Getting Started

Terms

Dynamically-typed Languages
IntelliSense
Refactoring
Source maps
Statically-typed Languages
Transpiling
Type safety

Summary

- Programming languages divide into two categories: statically-typed and dynamically-typed.
- In statically-typed languages (eg C++, C#, Java, etc), the type of variables is set at compile-time and cannot change later.
- In dynamically-typed languages (eg Python, JavaScript, Ruby), the type of variables is determined at run-time and can change later.
- TypeScript is essentially JavaScript with static typing and some additional features that help us write more concise and robust code.
- Most IDEs and code editors supporting TypeScript provide incredible IntelliSense and auto-completion. So we get active hints as we code. A great productivity booster!

- By providing type information in our code, we get better refactoring support in most IDEs and code editors.
- Refactoring means changing the structure of the code without changing its behavior.
- With TypeScript we can catch more bugs at compile time.
- Browsers don't understand TypeScript code. So we need to use the TypeScript compiler to compile and translate (or transpile) our TypeScript code into regular JavaScript for execution by browsers.
- Source maps are files that represent the mapping between TypeScript and JavaScript code. They're used for debugging.
- We can configure the TypeScript compiler by enabling/disabling various settings in tsconfig.json.