

Transpiling



Cory House

@housecor

reactjsconsulting.com



Here's the plan



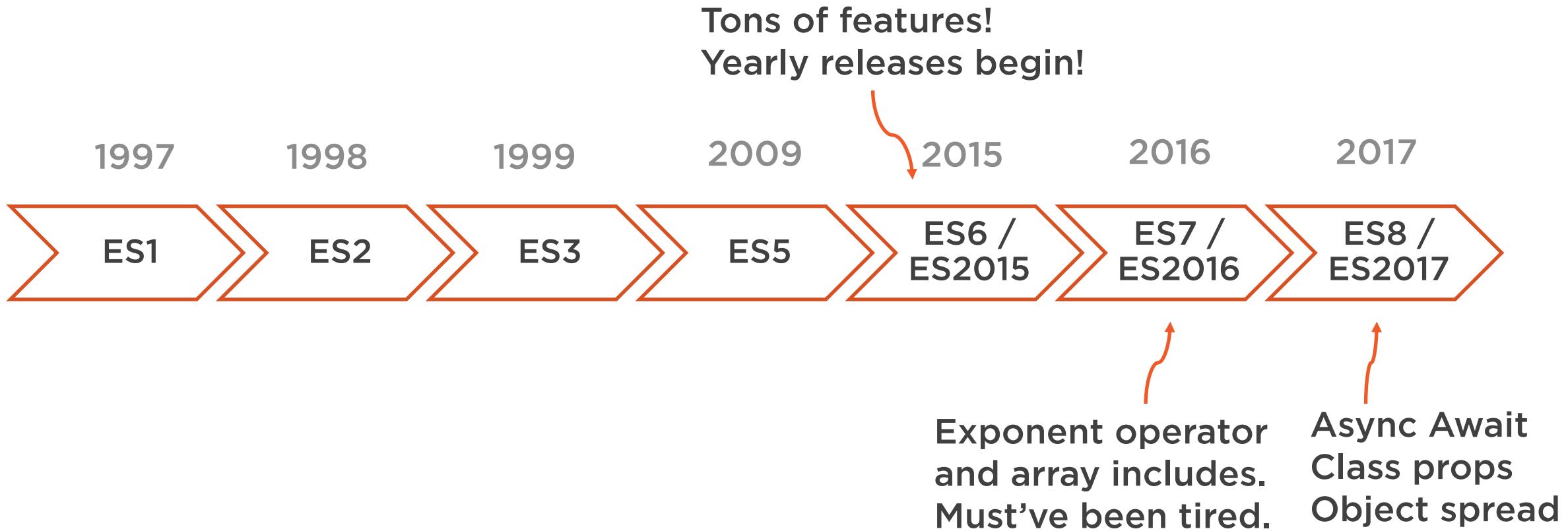
Why transpile? - History and future

Transpilers

Set up Babel



ECMAScript Versions



Choosing a Transpiler



https://github.com/jashkenas/coffeescript/wiki/list-of-languages-that-compile-to-js

Inbox (11) - housecor@gmail.c... Loader cannot parse jsx · Issu... samccone/The-cost-of-transp... Icon Library Search | Pluralsig... Learn ES2015 · Babel List of languages that compile... typescript – Google Search (1) Twitter / Notifications

This repository Search Pull requests Issues Gist Watch 591 Star 13,181 Fork 1,798

Code Issues 310 Pull requests 37 Wiki Pulse Graphs

jashkenas / coffeescript

Watch 591

Star 13,181

Fork 1,798

List of languages that compile to JS

Pierre Quentel edited this page 3 days ago · 530 revisions

Edit New Page

CoffeeScript Family (& Friends)

- [CoffeeScript Unfancy JavaScript](#)
stars 13k forks 1798 issues 347 open
- [CoffeeScript II: The Wrath of Khan](#) Rewrite of the CS compiler
stars 2k forks 118 issues 106 open

Family (share genes with CoffeeScript)

