support@mytectra.com Call Us: +91-90191-91856

Login / Register

myTectra

POPULAR COURSES IT COURSES BUSINESS COURSES CORPORATE TRAINING

DIGITAL

Search For The Courses, Software or Skills You Want to Learn...

SEARCH

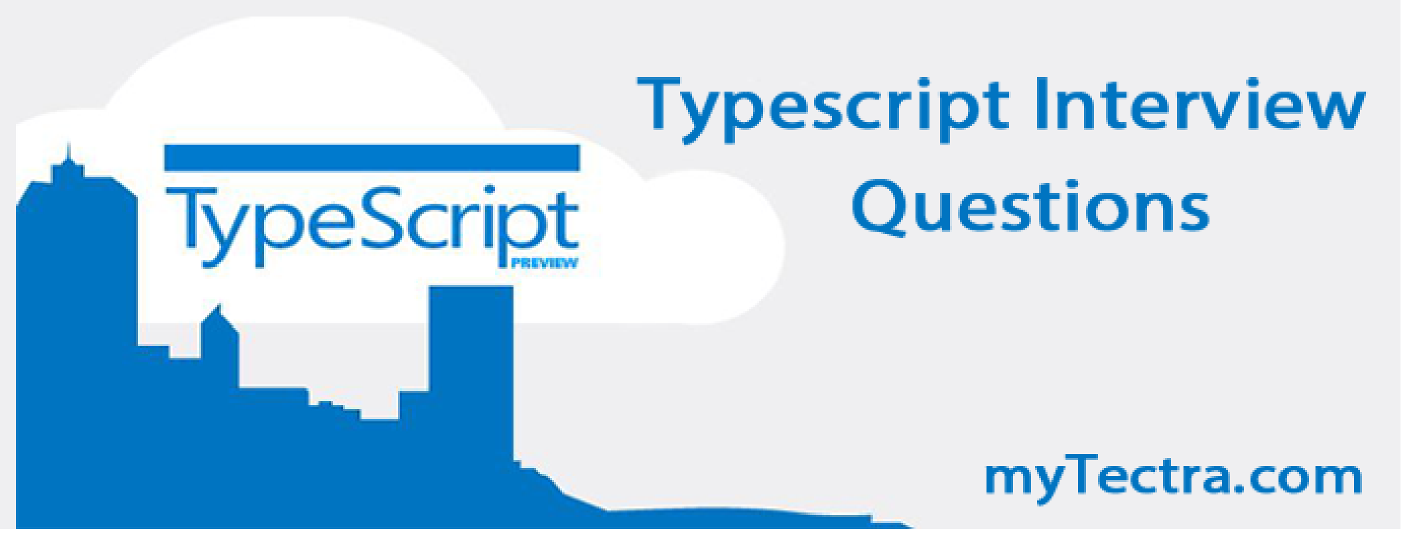
PREVIOUS ARTICLE

Java Web Services Interview Questions and Answers

NEXT ARTICLE

Advanced C++ Programming Interview Questions and Answers

Top 25 Typescript Interview Questions and Answers (updated 2019)



BY VENKATESAN M

May 25, 2017 in INFORMATION TECHNOLOGIES (IT) No Comments 1058

Q1. What is TypeScript and Why Do We Need It?

Ans. JavaScript is the only client side language universally supported by all browsers. But JavaScript is not the best designed language. It's not a class-based object-oriented language, doesn't support class based inheritance, unreliable dynamic typing and lacks in compile time error checking. And TypeScript addresses all these problems. In other words, TypeScript is an attempt to "fix" JavaScript problems.

TypeScript is a free and open source programming language developed and maintained by Microsoft. It is a strict superset of JavaScript, and adds **optional static typing** and **class-based object-oriented programming** to the language. TypeScript is quite easy to learn and use for developers familiar with C#, Java and all strong typed languages. At the end of day "TypeScript is a language that generates plain JavaScript files."

