Babel is a JavaScript compiler.

Use next generation JavaScript, today.

Setup (/docs/setup)

Try it Out (http://babeljs.io/repl/#?babili=false 2&experimental=false&loose=false&spec=false&code=%5B

Latest From Our Blog: The State of Babel (/blog/2016/12/07/the-state-of-babel)

Babel transforms your JavaScript

You put JavaScript in

JavaScript

```
[1,2,3].map(n => n + 1);
```

And get JavaScript out

JavaScript

```
[1,2,3].map(function(n) {
  return n + 1;
});
```

Start by installing the Babel CLI and a preset

Shell

```
npm install --save-dev babel-cli babel-preset-env
```

Create a `.babelrc` (/docs/usage/babelrc) file in your project (or use your package.json (/docs/usage/babelrc#use-via-package-json))

JSON

```
{
  "presets": ["env"]
}
```

ES2015 and beyond

Babel has support for the latest version of JavaScript through syntax transformers. These plugins (https://babeljs.io/docs/plugins/) allow you to use new syntax, **right now** without waiting for browser support. Check out our env preset (https://babeljs.io/docs/plugins/preset-env) to get started.

You can install this preset with

Shell

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

and add `"env"` to your `.babelrc` presets array.

- ✓ Arrow functions (/docs/plugins/transform-es2015-arrow-functions)
- ✓ Async functions (/docs/plugins/syntax-async-functions)
- ✓ Async generator functions (/docs/plugins/syntax-async-generators)
- ✓ Block scoping (/docs/plugins/transform-es2015-block-scoping)
- ✓ Block scoped functions (/docs/plugins/transform-es2015-block-scoped-functions)
- ✓ Classes (/docs/plugins/transform-es2015-classes)
- ✓ Class properties (/docs/plugins/transform-class-properties)
- ✓ Computed property names (/docs/plugins/transform-es2015-computed-properties)
- ✓ Constants (/docs/plugins/check-es2015-constants)
- ✓ Decorators (/docs/plugins/transform-decorators)
- ✓ Default parameters (/docs/plugins/transform-es2015-parameters)
- ✓ Destructuring (/docs/plugins/transform-es2015-destructuring)
- ✓ Do expressions (/docs/plugins/transform-do-expressions)
- ✓ Exponentiation operator (/docs/plugins/transform-exponentiation-operator)
- ✓ For-of (/docs/plugins/transform-es2015-for-of)
- ✓ Function bind (/docs/plugins/transform-function-bind)
- ✓ Generators (/docs/plugins/transform-regenerator)
- ✓ Modules (/docs/plugins/#modules)
- ✓ Module export extensions (/docs/plugins/transform-export-extensions)
- ✓ New literals (/docs/plugins/transform-es2015-literals)
- ✓ Object rest/spread (/docs/plugins/transform-object-rest-spread)
- ✓ Property method assignment (/docs/plugins/transform-es2015-shorthand-properties)
- ✓ Property name shorthand (/docs/plugins/transform-es2015-shorthand-properties)
- ✓ Rest parameters (/docs/plugins/transform-es2015-parameters)
- ✓ Spread (/docs/plugins/transform-es2015-parameters)
- ✓ Sticky regex (/docs/plugins/transform-es2015-sticky-regex)
- ✓ Template literals (/docs/plugins/transform-es2015-template-literals)
- ✓ Trailing function commas (/docs/plugins/syntax-trailing-function-commas)
- ✓ Type annotations (/docs/plugins/transform-flow-strip-types)
- ✓ Unicode regex (/docs/plugins/transform-es2015-unicode-regex)

Learn more about ES2015 → (/docs/learn-es2015/)

Polyfill

Since Babel only transforms syntax (like arrow functions), you can use babel-polyfill in order to support new globals such as Promise or new native methods like String.padStart (left-pad). It uses core-js (https://github.com/zloirock/core-js) and regenerator (https://facebook.github.io/regenerator/). Check out our babel-polyfill (/docs/usage/polyfill) docs for more info.

You can install the polyfill with

Shell

```
npm install --save-dev babel-polyfill
```

Use it by requiring it at the top of the entry point to your application or in your bundler config.

