aershov24 / **full-stack-interview-questions** Public

🔴 More than ~3877 Full Stack, Coding & System Design Interview Questions And Answers sourced from all around the Internet to help you to prepare to an interview, conduct one, mock your lead dev or completely ignore. Find more questions and answers on 👉

www.fullstack.cafe

☆ **602** stars **201** forks

☆ Star ▾ Notifications

Code Issues 5 Pull requests Actions Security Insights

master ▾ Go to file

aershov fix ... on Aug 26, 2021 **38**

View code

3877 Full-Stack, Coding and System Design Interview Questions (ANSWERED) To Land Your Next Six-Figure Job Offer from FullStack.Cafe

FullStack.Cafe is a biggest hand-picked collection of top technical interview questions for junior and experienced full-stack and web developers with more that 3877 tech interview questions and answers. Prepare for your next tech interview and land 6-figure job offer in no time.

🔴 Get All 3877 Answers + PDFs on FullStack.Cafe - Kill Your Tech & Coding Interview

Machine Learning & Data Science Interview Questions 🤖🤖🤖

👉 For 1299 ML & DataScience Interview Questions Check MLStack.Cafe - Kill Your Machine Learning, Data Science & Python Interview

Table of Contents

- Angular
- AngularJS
- Azure
- Behavioral
- Big Data
- Blockchain
- Bootstrap
- C#
- CSS
- Career
- Clojure
- Code Problems
- Data Science
- Data Structures
- Design Patterns
- DevOps
- Docker
- Entity Framework
- Flutter
- Git
- Golang
- GraphQL
- HTML5
- Ionic
- JSON
- Java
- JavaScript
- Kotlin
- LINQ
- Laravel
- MSMQ
- Machine Learning
- Microservices

[⬆️] .NET Core Interview Questions

Q1: What is the difference between String and string in C#? ⭐

Answer: `string` is an alias in C# for `System.String`. So technically, there is no difference. It's like `int` vs. `System.Int32`.

As far as guidelines, it's generally recommended to use `string` any time you're referring to an object.

```
string place = "world";
```

Likewise, it's generally recommended to use `String` if you need to refer specifically to the class.

```
string greet = String.Format("Hello {0}!", place);
```

Source: *blogs.msdn.microsoft.com*

Q2: What is .NET Standard? ⭐

Answer: The **.NET Standard** is a formal specification of .NET APIs that are intended to be available on all .NET implementations.

Source: *docs.microsoft.com*

Q3: What is .NET Core? ⭐

Answer: The .NET Core platform is a new .NET stack that is optimized for open source development and agile delivery on NuGet.

.NET Core has two major components. It includes a small runtime that is built from the same codebase as the .NET Framework CLR. The .NET Core runtime includes the same GC and JIT (RyuJIT), but doesn't include features like Application Domains or Code Access Security. The runtime is delivered via NuGet, as part of the ASP.NET Core package.

.NET Core also includes the base class libraries. These libraries are largely the same code as the .NET Framework class libraries, but have been factored (removal of dependencies) to enable to ship a smaller set of libraries. These libraries are shipped as `System.*` NuGet packages on NuGet.org.

Source: *stackoverflow.com*

Q4: What is the .NET Framework? ⭐

Answer: The .NET is a Framework, which is a collection of classes of reusable libraries given by Microsoft to be used in other .NET applications and to develop, build and deploy many types of applications on the Windows platform including the following:

- Console Applications
- Windows Forms Applications
- Windows Presentation Foundation (WPF) Applications
- Web Applications
- Web Services
- Windows Services
- Services-oriented applications using Windows Communications Foundation (WCF)
- Workflow-enabled applications using Windows Workflow Foundation(WF)

Source: *c-sharpcorner.com*

Q5: What's the difference between SDK and Runtime in .NET Core? ⭐⭐

Answer:

- The SDK is all of the stuff that is needed/makes developing a .NET Core application easier, such as the CLI and a compiler.

- The runtime is the "virtual machine" that hosts/runs the application and abstracts all the interaction with the base operating system.

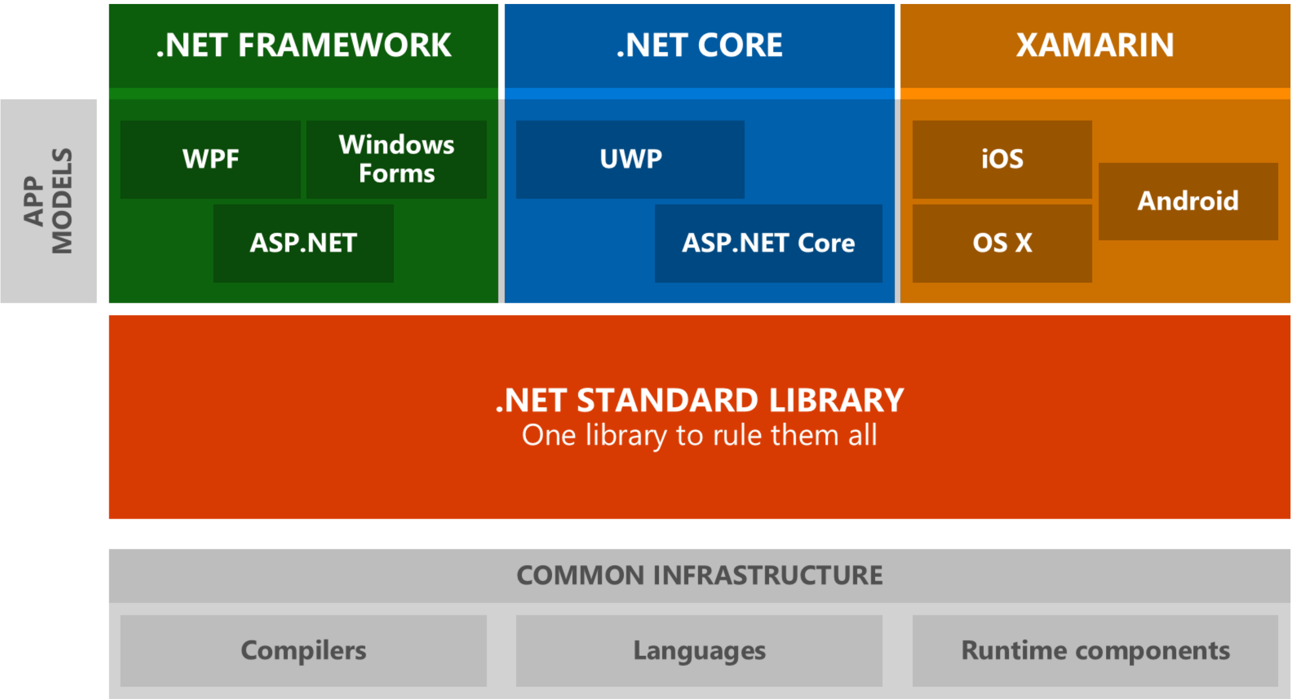
Source: *stackoverflow.com*

Q6: What is .NET Standard and why we need to consider it? ⭐⭐

Answer:

1. **.NET Standard** solves the code sharing problem for .NET developers across all platforms by bringing all the APIs that you expect and love across the environments that you need: desktop applications, mobile apps & games, and cloud services:
2. **.NET Standard** is a **set of APIs** that **all** .NET platforms **have to implement**. This **unifies the .NET platforms** and **prevents future fragmentation**.
3. **.NET Standard 2.0** will be implemented by **.NET Framework**, **.NET Core**, and **Xamarin**. For **.NET Core**, this will add many of the existing APIs that have been requested.
4. **.NET Standard 2.0** includes a compatibility shim for **.NET Framework** binaries, significantly increasing the set of libraries that you can reference from your .NET Standard libraries.

5. **.NET Standard will replace Portable Class Libraries (PCLs)** as the tooling story for building multi-platform .NET libraries.



Source: *stackoverflow.com*

Q7: What is the difference between decimal, float and double in .NET? ⭐⭐

Details: When would someone use one of these?

Answer: Precision is the main difference.

- Float - 7 digits (32 bit)
- Double-15-16 digits (64 bit)
- Decimal -28-29 significant digits (128 bit)

As for what to use when:

- For values which are "naturally exact decimals" it's good to use decimal. This is usually suitable for any concepts invented by humans: financial values are the most obvious example, but there are others too. Consider the score given to divers or ice skaters, for example.

- For values which are more artefacts of nature which can't really be measured exactly anyway, float/double are more appropriate. For example, scientific data would usually be represented in this form. Here, the original values won't be "decimally accurate" to start with, so it's not important for the expected results to maintain the "decimal accuracy". Floating binary point types are much faster to work with than decimals.

Source: *blogs.msdn.microsoft.com*

Q8: What are some characteristics of .NET Core? ⭐⭐

Answer:

- **Flexible deployment**: Can be included in your app or installed side-by-side user- or machine-wide.
- **Cross-platform**: Runs on Windows, macOS and Linux; can be ported to other OSes. The supported Operating Systems (OS), CPUs and application scenarios will grow over time, provided by Microsoft, other companies, and individuals.
- **Command-line tools**: All product scenarios can be exercised at the command-line.
- **Compatible**: .NET Core is compatible with .NET Framework, Xamarin and Mono, via the .NET Standard Library.
- **Open source**: The .NET Core platform is open source, using MIT and Apache 2 licenses. Documentation is licensed under CC-BY. .NET Core is a .NET Foundation project.
- **Supported by Microsoft**: .NET Core is supported by Microsoft, per .NET Core Support

Source: *stackoverflow.com*

Q9: What is an unmanaged resource? ⭐⭐

Answer: Use that rule of thumb:

- If you found it in the Microsoft .NET Framework: *it's managed*.
- If you went poking around MSDN yourself, *it's unmanaged*.

Anything you've used P/Invoke calls to get outside of the nice comfy world of everything available to you in the .NET Framwork is unmanaged – and you're now *responsible* for cleaning it up.

Source: *stackoverflow.com*

Q10: What is CTS? ⭐⭐

Answer: The **Common Type System (CTS)** standardizes the data types of all programming languages using .NET under the umbrella of .NET to a common data type for easy and smooth communication among these .NET languages.

CTS is designed as a singly rooted object hierarchy with `System.Object` as the base type from which all other types are derived. CTS supports two different kinds of types:

1. **Value Types**: Contain the values that need to be stored directly on the stack or allocated inline in a structure. They can be built-in (standard primitive types), user-defined (defined in source code) or enumerations (sets of enumerated values that are represented by labels but stored as a numeric type).
2. **Reference Types**: Store a reference to the value's memory address and are allocated on the heap. Reference types can be any of the pointer types, interface types or self-describing types (arrays and class types such as user-defined classes, boxed value types and delegates).

Source: *c-sharpcorner.com*

Q11: What is the difference between .NET Core and Mono? ⭐⭐

Answer: To be simple:

- Mono is third party implementation of .Net Framework for Linux/Android/iOs
- .Net Core is Microsoft's own implementation for same.

Source: *stackoverflow.com*

Q12: What is MSIL? ⭐⭐

Answer: When we compile our .NET code then it is not directly converted to native/binary code; it is first converted into intermediate code known as MSIL code which is then interpreted by the CLR. MSIL is independent of hardware and the operating system. Cross language relationships are possible since MSIL is the same for all .NET languages. MSIL is further converted into native code.

Source: *c-sharpcorner.com*

Q13: What is a .NET application domain? ⭐⭐

Answer: It is an isolation layer provided by the .NET runtime. As such, App domains live with in a process (1 process can have many app domains) and have their own virtual address space.

App domains are useful because:

- They are less expensive than full processes
- They are multithreaded

- You can stop one without killing everything in the process
- Segregation of resources/config/etc
- Each app domain runs on its own security level

Source: *stackoverflow.com*

Q14: What is CLR? ⭐⭐

Answer: The **CLR** stands for Common Language Runtime and it is an Execution Environment. It works as a layer between Operating Systems and the applications written in .NET languages that conforms to the Common Language Specification (CLS). The main function of Common Language Runtime (CLR) is to convert the Managed Code into native code and then execute the program.

Source: *c-sharpcorner.com*

Q15: Name some CLR services? ⭐⭐

Answer: CLR services

- Assembly Resolver
- Assembly Loader
- Type Checker
- COM marshalled
- Debug Manager
- Thread Support
- IL to Native compiler
- Exception Manager
- Garbage Collector

Source: *c-sharpcorner.com*

Q16: Talk about new .csproj file? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q17: Explain what is included in .NET Core? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q18: Is there a way to catch multiple exceptions at once and without code duplication? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q19: What about NuGet packages and packages.config? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q20: Why to use of the IDisposable interface? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q21: When should we use .NET Core and .NET Standard Class Library project types? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q22: What is the difference between Class Library (.NET Standard) and Class Library (.NET Core)? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q23: What is the difference between .NET Standard and PCL (Portable Class Libraries)? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q24: Explain the difference between Task and Thread in .NET ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q25: What is FCL? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q26: What is Kestrel? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q27: What is implicit compilation? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q28: What is JIT compiler? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q29: What is .NET Standard? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q30: What is Explicit Compilation? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q31: What are the benefits of explicit compilation? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q32: Explain the difference between “managed” and “unmanaged” code? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q33: What's the difference between .NET Core, .NET Framework, and Xamarin? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q34: What is difference between .NET Core and .NET Framework? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q35: Explain two types of deployment for .NET Core applications ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q36: What does Common Language Specification (CLS) mean? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q37: What is CoreCLR? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q38: What's is BCL? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q39: What is the difference between CIL and MSIL (IL)? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q40: How to choose the target version of .NET Standard library? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q41: Why does .NET use a JIT compiler instead of just compiling the code once on the target machine? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q42: What is the difference between AppDomain, Assembly, Process, and a Thread? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q43: What are benefits of using JIT? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q44: What is the difference between .NET Framework/Core and .NET Standard Class Library project types? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q45: What's the difference between RyuJIT and Roslyn? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q46: Why does .NET Standard library exist? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q47: Explain how does Asynchronous tasks (Async/Await) work in .NET? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q48: Explain Finalize vs Dispose usage? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q49: How many types of JIT Compilations do you know? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q50: What is the difference between Node.js async model and async/await in .NET? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q51: Could you name the difference between .Net Core, Portable, Standard, Compact, UWP, and PCL? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] ADO.NET Interview Questions

Q1: What is ADO.NET? ⭐

Answer: ADO stands for Active Data Object and ADO.NET is a set of .NET libraries for ADO. NET is a collection of managed libraries used by .NET applications for data source communication using a driver or provider:

- Enterprise applications handle a large amount of data. This data is primarily stored in relational databases, such as Oracle, SQL Server, and Access and so on. These databases use Structured Query Language (SQL) for retrieval of data.
- To access enterprise data from a .NET application, an interface was needed. This interface acts as a bridge between an RDBMS system and a .NET application. ADO.NET is such an interface that is created to connect .NET applications to RDBMS systems.
- In the .NET framework, Microsoft introduced a new version of Active X Data Objects (ADO) called ADO.NET. Any .NET application, either Windows based or web based, can interact with the database using a rich set of classes of the ADO.NET library. Data can be accessed from any database using connected or disconnected architecture.

Source: *c-sharpcorner.com*

Q2: What is exactly meaning of disconnected and connected approach in ADO.NET? ⭐⭐

Answer: In short:

- **Disconnected** = Make Connection , Fetch Data , Close Connection
- **Connected** = Make Connection , Keep Connection alive , Close Connection when close is called.

The ADO.net architecture, in which connection must be kept open till the end to retrieve and access data from database is called as *connected architecture*. Connected architecture is built on the these types - `connection` , `command` , `datareader`

The ADO.net architecture, in which connection will be kept open only till the data retrieved from database, and later can be accessed even when connection to database is closed is called as *disconnected architecture*. Disconnected architecture of ADO.net is built on these types - `connection`, `dataadapter`, `commandbuilder` and `dataset` and `dataview`.

Source: *stackoverflow.com*

Q3: Describe when you would use the DataView in ADO.NET? ⭐⭐

Answer: A **DataView** enables you to create different views of the data stored in a DataTable, a capability that is often used in data binding applications. Using a DataView, you can expose the data in a table with different sort orders, and you can filter the data by row state or based on a filter expression. A DataView provides a dynamic view of data whose content, ordering, and membership reflect changes to the underlying DataTable as they occur. This is different from the Select method of the DataTable, which returns a DataRow array from a table per particular filter and/or sort order and whose content reflects changes to the underlying table, but whose membership and ordering remain static. The dynamic capabilities of the DataView make it ideal for data-binding applications.

Source: *stackoverflow.com*

Q4: What is the SqlCommandBuilder? ⭐⭐

Answer: CommandBuilder helps you to generate update, delete, and insert commands on a single database table for a data adapter. Similar to other objects, each data provider has a command builder class. The OleDbCommandBuilder, SqlCommonBuilder, and OdbcCommandBuilder classes represent the CommonBuilder object in the OleDb, Sql, and ODBC data providers.

Source: *c-sharpcorner.com*

Q5: What is the DataAdapter Object in ADO.NET? ⭐⭐

Answer: A **DataAdapter** is used to retrieve data from a data source and populate tables within a `DataSet`. Data Adapters form the bridge between a data source and a dataset. The `DataAdapter` also resolves changes made to the `DataSet` back to the data source. The `DataAdapter` uses the `Connection` object of the .NET Framework data provider to connect to a data source, and it uses `Command` objects to retrieve data from and resolve changes to the data source.

A `DataAdapter` supports mainly the following two methods:

- **Fill():** The Fill method populates a dataset or a data table object with data from the database. It retrieves rows from the data source using the SELECT statement specified by an associated select command property. The Fill method leaves the connection in the same state as it encountered before populating the data.
- **Update():** The Update method commits the changes back to the database. It also analyzes the RowState of each record in the DataSet and calls the appropriate INSERT, UPDATE, and DELETE statements.

Source: *c-sharpcorner.com*

Q6: What is the basic difference between ADO.NET and Entity Framework? ⭐⭐

Answer: ADO.NET Entity Framework is an ORM (object-relational mapping) which creates a higher abstract object model over ADO.NET components. ADO.NET is a layer closer to the database (datatables, datasets and etc...). The main and the only benefit of EF is it auto-generates code for the Model (middle layer), Data Access Layer, and mapping code, thus reducing a lot of development time. Consider the following example:

ADO.NET:

```
DataTable table = adoDs.Tables[0];
for (int j = 0; j < table.Rows.Count; j++)
{
    DataRow row = table.Rows[j];

    // Get the values of the fields
    string CustomerName =
        (string)row["Customername"];
    string CustomerCode =
        (string)row["CustomerCode"];
}
```

EF:

```
foreach (Customer objCust in obj.Customers)
{}
```

Source: *stackoverflow.com*

Q7: What is Connection Pooling in ADO.NET? ⭐⭐

Answer: ADO.NET uses a technique called **connection pooling**, which minimizes the cost of repeatedly opening and closing connections. Connection pooling reuses existing active connections with the same connection string instead of creating new connections when a request is made to the database. It involves the use of a connection manager that is responsible for maintaining a list, or pool, of available connections for a given connection string. Several pools exist if different connection strings ask for connection pooling.

Source: *c-sharpcorner.com*

Q8: What is SqlCommand Object? ⭐⭐

Answer: The **SqlCommand** carries the SQL statement that needs to be executed on the database. SqlCommand carries the command in the CommandText property and this property will be used when the SqlCommand calls any of its execute methods.

- The Command Object uses the connection object to execute SQL queries.
- The queries can be in the form of Inline text, Stored Procedures or direct Table access.
- An important feature of Command object is that it can be used to execute queries and Stored Procedures with Parameters.
- If a select query is issued, the result set it returns is usually stored in either a DataSet or a DataReader object.

The three important methods exposed by the SqlCommand object is shown below:

- ExecuteScalar
- ExecuteNonQuery
- ExecuteReader

Source: *c-sharpcorner.com*

Q9: What are the ADO.NET components? ⭐⭐

Answer: ADO.NET components categorized in three modes:

- disconnected,
- common or shared and
- the .NET data providers.

The disconnected components build the basic ADO.NET architecture. You can use these components (or classes) with or without data providers. For example, you can use a `DataTable` object with or without providers and shared or common components are the base classes for data providers. Shared or common components are the base classes for data providers and shared by all data providers. The data provider components are specifically designed to work with different kinds of data sources. For example, ODBC data providers work with ODBC data sources and OleDb data providers work with OLE-DB data sources.

Source: *c-sharpcorner.com*

Q10: How can you define the DataSet structure? ⭐⭐

Answer: A **DataSet** object falls in disconnected components series. The `DataSet` consists of a collection of tables, rows, columns and relationships.

`DataSet` contains a collection of `DataTables` and the `DataTable` contains a collection of `DataRows`, `DataRelations`, and `DataColumns`. A `DataTable` maps to a table in the database.

Source: *c-sharpcorner.com*

Q11: What do you understand by DataRelation class? ⭐⭐

Answer: The **DataRelation** is a class of disconnected architecture in the .NET framework. It is found in the System.Data namespace. It represents a relationship between database tables and correlates tables on the basis of matching column.

Source: *c-sharpcorner.com*

Q12: How could you control connection pooling behavior? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q13: What is the difference between ExecuteScalar, ExecuteReader and ExecuteNonQuery? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q14: What is the difference between DataView, DataTable and DataSet? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q15: Could you explain me some of the main differences between Connection-oriented access and connectionless access in ADO.NET? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q16: What is the difference between Integrated Security = True and Integrated Security = SSPI? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q17: What is Unit Of Work? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q18: What are the differences between using SqlDataAdapter vs SqlDataReader for getting data from a DB? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q19: Mention what is the difference between ADO.NET and classic ADO? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q20: Can you explain the difference between a DataReader, a DataAdapter, a Dataset, and a DataView? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q21: Name types of transactions in ADO.NET ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q22: Where should I use disconnected architecture approach? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q23: Where should I use connected architecture approach? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q24: Is there anything faster than SqlDataReader in .NET? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q25: Could you explain some benefits of Repository Pattern? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q26: What is the difference between OLE DB and ODBC data sources? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q27: How could you monitor connection pooling behavior? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q28: Is it necessary to manually close and dispose of SqlDataReader? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q29: What's better: DataSet or DataReader? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q30: What is the difference between ADODB, OLEDB and ADO.NET? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q31: What is the best and fast way to insert 2 million rows of data into SQL Server? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q32: Under what scenarios would setting pooling=false in an ADO.NET connection string be of value when connecting to SQL Server? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q33: Name some problems that could occur with connection pooling ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] ASP.NET Interview Questions

Q1: What is ViewData? ⭐

Answer: Viewdata contains the key, value pairs as dictionary and this is derived from class—“ViewDataDictionary“. In action method we are setting the value for viewdata and in view the value will be fetched by typecasting.

Source: *medium.com*

Q2: What is ASP.Net? ⭐

Answer: It is a framework developed by Microsoft on which we can develop new generation web sites using web forms(aspx), MVC, HTML, Javascript, CSS etc. Its successor of Microsoft Active Server Pages(ASP). Currently there is ASP.NET 4.0, which is used to develop web sites. There are various page extensions provided by Microsoft that are being used for web site development. Eg: aspx, asmx, ascx, ashx, cs, vb, html, XML etc.

Source: *guru99.com*

Q3: Talk about Logging in ASP.NET Core? ⭐⭐

Answer: Logging is built-in and you get access to structured logs from the ASP.NET Core host itself to your application. With tools like Serilog, you can extend your logging easily and save your logs to file, Azure, Amazon or any other output provider. You can configure verbosity and log levels via configuration (appsettings.json by default), and you can configure log levels by different categories.

Source: *talkingdotnet.com*

Q4: Explain startup process in ASP.NET Core? ⭐⭐

Answer: Everything starts from Program.cs

```
public static void Main(string[] args)
{
    BuildWebHost(args).Run();
}

public static IWebHost BuildWebHost(string[] args) =>
    WebHost.CreateDefaultBuilder(args)
        .UseStartup<Startup>()
        .Build();
```

CreateDefaultBuilder extension method will create a default configuration which will look first into `appsettings.json` files then will look for Environment variables and at the end, it will use command line arguments.

This part will also set up default logger sources (debug and console) and load the settings for logging from appsettings.json.

After the `CreateDefaultBuilder` finishes, then `Startup` class is executed. First, the constructor code is executed. After that, services are added to DI container via `AddServices` method that lives in Startup class. After that, an order of middleware that will handle every incoming request is set up.

Source: *codingblast.com*

Q5: What exactly is an application pool? What is its purpose? ⭐⭐

Answer: Application pools allow you to isolate your applications from one another, even if they are running on the same server. This way, if there is an error in one app, it won't take down other applications.

Additionally, applications pools allow you to separate different apps which require different levels of security.

Source: *stackoverflow.com*

Q6: How you can add an event handler? ⭐⭐

Answer: ** **Using the Attributes property of server side control.

e.g.

```
btnSubmit.Attributes.Add("onMouseOver","JavascriptCode();")
```

Source: *guru99.com*

Q7: What's the use of Response.Output.Write()? ⭐⭐

Answer: We can write formatted output using Response.Output.Write().

Source: *guru99.com*

Q8: How to configure your ASP.NET Core app? ⭐⭐

Answer: Another crucial part of ASP.NET Core Framework is Configuration. Also, it is part of Dependency Injection. Use it anywhere in your code with an option to reload on changes of configuration values from sources (appsettings.json, environment variables, command line arguments, etc.). It is also easy to override, extend and customize the Configuration. No more extensive configurations in web.config, the preferred way now is ***appsettings.json*** in combination with a mix of Environment variables and cmd-line args.

Source: *talkingdotnet.com*

Q9: What is ASP.NET Core? ⭐⭐

Answer: ASP.NET Core is a brand new cross-platform web framework built with .NET Core framework. It is not an update to existing ASP.NET framework. It is a complete rewrite of the ASP.NET framework. It works with both .NET Core and .NET Framework.

Main characterestics of ASP.NET Core:

- DI Container which is quite simple and built-in. You can extend it with other popular DI containers
- Built-in and extensible structured logging. You can redirect output to as many sources as you want (file, Azure, AWS, console)
- Extensible strongly typed configuration, which can also be used to reload at run-time
- Kestrel – new, cross-platform and super fast web server which can stand alone without IIS, Nginx or Apache
- New, fully async pipeline. It is easily configured via middleware
- ASP.NET All meta package which improves development speed, and enables you to reference all Microsoft packages for ASP.NET Core and it will deploy only those that are being used by your code
- There is no *web.config*. We now use *appsettings.json* file in combination with other sources of configuration (command line args, environment variables, etc.)
- There is no _Global._asax – We have *Startup.cs* which is used to set up Middleware and services for DI Container.

Source: *talkingdotnet.com*

Q10: What is the difference between ASP.NET and ASP.NET MVC? ⭐⭐

Answer: ASP.NET, at its most basic level, provides a means for you to provide general HTML markup combined with server side "controls" within the event-driven programming model that can be leveraged with VB, C#, and so on. You define the page(s) of a site, drop in the controls, and provide the programmatic plumbing to make it all work.

ASP.NET MVC is an application framework based on the Model-View-Controller architectural pattern. This is what might be considered a "canned" framework for a specific way of implementing a web site, with a page acting as the "controller" and dispatching requests to the appropriate pages in the application. The idea is to "partition" the various elements of the application, eg business rules, presentation rules, and so on.

Think of the former as the "blank slate" for implementing a site architecture you've designed more or less from the ground up. MVC provides a mechanism for designing a site around a pre-determined "pattern" of application access, if that makes sense. There's more technical detail to it than that, to be sure, but that's the nickel tour for the purposes of the question.

Source: *stackoverflow.com*

Q11: What is ViewState? ⭐⭐

Answer: View State is the method to preserve the Value of the Page and Controls between round trips. It is a Page-Level State Management technique. View State is turned on by default and normally serializes the data in every control on the page regardless of whether it is actually used during a post-back.

A web application is stateless. That means that a new instance of a page is created every time when we make a request to the server to get the page and after the round trip our page has been lost immediately

Source: *c-sharpcorner.com*

Q12: Can ASP.NET Core work with the .NET framework? ⭐⭐

Answer: Yes. This might surprise many, but ASP.NET Core works with .NET framework and this is officially supported by Microsoft.

ASP.NET Core works with:

- .NET Core framework
- .NET framework

Source: *talkingdotnet.com*

Q13: What is the good practice to implement validations in aspx page? ⭐⭐

Answer: Client-side validation is the best way to validate data of a web page. It reduces the network traffic and saves server resources.

Source: *guru99.com*

Q14: What is a postback? ⭐⭐

Answer: A **postback** originates from the client browser. Usually one of the controls on the page will be manipulated by the user (a button clicked or dropdown changed, etc), and this control will initiate a postback. The state of this control, plus all other controls on the page (known as the View State) is Posted Back to the web server.

Source: *stackoverflow.com*

Q15: What is the file extension of ASP.NET web service? ⭐⭐

Answer: Web services have file extension `.asmx`.

Source: *guru99.com*

Q16: How can we prevent browser from caching an ASPX page? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q17: In which event of page cycle is the ViewState available? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q18: From which base class all Web Forms are inherited? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q19: What is the meaning of Unobtrusive JavaScript? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q20: Explain JSON Binding? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q21: What is new in ASP.NET Core 2, compared to ASP.NET Core 1? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q22: What are the sub types of ActionResult? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q23: What exactly is the difference between .NET Core and ASP.NET Core? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q24: What is the difference between Server.Transfer and Response.Redirect? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q25: Where the viewstate is stored after the page postback? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q26: How do you register JavaScript for webcontrols? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q27: Explain usage of Dependency Injection in ASP.NET Core ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q28: What are the different validators in ASP.NET? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q29: List the events in ASP.NET page life cycle ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q30: What is ViewState? How is it encoded? Is it encrypted? Who uses ViewState? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q31: Can we add code files of different languages in App_Code folder? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q32: What are the different types of caching? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q33: What are the event handlers that we can have in Global.asax file? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q34: Explain Middleware in ASP.NET Core? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q35: How long the items in ViewState exists? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q36: What is the difference between an HtmlInputCheckBox control and an HtmlInputRadioButton control? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q37: In which event are the controls fully loaded? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q38: Which type if caching will be used if we want to cache the portion of a page instead of whole page? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q39: How we can force all the validation controls to run? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q40: List the major built-in objects in ASP.NET? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q41: What is HttpModule in ASP.Net? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q42: What is RedirectPermanent in ASP.Net? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q43: What are the different types of cookies in ASP.NET? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q44: What is the difference between <system.web> and <system.webServer>? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q45: What is an HttpHandler in ASP.NET? Why and how is it used? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q46: What is the difference between Web Service and WCF Service? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q47: What is the difference between web config and machine config? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q48: Is it possible to create web application with both webforms and mvc? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q49: What is the difference between ASP.NET Core Web (.NET Core) vs ASP.NET Core Web (.NET Framework)? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q50: What are the different Session state management options available in ASP.NET? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q51: What is the difference between 'classic' and 'integrated' pipeline mode in IIS7? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q52: How can we apply Themes to an asp.net application? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q53: How to choose between ASP.NET 4.x and ASP.NET Core? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q54: What is Katana? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q55: What exactly is OWIN and what problems does it solve? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q56: Name some ASP.NET WebForms disadvantages over MVC? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q57: What is the difference between a web API and a web service? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q58: What is Cross Page Posting? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q59: What is the equivalent of WebForms in ASP.NET Core? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q60: Are static class instances unique to a request or a server in ASP.NET? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] ASP.NET MVC Interview Questions

Q1: What is Layout in MVC? ⭐

Answer: Layout pages are similar to master pages in traditional web forms. This is used to set the common look across multiple pages. In each child page we can find—

```
@{
Layout = “~/Views/Shared/TestLayout1.cshtml”;
}
```

This indicates child page uses TestLayout page as it’s master page.

Source: *medium.com*

Q2: Explain Bundle.Config in MVC4? ⭐⭐

Answer: “BundleConfig.cs” in MVC4 is used to register the bundles by the bundling and minification system. Many bundles are added by default including jQuery libraries like—jquery.validate, Modernizr, and default CSS references.

Source: *medium.com*

Q3: What is Razor View Engine? ⭐⭐

Answer: Razor is the first major update to render HTML in MVC 3. Razor was designed specifically for view engine syntax. Main focus of this would be to simplify and code-focused templating for HTML generation. Below is the sample of using Razor:

```
@model MvcMusicStore.Models.Customer
@{ViewBag.Title = “Get Customers”;}
<div class=”cust”> <h3><em>@Model.CustomerName</em> </h3>
```

Source: *medium.com*

Q4: What is the use of ViewModel in MVC? ⭐⭐

Answer: ViewModel is a plain class with properties, which is used to bind it to strongly typed view. ViewModel can have the validation rules defined for its properties using data annotations.

Source: *medium.com*

Q5: What you mean by Routing in MVC? ⭐⭐

Answer: Routing is a pattern matching mechanism of incoming requests to the URL patterns which are registered in route table. Class—“UrlRoutingModule” is used for the same process.

Source: *medium.com*

Q6: What are Actions in MVC? ⭐⭐

Answer: Actions are the methods in Controller class which is responsible for returning the view or json data. Action will mainly have return type—“ActionResult” and it will be invoked from method—“InvokeAction()” called by controller.

Source: *medium.com*

Q7: What are the advantages of MVC over ASP.NET? ⭐⭐

Answer:

- Provides a clean separation of concerns among UI (Presentation layer), model (Transfer objects/Domain Objects/Entities) and Business Logic (Controller).
- Easy to UNIT Test
- Improved reusability of model and views. We can have multiple views which can point to the same model and vice versa.

- Improved structuring of the code

Source: *medium.com*

Q8: What are Scaffold templates in MVC? ⭐⭐

Answer: Scaffolding in ASP.NET MVC is used to generate the Controllers, Model and Views for create, read, update, and delete (CRUD) functionality in an application. The scaffolding will be knowing the naming conventions used for models and controllers and views.

Source: *medium.com*

Q9: Can you explain Model, Controller and View in MVC? ⭐⭐

Answer:

- **Model**—It's a business entity and it is used to represent the application data.
- **Controller**—Request sent by the user always scatters through controller and it's responsibility is to redirect to the specific view using View() method.
- **View**—It's the presentation layer of MVC.

Source: *medium.com*

Q10: What is Razor Pages? ⭐⭐

Answer: Razor Pages is a new feature of ASP.NET Core that makes coding page-focused scenarios easier and more productive.

With Razor Pages, you have this one Razor file (*.cshtml*), and the code for a single page lives inside of that file, and that file also represents the URL structure of the app. Therefore, you got everything inside of one file, and it just works.

However, you can separate your code to the *code behind* file with *.cshtml.cs* extension. You would usually have your view model and handlers (like action methods in MVC) in that file and handle the logic there. Of course, you could also have your view model moved to separate place.

Since Razor Pages is part of the MVC stack, you can use anything that comes with MVC inside of our Razor Pages.

Source: *codingblast.com*

Q11: Explain Sections is MVC? ⭐⭐

Answer: Section are the part of HTML which is to be rendered in layout page. In Layout page we will use the below syntax for rendering the HTML –

```
@RenderSection(“TestSection”)
```

And in child pages we are defining these sections as shown below –

```
@section TestSection{
     <h1>Test Content</h1>
}
```

Source: *medium.com*

Q12: What are Non Action methods in MVC? ⭐⭐

Answer: In MVC all public methods have been treated as Actions. So if you are creating a method and if you do not want to use it as an action method then the method has to be decorated with "NonAction" attribute as shown below:

```
[NonAction]
public void TestMethod()
{
   // Method logic
}
```

Source: *stackoverflow.com*

Q13: Can a view be shared across multiple controllers? If Yes, How we can do that? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q14: What is the difference between ViewBag and ViewData in MVC? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q15: What is the difference between ViewResult() and ActionResult() in ASP.NET MVC? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q16: What are HTML Helpers in MVC? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q17: Can you explain the page life cycle of MVC? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q18: What is Attribute Routing in MVC? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q19: What is PartialView in MVC? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q20: Can you explain RenderBody and RenderPage in MVC? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q21: Explain the methods used to render the views in MVC? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q22: Explain ASP.NET WebApi vs MVC? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q23: What are some of the advantages of using ASP.Net MVC vs Web Forms? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q24: What is the "HelperPage.IsAjax" Property? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q25: What is RouteConfig.cs in MVC 4? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q26: Why to use Html.Partial in MVC? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q27: What are Validation Annotations? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q28: Explain TempData in MVC? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q29: How route table has been created in ASP.NET MVC? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q30: Explain Dependency Resolution? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q31: What is Separation of Concerns in ASP.NET MVC? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q32: What are AJAX Helpers in MVC? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q33: What is Html.RenderPartial? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] ASP.NET Web API Interview Questions

Q1: What is ASP.NET Web API? ⭐

Answer: ASP.NET Web API is a framework that simplifies building HTTP services for broader range of clients (including browsers as well as mobile devices) on top of .NET Framework.

Using ASP.NET Web API, we can create non-SOAP based services like plain XML or JSON strings, etc. with many other advantages including:

- Create resource-oriented services using the full features of HTTP
- Exposing services to a variety of clients easily like browsers or mobile devices, etc.

Source: *codeproject.com*

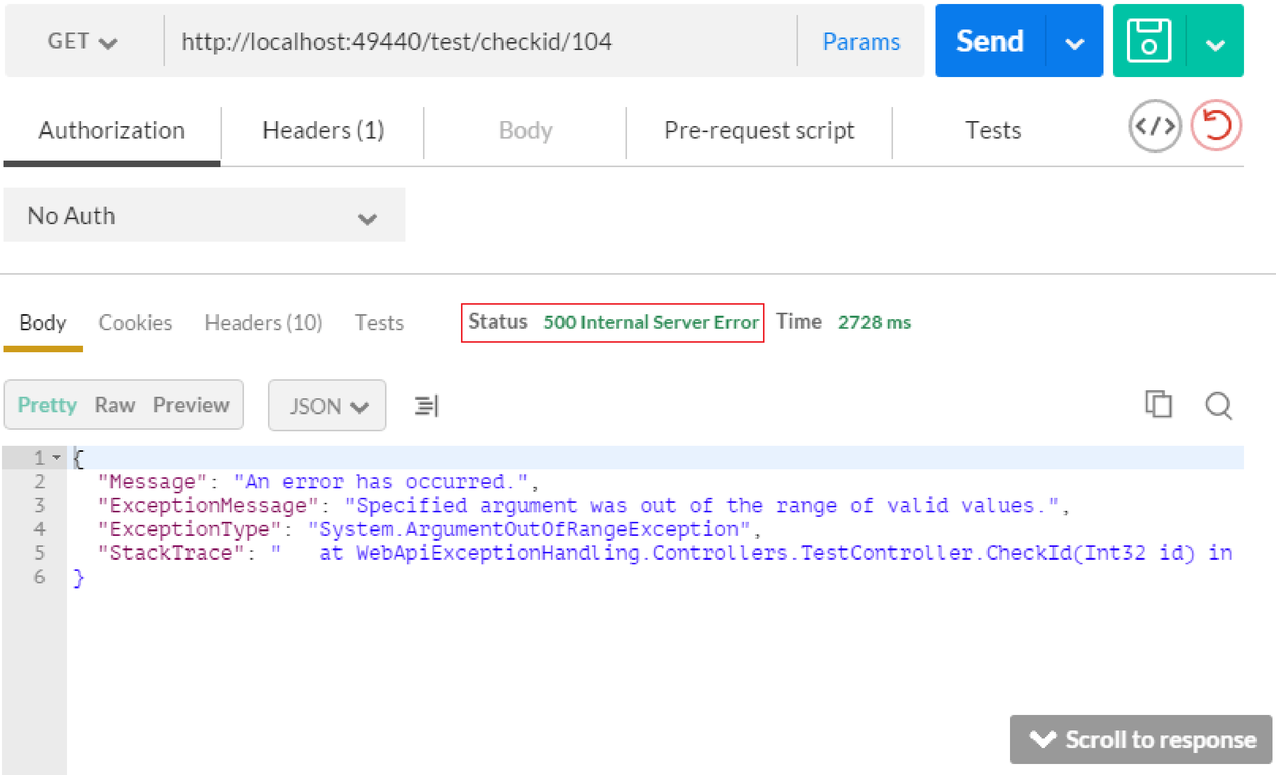
Q2: Which status code used for all uncaught exceptions by default? ⭐⭐

Answer: 500 – Internal Server Error

Consider:

```
[Route("CheckId/{id}")]
[HttpGet]
public IHttpActionResult CheckId(int id)
{
    if(id > 100)
    {
        throw new ArgumentOutOfRangeException();
    }
    return Ok(id);
}
```

And the result:



http://localhost:49440/test/checkid/104

No Auth

Status 500 Internal Server Error Time 2728 ms

```
{
  "Message": "An error has occurred.",
  "ExceptionMessage": "Specified argument was out of the range of valid values.",
  "ExceptionType": "System.ArgumentOutOfRangeException",
  "StackTrace": "   at WebApiExceptionHandling.Controllers.TestController.CheckId(Int32 id) in
}
```

Source: *docs.microsoft.com*

Q3: What are the Advantages of Using ASP.NET Web API? ⭐⭐

Answer: Using ASP.NET Web API has a number of advantages, but core of the advantages are:

- It works the HTTP way using standard HTTP verbs like `GET` , `POST` , `PUT` , `DELETE` , etc. for all CRUD operations
- Complete support for routing
- Response generated in JSON or XML format using `MediaTypeFormatter`
- It has the ability to be hosted in IIS as well as self-host outside of IIS

- Supports Model binding and Validation
- Support for OData

Source: *codeproject.com*

Q4: What New Features are Introduced in ASP.NET Web API 2.0? ⭐⭐

Answer: More new features introduced in ASP.NET Web API framework v2.0 are as follows:

- Attribute Routing
- External Authentication
- CORS (Cross-Origin Resource Sharing)
- OWIN (Open Web Interface for .NET) Self Hosting
- `IHttpActionResult`
- Web API OData

Source: *codeproject.com*

Q5: What exactly is OAuth (Open Authorization)? ⭐⭐

Answer: OAuth (Open Authorization) is an open standard for access granting/deligation protocol. It used as a way for Internet users to grant websites or applications access to their information on other websites but without giving them the passwords. It does not deal with authentication.

Basically there are three parties involved: oAuth Provider, oAuth Client and Owner.

- oAuth Client (Application Which wants to access your credential)
- oAuth Provider (eg. facebook, twitter...)
- Owner (the person with facebook,twitter.. account)

Source: *stackoverflow.com*

Q6: Explain the usage of HttpResponseMessage? ⭐⭐

Answer: `HttpResponseMessage` works with HTTP protocol to return the data with status/error.

Source: *c-sharpcorner.com*

Q7: What is the difference between ApiController and Controller? ⭐⭐

Answer:

- Use **Controller** to render your normal views.
- **ApiController** action only return data that is serialized and sent to the client.

