to leave a lot of stuff aside, and we won't be exhaustive on the rest, but it will be a great starting point. If you already know ES6, you can skip these pages. And if you don't, you will learn some pretty amazing things that will be useful to you even if you end up not using Angular in the future!

3.1. Transpilers

ES6 has just reached its final state, so it's not yet fully supported by every browser. And, of course, some browsers will always be late to this game (even if, for once, Microsoft is doing a good job with Edge). You might be thinking: what's the use of all this, if I need to be careful on what I can use? And you'd be right, because there aren't that many apps that can afford to ignore older browsers. But, since virtually every JS developer who has tried ES6 wants to write ES6 apps, the community has found a solution: a transpiler.

A transpiler takes ES6 source code and generates ES5 code that can run in every browser. It even generates the source map files, which allows to debug directly the ES6 source code from the browser. At the time of writing, there are two main alternatives to transpile ES6 code:

- Traceur, a Google project
- Babeljs, a project started by a young developer, Sebastian McKenzie (17 years old at the time, yeah, that hurts me too), with a lot of diverse contributions.

Each has its own pros and cons. For example, Babeljs produces a more readable source code than Traceur. But Traceur is a Google project, so, of course, Angular and Traceur play well together. The source code of Angular itself was at first transpiled with Traceur, before switching to TypeScript. TypeScript is an open source language developed by Microsoft. It's a typed superset of JavaScript that compiles to plain JavaScript, but we'll dive into it very soon.

Let's be honest Babel has waaaay more steam than Traceur, so I would advice you to use it. It is quickly becoming the de-facto standard in this area.