Write the code you want to write

JavaScript build tooling at Fastmail

Joe Woods, PhillyJS, 20230112

Joe Woods
Senior Client Application Development Lead @ Fastmail
@joe@oldinternet.net

github.com

Search or jump to... / Pull requests Issues Codespaces Marketplace Explore

tc39 / proposal-decorators Public

Edit Pins | Watch 166 | Fork 105 | Star 2.3k

Code | Issues 26 | Pull requests 3 | Actions | Security | Insights

master | 54 branches | 0 tags | Go to file | Add file | Code

pzuraq Merge pull request #483 from trusktr/patch-1 ... 26a9d33 on Oct 1, 2022 277 commits

meetings	Update 2019-01-15.md	4 years ago
.gitignore	Script to build spec 2 (#235)	4 years ago
EXTENSIONS.md	Fix missing code block terminator	last year
README.md	update README to not limit class decorators to constructibles	3 months ago

README.md

Decorators

Stage: 3

Decorators are a proposal for extending JavaScript classes which is widely adopted among developers in transpiler environments, with broad interest in standardization. TC39 has been iterating on decorators proposals for over five years. This document describes a new proposal for decorators based on elements from all past proposals.

This README describes the current decorators proposal, which is a work in progress. For previous iterations of this proposal, see the commit history of this repository.

About

Decorators for ES6 classes

arai-a.github.io/ecma262-compare/?p...

Readme
2.3k stars
166 watching
105 forks

Contributors 41

+ 30 contributors

Environments 1

github-pages Active

* Stage 0: Here’s an idea.
* Stage 1: How is this going to work?
* Stage 2: Here’s how it’s going to work.
* Stage 3: Here’s how we know it’s going to work, having survived *the enemy*
* Stage 4: It’s JavaScript

Options

▶ History

`version`

`"2022-03"`, `"2021-12"`, `"2018-09"` or `"legacy"`. Defaults to `"2018-09"`.

Selects the decorators proposal to use:

- `"2022-03"` is the proposal version that reached consensus for Stage 3 in the March 2022 TC39 meeting. You can read more about it at `tc39/proposal-decorators@8ca65c046d`.
- `"2021-12"` is the proposal version as it was presented to TC39 in Dec 2021. You can read more about it at `tc39/proposal-decorators@d6c056fa06`.
- `"2018-09"` is the proposal version that was initially promoted to Stage 2 presented to TC39 in Sept 2018. You can read more about it at `tc39/proposal-decorators@7fa580b40f`.
- `legacy` is the original Stage 1 proposal, defined at `wycats/javascript-decorators@e1bf8d41bf`.

https://babeljs.io/docs/en/babel-plugin-proposal-decorators

JS function logged(value, { kind, name }) { Untitled-1 1

```js
function logged(value, { kind, name }) {
  if (kind === "class") {
    return class extends value {
      constructor(...args) {
        super(...args);
        console.log(
          `constructing an instance of ${name} with arguments ${args.join(
            ", "
          )}`
        );
      }
    };
  }

  // ...
}

@logged
class C {}

new C(1);
// constructing an instance of C with arguments 1
```

VSCodium

JS class C {} Untitled-1

```javascript
class C {}

C = logged(C, {
  kind: "class",
  name: "C",
}) ?? C;

new C(1);
```

Ln 8, Col 10 Spaces: 4 UTF-8 LF {} JavaScript ✓ Prettier

VSCodium

JS class C { Untitled-1 1

```
class C {
    accessor x = 1;
}


class C {
    @logged accessor x = 1;
}
```

JS class C { Untitled-2

```
class C {
    #x = 1;

    get x() {
        return this.#x;
    }

    set x(val) {
        this.#x = val;
    }
}
```

