

[Home](#) / [Articles](#) / [Career Development](#)[Sameeksha Medewar](#) | 08 Aug, 2023

Top 50 Typescript Interview Questions and Answers in 2024



TypeScript is a free, open-source programming language you can apply to any application supported by [JavaScript](#) — meaning both front-end and back-end programming.

Designed and maintained by Microsoft, TypeScript is a superset of JavaScript. However, unlike JavaScript, TypeScript performs type checking, making it a suitable choice for many large and complex web applications.

Itching for a web development career move? You're probably scouring job ads in search of the perfect opportunity. Learning TypeScript will open many



Keep reading for our roundup of 50 TypeScript interview questions and answers to pass your next interview with flying colors!

Top Typescript Interview Questions for All Experience Levels

Are you a TypeScript pro and looking for a career jump? Or maybe you're fresh in the programming game and seeking an entry-level gig.

Whether you're a genius or a newcomer to TypeScript, these interview questions will help you on your career journey. We've split our list into basic, intermediate, and advanced TypeScript questions.

TypeScript Basic Interview Questions

If you are a newbie and with TypeScript experience, expect the following interview questions on TypeScript.

1. What is TypeScript?

TypeScript is a superset of JavaScript used to create

> **hackr.io**



Login



share the same initial semantics and syntax.

You can convert TypeScript code to ES5, allowing for important features like decorators, generics and interfaces, enums, and modules. You'll also find native JavaScript features like the spread arrow function and



- **Quick Code:** TypeScript reuses existing JavaScript code and supports JavaScript libraries, allowing for cleaner and quicker code.
- **Static Typing:** Programmers can easily identify issues as they are being debugged, speeding up the testing and debugging process.
- **User-Friendly:** Developers can create robust and user-friendly web apps using highly productive [development tools](#) through TypeScript.
- **Strong History:** Although you can change or update array elements' values, you cannot delete them.

3. What is “any” in TypeScript?

When building an application, you might need to describe variables you're unfamiliar with. These values could originate from dynamic content with variable type. We want to forego type-checking in these circumstances and allow the values to pass compile-time checks instead. To achieve this, we assign these the *any* type label. We can store *any* type in a variable without running into type errors.

4. Differentiate between 'let', 'var', and 'const' in TypeScript.

let: Allows you to define variables with both global and local scopes.

Global variable declaration: Using *let* outside a function.

Local variable declaration: Using *let* inside a function.

next. Enables you to define only variables with a global



const: Similar to *let*, except you cannot modify a const variable's value after its defined.

5. What is the 'tsconfig.json' file?

You can list various settings in the tsconfig.json file, which is in JSON format. This tells the compiler how to compile a project. This file is the TypeScript project root because it is in the directory.

6. What is "never" in TypeScript?

The "never" type represents values that have never occurred. Additionally, variables constricted by type guards that can never be true also acquire the type never.

For example, never is a function expression's return type.

7. What is the extension of TypeScript?

TypeScript Definition file (with .d.ts extension) provides definition for external JavaScript libraries.

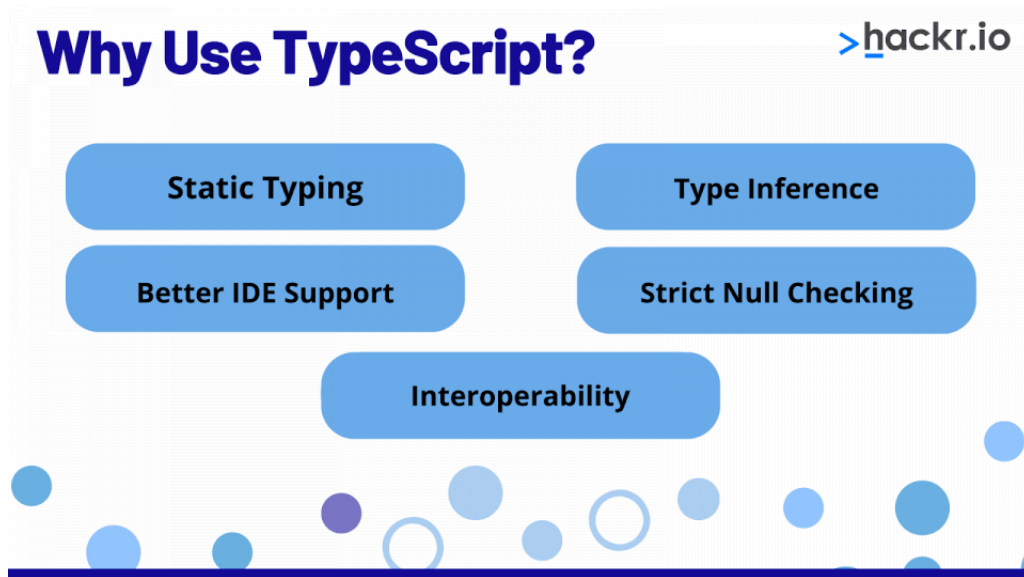
8. How are TypeScript and JavaScript different?

