

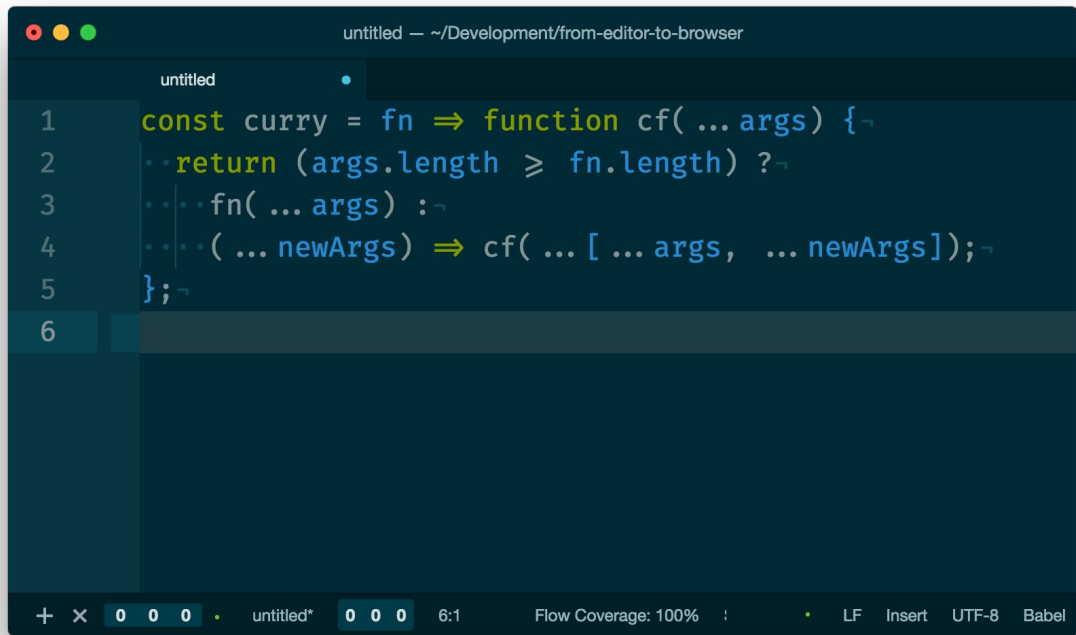
Problem #1

We are 22 frontend developers
(according to the Slack channel)

Different backgrounds

- Computer Science graduates
- Multimedia techs
- Designers
- Others...

Different writing styles



```
untitled — ~/Development/from-editor-to-browser  
untitled  
1  const curry = fn => function cf( ... args) {  
2    ··return (args.length ≥ fn.length) ?  
3    ··|·fn( ... args) :  
4    ··|·( ... newArgs) => cf( ... [ ... args, ... newArgs]);  
5  };  
6
```

The screenshot shows a code editor window titled 'untitled' with a file path of '~/Development/from-editor-to-browser'. The code is written in JavaScript and uses a complex arrow function syntax. The function 'curry' is defined as 'fn => function cf(... args) { ... }'. The body of 'cf' is a ternary expression: 'return (args.length ≥ fn.length) ? fn(... args) : (... newArgs) => cf(... [... args, ... newArgs]);'. The code is formatted with syntax highlighting and line numbers. The bottom status bar shows 'Flow Coverage: 100%' and other editor settings.

(some of us are more cryptic than others)