- [Coco](#) A CoffeeScript dialect that aims to be more radical and practical, also acts as a test bed for features that get imported in CoffeeScript.
stars 494 forks 40 issues 41 open
 - [LiveScript](#) is a fork of Coco that is much more compatible with CoffeeScript, more functional, and with more features.
stars 2k forks 144 issues 160 open
- [IcedCoffeeScript](#) A CoffeeScript dialect that adds support for `await` and `defer` keywords which simplify async control flow.
stars 721 forks 59 issues 76 open
- [Parsec CoffeeScript](#) CS based on parser combinators. The project's aim is to add static metaprogramming (i.e. macros + syntax extensibility) to Coffee Script (CS), similar to how Metalua adds such features to Lua. The resulting compiler, once merged with the official compiler, should be usable as a drop-in replacement for it.
stars 116 forks 13 issues 3 open
- [Contracts.coffee](#) A dialect of CoffeeScript that adds built-in support for contracts.
stars 225 forks 6 issues 28 open

Pages 11

Home

[HowTo] Compiling and Setting Up Build Tools

[Howto] Hacking on the CoffeeScript Compiler

Build tools

Common Gotchas

FAQ

In The Wild

List of languages that compile to JS

Text editor plugins

Uniform Type Identifiers

Web framework plugins

Clone this wiki locally

<https://github.com/jashkenas/coffeescript>

Clone in Desktop



Popular Transpilers



Babel



TypeScript



Why Babel?

BABEL

Modern, standards-based JS, today.



Why TypeScript?



- Superset of JavaScript**
- Enhanced autocompletion**
- Safer refactoring**
- Clearer intent**



TypeScript

ES6

ES5



TypeScript vs Babel

TypeScript	Babel
Enhanced Autocomplete	Write standardized JS
Enhanced readability	Leverage full JS Ecosystem
Safer refactoring	Use experimental features earlier
Additional non-standard features	No type defs, annotations required ES6 imports are statically analyzable



Popular Transpilers



Babel



TypeScript





Guides

[What is Babel?](#)

[Usage Guide](#)

[Configure Babel](#)

[Learn ES2015](#)

[Upgrade to Babel 7](#)

[Upgrade to Babel 7 \(API\)](#)

General

[Editors](#)

[Plugins](#)

[Presets](#)

[Caveats](#)

[FAQ](#)

[Roadmap](#)

Usage

Presets

EDIT

Don't want to assemble your own set of plugins? No problem! Presets can act as an array of Babel plugins or even a sharable [options config](#).

Official Presets

We've assembled some for common environments:

- [@babel/preset-env](#)
- [@babel/preset-flow](#)
- [@babel/preset-react](#)
- [@babel/preset-typescript](#)

Many other community maintained presets are available [on npm!](#)

Official Presets

[Stage-X \(Experimental Presets\)](#)

[Creating a Preset](#)

[Preset Paths](#)

[Preset Shorthand](#)

[Preset Ordering](#)

[Preset Options](#)

Guides

[What is Babel?](#)[Usage Guide](#)[Configure Babel](#)[Learn ES2015](#)[Upgrade to Babel 7](#)[Upgrade to Babel 7 \(API\)](#)

General

[Editors](#)[Plugins](#)[Presets](#)[Caveats](#)[FAQ](#)[Roadmap](#)

Usage

@babel/preset-env

EDIT

`@babel/preset-env` is a smart preset that allows you to use the latest JavaScript without needing to micromanage which syntax transforms (and optionally, browser polyfills) are needed by your target environment(s). This both makes your life easier and JavaScript bundles smaller!

- [Install](#)
- [How Does it Work?](#)
- [Browserslist Integration](#)
- [Options](#)

Install

With npm:

Shell

Copy

```
npm install --save-dev @babel/preset-env
```

Install

[How Does it Work?](#)[Browserslist Integration](#)[Options](#)[targets](#)[bugfixes](#)[spec](#)[loose](#)[modules](#)[debug](#)[include](#)[exclude](#)[useBuiltIns](#)[corejs](#)[forceAllTransforms](#)[configPath](#)[ignoreBrowserslistConfig](#)[browserslistEnv](#)[shippedProposals](#)



Guides

- What is Babel?
- Usage Guide
- Configure Babel
- Learn ES2015
- Upgrade to Babel 7
- Upgrade to Babel 7 (API)

General

- Editors
- Plugins
- Presets
- Caveats
- FAQ
- Roadmap

Usage

Presets

EDIT

Don't want to assemble your own set of plugins? No problem! Presets can act as an array of Babel plugins or even a sharable [options](#) config.