Javascript is a dynamically typed language; thus, there is no compiler to tell you if something is wrong. Instead, you must run the code to assess proper function. Furthermore, since everything in JS is stated as a variable rather than a type, the type cannot be guaranteed until runtime. So, maintaining a sizable code base with JavaScript is challenging.

Contrarily, TypeScript is statically typed and includes [type checking](#) to ensure that our applications are bug-free.

9. Why do we need TypeScript?





Here are some top reasons why companies need TypeScript:

Static Typing: TypeScript allows JavaScript to have type support, which otherwise is dynamically typed.

Type Inference: TypeScript uses type inference to simplify typing. The types are still present to prevent run-time errors.

Better IDE Support: TypeScript programming is a vast improvement over JavaScript development. Numerous [IDEs](#), including Atom, Visual Studio & VS Code, IntelliJ/WebStorm, and Sublime have great TypeScript support.

Strict Null Checking: In JavaScript programming, you'll often run into errors like the inability to read property 'x' of undefined are frequent. Since you cannot use a variable unknown to the TypeScript compiler, you'll avoid these mistakes with TypeScript.

Interoperability: Because TypeScript and JavaScript are closely related, TypeScript offers excellent interoperability. This means you can write the two coding languages alongside each other.

Using TypeScript has the following advantages:

- **Quick:** Its shared syntax and semantics with JavaScript make code writing quicker.
- **Integration:** TypeScript reuses JavaScript code, allowing it to seamlessly integrate with existing [JavaScript frameworks](#) and libraries.
- **ES6 and ES7 capabilities:** TypeScript has capabilities for ES6 and ES7 that operate in JavaScript ES5 engines like [Node.js](#).

11. What are TypeScript's drawbacks?

- **Manual typing:** If you implement strict TypeScript for your projects, you might have to type the code manually, which is time-consuming.
- **Properties requirements:** [Unit testing](#) is trickier with TypeScript because you must use specific types with each property.
- **Extra step for converting to JavaScript:** You must undergo a compilation step to convert TypeScript into JavaScript.
- **Definition file requirement:** Any third-party library must have a definition file before use. Additionally,



12. Who created TypeScript?

Anders Hejlberg developed TypeScript while working for Microsoft. You might know him as a key contributor to the [C# language](#) development team. Version 0.8 of TypeScript was first made available on October 1, 2012, under that name.

However, Microsoft maintains TypeScript under the Apache 2 license.

Understanding TypeScript - 2024 Edition

13. Which stable version of TypeScript is currently available?

TypeScript's most recent stable release, version 3.2, was made available on September 30, 2018. Any browser that supports the ECMAScript 2015 framework can execute TypeScript because it transpiles to straightforward JavaScript code. It also supports with the most recent and developing JavaScript features.

