Choose Interview Topic

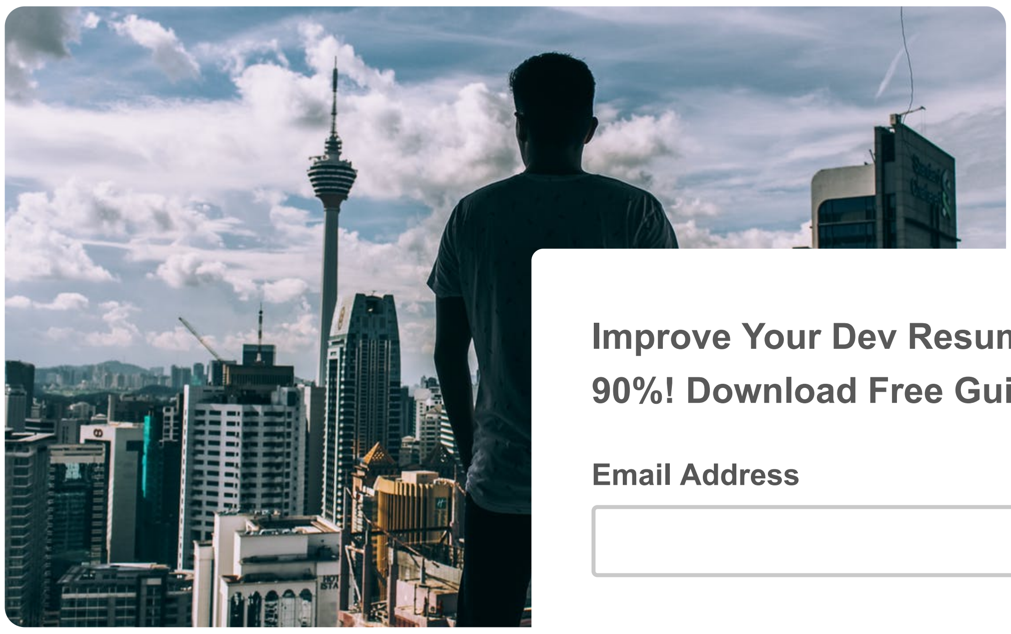
Click on a topic to explore Q&As

.NET Core 51	ADO.NET 33	ASP.NET 50	ASP.NET MVC 31	ASP.NET Web API 33	
AWS 44	Agile & Scrum 52	Angular 68	AngularJS 62	Azure 58	Behavioral 112
Bootstrap 38	C# 112	CSS 50	Code Problems 26	Data Structures 30	
Design Patterns 45	DevOps 36	Docker 55	Entity Framework 57	Git 26	GraphQL 13
HTML5 55	Java 103	JavaScript 116	LINQ 35	Laravel 41	MongoDB 79
Node.js 95	OOP 52	Questions to Ask 39	React 147	Redux 30	Ruby 84
Ruby on Rails 72	SOA & REST API 67	SQL 56	Software Architecture 38	Software Testing 26	
T-SQL 36	TypeScript 39	UX Design 13	Vue.js 41	WCF 32	Web Security 58
Webpack 32	jQuery 51				

#TypeScript

20 Basic TypeScript Interview Questions (2018 Edition)

7



ECMAScript 2015 and future proposals, like async functions and decorators, to help build robust components. TypeScript compiles to clean, simple JavaScript code which runs on any browser, in Node.js, or in any JavaScript engine that supports ECMAScript 3 (or newer).

Q1: What is TypeScript and why would I use it in place of JavaScript?

Topic: **TypeScript**
Difficulty: ⭐

Answer:
TypeScript is a superset of JavaScript which primarily provides optional static typing, classes and interfaces. One of the big benefits is to enable IDEs to provide a richer environment for spotting common errors as *you type the code*. For a large JavaScript project, adopting TypeScript might result in more robust software, while still being deployable where a regular JavaScript application would run.

In details:

- TypeScript supports new ECMAScript standards and compiles them to (older) ECMAScript targets of your choosing. This means that you can use features of ES2015 and beyond, like modules, lambda functions, classes, the spread operator, destructuring, today.
- JavaScript code is valid TypeScript code; TypeScript is a superset of JavaScript.
- TypeScript adds type support to JavaScript. The type system of TypeScript is relatively rich and includes: interfaces, enums, hybrid types, generics, union and intersection types, access modifiers and much more. TypeScript makes typing a bit easier and a lot less explicit by the usage of type inference.
- The development experience with TypeScript is a great improvement over JavaScript. The IDE is informed in real-time by the TypeScript compiler on its rich type information.
- With strict null checks enabled (`--strictNullChecks` compiler flag) the TypeScript compiler will not allow undefined to be assigned to a variable unless you explicitly declare it to be of nullable type.
- To use TypeScript you need a build process to compile to JavaScript code. The TypeScript compiler can inline source map information in the generated .js files or create separate .map files. This makes it possible for you to set breakpoints and inspect variables during runtime directly on your TypeScript code.
- TypeScript is open source (Apache 2 licensed, see github) and backed by Microsoft. *Anders Hejlsberg*, the lead architect of C# is spearheading the project.

