TS TypeScript

STATIC TYPING FOR JAVASCRIPT

PANU HORSMALAHTI

My TypeScript background

- Started using TypeScript around 0.9 release (2013)
- Shipped ~10 projects with TypeScript
- Maintainer of gulp-tslint (https://www.npmjs.com/package/gulp-tslint)

Typing in JavaScript

- Dynamic typing
- Academically: "static typing with a single type"
- Dynamic typing popular around 90s / early 00s (JavaScript, Python, Ruby, PHP…)

Dynamic typing problems

- Misses a class of errors, including:
  - typos
  - missing properties / functions
  - wrong parameters
  - objects missing properties
- Theory: "Well-typed programs cannot 'go wrong'." - Robin Milner
- No standard language for describing types, JSDoc often impractical
- Refactoring time consuming, difficult, dangerous and error prone
- Lack of proper intellisense/autocomplete/IDE UX
- Reading code is time consuming
- Allows bad code

Dynamic typing benefits

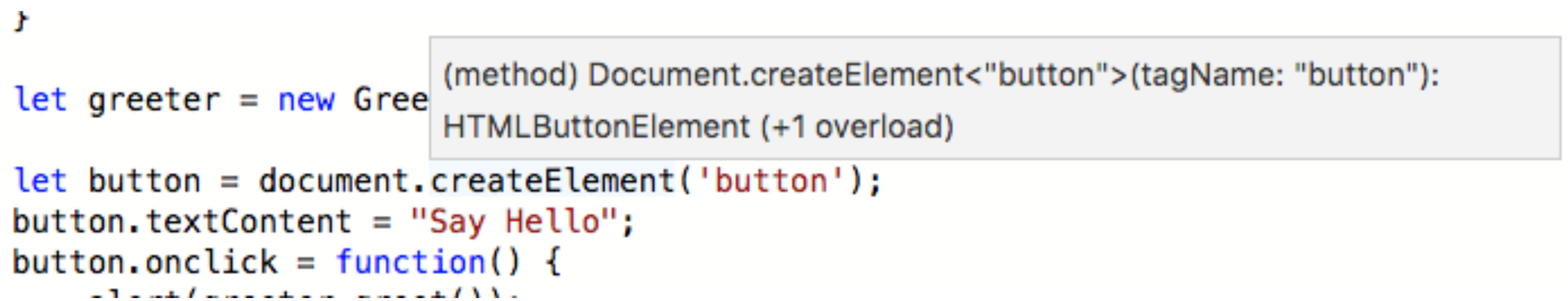
- Less typing ☺
- Suitable for small projects, trivial applications and small scripts
- Simpler language -> easier to learn
- No compilation

Static typing problems

- More complexity to learn when learning the language
- Compile/build chain needs to be configured
- TypeScript-specific: lack of type definitions, bad quality types for 3rd party libraries
- More time consuming to write and maintain types
- Fewer good alternatives for an IDE

Static typing benefits

- **Catches a large class of programming bugs**
- Unit tests can concentrate on the important stuff
- Proper intellisense / IDE integration
- Refactoring is easy
- Incentives to document external APIs, business domain concepts and interfaces

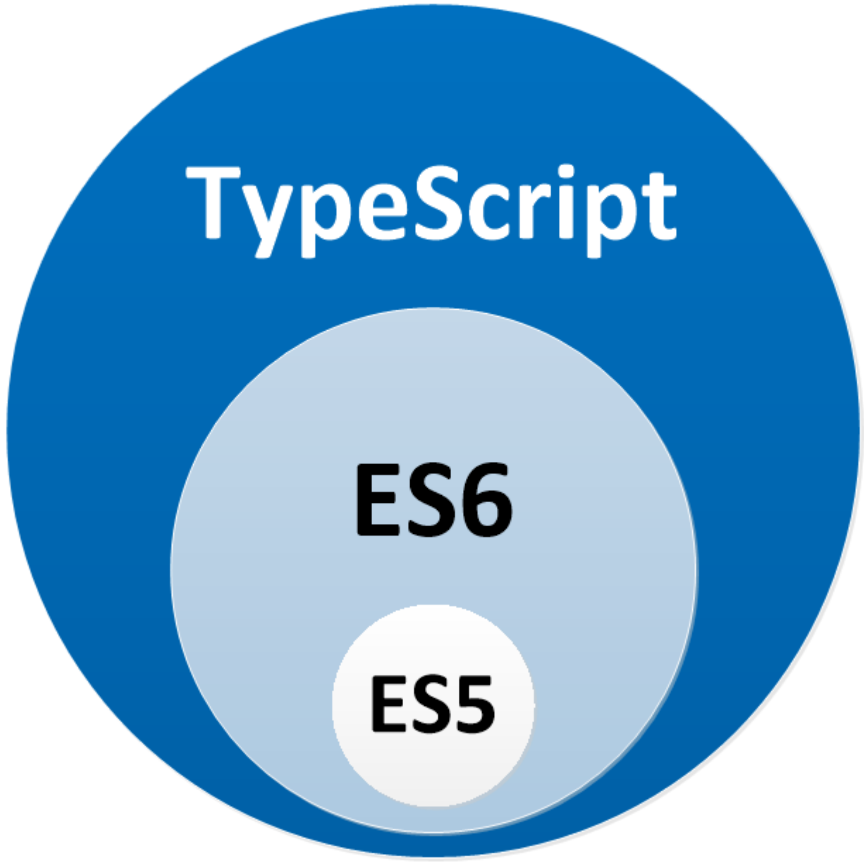


```
}

let greeter = new Gree

let button = document.createElement('button');
button.textContent = "Say Hello";
button.onclick = function() {
```

```
(method) Document.createElement<"button">(tagName: "button"):
HTMLButtonElement (+1 overload)
```

TypeScript

- Microsoft's statically typed "superset" of JavaScript
- Released in 2012
- Free and open source
- 2.3 just released, in active development
- Gradual, opt-in typing for JavaScript
- Compiles down to ES7/ES6/ES5/ES3 JavaScript
- Works with cli, grunt, gulp, babel, …
- Battle-tested, widely used in production
- Large scale JavaScript applications
- Compiler & language service
- tslint: TypeScript linter
- Angular >=2
- Alternatives: Flow



Basics

…

TypeScript - more

- .jsx support (.tsx)
- IDE plugins widely available (Atom, Sublime, Visual Studio Code, ...)
- TypeScript -> Babel pipeline
- Reads .js and .ts files
- --strict
- https://github.com/DefinitelyTyped/DefinitelyTyped
- Type definitions from npm

TypeScript – The Bad Parts

- Inadequate, missing, low-quality or outdated definitions
- Slightly slower build times
- Debugging
- Type inference problems (e.g. this.)

Future of typing & TypeScript

- Optimize performance using types (VMs already try)
- Chrome's strong mode / SoundScript experiment
- WebAssembly
- Reflection in TypeScript

What about Flow?

- Type checker for JavaScript
- Facebook's alternative to TypeScript
- Newer, less users
- TypeScript 2.x -> feature parity
- Angular2 / React
- IDE integration / tooling lacking
- Worse autocomplete
- Slightly better type safety
- Lacking typing definition files
- Documentation/tutorials WIP
- https://github.com/niieani/typescript-vs-flowtype

Questions / comments?