Ln 7, Col 5 Spaces: 4 UTF-8 LF {} JavaScript Prettier

Overture.js

VSCodium

JS const PhoneVerifyView = Class({ Untitled-1 1

```javascr
const PhoneVerifyView = Class({
    // ...
    reset: function () {
        this.set('error', '').set('hasChecked', false);
        if (this.get('state') === GET_NUMBER) {
            this.set('verificationCode', '');
        }
    }.observes('state'),

```

Ln 9, Col 1 Spaces: 4 UTF-8 LF {} JavaScript ✓ Prettier

VSCodium

JS class PhoneVerifyView { Untitled-1 3

```javascript
class PhoneVerifyView {
    // ...
    @observes('state')
    reset() {
        this.error = '';
        this.hasChecked = false;
        if (this.state === GET_NUMBER) {
            this.verificationCode = '';
        }
    },

```

Ln 11, Col 1 Spaces: 4 UTF-8 LF {} JavaScript ✓ Prettier

```
this.error = '';
this.
if (t
```

(property) PhoneVerifyView.error: string

Turning JavaScript into other JavaScript

minifying
bundling
transpiling

esbuild, snowpack, webpack, rollup, babel, swc, bublé, gulp, even TypeScript

rollup
core-js
bublé
terser

(also other fun stuff like postcss and typescript)

rollup

core-js

swc

(with other fun stuff still)

	local	test server
terser + buble	8:45.21	4:45.413
swc	5:59.73	3:10.776
time savings	2:45.48	1:34.637

Rust

52 lines (43 sloc) | 839 Bytes

Raw Blame

Decorators

Decorator

```
interface Decorator <: Node {
    type: "Decorator";
    expression: Expression;
}
```

AccessorProperty

```
interface AccessorProperty <: Node {
    type: "AccessorProperty";
    key: Expression | PrivateIdentifier;
    value: Expression | null;
    computed: boolean;
    static: boolean;
    decorators: [ Decorator ];
}
```

Class

```
extend interface Class {
    decorators: [ Decorator ];
}
```

ClassBody

https://github.com/estree/estree/blob/master/stage3/decorators.md

astexplorer.net

AST Explorer | Snippet | JavaScript | swc | Transform | default | ? Parser: swc-1.2.146

```
1 class PhoneVerifyView {
2     @observes('state')
3     reset() {
4         this.error = '';
5         this.hasChecked = false;
6         if (this.state === GET_NUMBER) {
7             this.verificationCode = '';
8         }
9     }
10 }
11
```

Tree | JSON 2ms

☑ Autofocus ☑ Hide methods ☑ Hide empty keys ☑ Hide location data ☑ Hide type keys

```
- ClassDeclaration {
  + identifier: Identifier {span, value, optional}
    declare: false
  + span: {ctxt}
    decorators: [ ]
  - body: [
    - ClassMethod {
      + span: {ctxt}
      - key: Identifier {
        + span: {ctxt}
          value: "reset"
          optional: false
        }
      - function: {
          params: [ ]
        - decorators: [
          - Decorator = $node {
            + span: {ctxt}
            - expression: CallExpression {
              + span: {ctxt}
              + callee: Identifier {span, value, optional}
              + arguments: [1 element]
              }
            }
          ]
        + span: {ctxt}
        - body: BlockStatement {
          + span: {ctxt}
          + stmts: [3 elements]
          }
```

Built with React, Babel, Font Awesome, CodeMirror, Express, and webpack | GitHub | Build: dda6ea7

https://tjwds.url.lol/ast

write-the-code-you-want-to-write

index.js

```js
function observes() {
  console.log("decorator invoked :-)");
}

class TestClass {
  @observes("state")
  doSomething() {
    console.log("You did something");
  }
}

console.log(TestClass);
```

.swcrc

You, 16 seconds ago | 1 author (You)

```json
{
    "jsc": {
        "parser": {
            "syntax": "ecmascript",
            "decorators": true
        },
        "target": "es5"
    },
    "minify": false
}
```