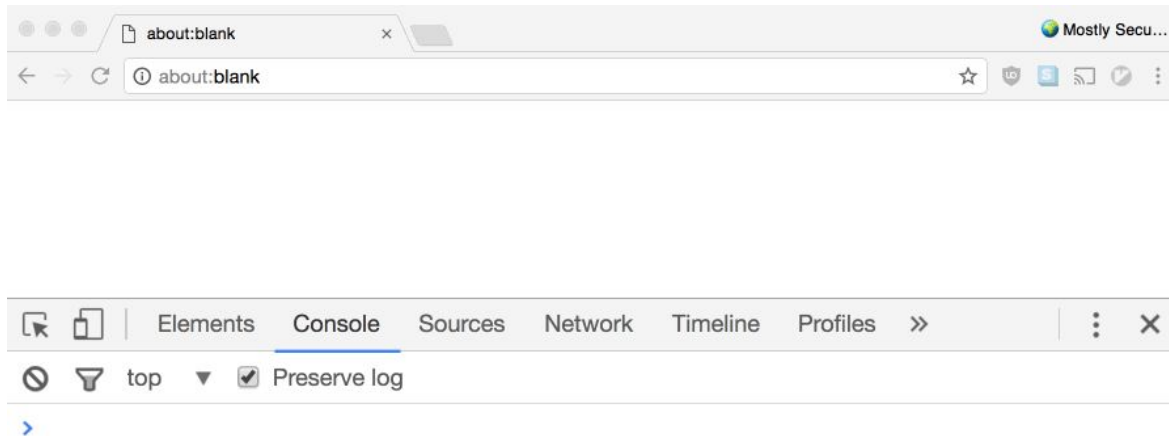
Problem #2



JavaScript was made in 10 days



image taken from: http://xahlee.info/comp/js_book_man_made_complexity.html

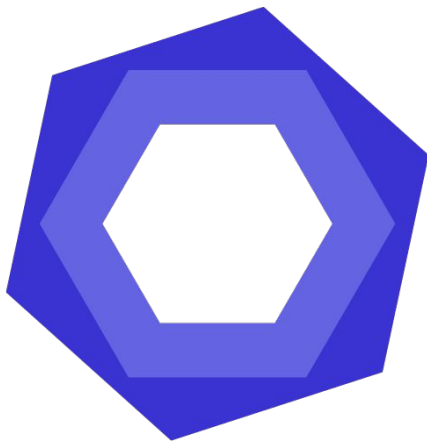


JavaScript tries to always return a value for any expression, even when it's nonsensical

Problem #3



Solution #1



The pluggable linting utility for JavaScript and JSX

eslint-config - npm search

NPM, Inc. [US] <https://www.npmjs.com/search?q=eslint-config>

No Problem, Meatbag npm Enterprise features pricing documentation support

npm eslint-config sign up or log in

Search Criteria

- ☒ Best Overall
- ☐ Quality
- ☐ Popularity
- ☐ Maintenance

Advanced Search

Click to try them out

scope:request

keywords:express

maintainer:bcoe

2396 packages found for "eslint-config"

eslint-config dthompson

Linting Config

0.3.0

eslint-config-airbnb ljharb

Airbnb's ESLint config, following our styleguide

14.1.0

eslint-config-airbnb-base ljharb

Airbnb's base JS ESLint config, following our styleguide

Extensible

01-dangerous-constructs.js — ~/Development/from-editor-to-browser

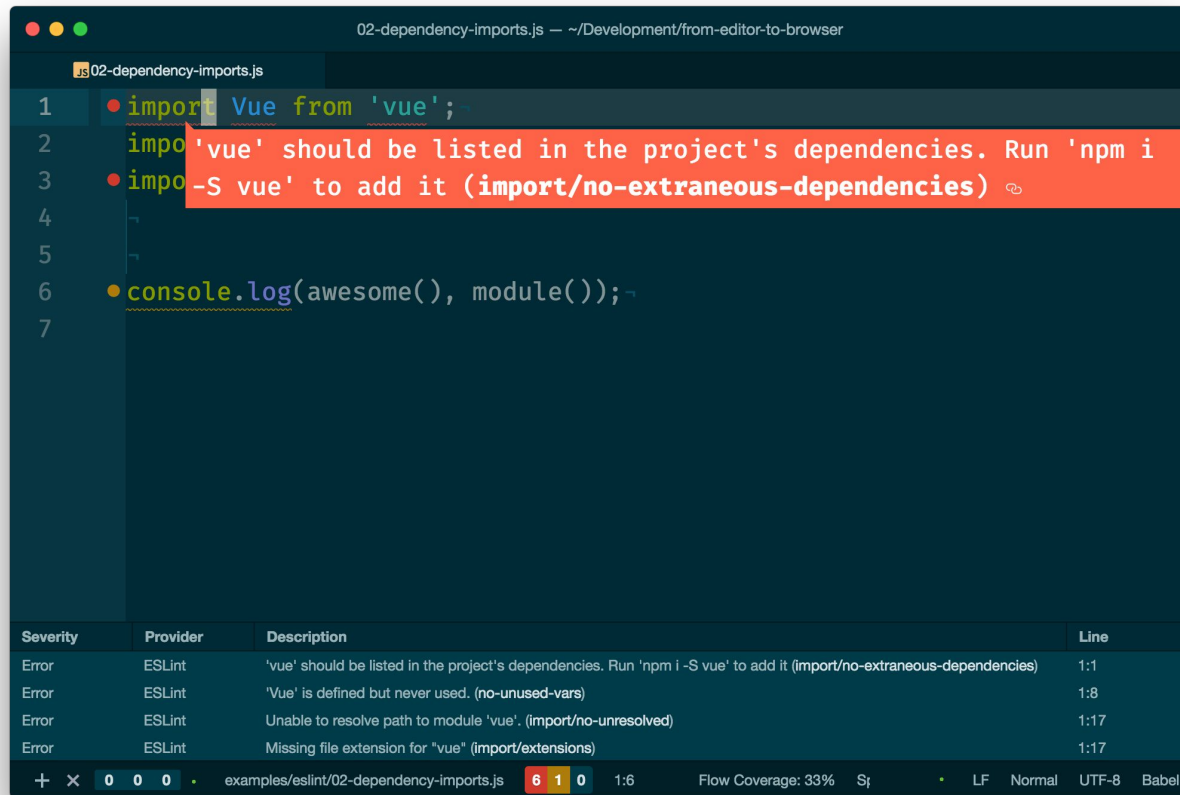
js01-dangerous-constructs.js

```
1 // this is bad, but ESLint throws a warning
2 function toNumber(str) {
3   return eval(str);
4 }
5
6 // this is also bad, but ESLint gets our backs
7 function call(str) {
8   return Function(str);
9 }
10
11 // We can't blame JavaScript for this one, and ESLint is not perfect
12 function throttle(str) {
13   return setTimeout(str);
14 }
15
16
```