Consider:

```
public class TweetsController : Controller {
  // GET: /Tweets/
  [HttpGet]
  public ActionResult Index() {
    return Json(Twitter.GetTweets(), JsonRequestBehavior.AllowGet);
  }
}
```

or

```
public class TweetsController : ApiController {
  // GET: /Api/Tweets/
  public List<Tweet> Get() {
    return Twitter.GetTweets();
  }
}
```

Source: *stackoverflow.com*

Q8: What are main return types supported in Web API? ⭐⭐

Answer: A Web API controller action can return following values:

- Void – It will return empty content
- HttpResponseMessage – It will convert the response to an HTTP message.
- IHttpActionResult – internally calls ExecuteAsync to create an HttpResponseMessage
- Other types – You can write the serialized return value into the response body

Source: *career.guru99.com*

Q9: What are the differences between WebAPI and WebAPI 2? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q10: How to Restrict Access to Web API Method to Specific HTTP Verb? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q11: What is Attribute Routing in ASP.NET Web API 2.0? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q12: Name types of Action Results in Web API 2 ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q13: Compare WCF vs ASP.NET Web API? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q14: Explain the difference between WCF RESTful Service vs ASP.NET Web API? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q15: Is it True that ASP.NET Web API has Replaced WCF? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q16: What's the difference between REST & RESTful? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q17: Explain the difference between MVC vs ASP.NET Web API ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q18: Why are the "FromBody" and "FromUri" attributes needed in ASP.NET Web API`? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q19: What is ASP.NET Web API OData? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q20: Explain briefly CORS(Cross-Origin Resource Sharing)? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q21: Can we use Web API with ASP.NET Web Form? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q22: How Can We Provide an Alias Name for ASP.NET Web API Action? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q23: What is Delegating Handler? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q24: How to register exception filter globally? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q25: What's the difference between OpenID and OAuth? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q26: How to Return View from ASP.NET Web API Method? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q27: Explain advantages/disadvantages of using HttpModule vs DelegatingHandler? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q28: Could you clarify what is the best practice with Web API error management? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q29: What is difference between OData and REST web services? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q30: Explain briefly OWIN (Open Web Interface for .NET) Self Hosting? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q31: Explain the difference between WCF, Web API, WCF REST and Web Service? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q32: Why should I use IHttpActionResult instead of HttpResponseMessage? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q33: What is difference between WCF and Web API and WCF REST and Web Service? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] AWS Interview Questions

Q1: What is AWS? ⭐

Answer: AWS stands for Amazon Web Services and is a platform that provides database storage, secure cloud services, offering to compute power, content delivery, and many other services to develop business levels.

Source: *onlineinterviewquestions.com*

Q2: Explain the key components of AWS? ⭐

Answer:

- **Simple Storage Service (S3)**: S3 is most widely used AWS storage web service.
- **Simple E-mail Service (SES)**: SES is a hosted transactional email service and allows one to fluently send deliverable emails using a RESTFUL API call or through a regular SMTP.
- **Identity and Access Management (IAM)**: IAM provides improved identity and security management for AWS account.
- **Elastic Compute Cloud (EC2)**: EC2 is an AWS ecosystem central piece. It is responsible for providing on-demand and flexible computing resources with a "pay as you go" pricing model.
- **Elastic Block Store (EBS)**: EBS offers continuous storage solution that can be seen in instances as a regular hard drive.
- **CloudWatch**: CloudWatch allows the controller to outlook and gather key metrics and also set a series of alarms to be notified if there is any trouble.

Source: *whizlabs.com*

Q3: What is buckets in AWS? ⭐

Answer: An Amazon S3 bucket is a public cloud storage resource available in Amazon Web Services' (AWS) Simple Storage Service (S3), an object storage offering. Amazon S3 buckets, which are similar to file folders, store objects, which consist of data and its descriptive metadata.

By default, you can create up to 100 buckets in each of your AWS accounts. If you need more buckets, you can increase your bucket limit by submitting a service limit increase.

Source: *whizlabs.com*

Q4: What is AWS Cloudfront? ⭐⭐

Answer: Amazon **CloudFront** is a content delivery network (CDN) offered by Amazon Web Services. Content delivery networks provide a globally-distributed network of proxy servers which cache content, such as web videos or other bulky media, more locally to consumers, thus improving access speed for downloading the content.

Source: *en.wikipedia.org*

Q5: What do you mean by AMI? What does it include? ⭐⭐

Answer: AMI stands for the term **Amazon Machine Image**. It’s an AWS template which provides the information (an application server, and operating system, and applications) required to perform the launch of an instance. This AMI is the copy of the AMI that is running in the cloud as a virtual server. You can launch instances from as many different AMIs as you need. AMI consists of the followings:

- A root volume template for an existing instance
- Launch permissions to determine which AWS accounts will get the AMI in order to launch the instances
- Mapping for block device to calculate the total volume that will be attached to the instance at the time of launch

Source: *whizlabs.com*

Q6: How can I download a file from EC2? ⭐⭐

Answer: Use scp:

```
scp -i ec2key.pem username@ec2ip:/path/to/file .
```

Source: *stackoverflow.com*

Q7: Is it possible to clone a EC2 instance data? ⭐⭐

Answer: You can make an AMI of an existing instance, and then launch other instances using that AMI.

Source: *stackoverflow.com*

Q8: What is AWS Data Pipeline? ⭐⭐

Answer: AWS Data Pipeline is a web service that you can use to automate the movement and transformation of data. With AWS Data Pipeline, you can define data-driven workflows, so that tasks can be dependent on the successful completion of previous tasks.

Source: *docs.aws.amazon.com*

Q9: Explain the features of Amazon EC2 services ⭐⭐

Answer: Amazon EC2 services have following features:

- Virtual Computing Environments
- Proffers Persistent storage volumes
- Firewall validating you to specify the protocol
- Pre-configured templates
- Static IP address for dynamic Cloud Computing

Source: *whizlabs.com*

Q10: What is the connection between AMI and Instance? ⭐⭐

Answer: Many different types of *instances* can be launched from one *AMI*. The type of an instance generally regulates the hardware components of the host computer that is used for the instance. Each type of instance has distinct computing and memory efficacy.

Once an instance is launched, it casts as host and the user interaction with it is same as with any other computer but we have a completely controlled access to our instances. AWS developer interview questions may contain one or more AMI based questions, so prepare yourself for the AMI topic very well.

Source: *whizlabs.com*

Q11: Are S3 buckets region specific? ⭐⭐

Answer: Yes, buckets exist in a specific region and you need to specify that region when you create a bucket. Amazon S3 creates bucket in a region you specify. You can choose any AWS region that is geographically close to you to optimize latency, minimize costs, or address regulatory requirements.

Source: *stackoverflow.com*

Q12: What is AWS Direct Connect? ⭐⭐

Answer: AWS Direct Connect bypasses the public Internet and establishes a secure, dedicated connection from your infrastructure into AWS. With established connectivity via AWS Direct Connect, you can access your Amazon VPC and all AWS services.

Source: *coresite.com*

Q13: What is AWS EBS? ⭐⭐

Answer: Amazon Elastic Block Store (Amazon EBS) provides persistent block storage volumes for use with Amazon EC2 instances in the AWS Cloud. Each Amazon EBS volume is automatically replicated within its Availability Zone to protect you from component failure, offering high availability and durability.

Source: *aws.amazon.com*

Q14: What is AWS Lambda? ⭐⭐

Answer: AWS Lambda is a serverless compute service that runs your code in response to events and automatically manages the underlying compute resources for you. You can use AWS Lambda to extend other AWS services with custom logic, or create your own back-end services that operate at AWS scale, performance, and security.

Source: *aws.amazon.com*

Q15: What is AWS DynamoDB? ⭐⭐

Answer: Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. With DynamoDB, you can create database tables that can store and retrieve any amount of data, and serve any level of request traffic.

Source: *docs.aws.amazon.com*

Q16: What is AWS EMR? ⭐⭐

Answer: Amazon Elastic MapReduce (EMR) is an Amazon Web Services (AWS) tool for big data processing and analysis. Amazon EMR offers the expandable low-configuration service as an easier alternative to running in-house cluster computing.

Amazon EMR is based on Apache Hadoop, a Java-based programming framework that supports the processing of large data sets in a distributed computing environment. MapReduce is a software framework that allows developers to write programs that process massive amounts of unstructured data in parallel across a distributed cluster of processors or stand-alone computers.

Source: *searchaws.techtarget.com*

Q17: Is data stored in S3 is always encrypted? ⭐⭐

Answer: By default data on S3 is not encrypted, but all you could enable server-side encryption in your object metadata when you upload your data to Amazon S3. As soon as your data reaches S3, it is encrypted and stored.

Source: *aws.amazon.com*

Q18: Can we attach single EBS to multiple EC2s same time? ⭐⭐

Answer: No. After you create a volume, you can attach it to any EC2 instance in the same Availability Zone. An EBS volume can be attached to **only one EC2 instance at a time**, but multiple volumes can be attached to a single instance.

Source: *docs.aws.amazon.com*

Q19: What is AWS API gateway? ⭐⭐

Answer: Amazon **API Gateway** is an AWS service that enables developers to create, publish, maintain, monitor, and secure APIs at any scale. You can create APIs that access AWS or other web services, as well as data stored in the AWS Cloud.

Source: *aws.amazon.com*

Q20: What is AWS Direct Connect? ⭐⭐

Answer: Using **AWS Direct Connect**, you can establish private connectivity between AWS and your datacenter, office, or colocation environment, which in many cases can reduce your network costs, increase bandwidth throughput, and provide a more consistent network experience than Internet-based connections.

Source: *aws.amazon.com*

Q21: What are the security best practices for Amazon EC2 instances? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q22: Can I automatically start and terminate my Amazon instance using Amazon API? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q23: Can we still continue working on EBS while creating snapshot of it? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q24: What is AWS Auto Scaling? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q25: What is AWS Auto Scaling group? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q26: What is the maximum size of a single S3 object? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q27: What is AWS bucket policy? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q28: Does AWS has the option for vertical "auto" scaling of EC2 instance? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q29: What is AWS WAF? What are the potential benefits of using WAF? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q30: How to get the instance id from within an EC2 instance? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q31: What is AWS Cloudwatch? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q32: What is the difference between Amazon EC2 and AWS Elastic Beanstalk? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q33: How many storage options are there for EC2 Instance? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q34: What is AWS Route 53? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q35: How would you implement vertical auto scaling of EC2 instance? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q36: What is Amazon Kinesis? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q37: How to find a region from within an EC2 instance? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q38: Where are EC2 snapshots stored? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q39: When should one use the following: Amazon EC2, Google App Engine, Microsoft Azure and Salesforce.com? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q40: When to use Amazon CloudFront and when S3? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q41: Our EC2 micro instance occasionally runs out of memory. Other than using a larger instance size, what else can be done? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q42: What is difference between Lightsail and EC2? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q43: What is the underlying hypervisor for EC2? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q44: How to safely upgrade an Amazon EC2 instance from t1.micro to large? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] Agile & Scrum Interview Questions

Q1: What is ASP.NET MVC? ⭐

Answer: ASP.NET MVC is a web application Framework. It is light weight and highly testable Framework. MVC separates application into three components—Model, View and Controller.

Source: *medium.com*

Q2: What is Scrum? ⭐

Answer: Scrum is one of the most popular frameworks for implementing *Agile*. Many people think scrum and agile are the same thing but they're not.

With scrum, the product is built in a series of fixed-length iterations called sprints that give teams a framework for shipping software on a regular cadence.

Source: *atlassian.com*

Q3: What is sprint in Scrum? ⭐

Answer: In the Scrum methodology a **sprint** is the basic unit of development. Scrum sprints correspond to Agile iterations.

Each sprint starts with

- a **planning meeting**, where the tasks for the sprint are identified and an estimated commitment for the **sprint goal** is made.

A Sprint ends with

- a **review or retrospective meeting** where the progress is reviewed and lessons for the next sprint are identified. During each sprint, the team creates finished portions of a product.

Source: *stackoverflow.com*

Q4: Name roles in Scrum ⭐

Answer: Three essential roles for scrum success are:

- **The Product Owner** are the champions for their product. They are focused on understanding business and market requirements, then prioritizing the work to be done by the engineering team accordingly.
- ** The Scrum Master** are the champion for scrum within their team. They coach the team, the product owner, and the business on the scrum process and look for ways to fine-tune their practice of it.
- **The Scrum Team** are the champions for sustainable development practices. Scrum teams are cross-functional, "the development team" includes testers, designers, and ops engineers in addition to developers.

Source: *atlassian.com*

Q5: What is User Stories? ⭐

Answer: User stories are features customers might want to see in their software. They are written on index cards to encourage face-to-face communication. Typically no more than a couple days work, they form the basis of our Agile plans.

Q6: What is an epic, user stories and task? ⭐

Answer: Epic: A customer described software feature that is itemized in the product backlog is known as epic. Epics are sub-divided into stories.

User Stories: From the client perspective user stories are prepared which defines project or business functions, and it is delivered in a particular sprint as expected.

Task: Further down user stories are broken down into different task

Source: *career.guru99.com*

Q7: Explain what is Refactoring? ⭐

Answer: To improve the performance, the existing code is modified; this is re-factoring. During re-factoring the code functionality remains same.

Source: *career.guru99.com*

Q8: What is an Agile iteration? ⭐

Answer: An Agile **iteration** is a short one to two week period where a team takes most important user stories, builds them completely and deliver as running-tested-software to the customer. Analysis, design, coding, testing happen during an iteration.

Q9: Name some types of meetings or ceremonies in Scrum ⭐⭐

Answer: Scrum calls for four ceremonies that bring structure to each sprint:

- **Sprint planning**: A team planning meeting that determines what to complete in the coming sprint.
- **Daily stand-up**: Also known as a daily scrum, a 15-minute mini-meeting for the software team to sync.
- **Sprint demo**: A sharing meeting where the team shows what they've shipped in that sprint.
- **Sprint retrospective**: A review of what did and didn't go well with actions to make the next sprint better.

Source: *atlassian.com*

Q10: If a timebox plan needs to be reprioritized who should re-prioritise it? ⭐⭐

Answer: If a timebox plan needs to be reprioritized it should include whole team, product owner, and developers.

Source: *career.guru99.com*

Q11: Mention the key difference between sprint backlog and product backlog? ⭐⭐

Answer:

- **Product backlog**: It contains a list of all desired features and is owned by the product owner

- **Sprint backlog**: It is a subset of the product backlog owned by development team and commits to deliver it in a sprint. It is created in Sprint Planning Meeting

Source: *career.guru99.com*

Q12: What is Agile? ⭐⭐

Answer: Agile is a time boxed, **iterative approach (framework) to software delivery** that builds software incrementally from the start of the project, instead of trying to deliver it all at once near the end.

It works by breaking projects down into little bits of user functionality called **user stories**, prioritizing them, and then continuously delivering them in short two week cycles called **iterations**.

Agile refers to any process that aligns with the concepts of the Agile Manifesto.

Source: *agilemanifesto.org*

Q13: Explain in Agile, burn-up and burn-down chart? ⭐⭐

Answer: To track the project progress burnup and burn down, charts are used

- Burnup Chart: It shows the progress of stories done over time
- Burndown Chart: It shows how much work was left to do overtime

Source: *career.guru99.com*

Q14: What is Sprint Planning? ⭐⭐

Answer: The work to be performed in the Sprint is planned at the **Sprint Planning**. This plan is created by the collaborative work of the entire Scrum Team.

Sprint Planning answers the following:

- What can be delivered in the Increment resulting from the upcoming Sprint?
- How will the work needed to deliver the Increment be achieved?

The Sprint Goal is an objective set for the Sprint that can be met through the implementation of Product Backlog.

Source: *scrum.org*

Q15: Explain difference between a Product and a Sprint Backlog ⭐⭐

Answer:

- The **Product Backlog** is an ordered list of everything that is known to be needed in the product. It is the single source of requirements for any changes to be made to the product.

- The **Sprint Backlog** is the set of Product Backlog items selected for the Sprint during the Sprint Planning, plus a plan for delivering the product Increment and realizing the Sprint Goal.

Source: *scrum.org*

Q16: What is story points/efforts/ scales? ⭐⭐

Answer: It is used to discuss the difficulty of the story without assigning actual hours. The most common scale used is a Fibonacci sequence (1, 2, 3, 5, 8,1 3,....100) although some teams use linear scale (1, 2, 3, 4....), Powers of 2 (1, 2, 4, 8......) and cloth size (XS, S ,M, L, XL)

Source: *career.guru99.com*

Q17: How is Agile different from other software delivery aproaches? ⭐⭐

Answer:

- Analysis, design, coding, and testing are continuous activities
- Development is iterative
- Planning is adaptive
- Roles blur
- Scope can vary
- Requirements can change
- Working software is the primary measure of success

Q18: Have you ever used Scrum Task Board? ⭐⭐

Answer: In Scrum the *task board* is a visual display of the progress of the Scrum team during a sprint. It presents a snapshot of the current sprint backlog allowing everyone to see which tasks remain to be started, which are in progress and which are done.

Consider the following layout of the task board:

- Stories
- To Do
- In Progress
- Testing
- Done

Source: *manifesto.co.uk*

Q19: Explain what does it mean by product roadmap? ⭐⭐

Answer: A **product roadmap** is referred for the holistic view of product features that create the product vision.

Source: *career.guru99.com*

Q20: Explain what is Velocity in Agile? ⭐⭐

Answer: Velocity is a metric that is calculated by addition of all efforts estimates related with user stories completed in an iteration. It figures out how much work Agile can complete in a sprint and how much time will it need to finish a project.

Source: *career.guru99.com*

Q21: Mention what should a burndown chart should highlight? ⭐⭐

Answer: The burn-down chart shows the remaining work to complete before the timebox (iteration) ends.

Source: *career.guru99.com*

Q22: What is test driven development? ⭐⭐

Answer: Test driven development (TDD) is also known as test-driven design. In this method, developer first writes an automated test case which describes new function or improvement and then creates small codes to pass that test, and later re-factors the new code to meet the acceptable standards.

Source: *career.guru99.com*

Q23: Explain what is Scrum ban? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q24: What does project velocity mean? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q25: Can you explain the purpose of a burndown chart? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q26: What is the Agile Manifesto? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q27: What does the Scrum Framework consist from? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q28: What are four Agile Manifesto values? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q29: Explain the difference between Extreme programming and Scrum? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q30: What are some methodologies used to implement Agile? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q31: Mention what are the challenges involved in Agile software development? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q32: What are the qualities of a good Agile tester should have? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q33: What is a Sprint Review? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q34: What is Acceptance Criteria? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q35: Mention what are the advantages of maintaining consistent iteration length throughout the project? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q36: Mention in detail what are the role's of Scrum Master? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q37: Mention what is the difference between Scrum and Agile? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q38: Mention what are the Agile quality strategies? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q39: Why Continuous Integration is important for Agile? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q40: What is a Sprint Retrospective? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q41: What are the Scrum values? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q42: What the Scrum theory is based on? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q43: When not to use Agile? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q44: In Agile mention what is the difference between the Incremental and Iterative development? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q45: Explain how you can measure the velocity of the sprint with varying team capacity? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q46: What is Scrum Increment? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q47: Explain main differences between Scrum and Agile? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q48: Name the 12 Agile Principles ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q49: What are the benefits of Burn Up chart? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q50: What is the Scrum's definition of "Done"? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q51: Provide some examples of burn-up chart ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q52: Explain what is Spike and Zero sprint in Agile? What is the purpose of it? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] Android Interview Questions

Q1: Mention the difference between RelativeLayout and LinearLayout? ⭐

Answer:

- **Linear Layout** — Arranges elements either vertically or horizontally. i.e. in a row or column.
- **Relative Layout** — Arranges elements relative to parent or other elements.

Source: *android.jlelse.eu*

Q2: What is the difference between Bitmap and Drawable in Android? ⭐

Answer:

- A **Bitmap** is a representation of a bitmap image (something like java.awt.Image).
- A **Drawable** is an abstraction of "something that can be drawn". It could be a Bitmap (wrapped up as a BitmapDrawable), but it could also be a solid color, a collection of other Drawable objects, or any number of other structures.

Source: *stackoverflow.com*

Q3: What is a difference between Spannable and String? ⭐

Answer: A **Spannable** allows to attach formatting information like bold, italic, ... to sub-sequences ("spans", thus the name) of the characters. It can be used whenever you want to represent "rich text".

Source: *stackoverflow.com*

Q4: What is an Activity? ⭐

Answer: An **activity** provides the window in which the app draws its UI. This window typically fills the screen, but may be smaller than the screen and float on top of other windows. Generally, one activity implements one screen in an app. For instance, one of an app's activities may implement a Preferences screen, while another activity implements a Select Photo screen.

Source: *github.com*

Q5: What is Armv7? ⭐⭐

Answer: There are 3 CPU architectures in Android:

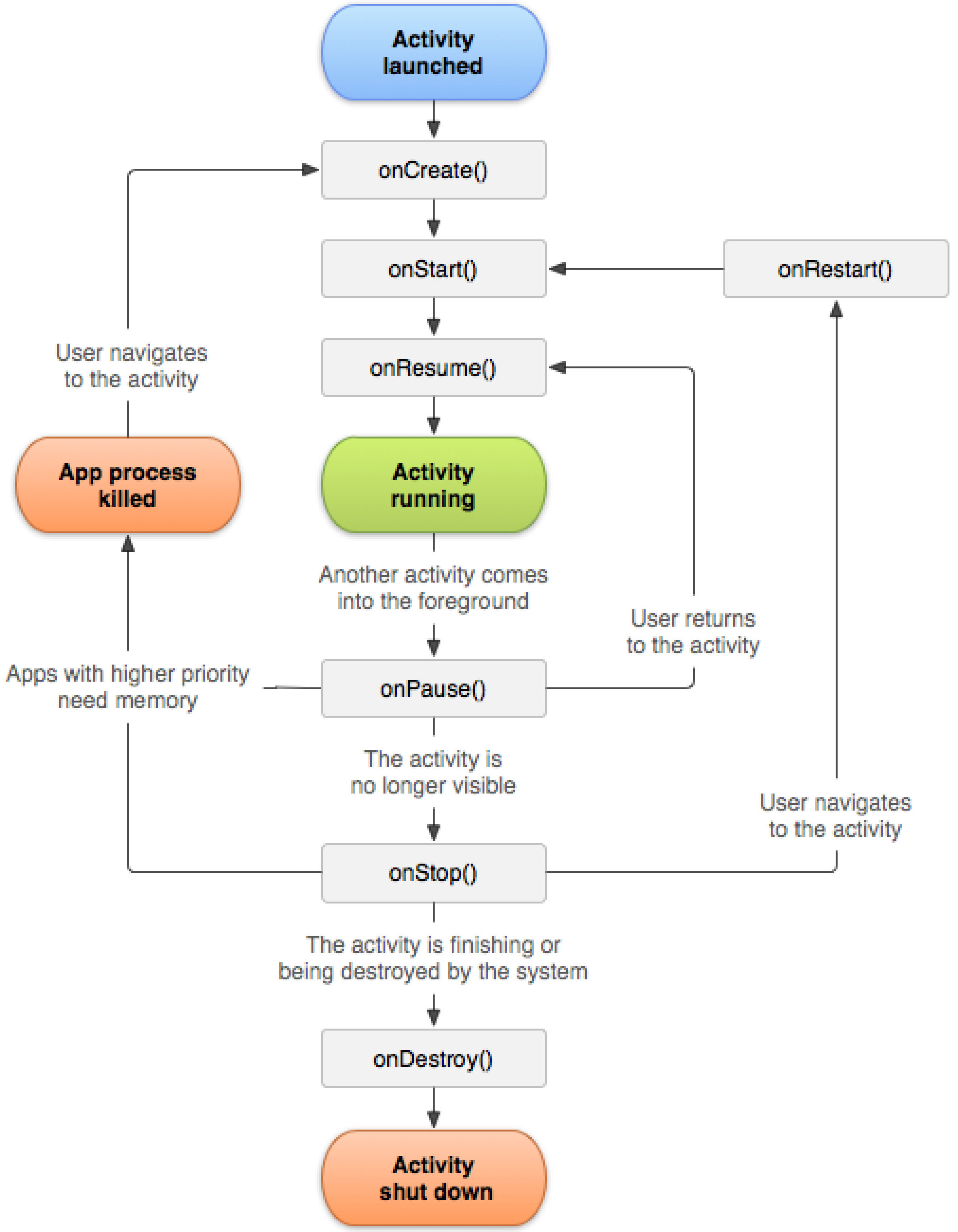
- ***ARMv7*** is the most common as it is optimised for battery consumption.
- ***ARM64*** is an evolved version of that that supports 64-bit processing for more powerful computing.
- ***ARMx86***, is the least used for these three, since it is not battery friendly. It is more powerful than the other two.

Source: *android.jlelse.eu*

Q6: Explain activity lifecycle ⭐⭐

Answer: As a user navigates through, out of, and back to your app, the Activity instances in your app transition through different states in their lifecycle.

To navigate transitions between stages of the activity lifecycle, the Activity class provides a core set of six callbacks: `onCreate()` , `onStart()` , `onResume()` , `onPause()` , `onStop()` , and `onDestroy()` . The system invokes each of these callbacks as an activity enters a new state.


Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Source: *developer.android.com*

Q7: How can I get the context in a fragment? ⭐⭐

Answer: You can use `getActivity()` , which returns the activity associated with a fragment. The activity is a context (since Activity extends `Context`).

You can also override the `onAttach` method of fragment:

```
public static class DummySectionFragment extends Fragment{
...
    @Override
    public void onAttach(Activity activity) {
        super.onAttach(activity);
```

```
            DBHelper = new DatabaseHelper(activity);
        }
    }
```

Source: *stackoverflow.com*

Q8: What is View Group? How are they different from Views? ⭐⭐

Answer: View: View objects are the basic building blocks of User Interface(UI) elements in Android. View is a simple rectangle box which responds to the user's actions. Examples are EditText, Button, CheckBox etc. View refers to the `android.view.View` class, which is the base class of all UI classes.

ViewGroup: ViewGroup is the invisible container. It holds View and ViewGroup. For example, LinearLayout is the ViewGroup that contains Button(View), and other Layouts also. ViewGroup is the base class for Layouts.

Source: *android.jlelse.eu*

Q9: Is it possible to implement the model–view–controller pattern in Java for Android? ⭐⭐

Answer: In Android you **don't have MVC**, but you have the following:

- You define your user interface in various XML files by resolution, hardware, etc.
- You define your resources in various XML files by locale, etc.
- You extend clases like ListActivity, TabActivity and make use of the XML file by inflaters.
- You can create as many classes as you wish for your business logic.
- A lot of Utils have been already written for you - DatabaseUtils, Html.

Source: *stackoverflow.com*

Q10: What's the difference between onCreate() and onStart()? ⭐⭐

Answer:

- The `onCreate()` method is called once during the Activity lifecycle, either when the application starts, or when the Activity has been destroyed and then recreated, for example during a configuration change.
- The `onStart()` method is called whenever the Activity becomes visible to the user, typically after `onCreate()` or `onRestart()` .

Source: *android.jlelse.eu*

Q11: Explain the build process in Android ⭐⭐

Answer:

1. First step involves compiling the resources folder (/res) using the aapt (android asset packaging tool) tool. These are compiled to a single class file called R.java. This is a class that just contains constants.
2. Second step involves the java source code being compiled to .class files by javac, and then the class files are converted to Dalvik bytecode by the "dx" tool, which is included in the sdk 'tools'. The output is classes.dex.
3. The final step involves the android apkbuilder which takes all the input and builds the apk (android packaging key) file.

Source: *android.jlelse.eu*

Q12: What is an Intent in Android? ⭐⭐

Answer: An Intent is basically a message that is passed between components (such as Activities, Services, Broadcast Receivers, and Content Providers).So, it is almost equivalent to parameters passed to API calls. The fundamental differences between API calls and invoking components via intents are:

- API calls are synchronous while intent-based invocations are asynchronous.
- API calls are compile-time binding while intent-based calls are run-time binding.

To listen for an broadcast intent (like the phone ringing, or an SMS is received), you implement a **broadcast receiver**, which will be passed the intent. To declare that you can handle another's app intent like "take picture", you declare an intent filter in your app's manifest file.

If you want to fire off an intent to do something, like pop up the dialer, you fire off an intent saying you will.

An Intent provides a facility for performing late runtime binding between the code in different applications.

Source: *stackoverflow.com*

Q13: What is the most appropriate way to store user settings in Android application? ⭐⭐

Answer: In general **SharedPreferences** are your best bet for storing preferences, so in general I'd recommend that approach for saving application and user settings.

The only area of concern here is what you're saving. Passwords are always a tricky thing to store, and I'd be particularly wary of storing them as clear text. The Android architecture is such that your application's SharedPreferences are sandboxed to prevent other applications from being able to access the values so there's some security there, but physical access to a phone could potentially allow access to the values.

Source: *stackoverflow.com*

Q14: In what situation should one use RecyclerView over ListView? ⭐⭐

Answer: RecyclerView was created as a **ListView** improvement, so yes, you can create an attached list with **ListView** control, but using **RecyclerView** is easier as it:

- Reuses cells while scrolling up/down - this is possible with implementing View Holder in the ListView adapter, but it was an optional thing, while in the RecycleView it's the default way of writing adapter.
- Decouples list from its container - so you can put list items easily at run time in the different containers (linearLayout, gridLayout) with setting LayoutManager.

To conclude, **RecyclerView** is a more flexible control for handling "list data" that follows patterns of delegation of concerns and leaves for itself only one task - recycling items.

Source: *stackoverflow.com*

Q15: Explain briefly all the Android application components ⭐⭐

Answer: App components are the essential building blocks of an Android app. Each component is an entry point through which the system or a user can enter your app.

There are four different types of app components:

- **Activities** - An activity is the entry point for interacting with the user. It represents a single screen with a user interface.
- **Services** - A service is a general-purpose entry point for keeping an app running in the background for all kinds of reasons. It is a component that runs in the background to perform long-running operations or to perform work for remote processes.
- **Broadcast receivers** - A broadcast receiver is a component that enables the system to deliver events to the app outside of a regular user flow, allowing the app to respond to system-wide broadcast announcements.

- **Content providers** - A content provider manages a shared set of app data that you can store in the file system, in a SQLite database, on the web, or on any other persistent storage location that your app can access.

Source: *developer.android.com*

Q16: What is 'Context' on Android? ⭐⭐

Answer: The documentation itself provides a rather straightforward explanation: The Context class is an “Interface to global information about an application environment".

We may assume a **Context** is a handle to the system; it provides services like resolving resources, obtaining access to databases and preferences, and so on. An Android app has activities. Context is like a handle to the environment your application is currently running in. The activity object inherits the Context object.

Source: *stackoverflow.com*

Q17: What is the Dalvik Virtual Machine? ⭐⭐

Answer: The **Dalvik Virtual Machine (DVM)** is an android virtual machine optimized for mobile devices. It optimizes the virtual machine for memory, battery life and performance.

The Dex compiler converts the class files into the `.dex` file that run on the Dalvik VM. Multiple class files are converted into one dex file.

Source: *www.javatpoint.com*

Q18: Tell about Constraint Layout ⭐⭐

Answer: ConstraintLayout allows you to create large and complex layouts with a flat view hierarchy (no nested view groups). It's similar to **RelativeLayout** in that all views are laid out according to relationships between sibling views and the parent layout, but it's more flexible than RelativeLayout and easier to use with Android Studio's Layout Editor.

Intention of ConstraintLayout is to optimize and flatten the view hierarchy of your layouts by applying some rules to each view to avoid nesting.

Source: *developer.android.com*

Q19: What is ADB and what is it used for? ⭐⭐

Answer: ADB is the acronym for Android Debug Bridge, which is part of the Android SDK (Software Development Kit). It uses a client-server-model (i.e. adbd, the ADB daemon, is running on the device and can be connected to), and in most cases is used via an USB connection. It is also possible to use it via WiFi (wireless adb).

There's nothing you need to install on your Android device, as the ADB daemon (adbd) is already integrated into the Android OS. It is usually accessed via a command line interface from the PC, where either the full Android SDK is installed (several 30 MB download archive currently), or a massively stripped-down version for "non-developers", sometimes referred to as "Mini ADB" or "ADB essentials" (for Linux, this is only the adb executable; for Windows it's adb.exe plus two or three .dll files).

Source: *developer.android.com*

Q20: What is Dalvik? ⭐⭐

Answer: Dalvik is a Just In Time (JIT) compiler. By the term JIT, we mean to say that whenever you run your app in your mobile device then that part of your code that is needed for execution of your app will only be compiled at that moment and rest of the code will be compiled in the future when needed. The JIT or Just In Time compiles only a part of your code and it has a smaller memory footprint and due to this, it uses very less physical space on your device.

Source: *blog.mindorks.com*

Q21: What types of Context do you know? ⭐⭐

Answer: The are mainly two types of context:

- **Application Context**: It is an instance that is the singleton and can be accessed in activity via `getApplicationContext()` . This context is tied to the lifecycle of an application. The application context can be used where you need a context whose lifecycle is separate from the current context or when you are passing a context beyond the scope of activity.

- **Activity Contex**t: This context is tied to the lifecycle of an activity. The activity context should be used when you are passing the context in the scope of an activity or you need the context whose lifecycle is attached to the current context.

Source: *blog.mindorks.com*

Q22: How do I pass data between Activities in Android application? ⭐⭐

Details: I have a scenario where, after logging in through a login page, there will be a sign-out button on each activity. Can you guide me on how to keep session id available to all activities?

Answer: The easiest way to do this would be to pass the session id to the signout activity in the **Intent** you're using to start the activity:

```
Intent intent = new Intent(getBaseContext(), SignoutActivity.class);
intent.putExtra("EXTRA_SESSION_ID", sessionId);
startActivity(intent);
```

Access that intent on next activity:

```
String sessionId = getIntent().getStringExtra("EXTRA_SESSION_ID");
```

Source: *stackoverflow.com*

Q23: How does the OutOfMemory happens? ⭐⭐

Answer: Out of memory error is very common error when you are developing for a application that deals with multiple images sets or large bitmaps or some Animation stuff. In Android, every application runs in a Linux Process. Each Linux Process has a Virtual Machine (Dalvik Virtual Machine) running inside it. There is a limit on the memory a process can demand and it is different for different devices and also differs for phones and tablets. When some process demands a higher memory than its limit it causes a error i.e **Out of memory** error.

There are number of reasons why we get a Out of memory errors. Some of those are:

1. You are doing some operation that continuously demands a lot of memory and at some point it goes beyond the max heap memory limit of a process.

2. You are leaking some memory i.e you didn't make the previous objects you allocated eligible for Garbage Collection (GC). This is called Memory leak.

3. You are dealing with large bitmaps and loading all of them at run time. You have to deal very carefully with large bitmaps by loading the size that you need not the whole bitmap at once and then do scaling.

Source: *blogs.innovationm.com*

Q24: What is a ContentProvider and what is it typically used for? ⭐⭐

Answer: A **content provider** manages access to a central repository of data. A provider is part of an Android application, which often provides its own UI for working with the data. However, content providers are primarily intended to be used by other applications, which access the provider using a provider client object.

Typically you work with content providers in one of two scenarios;

- you may want to implement code to access an existing content provider in another application, or
- you may want to create a new content provider in your application to share data with other applications.

Source: *developer.android.com*

Q25: What is an AsyncTask? ⭐⭐

Answer: `AsyncTask` is one of the easiest ways to implement parallelism in Android without having to deal with more complex methods like Threads. Though it offers a basic level of parallelism with the UI thread, it should not be used for longer operations (of, say, not more than 2 seconds).

AsyncTask has four methods

- `onPreExecute()`
- `doInBackground()`
- `onProgressUpdate()`
- `onPostExecute()`

where `doInBackground()` is the most important as it is where background computations are performed.

Source: *stackoverflow.com*

Q26: Why is it recommended to use only the default constructor to create a Fragment? ⭐⭐

Answer: In short, Fragments need to have a no-args constructor for the Android system to instantiate them. Your Fragment subclasses need a public empty constructor as this is what's being called by the framework.

It is used in the case when device has to restore the state of a fragment. No data will be passed and a default fragment will be created and then the state will be restored. Since the system has no way to know what you passed in your constructor or your `newInstance`, default constructor will be used and saved bundle should be passed via onCreate after the fragment is actually instantiated with the default constructor.

Source: *www.quora.com*

Q27: How to persist data in an Android app? ⭐⭐

Answer: There are basically four different ways to store data in an Android app:

1. Shared Preferences - to save primitive data in key-value pairs
2. Internal Storage - you need to store data to the device filesystem, but you do not want any other app (even the user) to read this data
3. External Storage - you might want the user to view the files and data saved by your app
4. SQLite database

Source: *www.androidauthority.com*

Q28: Explain Android notification system ⭐⭐

Answer: A **notification** is a message that Android displays outside your app's UI to provide the user with reminders, communication from other people, or other timely information from your app. Users can tap the notification to open your app or take an action directly from the notification.

Notifications appear to users in different locations and formats, such as an icon in the status bar, a more detailed entry in the notification drawer, as a badge on the app's icon, and on paired wearables automatically. Beginning with Android 5.0, notifications can appear on the lock screen.

Starting in Android 8.0 (API level 26), all notifications must be assigned to a channel or it will not appear. By categorizing notifications into channels, users can disable specific notification channels for your app (instead of disabling all your notifications), and users can control the visual and auditory options for each channel—all from the Android system settings.

Source: *developer.android.com*

Q29: What is Handler and what is it used for? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q30: Explain key differences between Service and IntentService ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q31: What is Explicit Intent? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q32: How would you support different screen sizes? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q33: What is Implicit Intent? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q34: What is Intent Filter? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q35: What does LayoutInflater in Android do? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q36: Android Log.v(), Log.d(), Log.i(), Log.w(), Log.e(). When to use each one? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q37: What are the differences between onCreate(), onCreateView(), and onActivityCreated() in fragments and what would they each be used for? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q38: What is the difference between onCreate() and onCreateView() lifecycle methods in Fragment? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q39: How to declare global variables in Android? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q40: What is the actual differences between a activity context and application context? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q41: What are the permission protection levels in Android? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q42: What is the difference between Activity and Context? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q43: How would you preserve Activity state during a screen rotation? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q44: When to use Android's ArrayMap instead of a HashMap? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q45: What are the differences between ArrayList and ArrayMap? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q46: What are dex files are used for? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q47: What is an Android PendingIntent? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q48: Explain how HashMap works ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q49: What is the difference between compileSdkVersion and targetSdkVersion? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q50: What are Android Annotations and what are they used for? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q51: How could you pass data between activities without Intent? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q52: What is the difference between invisible and gone for the View visibility status? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q53: What is ART? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q54: What is Android Data Binding? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q55: What is RenderScript and when should we (really) use it? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q56: Why fragments, and when to use fragments instead of activities? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q57: What are retained fragments? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q58: What is the difference between Service & Intent Service? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q59: Describe Different Types of Services in Android ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q60: What is the support library? Why was it introduced? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q61: What is DDMS and what can you do with it? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q62: What is the best way to update the screen periodically? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q63: What is a JobScheduler? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q64: What is the ViewHolder pattern? Why should we use it? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q65: What is the difference between ListView and RecyclerView? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q66: What is a LocalBroadcastManager? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q67: What is the difference between AsyncTask and Thread/Runnable? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q68: What is the difference between a Bundle and an Intent? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q69: What is the difference between Adapter and Loader in Android? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q70: When to use Android Loaders? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q71: Isn't android's Bundle functionally equivalent with a Map? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q72: What is Parcelable in Android? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q73: When to use Fragments vs Activities? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q74: How would you communicate between two Fragments? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q75: What is the difference between Handler vs AsyncTask vs Thread? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q76: How can two distinct Android apps interact? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q77: Explain String vs StringBuilder vs SpannedString vs SpannableString vs SpannableStringBuilder vs CharSequence ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q78: Is a Dalvik virtual machine instance created for each application? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q79: What is the Android NDK? How can one use it? Why should one use it? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q80: What is the StrictMode? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q81: When is it necessary, or better to use a SurfaceView instead of a View? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q82: Can you manually call the Garbage collector? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q83: What is the difference between Android timer and a Handler to do action every N seconds? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q84: What is Broadcast Receiver? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q85: How to avoid reverse engineering of an APK file? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q86: What are some differences between ART and Dalvik? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q87: What are Android Architecture Components? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q88: Why to consider FlatBuffers over JSON? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q89: What is the difference between ANR and crash in Android? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q90: How do you handle Bitmaps in Android as it takes too much memory? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q91: Explain how ArrayMap works ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q92: What is the difference between ArrayMap vs HashMap? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q93: Discuss Singletons vs. Application Context for app-global state ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q94: What is the difference between getContext() , getApplicationContext() , getBaseContext() , and "this"? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q95: What is Doze? What about App Standby? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q96: What is a ThreadPool? And is it more effective than using several separate Threads? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q97: How can I use AsyncTask in different Activities? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q98: What is AIDL? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q99: What are some best practices to avoid memory leaks on Android? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q100: When to use SparseArray vs HashMap? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q101: Provide some tips to reduce battery usage in an android application ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q102: What are some difference between Parcelable and Serializable? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q103: What are best practices for storing and protecting private API keys in applications? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q104: What is a Sticky Broadcast? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q105: What is the relationship between Looper, Handler and MessageQueue in Android? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q106: When would you use AIDL? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q107: Explain reasons why not to use getApplicationContext()? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q108: When to use AIDL vs Messenger Queue? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q109: What is the difference between Local, Normal, Ordered and Sticky broadcasts? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q110: Explain when would you call getApplicationContext() and why? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q111: What is Intent vs Sticky Intent vs Pending Intent? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q112: What happens if the user navigates away or closes the app while I still have a reference to the Activity the user just closed in my AsyncTask? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q113: What is the onTrimMemory method? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] Angular Interview Questions

Q1: What is Routing Guard in Angular? ⭐

Answer: Angular's route guards are interfaces which can tell the router whether or not it should allow navigation to a requested route. They make this decision by looking for a true or false return value from a class which implements the given guard interface.

Source: *medium.com*

Q2: What is a module, and what does it contain? ⭐

Answer: An Angular module is set of Angular basic building blocks like component, directives, services etc. An app can have more than one module.

A module can be created using `@NgModule` decorator.

```
@NgModule({
  imports:      [ BrowserModule ],
  declarations: [ AppComponent ],
  bootstrap:    [ AppComponent ]
})
export class AppModule { }
```

Source: *stackoverflow.com*

Q3: What are pipes? Give me an example. ⭐

Answer: A **pipe** takes in data as input and transforms it to a desired output. You can chain pipes together in potentially useful combinations. You can write your own custom pipes. Angular comes with a stock of pipes such as `DatePipe`, `UpperCasePipe`, `LowerCasePipe`, `CurrencyPipe`, and `PercentPipe`.

Consider:

```
<p>The hero's birthday is {{ birthday | date }}</p>
```

In this page, you'll use pipes to transform a component's birthday property into a human-friendly date.

Source: *angular.io*

Q4: What is the minimum definition of a component? ⭐⭐

Answer: The absolute minimal configuration for a `@Component` in Angular is a template. Both template properties are set to optional because you have to define either `template` or `templateUrl`.

