

This Will **Flow** Your Mind



@tryggvigy



How To Find A **Title** For Your Talk

*for dummies

STEP 1

← → ↻ ⓘ pungenerator.org/puns?q=flow&whole_words=0

Pun Generator

☒ Include partial word matches

STEP 2

Let a thousand fFlowers bloom	Let a thousand flowers bloom
Know which way the wind bFlows	Know which way the wind blows
FelFlow traveller	Fellow traveller
BFlown to smithereens	Blown to smithereens
BFlow your own trumpet	Blow your own trumpet
<u>BFlow your mind</u>	Blow your mind
BeFlow the salt	Below the salt
BeFlow the belt	Below the belt
Flow your hat into the ring	Throw your hat into the ring
Flow in the towel	Throw in the towel
Stone's Flow	Stone's throw
A stone's Flow	A stone's throw
1991 HalFloween blizzard	1991 Halloween blizzard
The Perks Of Being A WallfFlower	The Perks Of Being A Wallflower
The Legend of Sleepy HolFlow	The Legend of Sleepy Hollow
Harry Potter and the Deathly HalFlows	Harry Potter and the Deathly Hallows



STEP 3



THANK YOU

This Will Flow Your Mind



@tryggvigy



1. WHAT IS IT

2. WHY USE IT

3. WHAT CAN IT DO

4. WHAT CAN IT NOT DO

5. LESSONS LEARNED

WHAT IS IT

WHAT IS IT



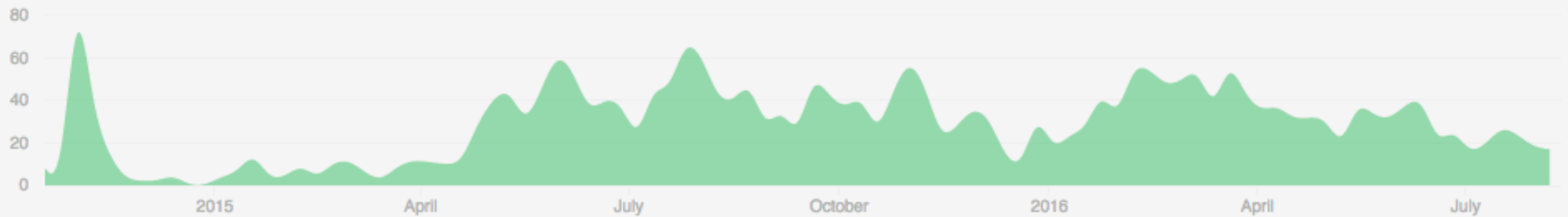
A STATIC TYPE CHECKER FOR JAVASCRIPT

Flow

Oct 26, 2014 – Aug 19, 2016

Contributions: **Commits** ▼

Contributions to master, excluding merge commits



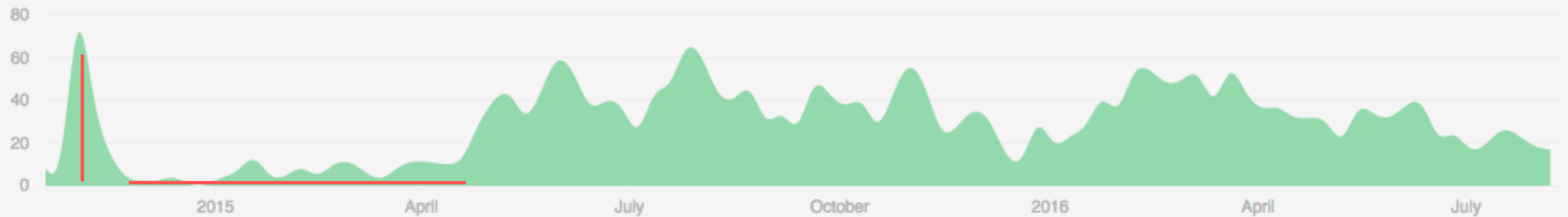
