

TypeScript

...

Objectives

- Review Resume/Profile Lab
- Set up Cloud9 for TypeScript Development
- What is TypeScript
- Variables - let & const
- Data Types
- Arithmetic Operators
- Comparison Operators
- Logical Operators
-
- TypeOf

Set Up Cloud9 for TypeScript Development

Step 1 - Create a new BLANK application in cloud9

Step 2 - npm init : we will hit enter all the way until we have to type yes

Step 3 - touch mycode.ts

Step 4 - Inside the package.json file we will change main to the name of our .ts file

Step 5 - Type some typescript code > save > and do tsc mycode.ts > run >

Step 6 - Create a tsconfig file by doing tsc --init in terminal

Step 7 - watch mode tsc mycode.ts --w

What is TypeScript?

TypeScript is a superset of JavaScript which primarily provides optional static typing Classes, and Interfaces. One of the big benefits is to enable IDE's to provide a richer environment for spotting common errors as you type the code. TypeScript compiles to JavaScript.

What is a compiler?

A **compiler** is a software program that transforms source code written by a developer in a high level programming language into low level object code(binary code) in machine language, which can be understood by the processor

Variables: let & Const

What is a variable?

A variable is a value that can change, depending on conditions or on information passed to the program. It's like a variable like a box to hold something that can change!

There are two ways of declaring variables in TypeScript.

Let - used to hold variables that can change.

Const - used to hold variables that do not change

Example of using let to declare a variable

```
let myName: string = "Mikaila";  
console.log(myName);
```

How we print something to the console?

Example of using const to declare a variable

```
const myAge: number = 19;
```

```
console.log(myAge);
```

If we try to assign another age to this variable the compiler will give us an error

We have several data types in TypeScript such as ...

String: used to define text or letters

Number: used to defined variables that must be numbers or decimals

Boolean: used to hold data that will be True or False

Any: used to hold data types that can change to any type such as string or number

Array: used to hold a list of data with indexes starting at 0

Tuples: are like arrays but with mixed types and the order is important

Enums: used to make numbers more expressive

Arithmetic Operators

Addition Plus sign (+)

Subtraction Minus sign (-)

Multiplication Multiplication sign (*)

Division Division sign (/)

Modulus Remainder (%)

PostFix Increment and Decrement (x++) & x(--)

Prefix Increment and Decrement (++x) & (--x)

Comparison Operators - Assuming X = 5

`x > 10` = false

`x < 10` = true

`x >= 5` = true

`x <= 100` = true

`x == "5"` = true //type coercion it turns them into same format then compares

`x === "5"` = false // does not convert it compares strings and numbers

`x != b` = true

`x !== "5"` = true

Logical Operators

`&&` AND both sides need to be true

`||` OR One side needs to be true

`!` NOT if something was true it makes it false (vice versa)

Using Operators for calculations

P. E. D. M. A. S rules applies -

The grouping operator () controls the precedence of evaluation of expressions

```
Let x : number = 500;
```

```
Let y : number = 400;
```

```
Let z : number = 10;
```

```
Let a : number = 2;
```

```
Let Answer = (x - y + z ) * a ; // How to fix ?
```

```
Console.log(Answer);
```

Ternary Condition Operator

It takes three operands such as condition ? val1 : val2

```
Let age: number = 20;
```

```
Let status : number = ( age >= 21 ) ? 'can drink' : 'cannot drink';
```

```
console.log(status);
```

Conditional statements { if statements }

We have two main conditional statements in TypeScript.
The if else statement and a switch statement

IF STATEMENT

```
If (true) {  
  console.log(true);  
} else{  
  console.log(false);  
}
```

Using the OR || Comparison Operator

```
let n: string = 'netflix';
```

```
let h: string = 'hulu';
```

```
let userInput: string = n;
```

```
if( (userInput == n) || (userInput == h) ){
```

```
    console.log('I will be streaming movies on netflix OR hulu this weekend' );
```

```
}else{
```

```
    console.log('I will be studying this weekend');
```

```
}
```

Using the AND && Comparison Operator

```
let n: string = 'netflix';  
let h: string = 'hulu';  
let userInput1: string = 'netflix';  
let userInput2: string = h;  
  
if((userInput1 == n) && (userInput2 == h)){  
    console.log(' I will be streaming movies on netflix and hulu this weekend ');  
}else{  
    console.log('I will be studying this weekend');  
  
}
```


Switch statement

```
switch(value){  
    Case 0:  
    console.log(print something.....);  
    break;  
  
    Case 1:  
    console.log(print something else.....);  
    Break;  
  
    Default:  
    console.log(print something else.....);  
    break  
}
```

TypeOf

Since static typing is optional in TypeScript you may come across data and need to know the type of data it is. That's when we use TypeOf as a mechanism to check

```
let myName = "Mikaila";
```

```
console.log(typeof(myName));
```

will print String