The Function constructor is eval. (no-new-func)

Severity	Provider	Description	Line
Error	ESLint	eval can be harmful. (no-eval)	3:10
Error	ESLint	The Function constructor is eval. (no-new-func)	8:10

+ × 0 0 0 • examples/eslint/01-dangerous-constructs.js 2 0 0 8:14 Flow Coverage: 79% • LF Normal UTF-8 Babel



03-browser-compatibility.js — ~/Development/from-editor-to-browser

03-browser-compatibility.js

1

2

3

4

5

6

```
export default () => fetch('https://api.twitter.com/');  
  
export const observer = () => new IntersectionObserver(() => {});
```

IntersectionObserver is not supported
in Safari 8, iOS Safari 8.1-8.4, IE
Mobile 10, IE 8, Firefox 48, Edge 12
(compat/compat) [🔗](#)

Severity	Provider	Description	Line
Error	ESLint	fetch is not supported in Safari 8, iOS Safari 8.1-8.4, IE Mobile 10, IE 8, Edge 12 (compat/compat)	2:22
Error	ESLint	IntersectionObserver is not supported in Safari 8, iOS Safari 8.1-8.4, IE Mobile 10, IE 8, Firefox 48, Edge 12 (compat/compat)	5:31

+ × 0 0 0

examples/eslint/03-browser-compatibility.js 2 0 0 5:32

Flow Coverage: 78%

LF Normal UTF-8 Babel

05-react.jsx — ~/Development/from-editor-to-browser

06-a11y.jsx 05-react.jsx

```
1 import { Component } from 'react';
2
3 class Awesome extends Component {
4   render() {
5     return (
6       <div className="my-awesome-component">
7         Hello World
8         {this.props.message && this.props.message}
9       </div>
10     );
11   }
12 }
13
14 export default Awesome;
```

'message' is missing in props validation (react/prop-types)

Severity	Provider	Description	Line
Error	ESLint	Component should be written as a pure function (react/prefer-stateless-function)	3:1
Error	ESLint	'React' must be in scope when using JSX (react/react-in-jsx-scope)	6:7
Error	ESLint	'message' is missing in props validation (react/prop-types)	8:21
Error	ESLint	'message' is missing in props validation (react/prop-types)	8:43

+ × 0 0 0 • examples/eslint/05-react.jsx 4 0 0 10:6 Flow Coverage: 63% Spaces: 2 • LF Normal UTF-8 Babel

