Topic 5 - TypeScript

From JavaScript to TypeScript

- Intro to TypeScript
- Getting Started - NPM
- ECMAScript 6 Importance
- Types and Functions
- Interfaces and ENUMs
- Class and Inheritance
- Modular Development
- Decorators

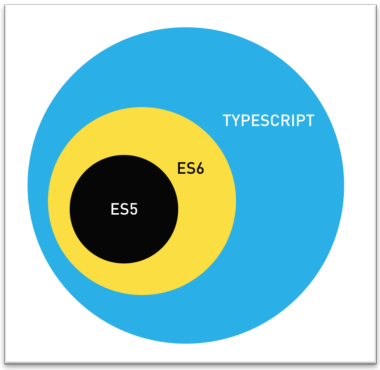
Intro to TypeScript

What is TS?

What is TypeScript?

Brief Overview

- Programming Language developed by Microsoft in 2012
- Typescript is a strongly typed version of JavaScript
  - a superset of JavaScript – any JS code is also TS code
  - TS compiles to plain JavaScript
- Open source and platform independent



What is TypeScript?

Motivation

- Motivation for TypeScript:
  - Provide an *optional type system* for JavaScript.
    - Implicit and Explicit types
    - Structural types
    - Prototype-based object chaining
  - Provide planned features from future JavaScript editions to current JavaScript engines

What is TypeScript?

TypeScript's History

- TypeScript was introduced by Microsoft in 2010 and made public in October 2012.
- Lacked browser and IDE support at first but soon development of plugins by open-source code editors such as Sublime, Atom, Emacs, etc.
- Visual Studio Code adopted the TypeScript compiler after the 2013 release (do not use any VS Code editions before 2013).

Advantages of TypeScript

- Low Risk: Compiled to JavaScript output
  - Easy to debug
  - Makes it simple to migrate to/from TypeScript
- Better structuring: modules, classes, interfaces
- ES6 features (classes, modules…)
- Optional types and IDE support
  - Compile time checks
  - Auto-complete
  - Automatic type inference

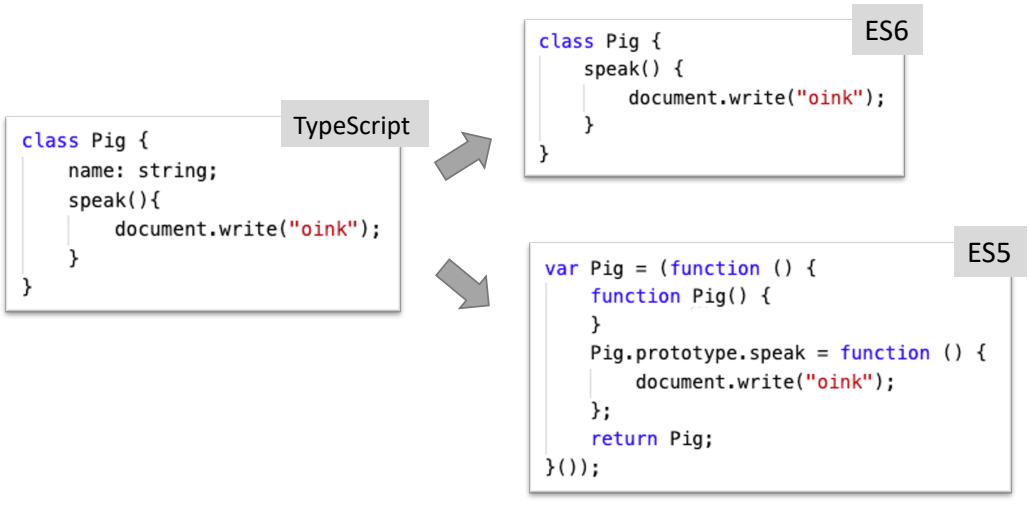
Disadvantages of TypeScript

- More difficult to learn (extra syntax)
- Extra compile step and source-mapping
- Leads to overconfidence
- Unnecessarily complicated for smaller projects

Getting Started with TypeScript

Transpiler

A JavaScript transpiler “compiles” TypeScript code into JavaScript code



Getting Started with TypeScript

Needed Software

- You will need a Text editor or IDEs with a TypeScript compiler. Luckily, it is readily available:

Getting Started with TypeScript

Software Installation

- Compiler Installation involves can be done in two ways

  1. Install IDE extension

     i.e. VSCode extension using Extensions > Manage Extensions

  2. Command-line interface via NPM (use a global install)

  3. Use TypeScript online playground

- For more info: https://www.typescriptlang.org/download

Getting Started with TypeScript

Software Installation – Option 2 - NPM

- For the command line interface option, you must install Node.js and NPM
- Node.js (https://www.nodejs.org)
  - A development framework based on Google's V8 JavaScript engine.
  - Code is written in JavaScript and then compiled into machine code by V8 to be executed.

Getting Started with TypeScript

Software Installation – Option 2 - NPM

- The **Node Package Manager** is a set of command line tools, or CLI's, that keep track of small software applications called packages
  - Packages are folders reusable code (much like what we've done so far) that can add extra functionality or to modularize your application
  - NPM is a quick way for developers to share code with other developers
  - Packages can be installed locally to your application or globally with your installation of Node.js
  - Essentially these packages make our lives easier

Getting Started with TypeScript

Software Installation – Option 2 - NPM

- When you install a package locally, NPM creates a node_modules folder if not exist.

```
npm install <package>
```

- NPM looks for a default file in the root of the application called "package.json". To create one, use the command

```
npm init
```

- To port your application, delete the node_modules folder and run

```
npm install
```

in the new environment

Getting Started with TypeScript

Software Installation – Option 2 - NPM

- NPM packages are for many different purposes. They can be even in the form of command line tool (like NPM itself)
- As a rule of thumb, install globally, those packages that you will be using in every application, like TS and a local server.
- To install globally, use the –g flag in the NPM command. This adds that package to the system path.

```
$ npm install -g typescript
$ npm install -g lite-server
```

NPM

Miscellaneous Tools

- Packages can come in many flavors:
  - For development purposes
  - App dependence
- An application dependence - the application will not run without these packages.
- A development dependency - that it is some utility that is required only during the development phase.
  - For example, tests, auto-compiling, transpilers, workflows, etc.

```
npm install --save-dev <package>
```


Getting Started with TypeScript

Software Installation – Option 3 – TS Sandbox

- **The TypeScript Playground**
  - TypeScript compiler written in TypeScript
  - http://www.typescriptlang.org/Playground/

Building a TypeScript Project

… via CLI

- The TypeScript compiler requires at least two files to compile:
  - The .ts file(s) – the file(s) that contains your TypeScript code
  - The tsconfig.conf file – contains the compiler options for the transpiler
- We can invoke the `tsc` command to compile our TypeScript code in JavaScript:

  ```
  $ tsc -w
  ```

  the `-w` flag indicates to keep a process alive and actively look for changes in the .ts files

Building a TypeScript Project

tsconfig.json file / local server

- The configuration for our TypeScript project goes in a file called `tsconfig.json`
- When it starts, the TypeScript compiler looks for this file in the root folder.
  - If the compiler finds it, it treats the whole folder and its subfolders as one big project
- Alternatively, we can tell TypeScript exactly which files to compile by using the "files" property in the tsconfig.json file.

Building a TypeScript Project

Local server

- If our TypeScript application requires special System privilege commands (i.e. importing modules), we will need to run a local server such as lite-server or webpack