TypeScript Basics

Brice Wilson

@brice_wilson www.BriceWilson.net

Overview

Declaring variables and constants

- var
- let
- const

Specifying types

Basic data structures

- enums
- arrays
- tuples

Declaring Variables with var, let, and const

var	let and const
Globally available in the function in which it is declared	Only available in the block in which it is declared
“Hoisted” to the top of the function	Not “hoisted” to the top of the block
Variable name may be declared a second time in the same function	Variable name may only be declared once per block

var Versus let

```
function ScopeTest() {
    if(true) {
        var foo = 'use anywhere';
        let bar = 'use in this block';
        // do some more stuff
    }
    console.log(foo); // works!!
    console.log(bar); // error!!
}
```

Basic Types

- Boolean
- Number
- String
- Array
- Enum
- Any
- Void

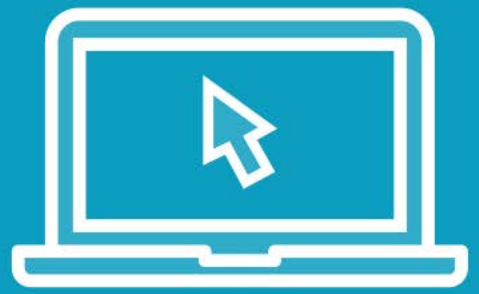
Type Inference

```
let myString = 'this is a string';
myString = 42; // error!!

function ReturnNumber() {
    return 42;
}
let anotherString = 'this is also a string';
anotherString = ReturnNumber(); // error!!
```

Adding Type Annotations

```
let myString: string = 'this is a string';
myString = 42; // error!!

function ReturnNumber(): number {
    return 42;
}
let anotherString: string = 'this is also a string';
anotherString = ReturnNumber(); // error!!
```

Demo



Declaring variables and constants

Adding type annotations

Enums

```
enum Category { Biography, Poetry, Fiction }; // 0, 1, 2
enum Category { Biography = 1, Poetry, Fiction }; // 1, 2, 3
```

Enums

```
enum Category { Biography, Poetry, Fiction }; // 0, 1, 2
enum Category { Biography = 1, Poetry, Fiction }; // 1, 2, 3
enum Category { Biography = 5, Poetry = 8, Fiction = 9 }; // 5, 8, 9

let favoriteCategory: Category = Category.Biography;
```

Enums

```
enum Category { Biography, Poetry, Fiction }; // 0, 1, 2
enum Category { Biography = 1, Poetry, Fiction }; // 1, 2, 3
enum Category { Biography = 5, Poetry = 8, Fiction = 9 }; // 5, 8, 9

let favoriteCategory: Category = Category.Biography;

console.log(favoriteCategory); // 5
let categoryString = Category[favoriteCategory]; // Biography
```

```
let strArray1: string[] = ['here', 'are', 'strings'];

let strArray2: Array<string> = ['more', 'strings', 'here'];

let anyArray: any[] = [42, true, 'banana'];
```

Arrays

Can be declared two different ways

Accessed and used much like JavaScript arrays

Declare as an array of “any” to store any type in the same array

```
let myTuple: [number, string] = [25, 'truck'];
```

Tuples

Array where types for first few elements are specified

Types do not have to be the same

```
let myTuple: [number, string] = [25, 'truck'];
let firstElement = myTuple[0]; // 25
let secondElement = myTuple[1]; // truck


// other elements can have numbers or strings
myTuple[2] = 100;
myTuple[2] = 'this works!’;
```

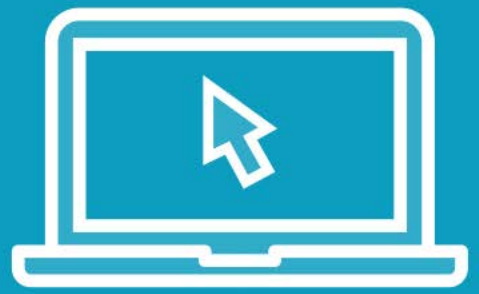
Tuples

Array where types for first few elements are specified

Types do not have to be the same

Additional elements can be any type from those previously specified

Demo



Using enums

Declaring arrays

Summary

Declare variables

Specify types

Use enums and arrays