- ✓ ArrayBuffer (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/ArrayBuffer)
- ✓ Array.from (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/from)
- ✓ Array.of (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/of)
- ✓ Array#copyWithin (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/copyWithin)
- ✓ Array#fill (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/fill)
- ✓ Array#find (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/find)
- ✓ Array#findIndex (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/findIndex)
- ✓ Function#name (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Function/name)
- ✓ Map (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Map)

- ✓ Math.acosh (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/acosh)
- ✓ Math.hypot (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/hypot)
- ✓ Math.imul (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/imul)
- ✓ Number.isNaN (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Number/isNaN)
- ✓ Number.isInteger (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Number/isInteger)
- ✓ Object.assign (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/assign)
- ✓ Object.getOwnPropertyDescriptors (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/getOwnPropertyDescriptors)
- ✓ Object.is (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/is)
- ✓ Object.entries (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/entries)
- ✓ Object.values (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/values)
- ✓ Object.setPrototypeOf (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/setPrototypeOf)
- ✓ Promise (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise)
- ✓ Reflect (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Reflect)
- ✓ RegExp#flags (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/RegExp/flags)
- ✓ Set (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Set)
- ✓ String#codePointAt (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/codePointAt)
- ✓ String#endsWith (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/endsWith)
- ✓ String.fromCodePoint (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/fromCodePoint)
- ✓ String#includes (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/includes)
- ✓ String.raw (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/raw)
- ✓ String#repeat (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/repeat)
- ✓ String#startsWith (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/startsWith)
- ✓ String#padStart (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/padStart)
- ✓ String#padEnd (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/padEnd)
- ✓ Symbol (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Symbol)
- ✓ WeakMap (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/WeakMap)
- ✓ WeakSet (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/WeakSet)

Learn about more features → (https://github.com/zloirock/core-js#index)

JSX and Flow

Babel can convert JSX syntax and strip out type annotations. Check out our React preset (https://babeljs.io/docs/plugins/preset-react/) to get started. Use it together with the babel-sublime (https://github.com/babel/babel-sublime) package to bring syntax highlighting to a whole new level.

You can install this preset with

Shell

```
npm install --save-dev babel-preset-react
```

and add `"react"` to your `.babelrc` presets array.

JavaScript

```
export default React.createClass({
  getInitialState() {
    return { num: this.getRandomNumber() };
  },

  getRandomNumber(): number {
    return Math.ceil(Math.random() * 6);
  },

  render(): any {
    return <div>
      Your dice roll:
      {this.state.num}
    </div>;
  }
});
```

Learn more about JSX (https://facebook.github.io/jsx/) and Flow (http://flowtype.org/)

Pluggable

Babel is built out of plugins. Compose your own transformation pipeline using existing plugins or write your own. Easily use a set of plugins by using or creating a preset (/docs/plugins/#presets). Learn more → (/docs/plugins/)

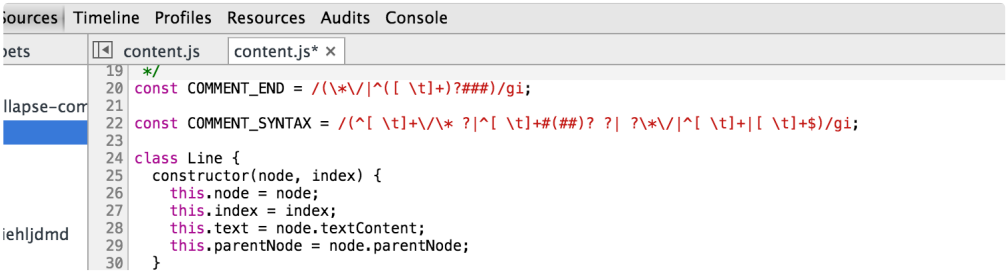
Create a plugin on the fly with astexplorer.net (https://astexplorer.net/#/KJ8AjD6maa)

JavaScript

```
// A plugin is just a function
export default function ({types: t}) {
  return {
    visitor: {
      Identifier(path) {
        let name = path.node.name;
        // reverse the name: JavaScript -> tpircSavaJ
        path.node.name = name.split('').reverse().join('');
      }
    }
  };
}
```

Debuggable

Source map support so you can debug your compiled code with ease.



Who's using Babel? (/users/)

(https://www.mozilla.org/) (https://yahoo.com/) (https://paypal.com)

Meet more Users (/users/)

Babel (https://github.com/babel/babel) · Distributed under MIT License (https://github.com/babel/babel/blob/master/LICENSE) · Code of Conduct (https://github.com/babel/babel/blob/master/CODE_OF_CONDUCT.md)

Looking for Babel 5.x docs? (http://henryzoo.com/babel.github.io/) · Found an issue with the docs? Report it here (https://github.com/babel/babel.github.io/issues/new).