Statically Typed JavaScript

TypeScript and Flow
Make Software a competitive advantage

Custom Software Consulting and Development
Based on Your Business Goals and a Deep Understanding of the Technology Landscape
Multibillion-Dollar, Fortune 500 Organizations Trust New Venture Software

New Venture Software is an expert software consulting, custom software and user experience development company.

Defined by its passion for building software the right way, New Venture Software delivers amazing software experiences through technological innovation, thoughtful user experience design and flawlessly built software solutions.

References are available upon request.
Say NO to the slow and painful

DEATH BY POWERPOINT
Agenda

- Type
- Systems
- TypeScript
- Flow
- Q&A
What is a Type System?
\[
\frac{x : \sigma \in \Gamma \quad \tau = \text{inst} (\sigma)}{\Gamma \vdash x : \tau} \quad \text{[Var]}
\]

\[
\frac{\Gamma \vdash e_0 : \tau_0 \quad \Gamma \vdash e_1 : \tau_1 \quad \tau' = \text{newvar} \quad \text{unify}(\tau_0, \tau_1 \rightarrow \tau')}{\Gamma \vdash e_0 \; e_1 : \tau'} \quad \text{[App]}
\]

\[
\frac{\tau = \text{newvar} \quad \Gamma, x : \tau \vdash e : \tau'}{\Gamma \vdash \lambda x . e : \tau \rightarrow \tau'} \quad \text{[Abs]}
\]

\[
\frac{\Gamma \vdash e_0 : \tau \quad \Gamma, x : \tilde{\Gamma}(\tau) \vdash e_1 : \tau'}{\Gamma \vdash \text{let} \; x = e_0 \; \text{in} \; e_1 : \tau'} \quad \text{[Let]}
\]
What is a Type System?

Set of rules that assigns a property called type to the various constructs of a computer program, such as variables, expressions, functions or modules.
Nominal vs Structural

Nominal: compatibility and equivalence of data types is determined by explicit declarations and/or the name of the types.

Structural: type compatibility and equivalence are determined by the type's actual structure or definition.
**{Static vs Dynamic}**

**Static type checking** is the process of verifying the type safety of a program based on analysis of a program's text (source code).

**Dynamic type checking** is the process of verifying the type safety of a program at runtime.
YOU COME TO ME AT RUNTIME

TO TELL ME THE CODE YOU ARE EXECUTING DOES NOT COMPILE
What is TypeScript?

- Language created by Microsoft.
- Has optional static typing.
- Compiles to JavaScript.
- Inherits concepts from C#.
- Provides language service API.
It's always better to catch errors at compile time rather than at runtime.
Benefits of TypeScript

- Due to static typing, it's more predictable.
- Due to modules, namespaces and stronger OOP, it scales better for larger apps.
- Due to compilation step, some errors are caught compile-time, not run-time.
Installing TypeScript

Using the Node Package Manager.

`npm install --global typescript`
TypeScript is written in `.ts` files, which can't be used directly in the browser. It need to be compiled to vanilla `.js` first.

```
tsc main.ts
```
Specifies the way TS is compiled.
(autogeneratable with tsc --init)

```json
{ "compilerOptions": {
  "target": "es5", // Sets the output JS's version
  "module": "commonjs", // Sets the module loader
  "outDir": "dist", // Sets output JS files' location
  "sourceMap": true, // Allows debugging
  "noEmitOnError": true // Do not compile if errors
}
}
Language Features
Static Type System

“Strongly typed languages reduce bugs by 15%.”
Basic Types

predefined in the language:

number, string, boolean, Array, enum, undefined, null, tuples, any, void, never
Live Demo

SAY HELLO TO

MY TYPE CHECKER
Complex Types

created by the developer
class Employee {
    name: string;

    constructor(name: string) {
        this.name = name;
    }

    greet(): string {
        return 'Hi, my name is #{this.name}';
    }
}
interface MyInterface {
    member: number;
    optionalMember?: boolean;

    myMethod(param: string[]): number;
}

const instance: MyInterface = ...
{Live Demo}

Generics

Class<T>

Creating a component that can work over a variety of types rather than a single one.
Live Demo

{Modules}
Live Demo

Or. It. Didn't. Happen.
EcmaScript Next
Live Demo

TypeScript also provides tooling and language services for autocompletion, code navigation and refactoring.
Plugins for:

- Tide(Emacs)
- VS Code TypeScript Support
- TypeScript-Sublime-Plugin(Sublime Text)
Type Definition Files

{lib}.d.ts

Distributed via NPM

npm install --save @types/jquery
{TypeScript and Angular}
Flow is a **static type checker** for JavaScript.
Flow

- Developed by Facebook
- Not a compiler, but checker
- Goal: No runtime errors
Flow checks your code for errors through static type annotations.

```javascript
// @flow
function square(n: number): number {
    return n * n;
}

square('2'); // Error!
```
A lot of the time, **Flow** can understand your code without any types at all.

```
// @flow
function square(n) {
  return n * n;
}

square('2'); // Error!
```

*TypeScript doesn't catch this without the "noImplicitAny" flag.*
Type annotations can also be written as comments.

```javascript
// @flow
function strLen(x) /* : string */ {
  return x.length;
}
strLen('Hello, world!');
```
Live Demo

Flow only does type checking and relies on Babel or flow-remove-types or some other tool to remove type annotations.

TypeScript implements both a type checker and a compiler that emits plain JavaScript.
When developers argue about Static vs Dynamic Typing.
Should I use a type checker?

- there seems to be little or no impact on productivity
- initial effort to introduce a checker is low
- but: a type system is a complex thing
My recommendation

- if your project does not live for long: no
- if your project is really simple: no
- if there is a chance you will need to refactor the thing: yes
- if your system is very important or even crucial for the success of your company: yes
- if people enter or leave your team frequently: yes
Questions?
Thank You!

denis@newventuresoftware.com

github.com/deniskyashif

@deniskyashif