Official Presets

We've assembled some for common environments:

- [@babel/preset-env](#)
- [@babel/preset-flow](#)
- [@babel/preset-react](#)
- [@babel/preset-typescript](#)

Many other community maintained presets are available [on npm!](#)

Official Presets

- Stage-X (Experimental Presets)
- Creating a Preset
- Preset Paths
- Preset Shorthand
- Preset Ordering
- Preset Options



Stage-X (Experimental Presets)

Guides

[What is Babel?](#)[Usage Guide](#)[Configure Babel](#)[Learn ES2015](#)[Upgrade to Babel 7](#)[Upgrade to Babel 7 \(API\)](#)

General

[Editors](#)[Plugins](#)[Presets](#)[Caveats](#)[FAQ](#)[Roadmap](#)

Usage

Any transforms in stage-x presets are changes to the language that haven't been approved to be part of a release of JavaScript (such as ES6/ES2015).

Subject to change

These proposals are subject to change so **use with extreme caution**, especially for anything pre stage-3. We plan to update stage-x presets when proposals change after each TC39 meeting when possible.

The TC39 categorizes proposals into the following stages:

- Stage 0 - Strawman: just an idea, possible Babel plugin.
- Stage 1 - Proposal: this is worth working on.
- Stage 2 - Draft: initial spec.
- Stage 3 - Candidate: complete spec and initial browser implementations.
- Stage 4 - Finished: will be added to the next yearly release.

[Official Presets](#)[Stage-X \(Experimental Presets\)](#)[Creating a Preset](#)[Preset Paths](#)[Preset Shorthand](#)[Preset Ordering](#)[Preset Options](#)



Guides

[What is Babel?](#)

[Usage Guide](#)

[Configure Babel](#)

[Learn ES2015](#)

[Upgrade to Babel 7](#)

[Upgrade to Babel 7 \(API\)](#)

General

[Editors](#)

[Plugins](#)

[Presets](#)

[Caveats](#)

[FAQ](#)

[Roadmap](#)

Usage

Presets

[EDIT](#)

Don't want to assemble your own set of plugins? No problem! Presets can act as an array of Babel plugins or even a sharable [options config](#).

Official Presets

We've assembled some for common environments:

- [@babel/preset-env](#)
- [@babel/preset-flow](#)
- [@babel/preset-react](#)
- [@babel/preset-typescript](#)

Many other community maintained presets are available [on npm!](#)

Official Presets

[Stage-X \(Experimental Presets\)](#)

[Creating a Preset](#)

[Preset Paths](#)

[Preset Shorthand](#)

[Preset Ordering](#)

[Preset Options](#)

Babel Configuration Styles

.babelrc

Not npm specific
Easier to read since isolated

package.json

One less file in your project

```
{  
  "name": "my-package",  
  "version": "1.0.0",  
  "babel": {  
    // my babel config here  
  }  
}
```



Build Script JS Style

Plain JS

No waiting for transpile = faster

No transpiler dependency

Transpiled

Enjoy the latest features

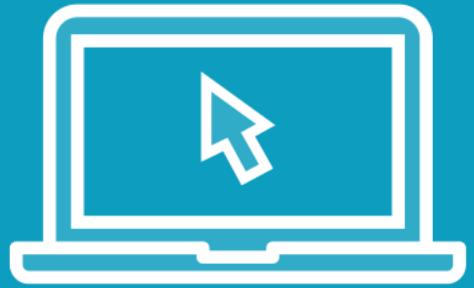
Consistent coding style

Use the same linting rules everywhere

Can eventually remove transpiler



Demo



Transpiling with Babel



Wrap Up



Transpiling is our present...and future

Transpilers

- Babel, TypeScript, dozens more

Configuring Babel

- .babelrc vs package.json
- Experimental features
- Transpiling build scripts

Set up Babel

Next up: Let's set up a bundler!

