TypeScript Notes

* Typescript is an open-source, object-oriented language
* Typescript is a superset of JavaScript
* It extends JavaScript by adding data types, classes, and other object-oriented features with type-checking
* It supports object-oriented programming features like data types, classes, enums, etc.

Features of TypeScript

1. **Cross-Platform:** TypeScript runs on any platform that JavaScript runs on. The TypeScript compiler can be installed on any Operating System such as Windows, macOS, and Linux.
2. **Object-Oriented Language**: TypeScript provides powerful features such as Classes, Interfaces, and Modules. You can write pure object-oriented code for client-side as well as server-side development.
3. **Static type-checking**: TypeScript uses static typing. This is done using type annotations. It helps type checking at compile time. Thus, you can find errors while typing the code without running your script each time. Additionally, using the type inference mechanism, if a variable is declared without a type, it will be inferred based on its value.
4. **ES 6 Features**: TypeScript includes most features of planned ECMAScript 2015 (ES 6, 7) such as class, interface, Arrow functions etc.

TypeScript Variable

* TypeScript follows the same rules as JavaScript for variable declarations. Variables can be declared using: var, let, and const.

var

* Variables in TypeScript can be declared using var keyword, same as in JavaScript. The scoping rules remains the same as in JavaScript.

let

* The let declarations follow the same syntax as var declarations. Unlike variables declared with var, variables declared with let have a block-scope. This means that the scope of let variables is limited to their containing block, e.g. function, if else block or loop block.

**Advantages of using let over var**

1. Block-scoped let variables cannot be read or written to before they are declared.
2. Let variables cannot be re-declared
3. The TypeScript compiler will give an error when variables with the same name (case sensitive) are declared multiple times in the same block using let.

**Const**

* Variables can be declared using const similar to var or let declarations. The const makes a variable a constant where its value cannot be changed. Const variables have the same scoping rules as let variables.
* const num:number = 100;
* Const variables must be declared and initialized in a single statement. Separate declaration and initialization is not supported.

**TypeScript - Functions**

* Functions are the primary blocks of any program. In JavaScript, functions are the most important part since the JavaScript is a functional programming language. With functions, you can implement/mimic the concepts of object-oriented programming like classes, objects, polymorphism, and, abstraction.
* Functions ensure that the program is maintainable and reusable, and organized into readable blocks. While TypeScript provides the concept of classes and modules, functions still are an integral part of the language.
* In TypeScript, functions can be of two types: named and anonymous.

**Named Functions**

* A named function is one where you declare and call a function by its given name.

**Anonymous Function**

* An anonymous function is one which is defined as an expression. This expression is stored in a variable. So, the function itself does not have a name. These functions are invoked using the variable name that the function is stored in.

**Function Parameters**

* Parameters are values or arguments passed to a function. In TypeScript, the compiler expects a function to receive the exact number and type of arguments as defined in the function signature. If the function expects three parameters, the compiler checks that the user has passed values for all three parameters i.e. it checks for exact matches

# TypeScript - Classes

* TypeScript introduced classes to avail the benefit of object-oriented techniques like encapsulation and abstraction. The class in TypeScript is compiled to plain JavaScript functions by the TypeScript compiler to work across platforms and browsers.
* A class can include the following:
* Constructor
* Properties
* Methods

**Constructor**

* The constructor is a special type of method which is called when creating an object. In TypeScript, the constructor method is always defined with the name "constructor".
* An object of the class can be created using the new keyword:
* let emp = new Employee();

## **Inheritance**

Just like object-oriented languages such as Java and C#, TypeScript classes can be extended to create new classes with inheritance, using the keyword extends