Ln 13, Col 1 Spaces: 2 UTF-8 LF JavaScript Prettier

https://github.com/tjwds/write-the-code-you-want-to-write
https://swc.rs/playground

joe@woods-fm — ..want-to-write

```
    if (n === "Map" || n === "Set") return Array.from(n);
    if (n === "Arguments" || /^(?:Ui|I)nt(?:8|16|32)(?:Clamped)?Array$/.test(n)) return
_arrayLikeToArray(o, minLen);
}
function observes() {
    console.log("decorator invoked :-)");
}
var TestClass = _decorate([], function(_initialize) {
    var TestClass = function TestClass() {
        "use strict";
        _classCallCheck(this, TestClass);
        _initialize(this);
    };
    return {
        F: TestClass,
        d: [
            {
                kind: "method",
                decorators: [
                    observes("state")
                ],
                key: "doSomething",
                value: function doSomething() {
                    console.log("You did something");
                }
            }
        ]
    };
});
console.log(TestClass);



projects/write-the-code-you-want-to-write ⬢ v18.10.0 2s
➜
```

joe@woods-fm — ..want-to-write

```
    };
});
console.log(TestClass);


//#sourceMappingURL=data:application/json;charset=utf-8; base64,IntcInZlcnNpb25cIjozLFw
291cmNlc1wiOltdLFwibmFtZXNcIjpbXSxcIm1hcHBpbmdzXCI6XCJcIixcImZpbGVcIjpcInN0ZG91dFwifSI=
projects/write-the-code-you-want-to-write ⬢ v18.10.0 2s
➜ npm run build | wc
     416    1524   15803
[
projects/write-the-code-you-want-to-write ⬢ v18.10.0 2s
➜
```

write-the-code-you-want-to-write

spreadExample.js U

examples > spreadExample.js > ...

```js
const foo = [0, 1]
const bar = [...foo];

```

spreadExampleCompiled.js U

examples > spreadExampleCompiled.js > _nonIterableSpread

```js
function _arrayLikeToArray(arr, len) {
    if (len == null || len > arr.length) len = arr.length;
    for(var i = 0, arr2 = new Array(len); i < len; i++)arr2[i] = arr[i];
    return arr2;
}
function _arrayWithoutHoles(arr) {
    if (Array.isArray(arr)) return _arrayLikeToArray(arr);
}
function _iterableToArray(iter) {
    if (typeof Symbol !== "undefined" && iter[Symbol.iterator] != null || iter["@@iterator"] != null) return Array.from(iter);
}
function _nonIterableSpread() {
    throw new TypeError("Invalid attempt to spread non-iterable instance.\\nIn order to be iterable, non-array objects must have a [Symbol.iterator]() method.");
}
function _toConsumableArray(arr) {
    return _arrayWithoutHoles(arr) || _iterableToArray(arr) || _unsupportedIterableToArray(arr) || _nonIterableSpread();
}
function _unsupportedIterableToArray(o, minLen) {
    if (!o) return;
    if (typeof o === "string") return _arrayLikeToArray(o, minLen);
    var n = Object.prototype.toString.call(o).slice(8, -1);
    if (n === "Object" && o.constructor) n = o.constructor.name;
    if (n === "Map" || n === "Set") return Array.from(n);
    if (n === "Arguments" || /^(?:Ui|I)nt(?:8|16|32)(?:Clamped)?Array$/.test(n)) return _arrayLikeToArray(o, minLen);
}
var foo = [
    0,
    1
];
var bar = _toConsumableArray(foo);


//#sourceMappingURL=data:application/json;charset=utf-8; base64,
```