🔗 **Source:** stackoverflow.com
💡 **Don't miss:** 39 More TypeScript Interview Questions & Answers

Q2: Explain generics in TypeScript

Topic: **TypeScript**
Difficulty: ⭐

Generics are able to create a component or function to work over a variety of types rather than a single one.

```
/** A class definition with a generic parameter */
class Queue<T> {
  private data = [];
  push = (item: T) => this.data.push(item);
  pop = (): T => this.data.shift();
}

const queue = new Queue<number>();
queue.push(0);
queue.push("1"); // ERROR : cannot push a string. Only numbe
```

🔗 **Source:** basarat.gitbooks.io
💡 **Don't miss:** 39 More TypeScript Interview Questions & Answers

Difficulty: ☆ ☆

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.

Source: jonathanmh.com
Don't miss: 39 More TypeScript Interview Questions & Answers

Q4: How could you check null and undefined in TypeScript?

Topic: **TypeScript**
Difficulty: ☆ ☆

Just use:

```
if (value) {
}
```

It will evaluate to `true` if `value` is not:

- `null`
- `undefined`
- `NaN`
- empty string `''`
- `0`
- `false`

TypesScript includes JavaScript rules.

Source: stackoverflow.com
Don't miss: 39 More TypeScript Interview Questions & Answers

All TypeScript Q&A

Q5: How to implement class constants in TypeScript?

Topic: **TypeScript**
Difficulty: ☆ ☆

In TypeScript, the `const` keyword cannot be used to declare class properties. Doing so causes the compiler to an error with "A class member cannot have the 'const' keyword." TypeScript 2.0 has the `readonly` modifier:

```
class MyClass {
    readonly myReadonlyProperty = 1;

    myMethod() {
        console.log(this.myReadonlyProperty);
    }
}

new MyClass().myReadonlyProperty = 5; // error, readonly
```

Source: stackoverflow.com
Don't miss: 39 More TypeScript Interview Questions & Answers

Topic: **TypeScript**
Difficulty: ★ ★

`.map` files are source map files that let tools map between the emitted JavaScript code and the TypeScript source files that created it. Many debuggers (e.g. Visual Studio or Chrome's dev tools) can consume these files so you can debug the TypeScript file instead of the JavaScript file.

Source: stackoverflow.com
Don't miss: 39 More TypeScript Interview Questions & Answers

Q7: What is getters/setters in TypeScript?

Topic: **TypeScript**
Difficulty: ★ ★

TypeScript supports **getters/setters** as a way of intercepting accesses to a member of an object. This gives you a way of having finer-grained control over how a member is accessed on each object.

```
class foo {
  private _bar:boolean = false;

  get bar():boolean {
    return this._bar;
  }
  set bar(theBar:boolean) {
    this._bar = theBar;
  }
}

var myBar = myFoo.bar;  // correct (get)
myFoo.bar = true;  // correct (set)
```

Source: typescriptlang.org
Don't miss: 39 More TypeScript Interview Questions & Answers

All TypeScript Q&A

Q8: Could we use TypeScript on backend and how?

Topic: **TypeScript**
Difficulty: ★ ★

Typescript doesn't only work for browser or frontend code, you can also choose to write your backend applications. For example you could choose Node.js and have some additional type safety and the other abstraction that the language brings.

1. Install the default Typescript compiler

```
npm i -g typescript
```

2. The TypeScript compiler takes options in the shape of a tsconfig.json file that determines where to put built files and in general is pretty similar to a babel or webpack config.

```
{
  "compilerOptions": {
    "target": "es5",
    "module": "commonjs",
    "declaration": true,
    "outDir": "build"
  }
}
```

3. Compile ts files

```
tsc
```

```
node build/index.js
```

Source: jonathanmh.com
Don't miss: 39 More TypeScript Interview Questions & Answers

Q9: What are different components of TypeScript?

Topic: **TypeScript**
Difficulty: ☆ ☆ ☆

There are mainly 3 components of TypeScript .

1. **Language** – The most important part for developers is the new language. The language consist 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 concating different files to 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.

Source: talkingdotnet.com
Don't miss: 39 More TypeScript Interview Questions & Answers

Q10: Is that TypeScript code valid? Explain why.

Topic: **TypeScript**
Difficulty: ☆ ☆ ☆

Consider:

```
class Point {
    x: number;
    y: number;
}

interface Point3d extends Point {
    z: number;
}

let point3d: Point3d = {x: 1, y: 2, z: 3};
```

Answer:
Yes, the code is valid. A class declaration creates two things: a *type* representing instances of the class and a *constructor function*. Because classes create types, you can use them in the same places you would be able to use interfaces.