As stated on Typescript official website, "TypeScript lets you write JavaScript the way you really want to. TypeScript is a typed superset of JavaScript that compiles to plain JavaScript. Any browser. Any host. Any OS. Open Source." Where "**typed**" means that it considers the types of variables, parameters and functions.

Q2. What are the Benefits of TypeScript?

Ans. TypeScript has following benefits:

- It helps in code structuring
- Use class based object oriented programming

- Offers type checking
- Compile time error checking
- Intellisense

Q3. What are Different Components of TypeScript?

Ans. There are mainly 3 components of TypeScript:

1. **Language**– The most important part for developers is the new language. The language consists of new syntax, keywords and allows you to write TypeScript.
2. **Compiler**– The TypeScript compiler is open source, cross-platform and open specification, and is written in TypeScript. Compiler will compile your TypeScript into JavaScript. And it will also emit error, if any. It can also help in concatenating different files to a single output file and in generating source maps.
3. **Language Service**– TypeScript language service which powers the interactive TypeScript experience in Visual Studio, VS Code, Sublime, the TypeScript playground and other editor.

Q4. How Can You Get TypeScript and Install It?

Ans. TypeScript can be installed and managed via npm, the Node.js package manager. To install TypeScript, first ensure the npm is installed properly. And then run the following command to install TypeScript globally on your system.

Hide Copy Code

```
npm install -g typescript
```

TypeScript is included in Visual Studio 2013 Update 2 and Visual Studio 2015 by default. TypeScript also provides support for other editors like Visual Studio Code, sublime, Emacs and Vim.

Q5. How Do You Compile TypeScript Files?

Ans. The extension for any TypeScript file is ***".ts"***. And any JavaScript file is TypeScript file as it is a superset of JavaScript. So change extension of "*.js*" to "*.ts*" file and your TypeScript file is ready. To compile any *.ts* file into *.js*, use the following command.

Hide Copy Code

```
tsc <TypeScript File Name>
```

For example, to compile "*Helloworld.ts*":

```
tsc helloworld.ts
```

And the result would be *helloworld.js*.

Q6. Is It Possible to Combine Multiple .ts Files into a Single .js File?

Ans. Yes, it's possible. While compiling add –outFILE [OutputJSFileName] option.

Hide Copy Code

```
tsc –outFile comman.js file1.ts file2.ts file3.ts
```

This will compile all 3 "*.ts*" file and output into single "*comman.js*" file. And what will happen if you don't provide a output file name.

Hide Copy Code

```
tsc –outFile file1.ts file2.ts file3.ts
```

In this case, *file2.ts* and *file3.ts* will be compiled and the output will be placed in *file1.ts*. So now your *file1.ts* contains JavaScript code.

Q7. Is It Possible to Compile .ts Automatically with Real Time Changes in .ts File?

Ans. Yes. Using –watch compiler option, this can be achieved.

Hide Copy Code

```
tsc –watch file1.ts
```

This will first compile *file1.ts* in *file.js* and watch for the file changes. As soon as there is any change detected, it will compile it again. But you need to ensure that command prompt must not be closed with –watchoption.

Ans. The answer is **YES**. There are 4 main principles to Object Oriented Programming: Encapsulation, Inheritance, Abstraction, and Polymorphism. TypeScript can implement all four of them with its smaller and cleaner syntax. Read Write Object-Oriented JavaScript with TypeScript.

Q9. Which Object Oriented Terms are Supported by TypeScript?

Ans. TypeScript supports the following object oriented terms:

- Modules
- Classes
- Interfaces
- Data Types
- Member functions

Q10. Which are the Different Data Types Supported by TypeScript?