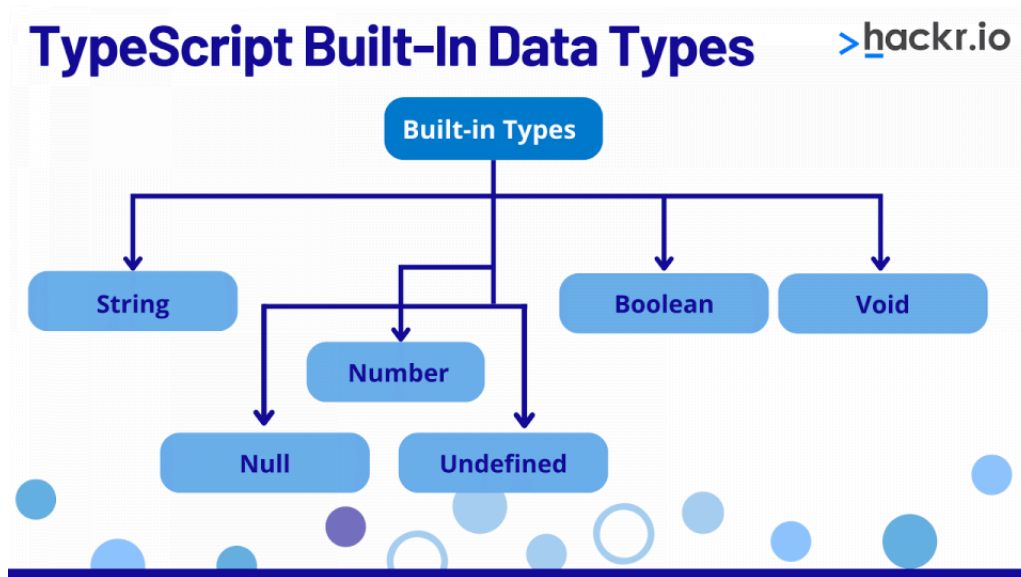
14. What are design patterns?

Design patterns are reusable pieces of code used to solve common software problems. Your projects will produce more modular, scalable, and optimized software if you employ design patterns. Since you'll be able to immediately identify a code's design pattern, understanding that code will be easier.

Common design patterns include Singleton, Factory Method, Decorator, Facade, and more.

15. List all of TypeScript's built-in data types.





- **Number:** This represents values of the number type. In TypeScript, the integers are kept as floating-point values.
- **String:** A string is a collection of characters encoded using Unicode UTF-16.
- **Boolean:** Logic value represented by a boolean. When using the Boolean type, we only receive true or false as the output.
- **Null:** Null designates a variable with an unknown value. Directly referencing a null type value is not feasible.
- **Undefined:** All uninitialized variables are represented by the Undefined type.



16. What are TypeScript variables, and how do you create them?

A variable is a specifically named memory area used to hold values. A colon (:) is placed after the variable name and followed by the type when declaring a variable in TypeScript.

Here are some guidelines to follow when declaring a variable in TypeScript:

- The variable name must have numbers or numbers.
- The name cannot begin with a digit.
- The only special characters you can use in the name are the dollar sign (\$) and underscore (_).

Intermediate TypeScript Interview Questions for Experienced Professionals

Here are some top TypeScript interview questions for the developers with some experience:

17. What do TypeScript interfaces do?

Interfaces (structures) define your application contract. The interface specifies the syntax classes you must use. Only members' declarations are included; the derived class is in charge of defining the members. The TypeScript compiler uses an interface to type-check objects and determines whether they have a particular structure.

```
interface interface_name
```



18. What is Singleton Design Pattern

Singleton is a design pattern for object production. It belongs to the category of "creative" design patterns, and permits us to have a single instance of a class.

Singleton objects are frequently used to provide options for configuring your application or a global environment. For instance, when your application starts up, it creates or initializes a global environment and fills it with configuration parameters already provided or pre-done. Several parts of your application may use these configuration settings.

19. What do TypeScript modules do?

TypeScript modules group relevant functions, variable, interfaces, and classes. You can use modules in their own scope (not the global scope).

In essence, you cannot directly access a module's defined variables, functions, classes, and interfaces from outside the module. You can use the export keyword to build a module, and the import keyword to include a module within another module.

Example:

```
module Module {  
  class A {  
    export sum(x, y){  
      return x+y;  
    }  
  }  
}
```

20. What distinguishes an internal module from an external module?

Internal Module



feature of Typescript's earlier iteration, and they group interfaces, functions, variables, and classes. You can use ModuleDeclarations to define the name and body of internal modules.

External Module

You can hide module definitions' internal statements with external modules, allowing only the methods and arguments connected to the defined variable to remain visible.

21. What is Model-View-Controller (MVC) architecture?

The primary MVC tenet is a program's partition into three parts:

- The model, which serves as the database blueprint;
- The controller, which regulates the interaction between the View and the Model; and The View (the UI and presentation).

Model

Models are used to describe the business logic and data form. In addition to accessing and storing object model states in a database, it preserves application data. It responds to user requests for data reading and changes the data as well. When receiving or publishing data, the Model interacts with the database, receives queries related to the database, processes them, and then sends the results back to the user through the browser.

