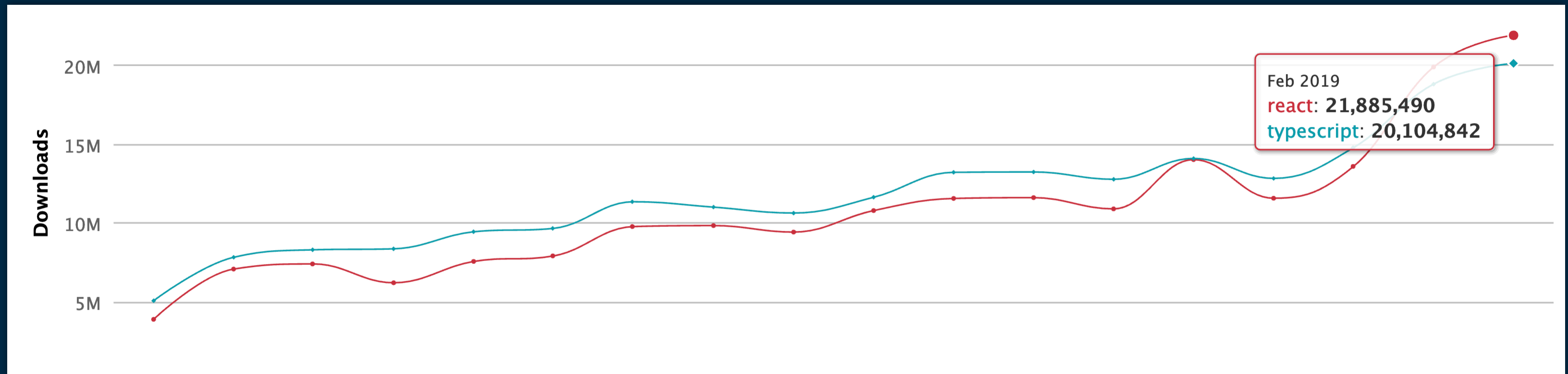


What's TypeScript?

- ▶ An open-source **typed, syntactic superset** of JavaScript, developed by **Microsoft**
- ▶ Compiles to readable JavaScript
- ▶ Comes in three parts: **Language, Language Server** and **Compiler**
- ▶ Works seamlessly with Babel 7

What's TypeScript?

- ▶ Since 2017...
 - ▶ +300% increase in downloads, now tied w/ React
 - ▶ 2018 NPM survey: 46% of respondents use TypeScript



...but why add types?

- ▶ Encode constraints and assumptions, as part of developer intent
- ▶ Catch common mistakes (i.e. incomplete refactors)
- ▶ Move some runtime errors to compile time
- ▶ Provide your consumers (including you) with a great DX

Type Systems & Type Equivalence



```
function validateInputField(input: HTMLInputElement) {
    /* ... */
}
```

```
validateInputField(x);
```

Can we regard **x** as an
HTMLInputElement?

- ▶ **Nominal Type Systems** answer this question based on whether **x** is an instance of a class/type named **HTMLInputElement**
- ▶ **Structural Type Systems** only care about the **shape** of an object.
This is how typescript works!

Object Shapes

JS

- ▶ When we talk about the **shape** of an object, we're referring to the names of properties and types of their values

Car

Make String
Model String
Year Number



interface **hello** {}
type **hello**

- ✓ Call, Construct and Index signatures
- ✓ "Open interfaces"
- ✓ Access modifier keywords
- ✓ Heritage clauses ("extends", "implements")

[./notes/3-interface-type-basics.ts](#)

Converting JS ▶ TS

3 STEPS FOR SUCCESS

1. Compiling in “loose mode”

- ▶ Start with tests passing
- ▶ Rename all .js to .ts, allowing implicit any
- ▶ Fix only things that are not type-checking, or causing compile errors
- ▶ **Be careful to avoid changing behavior**
- ▶ Get tests passing again

2. Explicit Any

- ▶ Start with tests passing
- ▶ Ban implicit any `("noImplicitAny": true,)`
- ▶ Where possible, provide a specific and appropriate type
 - ▶ Import types for dependencies from DefinitelyTyped
 - ▶ otherwise **explicit** any
- ▶ Get tests passing again

3. Squash explicit anys, enable strict mode

- ▶ Incrementally, in small chunks...
- ▶ Enable strict mode

```
"strictNullChecks": true,  
  "strict": true,  
  "strictFunctionTypes": true,  
  "strictBindCallApply": true
```
- ▶ Replace explicit anys w/ more appropriate types
- ▶ Try really hard to avoid unsafe casts