Source: typescriptlang.org
Don't miss: 39 More TypeScript Interview Questions & Answers

All TypeScrip

Q11: Explain how and why we could use pr

Topic: **TypeScript**
Difficulty: ☆ ☆ ☆

Decorators can be used to modify the behavior of a class or becom
instance, if your framework has methods with restricted access rec
method decorator to deny access to non-administrative users, or ar
ability to modify it.

```
    post() { }

    @admin
    delete() { }

    @owner
    put() { }
}
```

Source: www.sitepen.com

Don't miss: 39 More TypeScript Interview Questions & Answers

Q12: Are strongly-typed functions as parameters possible in TypeScript?

Topic: **TypeScript**
Difficulty: ★ ★ ★

Consider the code:

```
class Foo {
    save(callback: Function) : void {
        //Do the save
        var result : number = 42; //We get a number from the save operation
        //Can I at compile-time ensure the callback accepts a single parameter of type number somehow?
        callback(result);
    }
}

var foo = new Foo();
var callback = (result: string) : void => {
    alert(result);
}
foo.save(callback);
```

Can you make the result parameter in `save` a type-safe function? Rewrite the code to demonstrate.

Answer:

In TypeScript you can declare your **callback type** like:

```
type NumberCallback = (n: number) => any;

class Foo {
    // Equivalent
    save(callback: NumberCallback): void {
        console.log(1)
        callback(42);
    }
}

var numCallback: NumberCallback = (result: number) : void => {
    console.log("numCallback: ", result.toString());
}

var foo = new Foo();
foo.save(numCallback)
```

Source: stackoverflow.com

Don't miss: 39 More TypeScript Interview Questions & Answers

Q13: How can you allow classes defined in module?

Topic: **TypeScript**
Difficulty: ★ ★ ★

Classes define in a module are available within the module. Outside

```
        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.

```
module Vehicle {
    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");
```

Source: http://www.talkingdotnet.com

Don't miss: 39 More TypeScript Interview Questions & Answers

All TypeScript Q&A

Q14: Does TypeScript supports function overloading?

Topic: **TypeScript**
Difficulty: ⭐ ⭐ ⭐

Yes, TypeScript does support function overloading but the implementation is a bit different if we compare it to OO languages. We are creating just one function and a number of declarations so that TypeScript doesn't give compile errors. When this code is compiled to JavaScript, the concrete function alone will be visible. As a JavaScript function can be called by passing multiple arguments, it just works.

```
class Foo {
    myMethod(a: string);
    myMethod(a: number);
    myMethod(a: number, b: string);
    myMethod(a: any, b?: string) {
        alert(a.toString());
    }
}
```

Source: typescriptlang.org

Don't miss: 39 More TypeScript Interview Questions & Answers

Q15: Explain why that code is marked as W

Topic: **TypeScript**
Difficulty: ⭐ ⭐ ⭐ ⭐

```
/* WRONG */
interface Fetcher {
    getObject(done: (data: any, elapsedTime?: number) => voi
}
```

Q16: How would you overload a class constructor in TypeScript?

Topic: **TypeScript**
Difficulty: ⭐⭐⭐⭐

Sign In To See Answer

Q17: What is the difference between "interface vs type" statements?

Topic: **TypeScript**
Difficulty: ⭐⭐⭐⭐

```
interface X {
    a: number
    b: string
}

type X = {
    a: number
    b: string
};
```

Answer:

Sign In To See Answer

Q18: Explain when to use "declare" keyword in TypeScript

Topic: **TypeScript**
Difficulty: ⭐⭐⭐⭐⭐

Sign In To See Answer

Q19: What are Ambients in TypeScripts and when to use them?

Topic: **TypeScript**
Difficulty: ⭐⭐⭐⭐⭐

Sign In To See Answer

Q20: Is it possible to generate TypeScript declaration files from JS library?

Topic: **TypeScript**
Difficulty: ⭐⭐⭐⭐⭐

Sign In To See Answer

> *Thanks 🙌 for reading and good luck on your interview!*
> *Please share this article with your fellow devs if you like it!*
> *Check more FullStack Interview Questions & Answers on 👉 www.*

Improve Dev Resume Re
Get Your Free (

email address

Explore All

Recommended For You

Subscribe to Blog

35 LINQ Interview Questions and Answers in 2019
#LINQ

Read More...

Top 26+ React Interview Questions (2018 Edition)
#React #Redux

Read More...

27 C# Interview Questions and Answers to Know in 2019
#.NET Core #C# #LINQ

Read More...

50 Common Front End Developer Interview Questions [2019 Edition]
#AngularJS #CSS #HTML5

Read More...

20 .NET Core Interview Questions and Answers
#.NET Core #ASP.NET

Read More...

20 Vue.js Interview Questions And Answers (2018 Updated)
#Vue.js

Read More...