Using Variables, Types, and Enums



Dan Wahlin
@DanWahlin

John Papa @John_Papa



Overview



Benefits of types

Built-in types

Creating and using typed variables

Creating and using enums

Benefits of Types







Benefits of Types

What is the result of calling the add() function?

```
add.js

function add(x, y) {
   return x + y;
}

const result = add(2, '2');
```

result is 22

How Can TypeScript Types Help?

TypeScript allows types to be added to variables and parameters.

Adding TypeScript "guardrails"

```
function add(x: number, y: number) {
    return x + y;
}

Argument of type 'string' is not assignable to parameter of type 'number'. ts(2345)

View Problem (\text{\text{YF8}} \text{ No quick fixes available}

const result = add(2, '2');
```

Built-in Types



Built-in Types

string

boolean

number

array



Defining Types

```
variableName: type
let firstName: string;
```

function add(x: number, y: number) { }

Pattern for defining a type

Define a variable with a type

Define parameters with a type

Using Built-in TypeScript Types

TypeScript types can be used on variables and parameters following the *variableName*: *type* pattern.



Types Add Development Guardrails



Additional Built-in Types

undefined null void any

Additional Types

```
let product: any;
let firstName: string | undefined | null;
function log(msg: string) : void { }
```

Variable can hold any value

Define a "union type"

Function returns no value

Using Additional TypeScript Types

Additional types such as any, void, undefined, null, and others exist in TypeScript.

In cases where a variable can be one or more types a union type can be used.



```
enum ProductType {
    Sports,
    HomeGoods,
    Groceries
}

let productType = ProductType.Sports;
```

Enums represent a set of named constants

Select from a set of values

Using Enums

An enum represent a set of name constants.

Enums are not a "type-level" extension of JavaScript. They generate JavaScript code that is used at runtime.

Creating and Using Typed Variables



Creating and Using Enums



Summary



TypeScript provides "guardrails" that can help catch errors early

Key built-in types include:

- string
- number
- boolean
- array
- any

Enums minimize "magic strings" in code