Flow

WHAT IS IT

Oct 26, 2014 – Aug 19, 2016

Contributions: **Commits** ▼

Contributions to master, excluding merge commits

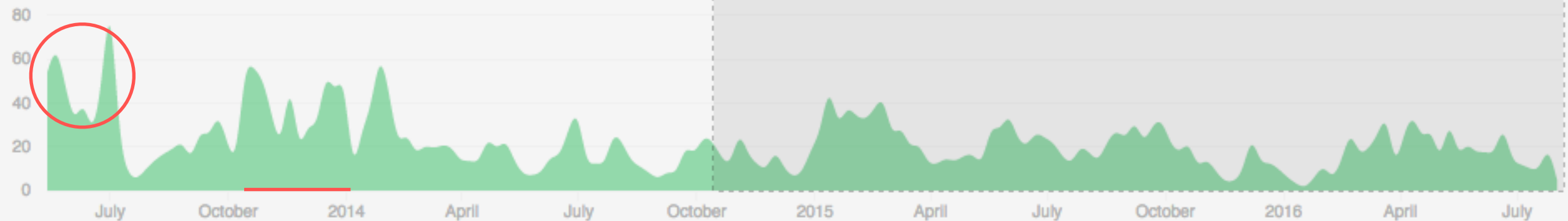


React

Oct 26, 2014 – Aug 19, 2016

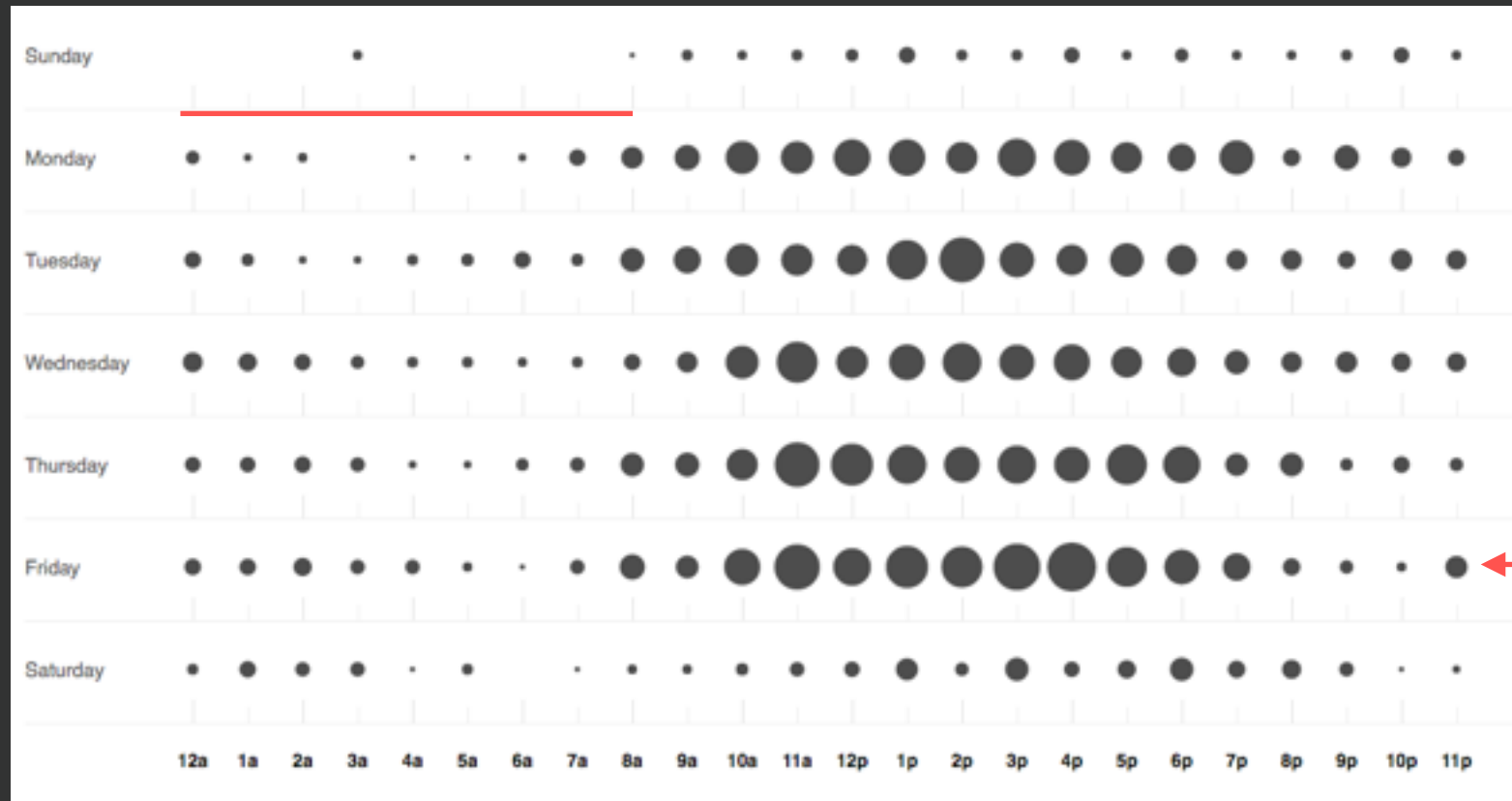
Contributions: **Commits** ▼

Contributions to master, excluding merge commits

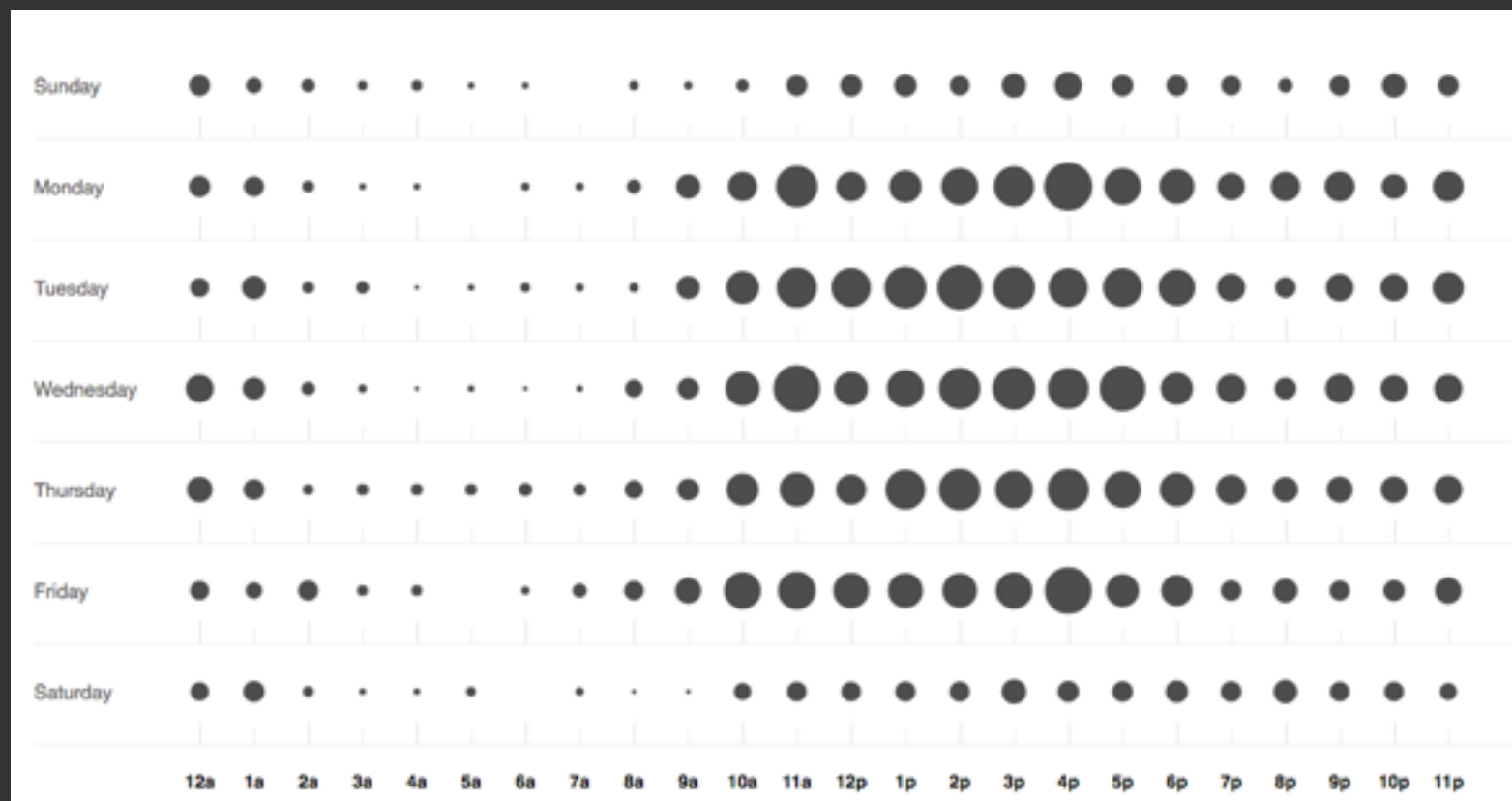


Flow

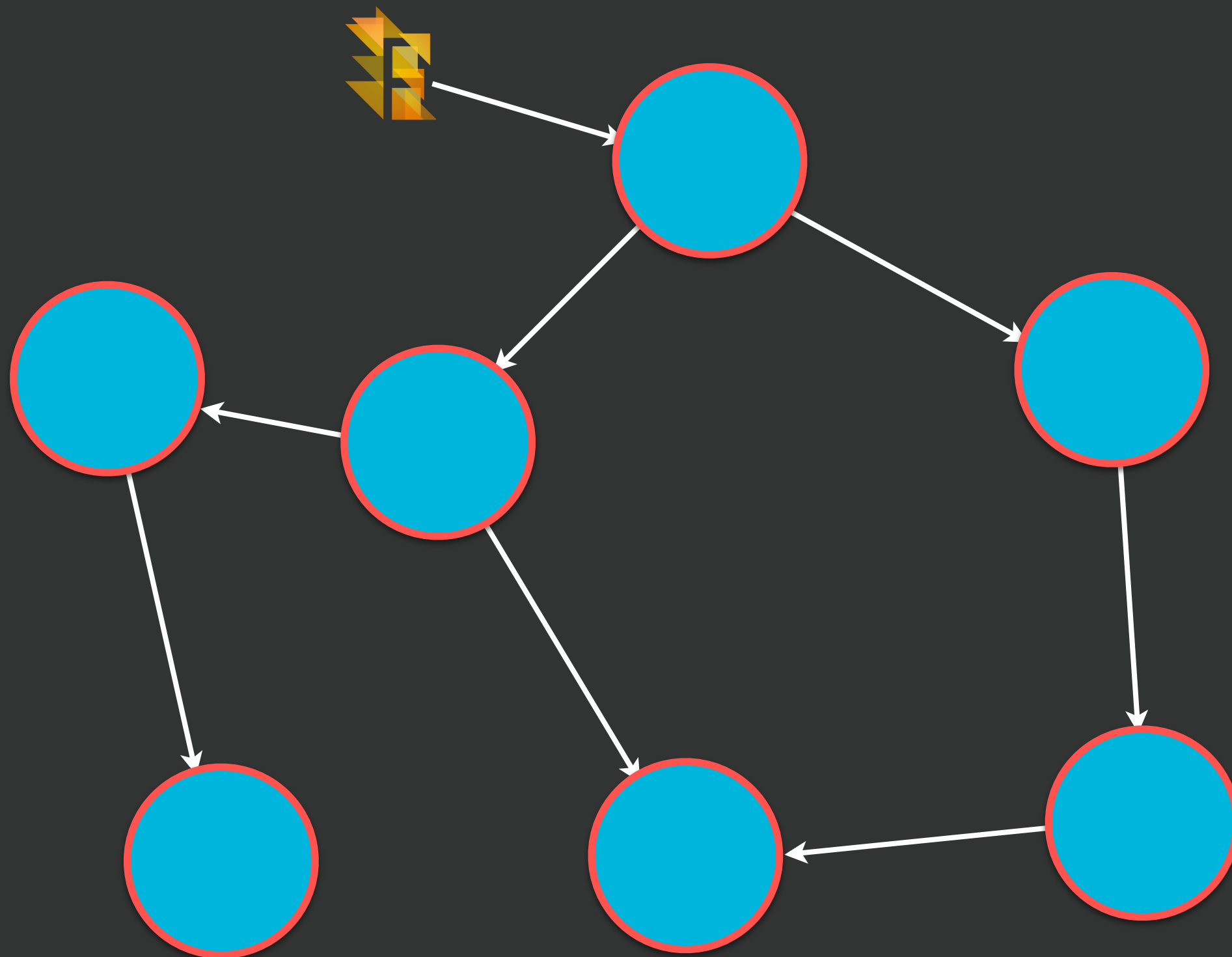
WHAT IS IT



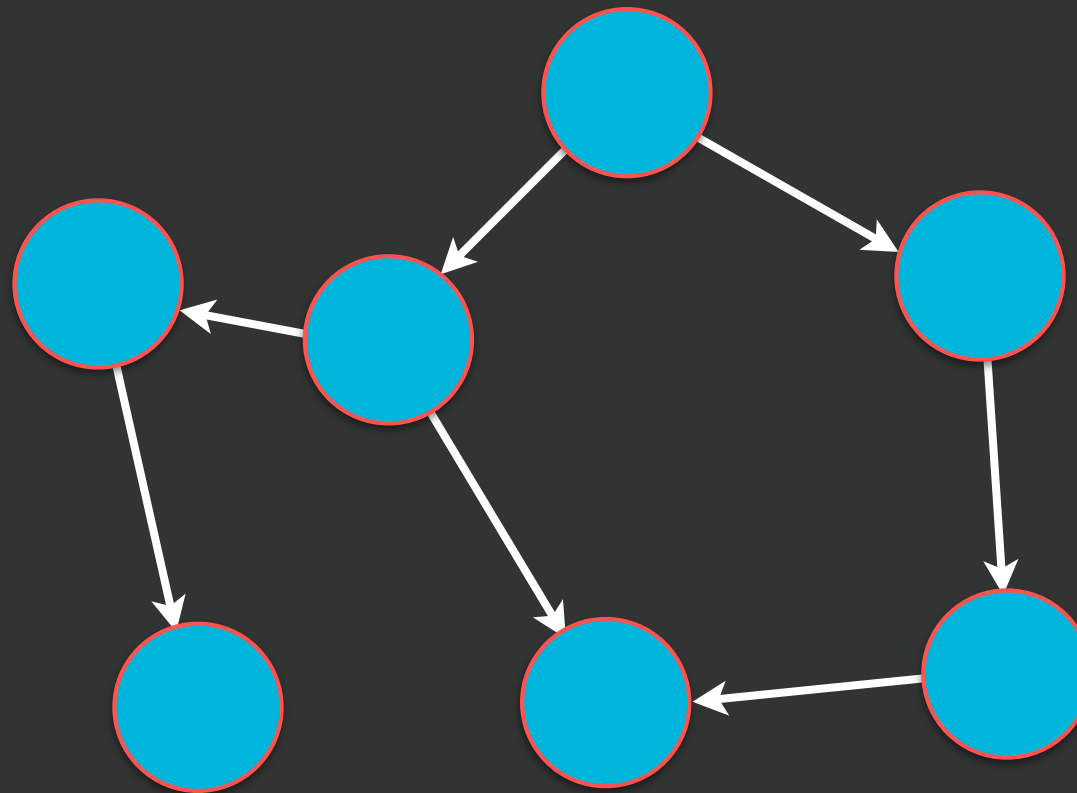
React



WHAT IS IT



FLOW CHART



```
// @flow

function addBangToNumber(num: number): string {
  return num + '!'
}
```

WHY USE IT


```
function enforceObject(x) {  
  if (typeof x === "object")  
    return x  