When you don't define them, you will get an exception like this:

```
No template specified for component 'ComponentName'
```

A selector property is not required, as you can also use your components in a route.

Source: *stackoverflow.com*

Q5: What's the difference between an Angular component and module? ⭐⭐

Answer: *Components* control views (html). They also communicate with other components and services to bring functionality to your app.

Modules consist of one or more components. They do not control any html. Your modules declare which components can be used by components belonging to other modules, which classes will be injected by the dependency injector and which component gets bootstrapped. Modules allow you to manage your components to bring modularity to your app.

Source: *stackoverflow.com*

Q6: How would you run unit test? ⭐⭐

Answer: The Angular CLI downloads and install everything you need to test an Angular application with the Jasmine test framework.

The project you create with the CLI is immediately ready to test. Just run this one CLI command:

```
ng test
```

Source: *angular.io*

Q7: What is a service, and when will you use it? ⭐⭐

Answer: *Angular services* are singleton objects which get instantiated only once during the lifetime of an application. They contain methods that maintain data throughout the life of an application, i.e. data does not get refreshed and is available all the time. The main objective of a service is to organize and share business logic, models, or data and functions with different components of an Angular application.

The *separation of concerns* is the main reason why Angular services came into existence. An Angular service is a stateless object and provides some very useful functions.

Source: *dzone.com*

Q8: What is interpolation? ⭐⭐

Answer: Interpolation is a special syntax that Angular converts into property binding. It's a convenient alternative to property binding. It is represented by double curly braces({{}}). The text between the braces is often the name of a component property. Angular replaces that name with the string value of the corresponding component property. Let's take an example,

```
<h3>
  {{title}}
  <img src="{{url}}" style="height:30px">
</h3>
```

In the example above, Angular evaluates the title and url properties and fills in the blanks, first displaying a bold application title and then a URL.

Source: *github.com/sudheerj*

Q9: What is a bootstrapping module? ⭐⭐

Answer: Every application has at least one Angular module, the root module that you bootstrap to launch the application is called as bootstrapping module. It is commonly known as AppModule. The default structure of AppModule generated by AngularCLI would be as follows,

```
/* JavaScript imports */
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { HttpClientModule } from '@angular/common/http';

import { AppComponent } from './app.component';

/* the AppModule class with the @NgModule decorator */
@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    FormsModule,
    HttpClientModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

Source: *github.com/sudheerj*

Q10: What is the equivalent of ngShow and ngHide in Angular? ⭐⭐

Answer: Just bind to the `hidden` property

Source: *stackoverflow.com*

Q11: What are observables? ⭐⭐

Answer: Observables are declarative which provide support for passing messages between publishers and subscribers in your application. They are mainly used for event handling, asynchronous programming, and handling multiple values. In this case, you define a function for publishing values, but it is not executed until a consumer subscribes to it. The subscribed consumer then receives notifications until the function completes, or until they unsubscribe.

Source: *github.com/sudheerj*

Q12: What is an observable? ⭐⭐

Answer: An Observable is a unique Object similar to a Promise that can help manage async code. Observables are not part of the JavaScript language so we need to rely on a popular Observable library called RxJS. The observables are created using new keyword. Let see the simple example of observable,

```
import { Observable } from 'rxjs';

const observable = new Observable(observer => {
  setTimeout(() => {
    observer.next('Hello from a Observable!');
  }, 2000);
});`
```

Source: *github.com/sudheerj*

Q13: What is a component? Why would you use it? ⭐⭐

Answer: *Components* are the most basic building block of an UI in an Angular application. An Angular application is a tree of Angular components. Angular components are a subset of directives. Unlike directives, components always have a template and only one component can be instantiated per an element in a template.

A component must belong to an NgModule in order for it to be usable by another component or application. To specify that a component is a member of an NgModule, you should list it in the `declarations` field of that NgModule.

```
@Component({selector: 'greet', template: 'Hello {{name}}!'})
class Greet {
  name: string = 'World';
}
```

Source: *angular.io*

Q14: What is an observer? ⭐⭐

Answer: Observer is an interface for a consumer of push-based notifications delivered by an Observable. It has below structure,

```
interface Observer<T> {
  closed?: boolean;
  next: (value: T) => void;
  error: (err: any) => void;
  complete: () => void;
}
```

A handler that implements the Observer interface for receiving observable notifications will be passed as a parameter for observable as below,

```
myObservable.subscribe(myObserver);
```

Note: If you don't supply a handler for a notification type, the observer ignores notifications of that type.

Source: *github.com/sudheerj*

Q15: What is the purpose of base href tag? ⭐⭐

Answer: The routing application should add element to the index.html as the first child in the tag inorder to indicate how to compose navigation URLs. If app folder is the application root then you can set the href value as below

```
<base href="/">
```

Source: *github.com/sudheerj*

Q16: You have an HTML response I want to display. How do I do that? ⭐⭐

Answer: The correct syntax is the following:

```
<div [innerHTML]="theHtmlString"></div>
```

Working in `5.2.6`

Source: *medium.com*

Q17: What is the difference between Structural and Attribute directives in Angular? ⭐⭐

Answer:

- **Structural directives** are used to alter the DOM layout by removing and adding DOM elements. It is far better in changing the structure of the view. Examples of Structural directives are NgFor and Nglf.
- **Attribute Directives** These are being used as characteristics of elements. For example, a directive such as built-in NgStyle in the template Syntax guide is an attribute directive.

Source: *onlineinterviewquestions.com*

Q18: How can I select an element in a component template? ⭐⭐

Answer: You can get a handle to the DOM element via ElementRef by injecting it into your component's constructor:

```
constructor(myElement: ElementRef) { ... }
```

Source: *medium.com*

Q19: What is the equivalent of "ngShow" and "ngHide" in Angular? ⭐⭐

Answer: Just bind to the hidden property:

```
[hidden]="!myVar"
```

Source: *medium.com*

Q20: What is the difference between *ngIf vs [hidden]? ⭐⭐

Answer: `*ngIf` effectively removes its content from the DOM while `[hidden]` modifies the `display` property and only instructs the browser to not show the content but the DOM still contains it.

Source: *medium.com*

Q21: What are the differences between AngularJS (angular 1.x) and Angular (Angular 2.x and beyond)? ⭐⭐

Answer: Angular and AngularJS is basically a different framework with the same name.

Angular is more ready for the current state of web standards and the future state of the web (ES6\7, immutiablity, components, shadow DOM, service workers, mobile compatibilty, modules, typescript and so on and so on...)

Angular killed many main features in AngularJS like - controllers, $scope, directives (replaced with `@component` annotations), the module definition, and much more, even simple things like ng-repeat has not left the same as it was.

Also:

1. They added an angular `cli`.
2. Your angular code is written in ES6 Typescript and it compiles at runtime to Javascript in the browser.
3. You bind to your HTML similarly like how you would if in an Angular 1 directive. So variable like $scope and $rootScope have been deprecated.

Source: *stackoverflow.com*

Q22: What are some differences between Angular 2 and 4? ⭐⭐

Answer: Just to name a few:

- Improvements in AOT,
- allowing the "else" clause in ngIf,
- support for TypeScript 2.1
- breaking out the animations package

Source: *github.com/WebPredict*

Q23: What is the difference between "@Component" and "@Directive" in Angular? ⭐⭐

Answer:

- **Directives** add behaviour to an existing DOM element or an existing component instance.
- **A component**, rather than adding/modifying behaviour, actually creates its own view (hierarchy of DOM elements) with attached behaviour.

Write a component when you want to create a reusable set of DOM elements of UI with custom behaviour. Write a directive when you want to write reusable behaviour to supplement existing DOM elements.

Source: *medium.com*

Q24: How would you protect a component being activated through the router? ⭐⭐

Answer: The Angular router ships with a feature called guards. These provide us with ways to control the flow of our application. We can stop a user from visitng certain routes, stop a user from leaving routes, and more. The overall process for protecting Angular routes:

- Create a guard service: `ng g guard auth`
- Create `canActivate()` or `canActivateChild()` methods
- Use the guard when defining routes

```
// import the newly created AuthGuard
const routes: Routes = [
  {
    path: 'account',
    canActivate: [AuthGuard]
  }
];
```

Some other available guards:

- `CanActivate` : Check if a user has access
- `CanActivateChild` : Check if a user has access to any of the child routes
- `CanDeactivate` : Can a user leave a page? For example, they haven't finished editing a post
- `Resolve` : Grab data before the route is instantiated
- `CanLoad` : Check to see if we can load the routes assets

Source: *scotch.io*

Q25: What does this line do? ⭐⭐

Details:

```
@HostBinding('[class.valid]') isValid;
```

Answer: `@HostBinding` lets you set properties on the element or component that hosts the directive.

The code applies the css class `valid` to whatever is using this directive conditionally based on the value of isValid.

Source: *alligator.io*

Q26: What is router outlet? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q27: What is difference between "declarations", "providers" and "import" in NgModule? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q28: What's new in Angular 6 and why shall we upgrade to it? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q29: Why would you use a spy in a test? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q30: What is TestBed? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q31: What is Protractor? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q32: What is the point of calling "renderer.invokeElementMethod(rendererEl, methodName)"? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q33: How would you control size of an element on resize of the window in a component? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q34: What is AOT? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q35: What is Redux and how does it relate to an Angular app? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q36: What is the use of codelyzer? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q37: How to inject base href? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q38: How to bundle an Angular app for production? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q39: When would you use eager module loading? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q40: What are the Core Dependencies of Angular 7? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q41: Why Incremental DOM Has Low Memory Footprint? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q42: What are the ways to control AOT compilation? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q43: What is Angular Universal? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q44: Do I need a Routing Module always? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q45: What is the purpose of Wildcard route? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q46: What is activated route? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q47: What is router state? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q48: What is Reactive Programming and how to use one with Angular? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q49: Why should `ngOnInit` be used, if we already have a `constructor`? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q50: What are dynamic components? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q51: Explain how custom elements works internally? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q52: What are custom elements? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q53: What are the utility functions provided by RxJS? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q54: How do you perform error handling in observables? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q55: What is multicasting? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q56: What is the difference between promise and observable? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q57: Can you explain the difference between `Promise` and `Observable` in Angular? In what scenario can we use each case? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q58: What is subscribing? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q59: How do you perform Error handling for HttpClient? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q60: What is the difference between `@Component` and `@Directive` in Angular? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q61: Explain the difference between "Constructor" and "ngOnInit" ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q62: What is a parameterized pipe? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q63: How do you categorize data binding types? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q64: Explain the difference between `Promise` and `Observable` in Angular? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q65: What happens if you use script tag inside template? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q66: What is the option to choose between inline and external template file? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q67: What's new in Angular 8? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q68: Angular 8: What is Bazel? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q69: Angular 8: What is Angular Ivy? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q70: Angular 8: Explain Lazy Loading in Angular 8? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q71: Name some security best practices in Angular ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q72: How to set headers for every request in Angular? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q73: What is ngUpgrage? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q74: What are the mapping rules between Angular component and custom element? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q75: Why would you use renderer methods instead of using native element methods? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q76: Angular 9: What are some new features in Angular 9? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q77: Do I need to bootstrap custom elements? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q78: How do you create application to use scss? What changed for Angular 6? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q79: Name and explain some Angular Module Loading examples ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q80: What does "detectChanges" do in Angular jasmine tests? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q81: Why would you use lazy loading modules in Angular app? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q82: When does a lazy loaded module is loaded? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q83: What are the lifecycle hooks for components and directives? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q84: When should I store the "Subscription" instances and invoke `unsubscribe()` during the NgOnDestroy life cycle and when can I simply ignore them? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q85: What would be a good use for NgZone service? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q86: What is Ivy Renderer? Is it supported by Angular 7? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q87: What is incremental DOM? How is it different from virtual DOM? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q88: What is the difference between pure and impure pipe? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q89: What is Zone in Angular? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q90: How to detect a route change in Angular? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q91: What are the advantages with AOT? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q92: Could I use jQuery with Angular? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q93: How would you insert an embedded view from a prepared TemplateRef? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q94: Are there any pros/cons (especially performance-wise) in using local storage to replace cookie functionality? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q95: What does a just-in-time (JIT) compiler do (in general)? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q96: Angular 8: Why we should use Bazel for Angular builds? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q97: What is the need for SystemJS in Angular? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q98: What is Reactive programming and how does it relate to Angular? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q99: Explain the purpose of Service Workers in Angular ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q100: Why do we need compilation process? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q101: Angular 9: How Would You Compare View Engine vs Ivy? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q102: What is the Angular equivalent to an AngularJS "$watch"? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q103: Why Incremental DOM is Tree Shakable? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q104: What is the difference between BehaviorSubject vs Observable? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q105: What's new in Angular 7? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q106: Name some differences between SystemJS vs WebPack? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q107: What are observable creation functions? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q108: Is there no equivalent to `$scope.emit()` or `$scope.broadcast()` in Angular? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q109: What is the default compilation for Angular 5? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q110: Could you provide some particular examples of using ngZone? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q111: Why angular uses url segment? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q112: When to use query parameters versus matrix parameters? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q113: Angular 8: How does Ivy affect the (Re)build time? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q114: Angular 8: What are some changes in Location module? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q115: Do you know how you can run angularJS and angular side by side? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q116: How would you extract webpack config from angular cli project? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q117: Just-in-Time (JiT) vs Ahead-of-Time (AoT) compilation. Explain the difference. ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q118: Angular 9: What is Locality principle for Ivy? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q119: Angular 9: Explain improvements in Tree-Shaking ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q120: Why did the Google team go with incremental DOM instead of virtual DOM? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] AngularJS Interview Questions

Q1: Why to use AngularJS? ⭐

Answer: There are following reasons to choose AngularJS as a web development framework:

1. It is based on MVC pattern which helps you to organize your web apps or web application properly.
2. It extends HTML by attaching directives to your HTML markup with new attributes or tags and expressions in order to define very powerful templates.
3. It also allows you to create your own directives, making reusable components that fill your needs and abstract your DOM manipulation logic.
4. It supports two-way data binding i.e. connects your HTML (views) to your JavaScript objects (models) seamlessly. In this way any change in model will update the view and vice versa without any DOM manipulation or event handling.
5. It encapsulates the behavior of your application in controllers which are instantiated with the help of dependency injection.
6. It supports services that can be injected into your controllers to use some utility code to fullfil your need. For example, it provides $http service to communicate with REST service.
7. It supports dependency injection which helps you to test your angular app code very easily.
8. Also, AngularJS is mature community to help you. It has widely support over the internet.

Source: *github.com/krosti*

Q2: What is the difference between "ng-show"/"ng-hide" and "ng-if" directives? ⭐

Answer: `ng-show` / `ng-hide` will always insert the DOM element, but will display/hide it based on the condition. `ng-if` will not insert the DOM element until the condition is not fulfilled.

`ng-if` is better when we needed the DOM to be loaded conditionally, as it will help load page bit faster compared to `ng-show` / `ng-hide` .

We only need to keep in mind what the difference between these directives is, so deciding which one to use totally depends on the task requirements.

Source: *codementor.io*

Q3: Does AngularJS has dependency on jQuery? ⭐⭐

Answer: AngularJS has no dependency on jQuery library. But it can be used with jQuery library.

Source: _ github.com/krosti_

Q4: How do you hide an HTML element via a button click in AngularJS? ⭐⭐

Answer: This can be done by using the `ng-hide` directive in conjunction with a controller to hide an HTML element on button click.

```
<div ng-controller="MyCtrl">
    <button ng-click="hide()">Hide element</button>
    <p ng-hide="isHide">Hello World!</p>
</div>
```

```
function MyCtrl($scope) {
        $scope.isHide = false;
        $scope.hide = function () {
                $scope.isHide = true;
        };
}
```

Source: *codementor.io*

Q5: What is a singleton pattern and where we can find it in AngularJS? ⭐⭐

Answer: Is a great pattern that restricts the use of a class more than once. We can find singleton pattern in angular in dependency injection and in the services.

In a sense, if the you do 2 times `new Object()` without this pattern, the you will be alocating 2 pieces of memory for the same object. With singleton pattern, if the object exists, it'll be reused.

Source: *codementor.io*

Q6: What are the AngularJS features? ⭐⭐

Answer: The features of AngularJS are listed below:

1. Modules
2. Directives

3. Templates
4. Scope
5. Expressions
6. Data Binding
7. MVC (Model, View & Controller)
8. Validations
9. Filters
10. Services
11. Routing
12. Dependency Injection
13. Testing

Source: *github.com/krosti*

Q7: When dependent modules of a module are loaded? ⭐⭐

Answer: A module might have dependencies on other modules. The dependent modules are loaded by angular before the requiring module is loaded.

In other words the configuration blocks of the dependent modules execute before the configuration blocks of the requiring module. The same is true for the run blocks. Each module can only be loaded once, even if multiple other modules require it.

Source: *github.com/krosti*

Q8: What is Angular's prefixes $ and $$? ⭐⭐

Answer: To prevent accidental name collisions with your code, Angular prefixes names of public objects with $ and names of private objects with `$$`. So, do not use the `$` or `$$` prefix in your code.

Source: *github.com/krosti*

Q9: What are Filters in AngularJS? ⭐⭐

Answer: *Filters* are used to format data before displaying it to the user. They can be used in view templates, controllers, services and directives. There are some built-in filters provided by AngularJS like as `Currency`, `Date`, `Number`, `OrderBy`, `Lowercase`, `Uppercase` etc. You can also create your own filters.

Filter Syntax:

```
{{ expression | filter}}
```

Source: *github.com/krosti*

Q10: What are Directives in AngularJS? ⭐⭐

Answer: AngularJS directives are a combination of AngularJS template markups (HTML attributes or elements, or CSS classes) and supporting JavaScript code. The JavaScript directive code defines the template data and behaviors of the HTML elements.

AngularJS directives are used to extend the HTML vocabulary i.e. they decorate html elements with new behaviors and help to manipulate html elements attributes in interesting way.

There are some built-in directives provided by AngularJS like as ng-app, ng-controller, ng-repeat, ng-model etc.

Source: *github.com/krosti*

Q11: What are Directives? ⭐⭐

Answer: Directives are markers on a DOM element (such as an attribute, element name, comment or CSS class) that tell AngularJS's HTML compiler ($compile) to attach a specified behavior to that DOM element (e.g. via event listeners), or even to transform the DOM element and its children.

Angular comes with a set of these directives built-in, like `ngBind` , `ngModel` , and `ngClass` . Much like you create controllers and services, you can create your own directives for Angular to use. When Angular bootstraps your application, the HTML compiler traverses the DOM matching directives against the DOM elements.

Source: *codementor.io*

Q12: Explain what is a "$scope" in AngularJS ⭐⭐

Answer: Scope is an object that refers to the application model. It is an execution context for expressions. Scopes are arranged in hierarchical structure which mimic the DOM structure of the application. Scopes can watch expressions and propagate events. Scopes are objects that refer to the model. They act as glue between controller and view.

Source: *codementor.io*

Q13: What directive would you use to hide elements from the HTML DOM by removing them from that DOM not changing their styling? ⭐⭐

Answer: The `ngIf` Directive, when applied to an element, will remove that element from the DOM if it's condition is false.

Source: *codementor.io*

Q14: What is the difference between one-way binding and two-way binding? ⭐⭐

Answer:

- One way binding implies that the scope variable in the html will be set to the first value its model is bound to (i.e. assigned to)
- Two way binding implies that the scope variable will change it's value everytime its model is assigned to a different value

Source: *codementor.io*

Q15: What is auto bootstrap process in AngularJS? ⭐⭐

Answer: Angular initializes automatically upon `DOMContentLoaded` event or when the angular.js script is downloaded to the browser and the `document.readyState` is set to `complete`. At this point AngularJS looks for the `ng-app` directive which is the root of angular app compilation and tells about AngularJS part within DOM. When the `ng-app` directive is found then Angular will:

1. Load the module associated with the directive.
2. Create the application injector.
3. Compile the DOM starting from the ng-app root element. This process is called auto-bootstrapping.

```
<html>
<body ng-app="myApp">
<div ng-controller="Ctrl"> Hello {{msg}}!
</div>
    <script src="lib/angular.js"></script>
    <script>
var app = angular.module('myApp', []); app.controller('Ctrl', function ($scop
              $scope.msg = 'World';
          });
    </script>
</body>
</html>
```

Source: *github.com/krosti*

Q16: How would you specify that a scope variable should have one-time binding only? ⭐⭐

Answer: By using “ `::` ” in front of it.

Source: *codementor.io*

Q17: What is scope in AngularJS? ⭐⭐

Answer: Scope is a JavaScript object that refers to the application model. It acts as a context for evaluating angular expressions. Basically, it acts as glue between controller and view.

Scopes are hierarchical in nature and follow the DOM structure of your AngularJS app. AngularJS has two scope objects: **$rootScope** and **$scope**.

Source: *github.com/krosti*

Q18: How do you disable a button depending on a checkbox’s state? ⭐⭐

Answer: We can use the `ng-disabled` directive and bind its condition to the checkbox’s state.

```
<body ng-app>
    <label>
        <input type="checkbox" ng-model="checked" />Disable Button
    </label>
    <button ng-disabled="checked">Select me</button>
</body>
```

Source: *codementor.io*

Q19: What is scope hierarchy? ⭐⭐

Answer: The **$scope** object used by views in AngularJS are organized into a hierarchy. There is a root scope, and the **$rootScope** can has one or more child scopes. Each controller has its own **$scope** (which is a child of the **$rootScope**), so whatever variables you create on $scope within controller, these variables are accessible by the view based on this controller.

For example, suppose you have two controllers: ParentController and ChildController as given below:

```
<html>
  <head>
    <script src="lib/angular.js"></script>
    <script>
      var app = angular.module('ScopeChain', []); app.controller("parentContr
        $scope.managerName = 'Shailendra Chauhan';
        $scope.$parent.companyName = 'Dot Net Tricks'; //attached to $rootSco
      });
      app.controller("childController", function ($scope, $controller) {
                 $scope.teamLeadName = 'Deepak Chauhan';
             });

    </script>
  </head>
  <body ng-app="ScopeChain">
    <div ng-controller="parentController ">
      <table style="border:2px solid #e37112">
        <caption>Parent Controller</caption>
        <tr>
          <td>Manager Name</td>
          <td>{{managerName}}</td>
        </tr>
        <tr>
          <td>Company Name</td>
          <td>{{companyName}}</td>
        </tr>
        <tr>
          <td>
            <table ng-controller="childController" style="border:2px solid #4
              <caption>Child Controller</caption>
              <tr>
                <td>Team Lead Name</td>
                <td>{{ teamLeadName }}</td>
              </tr>
              <tr>
                <td>Reporting To</td>
                <td>{{managerName}}</td>
              </tr>
              <tr>
                <td>Company Name</td>
                <td>{{companyName}}</td>
              </tr>
            </table>
          </td>
        </tr>
      </table>
    </div>
  </body>
</html>
```

Source: *github.com/krosti*

Q20: How do you share data between controllers? ⭐⭐

Answer: Create an AngularJS service that will hold the data and inject it inside of the controllers.

Using a service is the cleanest, fastest and easiest way to test. However, there are couple of other ways to implement data sharing between controllers, like:

– Using `events`
– Using `$parent`, `nextSibling`, `controllerAs`, etc. to directly access the controllers
– Using the `$rootScope` to add the data on (not a good practice)

The methods above are all correct, but are not the most efficient and easy to test.

Source: *codementor.io*

Q21: What are the basic steps to unit test an AngularJS filter? ⭐⭐

Answer:

- Inject the module that contains the filter.
- Provide any mocks that the filter relies on.
- Get an instance of the filter using `$filter('yourFilterName')`.
- Assert your expectations.

Source: *codementor.io*

Q22: What are the basic steps to unit test an AngularJS filter? ⭐⭐

Answer:

1. Inject the module that contains the filter.
2. Provide any mocks that the filter relies on.
3. Get an instance of the filter using `$filter('yourFilterName')`.
4. Assert your expectations.

Source: *codementor.io*

Q23: What are the advantage of AngularJS? ⭐⭐

Answer: There are following advantages of AngularJS:

1. **Data Binding** - AngularJS provides a powerful data binding mechanism to bind data to HTML elements by using scope.
2. **Customize & Extensible** - AngularJS is customized and extensible as per you requirement. You can create your own custom components like directives, services etc.
3. **Code Reusability** - AngularJS allows you to write code which can be reused. For example custom directive which you can reuse.
4. **Support** – AngularJS is mature community to help you. It has widely support over the internet. Also, AngularJS is supported by Google which gives it an advantage.
5. **Compatibility** - AngularJS is based on JavaScript which makes it easier to integrate with any other JavaScript library and runnable on browsers like IE, Opera, FF, Safari, Chrome etc.
6. **Testing** - AngularJS is designed to be testable so that you can test your AngularJS app components as easy as possible. It has dependency injection at its core, which makes it easy to test.

Source: *github.com/krosti*

Q24: Explain what is services in AngularJS ⭐⭐

Answer: In AngularJS *services* are the singleton objects or functions that are used for carrying out specific tasks. It holds some business logic and these function can be called as controllers, directive, filters and so on.

Source: *github.com/krosti*

Q25: Explain what is directive and mention what are the different types of Directive? ⭐⭐

Answer: During compilation process when specific HTML constructs are encountered a behaviour or function is triggered, this function is referred as *directive*. It is executed when the compiler encounters it in the DOM.

Different types of directives are:

- Element directives
- Attribute directives
- CSS class directives
- Comment directives

Source: *guru99.com*

Q26: How would you validate a text input field for a twitter username, including the @ symbol? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q27: Explain what is the difference between link and compile in AngularJS? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q28: How would you react on model changes to trigger some further action? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q29: What is jQLite/jQuery Lite? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q30: What should be the maximum number of concurrent “watches”? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q31: What is a digest cycle in AngularJS? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q32: Where should we implement the DOM manipulation in AngularJS? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q33: Is it a good or bad practice to use AngularJS together with jQuery? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q34: If you were to migrate from Angular 1.4 to Angular 1.5, what is the main thing that would need refactoring? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q35: Explain what Angular JS routes does? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q36: Explain what is Angular Expression? Explain what is key difference between angular expressions and JavaScript expressions? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q37: What is restrict option in directive? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q38: How would you make an Angular service return a promise? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q39: What is the role of services in AngularJS and name any services made available by default? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q40: How do you reset a "$timeout", "$interval()", and disable a "$watch()"? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q41: What are different ways to invoke a directive? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q42: What is the role of ng-app, ng-init and ng-model directives? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q43: How to access jQLite? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q44: What is an interceptor? What are common uses of it? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q45: What is manual bootstrap process in AngularJS? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q46: What makes the `angular.copy()` method so powerful? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q47: Explain what is linking function and type of linking function? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q48: Explain what is injector? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q49: When creating a directive, it can be used in several different ways in the view. Which ways for using a directive do you know? How do you define the way your directive will be used? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q50: When should you use an attribute versus an element? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q51: Explain what is DI (Dependency Injection) and how an object or function can get a hold of its dependencies? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q52: Explain how `$scope.$apply()` works? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q53: How AngularJS is compiled? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q54: What is DDO (Directive Definition Object)? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q55: Can you define multiple restrict options on a directive? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q56: What is the difference between $scope and scope? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q57: How would you implement application-wide exception handling in your Angular app? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q58: What is $scope and $rootScope? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q59: How would you programatically change or adapt the template of a directive before it is executed and transformed? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q60: How AngularJS compilation is different from other JavaScript frameworks? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q61: How Directives are compiled? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q62: What are Compile, Pre and Post linking in AngularJS? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] Azure Interview Questions

Q1: What are the benefits of severless applications? ⭐

Answer:

- Avoid managing servers
- Flexible scaling by demand
- Pay for time and resources it takes to execute your code

Q2: What is Azure Cloud Service? ⭐

Answer: By creating a cloud service, you can deploy a multi-tier web application in Azure, defining multiple roles to distribute processing and allow flexible scaling of your application. A cloud service consists of one or more web roles and/or worker roles, each with its own application files and configuration. Azure Websites and Virtual Machines also enable web applications on Azure. The main advantage of cloud services is the ability to support more complex multi-tier architectures

Source: *mindmajix.com*

Q3: What is a web role? ⭐

Answer: A web role provides a dedicated Internet Information Services (IIS) web-server used for hosting front-end web applications.

Source: *mindmajix.com*

Q4: What is Azure Functions? ⭐

Answer: Azure Functions is a solution for easily running small pieces of code, or "functions," in the cloud. We can write just the code we need for the problem at hand, without worrying about a whole application or the infrastructure to run it and use language of our choice such as C#, F#, Node.js, Java, or PHP. Azure Functions lets us develop serverless applications on Microsoft Azure.

Q5: What is serverless computing? ⭐

Answer: Serverless computing is the abstraction of servers, infrastructure, and operating systems. When you build serverless apps you don't need to provision and manage any servers, so you can take your mind off infrastructure concerns. Serverless computing is driven by the reaction to events and triggers happening in near-real-time—in the cloud.

As a fully managed service, server management and capacity planning are invisible to the developer and billing is based just on resources consumed or the actual time your code is running.

Q6: Is Azure Table Storage Nosql? ⭐⭐

Answer: Azure Table storage is a service that stores structured NoSQL data in the cloud, providing a key/attribute store with a schemaless design.

Source: *docs.microsoft.com*

Q7: What is Kudu? ⭐⭐

Answer: Every Azure App Service web application includes a "hidden" service site called **Kudu**.

Kudu Console for example is a debugging service for Azure platform which allows you to explore your web app and surf the bugs present on it, like deployment logs, memory dump, and uploading files to your web app, and adding JSON endpoints to your web apps, etc.

Q8: What is a role instance? ⭐⭐

Answer: A role instance is a virtual machine on which the application code and role configuration run. A role can have multiple instances, defined in the service configuration file.

Source: *mindmajix.com*

Q9: What is a guest operating system? ⭐⭐

Answer: The guest operating system for a cloud service is the operating system installed on the role instances (virtual machines) on which your application code runs.

Source: *mindmajix.com*

Q10: What is Azure Blob Storage? ⭐⭐

Answer: *Azure Blob storage* is Microsoft's object storage solution for the cloud. Blob storage is optimized for storing massive amounts of unstructured data, such as text or binary data. Azure Storage offers three types of blobs:

- **Block blobs** store text and binary data, up to about 4.7 TB. Block blobs are made up of blocks of data that can be managed individually.
- **Append blobs** are made up of blocks like block blobs, but are optimized for append operations. Append blobs are ideal for scenarios such as logging data from virtual machines.
- **Page blobs** store random access files up to 8 TB in size. Page blobs store the VHD files that back VMs.

Source: *docs.microsoft.com*

Q11: How to include external dll into Azure Function? ⭐⭐

Answer:

- Add the assembly to the BIN directory using KUDU
- Include the assembly and code the Azure Function to use it
- Add the using declaration so that the methods within the DLL can be accessed.

```
#r "D:\home\site\wwwroot\GreetingsAssemblyReference\bin\benjamin.dll"

