

# ES2015 Introduction

## An Introduction

- Let's see some newer JS features!
- Everything covered here is part of ES2015 (what is that??)
- Some of these features aren't supported in IE

## Before we start.....A brief history

- Browser wars of the 90s
- JS Chaos, standardization necessary
- Enter ECMA!

## What is ECMAScript?

- The “standard” for the JS language!
- Started during the browser wars between Microsoft and Netscape
- Browsers/Node implement the language based on this standard

## What is ECMA?

- standards organization for information and communication systems.
- Who makes the decisions?

## Introducing TC39

- Technical Committee 39 maintains and updates the ECMAScript standard/
- They are “in charge” of adding new features
- They evaluate proposals for complementary or additional technologies.

## Evolving JS - The old way

- New features were batched together into large releases
- These releases were named with version numbers (ES3, ES5, etc.)

- Versions released infrequently, often with years between

## **Evolving JS - The new way!**

- Updates to the standard are published yearly
- Any new features ready each year are added together
- Versions named by year (ES2016, ES2017, etc.)
- Introduced a new, formalized proposal process

## **Stages of TC39**

### **Stage 0: Strawman**

- It all starts with an idea!
- Non-formal proposals live here

Any discussion, idea, change, or addition which has not yet been submitted as a formal proposal is considered to be a “strawman” proposal at this stage. Only members of TC39 can create these proposals, and there’s over a dozen active strawman proposals today.

### **Stage 1: Proposal**

- It’s an “official” proposal
- Polyfills / plugins will be introduced usually at this time

Very unlikely that implementers are working on proposals

### **Stage 2: Draft**

- Still a bit too early for us to cover!
- Sometimes implementers will experiment adding this to browsers/environments

### **Stage 3: Candidate**

- Proposals in this stage are candidate recommendations.
- Unlikely to change beyond fixes to issues identified in the wild.

- Proposals move to this level with at least one browser implementation or when supported by a build-time transpiler like Babel.

## **Stage 4: Finished**

Proposals get to this stage when there are at least two independent implementations that pass acceptance tests.

## **Everything in this unit is Stage 4**

And has been implemented by almost all modern browsers

Let's get started!