



TypeScript

International JavaScript Conference
April 2018





 @Srushtika



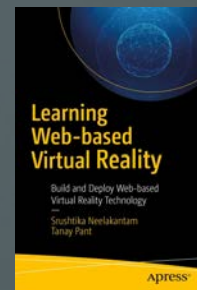
n.srushtika@gmail.com



Hi!

I'm Srushtika

Developer Advocate



TypeScript



Why?

JS

JAVASCRIPT

Hipster much?

In a regular project cycle



Write new code



Document



Maintain



Write new code

In a regular project cycle



Write new code



Document



Maintain



Write new code

TypeScript



What?

```
Srushtikas-Air:Ably srushtika$ node hello.js
/Users/srushtika/Ably/hello.js:2
console.log(customer_name.toLowerCase())
                        ^
```

```
TypeError: customer_name.toLowerCase is not a function
    at Object.<anonymous> (/Users/srushtika/Ably/hello.js:2:18)
    at Module._compile (module.js:398:26)
    at Object.Module._extensions..js (module.js:405:10)
    at Module.load (module.js:344:32)
    at Function.Module._load (module.js:301:12)
    at Function.Module.runMain (module.js:430:10)
    at startup (node.js:141:18)
    at node.js:980:3
```




```
let customer_name = 222  
console.log(customer_name.toLowerCase())
```

TypeScript



How?

Types

Interfaces

Classes & OOP

Migrating to TypeScript?

Here's how we did it at Ably

Understanding the problem

Realization

- Explosive growth in codebase
- Growth in the development team
- Long onboarding time
- Long and complex refactoring of existing code

Research

Flow vs TypeScript:

go.ably.io/typescript-vs-flow

Progressive Migration

- Setup TypeScript configuration
- Convert files that have most active development
- Use of a designated procedure to convert each file

Migration Steps

Refactoring and Documentation

Tips & Tricks

noImplicitAny

Object vs any



```
interface MyInterface {  
    b: Object  
}
```

```
let someVar = (a : MyInterface) => alert(a.b.nomethod())
```

Function overloading

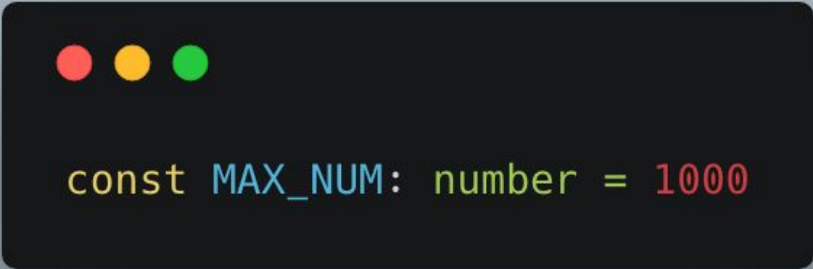
```
type Callback = (message: string) => void

/* overload list */
function hello (name: string, callback: Callback)
function hello (callback: Callback)

/* implementation */
function hello (name: string | Callback, callback?: Callback) {
  //logic
}
```

Function return types

const variables and types? Really?



```
const MAX_NUM: number = 1000
```

Interfaces with Classes

```
interface myInterface {
  myName: string
}
function Greeter(name: myInterface) {
  console.log('Hi' + name.myName)
}
class Conference implements myInterface {
  constructor(public name: string){}
}

const confA = new Conference('iJS')
Greeter(confA); /* 'Hi confA' */

const confB = { name: 'JavaScriptConf' };
Greeter(confB); /* 'Hi JavaScriptConf' */
```


John Diamond

<https://github.com/jdmnd>

Thankyou!

@Srushtika | Ably Realtime | iJS | 2018