Introduction to Typescript

Class 10 Course Content

Preparation

GOALS

By the end of this lesson, you will be able to:

1. **Install Typescript Globally or Locally**
2. **Comprehend Primitive Types in JavaScript**
3. **Implement Complex Types like arrays & objects**
4. **Utilize Typescript Generics**
5. **Type Classes Using Interfaces**

CONCEPTS

- **Typescript**: *Typescript* is a superset of Javascript, adding additional features such as static typing. Typescript helps developers to develop large applications to improve error catching and standardization.

Walkthrough

Go through the starter code

STEP 1: INSTALLING TYPESCRIPT

Aim: Install Typescript on your computer

| *https://www.typescriptlang.org/* |

- [ ] **Walkthrough downloading Typescript**
  - Go to Typescriptlang.org
  - Click on install typescript on your computer via `npm`
  - Read the documentation

🔻

| *Terminal* |

- [ ] **Walkthrough Installing Typescript locally**
  - Navigate to the correct path
  - Initialize your code using `npm init -y`
  - Install typescript locally using `npm install typescript`
  - Change the `main.js` file to a typescript file by replacing `.js` with `.ts`
  - Update the HTML to point to `main.ts` in the `<script>` tag `src` attribute

- Invoke the typescript compiler using `npx tsc main.ts`
- Add the type of `number` to both arguments in the `addNumbers` function
- Comment out or Delete the lines giving errors & compile your code again

```
npm init -y

npm install typescript
```

🔻

Check: Ensure you have correctly downloaded Typescript

- Should you install Typescript globally or locally? Why?
- What command allows you to compile a Typescript file into JavaScript?

STEP 2: BASE TYPES & PRIMATIVES

Aim: Learn the basics of Typescript types

| *./basics.ts* |

- Create the `basics.ts` file
- ☐ **Walkthrough Variables with the "number" Type**

```
// * PRIMATIVE TYPES START * \\
// Numbers: 0 1 2 3
let numOfStudents: number;

numOfStudents = 2;
// numOfStudents = "3"
```

🔻

- ☐ **Walkthrough Variables with the "string" Type**

```
// Strings: ""
let username: string;

username = "";
username = "123";
// username = 123
username = "Will_Wilder";
```

🔻

- [ ] **Walkthrough Variables with the "boolean" Type**

```
// Booleans: true or false
let isInstructor: boolean;

isInstructor = false;
isInstructor = true;
```

🔻

Check: Ensure you understand the primative Typescript types

- What are the main three Primitive Typescript Types?
- Should you use lowercase or uppercase to define a type in Typescript?

STEP 3: ARRAY & OBJECT TYPES

Aim: Learn the complex Typescript types such as objects & arrays

| *./complex.ts* |

- Create the `complex.ts` file
- [ ] **Walkthrough variables holding array values**

```
// * COMPLEX TYPES START * \\
// Arrays
let students: string[];

students = ["Quinton", "Betty", "Mary"];

let mixedBag: any[];

mixedBag = [2, "Hello", false];
```

🔻

- [ ] **Walkthrough variables containing object variables**

```
// Objects
let student: {
  name: string;
  age: number;
  isStudent: boolean;
};

student = {
  name: "James",
```

```
  age: 33,
  isStudent: true,
  //   hasPets: true
};
```

- [ ] **Walkthrough type Inference & Union types**

```
// Type inference & Unions
let course = "Codefi Coding Bootcamp";
// course = 123 // Type intferences causes an error

let bootcamp: string | number;
bootcamp = "Codefi Coding Bootcamp";
bootcamp = 123;
```

- [ ] **Walkthrough type Aliases**

```
// Type Aliases
type CodefiPerson = {
  name: string;
  age: number;
  isStudent: boolean;
};

let randomStudent: CodefiPerson = {
  name: "James",
  age: 33,
  isStudent: true,
};
let currCodeCoach: CodefiPerson = {
  name: "Chris",
  age: 99,
  isStudent: false,
};
```

- [ ] **Walkthrough typing functions & parameters**

```
// Functions & Parameters
function add(a: number, b: number): number {
  return a + b;
}
```

```
function printName(name: string): void {
  console.log(name);
}
```

🔻

Check: Ensure you understand the complex Typescript types

- How do you define an array full of only strings?
- What is the name for when a value can have two different types
- Explain explicit Typing.
- What is a "Type Alias"?

STEP 4: TYPESCRIPT GENERICS

Aim: Learn about and utilize Typescript Generics

| *./complex.js* |

- ☐ **Walkthrough Generics**

```
// * GENERICS START * \\
function getId<Type>(value: Type): Type {
  return value;
}

let userOneId = getId<String>("stringId_userOne");
let userTwoId = getId<Number>(34);
let userThreeHasId = getId<Boolean>(true);
```

🔻

Check: Ensure you understand Generics and can implement a simplified version

- When do you want to use Generics?
- What is one "real life" use-case for Generics? (You may need to look this one up!)

STEP 5: CLASSES & INTERFACES IN TYPESCRIPT

Aim: Implement Classes & Interfaces in Typescript

| *./complex.ts* |

- ☐ **Walkthrough Creating a Student Class**

```
// * CLASSES & INTERFACES START * \\
class Student implements StudentInterface {
  constructor(
```

```
    public first: string,
    public last: string,
    private courses: string[]
  ) {}

  enroll(courseName: string) {
    this.courses.push(courseName);
  }

  listCourses() {
    return this.courses.slice();
  }
}

const studentOne = new Student("Will", "Wilder", ["Codefi Front-End
Bootcamp"]);
studentOne.enroll("Codefi Back-End Bootcamp");

// studentOne.courses
studentOne.listCourses();

console.log("studentOne:", studentOne);
// CLASSES & INTERFACES END \\
```

🔻

- [ ] **Walkthrough Implementing a StudentInterface**

```
interface StudentInterface {
  first: string;
  last: string;
  enroll: (courseName: string) => void;
  listCourses: () => string[];
}
```

🔻

Check: Ensure you have correctly downloaded Typescript

- What is the difference between "Type Aliases" and "Type Interfaces"?

Review

ACCOMPLISHMENTS

Congratulations yet again! 🎊 🎉

Feel proud that **you learned something new and valuable today**.

Learning to code is a journey, and you are taking the necessary steps to improve your skills and opportunities for the future.

Good on you!

Specifically, we learned how to:

- Install Typescript
- Implement basic and advanced Typescript Types
- Know when and where to use specific types and interfaces

RESOURCES

Typescript Documentation *(Articles)*

Typescript - the Basics *(Video)*

TypeScript Course for Beginners 2021 - Learn TypeScript from Scratch! *(Course)*