Ans. TypeScript supports the following data types:

- Booleanvar bValue: boolean = false;
- Numbervar age: number = 16;
- Stringvar name: string = “jon”;
- Arrayvar list:number[] = [1, 2, 3];
- Enum

Hide Copy Code

```
enum Color {Red, Green, Blue};

var c: Color = Color.Green;
```

- Anyvar unknownType: any = 4;
- Void

Hide Copy Code

```
function NoReturnType(): void {

}
```

Q11. How TypeScript is Optionally Statically Typed Language?

Ans. TypeScript is referred as optionally statically typed, which means you can ask the compiler to ignore the type of a variable. Using any data type, we can assign any type of value to the variable. TypeScript will not give any error checking during compilation.

Hide Copy Code

```
var unknownType: any = 4;

unknownType = “Okay, I am a string”;

unknownType = false; // A boolean.
```

Q12. What are Modules in TypeScript?

Ans. Modules are the way to organize code in TypeScript. Modules don’t have any features, but can contain classes and interfaces. It is same like namespace in C#.

Q13. What are Classes in TypeScript?

Ans. The concept of classes is very similar to .NET/Java. A Class can have constructor, member variables, properties and methods. TypeScript also allows access modifiers “private” and “public” for member variables and functions.

Q14. How Do You Implement Inheritance in TypeScript?

Ans. Using extends keyword, we can implement inheritance.

```
class Animal {

public domestic:boolean;

constructor(public name: string) { }

}



class Cat extends Animal {

constructor(name: string, domestic: boolean)

{

super(name);

this.domestic = true;

}

}



class Tiger extends Animal {

constructor(name: string, domestic: boolean)

{

super(name);

this.domestic = false;

}

}
```

Q15. How to Call Base Class Constructor from Child Class in TypeScript?

Q16. What is the Default Access Modifier for Members of a Class in TypeScript?

Ans. In TypeScript, each member of class is public by default.

Q17. How Can You Allow Classes Defined in a Module to be Accessible Outside of the Module?

Ans. Classed define in a module are available within the module. Outside the module, you can’t access them.

Hide Copy Code

```
module Vehicle {

class Car {

constructor (

public make: string,

public model: string) { }

}

var audiCar = new Car(“Audi”, “Q7”);

}

// This won’t work

var fordCar = Vehicle.Car(“Ford”, “Figo”);
```

As per above code, fordCar variable will give us compile time error. To make classes accessible outside module, use export keyword for classes.

Hide Copy Code

```
export class Car {

constructor (

public make: string,

public model: string) { }

}

var audiCar = new Car(“Audi”, “Q7”);

}

// This works now

var fordCar = Vehicle.Car(“Ford”, “Figo”);
```

Q18. How Does TypeScript Support Optional Parameters in Function as in JavaScript Every Parameter is Optional for a Function?

Ans. In JavaScript, every parameter is considered optional. If no value is supplied, then it is treated as undefined. So while writing functions in TypeScript, we can make a parameter optional using the **“?”** after the parameter name.

Hide Copy Code

```
function Demo(arg1: number, arg2? :number) {

}
```

So, arg1 is always required and arg2 is an optional parameter. Remember, **optional parameters must follow required parameters.** If we want to make arg1 optional, instead of arg2, then we need to change the order and arg1 must be put after arg2.

Hide Copy Code

```
function Demo(arg2: number, arg1? :number) {

}
```

Similar to optional parameters, default parameters are also supported.

```
function Demo(arg1: number, arg2 = 4) {
}
```

Q19. Does TypeScript Support Function Overloading as JavaScript Doesn’t Support Function Overloading?

Ans. Yes, TypeScript does support function overloading. But the implementation is odd. When you overload in TypeScript, you only have one implementation with multiple signatures.

Hide Copy Code

```
class Customer {
name: string;
Id: number;

add(Id: number);
add(name:string);
add(value: any) {
if (value && typeof value == “number”) {
//Do something
}
if (value && typeof value == “string”) {
//Do Something
}
}
}
```

The any data type can take any type of data. The implementation then checks for the type of the supplied parameter and executes a different piece of code based on supplier parameter type.