View



the request to access particular resources, like a specific web page, using a web browser or a mobile app. You can use tools and frameworks like Vue or [React](#), or write the view entirely in HTML.

Controller

So what happens when you want to request data and receive a response? You would need the appropriate business logic — an algorithm that receives a user request, verifies it, or sends it to another component. This responsibility falls on the controllers.

22. What are Mixins?

When extending functionality in JavaScript, we can consider inheriting from Mixins as a strategy. Each newly created object has a prototype that allows the object to inherit additional properties. The ability to define properties for any number of object instances is even more significant than the ability to inherit from other object prototypes. You can use this information to encourage function reuse.

With the least amount of complexity possible, mixins enable objects to exchange or borrow functionalities. The pattern allows us a reasonably flexible approach to share functionality from not just one Mixin, but actually several through multiple inheritance because it works well with JavaScript's object prototypes. Mixins help a system increase function reuse and reduce functional repetition.

23. What is JSON?

JSON describes data structures and objects with its text-based standard. You might use JSON for the client- or server-side programming every so often.

24. What is the “scope” of a variable?



frequently used to indicate the range of a variable.

Global: A variable available across the entire program has a global scope. These variables are declared independently of any programming block or function. A global variable would typically be specified at the class level and accessible across that class in [object-oriented programming](#). It is accessible and modifiable in all of that class's functions.

Local: A variable's declared scope is the only area you can access it. They are conceived of as existing only while that function or code block is loaded into memory because they are defined inside of it. Only the function in which the variable is declared can access it or make changes to it.

25. What is the “declare” keyword in Typescript?

The "declare" keyword informs the compiler that the specified entity already exists in future code and can be referenced by other code. It is also used to indicate the statement does not require compilation into any JavaScript.

Perhaps it is a script that originates from another domain. When the script is evaluated, it will produce an object with some practical API methods and assign it to a global scope identifier. This statement is used by the compiler to statically check other code; nevertheless, no JavaScript is trans compiled into the result.

26. What are generics in TypeScript?

Generics are a feature of many [programming languages](#) that define the same functions of variables for different kinds — they are not exclusive to Typescript. Normally, you write a function for an operation that accepts an integer, but the compiler



generalizing the types, generics accomplish the same goal for you.

Imagine you wish to create an integer data structure. Later, you want to create a string data structure. This results in a significant amount of the same code being repeated twice in the majority of the languages, which requires double the work. This extra effort is reduced while using Generics in your programs.

27. What is JSX?

JSX is an embedded JavaScript that can be converted from an embeddable XML-like syntax.

28. What are assertions in programming?

Assertions are conditions that must be met in order for the program to function properly. It's common to use assertions as prerequisites before or after calling a function or method.

Preconditions list prerequisites that must exist to successfully call the function or method. Post-conditions are conditions placed on the called function or method used to ensure a correct calculation. Assertions guarantee the code is not broken because of an invalid circumstance.

29. What rules apply for declaring Rest parameters?

Here are the three rules to declare Rest parameters:

- In a function, only one rest parameter is permitted.
- It must be a type of array.
- It must be the last argument in a parameter list.



The 'as' keyword type assertion prompts the compiler to address the object differently from how its type describes it.

TypeScript Advanced Interview Questions

Let us now throw light on some commonly asked TypeScript interview questions for experienced developers.

31. What does TypeScript's method overriding mean?

Method overriding is the practice of adding a method to a subclass or child class that already exists in the parent class. The base class methods are essentially redefined in the derived class or child class.

Rules for Overriding the Method:

- The method's name has to match that of the parent class.
- The parameter in the overriding method must be consistent with those in the parent class.
- An inheritance or an IS-A relationship must exist.

32. What are TypeScript ambients, and when are they required?

Ambient decorations inform the compiler about real source code located elsewhere. If the source changes, you must update the documentation; otherwise, you'll run into compiler issues.

33. What is the Lambda/Arrow function?

A function without a name is known as a lambda



anonymous functions — lambda functions, or arrow functions.

34. Explain TypeScript decorators.

Class, method, accessor, property, and parameter declarations can all be given a specific treatment by applying a decorator. The decorator annotates and includes metadata to the code.