06-a11y.jsx — ~/Development/from-editor-to-browser

06-a11y.jsx

```
3 // Static Elements have no semantic meaning, we need to add a role.
4 export default ({ callback }) => (
5   <div onClick={callback}>Hello</div>
6 );
7
8 // This is a violation of the rule (jsx-a11y/no-static-element-interactions)
9 // associated with an input
10 export const Input = () => {
11   <div>
12     <label>Username</label>
13     <input type="text" />
14   </div>;
15 };
16
```

Visible, non-interactive elements should not have mouse or keyboard event listeners (jsx-a11y/no-static-element-interactions)

Severity	Provider	Description	Line
Error	ESLint	'callback' is missing in props validation (react/prop-types)	4:19
Error	ESLint	Visible, non-interactive elements should not have mouse or keyboard event listeners (jsx-a11y/no-static-element-interactions)	5:3
Error	ESLint	Expected an assignment or function call and instead saw an expression. (no-unused-expressions)	11:3

+ x 0 0 0 examples/eslint/06-a11y.jsx 4 0 0 5:6 Flow Coverage: 71% Spaces: 2 LF Normal UTF-8 Babel


```
04-stylish-features.js — ~/Development/from-editor-to-browser
04-stylish-features.js
1
2 • b = 2,
3 • c = 3;
4
5
6 • const iDontLikeSemiColons = 42
7
8 • const iHateTyping = () => {
9 •   return function() { return 5 }
10 • }
11
12 • const iPrefer4Spaces = () => {
13 •   console.log('I am waaay too nested')
14 • }
15
```

Severity	Provider	Description	Line
Error	ESLint	Split 'const' declarations into multiple statements. (one-var)	1:1
Error	ESLint	'a' is assigned a value but never used. (no-unused-vars)	1:7
Error	ESLint	'b' is assigned a value but never used. (no-unused-vars)	2:3
Error	ESLint	'c' is assigned a value but never used. (no-unused-vars)	3:3

+ × 0 0 0 • examples/eslint/04-stylish-features.js* 16 2 0 14:1 Flow Coverage: 100% Spi • LF Normal UTF-8 Babel

Of the 264 available rules available out-of-the-box, 84 can be automatically fixed by ESLint

Solution #2



A STATIC TYPE CHECKER FOR
JAVASCRIPT

A yellow square containing the letters 'JS' in a bold, black, sans-serif font.

JS

- Helps to move fast, it's easy to prototype applications.
- Very permissive type system.
- A lot of implicit conversions: `{ } + [] === 0 // true`
- Only allows some degree of linting against likely errors.



- Types are a good form of documentation
- Errors at compile-time are better than runtime
- They provide metadata for better tooling
- Better performance in the browser (indirectly)



- Allows for gradual typing.
- Type Inference.
- It's only a checker and not a compiler.

js 01-primitive-types.js

```
1 // @flow
2
3 const list: Array<number> = [1, 2, 3, 4];
4
5 function toString(n: number): string {
6   return (n).toString();
7 }
8
9 type Hobbie = 'Reading' | 'Skating' | 'Music'
10
11 type Person = {
12   name: string,
13   age: number,
14   hobbies: Array<Hobbie>,
15   address?: {
16     street: string
17   }
18 }
```



0 0 0

examples/flowtype/01-primitive-types.js

2 1 0

13:15

Flow Coverage: 100% Sp



LF

Insert

UTF-8

Babel

02-primitive-types-after-flow.js — ~/Development/from-editor-to-browser

js02-primitive-types-after-flow.js

```
1  // .....
2
3  const list                = [1, 2, 3, 4];
4
5  function toString(n      )      {
6    ··return (n).toString();
7  }
8
9  .....
10 .....
11 .....
12 .....
13 .....
14 .....
15 .....
16 .....
17 .....
18 .....
```

+ × 0 0 0 · examples/flowtype/02-primitive-types-after-flow.js* 0 0 0 22:1 Flow Coverage: 10 · LF Insert UTF-8 Babel

01-primitive-types.js — ~/Development/from-editor-to-browser

js01-primitive-types.js

```
1 // @flow
2
3 function toString(n: number): string {
4   ··return (n).toString();
5 }
6
7 toString('Hello');
```

string This type is incompatible with the expected param type of number ·\

Severity	Provider	Description	Line
Error	Flow IDE	string This type is incompatible with the expected param type of number	7:10