using benjamin;
```

Q12: What is an Azure subscription? ⭐⭐

Answer: A Windows **Azure subscription** grants you access to Windows Azure services and to the Windows Azure Platform Management Portal. A Windows Azure subscription has two aspects:

- The Windows Azure account, through which resource usage is reported
- Services are billed.

Source: *blogs.msdn.microsoft.com*

Q13: What is Azure ARM? ⭐⭐

Answer: The **Azure Resource Manager (ARM)** is the service used to provision resources in your Azure subscription. It was first announced at Build 2014 when the new Azure portal (portal.azure.com) was announced and provides a new set of API's that are used to provision resources. The ARM is:

- Template-driven – Using templates to deploy all resources.
- Declarative – You declare the resources you want to have instead of imperative where you need to make rules.
- Idempotent – You can deploy the template over and over again without affecting the current state of resources.
- Multi-service – All services can be deployed using Azure Resource Manager, Website, Storage, VMs etc.
- Multi region - You can choose in which region you would like to deploy the resources.
- Extensible – Azure Resource Manager is extensible with more resource providers and thus resources.

Source: *azurestack.blog*

Q14: Explain the Azure ARM Templates ⭐⭐

Answer: An Azure Resource Template is a JSON file used to deploy resources with Azure Resource Manager. It defines:

- Parameters
- Variables
- Resources - the actual resources that you are going to deploy or update
- Outputs

Source: *onlinetech.com*

Q15: What is a cloud service role? ⭐⭐

Answer: A cloud service role is comprised of application files and a configuration. A cloud service can have two types of roles:

- web role
- worker role

Source: *mindmajix.com*

Q16: What is Azure Redis Cache? ⭐⭐

Answer: *Redis* is an open source (BSD licensed), in-memory data structure store, used as a database, cache and message broker. **Azure Redis Cache** is based on the popular open-source Redis cache. It gives you access to a secure, dedicated Redis cache, managed by Microsoft, and accessible from any application within Azure. It supports data structures such as strings, hashes, lists, sets, sorted sets with range queries, bitmaps, hyperloglogs and geospatial indexes with radius queries.

Source: *quora.com*

Q17: What is Azure Service Fabric? ⭐⭐

Answer: **Azure Service Fabric** is a distributed systems platform that makes it easy to package, deploy, and manage scalable and reliable micro-services. Service Fabric also addresses the significant challenges in developing and managing cloud applications. Developers and administrators can avoid complex infrastructure problems and focus on implementing mission-critical, demanding workloads that are scalable, reliable, and manageable. Service Fabric represents the next-generation middleware platform for building and managing these enterprise-class, tier-1, cloud-scale applications.

Source: *quora.com*

Q18: How can I use applications with Azure AD that I’m using on-premises? ⭐⭐

Answer: *Azure AD* gives you an easy and secure way to connect to the web applications you choose. You can access these applications in the same way you access your SaaS apps in Azure AD, no need for a VPN to change your network infrastructure.

Source: *quora.com*

Q19: What is Azure Resource Group? ⭐⭐

Answer: Resource groups (RG) in Azure is an approach to group a collection of assets in logical groups for easy or even automatic provisioning, monitoring, and access control, and for more effective management of their costs. The underlying technology that powers resource groups is the Azure Resource Manager (ARM).

Source: *onlinetech.com*

Q20: What Is Azure Key Vault? ⭐⭐

Answer: Key Vault help you safeguard cryptographic keys and other secrets used by your applications whenever they are On-Premise or in the cloud. More and more services on Azure are now integrating Azure Key Vault as their secret/key source for things like deployments, data or even disk encryption.

Source: *codeisahighway.com*

Q21: What is a Blob Container? ⭐⭐

Answer: A container organizes a set of blobs, similar to a folder in a file system. All blobs reside within a container. A storage account can contain an unlimited number of containers, and a container can store an unlimited number of blobs.

Source: *docs.microsoft.com*

Q22: How can you stop a VM using Power Shell? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q23: How can you retrieve the state of a particular VM? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q24: How can one create a VM in Azure CLI? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q25: What do you know about Azure WebJobs? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q26: What is Azure MFA? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q27: How can one create a Virtual Machine in Powershell? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q28: How much storage can I use with a virtual machine? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q29: Is it possible to add an existing VM to an availability set? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q30: What is a worker role? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q31: What is Azure VPN? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q32: What is the difference between Service Bus Queues and Storage Queues? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q33: What are the differences between Subscription Administrator and Directory Administrator? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q34: What is a VNet? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q35: What is key vault in Azure? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q36: What are stateful and stateless microservices for Service Fabric? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q37: Do scale sets work with Azure availability sets? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q38: What are Network Security Groups? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q39: What are Update Domains? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q40: What are Fault Domains? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q41: What is an Availability Set? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q42: What are virtual machine scale sets in Azure? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q43: What is deployment environments? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q44: What are Cloud Service Roles and why do we use them? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q45: What is Azure Table Storage? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q46: What is Azure Resource Manager and why we need to use one? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q47: What is Azure Search? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q48: What are Redis databases? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q49: Is it possible to create a Virtual Machine using Azure Resource Manager in a Virtual Network that was created using classic deployment? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q50: How to create a new storage account and container using Power Shell? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q51: What is the meaning of application partitions? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q52: What is the difference between “price,” “software price,” and “total price” in the cost structure for Virtual Machine offers in the Azure Marketplace? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q53: What is Azure VNET? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q54: How to create a Network Security Group and a Network Security Group Rule? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q55: How are Azure Marketplace subscriptions priced? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q56: What are special Azure Regions? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q57: Explain Azure NSG ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q58: What VPN types are supported by Azure? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] Behavioral Interview Questions

Q1: How do you spend your time outside work? ⭐

Answer:

Q2: Why are you interested in this opportunity? ⭐

Answer:

Q3: Why do you want to work for X company? ⭐

Answer: The interviewer is looking for similar things whether asking about company or position. The hiring manager wants to:

- Learn about your career goals and how this position fits into your plan
- Make sure that you are sincerely interested in the job and will be motivated to perform if hired
- Find out what you know about the company, industry, position (and if you took the time to research)
- Understand your priorities and preferences — which aspects of the company and/or job are appealing to you and why?

Source: *github.com/yangshun/tech-interview-handbook*

Q4: Why do you want to leave your current/last company? ⭐

Answer: Here are some things your interviewer is likely looking for:

- Did you leave for a good reason?
- Did you leave voluntarily?
- Did you leave on good terms?
- What are your work values?

Source: *biginterview.com*

Q5: A hammer and a nail cost $1.10 together, and the hammer costs one dollar more than the nail. How much does the nail cost? ⭐

Answer:

Source: *startups.co*

Q6: What are you looking for in your next role? ⭐

Answer:

Source: *github.com/yangshun/tech-interview-handbook*

Q7: How large was the last team that you worked on? ⭐

Answer:

Q8: Do you plan to advance your education while working here? ⭐

Answer:

Q9: What are your hobbies? ⭐

Answer:

Q10: Tell me the story of how you became who you are today and what made you apply to X. ⭐

Answer:

Q11: Why do you think you’re a good fit for this company? ⭐⭐

Answer:

Q12: What’s are your favorite five apps? ⭐⭐

Answer:

Q13: Why do you want to come work at a startup, as opposed to an established company? ⭐⭐

Answer:

Q14: Share one of your trips with us. ⭐⭐

Answer:

Q15: What project are you currently working on? ⭐⭐

Answer:

Q16: Do prefer to work at a single company for a long time or would you rather take a job that suits you at the time? ⭐⭐

Answer:

Q17: Can you give an example of a career goal that you set and how you went about meeting it? ⭐⭐

Answer:

Q18: How quickly can you learn to use a new technology? ⭐⭐

Answer:

Q19: What does "belong anywhere" mean to you? ⭐⭐

Answer:

Q20: Can you list any deal breakers that would deter you from working for an employer? ⭐⭐

Answer:

Q21: Are you more interested in a career that offers great compensation, flexibility or the chance to do something meaningful? ⭐⭐

Answer:

Q22: What are you excited about? ⭐⭐

Answer:

Q23: Why do people resist change? ⭐⭐

Answer:

Source: *github.com*

Q24: Can you tell me what part of your resume you are most proud of? ⭐⭐

Answer:

Q25: Tell me about your last project - what worked and what didn’t? ⭐⭐

Answer:

Q26: What would your previous boss say your biggest strength was? ⭐⭐

Answer:

Q27: How do you stay up to date with the latest technologies? ⭐⭐

Answer:

Q28: What is the hardest technical problem you have run into? ⭐⭐

Answer:

Q29: What have you built as your side project? ⭐⭐

Answer:

Q30: If someone has a different viewpoint to do a project like different programming language, how would handle this situation? ⭐⭐

Answer:

Q31: What are your personal goals? ⭐⭐

Answer:

Q32: What was the most fun thing you did recently? ⭐⭐

Answer:

Q33: How do you deal with difficult coworkers? Think about specific instances where you resolved conflicts. ⭐⭐

Answer:

Q34: State an experience about how you solved a technical problem. Be specific about the diagnosis and process. ⭐⭐

Answer:

Q35: What is your biggest strength and area of growth? ⭐⭐

Answer:

Q36: Have you worked in a distributed team before? What challenges did you face? ⭐⭐

Answer:

Q37: What's the most difficult part of being a member of a team for you? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q38: Tell me about a time your work responsibilities got a little overwhelming. What did you do? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q39: Where do you want to be in five years? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q40: Tell me about a time in which you had a conflict and needed to influence somebody else. ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q41: What can you actually do for us? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q42: How did you win over the difficult employees? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q43: What are your three strengths and three weaknesses? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q44: What does it mean to be a "Professional Developer"? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q45: Would you prefer working on Green Field or Brown Field projects? Why? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q46: Describe company X to your grandmother. ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q47: Tell me a time when you predicted something. ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q48: Tell me a situation where you would have done something differently from what you actually did. ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q49: In my professional experience have you worked on something without getting approval from your manager? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q50: Explain me your toughest project and the working architecture. ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q51: What makes good code good? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q52: If you were to open your own business in the future, what kind of business will you open and why? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q53: Would you rather be good at a lot of things or an expert at one thing? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q54: Tell me something you are learning right now. ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q55: Tell me about a challenge you faced recently in your role. How did you tackle it? What was the outcome? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q56: What does your best day of work look like? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q57: Tell me about a time when you had a conflict with a co-worker. ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q58: What is the best gift you have ever given or received? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q59: What is something that you had to push for in your previous projects? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q60: What’s the best advice you’ve recently received? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q61: Can you explain this gap in your resume? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q62: What is something that you don't want from your last internship/job? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q63: If you had an unlimited budget and you could go somewhere, where would you go? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q64: What large problems in the world would you solve today? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q65: What frustrates you? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q66: If you had an unlimited budget and you could buy one gift for one person, what would you buy and who would you buy it for? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q67: Tell me about a time you needed information from someone who wasn't responsive. What did you do? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q68: What is the most exceedingly bad misstep you've made at any point? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q69: What books have inspired you in you live? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q70: What is the biggest mistake that you made in your last position? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q71: What is something new that you can teach your interviewer in a few minutes? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q72: What are some of the new ideas you would implement in this position? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q73: Are you comfortable assuming responsibilities outside your job description? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q74: What is the most challenging aspect of your current project? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q75: Tell me something about yourself and why you'd be a good fit for the position. ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q76: What are some of the best and worst things about your current company? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q77: How do you respond to constructive criticism? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q78: Explain streaming and how you would implement it. ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q79: Apart from technical knowledge, what did you learn during your work at Y? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q80: Talk about a project you are most passionate about, or one where you did your best work. ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q81: What was the most difficult bug that you fixed in the past 6 months? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q82: Imagine it is your first day here at the company. What do you want to work on? What features would you improve on? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q83: How do you deal with a failed deadline? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q84: Do you prefer to work in a team or individually? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q85: How do you tackle challenges? Name a difficult challenge you faced while working on a project, how you overcame it, and what you learned. ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q86: What are the most interesting projects you have worked on and how might they be relevant to this company's environment? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q87: What is the craziest thing you’ve ever done? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q88: What is your superpower? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q89: What is something you had to persevere at for multiple months? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q90: Tell me about a time you had to give someone terrible news. ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q91: Imagine there's a perfect clone of yourself. Imagine that that clone is your boss. Would you like to work for him/her? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q92: Interview me ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q93: Why are Quora's answers better than Yahoo Answers' ones? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q94: Let's play a game: defend Cobol against modern languages, and try to find as many reasonable arguments as you can. ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q95: Where will you be in 10 years? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q96: You are my boss and I'm fired. Inform me. ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q97: I want to refactor a legacy system. You want to rewrite it from scratch. Argument. Then, switch our roles. ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q98: Your boss asks you to lie to the Company. What's your reaction? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q99: Tell me about a time you had a disagreement with your manager. ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q100: What would happen if you put a mirror in a scanner? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q101: As a software engineer you want both to innovate and to be predictable. How those 2 goals can coexist in the same strategy? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q102: Name a situation where you were impressed by a company's customer service. ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q103: If you were given $1 million dollars every year for the rest of your life, what would you do? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q104: What risks do you feel you should never take? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q105: What is something 90% of people disagree with you about? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q106: Is developing software an art, a craftsmanship or an engineering endeavour? Your opinion. ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q107: Tell me about a time you were uncomfortable and how you dealt with it. ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q108: In one word, describe yourself. ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q109: Who do you look to as a role model? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q110: What is the most constructive feedback you have received in your career? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q111: What is broken around you? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q112: If you could travel back in time, which advice would you give to your younger self? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] Big Data Interview Questions

Q1: What is the meaning of big data and how is it different? ⭐

Answer: Big data is the term to represent all kind of data generated on the internet. On the internet over hundreds of GB of data is generated only by online activity. Here, online activity implies web activity, blogs, text, video/audio files, images, email, social network activity, and so on. Big data can be referred to data created from all these activities. Data generated online is mostly in unstructured form. Big data will also include transactions data in the database, system log files, along with data generated from smart devices such as sensors, IoT, RFID tags, and so on in addition to online activities.

Big data needs specialized systems and software tools to process all unstructured data. In fact, according to some industry estimates almost 85% of data generated on the internet is unstructured. Usually, relational databases have a structured format and the database is centralized. Hence, with RDBMS processing can be quickly done using a query language such as SQL. On the other hand, big data is very large and is distributed across the internet and hence processing big data will need distributed systems and tools to extract information from them. Big data needs specialized tools such as Hadoop, Hive, or others along with high-performance hardware and networks to process them.

Source: *www.educba.com*

Q2: What are the characteristics of big data? ⭐

Answer: Big data has three main characteristics: Volume, Variety, and Velocity. Volume characteristic refers to the size of data. Estimates show that over 3 million GB of data is generated every day. Processing this volume of data is not possible in a normal personal computer or in a client-server network in an office environment with limited compute bandwidth and storage capacities. However, cloud services provide solutions to handle big data volumes and process them efficiently using distributed computing architectures.

Variety characteristic refers to the format of big data – structured or unstructured. Traditional RDBMS fits into the structured format. An example of unstructured data format is, a video file format, image files, plain text format, from web document or standard MS Word documents, all have unique formats, and so on. Also to note, RDBMS does not have the capacity to handle unstructured data formats. Further, all this unstructured data must be grouped and consolidated which creates the need for specialized tools and systems. In addition new, data is added each day, or each minute and data grows continuously. Hence big data is more synonymous with variety.

The velocity characteristic refers to the speed at which data is created and the efficiency required to process all the data. For example, Facebook is accessed by over 1.6 billion users in a month. Likewise, there are other social network sites, YouTube, Google services, etc. Such data streams must be processed using queries in real time and must be stored without data loss. Thus, velocity characteristic is important in big data processing. In addition, other characteristics include veracity and value. Veracity will determine the dependability and reliability of data and value is the value derived by organizations from big data processing.

Source: *www.educba.com*

Q3: Why is big data important for organizations? ⭐

Answer: This is the basic Big Data interview question asked in an interview. Big data is important because by processing big data, organizations can obtain insight information related to:

- Cost reduction
- Improvements in products or services
- To understand customer behaviour and markets
- Effective decision making
- To become more competitive

Source: *www.educba.com*

Q4: Name some tools or systems used in big data processing? ⭐

Answer: Big data processing and analysis can be done using:

- Hadoop
- Hive
- Pig
- Mahout

- Flume

Source: *www.educba.com*

Q5: How can big data support organizations? ⭐⭐

Answer: Big data has the potential to support organizations in many ways. Information extracted from big data can be used in:

- Better coordination with customers and stakeholders and to resolve problems
- Improve reporting and analysis for product or service improvements
- Customize products and services to selected markets
- Ensure better information sharing
- Support in management decisions
- Identify new opportunities, product ideas, and new markets
- Gather data from multiple sources and archive them for future reference
- Maintain databases, systems
- Determine performance metrics
- Understand interdependencies between business functions
- Evaluate organizational performance

Source: *www.educba.com*

Q6: Explain how big data can be used to increase business value? ⭐⭐

Answer: While understanding the need for analyzing big data, such analysis will help businesses to identify their position in markets, and help businesses to differentiate themselves from their competitors. For example, from the results of big data analysis, organizations can understand the need for customized products or can understand potential markets towards increasing revenue and value. Analyzing big data will involve grouping data from various sources to understand trends and information related to business. When big data analysis is done in a planned manner by gathering data from the right sources, organizations can easily generate business value and revenue by almost 5% to 20%. Some examples of such organizations are Amazon, Linkedin, WalMart, and many others.

Source: *www.educba.com*

Q7: What is big data solution implementation? ⭐⭐

Answer: Big data solutions are implemented at a small scale first, based on a concept as appropriate for the business. From the result, which is a prototype solution, the business solution is scaled further. This is the most popular Big Data interview questions asked in a Big Data interview Some of the best practices followed in the industry include:

- To have clear project objectives and to collaborate wherever necessary
- Gathering data from the right sources
- Ensure the results are not skewed because this can lead to wrong conclusions
- Be prepared to innovate by considering hybrid approaches in processing by including data from structured and unstructured types, include both internal and external data sources
- Understand the impact of big data on existing information flows in the organization

Source: *www.educba.com*

Q8: What are the steps involved in big data solutions? ⭐⭐

Answer: Big data solutions follow three standard steps in its implementation. They are:

Data ingestion: This step will define the approach to extract and consolidate data from multiple sources. For example, data sources can be social network feeds, CRM, RDBMS, etc. The data extracted from different sources is stored in a Hadoop distributed file system (HDFS).

Data storage: This is the second step, the extracted data is stored. This storage can be in HDFS or HBase (NoSQL database).

Process the data: This is the last step. The data stored must be processed. Processing is done using tools such as Spark, Pig, MapReduce, and others.

Source: *www.educba.com*

[⬆️] Blockchain Interview Questions

Q1: What is blockchain? ⭐

Answer: Blockchain is a secure distributed ledger (data structure or database) that maintains a continuously growing list of ordered records, called “blocks”, that are linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data.

By design, a blockchain is resistant to modification of the data. It is "an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way".

Once recorded, the data in any given block cannot be altered retroactively without alteration of all subsequent blocks, which requires consensus of the network majority.

Source: *en.wikipedia.org*

Q2: Explain the common structure of blockchains ⭐⭐

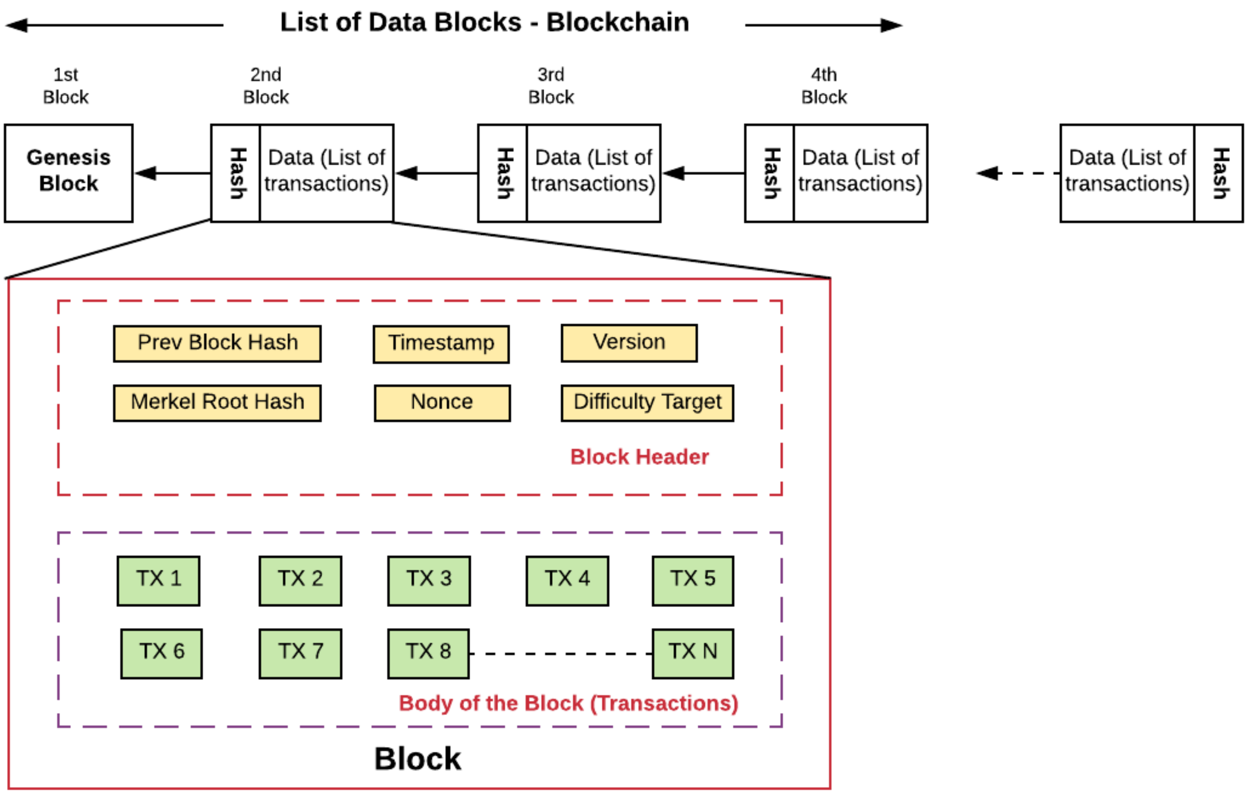
Answer: Blockchains are composed of three core parts:

- **Block**: A list of transactions recorded into a ledger over a given period. The size, period, and triggering event for blocks is different for every blockchain.
- **Chain**: A hash that links one block to another, mathematically “chaining” them together.
- **Network**: The network is composed of “full nodes.” Think of them as the computer running an algorithm that is securing the network. Each node contains a complete record of all the transactions that were ever recorded in that blockchain.

Source: *dummies.com*

Q3: What is the blockchain data structure? ⭐⭐

Answer: Basically **the blockchain data structure** is explained as a back-linked record of blocks of transactions, which is ordered. It can be saved as a file or in a plain database. Each block can be recognized by a hash, created utilizing the SHA256 cryptographic hash algorithm on the header of the block. Each block mentions a former block, also identified as the parent block, in the “previous block hash” field, in the block header.



Source: *cryptoticker.io*

Q4: What is the Genesis Block? ⭐⭐

Answer: The **first block in any blockchain **is termed the **genesis block**. If you start at any block and follow the chain backwards chronologically, you will arrive at the genesis block. The genesis block is statically encoded within the client software, that it cannot be changed. Every node can identify the genesis block's hash and structure, the fixed time of creation, and the single transactions within. Thus every node has a secure "root" from which is possible to build a trusted blockchain on.

Source: *linkedin.com*

Q5: What is blockchain transaction? ⭐⭐

Answer: Transactions are the things that give a blockchain purpose. They are the smallest building blocks of a blockchain system. Transactions generally consist of:

- a recipient address,
- a sender address,
- and a value.

This is not too different from a standard transaction that you would find on a credit card statement.

A transaction *changes the state* of the agreed-correct blockchain. A blockchain is a shared, decentralized, distributed state machine. This means that all nodes (users of the blockchain system) independently hold their own copy of the blockchain, and the current known "state" is calculated by processing each transaction in order as it appears in the blockchain.

Source: *pluralsight.com*

Q6: What is the purpose of a blockchain node? ⭐⭐

Answer: A blockchain exists out of blocks of data. These blocks of data are stored on nodes (compare it to small servers). **Nodes** can be any kind of device (mostly computers, laptops or even bigger servers). Nodes form the infrastructure of a blockchain.

All nodes on a blockchain are connected to each other and they constantly exchange the latest blockchain data with each other so all nodes stay up to date. They store, spread and preserve the blockchain data, so theoretically a blockchain exists on nodes.

A **full node** is basically a device (like a computer) that contains a full copy of the transaction history of the blockchain.

Source: *lisk.io*

Q7: Why does Blockchain need coins or tokens? ⭐⭐

Answer: *Tokens/Coins are used as a medium of exchange between the states*. They are digital assets built in to perform a specific function within a blockchain.

When someone does a transaction, there is a *change of state*, and coins are moved from one address to another address. Apart from that, transactions contain some additional data; this data can be mutated through the change of state. For this reason, blockchains need coins or tokens to incentivize the participants to join their networks.

Source: *mindmajix.com*

Q8: What is proof-of-work? ⭐⭐

Answer: A **proof of work** is a piece of data which is difficult (costly, time-consuming) to produce but easy for others to verify and which satisfies certain requirements. Producing a proof of work can be a random process with low probability so that a lot of trial and error is required on average before a valid proof of work is generated. Difficulty is a measure of how difficult it is to find a hash below a given target.

Source: *en.bitcoin.it*

Q9: What is deterministic behavior? ⭐⭐

Answer: If A + B = C, then no matter what the circumstances, A+B will always be equal to C. That is called deterministic behavior.

Hash functions are deterministic, meaning A's hash will always be H(A).

Source: *blockgeeks.com*

Q10: Explain what do nodes do? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q11: Why is the blockchain immutable? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q12: What are the core components of blockchain architecture? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q13: What is mining difficulty? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q14: What is a smart contract? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q15: How Are Blockchain And Distributed Ledger Different? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q16: What are some advantages of using Merke Trees? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q17: Explain why there is a fixed supply of bitcoins? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q18: What is a hashing function? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q19: What is RSA algorithm? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q20: What are the major elements of the blockchain ecosystem? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q21: What is DApp or Decentralised Application? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q22: What is Merkle Trees? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q23: How do verifiers check if a block is valid? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q24: Explain why a blockchain needs tokens to operate ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q25: What is a trapdoor function, and why is it needed in blockchain development? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q26: What is block data structure in blockchain? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q27: What's the difference between distributed hashtable technology and the bitcoin blockchain? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q28: What is a stealth address? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q29: What is nonce? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q30: What Is a Blockchain Consensus Algorithm? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q31: What Is a Proof of Stake? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q32: What is the difference between PoW and PoS? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q33: Explain what is target hash? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q34: What is off-chain transaction? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q35: Name some widespread platforms for developing blockchain applications ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q36: How is hard fork different from the soft fork in blockchain? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q37: What determines the mining difficulty? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q38: What is a 51% attack? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q39: What are miners really solving? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q40: Is it possible to brute force bitcoin address creation in order to steal money? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q41: What can an attacker with 51% of hash power do? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q42: Why is Git not considered a “block chain”? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] Bootstrap Interview Questions

Q1: Explain Bootstrap? ⭐

Answer: Bootstrap is a platform for web development that is based on front-end framework and creates exceptional responsive designs. It is fast, easy and has multiple templates designed using HTML, and CSS. These templates are used for forms, tables, buttons, typography, modals, tables, navigation, carousels and images. Bootstrap also has Javascript plugins, which are optional. Bootstrap is preferred for developing mobile web applications.

Source: *medium.com/@onlineinerview*

Q2: Explain what is Bootstrap? ⭐

Answer: Bootstrap is CSS/Javascript framework for building the rich web applications with minimal effort. This framework emphasis more on building mobile web applications.

Source: *medium.com/@alisonbenhar*

Q3: Explain the two codes that are used for code display in Bootstrap? ⭐⭐

Answer: There are two simple ways to display a code in Bootstrap:

- `<code>` tag: In case you wish to display an inline code
- `<pre>` tag: In case you have a code with several lines or even a block element

Source: *medium.com/@onlineinerview*

Q4: What are the types of layout available in Bootstrap? ⭐⭐

Answer: In Bootstrap there are two types of Layout available

- Fluid Layout: Fluid layout is used when you want to create a app that is 100% wide and use up all the width of the screen
- Fixed Layout: For a standard screen you will use fixed layout (940 px) option

Source: *medium.com/@alisonbenhar*

Q5: What will be the output of the following HTML code ⭐⭐

Answer: Consider:

```
<ul class="list-unstyled">
  <li>Item 1</li>
  <li>Item 2</li>
    <ul>
      <li>Nested item 2.1</li>
      <li>Nested item 2.2</li>
      <li>Nested item 2.3</li>
    </ul>
  <li>Item 3</li>
</ul>
```

What will be the output of the following HTML code?

Answer:

If we apply `.list-unstyled` to a list, it will remove the default list-style and left margin on the list items. But *only* for the immediate children. Main list items will be without any style, and nested list items will still have default unordered nested list-style.

Source: *toptal.com*

Q6: Explain why you prefer Bootstrap for website development? ⭐⭐

Answer: Bootstrap has features that are way better than other web development platforms. It provides an extensive browser support for almost every known browser such as Opera, Chrome, Firefox, Safari etc. With adequate knowledge of CSS and HTML, web development becomes easy on Bootstrap. Also, it supports mobile applications with the help of responsive design. It can adjust CSS as per the device, screen size etc. Instead of creating multiple files, it creates only a single file, which reduces any extra effort by the developer.

Source: *medium.com/@onlineinerview*

Q7: What are the key components of Bootstrap? ⭐⭐

Answer: In total, there are five key components of Bootstrap i.e. **CSS** (multiple CSS files), **Scaffolding (**essential for the basic system that consist of Grid system, background and link styles), **Layout Components:** (shares a list of all layouts), **JavaScript Plugins** (includes jQuery and JavaScript plugins) and **Customization ** (Allows customization of all components for a desired framework)

Source: *medium.com/@onlineinerview*

Q8: How many types of layout are available in Bootstrap? ⭐⭐

Answer: There are two major layouts for Bootstrap i.e. Fluid Layout and Fixed Layout. Fluid layout is necessary for creating an app that is 100 % wider and covers all the screen width. Fixed Layout is used only for a standard screen (940px). Both layouts can be used for creating a responsive design.

Source: *medium.com/@onlineinerview*

Q9: Why do we use Jumbotron in Bootstrap? ⭐⭐

Answer: *Jumbotron* has a very basic function in bootstrap i.e. highlighting a content. It could either be a slogan/uvp (unique value proposition) or probably a headline. It increases the heading size and gives a margin for content of the landing page. In order to implement Jumbotron in a Bootstrap use:

```
<div class=”jumbotron”>
```

Jumbotron can have any valid HTML along with other functions and classes.

Source: *medium.com/@onlineinerview*

Q10: What is Twitter Bootstrap? ⭐⭐

Answer: Bootstrap is a sleek, intuitive, and powerful mobile first front-end framework for faster and easier web development. It uses HTML, CSS and Javascript.

Source: *medium.com/@alisonbenhar*

Q11: Explain why to choose Bootstrap for building the websites? ⭐⭐

Answer: There are few reason why we choose Bootstrap for building websites

- Mobile Support: For mobile devices it provides full support in one single file rather than in separate file. It supports the responsive design including adjusting the CSS

based on the different types of device, size of the screen etc. It reduces extra effort for developers.
- Easy to learn: Writing application in bootstrap is easy if you know CSS and HTML
- Browser Support: It supports all the popular browsers like Firefox, Opera, Safari, IE etc.

Source: *medium.com/@alisonbenhar*

Q12: What global styles are applied as a part of Bootstrap's default typography? ⭐⭐

Answer: Bootstrap sets the global default font-size to `14px` , with a line-height of `1.428` . The default font is changed to `Helvetica` and `Arial` with `sans serif` fallback.

All these styles are applied to the `<body>` and all paragraphs, with the addition that `<p>` (paragraphs) receive a bottom margin of half their computed line-height, which is `10px` by default.

Source: *toptal.com*

Q13: What is the procedure to create Nav elements in Bootstrap? ⭐⭐

Answer: There are several styling navigation elements available on bootstrap and every style uses the same function i.e. class `.nav` . In order to create tabs or a tabular navigation, you can begin with a simple or rather basic unordered list using the function class `.nav` . To add the tabs the function class `.nav-tabs` can be used.

Source: *medium.com/@onlineinerview*

Q14: What are Glyphicons? ⭐⭐

Answer: Glyphicons are icon fonts which you can use in your web projects. Glyphicons Halflings are not free and require licensing, however their creator has made them available for Bootstrap projects free of cost.

To use the icons, simply use the following code just about anywhere in your code. Leave a space between the icon and text for proper padding.

```
<span class="glyphicon glyphicon-search"></span>
```

Source: *medium.com/@alisonbenhar*

Q15: When to use "lead" in Bootstrap? ⭐⭐

Answer: To add some emphasis to a paragraph, add class `.lead`. This will give you larger font size, lighter weight, and a taller line height.

Source: *medium.com/@alisonbenhar*

Q16: Explain the typography and links in Bootstrap. ⭐⭐

Answer: Bootstrap sets a basic global display (background), typography, and link styles:

- **Basic Global display** – sets *background-color: #fff;* on the
- **Typography** – uses the *@font-family-base*, *@font-size-base*, and *@line-height-base* attributes as the typographic base
- **Link styles** – sets the global link color via attribute *@link-color* and apply link underlines only on *:hover*.

Source: *medium.com/@alisonbenhar*

Q17: How do you make images responsive? ⭐⭐

Answer: Bootstrap 3 allows to make the images responsive by adding a class `.img-responsive` to the `<img>` tag. This class applies `max-width: 100%;` and `height: auto;` to the image so that it scales nicely to the parent element.

Source: *medium.com/@alisonbenhar*

Q18: What is missing for a tooltip to show properly? ⭐⭐

Answer: Consider:

```
<button type="button" class="btn btn-default" data-toggle="tooltip" data-plac
```

What is missing for a tooltip to show properly?

Answer

Bootstrap’s Tooltip plugin is not CSS-only, like other plugins are. For performance reasons, the Tooltip plugin is opt-in, and to use it you must initialize it using JavaScript with the following example code:

```
$(function () {
  $('[data-toggle="tooltip"]').tooltip();
});
```

Source: *medium.com/@alisonbenhar*

Q19: What do you mean by Bootstrap collapsing elements? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q20: What is a list group in Bootstrap and where does it finds its application? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q21: Explain Modal plugin in Bootstrap? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q22: What is a Bootstrap Container? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q23: Why do we use Bootstrap Carousel plugin? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q24: What is the role of media object in Bootstrap and how many types are available? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q25: What are the key components of Bootstrap? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q26: What is the step-wise procedure for creating basic or vertical forms? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q27: What do you mean by Bootstrap "well"? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q28: What is the difference between the following two lines of code? Explain your answer. ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q29: What will be the default Bootstrap look of the alert created with this following code? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q30: Explain what the following code does, and where they are useful ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q31: What is the role of pagination in bootstrap and what are their classifications? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q32: What is Normalize in Bootstrap? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q33: Consider the HTML code snippet below. What will the output be, and why? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q34: Explain column ordering in Bootstrap? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q35: What is screen reader in bootstrap documentation? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q36: How can you differentiate between Bootstrap and Foundation? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q37: How many different media queries are used by the Bootstrap grid system by default? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q38: What is the class sr-only used for? Is it important or can I remove it? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] C# Interview Questions

Q1: What is C#? ⭐

Answer: C# is the programming language for writing Microsoft .NET applications. C# provides the rapid application development found in Visual Basic with the power of C++. Its syntax is similar to C++ syntax and meets 100% of the requirements of OOPs like the following:

- Abstraction
- Encapsulation
- Polymorphism
- Inheritance

Source: *c-sharpcorner.com*

Q2: What is the difference between "continue" and "break" statements in C#? ⭐

Answer:

- using **break** statement, you can jump out of a loop
- using **continue** statement, you can jump over one iteration and then resume your loop execution

Source: *c-sharpcorner.com*

Q3: What are property Accessors? ⭐

Answer: The *get* and *set* portions or blocks of a property are called accessors. These are useful to restrict the accessibility of a property, the set accessor specifies that we can assign a value to a private field in a property and without the set accessor property it is like a read-only field. By the get accessor we can access the value of the private field, in other words it returns a single value. A Get accessor specifies that we can access the value of a field publically.

Source: *c-sharpcorner.com*

Q4: What is an Object? ⭐

Answer: According to MSDN, "*a class or struct definition is like a blueprint that specifies what the type can do. An object is basically a block of memory that has been allocated and configured according to the blueprint. A program may create many objects of the same class. Objects are also called instances, and they can be stored in either a named variable or in an array or collection. Client code is the code that uses these variables to call the methods and access the public properties of the object. In an object-oriented language such as C#, a typical program consists of multiple objects interacting dynamically*".

Objects helps us to access the member of a class or struct either they can be fields, methods or properties, by using the dot.

Source: *c-sharpcorner.com*

Q5: What is enum in C#? ⭐⭐

Answer: An **enum** is a value type with a set of related named constants often referred to as an enumerator list. The enum keyword is used to declare an enumeration. It is a primitive data type, which is user defined. An enum is used to create numeric constants in .NET framework. All the members of enum are of enum type. Their must be a numeric value for each enum type.

Some points about enum

- Enums are enumerated data type in C#.
- Enums are strongly typed constant. They are strongly typed, i.e. an enum of one type may not be implicitly assigned to an enum of another type even though the underlying value of their members are the same.
- Enumerations (enums) make your code much more readable and understandable.
- Enum values are fixed. Enum can be displayed as a string and processed as an integer.
- The default type is int, and the approved types are byte, sbyte, short, ushort, uint, long, and ulong.
- Every enum type automatically derives from System.Enum and thus we can use System.Enum methods on enums.
- Enums are value types and are created on the stack and not on the heap.

Source: *c-sharpcorner.com*

Q6: How is Exception Handling implemented in C#? ⭐⭐

Answer: Exception handling is done using four keywords in C#:

- **try** – Contains a block of code for which an exception will be checked.
- **catch** – It is a program that catches an exception with the help of exception handler.
- **finally** – It is a block of code written to execute regardless whether an exception is caught or not.
- **Throw** – Throws an exception when a problem occurs.

Source: *softwaretestinghelp.com*

Q7: Can "this" be used within a static method? ⭐⭐

Answer: We can't use *this* in static method because keyword *this* returns a reference to the current instance of the class containing it. Static methods (or any static member) do not belong to a particular instance. They exist without creating an instance of the class and call with the name of a class not by instance so we can't use this keyword in the body of static Methods, but in case of Extension Methods we can use it as the functions parameters.

Source: *c-sharpcorner.com*

Q8: Define Property in C#? ⭐⭐

Answer: Properties are members that provide a flexible mechanism to read, write or compute the values of private fields, in other words by the property we can access private fields. In other words we can say that a property is a return type function/method with one parameter or without a parameter. These are always public data members. It uses methods to access and assign values to private fields called accessors.

Source: *c-sharpcorner.com*

Q9: What is Boxing and Unboxing? ⭐⭐

Answer: Boxing and Unboxing both are used for type conversion but have some difference:

- **Boxing** - Boxing is the process of converting a value type data type to the object or to any interface data type which is implemented by this value type. When the CLR boxes a value means when CLR is converting a value type to Object Type, it wraps the value inside a System.Object and stores it on the heap area in application domain.

- **Unboxing** - Unboxing is also a process which is used to extract the value type from the object or any implemented interface type. Boxing may be done implicitly, but unboxing have to be explicit by code.

The concept of boxing and unboxing underlines the C# unified view of the type system in which a value of any type can be treated as an object.

Source: *c-sharpcorner.com*

Q10: What is the difference between string and StringBuilder in c#? ⭐⭐

Answer: String

- It's an immutable object that hold string value.
- Performance wise string is slow because its' create a new instance to override or change the previous value.
- String belongs to System namespace.

StringBuilder

- StringBuilder is a mutable object.
- Performance wise StringBuilder is very fast because it will use same instance of StringBuilder object to perform any operation like insert value in existing string.
- StringBuilder belongs to System.Text.Stringbuilder namespace.

Source: *c-sharpcorner.com*

Q11: What are partial classes? ⭐⭐

Answer: A **partial** class is only use to splits the definition of a class in two or more classes in a same source code file or more than one source files. You can create a class definition in multiple files but it will be compiled as one class at run time and also when you'll create an instance of this class so you can access all the methods from all source file with a same object. Partial classes can be create in the same namespace it's doesn't allowed to create a partial class in different namespace.

Source: *c-sharpcorner.com*

Q12: Filter out the first 3 even numbers from the list using LINQ ⭐⭐

Answer:

```
var evenNumbers = List
   .Where(x => x % 2 ==0)
   .Take(3)
```

Source: *medium.com/*

Q13: Why to use “finally” block in C#? ⭐⭐

Answer: Finally block will be executed irrespective of exception. So while executing the code in try block when exception is occurred, control is returned to catch block and at last `finally` block will be executed. So closing connection to database / releasing the file handlers can be kept in `finally` block.

Source: *a4academics.com*

Q14: What are nullable types in C#? ⭐⭐

Answer: C# provides a special data types, the **nullable types**, to which you can assign normal range of values as well as null values.

For example, you can store any value from -2,147,483,648 to 2,147,483,647 or null in a `Nullable<Int32>` variable. Similarly, you can assign true, false, or null in a `Nullable<bool>` variable.

Source: *tutorialspoint.com*

Q15: What are generics in C#? ⭐⭐

Answer: Generics allow you to delay the specification of the data type of programming elements in a class or a method, until it is actually used in the program. In other words, generics allow you to write a class or method that can work with any data type.

Source: *c-sharpcorner.com*

Q16: What is Managed or Unmanaged Code? ⭐⭐

Answer:

- **Managed Code** - The code, which is developed in .NET framework is known as managed code. This code is directly executed by CLR with the help of managed code execution. Any language that is written in .NET Framework is managed code.
- **Unmanaged Code** - The code, which is developed outside .NET framework is known as unmanaged code. Applications that do not run under the control of the CLR are said to be unmanaged, and certain languages such as C++ can be used to write such applications, which, for example, access low - level functions of the operating system. Background compatibility with the code of VB, ASP and COM are examples of unmanaged code.

Source: *c-sharpcorner.com*

Q17: What are reference types in C#? ⭐⭐

Answer: The **reference types** do not contain the actual data stored in a variable, but they contain a reference to the variables.

In other words, they refer to a memory location. Using multiple variables, the reference types can refer to a memory location. If the data in the memory location is changed by one of the variables, the other variable automatically reflects this change in value. Example of built-in reference types are: object, dynamic, and string.

Source: *tutorialspoint.com*

Q18: What you understand by Value types and Reference types in C#.Net? ⭐⭐

Answer: In C# data types can be of two types: **Value Types** and **Reference Types**. Value type variables contain their object (or data) directly. If we copy one value type variable to another then we are actually making a copy of the object for the second variable. Value Type member will located into Stack and reference member will located in Heap always.

Source: *stackoverflow.com*

Q19: What is namespace in C#? ⭐⭐

Answer: A **namespace** is designed for providing a way to keep one set of names separate from another. The class names declared in one namespace does not conflict with the same class names declared in another.

Source: *tutorialspoint.com*

Q20: What is Serialization? ⭐⭐

Answer: Serialization means saving the state of your object to secondary memory, such as a file.

1. Binary serialization (Save your object data into binary format).
2. Soap Serialization (Save your object data into binary format; mainly used in network related communication).
3. XmlSerialization (Save your object data into an XML file).

Source: *c-sharpcorner.com*

Q21: In how many ways you can pass parameters to a method? ⭐⭐

Answer: There are three ways that parameters can be passed to a method:

- **Value parameters** – This method copies the actual value of an argument into the formal parameter of the function. In this case, changes made to the parameter inside the function have no effect on the argument.
- **Reference parameters** – This method copies the reference to the memory location of an argument into the formal parameter. This means that changes made to the parameter affect the argument.
- **Output parameters** – This method helps in returning more than one value.

Source: *tutorialspoint.com*

Q22: Can you return multiple values from a function in C#? ⭐⭐

Answer: Yes! Using output parameters. A return statement can be used for returning only one value from a function. However, using output parameters, you can return two values from a function.

Source: *tutorialspoint.com*

Q23: What is LINQ in C#? ⭐⭐

Answer: LINQ stands for Language Integrated Query. LINQ has a great power of querying on any source of data. The data source could be collections of objects, database or XML files. We can easily retrieve data from any object that implements the `IEnumerable<T>` interface.

Source: *c-sharpcorner.com*

Q24: Can multiple catch blocks be executed? ⭐⭐

Answer: No, Multiple catch blocks can't be executed. Once the proper catch code executed, the control is transferred to the finally block and then the code that follows the finally block gets executed.

Source: *guru99.com*

Q25: What is an Abstract Class? ⭐⭐

Answer: An **Abstract class** is a class which is denoted by abstract keyword and can be used only as a Base class. An Abstract class should always be inherited. An instance of the class itself cannot be created. If we do not want any program to create an object of a class, then such classes can be made abstract.

Any method in the abstract class does not have implementations in the same class. But they must be implemented in the child class.

Source: *softwaretestinghelp.com*

Q26: What are Custom Exceptions? ⭐⭐

Answer: Sometimes there are some errors that need to be handeled as per user requirements. Custom exceptions are used for them and are used defined exceptions.

Source: *guru99.com*

Q27: What is the difference between a struct and a class in C#? ⭐⭐

Answer: Class and struct both are the user defined data type but have some major difference:
**
Struct**

- The struct is value type in C# and it inherits from System.Value Type.
- Struct is usually used for smaller amounts of data.
- Struct can't be inherited to other type.
- A structure can't be abstract.
- No need to create object by new keyword.
- Do not have permission to create any default constructor.

Class

- The class is reference type in C# and it inherits from the System.Object Type.
- Classes are usually used for large amounts of data.
- Classes can be inherited to other class.
- A class can be abstract type.
- We can't use an object of a class with using new keyword.
- We can create a default constructor.

Source: *c-sharpcorner.com*

Q28: Why can't you specify the accessibility modifier for methods inside the interface? ⭐⭐

Answer: In an interface, we have virtual methods that do not have method definition. All the methods are there to be overridden in the derived class. That's why they all are public.

Source: *guru99.com*

Q29: What are the different types of classes in C#? ⭐⭐

Answer: The different types of class in C# are:

- **Partial class** – Allows its members to be divided or shared with multiple .cs files. It is denoted by the keyword *Partial.*
- **Sealed class** – It is a class which cannot be inherited. To access the members of a sealed class, we need to create the object of the class. It is denoted by the keyword *Sealed*.
- **Abstract class** – It is a class whose object cannot be instantiated. The class can only be inherited. It should contain at least one method. It is denoted by the keyword *abstract.*
- **Static class** – It is a class which does not allow inheritance. The members of the class are also static. It is denoted by the keyword *static*. This keyword tells the compiler to check for any accidental instances of the static class.

Source: *softwaretestinghelp.com*

Q30: What are dynamic type variables in C#? ⭐⭐

Answer: You can store any type of value in the dynamic data type variable. Type checking for these types of variables takes place at run-time.

Source: *tutorialspoint.com*

Q31: What is the difference between Equality Operator (==) and Equals() Method in C#? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q32: How encapsulation is implemented in C#? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q33: What is the difference between dynamic type variables and object type variables? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q34: What is extension method in C# and how to use them? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q35: What is lambda expressions in C#? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q36: What is Virtual Method in C#? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q37: What is the difference between ref and out keywords? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q38: What is delegates in C# and uses of delegates? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q39: What is sealed class in C#? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q40: What is an anonymous function in C#? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q41: What are the uses of “using” in C# ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q42: What is the difference between overloading and overriding? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q43: What is a Destructor in C#? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q44: Explain Anonymous type in C# ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q45: Refactor the code ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q46: What is Reflection in C#.Net? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q47: What is the output of the program below? Explain your answer. ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q48: What is difference between constants and readonly? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q49: Given an array of ints, write a C# method to total all the values that are even numbers. ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q50: What is the difference between Interface and Abstract Class? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q51: What is difference between Throw Exception and Throw Clause? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q52: What is the use of Null Coalescing Operator (??) in C#? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q53: What is the difference between constant and readonly in c#? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q54: Explain Code compilation in C# ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q55: What is the difference between Virtual method and Abstract method? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q56: What is scope of a Internal member variable of a C# class? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q57: Can Multiple Inheritance implemented in C# ? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q58: What are the different ways a method can be overloaded? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q59: Describe the accessibility modifier "protected internal". ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q60: What are the differences between a multidimensional array and an array of arrays in C#? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q61: Explain what is short-circuit evaluation in C# ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q62: Explain the difference between Select and Where ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q63: When to use ArrayList over array[] in c#? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q64: What is an Object Pool in .Net? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q65: Name difference between "is" and "as" operator in C# ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q66: What is the difference between Func<string,string> and delegate? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q67: What are pointer types in C#? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q68: What is marshalling and why do we need it? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q69: What is the difference between dispose and finalize methods in c#? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q70: What is scope of a Protected Internal member variable of a C# class? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q71: IEnumerable vs List - What to Use? How do they work? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q72: What is Indexer in C#? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q73: Can you create a function in C# which can accept varying number of arguments? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q74: Is operator overloading supported in C#? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q75: What interface should your data structure implement to make the "Where" method work? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q76: What is the use of conditional preprocessor directive in C#? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q77: What is the difference between System.ApplicationException class and System.SystemException class? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q78: What is difference between late binding and early binding in C#? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q79: Can we have only “try” block without “catch” block in C#? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q80: In try block if we add return statement whether finally block is executed in C#? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q81: What's the difference between StackOverflowError and OutOfMemoryError? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q82: What are the uses of delegates in C#? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q83: Why to use lock statement in C#? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q84: What is the "yield" keyword used for in C#? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q85: What is the output of the program below? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q86: What is the Constructor Chaining in C#? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q87: What is the output of the short program below? Explain your answer. ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q88: Is the comparison of time and null in the if statement below valid or not? Why or why not? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q89: What is the output of the program below? Explain your answer. ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q90: What is the best practice to have best performance using Lazy objects? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q91: Calculate the circumference of the circle ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q92: Could you explain the difference between destructor, dispose and finalize method? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q93: What are the differences between IEnumerable and IQueryable? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q94: What is the use of static constructors? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q95: What's the difference between the System.Array.CopyTo() and System.Array.Clone()? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q96: What is a preprocessor directives in C#? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q97: What is multicast delegate in C#? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q98: What are the benefits of a Deferred Execution in LINQ? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q99: What is the method MemberwiseClone() doing? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q100: List some different ways for equality check in .Net ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q101: What is deep or shallow copy concept in C#? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q102: Can you create sealed abstract class in C#? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q103: Could you explain the difference between Func vs. Action vs. Predicate? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q104: What are circular references? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q105: Explain the difference between IQueryable, ICollection, IList & IDictionary interfaces? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q106: in C#, when should we use abstract classes instead of interfaces with extension methods? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q107: Can you add extension methods to an existing static class? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q108: Implement the "Where" method in C# ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q109: What is the “volatile” keyword used for? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q110: What is jagged array in C#.Net and when to prefer jagged arrays over multi-dimensional arrays? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q111: Explain what is weak reference in C#? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q112: What is the difference between lambdas and delegates? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] CSS Interview Questions

Q1: Explain the three main ways to apply CSS styles to a web page ⭐

Answer: Using the inline style attribute on an element

```
<div>
    <p style="color: maroon;"></p>
</div>
```

Using a `<style>` block in the `<head>` section of your HTML

```
<head>
    <title>CSS Refresher</title>
    <style>
        body {
            font-family: sans-serif;
            font-size: 1.2em;
        }
    </style>
</head>
```

Loading an external CSS file using the `<link>` tag

```
<head>
    <title>CSS Refresher</title>
    <link rel="stylesheet" href="/css/styles.css" />
</head>
```

Source: *goskills.com*

Q2: What is CSS? ⭐

Answer: CSS stands for **Cascading Style Sheets**. CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

CSS was intended to allow web professionals to separate the content and structure of a website's code from the visual design.

Source: *w3schools.com*

Q3: How to use variables in Sass? ⭐

Answer: Think of variables as a way to store information that you want to reuse throughout your stylesheet. You can store things like colors, font stacks, or any CSS value you think you'll want to reuse. Sass uses the `$` symbol to make something a variable.

```
$font-stack:    Helvetica, sans-serif;
$primary-color: #333;

body {
  font: 100% $font-stack;
  color: $primary-color;
}
```

Source: *sass-lang.com*

Q4: What is variable interpolation in Sass? Provide some examples. ⭐⭐

Answer: If you want to use variables inside a string, you will have to use a process called **variable interpolation**. To use it you will have to wrap your variables in `#{}` .

Consider:

```
$name: 'Gajendar';
$author: 'Author : $name'; // 'Author : $name'

$author: 'Author : #{$name}';
// 'Author : Gajendar'
```

The interpolation method could be useful in situations where the value of a variable is determined by some conditional statements.

Source: *sitepoint.com*

Q5: What is a CSS rule? ⭐⭐

Answer: Web browsers apply **CSS rules** to a document to affect how they are displayed. A CSS rule is formed from:

- A **set of properties**, which have values set to update how the HTML content is displayed,

- A **selector**, which selects the element(s) you want to apply the updated property values to.

A set of CSS rules contained within a stylesheet determines how a webpage should look.

Source: *developer.mozilla.org*

Q6: What is DOM (Document Object Model) and how is it linked to CSS? ⭐⭐

Answer: The *Document Object Model (DOM)* is a cross-platform and language-independent *application programming interface* that treats an HTML, XHTML, or XML document as a tree structure wherein each node is an object representing a part of the document.

With the Document Object Model, programmers can create and build documents, navigate their structure, and add, modify, or delete elements and content. The DOM specifies interfaces which may be used to manage XML or HTML documents.

When a browser displays a document, it must combine the document's content with its style information. The browser converts HTML and CSS into the DOM (Document Object Model). The DOM represents the document in the computer's memory. It combines the document's content with its style.

Source: *en.wikipedia.org*

Q7: Have you played around with the new CSS Flexbox or Grid specs? ⭐⭐

Answer: Yes. Flexbox is mainly meant for 1-dimensional layouts while Grid is meant for 2-dimensional layouts.

Flexbox solves many common problems in CSS, such as vertical centering of elements within a container, sticky footer, etc. Bootstrap and Bulma are based on Flexbox, and it is probably the recommended way to create layouts these days. Have tried Flexbox before but ran into some browser incompatibility issues (Safari) in using `flex-grow` , and I had to rewrite my code using `inline-blocks` and math to calculate the widths in percentages, it wasn't a nice experience.

Grid is by far the most intuitive approach for creating grid-based layouts (it better be!) but browser support is not wide at the moment.

Source: *codeburst.io*

Q8: What is Sass? ⭐⭐

Answer: Sass or **Syntactically Awesome StyleSheets** is a *CSS* preprocessor that adds power and elegance to the basic language. It allows you to use variables, nested rules, mixins, inline imports, and more, all with a fully CSS-compatible syntax. Sass helps keep large stylesheets well-organized, and get small stylesheets up and running quickly.

A *CSS preprocessor* is a scripting language that extends CSS by allowing developers to write code in one language and then compile it into CSS.

Source: *sass-lang.com*

Q9: List out the key features for Sass? ⭐⭐

Answer: Key features for Sass include

- Full CSS3-compatible
- Language extensions such as nesting, variables, and mixins
- Many useful functions for manipulating colors and other values
- Advanced features like control directives for libraries
- Well-formatted, customizable output

Source: *career.guru99.com*

Q10: What existing CSS frameworks have you used locally, or in production? How would you change/improve them? ⭐⭐

Answer:

- **Bootstrap** - Slow release cycle. Bootstrap 4 has been in alpha for almost 2 years. Add a spinner button component, as it is widely used.
- **Semantic UI** - Source code structure makes theme customization extremely hard to understand. Its unconventional theming system is a pain to customize. Hardcoded config path within the vendor library. Not well-designed for overriding variables unlike in Bootstrap.
- **Bulma** - A lot of non-semantic and superfluous classes and markup required. Not backward compatible. Upgrading versions breaks the app in subtle manners.

Source: *codeburst.io*

Q11: List out the data types that Sass supports ⭐⭐

Answer: Sass supports seven main data types:

- **Numbers** - most of the time they are accompanied by a unit of some sort but they are still technically numbers. You can perform basic mathematical operations on these values.

