

Prototypal Inheritance

prototype-based inheritance and the introduction to classes in ES6

The concepts of prototypes and prototypal inheritance in ES5 are hard to understand for many developers transitioning from another programming language to JavaScript.

ES6 classes introduce *syntactic sugar* to make prototypes look like classical inheritance.

For this reason, some people applaud classes, as it makes JavaScript appear more familiar to them. Others seem to have launched a holy war against classes, claiming that the class syntax is flawed.

On some level, all opinions have merit. My advice to you is that the market is always right. Knowing classes gives you the advantage that you can maintain code written in the class syntax. It does not mean that you have to use it. If your judgment justifies that classes should be used, go for it.

I use classes on a regular basis, and my [React-Redux tutorials](#) also make use of the class syntax.

Not knowing the class syntax is a disadvantage.

Judgment on the class syntax, or offering alternatives are beyond the scope of this section. In the next lesson, I will talk about prototypal inheritance in more detail.