# JavaScript — WTF is ES6, ES8, ES 2017, ECMAScript… ?

## Learn JavaScript and ECMAScript history and naming conventions is this JavaScript Quickie!

Brandon Morelli Follow
Aug 28, 2017 · 3 min read

**Learning JavaScript can be confusing for a number of reasons. Don't let these acronyms confuse you any further. Let's break down the ES abbreviations in this JavaScript Quickie!**

### WTF is ECMAScript?

*Here's what happened long, long ago:*

1. JavaScript was originally named JavaScript in hopes of capitalizing on the success of Java.
2. Netscape then submitted JavaScript to ECMA International for Standardization. (ECMA is an organization that standardizes information)
3. This results in a new language standard, known as **ECMAScript**.

Put simply, **ECMAScript** is a standard. While **JavaScript** is the most popular *implementation* of that standard. JavaScript implements ECMAScript and builds on top of it.

## Okay, so 'ES'...?

**ES** is simply short for **ECMAScript**. Every time you see **ES** followed by a number, it is referencing an edition of ECMAScript. In fact, there are eight editions of ECMAScript published. Lets dive into them:

## ES1, ES2, ES3, ES4

***ES1**: June 1997—**ES2**: June 1998—**ES3**: Dec. 1999—**ES4**: Abandoned*

I've grouped all of these together. These were the first 4 editions of **ECMAScript**, and to save time, we wont go too in-depth. Just know that the first three editions were annual, and the fourth was abandoned due to political differences.

## ES5

*December 2009:* Nearly 10 years later, **ES5** was released in 2009. It would then take almost six years for the next version of **ECMAScript** to be released.

## ES6 / ES2015

*June 2015*: Perhaps the cause for all of your confusion begins here. You see, **ES6 and ES2015 are the same thing**.

**ES6** was the popularized name prior to release. However, the committee that oversees ECMAScript specifications made the decision to move to annual updates. With this change, the edition was renamed to **ES 2015** to reflect the year of release**.** Subsequent releases will therefor also be named according to the year they are released.

## ES2016 (ES7)

*June 2016*: Seventh edition of ECMAScript.

## ES2017 (ES8)

*June 2017*: Eighth edition of ECMAScript.

## ES.Next

You may have also seen ES.Next used online. This term is dynamic and references the next version of ECMAScript coming out.

## Why?

Each release brings updates and new features to the language.

. . .

## Key Takeaways:

- An update to ECMAscript can be expected annually.
- Initial Editions of ECMAScript are named numerically, increasing by 1: **ES1, ES2, ES3, ES4, ES5**
- New editions (starting with 2015) will be named ES followed by the year of release: **ES2015, ES2016, ES2017**
- **ECMAScript** is a standard. **JavaScript** is the most popular *implementation* of that standard. Other implementations include: SpiderMonkey, V8, and ActionScript.

Hopefully the next time you see ES in the wild you'll better understand what it's referencing!

I publish a few articles and tutorials each week, please consider entering your email here if you'd like to be added to my once-weekly email list.

## Want more advanced JavaScript?

Check out: **A Beginners Guide to Advanced JavaScript**

. . .

## If this post was helpful, please click the clap 👏 button below a few times to show your support! ⬇⬇