```
$size: 18;                  // A number
$px-unit: $size * 1px;      // A pixel measurement
$px-string: $size + px;     // A string
$px-number: $px-unit / 1px; // A number
```

- **Strings** - just like CSS, accepts both quoted and unquoted strings, even if they contain spaces

```
$website: 'SitePoint'; // Stores SitePoint
$name: 'Gajendar' + ' Singh';  // 'Gajendar Singh'
$date:  'Month/Year : ' + 3/2016; // 'Month/Year : 3/2016'
$date:  'Month/Year : ' + (3/2016); // 'Month/Year : 0.00149'
// This is because 3/2016 is evaluated first.
$variable: 3/2016;      // Evaluated to 0.00149
```

- **Colors** - CSS color expressions come under the `color` data type. You can refer to the colors in hexadecimal notation, as `rgb` , `rgba` , `hsl` and `hsla` values or use native keywords like `pink` , `blue` , etc.

```
$color: yellowgreen;           // #9ACD32
color: lighten($color, 15%);    // #b8dc70
color: darken($color, 15%);     // #6c9023
color: saturate($color, 15%);   // #a1e01f
color: desaturate($color, 15%); // #93ba45
color: (green + red);           // #ff8000
```

- **Booleans** - has only two possible values: `true` and `false`

```
$i-am-true: true;

body {
  @if not $i-am-true {
    background: rgba(255, 0, 0, 0.6);
  } @else {
    background: rgba(0, 0, 255, 0.6); // expected
  }
}
```

- **Null** - is commonly used to define an empty state, neither `true` or `false` . This is typically the value you want to set when defining a variable without a value, only to prevent the parser from crashing.

```
.foo {
  content: type-of(null); // null
  content: type-of(NULL); // string
  $bar: 'foo' + null; // invalid null operation: "foo plus null”.
}
```

- **Lists** - are just the Sass version of arrays. You can store multiple types of values in a list.

```
$font-list: 'Raleway','Dosis','Lato'; // Three comma separated elements
$pad-list: 10px 8px 12px; // Three space separated elements
$multi-list: 'Roboto',15px 1.3em; // This multi-list has two lists.
```

- **Maps** - Sass maps are like associative arrays. A map stores both keys and values associated with those keys.

```
$styling: (
  'font-family': 'Lato',
  'font-size': 1.5em,
  'color': tomato,
  'background': black
);

h1 {
  color: map-get($styling, 'color');
  background: map-get($styling, 'background');
}
```

Source: *career.guru99.com*

Q12: Explain CSS sprites, and how you would implement them on a page or site. ⭐ ⭐

Answer: *CSS sprites* combine multiple images into one single larger image. It is commonly used technique for icons (Gmail uses it).

- Use a sprite generator that packs multiple images into one and generate the appropriate CSS for it.

- Each image would have a corresponding CSS class with `background-image` , `background-position` and `background-size` properties defined.
- To use that image, add the corresponding class to your element.

Advantages:

- Reduce the number of HTTP requests for multiple images (only one single request is required per spritesheet). But with HTTP2, loading multiple images is no longer much of an issue.
- Advance downloading of assets that won’t be downloaded until needed, such as images that only appear upon `:hover` pseudo-states. Blinking wouldn't be seen.

Source: *codeburst.io*

Q13: What Selector Nesting in Sass is used for? ⭐⭐

Answer: Sass *let you nest* your CSS selectors in a way that follows the same visual hierarchy of your HTML. CSS, on the other hand, doesn't have any visual hierarchy.

Consider example (scss):

```
.parent {
  color: red;

  .child {
    color: blue;
  }
}
```

Result (css):

```
.parent {
  color: red;
}

.parent .child {
  color: blue;
}
```

Source: *sass-lang.com*

Q14: Explain what is a @extend directive used for in Sass? ⭐⭐

Answer: Using `@extend` lets you share a set of CSS properties from one selector to another. It helps keep your Sass very dry.

Consider:

```
%message-shared {
  border: 1px solid #ccc;
  padding: 10px;
  color: #333;
}

.message {
  @extend %message-shared;
}

.success {
  @extend %message-shared;
  border-color: green;
}

.error {
  @extend %message-shared;
  border-color: red;
}

.warning {
  @extend %message-shared;
  border-color: yellow;
}
```

CSS output:

```
.message, .success, .error, .warning {
  border: 1px solid #cccccc;
  padding: 10px;
  color: #333;
}

.success {
  border-color: green;
}

.error {
  border-color: red;
}

.warning {
  border-color: yellow;
}
```

Source: *career.guru99.comsass-lang.com*

Q15: Explain the CSS “box model” and the layout components that it consists of ⭐⭐

Answer: The CSS box model is a rectangular layout paradigm for HTML elements that consists of the following:

- **Content** - The content of the box, where text and images appear
- **Padding** - A transparent area surrounding the content (i.e., the amount of space between the border and the content)
- **Border** - A border surrounding the padding (if any) and content
- **Margin** - A transparent area surrounding the border (i.e., the amount of space between the border and any neighboring elements)

Source: *toptal.com*

Q16: What is the difference between classes and IDs in CSS? ⭐⭐

Answer:

- **IDs**—Meant to be unique within the document. Can be used to identify an element when linking using a fragment identifier. Elements can only have one id attribute.

- **Classes**—Can be reused on multiple elements within the document. Mainly for styling and targeting elements.

Source: *codeburst.io*

Q17: Describe floats and how they work ⭐⭐

Answer: *Float* is a CSS positioning property. Floated elements remain a part of the flow of the web page. This is distinctly different than page elements that use absolute positioning. Absolutely positioned page elements are removed from the flow of the webpage.

```
#sidebar {
  float: right; // left right none inherit
}
```

The CSS clear property can be used to be positioned below `left` / `right` / `both` floated elements.

Source: *css-tricks.com*

Q18: Explain the usage of "table-layout" property ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q19: How to create a zebra striped table with CSS? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q20: Have you ever worked with retina graphics? If so, when and what techniques did you use? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q21: What is a Mixin and how to use on? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q22: How is responsive design different from adaptive design? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q23: Explain your understanding of the box model and how you would tell the browser in CSS to render your layout in different box models. ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q24: What's the difference between SCSS and Sass? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q25: Describe pseudo-elements and discuss what they are used for. ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q26: How does CSS actually work (under the hood of browser)? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q27: What is CSS selectors? Name some. ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q28: What’s the difference between “resetting” and “normalizing” CSS? Which would you choose, and why? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q29: What's the difference between a `relative`, `fixed`, `absolute` and `static`ally positioned element? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q30: What are the advantages/disadvantages of using CSS preprocessors? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q31: What is CSS preprocessor and why to user one? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q32: How would you approach fixing browser-specific styling issues? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q33: What is a Grid System in CSS? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q34: What does Accessibility (a11y) mean? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q35: What does * { box-sizing: border-box; } do? What are its advantages? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q36: Have you ever used a grid system, and if so, what do you prefer? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q37: Explain the purpose of clearing floats in CSS ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q38: Can you explain the difference between coding a website to be responsive versus using a mobile-first strategy? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q39: How do you optimize your webpages for print? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q40: Describe z-index and how a stacking context is formed ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q41: Explain the basic rules of CSS Specificity ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q42: What is the '@content' directive used for? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q43: What are the different ways to visually hide content (and make it available only for screen readers)? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q44: What will be the CSS output for the following Sass code? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q45: Is there any reason you'd want to use `translate()` instead of `absolute` positioning, or vice-versa? And why? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q46: What the code fragment has the greater CSS specificity? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q47: What’s wrong with Sass nesting? Provide some example. ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q48: What clearfix methods do you know? Provide some examples. ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q49: How to style every element which has an adjacent item right before it? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q50: Write down a selector that will match any links end in .zip, .ZIP, .Zip etc... ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] Career Interview Questions

Q1: career test ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] Clojure Interview Questions

Q1: What is Clojure? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q2: What is Data-oriented programming? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q3: What does it mean "Clojure is immutable-first"? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q4: How do you make a web application using Clojure? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] Code Problems Interview Questions

Q1: Test divisors of three ⭐

Details: You will be given 2 parameters: a low and high number. Your goal is to print all numbers between low and high, and for each of these numbers print whether or not the number is divisible by 3. If the number is divisible by 3, print the word "div3" directly after the number.

Answer: We'll solve this problem by first creating a loop that will print each number from low to high. Once we have the code for that written, we'll add a conditional that will check if the number is evenly divisible by 3 by using the mod operator.

```
function test_divisors(low, high) {

  // we'll store all numbers and strings within an array
  // instead of printing directly to the console
  var output = [];

  for (var i = low; i <= high; i++) {

    // simply store the current number in the output array
    output.push(i);

    // check if the current number is evenly divisible by 3
    if (i % 3 === 0) { output.push('div3'); }

  }

  // return all numbers and strings
  return output;

}

test_divisors(2, 10);
```

Source: *coderbyte.com*

Q2: Sum of Array Plus One ⭐

Details: Write a function that takes an array of integers and returns the sum of the integers after adding 1 to each.

Answer:

```
// ES5 method is nice and clean
exports.es5 = function (array) {
  return array.reduce(function (memo, num) {
    return memo + num;
  }, array.length);
};

// Without array.reduce method isn't much different
exports.iterative = function (array) {
  var result = array.length;
  for (var i = 0; i < array.length; i++) {
    result += array[i];
  }

  return result;
};
```

Source: *github.com/blakeembrey*

Q3: String Rotation ⭐

Details: Find out if a string is a rotation of another string. E.g. `ABCD` is a rotation of `BCDA` but not `ACBD`.

Answer: First make sure `a` and `b` are of the same length. Then check to see if `b` is a substring of `a` concatenated with `a`:

```
module.exports = function (a, b) {
  return a.length === b.length && (a + a).indexOf(b) > -1;
};
```

Source: *github.com/blakeembrey*

Q4: Oddball sum ⭐

Details: Write a function called `oddball_sum` which takes in a list of numbers and returns the sum of all the odd elements. Try to solve with and without `reduce` function.

Answer: To solve this challenge we'll simply loop through the array while maintaining a final count, and every time an odd number is encountered we'll add it to the count.

Without `reduce`:

```
function oddball_sum(nums) {

  // final count of all odd numbers added up
  var final_count = 0;

  // loop through entire list
  for (var i = 0; i < nums.length; i++) {

    // we divide by 2, and if there is a remainder then
    // the number must be odd so we add it to final_count
    if (nums[i] % 2 === 1) {
      final_count += nums[i]
    }

  }

  return final_count;

}

oddball_sum([1, 2, 3, 4, 5]);
```

With `reduce`:

```
function oddball_sum(nums) {
  return nums.reduce(function(total, item){
        if (item % 2 === 1) {
                return total += item;
        }
        return total;
  });
}
```

Source: *prepwork.appacademy.io*

Q5: Simple clock angle ⭐

Details: You will be given a number `N` that represents where the minute hand currently is on a clock. Your program should return the angle that is formed by the minute hand and the `12` o'clock mark on the clock.

Answer: If the input is `15` then your program should return `90` because a `90`-degree angle is formed by the minute hand and the `12` o'clock mark on the clock. We'll solve this challenge by first calculating what angle is created by each minute passing on a clock. Once we calculate this number, we multiply it by the input to determine the final angle.

A method to solve such problems is to consider the rate of change of the angle in degrees per minute. The hour hand of a normal `12-hour` analogue clock turns `360°` in `12` hours (`720` minutes) or `0.5°` per minute. The minute hand rotates through `360°` in `60` minutes or `6°` per minute.

```
function simpleClockAngle(num) {

  // we got 6 because 360/60 = 6
  // 360 represents the full number of a degrees in a circle and
  // 60 is the number of minutes on a clock, so dividing these two numbers
  // gives us the number of degrees for one minute
  return 6 * num;

}

simpleClockAngle(15);
```

Source: *coderbyte.com*

Q6: Sum of several arrays ⭐

Details: You will be given an array of several arrays that each contain integers and your goal is to write a function that will sum up all the numbers in all the arrays. For example, if the input is `[[3, 2], [1], [4, 12]]` then your program should output `22` because `3 + 2 + 1 + 4 + 12 = 22`. Solve without and with `reduce`.

Answer: We will solve this challenge by looping through the entire array, and then looping through each inner array adding up all the numbers.

```
function sum_array(arr) {
  // store our final answer
  var sum = 0;
  // loop through entire array
  for (var i = 0; i < arr.length; i++) {
    // loop through each inner array
    for (var j = 0; j < arr[i].length; j++) {
      // add this number to the current final sum
      sum += arr[i][j];
    }
  }

  return sum;
}

sum_array([[3, 2], [1], [4, 12]]);
```

With `reduce`:

```
function sumArray(arr) {
  return arr.reduce((t, e) => t.concat(e)).reduce((t, e) => t + e)
}
```

Source: *coderbyte.com*

Q7: Lucky sevens ⭐

Details: Write a function called `lucky_sevens` which takes an array of integers and returns true if any three consecutive elements sum to 7.

Answer: To solve this challenge we'll simply loop through the array starting at the 3rd position, and checking if the number at this index plus the two previous elements sums to 7. We continue doing this as we loop through the entire array. Once we find three elements that sum to 7, we simply return `true` . If we reach the end of the array without finding elements that sum to 7, we return `false` .

```
function lucky_sevens(arr) {

  // if less than 3 elements then this challenge is not possible
  if (arr.length < 3) {
    return "not possible";
  }

  // because we know there are at least 3 elements we can
  // start the loop at the 3rd element in the array (i=2)
  // and check it along with the two previous elements (i-1) and (i-2)
  for (var i = 2; i < arr.length; i++) {
    if (arr[i] + arr[i-1] + arr[i-2] === 7) {
      return true;
    }
  }

  // if loop is finished and no elements summed to 7
  return false;

}

lucky_sevens([2, 1, 5, 1, 0]);
```

Source: *coderbyte.com*

Q8: Two sum problem ⭐⭐

Details: Given an integer `x` and a sorted array a of `N` distinct integers, design a linear-time algorithm to determine if there exists two distinct indices `i` and `j` such that `a[i] + a[j] == x`

For example, if the array is `[3, 5, 2, -4, 8, 11]` and the sum is `7`, your program should return `[[11, -4], [2, 5]]` because `11 + -4 = 7` and `2 + 5 = 7`.

Answer: The algorithm below makes use of hash tables which have a constant lookup time. As we pass through each element in the array, we check to see if S minus the current element exists in the hash table. We only need to loop through the array once, resulting in a running time of `O(n)` since each lookup and insertion in a hash table is `O(1)`.

```
// our two sum function which will return
// all pairs in the array that sum up to S
function twoSum(arr, S) {

  var sums = [];
  var hashTable = {};

  // check each element in array
  for (var i = 0; i < arr.length; i++) {

    // calculate S - current element
    var sumMinusElement = S - arr[i];

    // check if this number exists in hash table
    // if so then we found a pair of numbers that sum to S
    if (hashTable[sumMinusElement.toString()] !== undefined) {
      sums.push([arr[i], sumMinusElement]);
    }

    // add the current number to the hash table
    hashTable[arr[i].toString()] = arr[i];

  }

  // return all pairs of integers that sum to S
  return sums;

}

twoSum([3, 5, 2, -4, 8, 11], 7);
```

Source: *coderbyte.com*

Q9: Implement a queue using a linked list ⭐⭐

Answer: We will store a reference to the front and back of the queue in order to make enqueuing and dequeuing run in `O(1)` constant time. Every time we want to insert into the queue, we add the new element to the end of the linked list and update the back pointer. When we want to dequeue we return the first node in the linked list and update the front pointer.

```
// queue is initially empty
var Queue = {front: null, back: null};

// we will use a node to keep track of the elements
// in the queue which is represented by a linked list
function Node(data, next) {
  this.data = data;
  this.next = next;
}

// add elements to queue in O(1) time
function Enqueue(element) {
  var N = new Node(element, null);
  if (Queue.back === null) {
    Queue.front = N;
    Queue.back = N;
  } else {
    Queue.back.next = N;
    Queue.back = Queue.back.next;
  }
}

// remove first element from queue in O(1) time
function Dequeue() {
  if (Queue.front !== null) {
    var first = Queue.front;
    Queue.front = Queue.front.next;
    return first.data;
  } else {
    if (Queue.back !== null) { Queue.back = null; }
    return 'Cannot dequeue because queue is empty';
  }
}

Enqueue('a');
Enqueue('b');
Enqueue('c');
Dequeue();
```

Source: *codeisahighway.com*

Q10: Tree Level Order Print ⭐⭐

Details: Given a binary tree of integers, print it in level order. The output will contain space between the numbers in the same level, and new line between different levels.

Answer:

```
module.exports = function (root) {
  // Doing a breadth first search using recursion.
  (function walkLevel (children) {
    // Create a new queue for the next level.
    var queue = [],
        output;

    // Use the map function to easily join all the nodes together while pushi
    // it's children into the next level queue.
    output = children.map(function (node) {
      // Assuming the node has children stored in an array.
      queue = queue.concat(node.children || []);
      return node.value;
    }).join(' ');

    // Log the output at each level.
    console.log(output);

    // If the queue has values in it, recurse to the next level and walk
    // along it.
    queue.length && walkLevel(queue);
  })([root]);
};
```

Source: *ardendertat.com*

Q11: Stock maximum profit ⭐⭐

Details: You will be given a list of stock prices for a given day and your goal is to return the maximum profit that could have been made by buying a stock at the given price and then selling the stock later on.

For example if the input is:

```
[45, 24, 35, 31, 40, 38, 11]
```

then your program should return 16 because if you bought the stock at $24 and sold it at $40, a profit of $16 was made and this is the largest profit that could be made. If no profit could have been made, return -1.

Answer: We'll solve the challenge the following way:

1. Iterate through each number in the list.
2. At the ith index, get the `i+1` index price and check if it is larger than the ith index price.
3. If so, set `buy_price = i` and `sell_price = i+1`. Then calculate the profit: `sell_price - buy_price`.
4. If a stock price is found that is cheaper than the current `buy_price`, set this to be the new buying price and continue from step 2.
5. Otherwise, continue changing only the `sell_price` and keep `buy_price` set.

This algorithm runs in linear time, making only one pass through the array, so the running time in the worst case is `O(n)`.

```
function StockPicker(arr) {

  var max_profit = -1;
  var buy_price = 0;
  var sell_price = 0;

  // this allows our loop to keep iterating the buying
  // price until a cheap stock price is found
  var change_buy_index = true;

  // loop through list of stock prices once
  for (var i = 0; i < arr.length-1; i++) {

    // selling price is the next element in list
    sell_price = arr[i+1];

    // if we have not found a suitable cheap buying price yet
    // we set the buying price equal to the current element
    if (change_buy_index) { buy_price = arr[i]; }

    // if the selling price is less than the buying price
    // we know we cannot make a profit so we continue to the
    // next element in the list which will be the new buying price
    if (sell_price < buy_price) {
      change_buy_index = true;
      continue;
    }
    // if the selling price is greater than the buying price
    // we check to see if these two indices give us a better
    // profit then what we currently have
    else {
      var temp_profit = sell_price - buy_price;
      if (temp_profit > max_profit) { max_profit = temp_profit; }
      change_buy_index = false;
    }

  }

  return max_profit;

}

StockPicker([44, 30, 24, 32, 35, 30, 40, 38, 15]);
```

Source: *coderbyte.com*

Q12: Find Word Positions in Text ⭐⭐

Details: Given a text file and a word, find the positions that the word occurs in the file. We’ll be asked to find the positions of many words in the same file.

Answer: Since we’ll have to answer multiple queries, precomputation would be useful. We’ll build a data structure that stores the positions of all the words in the text file. This is known as inverted index.

```
module.exports = function (text) {
  var trie   = {},
      pos    = 0,
      active = trie; // Start the active structure as the root trie structure

  // Suffix a space after the text to make life easier
  text += ' ';

  // Loop through the input text adding it to the trie structure
  for (var i = 0; i < text.length; i++) {
    // When the character is a space, restart
    if (text[i] === ' ') {
      // If the current active doesn't equal the root, set the position
      if (active !== trie) {
        (active.positions = active.positions || []).push(pos);
      }
      // Reset the positions and the active part of the data structure
      pos    = i;
      active = trie;
      continue;
    }

    // Set the next character in the structure up
    active[text[i]] = (active[text[i]] || {});
    active = active[text[i]];
  }

  // Return a function that accepts a word and looks it up in the trie struct
  return function (word) {
    var i      = -1,
        active = trie;

    while (word[++i]) {
      if (!active[word[i]]) { return []; }
      active = active[word[i]];
    }

    return active.positions;
  };
};
```

Source: *github.com/blakeembrey*

Q13: Determine overlapping numbers in ranges ⭐⭐

Details: You will be given an array with `5` numbers. The first 2 numbers represent a range, and the next two numbers represent another range. The final number in the array is `X`. The goal of your program is to determine if both ranges overlap by at least `X` numbers. For example, in the array `[4, 10, 2, 6, 3]` the ranges `4` to `10` and `2` to `6` overlap by at least `3` numbers `(4, 5, 6)`, so your program should return `true`. Solve with and without looping.

If the array is `[10, 20, 4, 14, 4]` then the ranges are:

```
10 11 12 13 14 15 16 17 18 19 20
4 5 6 7 8 9 10 11 12 13 14
```

These ranges overlap by at least `4` numbers, namely: `10, 11, 12, 13, 14` so your program should return `true`.

Answer: With loop:

```
function OverlappingRanges(arr) {

  // keep a count of how many numbers overlap
  var counter = 0;

  // loop through one of the ranges
  for (var i = arr[0]; i < arr[1]; i++) {

    // check if a number within the first range exists
    // in the second range
    if (i >= arr[2] && i <= arr[3]) {
      counter += 1;
    }

  }

  // check if the numbers that overlap is equal to or greater
  // than the last number in the array
  return (counter >= arr[4]) ? true : false;
}

OverlappingRanges([4, 10, 2, 6, 3]);
```

Without loop:

```
function overlapping(input){
  var nums1 = listOfNums(input[0], input[1]);
  var nums2 = listOfNums(input[2], input[3]);
  var overlappingNum = 0;

  if(nums1[0] >= nums2[0] && nums1[0] <= nums2[1]){
    overlappingNum =  nums2[1] - nums1[0] + 1;
  } else {
    overlappingNum =  nums1[1] - nums2[0] + 1;
  }
  if(overlappingNum >= input[4]){
    return true;
  }
}

function listOfNums(a, b){
  var start = a;
  var end = b;
  if(a > b){
    start = b;
    end = a;
  }
  return [a, b];
}

var a = [4, 10, 2, 6, 3];
overlapping(a)
```

Source: *coderbyte.com*

Q14: Throttle Function Implementation ⭐⭐

Details: Write a function that accepts a function and timeout, `x`, in number of milliseconds. It returns a function that can only be executed once per `x` milliseconds. This can be useful for limiting the number of time and computation heavy function that are run. For example, making AJAX requests to an autocompletion API.

Once written, add a third parameter that will allow the function to be executed immediately if set to true. Otherwise the function will run at the end of the timeout period.

Answer:

```
module.exports = function (fn, delay, execAsap) {
  var timeout; // Keeps a reference to the timeout inside the returned functi

  return function () {
    // Continue to pass through the function execution context and arguments
    var that = this,
        args = arguments;

    // If there is no timeout variable set, proceed to create a new timeout
    if (!timeout) {
      execAsap && fn.apply(that, args);

      timeout = setTimeout(function () {
        execAsap || fn.apply(that, args);
        // Remove the old timeout variable so the function can run again
        timeout = null;
      }, delay || 100);
    }
  };
};
```

Source: *github.com/blakeembrey*

Q15: Dutch national flag sorting problem ⭐⭐

Details: For this problem, your goal is to sort an array of `0`, `1`, `2` but you must do this in place, in linear time and without any extra space (such as creating an extra array). This is called the *Dutch national flag sorting problem*. For example, if the input array is `[2,0,0,1,2,1]` then your program should output `[0,0,1,1,2,2]` and the algorithm should run in `O(n)` time.

Answer: The solution to this algorithm will require 3 pointers to iterate throughout the array, swapping the necessary elements.

1. Create a low pointer at the beginning of the array and a high pointer at the end of the array.
2. Create a mid pointer that starts at the beginning of the array and iterates through each element.
3. If the element at `arr[mid]` is a `2`, then swap `arr[mid]` and `arr[high]` and decrease the high pointer by `1`.
4. If the element at `arr[mid]` is a `0`, then swap `arr[mid]` and `arr[low]` and increase the low and mid pointers by `1`.
5. If the element at `arr[mid]` is a `1`, don't swap anything and just increase the mid pointer by `1`.

```
function swap(arr, i1, i2) {
  var temp = arr[i1];
  arr[i1] = arr[i2];
  arr[i2] = temp;
}

function dutchNatFlag(arr) {

  var low = 0;
  var mid = 0;
  var high = arr.length - 1;

  // one pass through the array swapping
  // the necessary elements in place
  while (mid <= high) {
    if      (arr[mid] === 0) { swap(arr, low++, mid++); }
    else if (arr[mid] === 2) { swap(arr, mid, high--); }
    else if (arr[mid] === 1) { mid++; }
  }

  return arr;

}
```

```
dutchNatFlag([2,2,2,0,0,0,1,1]);
```

Source: *coderbyte.com*

Q16: Step-by-step solution for step counting using recursion ⭐⭐

Details:

Suppose you want climb a staircase of N steps, and on each step you can take either 1 or 2 steps. How many distinct ways are there to climb the staircase? For example, if you wanted to climb 4 steps, you can take the following distinct number of steps:

```
1) 1, 1, 1, 1
2) 1, 1, 2
3) 1, 2, 1
4) 2, 1, 1
5) 2, 2
```

So there are 5 distinct ways to climb 4 steps. We want to write a function, using recursion, that will produce the answer for any number of steps.

Answer: The solution to this problem requires recursion, which means to solve for a particular `N`, we need the solutions for previous N's. The solution for N steps is equal to the solutions for `N - 1` steps plus `N - 2` steps.

```
function countSteps(N) {

  // just as in our solution explanation above, we know that to climb 1 step
  // there is only 1 solution, and for 2 steps there are 2 solutions
  if (N === 1) { return 1; }
  if (N === 2) { return 2; }

  // for all N > 2, we add the previous (N - 1) + (N - 2) steps to get
  // an answer recursively
  return countSteps(N - 1) + countSteps(N - 2);

}

// the solution for N = 6 will recursively be solved by calculating
// the solution for N = 5, N = 4, N = 3, and N = 2 which we know is 2
countSteps(6);
```

Source: *coderbyte.com*

Q17: Implement Bubble Sort ⭐⭐

Answer: The steps in the bubble sort algorithm are:

1. Loop through the whole array starting from index `1`
2. If the number in the array at index `i-1` is greater than i, swap the numbers and continue
3. Once the end of the array is reached, start looping again from the beginning
4. Once no more elements can be swapped, the sorting is complete

```
function swap(arr, i1, i2) {
  var temp = arr[i1];
  arr[i1] = arr[i2];
  arr[i2] = temp;
}

function bubblesort(arr) {

  var swapped = true;

  // keep going unless no elements can be swapped anymore
  while (swapped) {

    // set swapped to false so that the loop stops
    // unless two element are actually swapped
    swapped = false;

    // loop through the whole array swapping adjacent elements
    for (var i = 1; i < arr.length; i++) {
      if (arr[i-1] > arr[i]) {
        swap(arr, i-1, i);
        swapped = true;
      }
    }

  }

  return arr;

}

bubblesort([103, 45, 2, 1, 97, -4, 67]);
```

Source: *coderbyte.com*

Q18: Implement a queue using two stacks ⭐⭐

Answer: Suppose we push `a`, `b`, `c` to a stack. If we are trying to implement a queue and we call the dequeue method 3 times, we actually want the elements to come out in the order: `a`, `b`, `c`, which is in the opposite order they would come out if we popped from the stack. So, basically, we need to access the elements in the reverse order that they exist in the stack.

Algorithm for queue using two stacks:

1. When calling the enqueue method, simply push the elements into the stack 1.
2. If the dequeue method is called, push all the elements from stack 1 into stack 2, which reverses the order of the elements. Now pop from stack 2.

The worst case running time for implementing these operations using stacks is `O(n)` because you need to transfer n elements from stack 1 to stack 2 when a dequeue method is called. Pushing to stack 1 is simply `O(1)`.

```
// implement stacks using plain arrays with push and pop functions
var Stack1 = [];
var Stack2 = [];

// implement enqueue method by using only stacks
// and the push and pop functions
function Enqueue(element) {
  Stack1.push(element);
}

// implement dequeue method by pushing all elements
// from stack 1 into stack 2, which reverses the order
// and then popping from stack 2
function Dequeue() {
  if (Stack2.length === 0) {
    if (Stack1.length === 0) { return 'Cannot dequeue because queue is empty'
    while (Stack1.length > 0) {
      var p = Stack1.pop();
      Stack2.push(p);
    }
  }
  return Stack2.pop();
}

Enqueue('a');
Enqueue('b');
Enqueue('c');
Dequeue();
```

Source: *coderbyte.com*

Q19: Implement pow(a,b) without multiplication or division ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q20: Generate all balanced bracket combinations ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q21: All Permutations (Anagrams) of a String ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q22: Merge two sorted linked lists ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q23: Insert an interval into a list of sorted disjoint intervals ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q24: Find all string combinations consisting only of 0, 1 and ? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q25: Quickly calculate the cube root of 6 digit numbers ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q26: Transform Word ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] Data Science Interview Questions

Q1: What is Data Science? ⭐

Answer: Data Science is an interdisciplinary field of different scientific methods, techniques, processes, and knowledge that is used to transform the data of different types such as structured, unstructured and semi-structured data into the required format or representation.

Data Science concepts include different concepts such as statistics, regression, mathematics, computer science, algorithms, data structures and information science with also including some subfields such as data mining, machine learning, and databases etc.,

Data Science concept has recently evolved to a greater extent in the area of computing technology in order to perform data analysis on the existing data where the growth of data is in terms of an exponential with respect to time.

Data Science is the study of various types of data such as structured, semi-structured and unstructured data in any form or formats available in order to get some information out of it.

Data Science consists of different technologies used to study data such as data mining, data storing, data purging, data archival, data transformation etc., in order to make it efficient and ordered. Data Science also includes the concepts like Simulation, modeling, analytics, machine learning, computational mathematics etc.

Source: *www.educba.com*

Q2: What is the best Programming Language to use in Data Science? ⭐

Answer: Data Science can be handled by using programming languages like Python or R programming language. These two are the two most popular languages being used by the Data Scientists or Data Analysts. R and Python are open source and are free to use and came into existence during the 1990s.

Python and R have different advantages depending on the applications and required a business goal. Python is better to be used in the cases of repeated tasks or jobs and for data manipulations whereas R programming can be used for querying or retrieving datasets and customized data analysis.

Mostly Python is preferred for all types of data science applications where some time R programming is preferred in the cases of high or complex data applications. Python is easier to learn and has less learning curve whereas R has a deep learning curve.

Python is mostly preferred in all the cases which is a general-purpose programming language and can be found in many applications other than Data Science too. R is mostly seen in Data Science area only where it is used for data analysis in standalone servers or computing separately.

Source: *www.educba.com*

Q3: Why is data cleaning essential in Data Science? ⭐

Answer: Data cleaning is more important in Data Science because the end results or the outcomes of the data analysis come from the existing data where useless or unimportant need to be cleaned periodically as of when not required. This ensures the data reliability & accuracy and also memory is freed up.

Data cleaning reduces the data redundancy and gives good results in data analysis where some large customer information exists and that should be cleaned periodically. In the businesses like e-commerce, retail, government organizations contain large customer transaction information which is outdated and needs to be cleaned.

Depending on the amount or size of data, suitable tools or methods should be used to clean the data from the database or big data environment. There are different types of data existing in a data source such as dirty data, clean data, mixed clean and dirty data and sample clean data.

Modern data science applications rely on machine learning model where the learner learns from the existing data. So, the existing data should always be cleanly and well maintained to get sophisticated and good outcomes during the optimization of the system.

Source: *www.educba.com*

Q4: What is Linear Regression in Data Science? ⭐⭐

Answer: Linear Regression is a technique used in supervised machine learning algorithmic process in the area of Data Science. This method is used for predictive analysis.

Predictive analytics is an area within Statistical Sciences where the existing information will be extracted and processed to predict the trends and outcomes pattern. The core of the subject lies in the analysis of existing context to predict an unknown event.

The process of Linear Regression method is to predict a variable called target variable by making the best relationship between the dependent variable and an independent variable. Here dependent variable is outcome variable and also response variable whereas the independent variable is predictor variable or explanatory variable.

For example in real life, depending on the expenses occurred in this financial year or monthly expenses, the predictions happen by calculating the approximate upcoming months or financial years expenses.

In this method, the implementation can be done by using Python programming technique where this is the most important method used in Machine Learning technique under the area of Data Science.

Linear regression is also called Regression analysis that comes under the area of Statistical Sciences which is integrated together with Data Science.

Source: *www.educba.com*

Q5: What is A/B testing in Data Science? ⭐⭐

Answer: A/B testing is also called as Bucket Testing or Split Testing. This is the method of comparing and testing two versions of systems or applications against each other to determine which version of application performs better. This is important in the cases where multiple versions are shown to the customers or end users in order to achieve the goals.

In the area of Data Science, this A/B testing is used to know which variable out of the existing two variables in order to optimize or increase the outcome of the goal. A/B testing is also called Design of Experiment. This testing helps in establishing a cause and effect relationship between the independent and dependent variables.

This testing is also simply a combination of design experimentation or statistical inference. Significance, Randomization and Multiple Comparisons are the key elements of the A/B testing.

The significance is the term for the significance of statistical tests conducted. Randomization is the core component of the experimental design where the variables will be balanced. Multiple comparisons are the way of comparing more variables in the case of customer interests that causes more false positives resulting in the requirement of correction in the confidence level of a seller in the area of e-commerce.

A/B testing is the important one in the area of Data Science in predicting the outcomes.

Source: *www.educba.com*

[⬆️] Data Structures Interview Questions

Q1: What is data-structure? ⭐

Answer: Data structure availability may vary by programming languages. Commonly available data structures are:

- list,
- arrays,
- stack,
- queues,

- graph,
- tree etc.

Source: *tutorialspoint.com*

Q2: What is a graph? ⭐

Answer: A **graph** is a pictorial representation of a set of objects where some pairs of objects are connected by links. The interconnected objects are represented by points termed as vertices, and the links that connect the vertices are called edges.

Source: *tutorialspoint.com*

Q3: What is linear searching? ⭐

Answer: Linear search or sequential search is a method for finding a target value within a list. It sequentially checks each element of the list for the target value until a match is found or until all the elements have been searched. Linear search runs in at worst *linear time* and makes at most `n` comparisons, where `n` is the length of the list.

- Worst-case performance `O(n)`
- Best-case performance `O(1)`
- Average performance `O(n)`
- Worst-case space complexity `O(1)` iterative

In theory other search algorithms may be faster than linear search (for instance binary search), in practice even on medium-sized arrays (around 100 items or less) it might be infeasible to use anything else.

Source: *wikipedia.org*

Q4: What is algorithm? ⭐

Answer: Algorithm is a step by step procedure, which defines a set of instructions to be executed in certain order to get the desired output.

Source: *tutorialspoint.com*

Q5: What is linear data structure and what are common operations to perform on it? ⭐⭐

Answer: A *linear data-structure* has sequentially arranged data items. The next item can be located in the next memory address. It is stored and accessed in a sequential manner. **Array** and **list** are example of linear data structure.

The following operations are commonly performed on any data-structure:

- **Insertion** – adding a data item
- **Deletion** – removing a data item
- **Traversal** – accessing and/or printing all data items
- **Searching** – finding a particular data item
- **Sorting** – arranging data items in a pre-defined sequence

Source: *tutorialspoint.com*

Q6: What is an average case complexity of Bubble Sort? ⭐⭐

Answer: Bubble sort, sometimes referred to as sinking sort, is a simple sorting algorithm that repeatedly steps through the list to be sorted, compares each pair of adjacent items and swaps them if they are in the wrong order. The pass through the list is repeated until no swaps are needed, which indicates that the list is sorted.

Bubble sort has a worst-case and average complexity of `O(n2)` , where `n` is the number of items being sorted. Most practical sorting algorithms have substantially better worst-case or average complexity, often `O(n log n)` . Therefore, bubble sort is not a practical sorting algorithm.

Source: *wikipedia.org*

Q7: What examples of greedy algorithms do you know? ⭐⭐

Answer: The below given problems find their solution using greedy algorithm approach:

- Travelling Salesman Problem
- Prim's Minimal Spanning Tree Algorithm
- Kruskal's Minimal Spanning Tree Algorithm
- Dijkstra's Minimal Spanning Tree Algorithm
- Graph - Map Coloring
- Graph - Vertex Cover
- Knapsack Problem
- Job Scheduling Problem

Source: *tutorialspoint.com*

Q8: What are some examples of divide and conquer algorithms? ⭐⭐

Answer: The below given problems find their solution using divide and conquer algorithm approach:

- Merge Sort
- Quick Sort
- Binary Search
- Strassen's Matrix Multiplication
- Closest pair (points)

Source: *tutorialspoint.com*

Q9: What are some examples of dynamic programming algorithms? ⭐⭐

Answer: The below given problems find their solution using divide and conquer algorithm approach:

- Fibonacci number series
- Knapsack problem
- Tower of Hanoi
- All pair shortest path by Floyd-Warshall
- Shortest path by Dijkstra
- Project scheduling

Source: *tutorialspoint.com*

Q10: Why do we use stacks? ⭐⭐

Answer: In data-structure, **stack** is an Abstract Data Type (ADT) used to store and retrieve values in Last In First Out (LIFO) method.

Stacks follows LIFO method and addition and retrieval of a data item takes only `O(n)` time. Stacks are used where we need to access data in the reverse order or their arrival. Stacks are used commonly in recursive function calls, expression parsing, depth first traversal of graphs etc.

The below operations can be performed on a stack:

- **push()** – adds an item to stack
- **pop()** – removes the top stack item
- **peek()** – gives value of top item without removing it
- **isempty()** – checks if stack is empty
- **isfull()** – checks if stack is full

Source: *tutorialspoint.com*

Q11: Why do we use queues? ⭐⭐

Answer: Queue is an abstract data structure (ADS), somewhat similar to stack. In contrast to stack, queue is opened at both end. One end is always used to insert data (enqueue) and the other is used to remove data (dequeue). Queue follows First-In-First-Out (FIFO) methodology, i.e., the data item stored first will be accessed first.

As queues follows FIFO method, they are used when we need to work on data-items in exact sequence of their arrival. Every operating system maintains queues of various processes. Priority queues and breadth first traversal of graphs are some examples of queues.

The below operations can be performed on a queue:

- **enqueue()** – adds an item to rear of the queue
- **dequeue()** – removes the item from front of the queue
- **peek()** – gives value of front item without removing it
- **isempty()** – checks if stack is empty
- **isfull()** – checks if stack is full

Source: *tutorialspoint.com*

Q12: What is Selection Sort? ⭐⭐

Answer: Selection sort is in-place sorting technique. It divides the data set into two sub-lists: sorted and unsorted. Then it selects the minimum element from unsorted sub-list and places it into the sorted list. This iterates unless all the elements from unsorted sub-list are consumed into sorted sub-list.

Source: *tutorialspoint.com*

Q13: Why we need to do algorithm analysis? ⭐⭐

Answer: A problem can be solved in more than one ways. So, many solution algorithms can be derived for a given problem. We analyze available algorithms to find and implement the best suitable algorithm.

An algorithm are generally analyzed on two factors – time and space. That is, how much **execution** time and how much **extra space** required by the algorithm.

Source: *tutorialspoint.com*

Q14: What is the difference between Linear Search and Binary Search? ⭐⭐

Answer:

- A **linear search** looks down a list, one item at a time, without jumping. In complexity terms this is an `O(n)` search - the time taken to search the list gets bigger at the same rate as the list does.

- A **binary search** is when you start with the middle of a sorted list, and see whether that's greater than or less than the value you're looking for, which determines whether the value is in the first or second half of the list. Jump to the half way through the sublist, and compare again etc. In complexity terms this is an `O(log n)` search - the number of search operations grows more slowly than the list does, because you're halving the "search space" with each operation.

Comparing the two:

- Binary search requires the input data to be sorted; linear search doesn't
- Binary search requires an *ordering* comparison; linear search only requires equality comparisons
- Binary search has complexity `O(log n)`; linear search has complexity O(n)
- Binary search requires random access to the data; linear search only requires sequential access (this can be very important - it means a linear search can *stream* data of arbitrary size)

Source: *wikipedia.org*

Q15: What is asymptotic analysis of an algorithm? ⭐⭐

Answer: Asymptotic analysis of an algorithm, refers to defining the mathematical boundation/framing of its run-time performance. Using asymptotic analysis, we can very well conclude the best case, average case and worst case scenario of an algorithm.

Asymptotic analysis can provide three levels of mathematical binding of execution time of an algorithm:

- Best case is represented by Ω(n) notation.
- Worst case is represented by O(n) notation.
- Average case is represented by Θ(n) notation.

Source: *tutorialspoint.com*

Q16: Name some approaches to develop algorithms ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q17: What is Circular Queue and why will you use one? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q18: Why is Insertion sort better than Quick sort for small list of elements? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q19: Tell me something about Insertion sort? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q20: List some advantages of Insertion Sort ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q21: How Insertion sort and Selection sorts are different? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q22: What is Merge Sort and how it works? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q23: How Quick Sort works? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q24: What is Shell Sort? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q25: Is there ever a good reason to use Insertion Sort? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q26: Is there any advantages of Bubble Sort? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q27: What is Bucket Sort? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q28: What is Tim Sort and how would you compare it with Quick Sort? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q29: Why is Quick Sort better than Merge Sort? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q30: What is stability in sorting algorithms and why is it important? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] Design Patterns Interview Questions

Q1: What are the main categories of Design Patterns? ⭐

Answer: The Design patterns can be classified into three main categories:

- Creational Patterns
- Behavioral Patterns
- Functional Patterns

Source: *www.educba.com*

Q2: What is a pattern? ⭐

Answer: *Patterns* in programming are like recipes in cooking. They are not ready dishes, but instructions for slicing and dicing products, cooking them, serving them and so forth.

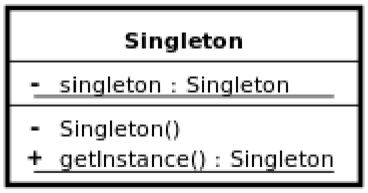
Pattern content As a rule, a pattern description consists of the following:

- a problem that the pattern solves;
- motivation for solving the the problem using the method suggested by the pattern;
- structures of classes comprising the solution;
- an example in one of the programming languages;
- a description of the nuances of pattern implementation in various contexts; relations with other patterns.

Source: *refactoring.guru*

Q3: What is Singleton pattern? ⭐

Answer: Singleton pattern comes under *creational* patterns category and introduces a single class which is responsible to create an object while making sure that only single object gets created. This class provides a way to access its only object which can be accessed directly without need to instantiate the object of the class.



Source: *refactoring.guru*

Q4: What is Design Patterns and why anyone should use them? ⭐

Answer: Design patterns are a well-described solution to the most commonly encountered problems which occur during software development.

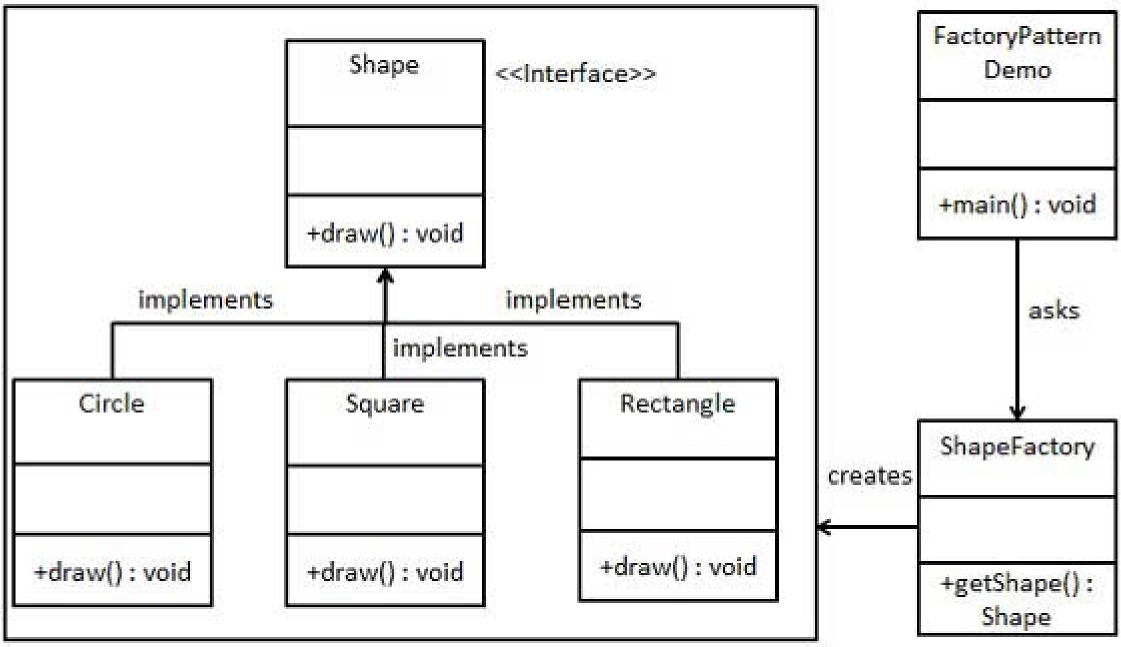
Design pattern represents the best practices evolved over a period of time by experienced software developers. They promote reusability which leads to a more robust and maintainable code.

Source: *www.educba.com*

Q5: What is Factory pattern? ⭐⭐

Answer: Factory pattern is one of most used design pattern and comes under *creational* patterns category.

In Factory pattern, we create object without exposing the creation logic to the client and refer to newly created object using a *common interface*.



Pro's:

- Allows you to hide implementation of an application seam (the core interfaces that make up your application)
- Allows you to easily test the seam of an application (that is to mock/stub) certain parts of your application so you can build and test the other parts
- Allows you to change the design of your application more readily, this is known as loose coupling

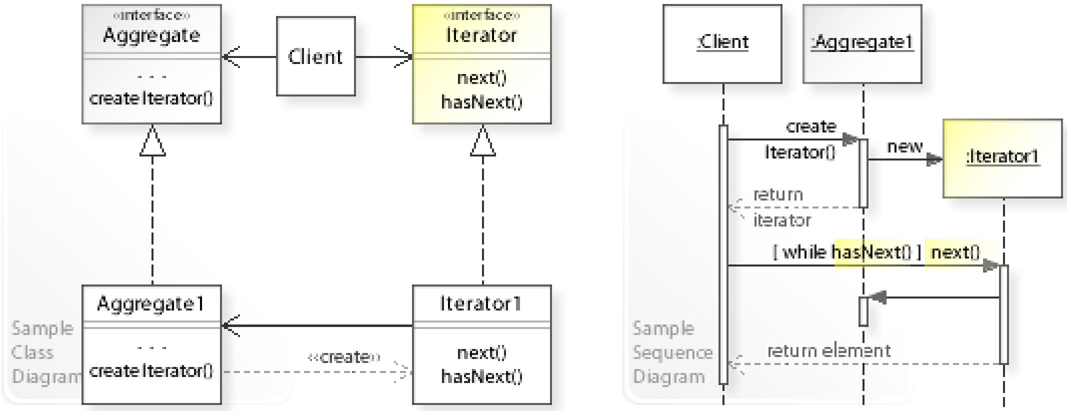
Con's

- Makes code more difficult to read as all of your code is behind an abstraction that may in turn hide abstractions.
- Can be classed as an anti-pattern when it is incorrectly used, for example some people use it to wire up a whole application when using an IOC container, instead use Dependency Injection.

Source: *tutorialspoint.com*

Q6: What is Iterator pattern? ⭐⭐

Answer: Iterator pattern is very commonly used design pattern in Java and .Net programming environment. This pattern is used to get a way to access the elements of a collection object in sequential manner without any need to know its underlying representation. Iterator pattern falls under *behavioral* pattern category.



Source: *tutorialspoint.com*

Q7: What is Inversion of Control? ⭐⭐

Answer:

Inversion of control is a broad term but for a software developer it's most commonly described as a pattern used for decoupling components and layers in the system.

For example, say your application has a text editor component and you want to provide spell checking. Your standard code would look something like this:

