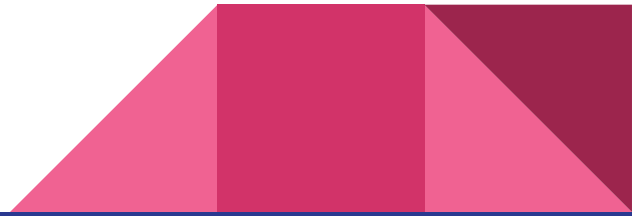
Learning TypeScript

Type-safe JavaScript

What is TypeScript?

- TypeScript is a syntactic superset of JavaScript which adds static typing.
- This basically means that TypeScript adds syntax on top of JavaScript, allowing developers to add types.
- TypeScript being a "Syntactic Superset" means that it shares the same base syntax as JavaScript, but adds something to it.



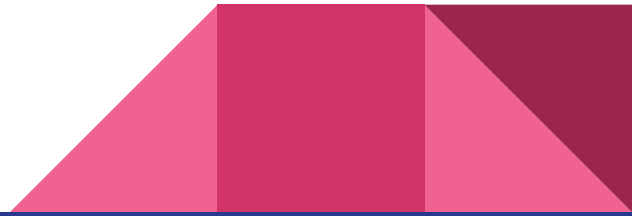
Why should I use TypeScript?

- JavaScript is a loosely typed language. It can be difficult to understand what types of data are being passed around in JavaScript.
- In JavaScript, function parameters and variables don't have any information! So developers need to look at documentation, or guess based on the implementation.
- TypeScript allows specifying the types of data being passed around within the code, and has the ability to report errors when the types don't match.
- For example, TypeScript will report an error when passing a string into a function that expects a number. JavaScript will not.

TypeScript uses compile time type checking. Which means it checks if the specified types match before running the code, not while running the code.

Why should I use TypeScript?

- A common way to use TypeScript is to use the official TypeScript compiler, which transpiler TypeScript code into JavaScript.
- Some popular code editors, such as Visual Studio Code, have built-in TypeScript support and can show errors as you write code.



Type Assignment

Type Assignment

When creating a variable, there are two main ways TypeScript assigns a type:

- Explicit
- Implicit

Type Assignment

Some Basic Types

- string
- number
- boolean

Type Assignment

Explicit Type

- Explicit type assignment are easier to read and more intentional.

```
let firstName: string = "Dylan";
```

Implicit Type

- Implicit TypeImplicit - TypeScript will "guess" the type, based on the assigned value:

```
let firstName = "Dylan";
```

Error In Type Assignment

Error In Type Assignment

- TypeScript will throw an error if data types do not match

```
let firstName: string = "Dylan"; // type string
firstName = 33; // attempts to re-assign the value to a different type
```

- Implicit type assignment would have made firstName less noticeable as a string, but both will throw an error:

```
let firstName = "Dylan"; // inferred to type string
firstName = 33; // attempts to re-assign the value to a different type
```

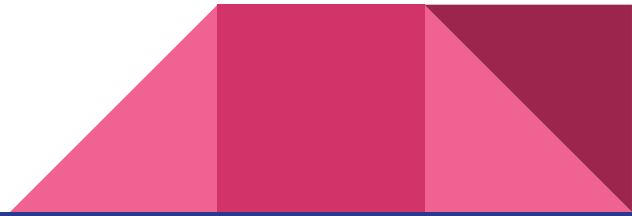
Note: Having TypeScript "guess" the type of a value is called infer.

Unable to Infer

Unable to Infer

- TypeScript may not always properly infer what the type of a variable may be. In such cases, it will set the type to any which disables type checking.

```
let rocker; // Type: any
```



Object Oriented Programming

Object Oriented Programming

Style of code, “how” we write and organize code

- Object-oriented programming (OOP) is a programming paradigm based on the concept of objects;

```
let rocker; // Type: any
```