Ln 3, Col 1 Spaces: 4 UTF-8 LF {} JavaScript Prettier

github.com

Contains Search Done

3 contributors

59 lines (59 sloc) | 5.52 KB

Raw Blame

```
{
    "compilerOptions": {
        /* Basic Options */
        "target": "es2015" /* Specify ECMAScript target version: 'ES3' (default), 'ES5', 'ES2015', 'ES2016', 'ES2017','ES2018' or 'ESNEXT'. */,
        "module": "commonjs" /* Specify module code generation: 'none', 'commonjs', 'amd', 'system', 'umd', 'es2015', or 'ESNext'. */,
        // "lib": [],                           /* Specify library files to be included in the compilation. */
        "allowJs": true /* Allow javascript files to be compiled. */,
        // "checkJs": true,                     /* Report errors in .js files. */
        // "jsx": "preserve",                   /* Specify JSX code generation: 'preserve', 'react-native', or 'react'. */
        "declaration": true /* Generates corresponding '.d.ts' file. */,
        // "declarationMap": true,              /* Generates a sourcemap for each corresponding '.d.ts' file. */
        "sourceMap": false /* Generates corresponding '.map' file. */,
        // "outFile": "./",                     /* Concatenate and emit output to single file. */
        "outDir": "./" /* Redirect output structure to the directory. */,
```

ECMAScript 5 Strict Mode - OTHER

Method of placing code in a "strict" operating context.

Usage: Global — % of all users: 98.72% + 0.02% = 98.74%

Current aligned | Usage relative | Date relative | Filtered | All

Chrome	Edge *	Safari	Firefox	Opera	IE	Chrome for Android	Safari on iOS *	Samsung Internet	Opera Mini *	Opera Mobile *	UC Browser for Android	Android Browser *	Firefox for Android	QQ Browser
		3.1-4												
4-12		[1] 5-5.1	2-3.6	10-11.5	6-9		3.2-4.3					2.1-2.3		
13-107	12-107	6-16.1	4-107	12.1-91	10		5-16.1	4-18.0		12-12.1		3-4.4.4		
108	108	16.2	108	92	11	108	16.2	19.0	all	72	13.4	108	107	13.1
109-111		16.3-TP	109-110				16.3							

Notes | Test on a real browser | Known issues (0) | Resources (3) | Feedback

[1] Partial support in older Safari refers to strict mode still accepting a lot of JS th

JavaScript statement: `try...catch`

Usage: Global — % of all users: 95.87%

Current aligned | Usage relative | Date relative | Filtered | All

Chrome	Edge *	Safari	Firefox	Opera	IE	Chrome for Android	Safari on iOS *	Samsung Internet	Opera Mini *	Opera Mobile *	UC Browser for Android	Android Browser *	Firefox for Android	QQ Browser
												2.1-4.3		
4-107	12-107	3.1-16.1	2-107	10-91	6-10		3.2-16.1	4-18.0		12-12.1		4.4-4.4.4		
108	108	16.2	108	92	11	108	16.2	19.0	all	72	13.4	108	107	13.1
109-111		16.3-TP	109-110				16.3							

Notes | Test on a real browser | Sub-features | Feedback

See full reference on MDN Web Docs.

Support data for this feature provided by:

MDN browser-compat-data

JavaScript built-in: Array: at

Usage % of all users ?

Global 88.28%

Current aligned | Usage relative | Date relative | Filtered | All

Chrome	Edge *	Safari	Firefox	Opera	IE	Chrome for Android	Safari on iOS *	Samsung Internet	Opera Mini *	Opera Mobile *	UC Browser for Android	Android Browser *	Firefox for Android	Br
4-91	12-91	3.1-15.3	2-89	10-77			3.2-15.3	4-15.0						
92-107	92-107	15.4-16.1	90-107	78-91	6-10		15.4-16.1	16.0-18.0		12-12.1		2.1-4.4.4		
108	108	16.2	108	92	11	108	16.2	19.0	all	72	13.4	108	107	1
109-111		16.3-TP	109-110				16.3							

https://blog.logrocket.com/using-spack-bundler-in-rust-to-speed-up-builds/

rollup

core-js

swc

(with other fun stuff still)

Write the code you want to write

JavaScript build tooling at Fastmail

Joe Woods, PhillyJS, 20230112