TypeScript

TypeScript is a typed superset of JavaScript that compiles to plain JavaScript.

TypeScript is pure object oriented with classes, interfaces and statically typed like C# or Java.

The popular JavaScript framework Angular 2.0 is written in TypeScript.

Mastering TypeScript can help programmers to write object-oriented programs and have them compiled to JavaScript, both on server side and client side.

Whitespace and Line Breaks

TypeScript ignores spaces, tabs, and newlines that appear in programs. You can use spaces, tabs, and newlines freely in your program and you are free to format and indent your programs in a neat and consistent way that makes the code easy to read and understand.

TypeScript is Case-sensitive

TypeScript is case-sensitive. This means that TypeScript differentiates between uppercase and lowercase characters.

Semicolons are optional

Each line of instruction is called a statement. Semicolons are optional in TypeScript.

class Greeting {

greet():void {

console.log("Hello World!!!")

}

}

var obj = new Greeting();

obj.greet();

Data Types

Bult-in types=number,void,string,null,boolean

Void

A void is a return type of the functions which do not return any type of value. It is used where no data type is available. A variable of type void is not useful because we can only assign undefined or null to them. An undefined data type denotes uninitialized variable, whereas null represents a variable whose value is undefined.

Any Type

It is the "super type" of all data type in TypeScript. It is used to represents any JavaScript value. It allows us to opt-in and opt-out of type-checking during compilation. If a variable cannot be represented in any of the basic data types, then it can be declared using "Any" data type. Any type is useful when we do not know about the type of value (which might come from an API or 3rd party library), and we want to skip the type-checking on compile time.

User-Defined Data Type=array,class,tuple,enum,functions,interface

Tuple

The Tuple is a data type which includes two sets of values of different data types. It allows us to express an array where the type of a fixed number of elements is known, but they are not the same. For example, if we want to represent a value as a pair of a number and a string, then it can be written as:

// Declare a tuple

let a: [string, number];

// Initialize it

a = ["hi", 8, "how", 5]; // OK

=======================

Interview Questions

**1) What is Typescript?**

TypeScript is a free and open-source programming language developed and maintained by Microsoft. It is a strongly typed superset of JavaScript that compiles to plain JavaScript. It is a language for application-scale JavaScript development. TypeScript is quite easy to learn and use for developers familiar with C#, Java and all strong typed languages.

TypeScript can be executed on Any browser, Any Host, and Any Operating System. TypeScript is not directly run on the browser. It needs a compiler to compile and generate in JavaScript file. TypeScript is the ES6 version of JavaScript with some additional features.

2) How is TypeScript different from JavaScript?

TypeScript is different from JavaScript in the following manner:

|  |  |  |
| --- | --- | --- |
| **SN** | **JavaScript** | **TypeScript** |
| 1 | It was developed by Netscape in 1995. | It was developed by Anders Hejlsberg in 2012. |
| 2 | JavaScript source file is in ".js" extension. | TypeScript source file is in ".ts" extension. |
| 3 | JavaScript doesn't support ES6. | TypeScript supports ES6. |
| 4 | It doesn't support strongly typed or static typing. | It supports strongly typed or static typing feature. |
| 5 | It is just a scripting language. | It supports object-oriented programming concept like classes, interfaces, inheritance, generics, etc. |
| 6 | JavaScript has no optional parameter feature. | TypeScript has optional parameter feature. |
| 7 | It is interpreted language that's why it highlighted the errors at runtime. | It compiles the code and highlighted errors during the development time. |
| 8 | JavaScript doesn't support modules. | TypeScript gives support for modules. |
| 9 | In this, number, string are the objects. | In this, number, string are the interface. |
| 10 | JavaScript doesn't support generics. | TypeScript supports generics. |