35. How are optional parameters supported in functions in TypeScript?

You can use optional parameters with the question mark symbol ('?') to avoid errors when calling functions without specifying parameter kinds. It shows optional parameters can be denoted by appending a "?" to those that may or may not receive a value.

36. Describe the difference between method overriding and method overloading?

Method overloading: When two or more methods in a class share the same name but have distinct parameters.

Method overriding: Refers to two methods that have the same parameters but distinct implementations in derived classes and superclasses.

37. What is the difference between argument and parameter?

Although these two terms are frequently used synonymously in [programming languages](#), they're not the same.

Parameter (also known as a formal parameter) refers to the variable specified in the function declaration.



For instance, the function definition "f(x) = x*10" uses the variable "x" as a parameter, and the function call "f(2)" uses the value "2" as the function argument. You can think of an argument as an instance and a parameter as a type.

38. What type of language is TypeScript?

TypeScript is an object-oriented, compiled language and JavaScript superset.

39. Does TypeScript support object-oriented programming concepts?

Yes, TypeScript supports object-oriented programming concepts like classes, interfaces, and inheritance.

40. How many components does TypeScript have?

TypeScript has three components:

- Language
- TypeScript Compiler
- TypeScript Language Service

41. How many types of comments does TypeScript support?

TypeScript supports single- and multi-line comments.

42. Are semicolons optional in TypeScript?

Yes, semicolons are optional in TypeScript.

43. What are the two types of enums?

Numeric enum and **String** enum are two types of enums



Numeric enum's first value is defaulted to 0.

45. What are TypeScript's three "simple types"?

Boolean, Number, and String are the three main 'simple types' in TypeScript.

46. What is the Facade Design Pattern?

A system's complexity is hidden by a facade pattern, making it easier to use. It adheres to a pattern of structural design where client programs can access the system. However, it conceals how it functions by giving clients a more user-friendly interface.

47. What is a closure?

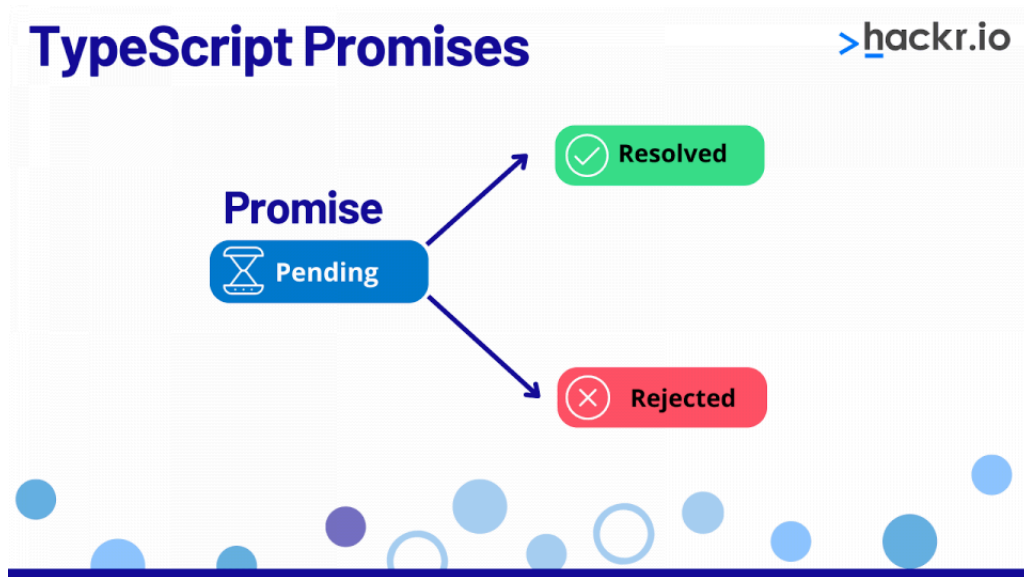
Simply said, a closure is a function that can access its outside scope even when operating outside of it. Every time a function is declared inside of another function in JavaScript, a closure is produced because each function has its own scope due to the function-scoped nature of the language.

48. How is a synchronous task different from an asynchronous task?

Synchronous: Programs must wait for the call to complete before continuing during synchronous calls. This means your application won't continue to operate until the API answer, which the user can experience as latency or performance lag. Synchronous API calls may be useful if your app has functionality that can only be used after the API response has been received.

Asynchronous: Asynchronous calls do not pause for the server to respond to an API call. A "callback" function is then called while your application is still running when the server responds to the call.