OR You can also use union type introduced in TypeScript 1.4. Union types let you represent a value which is one of multiple types.

Hide Copy Code

```
add(a:string|number) {

//do something

}
```

Using union type, you can typically remove the need for an overload.

Q20. Is It Possible to Debug Any TypeScript File?

Ans. Yes, it is possible. To debug it, you need *.js* source map file. If you are new to source map, read more here. So compile the *.ts* file with the –sourcemap flag to generate a source map file.

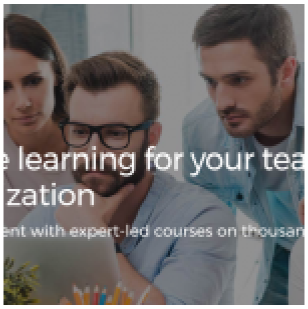
Hide Copy Code

```
tsc -sourcemap file1.ts
```

This will create *file1.js* and *file1.js.map*. And last line of *file1.js* would be reference of source map file.

Hide Copy Code

```
//# sourceMappingURL=file1.js.map
```



Q21. What is TypeScript Definition Manager and Why Do You Need It?

Ans. TypeScript Definition Manager (TSD) is a package manager to search and install TypeScript definition files directly from the community driven DefinitelyTyped repository. Let's see with an example.

Suppose, you want to use some jQuery code in your *.ts* file.

Hide Copy Code

```
$(document).ready(function() { //Your jQuery code });
```

And now when you try to compile it using tsc, you will get compile time error **Cannot find name "$".** That's because TypeScript can't understand what does "$" means. So somehow we need to inform TypeScript compiler that it belongs to jQuery. That's where TSD comes into play. You can download jQuery Type Definition file and include it in your *.ts* file. First, install TSD.

Hide Copy Code

```
npm install tsd -g
```

In your *typescript* directory, create a new typescript project by running:

Hide Copy Code

```
tsd init
```

Then install the definition file for jquery.

Hide Copy Code

```
tsd query jquery –action install
```

This will download and create a new directory containing your jquery definition file. The definition file ends with "*.d.ts*". So now include it by updating your typescript file to point to the jquery definition.

Hide Copy Code

```
/// <reference path="typings/jquery/jquery.d.ts" />

$(document).ready(function() { //To Do

});
```

Now try to compile again and this time, js will be generated without any error.

Ans. It’s quite possible that JavaScript libraries/frameworks don’t have TypeScript definition files and yet you want to use them without any errors. Te solution is to use the declare keyword. The declare keyword is used for **ambient declarations** where you want to define a variable that may not have originated from a TypeScript file.

Hide Copy Code

```
declare var unKnownLibrary;
```

TypeScript runtime will assign unKnownLibrary variable any type. You won’t get Intellisense in design time but you will be able to use the library in your code.

Q23. How to Generate TypeScript Definition File from Any .ts File?

Ans. You can generate TypeScript definition file from any *.ts* file via tsc compiler. Generating a TypeScript definition will make your TypeScript file reusable.

Hide Copy Code

```
tsc –declaration file1.ts
```

Q24. What is tsconfig.json File?

Ans. The presence of a *tsconfig.json* file in a directory indicates that the directory is the root of a TypeScript project. The *tsconfig.json* file specifies the root files and the compiler options required to compile the project. And using this file, we can streamline building TypeScript project. Below is a sample *tsconfig.json* file.

Hide Copy Code

```
{

“compilerOptions”: {

“removeComments”: true,

“sourceMap”: true

},

“files”: [
"othermodule.ts"

]

}
```

Within files section, define all the *.ts* files in the project. And when you invoke tsc without any other arguments with the above file in the current directory, it will compile all the files with the given compiler option settings.

Q25. What are the Disadvantages of TypeScript?

Ans. Well, TypeScript is great but there are some disadvantages as well.