  else  
    return {}  
}
```

```
function enforceObject(x) {  
  if (typeof x === "object")  
    return x // x can still be null since `typeof null === "object"`  
  else  
    return {}  
}
```

```
function enforceObject(x) {  
  if (typeof x === "object" && x !== null)  
    return x  
  else  
    return {}  
}
```

```
function enforceObject(x: ?Object): Object {  
  if (typeof x === "object" && x !== null)  
    return x  
  else  
    return {}  
}
```

**WHAT CAN
IT DO**

Autocompletion

WHAT CAN IT DO

```
1  /* @flow */  
2  
3  import actionTypes from 'constants/actionTypes';  
4  import type {Action} from 'models/';  
5  
6  export function throwError(error: Error): Action {  
7    return {  
8      };  
9  }  
10
```

Template strings

async/await

Promises

ES6+

Generators

Arrow functions

Modules

Classes


Destructuring

let/const




Spread


WHAT CAN IT DO




[>](#)

 This repository

[Pull requests](#) [Issues](#) [Gist](#)

 [flowtype](#) / [flow-typed](#)


 Watch 28  Unstar 478  Fork 84

[Code](#) [Issues 37](#) [Pull requests 27](#) [Wiki](#) [Graphs](#) [Releases 0](#)

Branch: **master**

[flow-typed](#) / [definitions](#) / [npm](#) /

[Create new file](#) [Find file](#) [History](#)

Latest commit ea99362 15 hours ago  **turadg** committed with **gabelevi** include html in Express render callback (#271) [...](#)

