**What major new features were in each JavaScript version? What version should I target?**

[Apr 28, 2018](https://medium.com/@dupski/what-major-new-features-were-in-each-javascript-version-what-version-should-i-target-25526c498687?source=post_page-----25526c498687----------------------) · 2 min read

So I was starting a new TypeScript project and wondering what JavaScript / ECMA Script version to target. We can currently choose from ES5, ES2015, ES2016, ES2017 and ES2018!

The two major things to consider when selecting your JavaScript target version are:

* Target Environment — Desktop Browser? Mobile? NodeJS?
* What JS features do I need — Classes? Modules? Async Functions?

We are fortunate to be in the age of the Evergreen Browser, so support for the latest features is very good in most environments, however there are always situations when users have older devices, or you are stuck on an older version of NodeJS.

For this reason, my general recommendation is to **target the oldest version that has the features you need**. This will give a good balance between wide target environment support and access to awesome new features :)

**ES Version Major Features**

Below I have summarised what I see as the major new features released in each ECMA Script version. For a comprehensive list, as well as full details of target environment support, check out Kangax’s awesome [compatibility table](http://kangax.github.io/compat-table/es2016plus/).

**ES5 (2009)**

* This is the baseline version of JS which you can generally assume all run-times (except really old ones!) will support.

**ES6 / ES2015**

1. Standard Modules — import and export
2. Standardised Promises
3. Classes & Inheritance
4. Block-scoped variables — letand const
5. Template Literals
6. Object destructing into variables
7. Generator functions
8. Map and Set data structures
9. Internationalisation for Strings, Numbers and Dates via Intl API
10. new function syntax using Arrow functions,
11. creation of Classes,
12. parameters with default values,
13. promises for asynchronous actions,

**ES7 / ES2016**

1. Array.includes()
2. Numeric exponent (power of) operator \*\*

**ES8 / ES2017**

1. Async Functions
2. Object.entries
3. String padding functions

**ES9 / ES2018**

* Object Rest/Spread const obj = { ...props };
* Asynchronous Iteration for await (...) {
* Promise finally() function
* Regular expression enhancements (lookbehind, named groups)

More Details:

<https://www.greycampus.com/blog/programming/java-script-versions>