- TypeScript is just another way to write JavaScript. It doesn't fix the problems of JavaScript. It just creates an illusion that it does.
- One more tool to learn.
- TypeScript has dependency on type definition files, if you wish to use any existing JavaScript libraries.
- Quality of type definition files is a concern as how can you be sure the definitions are correct?
- Frequent releases of new versions JavaScript library is also a pain area. Because if their type definition files are not updated then you can't use them instantly.
- In order to run the application in the browser, a compile step is required to transform TypeScript into JavaScript.
- Web developers are using JavaScript from decades and TypeScript doesn't bring anything new.
- To use any third party library, definition file is you need. And not all the third party library have definition file available.

Know More about TYPESCRIPT

Summary

Reviewer	Keerthi
Review Date	2017-05-25
Reviewed Item	This interview questions were useful.Thank You myTectra.
Author Rating	★★★★★

Post Views: 1,212

Share

TYPESCRIPT

TYPESCRIPT INTERVIEW QUESTIONS

TYPESCRIPT INTERVIEW QUESTIONS ANS ANSWERS

RECOMMENDED POSTS

Blockchain Interview Questions

January 29, 2019

CRM Interview Questions

May 26, 2018

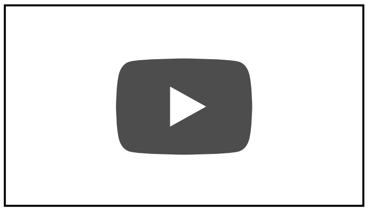
SAP BI/BW Frequently asked Interview Questions.

May 26, 2018

ONLINE TRAINING COMPANY OF THE YEAR 2017

INNOVATIVE ONLINE TRAINING COMPANIES

SERVICES EXCELLENCE AWARDS



BEST TRAINING COMPANY OF THE YEAR 2018

TAGS

angularjs interview questions automation testing with python aws aws interview questions

Data Science Interview Question Digital Marketing Digital Marketing Interview questions

django interview questions ext js ext js interview questions hadoop hadoop interview questions hbase

interview questions for CRM interview questions for Excel VBA

interview questions for maongodb ios interview questions and answers magento interview questions

mongo db interview questions node js node js interview questions oracle scm oracle scm interview questions pentaho bi pentaho bi interview questions

performance testing interview questions

PHP interview questions Python Interview Question ruby cucumber

ruby cucumber interview questions sap Bpc interview questions SEO

SEO Interview Question swift programming swift programming interview questions tableau

tableau interview questions TCL Interview Questions and Answers-2017 testing with python Typo3 cms

Typo3 cms interview questions web design web design interview questions zend framework

zend framework interview questions

CATEGORIES

- Digital Marketing
- Finance and Accounting
- Information Technologies (IT)
- Tips & Trick
- Work

Popular Recent Comments

ASP .Net Web API Essentials using C# Interview Questions
May 24, 2017

Accounting and Finantial Accounting Interview Questions and Answers
myTectra.com

Basic Accounting and Financial Accounting Interview Questions and Answers
May 17, 2017

C# Interview Questions and Answers for 5 years Experienced
March 17, 2018

June 12, 2017

Basic Electronics Interview Questions and Answers

April 2, 2018

SOCIAL

10P, IWWA Building, 2nd Floor,7th Main Road, BTM Layout 2nd Stage,Bangalore-560076,Karnataka, India

info@mytectra.com
support@mytectra.com

+91 9019191856
+1 (209) 222-4733 (USA)

ABOUT US

myTectra a global learning solutions company helps transform people and organization to gain real, lasting benefits.Join Today.Ready to Unlock your Learning Potential !

MYTECTRA

- Awards & Recognition
- Blog

- Become a Trainer
- Terms and Conditions
- Privacy Policy
- Contact Us

SERVICES

- Class Room Training
- Live Online Training
- IT Services
- Corporate Training
- Government Training
- University Training
- Online Self-Learning

NEWSLETTER SIGNUP

Get an email of every new post! We will never share your address.

Example: name@gmail.com