+ × 0 0 0

examples/flowtype/01-primitive-types.js 1 0 0 7:12

Flow Coverage: 100% Spa

LF Insert UTF-8 Babel

js 04-global-assignment.js

```
1 // @flow
2
3 import React from 'react';
4
5 const Button = ({ children }) => (
6   <div ref={e} => { this.button = e; }>
7     {children}
8   </div>
9 );
10
```



0 0 0

examples/flowtype/04-global-assignment.js

0 0 0

•

LF

Insert

UTF-8

Babel

js 04-global-assignment.js

```
1 // @flow
2
3 import React from 'react';
4
5 const Button = ({ children }) => (
6   <div ref={e} => { this.button = e; }>
7     {children}
8   </div>
9 );
```

property `button`
Property cannot be
assigned on global
object `

+ ×

0 0 0

examples/flowtype/04-global-assignment.js

1 0 0

•

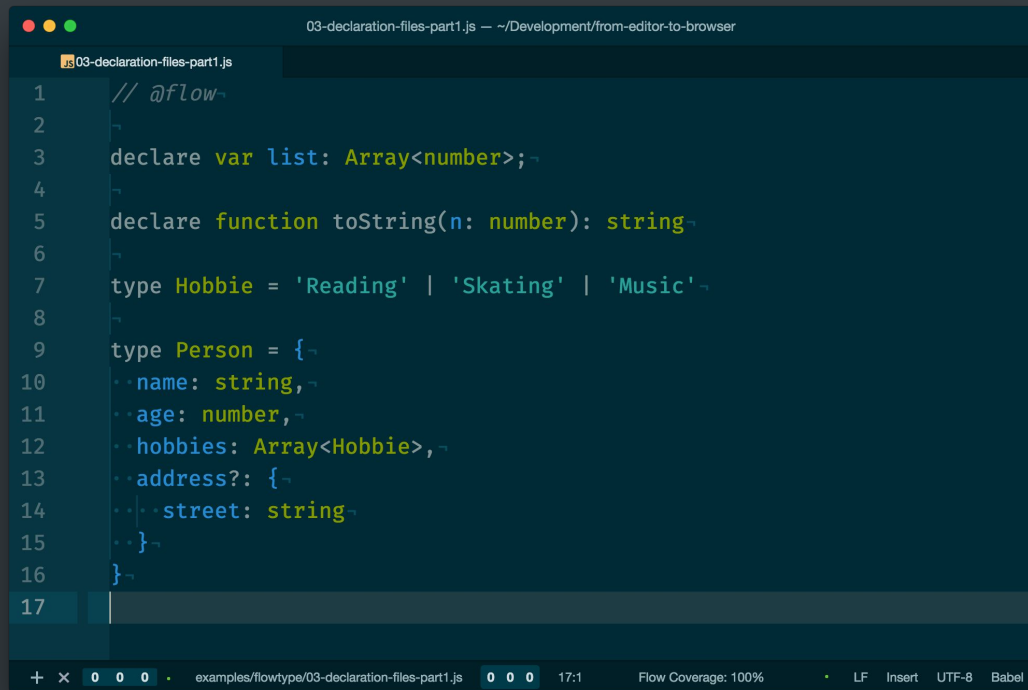
LF

Insert

UTF-8

Babel

If you don't like type annotations in your code:

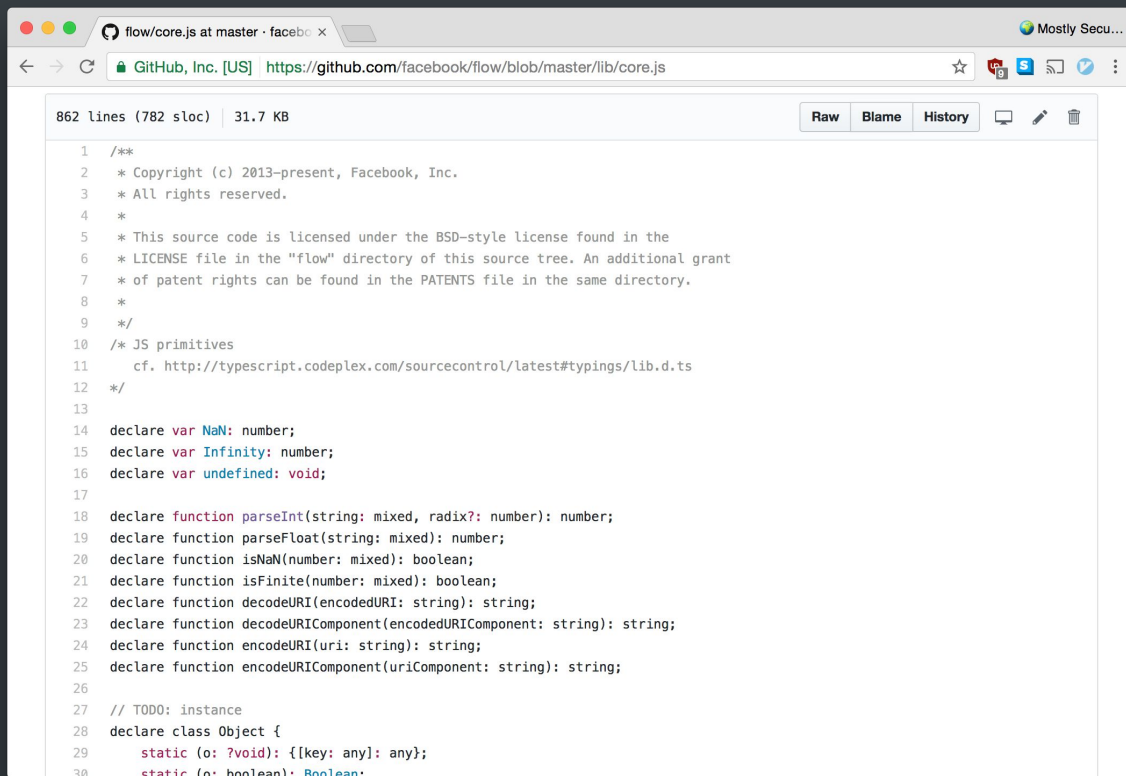


The screenshot shows a code editor window titled "03-declaration-files-part1.js" with a file icon on the left. The code is written in JavaScript with Flow type annotations. The editor has a dark theme with a light blue sidebar on the left showing line numbers 1 through 17. The code is as follows:

```
1 // @flow
2
3 declare var list: Array<number>;
4
5 declare function toString(n: number): string
6
7 type Hobbie = 'Reading' | 'Skating' | 'Music'
8
9 type Person = {
10   · name: string,
11   · age: number,
12   · hobbies: Array<Hobbie>,
13   · address?: {
14     · street: string
15   }
16 }
17
```

The status bar at the bottom shows the file path "examples/flowtype/03-declaration-files-part1.js", a progress indicator "0 0 0", the line number "17:1", and "Flow Coverage: 100%". On the right side of the status bar, there are icons for "LF", "Insert", "UTF-8", and "Babel".

Flow has built-in type definitions for JavaScript, React, Node and Browser APIs



```
862 lines (782 sloc) | 31.7 KB
Raw Blame History

1  /**
2   * Copyright (c) 2013-present, Facebook, Inc.
3   * All rights reserved.
4   *
5   * This source code is licensed under the BSD-style license found in the
6   * LICENSE file in the "flow" directory of this source tree. An additional grant
7   * of patent rights can be found in the PATENTS file in the same directory.
8   *
9   */
10 /* JS primitives
11    cf. http://typescript.codeplex.com/sourcecontrol/latest#typings/lib.d.ts
12 */
13
14 declare var NaN: number;
15 declare var Infinity: number;
16 declare var undefined: void;
17
18 declare function parseInt(string: mixed, radix?: number): number;
19 declare function parseFloat(string: mixed): number;
20 declare function isNaN(number: mixed): boolean;
21 declare function isFinite(number: mixed): boolean;
22 declare function decodeURI(encodedURI: string): string;
23 declare function decodeURIComponent(encodedURIComponent: string): string;
24 declare function encodeURI(uri: string): string;
25 declare function encodeURIComponent(uriComponent: string): string;
26
27 // TODO: instance
28 declare class Object {
29   static (o: ?void): {[key: any]: any};
30   static (o: boolean): Boolean;
```

What about third-party libs? (e.g.: lodash)





E2E tests

**Unit/Integration
Tests**

**Static-Analysis Tools
(Linting/Type checker)**

Questions?