Typescript is a superset of **JavaScript** which primarily provides optional static typing, classes, and interfaces. We have written some Refined interview questions on Typescript that helps developers to crack interview on TypeScript.

Top Typescript Interview Questions and Answers for Freshers and Experienced

Below are the list of some **Typescript Interview Questions and their answers** that may be asked by an interviewer in your interview

Q1. What is Typescript? How is it different from javascript?

Typescript is a free and open-source programming language which is designed and developed by Microsoft. It was designed by **Anders Hejlsberg** at **Microsoft**. Typescript is a **superset of JavaScript** which primarily provides optional static typing, classes and interfaces. Typescript is compiled to provide clean and simple JavaScript code which runs on any browser. It allows JavaScript developers for using highly productive development tools and practices like static checking and code refactoring.

Differences between Typescript and JavaScript are

1. JavaScript don't support Es6 while Typescript supports .
2. JavaScript build up reusable components by using unctions and prototype-based inheritance while Typescript supports Classes that allow programmer to think in more object oriented way .
3. JavaScript don't have any interfaces while Typescript has interfaces.
4. There is no static typing in JavaScript whereas there is static typing in Typescript.
5. JavaScript has no optional parameter feature while Typescript has optional parameter feature.

Q2. List some features of Typescript?

Features of Typescript are:-
- Typescript can be compiled to all major versions of Javascript(ES3,ES5,ES6,ES7).
- Typescript can be used for cross –browser development and is an open source project.
- Typescript is a superset of JavaScript that provides typed nature to your code.
- It is used to avoid the disadvantages of JavaScript like Typescript is designed to handle the needs of complex programs.
- Typescript was debuted as a beta programming language on October 1, 2012 and since then has gone through many versions with improved capabilities.
- Another key feature of Typescript is in version Typescript 2.6 that covers error suppression comments.
- Typescript is more predictable and is generally easier to debug.

Q3. List some benefits of using Typescript?

Following are some benefits of using Typescript

1. One of the biggest advantages of Typescript is its code completion and intelligence.
2. It provides the benefits of optional static typing .Here Typescript provides types that can be added to variables, functions, properties etc.
3. Typescript has the ability to compile down to a version of JavaScript that runs on all browsers.
4. Typescript tries to extend JavaScript. Compiler generates JavaScript.
5. Typescript is a backward compatible version of JavaScript that compiles to pure JavaScript which makes their writing easier and faster.
6. Typescript is purely object oriented programming which means it is a programming paradigm based on the concepts of objects.
7. Most important advantage is it offers a "compiler" that can convert to JavaScript equivalent code. And it has a concept of namespace defined by a "module".

Q4. Who developed Typescript and what is the current stable version of Typescript?

Typescript was developed by **Anders Hejlsberg** in 2012, who is the lead architect of C# and the creator of Delphi and Turbo Pascal. Typescript is a free and open source programming language maintained by Microsoft. It is a superset of JavaScript. It was developed for the development of large application. The current stable version of this programming language is **2.7.0-beta-20171212** which was released on **December 12, 2017**. Typescript compiles to simple JavaScript code which runs on any browser that supports ECMAScript 3 framework. It offers support for the latest and evolving JavaScript features.

Q5. Tell the minimum requirements for installing Typescript. Also, mention the steps involved in it.

Installing Typescript with the help of node and npm is recommended. Here, npm is used to install all the libraries and tools. Typescript should be installed globally using
```
npm install -g typescript
```

It installs a command line code **"tsc"** which will further be used to compile your Typescript code. Make sure that you check the version of Typescript installed on your system.

Following steps are involved for installing Typescript on your system:

1. Download and run the .msi installer for node.
2. Enter the command "node –v" to check if the installation was successful.
3. Type the following command in the terminal window to install Typescript:

```
npm install -g typescript
```

Q6. List the built-in types in Typescript.

These are also called the primitive types in Typescript. These are of 5 types: –

1. **Number type:** it is used to represent number type values and represents double precision floating point values.

   Syntax- var variable_name: number;

2. **String type:** it represents a sequence of characters stored as Unicode UTF-16 code. It is the same as JavaScript primitive type.

   Syntax- var variable_name: string;

3. **Boolean type**: in Typescript, it is used to represent a logical value. When we use the Boolean type, we get output only in true or false. It is also the same as JavaScript primitive type.

   Syntax- var variable_name: bool;

4. **Null type:** it represents a null literal and it is not possible to directly reference the null type value itself.

   Syntax- var variable_name:number=null;

5. **Undefined type:** it is the type of undefined literal. This type of built-in type is the sub-type of all the types.

   Syntax- var variable_name:number=undefined;

Q7. What are variables in Typescript? How to create a variable in Typescript?

A variable is a named space in the memory which stores a particular value. While declaring a variable in Typescript, certain rules should be followed like-
1. A variable name should contain alphabets and numeric digits.
2. It cannot contain spaces and special characters except underscore (_) and dollar ($) sign.
3. A variable name cannot begin with a digit.

