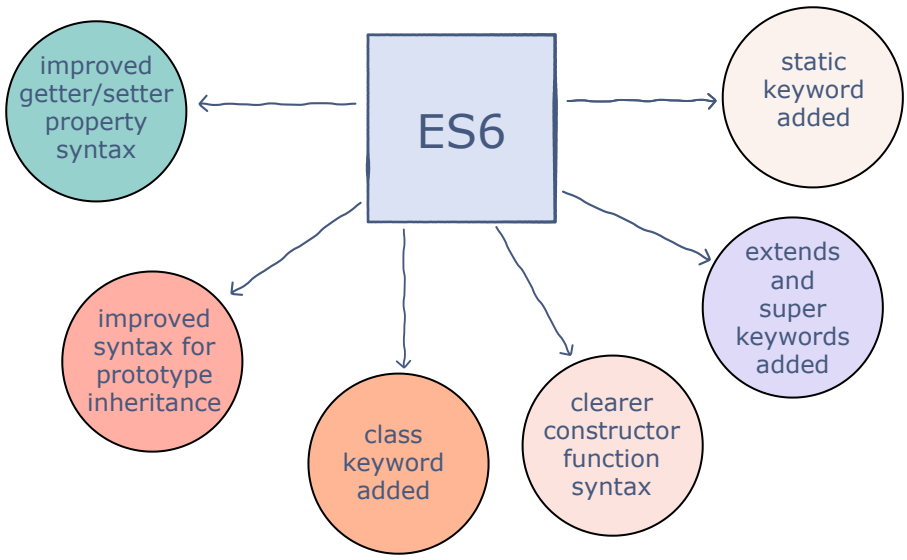
ES6 vs ES5

This lesson goes over the new features as well as the improvements made in existing features in the ES6 version of JavaScript.

WE'LL COVER THE FOLLOWING

What does ES6 offer?

It provides new features as well as a cleaner approach for some of the existing features in ES5.



Features in ES6

Explanation

`class` keyword

In the ES5 version, there are no classes; a *function* is used to make an object directly. However, the ES6 version uses the keyword `class` to define a class. The underlying concept is more or less the same. ES6 just cleans up the syntax. This will be discussed in detail in upcoming chapters.

getter/setter syntax

The object properties require get/set methods to be called on them to access/modify their values. In the ES5 version, they were not widely used because the syntax was not that clean, thus making them challenging to use. However, the ES6 version provides an improved, easy to use, and clear syntax for Get and Set methods.

constructor function syntax

In the ES5 version, constructor functions are declared using the `function` keyword, with the body of the code initializing the object properties upon its creation. The ES6 version, on the other hand, uses the `constructor` keyword to declare the constructor function which runs on object creation. This new syntax is clearer, hence making it easier to use.

`extends` keyword

In the ES5 version, *prototypal inheritance* is not an easy task. The code is hard to follow and takes a lot of effort to write as well as understand. The ES6 version offers an improved and cleaner syntax by using the *keyword* `extends` for setting up the inheritance relationship between parent and child.

`super` keyword

This is a new feature introduced in the ES6 version. `super` is used to call the constructor on the parent object that is being inherited by the child. It is used to avoid duplication of the parts of the constructor that are present in both the parent and child class.

`static` keyword

JavaScript did allow for static members to be declared in the ES5 version.

However, the ES6 version formalizes this by introducing the *keyword* `static`.

Now that you have had a basic introduction to OOP in JavaScript and the ES6 version, let's test your knowledge in the next lesson!