```
public class TextEditor {

    private SpellChecker checker;

    public TextEditor() {
        this.checker = new SpellChecker();
    }
}
```

What we've done here creates a dependency between the TextEditor and the SpellChecker. In an IoC scenario we would instead do something like this:

```
public class TextEditor {

    private IocSpellChecker checker;

    public TextEditor(IocSpellChecker checker) {
        this.checker = checker;
    }
}
```

You have *inverted control* by handing the responsibility of instantiating the spell checker from the TextEditor class to the caller.

```
SpellChecker sc = new SpellChecker; // dependency
TextEditor textEditor = new TextEditor(sc);
```

Source: *stackoverflow.com*

Q8: Can we create a clone of a singleton object? ⭐⭐

Answer: Yesl, we can but the purpose of Singleton Object creation is to have single instance serving all requests.

Source: *tutorialspoint.com*

Q9: Name types of Design Patterns? ⭐⭐

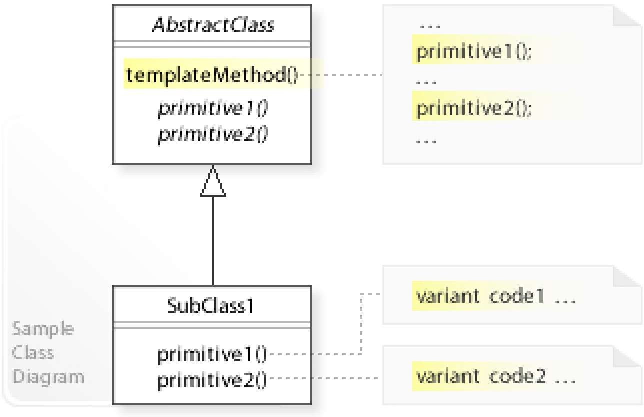
Answer: Design patterns can be classified in three categories: Creational, Structural and Behavioral patterns.

- Creational Patterns - These design patterns provide a way to create objects while hiding the creation logic, rather than instantiating objects directly using new opreator. This gives program more flexibility in deciding which objects need to be created for a given use case.
- Structural Patterns - These design patterns concern class and object composition. Concept of inheritance is used to compose interfaces and define ways to compose objects to obtain new functionalities.
- Behavioral Patterns - These design patterns are specifically concerned with communication between objects.

Source: *tutorialspoint.com*

Q10: What is Template pattern? ⭐⭐

Answer: In **Template pattern**, an abstract class exposes defined way(s)/template(s) to execute its methods. Its subclasses can override the method implementation as per need but the invocation is to be in the same way as defined by an abstract class. This pattern comes under *behavior* pattern category.



Source: *tutorialspoint.com*

Q11: What is Filter pattern? ⭐⭐

Answer: Filter pattern or **Criteria pattern** is a design pattern that enables developers to filter a set of objects using different criteria and chaining them in a decoupled way through logical operations. This type of design pattern comes under *structural* pattern as this pattern combines multiple criteria to obtain single criteria.

Filter design pattern is useful where you want to add filters dynamically or you are implementing multiple functionalities and most of them require different filter criteria to filter something. In that case instead of hard coding the filters inside the functionalities, you can create filter criteria and re-use it wherever required.

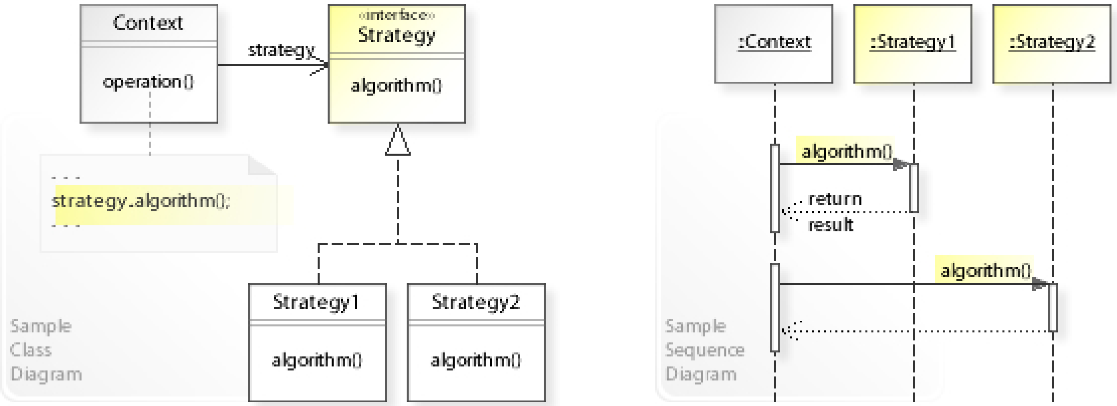
```
List<Laptop> laptops = LaptopFactory.manufactureInBulk();
AndCriteria searchCriteria = new AndCriteria(
  new HardDisk250GBFilter(),
  new MacintoshFilter(),
  new I5ProcessorFilter());
List<Laptop> filteredLaptops = searchCriteria.meets(laptops);
```

Source: *tutorialspoint.com*

Q12: What is Strategy pattern? ⭐⭐

Answer: In **Strategy pattern**, a class behavior or its algorithm can be changed at run time. This type of design pattern comes under *behavior* pattern.

In Strategy pattern, we create objects which represent various strategies and a context object whose behavior varies as per its strategy object. The strategy object changes the executing algorithm of the context object.



Source: *tutorialspoint.com*

Q13: What is Dependency Injection? ⭐⭐

Answer: *Dependency injection* makes it easy to create loosely coupled components, which typically means that components consume functionality defined by interfaces without having any first-hand knowledge of which implementation classes are being used.

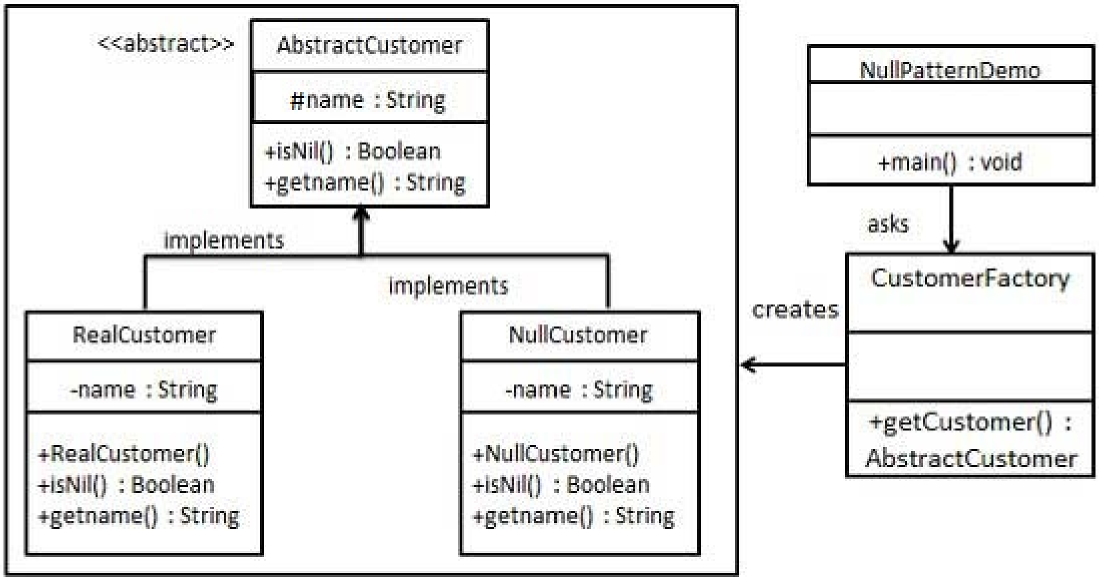
Dependency injection makes it easier to change the behavior of an application by changing the components that implement the interfaces that define application features. It also results in components that are easier to isolate for unit testing.

Source: *Pro ASP.NET Core MVC 2*

Q14: What is Null Object pattern? ⭐⭐

Answer: In **Null Object pattern**, a null object replaces check of NULL object instance. Instead of putting if check for a null value, Null Object reflects a do nothing relationship. Such Null object can also be used to provide default behaviour in case data is not available.

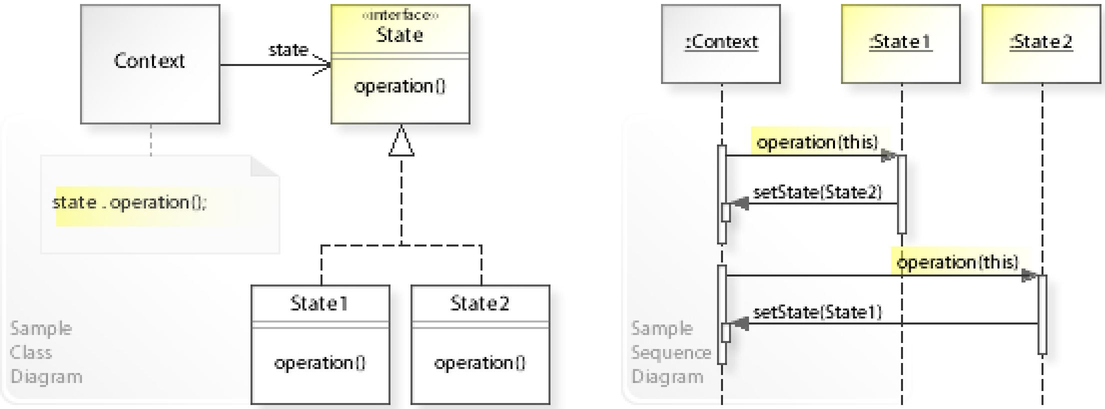
In Null Object pattern, we create an abstract class specifying various operations to be done, concrete classes extending this class and a null object class providing do nothing implementation of this class and will be used seamlessly where we need to check null value.



Source: *tutorialspoint.com*

Q15: What is State pattern? ⭐⭐

Answer: In **State pattern** a class behavior changes based on its state. This type of design pattern comes under *behavior* pattern. In State pattern, we create objects which represent various states and a context object whose behavior varies as its state object changes.

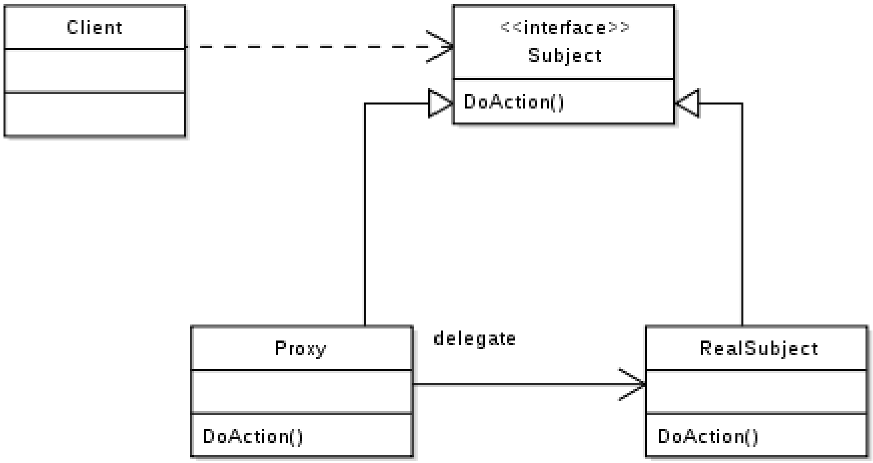


Source: *tutorialspoint.com*

Q16: What is Proxy pattern? ⭐⭐

Answer: In **proxy pattern**, a class represents functionality of another class. This type of design pattern comes under *structural* pattern.

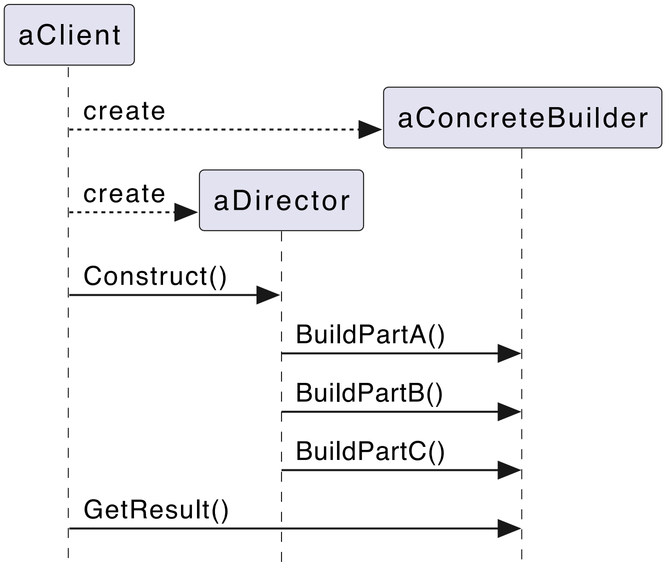
In proxy pattern, we create object having original object to interface its functionality to outer world.



Source: *tutorialspoint.com*

Q17: What is Builder pattern? ⭐⭐

Answer: *Builder pattern* builds a complex object using simple objects and using a step by step approach. This builder is independent of other objects.



The Director class is optional and is used to make sure that the building steps are executed in the *right order* with the right data by the right builder. It's about validation and delegation.

Builder/Director pattern's steps invocations could be semantically presented by *method chaining* or so called *Fluent Interface* syntax.

```
Pizza pizza = new Pizza.Builder()
                       .cheese(true)
                       .pepperoni(true)
                       .bacon(true)
                       .build();
```

Source: *tutorialspoint.com*

Q18: What are the difference between a static class and a singleton class? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q19: When should I use composite design pattern? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q20: What does “program to interfaces, not implementations” mean? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q21: What is Abstract Factory pattern? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q22: What is Decorator pattern? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q23: What is Prototype pattern? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q24: What is Memento pattern? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q25: Can you give any good explanation what is the difference between Proxy and Decorator? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q26: What is Adapter Pattern? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q27: What is Bridge pattern? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q28: What is the Chain of Responsibility pattern? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q29: What is Observer pattern? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q30: What is Command pattern? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q31: What is Interpreter pattern? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q32: What is Facade pattern? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q33: What is Mediator pattern? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q34: When would you use the Builder Pattern? Why not just use a Factory Pattern? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q35: Why would I ever use a Chain of Responsibility over a Decorator? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q36: What is Flyweight pattern? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q37: Explain usage of Service Locator Pattern ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q38: What is the difference between Strategy design pattern and State design pattern? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q39: How is Bridge pattern is different from Adapter pattern? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q40: Explain what is Composition over inheritance? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q41: Could you explain the difference between Façade vs. Mediator? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q42: Explain difference between the Facade, Proxy, Adapter and Decorator design patterns? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q43: What is the difference between the template patterns and the strategy pattern? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q44: What's the difference between the Dependency Injection and Service Locator patterns? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q45: Could you explain what is the "deadly diamond of death"? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] DevOps Interview Questions

Q1: Explain what is DevOps ? ⭐

Answer: DevOps is a newly emerging term in IT field, which is nothing but a practice that emphasizes the collaboration and communication of both software developers and other information-technology (IT) professionals. It focuses on delivering software product faster and lowering the failure rate of releases.

Source: *quora.com*

Q2: What is the most important thing DevOps helps us achieve? ⭐

Answer: The most important thing that DevOps helps us achieve is to get the changes into production as quickly as possible while minimising risks in software quality assurance and compliance. This is the primary objective of DevOps.

Source: *edureka.co*

Q3: What is meant by Continuous Integration? ⭐

Answer: *Continuous Integration (CI)* is a development practice that requires developers to integrate code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early.

Source: *edureka.co*

Q4: What is the need for DevOps? ⭐

Answer: Nowadays instead of releasing big sets of features, companies are trying to see if small features can be transported to their customers through a series of release trains. This has many advantages like quick feedback from customers, better quality of software etc. which in turn leads to high customer satisfaction. To achieve this, companies are required to:

1. Increase deployment frequency
2. Lower failure rate of new releases
3. Shortened lead time between fixes
4. Faster mean time to recovery in the event of new release crashing

DevOps fulfills all these requirements and helps in achieving seamless software delivery.

Source: *edureka.co*

Q5: Are you more Dev or Ops? ⭐

Answer: What the interview means is do you do more sysadmin work, or do you spend a lot of time working with developers on coding?

Source: *vminstall.com*

Q6: What is post mortem meetings? ⭐⭐

Answer: *Post mortem meeting* is a meeting where we discuss what went wrong and what steps should be taken so that failure doesn't happen again. Post mortem meetings are not about finding the one to be blamed, they are for preventing outages from reoccurring and planing redesign of the infrastructure so that downtime can be minimised. It is about learning from mistakes.

Source: *linoxide.com*

Q7: How is DevOps different from Agile/SDLC? ⭐⭐

Answer:

- Agile software development methodology focuses on the development of software.
- DevOps on the other hand is responsible for development as well as deployment of the software in the safest and most reliable way possible.

Source: *edureka.co*

Q8: What are the success factors for Continuous Integration? ⭐⭐

Answer:

- Maintain a code repository
- Automate the build
- Make the build self-testing
- Everyone commits to the baseline every day
- Every commit (to baseline) should be built
- Keep the build fast
- Test in a clone of the production environment
- Make it easy to get the latest deliverables
- Everyone can see the results of the latest build
- Automate deployment

Source: *edureka.co*

Q9: How have you handled failed deployments? ⭐⭐

Answer:

Source: *logz.io*

Q10: Why is Continuous monitoring necessary? ⭐⭐

Answer: *Continuous Monitoring* allows timely identification of problems or weaknesses and quick corrective action that helps reduce expenses of an organization. Continuous monitoring provides solution that addresses three operational disciplines known as:

- continuous audit
- continuous controls monitoring
- continuous transaction inspection

Source: *quora.com*

Q11: Which are the top DevOps tools? Which tools have you worked on? ⭐⭐

Answer: The most popular DevOps tools are:

- **Git**: Version Control System tool
- **Jenkins**: Continuous Integration tool
- **Selenium**: Continuous Testing tool
- **Puppet, Chef, Ansible**: Configuration Management and Deployment tools
- **Nagios**: Continuous Monitoring tool
- **Docker**: Containerization tool

Source: *edureka.co*

Q12: Mention what are the key aspects or principle behind DevOps? ⭐⭐

Answer: The key aspects or principle behind DevOps are:

- Infrastructure as code
- Continuous deployment
- Automation
- Monitoring
- Security

Source: *quora.com*

Q13: What's the next thing you would automate in your current workflow? ⭐⭐

Answer:

Source: *github.com*

Q14: Can we consider DevOps as an Agile methodology? ⭐⭐

Answer: *DevOps* is a movement to reconcile and synchronize development and production start through a set of good practices . Its emergence is motivated by a deep changing demands of business, who want to speed up the changes to stick closer to the requirements of business and the customer.

Source: *linoxide.com*

Q15: What is DevOps engineer's duty with regards to Agile development? ⭐⭐

Answer: DevOps engineer work very closely with Agile development teams to ensure they have an environment necessary to support functions such as automated testing, continuous Integration and continuous Delivery. DevOps engineer must be in constant contact with the developers and make all required parts of environment work seamlessly.

Source: *linoxide.com*

Q16: What does Containerization mean? ⭐⭐

Answer: *Containerisation* is a type of *virtualization* strategy that emerged as an alternative to traditional hypervisor-based virtualization.

In containerization, the operating system is shared by the different containers rather than cloned for each virtual machine. For example Docker provides a container virtualization platform that serves as a good alternative to hypervisor-based arrangements.

Source: *linoxide.com*

Q17: What is the function of CI (Continuous Integration) server? ⭐⭐

Answer: CI server function is to continuously integrate all changes being made and committed to repository by different developers and check for compile errors. It needs to build code several times a day, preferably after every commit so it can detect which commit made the breakage if the breakage happens.

Source: *linoxide.com*

Q18: What are the advantages of DevOps? ⭐⭐

Answer: Technical benefits:

- Continuous software delivery
- Less complex problems to fix
- Faster resolution of problems

Business benefits:

- Faster delivery of features
- More stable operating environments
- More time available to add value (rather than fix/maintain)

Source: *edureka.co*

Q19: What is the role of a configuration management tool in DevOps? ⭐⭐

Answer: *Configuration Management* tools' purpose is to automatize deployment and configuration of software on big number of servers. Most CM tools usually use agent architecture which means that every machine being manged needs to have agent installed.

One tool that uses agentless architecture is Ansible. It only requires SSH and Python. And if raw module is being used, not even Python is required because it can run raw bash commands. Other available and popular CM tools are Puppet, Chef, SaltStack.

Source: *linoxide.com*

Q20: Tell me about the worst-run/best-run outage you've been a part of. What made it bad/well-run? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q21: How would you make key aspects of a software system traceable? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q22: If something breaks in production, how do you know about it? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q23: What is the difference between resource allocation and resource provisioning? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q24: What is Chef? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q25: How do all DevOps tools work together? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q26: What are the differences between continuous integration, continuous delivery, and continuous deployment? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q27: Explain a use case for Docker ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q28: Explain Blue-Green Deployment Technique ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q29: How would you assess how “deployable” a system is? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q30: How would you prepare for a migration from one platform to another? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q31: Classify Cloud Platforms by category ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q32: What's the difference between a blue/green deployment and a rolling deployment? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q33: What do you know about serverless model? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q34: What is Vagrant and what is it used for? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q35: How is container different from a virtual machine? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q36: How would you deploy software to 5000 nodes? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q37: What is Continuous Monitoring? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q38: How would you introduce Continuous Delivery in a successful, huge company for which the change from Waterfall to Continuous Delivery would be not trivial, because of the size and complexity of the business? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q39: What is Canary Releasing? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] Docker Interview Questions

Q1: What is Docker? ⭐

Answer:

- Docker is a containerization platform which packages your application and all its dependencies together in the form of containers so as to ensure that your application works seamlessly in any environment be it development or test or production.
- Docker containers, wrap a piece of software in a complete filesystem that contains everything needed to run: code, runtime, system tools, system libraries etc. anything that can be installed on a server.
- This guarantees that the software will always run the same, regardless of its environment.

Source: *edureka.co*

Q2: What is Build Cache in Docker? ⭐⭐

Answer: When we build an Image, Docker will process each line in Dockerfile. It will execute the commands on each line in the order that is mentioned in the file. But at each line, before running any command, Docker will check if there is already an existing image in its cache that can be reused rather than creating a new image.

Source: *mindmajix.com*

Q3: What’s the difference between a repository and a registry? ⭐⭐

Answer:

- **Docker registry** is a service for hosting and distributing images (the default one is the Docker Hub).
- **Docker repository** is a collection of related Docker images (the same name but with different tags).

Source: *rafalgolarz.com*

Q4: What is the difference between ‘docker run’ and ‘docker create’? ⭐⭐

Answer: The primary difference is that using **'docker create'** creates a container in a stopped state.

Bonus point: You can use **'docker create'** and store an outputed container ID for later use. The best way to do it is to use **'docker run'** with **--cidfile FILE_NAME** as running it again won't allow to overwrite the file. A good practice is to keep well ogranised directory structure: /containers/web/server1/ws.cid containers/web/server3/ws.cid

Source: *rafalgolarz.com*

Q5: Can you remove ('docker rm') a container that is paused? ⭐⭐

Answer: No, to remove a container it must be stopped first.

Source: *rafalgolarz.com*

Q6: When would you use 'docker kill' or 'docker rm -f'? ⭐⭐

Answer: If you must stop the container really quickly… (someone pushed something to production on Friday evening?… ;))

Source: *rafalgolarz.com*

Q7: How to link containers? ⭐⭐

Answer: The simplest way is to use network port mapping. There's also the **- -link** flag which is deprecated.

Source: *rafalgolarz.com*

Q8: What is the difference between a Docker image and a container? ⭐⭐

Answer: An instance of an image is called a container. You have an image, which is a set of layers. If you start this image, you have a running container of this image. You can have many running containers of the same image.

You can see all your images with `docker images` whereas you can see your running containers with `docker ps` (and you can see all containers with `docker ps -a`).

So a running instance of an image is a container.

Source: *stackoverflow.com*

Q9: What type of applications - Stateless or Stateful are more suitable for Docker Container? ⭐⭐

Answer: It is preferable to create Stateless application for Docker Container. We can create a container out of our application and take out the configurable state parameters from application. Now we can run same container in Production as well as QA environments with different parameters. This helps in reusing the same Image in different scenarios. Also a stateless application is much easier to scale with Docker Containers than a stateful application.

Source: *mindmajix.com*

Q10: What are the most common instructions in Dockerfile? ⭐⭐

Answer:

Some of the common instructions in Dockerfile are as follows:

- **FROM**: We use FROM to set the base image for subsequent instructions. In every valid Dockerfile, FROM is the first instruction.
- **LABEL**: We use LABEL to organize our images as per project, module, licensing etc. We can also use LABEL to help in automation.
  In LABEL we specify a key value pair that can be later used for programmatically handling the Dockerfile.
- **RUN**: We use RUN command to execute any instructions in a new layer on top of the current image. With each RUN command we add something on top of the image and use it in subsequent steps in Dockerfile.
- **CMD**: We use CMD command to provide default values of an executing container. In a Dockerfile, if we include multiple CMD commands, then only the last instruction is used.

Source: *knowledgepowerhouse.com*

Q11: How to build envrionment-agnostic systems with Docker? ⭐⭐

Answer: There are three main features helping to achieve that:

- Volumes
- Environment variable injection
- Read-only file systems

Source: *rafalgolarz.com*

Q12: What is the difference between the `COPY` and `ADD` commands in a Dockerfile? ⭐⭐

Answer: Although `ADD` and `COPY` are functionally similar, generally speaking, `COPY` is preferred.

That’s because it’s more transparent than ADD. COPY only supports the basic copying of local files into the container, while ADD has some features (like local-only tar extraction and remote URL support) that are not immediately obvious. Consequently, the best use for ADD is local tar file auto-extraction into the image, as in ADD rootfs.tar.xz /.

Source: *stackoverflow.com*

Q13: What is the difference between CMD and ENTRYPOINT in a Dockerfile? ⭐⭐

Answer: Both `CMD` and `ENTRYPOINT` instructions define what command gets executed when running a container. There are few rules that describe their co-operation.

1. Dockerfile should specify at least one of `CMD` or `ENTRYPOINT` commands.
2. `ENTRYPOINT` should be defined when using the container as an executable.
3. `CMD` should be used as a way of defining default arguments for an `ENTRYPOINT` command or for executing an ad-hoc command in a container.
4. `CMD` will be overridden when running the container with alternative argumen

Source: *stackoverflow.com*

Q14: How do I transfer a Docker image from one machine to another one without using a repository, no matter private or public? ⭐⭐

Answer: You will need to save the Docker image as a tar file:

```
docker save – o <path for generated tar file> <image name>
```

Then copy your image to a new system with regular file transfer tools such as `cp` or `scp`. After that you will have to load the image into Docker:

```
docker load –i <path to image tar file>
```

Source: *stackoverflow.com*

Q15: Is there a way to identify the status of a Docker container? ⭐⭐

Answer: We can identify the status of a Docker container by running the command

```
docker ps –a
```

which will in turn list down all the available docker containers with its corresponding statuses on the host. From there we can easily identify the container of interest to check its status correspondingly.

Source: *mindmajix.com*

Q16: What is Docker image? ⭐⭐

Answer: Docker image is the source of Docker container. In other words, Docker images are used to create containers. Images are created with the build command, and they'll produce a container when started with run. Images are stored in a Docker registry such as `registry.hub.docker.com` because they can become quite large, images are designed to be composed of layers of other images, allowing a minimal amount of data to be sent when transferring images over the network.

Source: *edureka.co*

Q17: What is Docker container? ⭐⭐

Answer: Docker containers include the application and all of its dependencies, but share the kernel with other containers, running as isolated processes in user space on the host operating system. Docker containers are not tied to any specific infrastructure: they run on any computer, on any infrastructure, and in any cloud.

Source: *edureka.co*

Q18: What is Docker hub? ⭐⭐

Answer: Docker hub is a cloud-based registry service which allows you to link to code repositories, build your images and test them, stores manually pushed images, and links to Docker cloud so you can deploy images to your hosts. It provides a centralized resource for container image discovery, distribution and change management, user and team collaboration, and workflow automation throughout the development pipeline.

Source: *edureka.co*

Q19: Do I lose my data when the Docker container exits? ⭐⭐

Answer: There is no loss of data when any of your Docker containers exits as any of the data that your application writes to the disk in order to preserve it. This will be done until the container is explicitly deleted. The file system for the Docker container persists even after the Docker container is halted.

Source: *mindmajix.com*

Q20: What are the various states that a Docker container can be in at any given point in time? ⭐⭐

Answer: There are four states that a Docker container can be in, at any given point in time. Those states are as given as follows:

- Running
- Paused
- Restarting
- Exited

Source: *mindmajix.com*

Q21: What is the difference between “expose” and “publish” in Docker? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q22: Explain basic Docker usage workflow ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q23: Should I use Vagrant or Docker for creating an isolated environment? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q24: What is the difference between Docker Image and Layer? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q25: What happens if you add more than one CMD instruction to a Dockerfile? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q26: What is virtualisation? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q27: What is the difference between CMD and ENTRYPOINT in a Dockerfile? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q28: Can you create containers wihout their own PID namespace? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q29: Docker Compose vs. Dockerfile - which is better? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q30: What is Hypervisor? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q31: Could you explain what is Emulation? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q32: What is the default CPU limit set for a container? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q33: What is the purpose of EXPOSE command in Dockerfile? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q34: What is Docker Swarm? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q35: How will you monitor Docker in production? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q36: What is the preferred way of removing containers - ‘docker rm -f’ or ‘docker stop’ then followed by a ‘docker rm’? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q37: How can we control the startup order of services in Docker compose? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q38: What exactly do you mean by “Dockerized node”? Can this node be on-premises or in the cloud? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q39: How is Docker different from a virtual machine? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q40: Is it possible to generate a Dockerfile from an image? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q41: Can you explain dockerfile ONBUILD instruction? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q42: When you limit the memory for a container, does it reserve (guarantee) the memory? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q43: What is the difference between Docker RUN, CMD and ENTRYPOINT? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q44: What is an orphant volume and how to remove it? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q45: What are the different kinds of namespaces available in a Container? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q46: Is it good practice to run stateful applications on Docker? What are the scenarios where Docker best fits in? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q47: How virtualization works at low level? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q48: What is Paravirtualization? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q49: Can you run Docker containers natively on Windows? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q50: Name some limitations of containers vs VM ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q51: How does Docker run containers in non-Linux systems? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q52: How containers works at low level? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q53: Why did Docker jump from version 1.13 to 17.03? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q54: Why Docker compose does not wait for a container to be ready before moving on to start next service in dependency order? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q55: How to use Docker with multiple environments? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] Entity Framework Interview Questions

Q1: Explain what is ADO.NET entity framework? ⭐

Answer: ADO.NET Entity Framework is an ORM (Object Relational Mapping) framework developed by Microsoft. It is an extension of ADO.NET that provides an automated mechanism to access and store data in the database. With the help of ADO.NET, database can be accessed without much required programming or code.

Source: *career.guru99.com*

Q2: What are the benefits of using EF? ⭐

Answer: The main and the only benefit of EF is it auto-generates code for the Model (middle layer), Data Access Layer, and mapping code, thus reducing a lot of development time.

Source: *codeproject.com*

Q3: What is Entity Framework? ⭐

Answer: ADO.NET EF is an ORM (object-relational mapping) which creates a higher abstract object model over ADO.NET components. So rather than getting into dataset, datatables, command, and connection objects as shown in the below code, you work on higher level domain objects like customers, suppliers, etc.

Source: *codeproject.com*

Q4: What are scalar and navigation properties in Entity Framework? ⭐⭐

Answer:

- *Scalar properties* are those where actual values are contained in the entities. Normally a scalar property will map to a database field.
- *Navigation properties* help to navigate from one entity to another entity directly in the code.

Source: *codeproject.com*

Q5: What are the different ways of creating these domain / entity objects? ⭐⭐

Answer: Entity objects can be created in two ways: from a database structure, or by starting from scratch by creating a model.

Source: *codeproject.com*

Q6: What is pluralize and singularize in the Entity Framework? ⭐⭐

Answer: “Pluralize” and “Singularize” give meaningful naming conventions to objects. In simple words it says do you want to represent your objects with the below naming convention:

- One Customer record means “Customer” (singular).
- Lot of customer records means “Customer’s” (plural, watch the “s”)

Source: *codeproject.com*

Q7: What is migration in Entity Framework? ⭐⭐

Answer: Entity Framework introduced a migration tool that automatically updates the database schema when your model changes without losing any existing data or other database objects.

There are two kinds of Migration:

- Automated Migration

- Code-based Migration

Source: *entityframeworktutorial.net*

Q8: What is Code First approach in Entity Framework? ⭐⭐

Answer: In **Code First** approach we avoid working with the Visual Designer of Entity Framework. In other words the EDMX file is excluded from the solution. So you now have complete control over the context class as well as the entity classes.

Source: *codeproject.com*

Q9: How can we read records using Entity Framework classes? ⭐⭐

Answer: In order to browse through records you can create the object of the context class and inside the context class you will get the records.

For instance, in the below code snippet we are looping through a customer object collection. This customer collection is the output given by the context class `CustomermytextEntities`.