A promise is an object that a module, library, or function returns and that will either be fulfilled or rejected in the future.

50. What are enums?

An enum symbol on a variable means it can only take one value from a predetermined range. You must provide a list of all potential values when declaring an enum.

Will AI Replace Programmers?

Loading ad



Bonus Tips

Practicing TypeScript interview questions and answers will bring you closer to your career goals. But that's not all you can do!

TypeScript interviews might not only entail verbal questions. You might have to demonstrate your skills through coding exercises (TypeScript coding interview questions).

Here are some bonus tips for performing confidently on your TypeScript interview:

- **Pursue a Course:** Why not brush up on your skills with an online course? You might explore platforms like [Coursera](#) and Udacity, or try out some [TypeScript tutorials](#).
- **Build Projects:** If you're a beginner, you might not have too many TypeScript pieces on your portfolio. Luckily, you can add some by building a TypeScript project! Plenty of courses and tutorials offer you the tools to do this, and they act as a fabulous speaking point when elaborating on your skills.
- **Update Your Resume and Cover Letter:** A solid resume and cover letter will score you an interview in most cases. Ensure you update them with relevant, recent experience and demonstrate your work ethic as well. Remember to tailor every cover letter to the job ad, as you'll want to cater each



- **Mock Interviews:** You can practice these TypeScript questions alone, but you'll gain much more if you grab a friend for a mock interview. They can help you catch areas you need to brush up on.

Conclusion

If you're preparing for a TypeScript developer interview, you're perfectly capable of scoring that job! Just practice these TypeScript interview questions alone or with a friend to get comfortable with the material.

Are you applying for other developer jobs that use languages like Java or HTML? We've got you covered!

People are also reading:

- [HTML Interview Questions](#)



Frequently Asked Questions

1. What is TypeScript Used For?

TypeScript is a programming language for developing server-side applications.

2. What is True TypeScript?

True TypeScript performs variable type-checking, unlike JavaScript.

3. Is TypeScript Front End or Back End?

TypeScript is used for developing back-end applications.

4. What is the 'Symbol' in Typescript?

The symbol is a primitive data type in TypeScript. In addition to having no properties or immutable methods, a primitive data type is not an object. Similar types to the symbol type include number, string, boolean, etc. The Symbol function `Object() { [native code] }` is used to produce symbol values.



Explore More

[SHOW ALL](#)

By Sameeksha Medewar

Sameeksha is a freelance content writer for more than half and a year. She has a hunger to explore and learn new things. She possesses a bachelor's degree in Computer Science.

[View all post by the author](#)



[Subscribe](#)

Disclosure: Hackr.io is supported by its audience. When you purchase through links on our site, we may earn an affiliate commission.

In this article

- [Top Typescript Interview Questions for All Experience Levels](#)
- [Bonus Tips](#)
- [Conclusion](#)
- [Frequently Asked Questions](#)

Learn More

Top 48 Networking Interview Questions and Answers in 2024

[Computer Networks](#) [Career Development](#) [Interview Questions](#)



[TypeScript](#) [Courses](#) [Web Development](#)

Top 20 REST API Interview Questions & Answers [2024]

[Web Development](#) [Career Development](#) [Interview Questions](#)

Find Your Place In The World

Mid-Level
Software...
Boeing...
United States

[See Job](#)

Industrial
Operation...
Capgemini
Auburn Hills

[See Job](#)

Software
Developer...
DataAnno...
Cary
From \$40 a...

[See Job](#)

Director of
Paid Media
Citadel...
Exton

[See Job](#)

AI/ML
Engineer
Logistics...
Arlington

[See Job](#)

Cognos
Developer
Envision,...
St. Louis

[See Job](#)



Please login to leave comments

Always be in the loop.

Get news once a week, and don't worry — no spam.

Your email

Subscribe

Programming

DevOps

Data Science

Design

Articles

About us

Roadmaps

We ♥ Feedback



[Privacy Policy](#)

[Cookie Policy](#)

[Change Privacy Settings](#)

[Disclosure Policy](#)

[Terms and Conditions](#)

[Disclaimer](#)

[Refund Policy](#)

Follow us

Disclosure: This page may contain affiliate links, meaning when you click the links and make a purchase, we receive a commission.

A RAPTIVE PARTNER SITE