..		
array-uniq_v1.x.x	A few tweaks to the The Sindre Sorhus Suite (#184)	2 months ago
axios_v0.11.x	add <*> (#150)	2 months ago
axios_v0.13.x	Update: Fix typo, remove duplicate entries, and add baseUrl property (#...	10 days ago
bcrypt_v0.8.x	Add bcrypt module definitions (#197)	a month ago
big.js_v3.x.x	Add big.js flow definitions (#122)	3 months ago
bigi_v1.x.x	Bigi definitions (#205)	a month ago
bitcoinjs-lib_v2.x.x	Correcting bitcoinjs definitions (#238)	16 days ago
bull_v1.x.x	0.28 fix (#166)	2 months ago
bunyan_v1.x.x	Allow use of process.stdout as one of bunyan streams (#203)	a month ago
cache-manager_v2.x.x	cache-manager (#214)	24 days ago

WHAT CAN IT DO

```
3 type PromiseLike<R> = {  
4   · · then<U>(  
5     · · · onFulfill?: (value: R) => Promise<U> | U,  
6     · · · onReject?: (error: any) => Promise<U> | U  
7   · · ): Promise<U>;  
8 }
```

WHAT CAN IT DO

REACT

REACT - prop validation

WHAT CAN IT DO

```
1  // @flow
2
3  import React from 'react'
4
5  class CoffeeImage extends React.Component {
6    ..props: {
7      ..format: 'png' | 'jpg',
8    }
9
10   ..render() {
11     ..const {format} = this.props;
12     ..return (
13       ..<img
14         ..src={`https://assets.imgix.net/unsplash/coffee.JPG?w=900&usm=10&fm=${format}&q=95`}
15       ../>
16     ..)
17   }
18 }
19
20 function App() {
21   ..return (
22     ..<div>
23       ..<h1>Getting a craving for a cup of ☕ yet?</h1>
24     ..</div>
25   ..)
26 }
27
```

```
1 // @flow
2
3 import React from 'react'
4
5 const CoffeeImage = ({format}: {format: 'png' | 'jpg'}) =>
6   <img
7     src={`https://assets.imgix.net/unsplash/coffee.JPG?w=900&usm=10&fm=${format}&q=95`}
8   />
9
10 function App() {
11   return (
12     <div>
13       <h1>Getting a craving for a cup of ☕ yet?</h1>
14       {/* $FlowSupressError: expected png or jpg, got gif */}
15       <CoffeeImage format={'gif'} />
16     </div>
17   )
18 }
19
```

**WHAT
CAN IT
NOT DO**

Warn about additional props not declared in a components Props type

```
1  // @flow
2
3  const Hello = ({name} : {name: string}) =>
4  ..<div>Hello {name}</div>
5
6  Hello({name: 'Tryggvi', age: 23})
7
```

```
1  // @flow
2
3  type Person = { salutation: string, last: string }
4
5  const subtypeOfPerson = {
6    salutation: "Mr",
7    first: "Bob",
8    last: "Dobalina"
9  }
10
11  function takesExactlyPerson(person: Person): void {}
12
13  takesExactlyPerson(subtypeOfPerson); // should error
14
```

Non-native data structures

“The more things change the
more they stay *immutable*”

IMMUTABLE
IMMUTABLE
IMMUTABLE

WHAT CAN IT NOT DO


```
1  // @flow
2
3  import {Map, List} from 'immutable'
4
5  type Player = Map<
6    · 'firstName' | 'lastName' | 'age' | 'id',
7    · string | number
8  >
9  type StateKeys = 'players' | 'status'
10 type StateValues = Map<string, Player> | string
11 type AppState = Map<StateKeys, StateValues>
12
13 function getAge(state: AppState): number {
14   · return state.getIn(['players', '123', 'age'])
15 }
16
```

LESSONS LEARNED

**Super easy to
start using in an
existing project**


**You don't have
to type-check
everything**

```
1 // @flow
2
3 function ageTimesTen(x) {
4   return `Your age times ten: ${x*10}`
5 }
```



```
1 // @flow
2
3 function getFoo(obj: Object): string {
4   return obj.foo
5 }
6
```

Avoid type laziness



```
1 // @flow
2
3 function getFoo(obj: {foo: string}): string {
4   return obj.foo
5 }
6
```

**Types are great
documentation**



THANK YOU



@tryggvigy

<https://speakerdeck.com/tryggvigy/this-will-flow-your-mind>