You should always keep in mind, that a variable should be declared before being used. Variables in Typescript are declared by placing a "var" keyword prior to the variable name. A variable can be declared using 4 methods: –

- Declare its type and value in one statement.**Syntax-** var variable_name:string = value;
- Declare its type but no value.**Syntax-** var variable_name:string;
- Declare its value but no type.**Syntax-** var variable_name = value;
- Declare neither value nor type.**Syntax-** var variable_name;

Q8. What do you mean by interfaces? Explain them with reference to Typescript.

An interface is a syntactical contract which defines the syntax of any entity. Interfaces in Typescript define properties, methods, and events which are the members of the interface. It contains only the declaration of the members. It is responsible for deriving classes to define the members. It often helps in providing a standard structure that the deriving classes would follow. For declaring an interface, we make use of the "interface" keyword.

Syntax-

```
interface interface_name{
Statements;
}
```

Interfaces need not to be converted to JavaScript for execution. They have zero runtime JavaScript impact. Interfaces can define both the kind of keys which an array uses and the type of entry it contains.

Q9. How to compile a Typescript file?

A Typescript file is used to write more manageable and complex code in AppBuilder but before that, they need to be compiled to JavaScript. Here are the steps which are followed while compiling a Typescript file to JavaScript: –
1. It should be firstly verified that the Typescript engine has been enabled or not. In the title bar, click your username and select options.
2. In the project navigator, select and right-click the TS files that are to be compiled.
3. Choose compile to JavaScript.
4. Add a <script> reference for the JS file in the HTML code if needed.
5. To compile a typescript file via command line **tsc <TypeScript File Name>** command is used.

Q10. What do you understand by classes in Typescript? List some features of classes.

We all know that Typescript is a type of object-oriented JavaScript and supports object-oriented programming features like- classes, interfaces, etc. Here, a class in terms of object-oriented programming is a blueprint for creating objects. A class is used to encapsulate data for the object. A built-in support is provided by Typescript for this feature. The "class" keyword is used to declare a class in Typescript.
Example

```
class Greeter {
    greeting: string;
    constructor (message: string) {
        this.greeting = message;
    }
    greet() {
        return "Hello, " + this.greeting;
    }
}
```

Some features of a class are-

- Encapsulation i.e. creation of self-contained modules that bind processing functions to the data, takes

place.
- Classes are created in hierarchies and with the help of inheritance; the features of one class can be used in the other.
- Polymorphism takes place.
- Abstraction

Q11. Is Native Javascript supports modules?

Currently, Modules are not supported by Native JavaScript.To create and work with modules in Javascript you require an external like CommonJS.
Looking For Angular 2 Interview Questions. **Click here**

Q12. How to compile multiple Typescript files into a single file?

Using below command we can compile multiple typescript files into a single file
```
tsc --outFile outputfile.ts file1.ts file2.ts file3.ts  ... filen.ts
```

Q13. How to Call Base Class Constructor from Child Class in TypeScript?

super() function is used to called parent or base class constructor from Child Class.

Q14. What are Closures in Javascript?

Closures are nothing but a statefull functions.

A closure is an inner function that has access to outer function's variables in addition to it's own variable and global variables.In simple term a closure is a function inside a function.Closures using Lexical scoping in which function have access to global and outer variables.Below is a sample example of Javascript closure.

```
function employee() {
    var employee_dept = 'IT';

    return {
        getDept: function ()  {
          return employee_dept;
        },
        setDept: function (new_dept)  {
            employee_dept = new_name;
        }
    }
?
}
var emp1 = employee (); // In this juncture, the employee_dept outer function has
mjID.getDept(); // IT
mjID.setDept('Account'); // Changes the outer function's variable?
mjID.getDept(); //outputs Account
```

Closures prevents your important variable to be modified accidently.It can be only modified by itself.

Q15. List types of scopes available in Javascript?

There are two types of scopes are available in Javascript, they are
- Global Scope
- Local Scope

Q16. What are Modules in Typescript?

Modules are powerful way to share code between files.By using modules effectively in your project you can keep you file size small and dependencies clear.

> Modules are executed within their own scope, not in the global scope; this means that variables, functions, classes, etc. declared in a module are not visible outside the module unless they are explicitly exported using one of the export forms.

Creating a Module

```
module module_name{
 class xyz{
  export foo(x, y){
    return x*y;
   }
 }
}
```

Q17. What is namespace in Typescript? How to declare a namespace in Typescript?

Internal Modules are known as namespaces in Typescript.Namespaces are used to maintain the legacy code of typescript interally. A namespace is simply a way to logically group related classes or interfaces in a wrapper. Synatax for creating namespace in Typescript

```
namespace YourNameSpace {
    export class Class1 {  }
    export class Class2 {  }
}
```

Q18. Explain Decorators in Typescript? List type of Decorators available in Typescript?

Decorators are function that supports annotating or modifying classes its members.Its allow way to add both annotations and a meta-programming syntax for class declarations and members. Decorators are an experimental feature of Typescript and maybe change in future releases.

You can enable Decorators using the command line or editing your tsconfig.json

Enabling Decorators in TypeScript via command line

```
tsc --target ES5 --experimentalDecorators
```

Q19. What are Mixins?

In Javascript Mixins are another way of building up classes from reusable components is to build them by combining simpler partial classes.

Q20. What is default visibility for properties/methods in Typescript classes?

Public is default visibility for properties/methods in Typescript classes