```
CustomermytestEntities obj = new CustomermytestEntities();
foreach (Customer objCust in obj.Customers)
{}
```

Source: *codeproject.com*

Q10: What is the purpose of a DBContext class? ⭐⭐

Answer: You can think of `DbContext` as the database connection and a set of tables, and `DbSet` as a representation of the tables themselves. The `DbContext` allows you to link your model properties (presumably using the Entity Framework) to your database with a connection string.

Later, when you wish to refer to a database in your controller to handle data, you reference the `DbContext`.

Source: *stackoverflow.com*

Q11: What is Mapping? ⭐⭐

Answer: The Mapping will have the information on how the Conceptual Models are mapped to Storage Models.

Source: *a4academics.com*

Q12: What is Conceptual Model? ⭐⭐

Answer: Conceptual Models are the model classes which contain the relationships. These are independent of the database design.

Source: *a4academics.com*

Q13: What is Storage Model? ⭐⭐

Answer: Storage Models are our database design models, which contains database tables, views, stored procs and keys with relationships.

Source: *a4academics.com*

Q14: Mention in what all scenarios Entity Framework can be applicable? ⭐⭐

Answer: Entity Framework can be applicable in three scenarios

- If you have an existing database already or you want to build your database first than other parts of the application
- If your prime focus is your domain classes and then create the database from your domain classes
- If you want to design your database schema on the visual designer and create the classes and database

Source: *career.guru99.com*

Q15: Mention what is Code First approach and Model First Approach in Entity Framework? ⭐⭐

Answer: In Entity Framework,

- **Model First Approach:** In this approach we create entities, relationships directly on the design surface of EDMX.
- **Code Approach:** For code approach we avoid working with the visual designer or entity framework.

Source: *career.guru99.com*

Q16: What are the components of Entity Framework Architecture? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q17: What are the advantages of Model First Approach? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q18: What are the advantages and disadvantages of Database First Approach? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q19: How can we handle concurrency in Entity Framework? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q20: What are the different approaches supported in the Entity Framework to create Entity Model? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q21: What is EF Data Access Architecture? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q22: What is the importance of EDMX file in Entity Framework? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q23: Can you explain lazy loading in a detailed manner? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q24: What is the role of Entity Client Data Provider? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q25: What is eager loading? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q26: What are POCO classes in Entity Framework? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q27: How do I delete multiple rows in Entity Framework (without foreach)? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q28: What are complex types in Entity Framework? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q29: Could you explain the difference between Optimistic vs Pessimistic locking? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q30: What are the advantages/disadvantages of Code First Approach? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q31: Explain Lazy loading, Eager Loading, and Explicit Loading? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q32: Explain how you can load related entities in EF? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q33: What is Optimistic locking? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q34: What’s the difference between LINQ to SQL and Entity Framework? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q35: Name some differences between Express vs Recoverable messages ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q36: What is faster - ADO.NET or ADO.NET Entity Framework? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q37: Is DbContext thread safe? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q38: What are the disadvantages of using static DbContext? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q39: Can you explain CSDL, SSDL and MSL sections in an EDMX file? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q40: What is the difference between ObjectContext and DbContext? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q41: What is the difference between Code First, Model First and Database First? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q42: How can you enhance the performance of Entity Framework? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q43: Which type of loading is good in which scenario? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q44: When would you use EF6 vs EF Core? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q45: What is the difference between POCO, Code First, and simple EF approach? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q46: What type of system generated messages do you know? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q47: What are T4 templates? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q48: Could you explain pessimistic locking? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q49: What is the difference between Automatic Migration vs Code-base Migration? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q50: How can we do pessimistic locking in Entity Framework? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q51: How do I view the SQL generated by the Entity Framework? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q52: What difference does .AsNoTracking() make? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q53: What's the difference between .SaveChanges() and .AcceptAllChanges()? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q54: When would you use SaveChanges(false) + AcceptAllChanges()? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q55: What are the advantages and disadvantages of creating a global entities context for the application (i.e. one static instance)? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q56: What is client wins and store wins mode in Entity Framework concurrency? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q57: Can I use Entity Framework 6 (not core) in .Net Core? ⭐⭐⭐⭐⭐

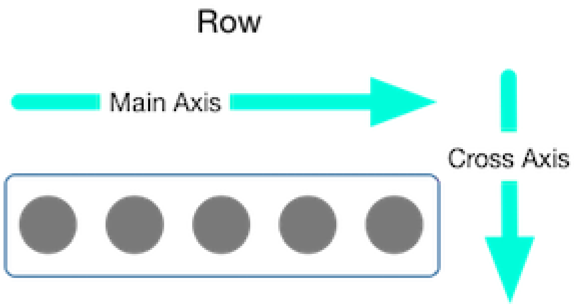
Read answer on 👉 FullStack.Cafe

[⬆️] Flutter Interview Questions

Q1: When to use main Axis Alignment and cross Axis Alignment? ⭐

Answer: For Row:

`mainAxisAlignment` = Horizontal Axis
`crossAxisAlignment` = Vertical Axis



For Column:

`mainAxisAlignment` = Vertical Axis
`crossAxisAlignment` = Horizontal Axis

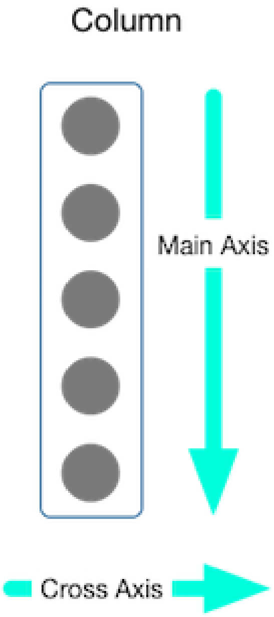


Image source

Source: *stackoverflow.com*

Q2: What is Flutter? ⭐

Answer: Flutter is an open-source UI toolkit from *Google* for crafting beautiful, natively compiled applications for desktop, web, and mobile from a single codebase. Flutter apps are built using the *Dart* programming language.

Source: *flutter.dev*

Q3: What is the pubspec.yaml file and what does it do? ⭐⭐

Answer:

- The `pubspec.yaml` file allows you to define the packages your app relies on, declare your assets like images, audio, video, etc.
- It allows you to set constraints for your app.
- For Android developers, this is roughly similar to a `build.gradle` file.

Source: *medium.com*

Q4: When should you use WidgetsBindingObserver? ⭐⭐

Answer: WidgetsBindingObserver should be used when we want to listen to the `AppLifecycleState` and call stop/start on our services.

Source: *www.filledstacks.com*

Q5: What is the difference between "main()" and "runApp()" functions in Flutter? ⭐⭐

Answer:

- `main ()` function came from *Java*-like languages so it's where all program started, without it, you can't write any program on Flutter even without UI.
- `runApp()` function should return *Widget* that would be attached to the screen as a root of the *Widget Tree* that will be rendered.

Source: *stackoverflow.com*

Q6: What is the difference between Expanded and Flexible widgets? ⭐⭐

Answer: `Expanded` is just a shorthand for `Flexible`

Using expanded this way:

```
Expanded(
        child: Foo(),
);
```

is strictly equivalent to:

```
Flexible(
        fit: FlexFit.tight,
        child: Foo(),
);
```

You may want to use `Flexible` over `Expanded` when you want a different `fit` , useful in some responsive layouts.

The difference between `FlexFit.tight` and `FlexFit.loose` is that loose will allow its child to have a maximum size while tight forces that child to fill all the available space.

Source: *stackoverflow.com*

Q7: How is Flutter different from a WebView based application? ⭐⭐

Answer:

- Code you write for a **WebView** or an app that runs similarly has to go through multiple layers to finally get executed (like Cordova for Ionic).** In essence, Flutter leapfrogs that by **compiling down to native **ARM** code to execute on both platforms.
- "Hybrid" apps are slow, sluggish and look different from the platform they run on. *Flutter* apps run much, much faster than their hybrid counterparts.
- It's much easier to access native components and sensors using plugins rather than using **WebView** which can't take full use of their platform.

Source: *medium.com*

Q8: What is the pubspec.yaml file and what does it do? ⭐⭐

Answer: The `Pubspec.yaml` allows you to define the packages your app relies on, declare your assets like images, audio, video, etc. It also allows you to set constraints for your app. For Android developers, this is roughly similar to a `build.gradle` file, but the differences between the two are also evident.

Source: *medium.com*

Q9: What is a "widget" and mention its importance in Flutter? ⭐⭐

Answer:

- Widgets are basically the UI components in Flutter.
- It is a way to describe the configuration of an *Element*.
- They are inspired from components in **React**.

Widgets are important in Flutter because everything within a Flutter application is a **Widget** , from a simple *"Text"* to *"Buttons"* to *"Screen Layouts"*.

Source: *stackoverflow.com*

Q10: What is Dart and why does Flutter use it? ⭐⭐

Answer: Dart is an *object-oriented*, *garbage-collected* programming language that you use to develop Flutter apps. It was also created by Google, but is open-source, and has community inside and outside Google. Dart was chosen as the language of **Flutter** for the following reason:

- Dart is **AOT** (Ahead Of Time) compiled to fast, predictable, native code, which allows almost all of Flutter to be written in Dart. This not only makes Flutter fast, virtually everything (including all the widgets) can be customized.
- Dart can also be **JIT** (Just In Time) compiled for exceptionally fast development cycles and game-changing workflow (including Flutter's popular sub-second stateful hot reload).
- Dart allows Flutter to avoid the need for a separate declarative layout language like *JSX* or *XML*, or separate visual interface builders, because Dart's declarative, programmatic layout is easy to read and visualize. And with all the layout in one language and in one place, it is easy for Flutter to provide advanced tooling that makes layout a snap.

Source: *hackernoon.com*

Q11: What is an App state? ⭐⭐

Answer:

- State that is not *ephemeral*, that you want to share across many parts of your app, and that you want to keep between user sessions, is what we call **application state** (sometimes also called *shared state*).
- Examples of application state:
  - User preferences
  - Login info
  - Notifications in a social networking app
  - The shopping cart in an e-commerce app
  - Read/unread state of articles in a news app

Source: *flutter.dev*

Q12: How many types of widgets are there in Flutter? ⭐⭐

Answer: There are two types of widgets:

1. **StatelessWidget**: A widget that does not require mutable state.

2. **StatefulWidget**: A widget that has mutable state.

Source: *proandroiddev.com*

Q13: What are the different build modes in Flutter? ⭐⭐

Answer:

- The Flutter tooling supports three modes when compiling your app, and a headless mode for testing.
- You choose a compilation mode depending on where you are in the development cycle.
- The modes are:
  - Debug
  - Profile
  - Release

Source: *flutter.dev*

Q14: What is Fat Arrow Notation in Dart and when do you use it? ⭐⭐

Answer: The fat arrow syntax is simply a short hand for returning an expression and is similar to `(){ return expression; }`.

The fat arrow is for returning a single line, braces are for returning a code block.

Only an expression—not a statement—can appear between the arrow (=>) and the semicolon (`;`). For example, you can’t put an *if* statement there, but you can use a *conditional* expression

```
// Normal function
void function1(int a) {
  if (a == 3) {
    print('arg was 3');
  } else {
    print('arg was not 3');
  }
}

// Arrow Function
void function2(int a) => print('arg was ${a == 3 ? '' : 'not '}3');
```

Source: *stackoverflow.com*

Q15: Does Flutter work like a browser? How is it different from a WebView based application? ⭐⭐

Answer: To answer this question simply: **Code you write for a WebView or an app that runs similarly has to go through multiple layers to finally get executed.** In essence, Flutter leapfrogs that by **compiling down to native ARM** code to execute on both platforms. “Hybrid” apps are slow, sluggish and look different from the platform they run on. Flutter apps run much, much faster than their hybrid counterparts. Also, it’s much easier to access native components and sensors using plugins rather than using WebViews which can’t take full use of their platform.

Source: *medium.com*

Q16: Do you know what Ephemeral state means? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q17: What is Streams in Flutter/Dart? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q18: Explain the different types of Streams? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q19: Why is the build() method on State and not Stateful Widget? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q20: Differentiate StatelessWidget and StatefulWidget? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q21: What are packages and plugins in Flutter? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q22: What are keys in Flutter and when to use it? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q23: Explain Navigator Widget and its push pop functions in Flutter? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q24: When do we use double.INFINITY? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q25: How to get difference of lists in Flutter/Dart? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q26: Differentiate between required and optional parameters in Dart ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q27: What are Null-aware operators? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q28: What is debug mode and when do you use it? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q29: What is profile mode and when do you use it? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q30: What is release mode and when do you use it? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q31: Differentiate between named parameters and positional parameters in Dart? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q32: How would you execute code only in debug mode? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q33: What is the difference between Scaffold and Container in Flutter? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q34: What is ScopedModel / BLoC Pattern? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q35: How is InheritedWidget different from Provider? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q36: What are some pros of Flutter? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q37: Name some cons of using Flutter? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q38: Where are the layout files? Why doesn't Flutter have layout files? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q39: Why do we pass functions to widgets? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q40: Differentiate between Hot Restart and Hot Reload? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q41: How do you check if an async void method is completed in Dart? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q42: How to declare async function as a variable in Dart? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q43: How to duplicate repeating items inside a Dart list? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q44: How is whenCompleted() different from then() in Future? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q45: When would you use App state or Ephemeral state over another? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q46: What is Future in Flutter/Dart? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q47: What is the purpose of SafeArea in Flutter? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q48: How do you convert a List into a Map in Dart? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q49: What is a difference between these operators "?? and ?." ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q50: What is the difference between React Native and Flutter in-depth? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q51: How does Dart AOT work? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q52: What are some pros and cons of Scoped Model vs BLoC and vice versa? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q53: What are Global Keys? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q54: What is a MediaQuery in Flutter and when do we use it? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q55: Why is exit(0) not preferred for closing an app? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q56: What's the difference between async and async* in Dart? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q57: What is the difference between double.INFINITY and MediaQuery? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q58: What are the similarities and differences of Future and Stream? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q59: Explain async, await in Flutter/Dart? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q60: How to compare two dates that are constructed differently in Dart? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q61: What does "non-nullable by default" mean in Dart? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q62: What does a class with a method named ._() mean in Dart/Flutter? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q63: Why should you use kReleaseMode instead of assert? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q64: How is AnimationController different from Timer? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q65: What is the difference between debug mode and profile mode? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q66: Why Are StatefulWidget and State Separate Classes? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q67: List some approaches for State management in Flutter ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q68: Explain Stateful Widget Lifecycle in details ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] Git Interview Questions

Q1: What is the command to write a commit message in Git? ⭐

Answer: Use:

```
git commit -a
```

-a on the command line instructs git to commit the new content of all tracked files that have been modified. You can use

```
git add <file>
```

or

```
git add -A
```

before git commit -a if new files need to be committed for the first time.

Source: *edureka.co*

Q2: What is difference between Git vs SVN? ⭐

Answer: The main point in Git vs SVN debate boils down to this: Git is a distributed version control system (DVCS), whereas SVN is a centralized version control system.

Source: *medium.com*

Q3: What is Git? ⭐

Answer: Git is a Distributed Version Control system (DVCS). It can track changes to a file and allows you to revert back to any particular change.

Its distributed architecture provides many advantages over other Version Control Systems (VCS) like SVN one major advantage is that it does not rely on a central server to store all the versions of a project’s files.

Source: *edureka.co*

Q4: What's the difference between a "pull request" and a "branch"? ⭐⭐

Answer:

- A **branch** is just a separate version of the code.

- A **pull request** is when someone take the repository, makes their own branch, does some changes, then tries to merge that branch in (put their changes in the other person's code repository).

Source: *stackoverflow.com*

Q5: What is Git fork? What is difference between fork, branch and clone? ⭐⭐

Answer:

- A **fork** is a remote, server-side copy of a repository, distinct from the original. A fork isn't a Git concept really, it's more a political/social idea.
- A **clone** is not a fork; a clone is a local copy of some remote repository. When you clone, you are actually copying the entire source repository, including all the history and branches.
- A **branch** is a mechanism to handle the changes within a single repository in order to eventually merge them with the rest of code. A branch is something that is within a repository. Conceptually, it represents a thread of development.

Source: *stackoverflow.com*

Q6: What is the difference between "git pull" and "git fetch"? ⭐⭐

Answer: In the simplest terms, `git pull` does a `git fetch` followed by a `git merge` .

- When you use `pull` , Git tries to automatically do your work for you. **It is context sensitive**, so Git will merge any pulled commits into the branch you are currently working in. `pull` **automatically merges the commits without letting you review them first**. If you don’t closely manage your branches, you may run into frequent conflicts.

- When you `fetch` , Git gathers any commits from the target branch that do not exist in your current branch and **stores them in your local repository**. However, **it does not merge them with your current branch**. This is particularly useful if you need to keep your repository up to date, but are working on something that might break if you update your files. To integrate the commits into your master branch, you use `merge` .

Source: *stackoverflow.com*

Q7: How does the Centralized Workflow work? ⭐⭐

Answer: The **Centralized Workflow** uses a central repository to serve as the single point-of-entry for all changes to the project. The default development branch is called master and all changes are committed into this branch.

Developers start by cloning the central repository. In their own local copies of the project, they edit files and commit changes. These new commits are stored locally.

To publish changes to the official project, developers *push* their local master branch to the central repository. Before the developer can publish their feature, they need to *fetch* the updated central commits and rebase their changes on top of them.

Compared to other workflows, the Centralized Workflow has no defined pull request or forking patterns.

Source: *atlassian.com*

Q8: You need to update your local repos. What git commands will you use? ⭐⭐

Answer: It’s a two steps process. First you fetch the changes from a remote named origin:

```
git fetch origin
```

Then you merge a branch master to local:

```
git merge origin/master
```

Or simply:

```
git pull origin master
```

If origin is a default remote and ‘master’ is default branch, you can drop it eg. `git pull` .

Source: *samwize.com*

Q9: How to undo the most recent commits in Git? ⭐⭐

Details: You accidentally committed wrong files to Git, but haven't pushed the commit to the server yet. How can you undo those commits from the local repository?

Answer:

```
$ git commit -m "Something terribly misguided"
$ git reset HEAD~                                 # copied the old head to
<< edit files as necessary >>
$ git add ...
$ git commit -c ORIG_HEAD                         # will open an editor, wh
```

Source: *stackoverflow.com*

Q10: Tell me the difference between HEAD, working tree and index, in Git? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q11: When should I use "git stash"? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q12: Explain the advantages of Forking Workflow ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q13: Could you explain the Gitflow workflow? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q14: You need to rollback to a previous commit and don't care about recent changes. What commands should you use? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q15: How to revert previous commit in git? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q16: What is "git cherry-pick"? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q17: What is a "bare git" repository? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q18: When would you use "git clone --bare" over "git clone --mirror" ? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q19: What is the "HEAD" in Git? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q20: How to remove a file from git without removing it from your file system? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q21: Can you explain what “git reset” does in plain english? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q22: Write down a sequence of git commands for a "Rebase Workflow" ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q23: What is difference between "git stash pop" and "git stash apply"? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q24: How do you make an existing repository bare? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q25: What is the difference between "git clone", "git clone --bare" and "git clone --mirror"? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q26: When would you use "git clone" over "git clone --bare" ? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q27: When would you use "git clone" over "git clone --mirror" ? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q28: Write down a git command to check difference between two commits ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q29: When do you use "git rebase" instead of "git merge"? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q30: Do you know how to easily undo a git rebase? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q31: How to amend older Git commit? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q32: What git command do you need to use to know who changed certain lines in a specific file? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q33: What are "git hooks"? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q34: What are the type of git hooks? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q35: What is "git bisect"? How can you use it to determine the source of a (regression) bug? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q36: How can you use "git bisect" to determine the source of a (regression) bug? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] Golang Interview Questions

Q1: What is Go? ⭐

Answer: Go is a general-purpose language designed with systems programming in mind. It was initially developed at Google in year 2007 by Robert Griesemer, Rob Pike, and Ken Thompson. It is strongly and statically typed, provides inbuilt support for garbage collection and supports concurrent programming. Programs are constructed using packages, for efficient management of dependencies. Go programming implementations use a traditional compile and link model to generate executable binaries.

Source: *tutorialspoint.com*

Q2: Is Go a new language, framework or library? ⭐

Answer: Go isn't a library and not a framework, it's a new language.

Go is mostly in the C family (basic syntax), with significant input from the Pascal/Modula/Oberon family (declarations, packages). Go does have an extensive library, called the runtime, that is part of every Go program. Although it is more central to the language, Go's runtime is analogous to libc, the C library. It is important to understand, however, that Go's runtime does not include a virtual machine, such as is provided by the Java runtime. Go programs are compiled ahead of time to native machine code.

Source: *golang.org*

Q3: Can you declared multiple types of variables in single declaration in Go? ⭐⭐

Answer: Yes. Variables of different types can be declared in one go using type inference.

```
var a, b, c =  3,  4,  "foo"
```

Source: *tutorialspoint.com*

Q4: What is a pointer? ⭐⭐

Answer: A **pointer variable** can hold the *address* of a variable.

Consider:

```
var x =  5  var p *int p =  &x
fmt.Printf("x = %d",  *p)
```

Here `x` can be accessed by `*p`.

Source: *tutorialspoint.com*

Q5: Can you return multiple values from a function? ⭐⭐

Answer: A Go function can return multiple values.

Consider:

```
package main
import "fmt"
```

```
func swap(x, y string) (string, string) {
   return y, x
}
func main() {
   a, b := swap("Mahesh", "Kumar")
   fmt.Println(a, b)
}
```

Source: *tutorialspoint.com*

Q6: What are some advantages of using Go? ⭐⭐

Answer: Go is an attempt to introduce a new, concurrent, garbage-collected language with fast compilation and the following benefits:

- It is possible to compile a large Go program in a few seconds on a single computer.
- Go provides a model for software construction that makes dependency analysis easy and avoids much of the overhead of C-style include files and libraries.
- Go's type system has no hierarchy, so no time is spent defining the relationships between types. Also, although Go has static types, the language attempts to make types feel lighter weight than in typical OO languages.
- Go is fully garbage-collected and provides fundamental support for concurrent execution and communication.
- By its design, Go proposes an approach for the construction of system software on multicore machines.

Source: *golang.org*

Q7: Why the Go language was created? ⭐⭐

Answer: Go was born out of frustration with existing languages and environments for systems programming.

Go is an attempt to have:

- an interpreted, dynamically typed language with
- the efficiency and safety of a statically typed, compiled language
- support for networked and multicore computing
- be fast in compilation

To meet these goals required addressing a number of linguistic issues: an expressive but lightweight type system; concurrency and garbage collection; rigid dependency specification; and so on. These cannot be addressed well by libraries or tools so a new language was born.

Source: *golang.org*

Q8: Explain this code ⭐⭐

Details: In Go there are various ways to return a struct value or slice thereof. Could you explain the difference?

```go
type MyStruct struct {
    Val int
}

func myfunc() MyStruct {
    return MyStruct{Val: 1}
}

func myfunc() *MyStruct {
    return &MyStruct{}
}

func myfunc(s *MyStruct) {
    s.Val = 1
}
```

Answer: Shortly:

- the first returns a copy of the struct,
- the second a pointer to the struct value created within the function,
- the third expects an existing struct to be passed in and overrides the value.

Source: *stackoverflow.com*

Q9: What is dynamic type declaration of a variable in Go? ⭐⭐

Answer: A *dynamic type variable declaration* requires compiler to interpret the type of variable based on value passed to it. Compiler don't need a variable to have type statically as a necessary requirement.

Source: *tutorialspoint.com*

Q10: What are Goroutines? ⭐⭐

Answer: Goroutines are functions or methods that run concurrently with other functions or methods. Goroutines can be thought of as light weight threads. The cost of creating a Goroutine is tiny when compared to a thread. Its common for Go applications to have thousands of Goroutines running concurrently.

Source: *golangbot.com*

Q11: Let's talk variable declaration in Go. Could you explain what is a variable "zero value"? ⭐⭐

Answer: Variable is the name given to a memory location to store a value of a specific type. There are various syntaxes to declare variables in go.

```
// 1 - variable declaration, then assignment
var age int
age = 29

// 2 - variable declaration with initial value
var age2 int = 29

// 3 - Type inference
var age3 = 29

// 4 - declaring multiple variables
var width, height int = 100, 50

// 5 - declare variables belonging to different types in a single statement
var (
      name1 = initialvalue1,
      name2 = initialvalue2
)
// 6 - short hand declaration
name, age4 := "naveen", 29 //short hand declaration
```

If a variable is not assigned any value, go automatically initialises it with the **zero value of the variable's type**. Go is strongly typed, so variables declared as belonging to one type cannot be assigned a value of another type.

Source: *golangbot.com*

Q12: What kind of type conversion is supported by Go? ⭐⭐

Answer: Go is very strict about **explicit typing**. There is no automatic type promotion or conversion. Explicit type conversion is required to assign a variable of one type to another.

Consider:

```
i := 55      //int
j := 67.8    //float64
sum := i + int(j) //j is converted to int
```

Source: *golangbot.com*

Q13: What is static type declaration of a variable in Go? ⭐⭐

Answer: *Static type variable declaration* provides assurance to the compiler that there is one variable existing with the given type and name so that compiler proceed for further compilation without needing complete detail about the variable. A variable declaration has its meaning at the time of compilation only, compiler needs actual variable declaration at the time of linking of the program.

Source: *tutorialspoint.com*

Q14: How to efficiently concatenate strings in Go? ⭐⭐

Details: In Go, a `string` is a primitive type, which means it is read-only, and every manipulation of it will create a new string.

So if I want to concatenate strings many times without knowing the length of the resulting string, what's the best way to do it?

Answer: Beginning with Go 1.10 there is a `strings.Builder`. A Builder is used to efficiently build a string using Write methods. It minimizes memory copying. The zero value is ready to use.

```
package main

import (
    "strings"
    "fmt"
)

func main() {
    var str strings.Builder
    for i := 0; i < 1000; i++ {
        str.WriteString("a")
    }

    fmt.Println(str.String())
}
```

Source: *stackoverflow.com*

Q15: What are the benefits of using Go programming? ⭐⭐

Answer: Following are the benefits of using Go programming:

- Support for environment adopting patterns similar to dynamic languages. For example type inference (`x := 0` is valid declaration of a variable `x` of type `int`).
- Compilation time is fast.
- In built concurrency support: light-weight processes (via goroutines), channels, select statement.
- Conciseness, Simplicity, and Safety.
- Support for Interfaces and Type embedding.
- The go compiler supports static linking. All the go code can be statically linked into one big fat binary and it can be deployed in cloud servers easily without worrying about dependencies.

Source: *tutorialspoint.com*

Q16: Does Go have exceptions? ⭐⭐

Answer: No, Go takes a different approach. For plain error handling, Go's **multi-value returns** make it easy to report an error without overloading the return value. Go code uses error values to indicate an abnormal state.

Consider:

```
func Open(name string) (file *File, err error)
```

```
f, err := os.Open("filename.ext")
if err != nil {
    log.Fatal(err)
}
// do something with the open *File f
```

Source: *golang.org*

Q17: What is "rune" type in Go? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q18: Name some advantages of Goroutines over threads ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q19: How to initialise a struct in Go? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q20: Is Go an object-oriented language? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q21: How to check if a map contains a key in Go? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q22: Is there a foreach construct in the Go language? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q23: Implement a function that reverses a slice of integers ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q24: Can Go have optional parameters? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q25: Have you worked with Go 2? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q26: What is the preferred way to handle configuration parameters for a Go program? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q27: What is the difference between the = and := operator? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q28: What would you do if you need a hash displayed in a fixed order? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q29: What is the difference between C.sleep() and time.Sleep()? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q30: What are the differences between unbuffered and buffered channels? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q31: How to copy map in Go? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q32: Why would you prefer to use an empty struct{}? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q33: How do you swap two values? Provide a few examples. ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q34: What is so special about constants in Go? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q35: How does Go compile so quickly? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q36: What might be wrong with the following small program? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q37: How to find a type of an object in Go? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q38: When is the init() function run? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q39: Briefly describe how GC works in GO? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q40: What is the difference, if any, in the following two slice declarations, and which one is more preferable? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q41: List the functions can stop or suspend the execution of current goroutine, and explain their differences. ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q42: What is $GOROOT and $GOPATH? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q43: What is the idiomatic Go equivalent of C's ternary operator? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q44: What are the use(s) for tags in Go? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q45: How can I check if two slices are equal? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q46: What is an idiomatic way of representing enums in Go? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q47: What is the malloc threshold of Map object? How to modify it? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q48: How to compare two interfaces in Go? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q49: When go runtime allocates memory from heap, and when from stack? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] GraphQL Interview Questions

Q1: Is GraphQL a Database Technology? ⭐

Answer: No. GraphQL is often confused with being a database technology. This is a misconception, GraphQL is a *query language* for APIs - not databases. In that sense it's database agnostic and can be used with any kind of database or even no database at all.

Source: *howtographql.com*

Q2: What is GraphQL? ⭐

Answer: GraphQL is a query language created by Facebook in 2012 which provides a **common interface between the client and the server for data fetching and manipulations**.

The client asks for various data from the GraphQL server via queries. The response format is described in the query and defined by the client instead of the server: they are called **client-specified queries**.
The structure of the data is not hardcoded as in traditional REST APIs - this makes retrieving data from the server more efficient for the client.

Source: *howtographql.com*

Q3: Is GraphQL only for React / Javascript Developers? ⭐

Answer: No. GraphQL is an API technology so it can be used in any context where an API is required.

On the *backend*, a GraphQL server can be implemented in any programming language that can be used to build a web server. Next to Javascript, there are popular reference implementations for Ruby, Python, Scala, Java, Clojure, Go and .NET.

Since a GraphQL API is usually operated over HTTP, any client that can speak HTTP is able to query data from a GraphQL server.

Note: GraphQL is actually transport layer agnostic, so you could choose other protocols than HTTP to implement your server.

Source: *howtographql.com*

Q4: What is an exclamation point in GraphQL? ⭐

Answer: That means that the field is non-nullable. By default, all types in GraphQL are nullable. When non-null is applied to the type of a field, it means that if the server resolves that field to `null`, the response will *fail validation*.

Source: *stackoverflow.com*

Q5: How to do Error Handling? ⭐⭐

Answer: A successful GraphQL query is supposed to return a JSON object with a root field called `"data"`. If the request fails or partially fails (e.g. because the user requesting the data doesn’t have the right access permissions), a second root field called `"errors"` is added to the response:

```
{
  "data": { ... },
  "errors": [ ... ]
}
```

Source: *howtographql.com*

Q6: Where is GraphQL useful? ⭐⭐

Answer: GraphQL helps where your **client needs a flexible response** format to avoid extra queries and/or massive data transformation with the overhead of keeping them in sync.

Using a GraphQL server makes it very easy for a client side developer to change the response format without any change on the backend.

With GraphQL, you can describe the required data in a more natural way. It can speed up development, because in application structures like **top-down rendering** in React, the required data is more similar to your component structure.

Source: *blog.risingstack.com*

Q7: What is difference between Mutation and Query? ⭐⭐

Answer: Technically any GraphQL *query* could be implemented to cause a data write. But there is a convention that any operations that cause writes should be sent explicitly via a *mutation*.

Besides the difference in the semantic, there is one important technical difference:

Query fields can be executed in parallel by the GraphQL engine while Mutation top-level fields MUST execute serially according to the spec.

Source: *stackoverflow.com*

Q8: What is GraphQL schema? ⭐⭐

Answer: Every GraphQL server has two core parts that determine how it works: a schema and resolve functions.

The *schema* is a model of the data that can be fetched through the GraphQL server. It defines what queries clients are allowed to make, what types of data can be fetched from the server, and what the relationships between these types are.

Consider:

```
type Author {
  id: Int
  name: String
  posts: [Post]
}
type Post {
  id: Int
  title: String
  text: String
  author: Author
}
type Query {
  getAuthor(id: Int): Author
  getPostsByTitle(titleContains: String): [Post]
}
schema {
  query: Query
}
```

Source: *dev-blog.apollodata.com*

Q9: How to do Authentication and Authorization? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q10: What kind of operations could GraphQL schema have? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q11: Explain the main difference between REST and GraphQL ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q12: Does GraphQL Support Offline Usage? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q13: How to do Server-side Caching? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q14: How to query all the GraphQL type fields without writing a long query? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q15: List the key concepts of the GraphQL query language ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q16: Can you make a GraphQL type both an input and output type? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q17: Are there any disadvantages to GraphQL? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q18: How to implement a set of GraphQL mutations in single transaction? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q19: How do you prevent nested attack on GraphQL server? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q20: Is it possible to use inheritance with GraphQL input types? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q21: What is AST in GraphQL? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q22: How to respond with different status code in GraphQL? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q23: What the criteria set is for deciding when to use GraphQL vs. HATEOAS? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q24: How would you model recursive data structures in GraphQL? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] HTML5 Interview Questions

Q1: What is an iframe and how it works? ⭐

Answer: An **iframe** is an **HTML document** which can be embedded inside another HTML page.

Example:

```
<iframe src="https://github.com" height="300px" width="300px"></iframe>
```

Source: *github.com/FuelFrontend*

Q2: Explain meta tags in HTML ⭐

Answer:

- **Meta tags** always go inside **head tag** of the HTML page
- **Meta tags** is always passed as name/value pairs
- **Meta tags** are not displayed on the page but intended for the browser
- **Meta tags** can contain information about **character encoding**, **description**, **title** of the document etc,

Example:

```
<!DOCTYPE html>
<html>
<head>
  <meta name="description" content="I am a web page with description">
  <title>Home Page</title>
</head>
<body>
```

```
</body>
</html>
```

Source: *github.com/FuelFrontend*

Q3: What is the purpose of the alt attribute on images? ⭐

Answer: The `alt` attribute provides alternative information for an image if a user cannot view it. The `alt` attribute should be used to describe any images except those which only serve a decorative purposes, in which case it should be left empty.

Source: *developer.mozilla.org*

Q4: Write a HTML table tag sequence that outputs the following ⭐

Answer: Write a HTML table tag sequence that outputs the following:

```
50 pcs 100 500
10 pcs 5 50
```

Answer:

```
<table>
  <tr>
    <td>50 pcs</td>
    <td>100</td>
    <td>500</td>
  </tr>
  <tr>
    <td>10 pcs</td>
    <td>5</td>
    <td>50</td>
  </tr>
</table>
```

Source: *toptal.com*

Q5: What were some of the key goals and motivations for the HTML5 specification? ⭐⭐

Answer: HTML5 was designed to replace HTML 4, XHTML, and the HTML DOM Level 2. The key goals and motivations behind the HTML5 specification were to:

- Deliver rich content (graphics, movies, etc.) without the need for additional plugins, such as Flash.
- Provide better semantic support for web page structure through new structural element tags.
- Provide a stricter parsing standard to simplify error handling, ensure more consistent cross-browser behaviour, and simplify compatibility with documents written to older standards.
- Provide better cross-platform support whether running on a PC, Tablet, or Smartphone.

Source: *toptal.com*

Q6: hat's the difference between an "attribute" and a "property" in HTML? ⭐⭐

Answer: Attributes are defined on the HTML markup but properties are defined on the DOM. To illustrate the difference, imagine we have this text field in our HTML: `<input type="text" value="Hello">`.

```
const input = document.querySelector('input');
console.log(input.getAttribute('value')); // Hello
console.log(input.value); // Hello
```

But after you change the value of the text field by adding "World!" to it, this becomes:

```
console.log(input.getAttribute('value')); // Hello
console.log(input.value); // Hello World!
```

Source: *github.com/yangshun*

Q7: Briefly describe the correct usage of the following HTML5 semantic elements: , , , ⭐⭐

Answer:

- `<header>` is used to contain introductory and navigational information about a section of the page. This can include the section heading, the author's name, time and date of publication, table of contents, or other navigational information.

- `<article>` is meant to house a self-contained composition that can logically be independently recreated outside of the page without losing it's meaining. Individual blog posts or news stories are good examples.

- `<section>` is a flexible container for holding content that shares a common informational theme or purpose.
- `<footer>` is used to hold information that should appear at the end of a section of content and contain additional information about the section. Author's name, copyright information, and related links are typical examples of such content.

Source: *w3schools.com*

Q8: How Can I Get Indexed Better by Search Engines? ⭐⭐

Answer: It is possible to get indexed better by placing the following two statements in the `<HEAD>` part of your documents:

```
<META NAME="keywords" CONTENT="keyword keyword keyword keyword">
<META NAME="description" CONTENT="description of your site">
```

Both may contain up to 1022 characters. If a keyword is used more than 7 times, the keywords tag will be ignored altogether. Also, you cannot put markup (other than entities) in the description or keywords list.

Source: *freejavaguide.com*

Q9: What is Character Encoding? ⭐⭐

Answer: To display an HTML page correctly, a web browser must know which character set (character encoding) to use. This is specified in the tag:

HTML4:

```
<meta http-equiv="Content-Type" content="text/html;charset=ISO-8859-1">
```

HTML5:

```
<meta charset="UTF-8">
```

Source: *w3schools.com*

Q10: What is the difference between span and div? ⭐⭐

Answer:

- `div` is a block element
- `span` is inline element

For bonus points, you could point out that it's illegal to place a block element inside an inline element, and that while `div` can have a `p` tag, and a `p` tag can have a `span` , it is not possible for `span` to have a `div` or `p` tag inside.

Source: *thatjsdude.com*

Q11: What is a self closing tag? ⭐⭐

Answer: In HTML5 it is not strictly necessary to close certain HTML tags. The tags that aren't required to have specific closing tags are called "self closing" tags.

An example of a self closing tag is something like a line break (`<br />`) or the meta tag (`<meta>`). This means that the following are both acceptable:

```
<meta charset="UTF-8">
...
<meta charset="UTF-8" />
```

Source: *blog.teamtreehouse.com*

Q12: How can you highlight text in HTML? ⭐⭐

Answer: If you are working with an HTML5 page, the `<mark>` tag can be a quick and easy way of highlighting or marking text on a page:

```
<mark>highlighted text</mark>
```

To highlight text with just HTML code and support for all browsers, set the background-color style, as shown in the example below, using the HTML tag.

```
<span style="background-color: #FFFF00">Yellow text.</span>
```

Source: *computerhope.com*

Q13: Can a web page contain multiple elements? What about elements? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q14: What are `data-` attributes good for? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q15: Have you used different HTML templating languages before? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q16: How do you change the direction of html text? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q17: What is WebSQL? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q18: What is an optional tag? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q19: How do you serve a page with content in multiple languages? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q20: What is the difference between
and
? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q21: What does a DOCTYPE do? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q22: Explain almost standard, full standard and quirks mode ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q23: When is it appropriate to use the small element? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q24: What is the purpose of cache busting and how can you achieve it? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q25: Describe the difference between a 'cookie', 'sessionStorage' and 'localStorage'. ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q26: What are `defer` and `async` attributes on a `<script>` tag? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q27: What is the DOM? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q28: Discuss the differences between an HTML specification and a browser's implementation thereof. ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q29: What are some differences that XHTML has compared to HTML? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q30: Where and why is the `rel="noopener"` attribute used? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q31: What is HTML5 Web Storage? Explain `localStorage` and `sessionStorage`. ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q32: What is WebSQL? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q33: Explain the difference between block elements and inline elements ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q34: How do you set IE compatibility mode? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q35: What's new in HTML 5? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q36: Explain the difference between cookies, session and local storage ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q37: What are Web Workers? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q38: Describe the difference between <script>, <script async> and <script defer>. ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q39: Why to use HTML5 semantic tags? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q40: What is WebP? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q41: What kind of things must you be wary of when designing or developing for multilingual sites? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q42: What is progressive rendering? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q43: What are the building blocks of HTML5? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q44: What's the difference between Full Standard, Almost Standard and Quirks Mode? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q45: Why you would use a `srcset` attribute in an image tag? Explain the process the browser uses when evaluating the content of this attribute. ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q46: HTML Markup Validity ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q47: How would you select svg or canvas for your site? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q48: What is an HTML preprocessor and are you using it? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q49: Why do I need a doctype and what does it do? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q50: What is the purpose of 'main' element? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q51: What are Web Components? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q52: What is accessibility & ARIA role means in a web application? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q53: Could you generate a public key in HTML? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q54: Why is it generally a good idea to position CSS s between and JS <script>s just before ? Do you know any exceptions? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q55: What is an IndexedDB? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] Ionic Interview Questions

Q1: What is Ionic Framework? ⭐

Answer: Ionic Framework is an open source UI toolkit for building performant, high-quality mobile and desktop apps using web technologies (HTML, CSS, and JavaScript). Ionic Framework is focused on the frontend user experience, or UI interaction of an app (controls, interactions, gestures, animations). Currently, Ionic Framework has official integrations with Angular and React, and support for Vue is in development.

Source: *ionicframework.com*

Q2: How can you test Ionic applications? ⭐⭐

Answer: Ionic v.1 applications are built using AngularJS. Angular has a rich set of test libraries and frameworks such as Jasmine and Karma test runner. These frameworks can be used to write unit tests for Ionic applications. Also, `ionic-CLI` provides `live reload` feature so the application can be tested in the browser. For example, the `ionic serve` command can be used to load the application in any browser. Thus, we can use Chrome Developer Tools or Mozilla Firefox with Firebug to debug and inspect Ionic applications.

Source: *toptal.com*

Q3: What is hybrid app development? ⭐⭐

Answer: Hybrid apps are developed using web technologies like HTML, CSS and Javascript, and then wrapped in a native application using platforms like Cordova. The apps are shown in its own embedded browser, like UIWebView in iOS and WebView in Android (not Safari or Chrome). This allows you to use any web-native framework for mobile app development.

Source: *netguru.com*

Q4: Can we work with Ionic > 1 and AngularJS? ⭐⭐

Answer: Unfortunately, **no**. Ionic (1) at a very high-level is essentially just a wrapper & directive/component library for AngularJS (1). In that same regard, Ionic 2 is built in the same way, utilizing all the benefits of Angular 2+.

Ionic 2 breaks away from being tied to the DOM in the browser, by using angular 2 which is the reason for the massive change between ionic 1.x and ionic 2.x. (Angular 2.x architecture is not tied down to the DOM unlike the Angular 1.x).

Source: *toptal.com*

Q5: How do you pass data from one view to another in Ionic applications? ⭐⭐

Answer: Ionic v.1 uses AngularJS and UI-router. It means you can use Angular services or UI-router's state `resolve` to pass data from one view to another. Since Angular services are singletons, data stored in services can be accessed across other Angular controllers.

As mentioned, UI-router provides a `resolve` configuration. For example:

```
$stateProvider
  .state('todos', {
    url: '/todos',
    controller: 'TodosCtrl',
    templateUrl: 'todos.html',
    resolve: {
      todos: function(TodosService) {
        return TodosService.getTodos()
      }
    }
  })
```

One advantage of `resolve` over stateful services is better testing: as `resolve` injects dependencies in the controller, it is easy to test them.

When using Ionic v.4 you have 3 options:

1. Using Query Params (bad)
2. Service and Resolve Function (legit)
3. Using Router Extras State (new since Angular 7.2)

```
openDetailsWithState() {
    let navigationExtras: NavigationExtras = {
      state: {
        user: this.user
      }
    };
    this.router.navigate(['details'], navigationExtras);
  }
```

Source: *toptal.com*

Q6: What is the difference between Cordova and Ionic? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q7: How would you compare Ionic vs Flutter? When would you choose one? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q8: How do you persist data between application launches using Ionic? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q9: What are some possible security issues with Ionic apps? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q10: What is the advantage of caching the views in Ionic apps? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q11: How can you detect a platform (Android or iOS) at runtime in Ionic application? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q12: How can you access mobile phone native functionality in Ionic applications, for example the camera? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q13: What is the difference between PhoneGap, Cordova, and Ionic? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q14: What’s the difference between “ionic build” and “ionic prepare”? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q15: What are the most prominent advantages and disadvantages of building applications using the Ionic framework? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q16: How many Types of Storage Available in Ionic Framework? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q17: What is Capacitor? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q18: How to use observables in the Ionic framework? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q19: What is Ionic Native? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q20: What is the difference between Capacitor and Cordova? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q21: What does it mean that Ionic became framework-agnostic? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q22: How can you render a 5000 item list in Ionic, without affecting scroll performance? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q23: How is the Ionic Framework v4 different from v3? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q24: If more than one component is trying to make an HTTP call to same URL, then how can you restrict making 2 network calls? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q25: What are the new features in Ionic 4? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q26: What Are The Ionic Lifecycle Events? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q27: What AOT and JIT and which is used by Ionic? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q28: Performance of Ionic application is bad on older Android devices. Why is this, and what can be done to improve it? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q29: What are some security measures should be made for Ionic app? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] JSON Interview Questions

Q1: What is JSON and why would I use it? ⭐

Answer: JSON (JavaScript Object Notation) is a lightweight format that is used for data interchanging. It is based on a subset of JavaScript language (the way objects are built in JavaScript). Some JavaScript is not JSON, and some JSON is not JavaScript.

Source: *stackoverflow.com*

Q2: What is the correct JSON content type? ⭐⭐

Answer: The MIME media type for JSON text is **application/json**. The default encoding is UTF-8. (Source: RFC 4627).

Source: *stackoverflow.com*

Q3: How should I parse a JSON string in JavaScript? ⭐⭐

Answer: The standard way to parse JSON in JavaScript is `JSON.parse()`

The JSON API was introduced with ES5 (2011) and has since been implemented in >99% of browsers by market share, and Node.js. Its usage is simple:

```
const json = '{ "fruit": "pineapple", "fingers": 10 }';
const obj = JSON.parse(json);
console.log(obj.fruit, obj.fingers);
```

Source: *stackoverflow.com*

Q4: What are the limitations and uses of JSON? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q5: Can I use comments inside a JSON file? If so, how? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q6: Why must one use JSON over XML? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q7: Are Javascript objects and JSON equivalent? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q8: Explain the structure of JSON ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q9: What are the differences between JSON and JSONP? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q10: Explain the difference between JSON.stringify() and JSON.parse() ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q11: What is the difference between YAML and JSON? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q12: Which data format is the right one for JSON? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q13: Is there a standard on JSON naming? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q14: What is JSONP, and why was it created? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q15: How could I parse a JSON string in older browser? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] Java Interview Questions

Q1: What is JVM? Why is Java called the “Platform Independent Programming Language”? ⭐

Answer: A Java virtual machine (JVM) is a process virtual machine that can execute Java bytecode. Each Java source file is compiled into a bytecode file, which is executed by the JVM. Java was designed to allow application programs to be built that could be run on any platform, without having to be rewritten or recompiled by the programmer for each separate platform. A Java virtual machine makes this possible, because it is aware of the specific instruction lengths and other particularities of the underlying hardware platform.

Source: *github.com/snowdream*

Q2: What is a Servlet? ⭐

Answer: The servlet is a Java programming language class used to process client requests and generate dynamic web content. Servlets are mostly used to process or store data submitted by an HTML form, provide dynamic content and manage state information that does not exist in the stateless HTTP protocol.

Source: *github.com/snowdream*

Q3: What is the Difference between JDK and JRE? ⭐

Answer:

- **The Java Runtime Environment (JRE) **is basically the Java Virtual Machine (**JVM**) where your Java programs are being executed. It also includes browser plugins for applet execution.

- **The Java Development Kit (JDK)** is the full featured Software Development Kit for Java, including the JRE, the compilers and tools (like JavaDoc, and Java Debugger), in order for a user to develop, compile and execute Java applications.

Source: *github.com/snowdream*

Q4: What is a JSP Page? ⭐

Answer: A **Java Server Page (JSP)** is a text document that contains two types of text:

- static data and
- JSP elements.

Static data can be expressed in any text-based format, such as HTML or XML. JSP is a technology that mixes static content with dynamically-generated content.

Source: *github.com/snowdream*

Q5: What is the difference between an Applet and a Java Application? ⭐

Answer:

- Applets are executed within a Java enabled browser, but a
- Java application is a standalone Java program that can be executed outside of a browser.

However, they both require the existence of a Java Virtual Machine (JVM). Furthermore, a Java application requires a main method with a specific signature, in order to start its execution. Java applets don't need such a method to start their execution. Finally, Java applets typically use a restrictive security policy, while Java applications usually use more relaxed security policies.

Source: *github.com/snowdream*

Q6: What are the two types of Exceptions in Java? Which are the differences between them? ⭐

Answer: Java has two types of exceptions: checked exceptions and unchecked exceptions.

1. **Unchecked exceptions **do not need to be declared in a method or a constructor's throws clause, if they can be thrown by the execution of the method or the constructor, and propagate outside the method or constructor boundary.

2. On the other hand, **checked exceptions** must be declared in a method or a constructor's throws clause.

Source: *github.com/snowdream*

Q7: How do I efficiently iterate over each entry in a Java Map? ⭐⭐

Answer: Consider:

```
Map<String, String> map = ...
for (Map.Entry<String, String> entry : map.entrySet()) {
    System.out.println(entry.getKey() + "/" + entry.getValue());
}
```

In Java 8 you can do it clean and fast using the new lambdas features:

```
final long[] i = {0};
map.forEach((k, v) -> i[0] += k + v);
```

Source: *stackoverflow.com*

Q8: Explain Serialization and Deserialization. ⭐⭐

Answer: Java provides a mechanism, called object serialization where an object can be represented as a sequence of bytes and includes the object's data, as well as information about the object's type, and the types of data stored in the object. Thus, serialization can be seen as a way of flattening objects, in order to be stored on disk, and later, read back and reconstituted. Deserialisation is the reverse process of converting an object from its flattened state to a live object.

Source: *github.com/snowdream*

Q9: What is the difference between an Interface and an Abstract class? ⭐⭐

Answer: Java provides and supports the creation both of **abstract** classes and **interfaces**. Both implementations share some common characteristics, but they differ in the following features:

- All methods in an interface are implicitly abstract. On the other hand, an abstract class may contain both abstract and non-abstract methods.
- A class may implement a number of Interfaces, but can extend only one abstract class.
- In order for a class to implement an interface, it must implement all its declared methods. However, a class may not implement all declared methods of an abstract class. Though, in this case, the sub-class must also be declared as abstract.
- Abstract classes can implement interfaces without even providing the implementation of interface methods.
- Variables declared in a Java interface is by default final. An abstract class may contain non-final variables.
- Members of a Java interface are public by default. A member of an abstract class can either be private, protected or public.
- An interface is absolutely abstract and cannot be instantiated. An abstract class also cannot be instantiated, but can be invoked if it contains a main method.

Source: *github.com/snowdream*

Q10: What are pass by reference and pass by value? ⭐⭐

Answer:

- When an object is **passed by value**, this means that a copy of the object is passed. Thus, even if changes are made to that object, it doesn't affect the original value.
- When an object is **passed by reference**, this means that the actual object is not passed, rather a reference of the object is passed. Thus, any changes made by the external method, are also reflected in all places.

Source: *github.com/snowdream*

Q11: What is the difference between processes and threads? ⭐⭐

Answer: The main difference between them is that

- a **Process** is a program which is executing some code and
- a **Thread** is an independent path of execution in the process.

A process can have more than one thread for doing independent task e.g. a thread for reading data from disk, a thread for processing that data and another thread for sending that data over the network.

Source: *github.com/snowdream*

Q12: What's the difference between sendRedirect and forward methods? ⭐⭐

Answer: The sendRedirect method creates a new request, while the forward method just forwards a request to a new target. The previous request scope objects are not available after a redirect, because it results in a new request. On the other hand, the previous request scope objects are available after forwarding. FInally, in general, the sendRedirect method is considered to be slower compare to the forward method.

Source: *github.com/snowdream*

Q13: Explain the architechure of a Servlet. ⭐⭐

Answer: The core abstraction that must be implemented by all servlets is the javax.servlet.Servlet interface. Each servlet must implement it either directly or indirectly, either by extending javax.servlet.GenericServlet or javax.servlet.http.HTTPServlet. Finally, each servlet is able to serve multiple requests in parallel using multithreading.

Source: *github.com/snowdream*

Q14: What are JSP actions? ⭐⭐

Answer: JSP actions use constructs in XML syntax to control the behavior of the servlet engine. JSP actions are executed when a JSP page is requested. They can be dynamically inserted into a file, re-use JavaBeans components, forward the user to another page, or generate HTML for the Java plugin.Some of the available actions are listed below:

- jsp:include – includes a file, when the JSP page is requested.
- jsp:useBean – finds or instantiates a JavaBean.
- jsp:setProperty – sets the property of a JavaBean.
- jsp:getProperty – gets the property of a JavaBean.
- jsp:forward – forwards the requester to a new page.
- jsp:plugin – generates browser-specific code.

Source: *github.com/snowdream*

Q15: What are Expressions? ⭐⭐

Answer: A JSP expression is used to insert the value of a scripting language expression, converted into a string, into the data stream returned to the client, by the web server. Expressions are defined between <% = and %> tags.

Source: *github.com/snowdream*

Q16: What are Decalarations? ⭐⭐

Answer: Declarations are similar to variable declarations in Java. Declarations are used to declare variables for subsequent use in expressions or scriptlets. To add a declaration, you must use the sequences to enclose your declarations.

Source: *github.com/snowdream*

Q17: What does the “static” keyword mean? Can you override private or static method in Java? ⭐⭐

Answer: The `static` keyword denotes that a member variable or method can be accessed, *without requiring an instantiation of the class to which it belongs*.

A user cannot override static methods in Java, because method overriding is based upon dynamic binding at runtime and static methods are statically binded at compile time. A static method is not associated with any instance of a class so the concept is not applicable.

Source: *github.com/snowdream*

Q18: What are the basic interfaces of Java Collections Framework? ⭐⭐

Answer: Java Collections Framework provides a well designed set of interfaces and classes that support operations on a collections of objects. The most basic interfaces that reside in the Java Collections Framework are:

- **Collection**, which represents a group of objects known as its elements.
- **Set**, which is a collection that cannot contain duplicate elements.
- **List**, which is an ordered collection and can contain duplicate elements.
- **Map**, which is an object that maps keys to values and cannot contain duplicate keys.

Source: *github.com/snowdream*

Q19: What are Directives? ⭐⭐

Answer: What are the different types of Directives available in JSP ? Directives are instructions that are processed by the JSP engine, when the page is compiled to a servlet. Directives are used to set page-level instructions, insert data from external files, and specify custom tag libraries. Directives are defined between < %@ and % >.The different types of directives are shown below:

- Include directive: it is used to include a file and merges the content of the file with the current page.
- Page directive: it is used to define specific attributes in the JSP page, like error page and buffer.
- Taglib: it is used to declare a custom tag library which is used in the page.

Source: *github.com/snowdream*

Q20: What is an Iterator? ⭐⭐

Answer: The Iterator interface provides a number of methods that are able to iterate over any Collection. Each Java Collection contains the Iterator method that returns an Iterator instance. Iterators are capable of removing elements from the underlying collection during the iteration.

Source: *github.com/snowdream*

Q21: How are the JSP requests handled? ⭐⭐

Answer: On the arrival of a JSP request, the browser first requests a page with a .jsp extension. Then, the Web server reads the request and using the JSP compiler, the Web server converts the JSP page into a servlet class. Notice that the JSP file is compiled only on the first request of the page, or if the JSP file has changed.The generated servlet class is invoked, in order to handle the browser's request. Once the execution of the request is over, the servlet sends a response back to the client. See how to get Request parameters in a JSP.

Source: *github.com/snowdream*

Q22: What is Function Overriding and Overloading in Java? ⭐⭐

Answer:

- Method **overloading** in Java occurs when two or more methods in the same class have the exact same name, but different parameters.

```
class Dog{
    public void bark(){
        System.out.println("woof ");
    }

    //overloading method
    public void bark(int num){
        for(int i=0; i<num; i++)
                System.out.println("woof ");
    }
}
```

- On the other hand, method **overriding** is defined as the case when a child class redefines the same method as a parent class. Overridden methods must have the same name, argument list, and return type. The overriding method may not limit the access of the method it overrides.

```
class Dog{
    public void bark(){
        System.out.println("woof ");
    }
}
class Hound extends Dog{
    public void sniff(){
        System.out.println("sniff ");
    }
    public void bark(){
        System.out.println("bowl");
    }
}

public class OverridingTest{
    public static void main(String [] args){
        Dog dog = new Hound();
        dog.bark();
    }
}
```

Source: *github.com/snowdream*

Q23: How HashMap works in Java? ⭐⭐

Answer: A HashMap in Java stores key-value pairs. The HashMap requires a hash function and uses hashCode and equals methods, in order to put and retrieve elements to and from the collection respectively. When the put method is invoked, the HashMap calculates the hash value of the key and stores the pair in the appropriate index inside the collection. If the key exists, its value is updated with the new value. Some important characteristics of a HashMap are its capacity, its load factor and the threshold resizing.

Source: *github.com/snowdream*

Q24: What differences exist between HashMap and Hashtable? ⭐⭐

Answer: There are several differences between `HashMap` and `Hashtable` in Java:

1. `Hashtable` is synchronized, whereas `HashMap` is not. This makes `HashMap` better for non-threaded applications, as unsynchronized Objects typically perform better than synchronized ones.

2. `Hashtable` does not allow `null` keys or values. `HashMap` allows one `null` key and any number of `null` values.

3. One of HashMap's subclasses is `LinkedHashMap` , so in the event that you'd want predictable iteration order (which is insertion order by default), you could easily swap out the `HashMap` for a `LinkedHashMap` . This wouldn't be as easy if you were using `Hashtable` .

Source: *stackoverflow.com*

Q25: What is the purpose Class.forName method? ⭐⭐

Answer: This method is used to method is used to load the driver that will establish a connection to the database.

Source: *github.com/snowdream*

Q26: What is JDBC? ⭐⭐

Answer: JDBC is an abstraction layer that allows users to choose between databases. JDBC enables developers to write database applications in Java, without having to concern themselves with the underlying details of a particular database.

Source: *github.com/snowdream*

Q27: What is the design pattern that Java uses for all Swing components? ⭐⭐

Answer: The design pattern used by Java for all Swing components is the Model View Controller (MVC) pattern.

Source: *github.com/snowdream*

Q28: How does Garbage Collection prevent a Java application from going out of memory? ⭐⭐

Answer: It doesn't! Garbage Collection simply cleans up unused memory when an object goes out of scope and is no longer needed. However an application could create a huge number of large objects that causes an `OutOfMemoryError` .

Source: *codementor.io*

Q29: What do you know about the big-O notation and can you give some examples with respect to different data structures? ⭐⭐

Answer: The Big-O notation simply describes how well an algorithm scales or performs in the worst case scenario as the number of elements in a data structure increases. The Big-O notation can also be used to describe other behavior such as memory consumption. Since the collection classes are actually data structures, we usually use the Big-O notation to chose the best implementation to use, based on time, memory and performance. Big-O notation can give a good indication about performance for large amounts of data.

Source: *github.com/snowdream*

Q30: What are the Data Types supported by Java? What is Autoboxing and Unboxing? ⭐⭐

Answer: The eight primitive data types supported by the Java programming language are:

- byte
- short
- int
- long
- float
- double
- boolean
- char

Autoboxing is the automatic conversion made by the Java compiler between the primitive types and their corresponding object wrapper classes. If the conversion goes the other way, this operation is called **unboxing**.

Source: *github.com/snowdream*

Q31: What is an Java Applet? ⭐⭐

Answer: A Java Applet is program that can be included in a HTML page and be executed in a java enabled client browser. Applets are used for creating dynamic and interactive web applications.

Source: *github.com/snowdream*

Q32: What will happen to the Exception object after exception handling? ⭐⭐

Answer: The Exception object will be garbage collected in the next garbage collection.

Source: *github.com/snowdream*

Q33: What is the importance of finally block in exception handling? ⭐⭐

Answer: A *finally* block will always be executed, whether or not an exception is actually thrown. Even in the case where the catch statement is missing and an exception is thrown, the finally block will still be executed. Last thing to mention is that the finally block is used to release resources like I/O buffers, database connections, etc.

Source: *github.com/snowdream*

Q34: What is the purpose of garbage collection in Java, and when is it used? ⭐⭐

Answer: The purpose of garbage collection is to identify and discard those objects that are no longer needed by the application, in order for the resources to be reclaimed and reused.

Source: *github.com/snowdream*

Q35: What does System.gc() and Runtime.gc() methods do? ⭐⭐

Answer: These methods can be used as a hint to the JVM, in order to start a garbage collection. However, this it is up to the Java Virtual Machine (JVM) to start the garbage collection immediately or later in time.

Source: *github.com/snowdream*

Q36: What is the difference between Exception and Error in java? ⭐⭐

Answer:

- An **Error** "indicates serious problems that a reasonable application should not try to catch."
- An **Exception** "indicates conditions that a reasonable application might want to catch."

Source: *github.com/snowdream*

Q37: What is reflection and why is it useful? ⭐⭐

Answer: The name **reflection** is used to describe code which is able to inspect other code in the same system (or itself) and to make modifications at runtime.

For example, say you have an object of an unknown type in Java, and you would like to call a 'doSomething' method on it if one exists. Java's static typing system isn't really designed to support this unless the object conforms to a known interface, but using reflection, your code can look at the object and find out if it has a method called 'doSomething' and then call it if you want to.

```
Method method = foo.getClass().getMethod("doSomething", null);
method.invoke(foo, null);
```

Source: *stackoverflow.com*

Q38: When does an Object becomes eligible for Garbage collection in Java ? ⭐⭐

Answer: A Java object is subject to garbage collection when it becomes unreachable to the program in which it is currently used.

Source: *github.com/snowdream*

Q39: Compare the sleep() and wait() methods in Java ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q40: Is there anything like static class in java? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q41: What is the importance of hashCode() and equals() methods? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q42: If an object reference is set to null, will the Garbage Collector immediately free the memory held by that object? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q43: When is the finalize() called? What is the purpose of finalization? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q44: What is the difference between throw and throws? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q45: What is the Java Classloader? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q46: What’s the difference between Enumeration and Iterator interfaces? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q47: How does finally block differ from finalize() method? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q48: Explain the life cycle of an Applet. ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q49: What happens when an applet is loaded? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q50: What is the tradeoff between using an unordered array versus an ordered array? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q51: What are the restrictions imposed on Java applets? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q52: What are untrusted applets? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q53: What is the applet security manager, and what does it provide? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q54: Can an enum be extended? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q55: Which Swing methods are thread-safe? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q56: What is the relationship between an event-listener interface and an event-adapter class? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q57: What’s the difference between a ClassNotFoundException and NoClassDefFoundError? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q58: What is Java Priority Queue? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q59: What is Comparable and Comparator interface? List their differences. ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q60: What is difference between ArrayList and LinkedList ? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q61: Explain the role of Driver in JDBC. ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q62: What is difference between Array and ArrayList ? When will you use Array over ArrayList? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q63: What is the advantage of PreparedStatement over Statement? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q64: What is the use of CallableStatement? Name the method, which is used to prepare a CallableStatement. ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q65: What is difference between fail-fast and fail-safe? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q66: What differences exist between Iterator and ListIterator? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q67: What are the advantages of JSP? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q68: Why Collection doesn’t extend Cloneable and Serializable interfaces? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q69: What are Scriptlets? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q70: What’s a deadlock? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q71: What is structure of Java Heap? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q72: What is meant by JSP implicit objects and what are they? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q73: What is the difference between an Applet and a Servlet? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q74: What is the difference between GenericServlet and HttpServlet? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q75: Explain the life cycle of a Servlet. ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q76: What is the difference between doGet() and doPost()? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q77: What is the difference between final, finalize and finally? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q78: What is a Server Side Include (SSI)? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q79: Explain different ways of creating a thread. Which one would you prefer and why? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q80: How and where are Annotations used in Java? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q81: What are the steps involved to make work a RMI program? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q82: What is the role of stub in RMI? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q83: Does Java support multiple inheritance? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q84: Is Java “pass-by-reference” or “pass-by-value”? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q85: How do I read/convert an InputStream into a String in Java? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q86: What is the volatile keyword useful for? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q87: What is a Constructor, Constructor Overloading in Java and Copy-Constructor? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q88: What is the difference between public, protected, package-private and private in Java? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q89: How threadsafe is enum in Java? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q90: What is the JIT? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q91: Can you access non static variable in static context? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q92: What do the ... dots in the method parameters mean? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q93: How can I synchornize two Java processes? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q94: How do I break out of nested loops in Java? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q95: What is a JavaBean exactly? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q96: Can == be used on enum? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q97: What are the differences between == and equals? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q98: What is the main difference between StringBuffer and StringBuilder? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q99: What's the advantage of using getters and setters? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q100: Why does Java have transient fields? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q101: What is static initializer? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q102: What is the difference between HashMap, LinkedHashMap and TreeMap in Java? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q103: What is the role of Remote Interface in RMI? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q104: What is the difference between applets loaded over the internet and applets loaded via the file system? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q105: What is the applet class loader, and what does it provide? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q106: What are the differences between a HashMap and a Hashtable in Java? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q107: What are some of the best practices relating to the Java Collection framework? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q108: What is Servlet Chaining? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q109: How do you find out what client machine is making a request to your servlet? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q110: When to use LinkedList over ArrayList in Java? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q111: What is the difference between a synchronized method and a synchronized block? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q112: What is Double Brace initialization in Java? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q113: What is RMI? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q114: How do I test a private function or a class that has private methods, fields or inner classes? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q115: What is the basic principle of RMI architecture? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q116: Explain the available thread states in a high-level. ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q117: Is it possible to call one constructor from another in Java? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q118: What is the difference between Serial and Throughput Garbage collector? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q119: What is the role of the java.rmi.Naming Class? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q120: What is the main difference between an inner class and a static nested class in Java? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q121: What is meant by binding in RMI? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q122: Given two double values d1, d2, what is the most reliable way to test their equality? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q123: How do you ensure that N threads can access N resources without deadlock? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q124: Does Java support default parameter values? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q125: Explain a use case for the Builder Design Pattern ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q126: What exactly is marker interface in Java? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q127: What is the purpose of using RMISecurityManager in RMI? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q128: Explain Marshalling and demarshalling. ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q129: Is null check needed before calling instanceof? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q130: What is Perm Gen space in Heap? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q131: Why is Spring MVC better than Servlets / JSP ? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q132: Why is char[] preferred over String for passwords? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q133: What does Connection pooling mean? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q134: What is an efficient way to implement a singleton pattern in Java? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q135: What's the difference between SoftReference and WeakReference in Java? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q136: Provide some examples when a finally block won't be executed in Java? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q137: Why isn’t String‘s .length() accurate? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q138: What's wrong with Double Brace initialization in Java? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q139: Why ArrayList are preferable in many more use-cases than LinkedList? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q140: Explain what will the code return ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q141: Compare volatile vs static variables in Java ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q142: What is DGC ? And how does it work? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q143: What are the layers of RMI Architecture? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q144: How does thread synchronization occurs inside a monitor? What levels of synchronization can you apply? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q145: What is the difference between HashSet and TreeSet? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q146: Does Garbage collection occur in permanent generation space in JVM? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q147: What does 'synchronized' mean? ⭐⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

[⬆️] JavaScript Interview Questions

Q1: What is Coercion in JavaScript?

Answer: In JavaScript conversion between different two build-in types called `coercion`. Coercion comes in two forms in JavaScript: *explicit* and *implicit*.

Here's an example of explicit coercion:

```
var a = "42";

var b = Number( a );

a;                              // "42"
b;                              // 42 -- the number!
```

And here's an example of implicit coercion:

```
var a = "42";

var b = a * 1;  // "42" implicitly coerced to 42 here

a;                              // "42"
b;                              // 42 -- the number!
```

Q2: Explain equality in JavaScript ⭐

Answer: JavaScript has both strict and type–converting comparisons:

- **Strict comparison (e.g., ===)** checks for value equality without allowing *coercion*
- **Abstract comparison (e.g. ==)** checks for value equality with *coercion* allowed

```
var a = "42";
var b = 42;

a == b;                 // true
a === b;                // false
```

Some simple equalityrules:

- If either value (aka side) in a comparison could be the `true` or `false` value, avoid `==` and use `===`.
- If either value in a comparison could be of these specific values (`0` , `""` , or `[]` -- empty array), avoid `==` and use `===`.
- In all other cases, you're safe to use `==`. Not only is it safe, but in many cases it simplifies your code in a way that improves readability.

Q3: What is typeof operator? ⭐

Answer: JavaScript provides a `typeof` operator that can examine a value and tell you what type it is:

```
var a;
typeof a;                               // "undefined"

a = "hello world";
typeof a;                               // "string"

a = 42;
typeof a;                               // "number"

a = true;
typeof a;                               // "boolean"

a = null;
typeof a;                               // "object" -- weird, bug

a = undefined;
typeof a;                               // "undefined"

a = { b: "c" };
typeof a;                               // "object"
```

Q4: What is the object type? ⭐

Answer: The object type refers to a compound value where you can set properties (named locations) that each hold their own values of any type.

```
var obj = {
        a: "hello world", // property
        b: 42,
        c: true
};

obj.a;          // "hello world", accessed with doted notation
obj.b;          // 42
obj.c;          // true

obj["a"];       // "hello world", accessed with bracket notation
obj["b"];       // 42
obj["c"];       // true
```

Bracket notation is also useful if you want to access a property/key but the name is stored in another variable, such as:

```
var obj = {
        a: "hello world",
        b: 42
};

var b = "a";

obj[b];                 // "hello world"
obj["b"];               // 42
```

Q5: Explain arrays in JavaScript ⭐

Answer: An `array` is an object that holds values (of any type) not particularly in named properties/keys, but rather in numerically indexed positions:

```
var arr = [
        "hello world",
        42,
        true
];

arr[0];                 // "hello world"
arr[1];                 // 42
arr[2];                 // true
arr.length;             // 3

typeof arr;             // "object"
```

Q6: What is Scope in JavaScript? ⭐

Answer: In JavaScript, each function gets its own *scope*. Scope is basically a collection of variables as well as the rules for how those variables are accessed by name. Only code inside that function can access that function's scoped variables.

A variable name has to be unique within the same scope. A scope can be nested inside another scope. If one scope is nested inside another, code inside the innermost scope can access variables from either scope.

Q7: What does "use strict" do? ⭐⭐

Answer: The `use strict` literal is entered at the top of a JavaScript program or at the top of a function and it helps you write safer JavaScript code by throwing an error if a global variable is created by mistake. For example, the following program will throw an error:

```
function doSomething(val) {
  "use strict";
  x = val + 10;
}`
```

It will throw an error because `x` was not defined and it is being set to some value in the global scope, which isn't allowed with `use strict` The small change below fixes the error being thrown:

```
function doSomething(val) {
  "use strict";
  var x = val + 10;
}
```

Source: *coderbyte.com*

Q8: Explain Null and Undefined in JavaScript ⭐⭐

Answer: JavaScript (and by extension TypeScript) has two bottom types: `null` and `undefined`. They are *intended* to mean different things:

- Something hasn't been initialized : `undefined`.
- Something is currently unavailable: `null`.

Q9: What's the difference between throw Error('msg') vs throw new Error('msg')? ⭐⭐

Details:

```
var err1 = Error('message');
var err2 = new Error('message');
```

Which one is correct and why?

Answer: Both are fine; the function call `Error(…)` is equivalent to the object creation expression `new Error(…)` with the same arguments.

Source: *stackoverflow.com*

Q10: Is there anyway to force using strict mode in Node.js? ⭐⭐

Answer: you can place

```
"use strict";
```

at the top of your file in **node >= 0.10.7**, but if you want your whole app to run in strict (**including external modules**) you can do this

```
node --use_strict
```

Source: *stackoverflow.com*

Q11: What's the difference between host objects and native objects? ⭐⭐

Answer:

- Native objects are objects that are part of the JavaScript language defined by the ECMAScript specification, such as `String` , `Math` , `RegExp` , `Object` , `Function` , etc.
- Host objects are provided by the runtime environment (browser or Node), such as `window` , `XMLHTTPRequest` , etc.

Source: *github.com/yangshun*

Q12: What is strict mode? ⭐⭐

Answer: *Strict Mode* is a new feature in ECMAScript 5 that allows you to place a program, or a function, in a "strict" operating context. This strict context prevents certain actions from being taken and throws more exceptions.

```
// Non-strict code...

(function(){
  "use strict";
  // Define your library strictly...
})();

// Non-strict code...
```

Q13: What is the difference between `==` and `===`? ⭐⭐

Answer: `==` is the abstract equality operator while `===` is the strict equality operator. The `==` operator will compare for equality after doing any necessary type conversions. The `===` operator will not do type conversion, so if two values are not the same type `===` will simply return `false`. When using `==`, funky things can happen, such as:

```
1 == '1'; // true
1 == [1]; // true
1 == true; // true
0 == ''; // true
0 == '0'; // true
0 == false; // true
```

My advice is never to use the `==` operator, except for convenience when comparing against `null` or `undefined`, where `a == null` will return `true` if `a` is `null` or `undefined`.

```
var a = null;
console.log(a == null); // true
console.log(a == undefined); // true
```

Source: *github.com/yangshun*

Q14: Explain the same-origin policy with regards to JavaScript. ⭐⭐

Answer: The same-origin policy prevents JavaScript from making requests across domain boundaries. An origin is defined as a combination of URI scheme, hostname, and port number. This policy prevents a malicious script on one page from obtaining access to sensitive data on another web page through that page's Document Object Model.

Source: *github.com/yangshun*

Q15: Make this work ⭐⭐

Details:

```
duplicate([1, 2, 3, 4, 5]); // [1,2,3,4,5,1,2,3,4,5]
```

Answer:

```
function duplicate(arr) {
  return arr.concat(arr);
}

duplicate([1, 2, 3, 4, 5]); // [1,2,3,4,5,1,2,3,4,5]
```

Source: *github.com/yangshun*

Q16: FizzBuzz Challenge ⭐⭐

Details: Create a for loop that iterates up to `100` while outputting **"fizz"** at multiples of `3`, **"buzz"** at multiples of `5` and **"fizzbuzz"** at multiples of `3` and `5`.

Answer: Check out this version of FizzBuzz:

```
for (let i = 1; i <= 100; i++) {
  let f = i % 3 == 0,
    b = i % 5 == 0;
  console.log(f ? (b ? 'FizzBuzz' : 'Fizz') : b ? 'Buzz' : i);
}
```

Source: *github.com/yangshun*

Q17: What is a Polyfill? ⭐⭐

Answer: A polyfill is essentially the specific code (or plugin) that would allow you to have some specific functionality that you expect in current or “modern” browsers to also work in other browsers that do not have the support for that functionality built in.

- Polyfills are not part of the HTML5 standard
- Polyfilling is not limited to Javascript

Source: *programmerinterview.com*

Q18: Why would you use something like the `load` event? Does this event have disadvantages? Do you know any alternatives, and why would you use those? ⭐⭐

Answer: The `load` event fires at the end of the document loading process. At this point, all of the objects in the document are in the DOM, and all the images, scripts, links and sub-frames have finished loading.

The DOM event `DOMContentLoaded` will fire after the DOM for the page has been constructed, but do not wait for other resources to finish loading. This is preferred in certain cases when you do not need the full page to be loaded before initializing.

Source: *github.com/yangshun*

Q19: Why is it, in general, a good idea to leave the global scope of a website as-is and never touch it? ⭐⭐

Answer: Every script has access to the global scope, and if everyone uses the global namespace to define their variables, collisions will likely occur. Use the module pattern (IIFEs) to encapsulate your variables within a local namespace.

Source: *github.com/yangshun*

Q20: What are some of the advantages/disadvantages of writing JavaScript code in a language that compiles to JavaScript? ⭐⭐

Answer: Some examples of languages that compile to JavaScript include CoffeeScript, Elm, ClojureScript, PureScript, and TypeScript.

Advantages:

- Fixes some of the longstanding problems in JavaScript and discourages JavaScript anti-patterns.
- Enables you to write shorter code, by providing some syntactic sugar on top of JavaScript, which I think ES5 lacks, but ES2015 is awesome.
- Static types are awesome (in the case of TypeScript) for large projects that need to be maintained over time.

Disadvantages:

- Require a build/compile process as browsers only run JavaScript and your code will need to be compiled into JavaScript before being served to browsers.
- Debugging can be a pain if your source maps do not map nicely to your pre-compiled source.
- Most developers are not familiar with these languages and will need to learn it. There's a ramp up cost involved for your team if you use it for your projects.

- Smaller community (depends on the language), which means resources, tutorials, libraries, and tooling would be harder to find.
- IDE/editor support might be lacking.
- These languages will always be behind the latest JavaScript standard.
- Developers should be cognizant of what their code is being compiled to—because that is what would actually be running, and that is what matters in the end.

Practically, ES2015 has vastly improved JavaScript and made it much nicer to write. I don't really see the need for CoffeeScript these days.

Source: *github.com/yangshun*

Q21: What language constructions do you use for iterating over object properties and array items? ⭐⭐

Answer: For objects:

- `for` loops - `for (var property in obj) { console.log(property); }`. However, this will also iterate through its inherited properties, and you will add an `obj.hasOwnProperty(property)` check before using it.
- `Object.keys()` - `Object.keys(obj).forEach(function (property) { ... })`. `Object.keys()` is a static method that will lists all enumerable properties of the object that you pass it.
- `Object.getOwnPropertyNames()` - `Object.getOwnPropertyNames(obj).forEach(function (property) { ... })`. `Object.getOwnPropertyNames()` is a static method that will lists all enumerable and non-enumerable properties of the object that you pass it.

For arrays:

- `for` loops - `for (var i = 0; i < arr.length; i++)`. The common pitfall here is that `var` is in the function scope and not the block scope and most of the time you would want block scoped iterator variable. ES2015 introduces `let` which has block scope and it is recommended to use that instead. So this becomes: `for (let i = 0; i < arr.length; i++)`.
- `forEach` - `arr.forEach(function (el, index) { ... })`. This construct can be more convenient at times because you do not have to use the `index` if all you need is the array elements. There are also the `every` and `some` methods which will allow you to terminate the iteration early.

Most of the time, I would prefer the `.forEach` method, but it really depends on what you are trying to do. `for` loops allow more flexibility, such as prematurely terminate the loop using `break` or incrementing the iterator more than once per loop.

Source: *github.com/yangshun*

Q22: What is let keyword in JavaScript? ⭐⭐

Answer: In addition to creating declarations for variables at the function level, ES6 lets you declare variables to belong to individual blocks (pairs of { .. }), using the `let` keyword.

Source: *github.com/getify*

Q23: Explain what a callback function is and provide a simple example. ⭐⭐

Answer: A `callback` function is a function that is passed to another function as an argument and is executed after some operation has been completed. Below is an example of a simple callback function that logs to the console *after* some operations have been completed.

```
function modifyArray(arr, callback) {
  // do something to arr here
  arr.push(100);
  // then execute the callback function that was passed
  callback();
}

var arr = [1, 2, 3, 4, 5];

modifyArray(arr, function() {
  console.log("array has been modified", arr);
});
```

Source: *coderbyte.com*

Q24: Being told that an unsorted array contains (n - 1) of n consecutive numbers (where the bounds are defined), find the missing number in O(n) time ⭐⭐

Answer:

```
// The output of the function should be 8
var arrayOfIntegers = [2, 5, 1, 4, 9, 6, 3, 7];
var upperBound = 9;
var lowerBound = 1;
```

```
findMissingNumber(arrayOfIntegers, upperBound, lowerBound); // 8

function findMissingNumber(arrayOfIntegers, upperBound, lowerBound) {
  // Iterate through array to find the sum of the numbers
  var sumOfIntegers = 0;
  for (var i = 0; i < arrayOfIntegers.length; i++) {
    sumOfIntegers += arrayOfIntegers[i];
  }

  // Find theoretical sum of the consecutive numbers using a variation of Gau
  // Formula: [(N * (N + 1)) / 2] - [(M * (M - 1)) / 2];
  // N is the upper bound and M is the lower bound

  upperLimitSum = (upperBound * (upperBound + 1)) / 2;
  lowerLimitSum = (lowerBound * (lowerBound - 1)) / 2;

  theoreticalSum = upperLimitSum - lowerLimitSum;

  return theoreticalSum - sumOfIntegers;
}
```

Source: *https://github.com/kennymkchan*

Q25: Remove duplicates of an array and return an array of only unique elements ⭐ ⭐

Answer:

```
// ES6 Implementation
var array = [1, 2, 3, 5, 1, 5, 9, 1, 2, 8];

Array.from(new Set(array)); // [1, 2, 3, 5, 9, 8]

// ES5 Implementation
var array = [1, 2, 3, 5, 1, 5, 9, 1, 2, 8];

uniqueArray(array); // [1, 2, 3, 5, 9, 8]

function uniqueArray(array) {
  var hashmap = {};
  var unique = [];

  for(var i = 0; i < array.length; i++) {
    // If key returns undefined (unique), it is evaluated as false.
    if(!hashmap.hasOwnProperty(array[i])) {
      hashmap[array[i]] = 1;
      unique.push(array[i]);
    }
  }

  return unique;
}
```

Source: *https://github.com/kennymkchan*

Q26: Explain Values and Types in JavaScript ⭐⭐

Answer: JavaScript has typed values, not typed variables. The following built-in types are available:

- `string`
- `number`
- `boolean`
- `null` and `undefined`
- `object`
- `symbol` (new to ES6)

Q27: How would you check if a number is an integer? ⭐⭐

Answer: A very simply way to check if a number is a decimal or integer is to see if there is a remainder left when you divide by 1.

```
function isInt(num) {
  return num % 1 === 0;
}

console.log(isInt(4)); // true
console.log(isInt(12.2)); // false
console.log(isInt(0.3)); // false
```

Source: *coderbyte.com*

Q28: Given a string, reverse each word in the sentence ⭐⭐

Details: For example `Welcome to this Javascript Guide!` should be become `emocleW ot siht tpircsavaJ !ediuG`

Answer:

```
var string = "Welcome to this Javascript Guide!";

// Output becomes !ediuG tpircsavaJ siht ot emocleW
var reverseEntireSentence = reverseBySeparator(string, "");

// Output becomes emocleW ot siht tpircsavaJ !ediuG
var reverseEachWord = reverseBySeparator(reverseEntireSentence, " ");

function reverseBySeparator(string, separator) {
  return string.split(separator).reverse().join(separator);
}
```

Source: *https://github.com/kennymkchan*

Q29: Write a function that would allow you to do this. ⭐⭐

Details:

```
var addSix = createBase(6);
addSix(10); // returns 16
addSix(21); // returns 27
```

Answer: You can create a closure to keep the value passed to the function `createBase` even after the inner function is returned. The inner function that is being returned is created within an outer function, making it a closure, and it has access to the variables within the outer function, in this case the variable `baseNumber` .

```
function createBase(baseNumber) {
  return function(N) {
    // we are referencing baseNumber here even though it was declared
    // outside of this function. Closures allow us to do this in JavaScript
    return baseNumber + N;
  }
}

var addSix = createBase(6);
addSix(10);
addSix(21);
```

Source: *coderbyte.com*

Q30: How would you use a closure to create a private counter? ⭐⭐

Answer: You can create a function within an outer function (a closure) that allows you to update a private variable but the variable wouldn't be accessible from outside the function without the use of a helper function.

```
function counter() {
  var _counter = 0;
  // return an object with several functions that allow you
  // to modify the private _counter variable
  return {
    add: function(increment) { _counter += increment; },
    retrieve: function() { return 'The counter is currently at: ' + _counter;
  }
}

// error if we try to access the private variable like below
// _counter;

// usage of our counter function
var c = counter();
c.add(5);
c.add(9);

// now we can access the private variable in the following way
c.retrieve(); // => The counter is currently at: 14
```

Source: *coderbyte.com*

Q31: Implement enqueue and dequeue using only two stacks ⭐⭐

Answer: *Enqueue* means to add an element, *dequeue* to remove an element.

```
var inputStack = []; // First stack
var outputStack = []; // Second stack

// For enqueue, just push the item into the first stack
function enqueue(stackInput, item) {
  return stackInput.push(item);
}

function dequeue(stackInput, stackOutput) {
  // Reverse the stack such that the first element of the output stack is the
  // last element of the input stack. After that, pop the top of the output t
  // get the first element that was ever pushed into the input stack
  if (stackOutput.length <= 0) {
    while(stackInput.length > 0) {
      var elementToOutput = stackInput.pop();
      stackOutput.push(elementToOutput);
    }
  }

  return stackOutput.pop();
}
```

Source: *https://github.com/kennymkchan*

Q32: How to check if an object is an array or not? Provide some code. ⭐⭐

Answer:

> The best way to find whether an object is instance of a particular class or not using `toString` method from `Object.prototype`

```
var arrayList = [1 , 2, 3];
```

One of the best use cases of type checking of an object is when we do method overloading in JavaScript. For understanding this let say we have a method called `greet` which take one single string and also a list of string, so making our `greet` method workable in both situation we need to know what kind of parameter is being passed, is it single value or list of value?

```
function greet(param) {
  if() {
    // here have to check whether param is array or not
  }
  else {
  }
}
```

However, in above implementation it might not necessary to check type for array, we can check for single value string and put array logic code in else block, let see below code for the same.

```
 function greet(param) {
   if(typeof param === 'string') {
   }
   else {
     // If param is of type array then this block of code would execute
   }
 }
```

Now it's fine we can go with above two implementations, but when we have a situation like a parameter can be `single value`, `array`, and `object` type then we will be in trouble.

Coming back to checking type of object, As we mentioned that we can use `Object.prototype.toString`

```
if(Object.prototype.toString.call(arrayList) === '[object Array]') {
  console.log('Array!');
}
```

If you are using `jQuery` then you can also used jQuery `isArray` method:

```
if($.isArray(arrayList)) {
  console.log('Array');
} else {
  console.log('Not an array');
}
```

FYI jQuery uses `Object.prototype.toString.call` internally to check whether an object is an array or not.

In modern browser, you can also use:

```
Array.isArray(arrayList);
```

`Array.isArray` is supported by Chrome 5, Firefox 4.0, IE 9, Opera 10.5 and Safari 5

Source: *github.com/ganqqwerty*

Q33: How to empty an array in JavaScript? ⭐⭐

Details:

```
var arrayList =  ['a', 'b', 'c', 'd', 'e', 'f'];
```

How could we empty the array above?

Answer: Method 1

```
arrayList = [];
```

Above code will set the variable `arrayList` to a new empty array. This is recommended if you don't have **references to the original array** `arrayList` anywhere else because It will actually create a new empty array. You should be careful with this way of empty the array, because if you have referenced this array from another variable, then the original reference array will remain unchanged, Only use this way if you have only referenced the array by its original variable `arrayList` .

For Instance:

```
var arrayList = ['a', 'b', 'c', 'd', 'e', 'f']; // Created array
var anotherArrayList = arrayList;  // Referenced arrayList by another variabl
arrayList = []; // Empty the array
console.log(anotherArrayList); // Output ['a', 'b', 'c', 'd', 'e', 'f']
```

Method 2

```
arrayList.length = 0;
```

Above code will clear the existing array by setting its length to 0. This way of empty the array also update all the reference variable which pointing to the original array. This way of empty the array is useful when you want to update all the another reference variable which pointing to `arrayList` .

For Instance:

```
var arrayList = ['a', 'b', 'c', 'd', 'e', 'f']; // Created array
var anotherArrayList = arrayList;  // Referenced arrayList by another variabl
arrayList.length = 0; // Empty the array by setting length to 0
console.log(anotherArrayList); // Output []
```

Method 3

```
arrayList.splice(0, arrayList.length);
```

Above implementation will also work perfectly. This way of empty the array will also update all the references of the original array.

```
var arrayList = ['a', 'b', 'c', 'd', 'e', 'f']; // Created array
var anotherArrayList = arrayList;  // Referenced arrayList by another variabl
arrayList.splice(0, arrayList.length); // Empty the array by setting length t
console.log(anotherArrayList); // Output []
```

Method 4

```
while(arrayList.length) {
  arrayList.pop();
}
```

Above implementation can also empty the array. But not recommended to use often.

Source: *github.com/ganqqwerty*

Q34: Write a "mul" function which will properly when invoked as below syntax. ⭐⭐

Details:

```
console.log(mul(2)(3)(4)); // output : 24
console.log(mul(4)(3)(4)); // output : 48
```

Answer:

```
function mul (x) {
  return function (y) { // anonymous function
    return function (z) { // anonymous function
      return x * y * z;
    };
  };
}
```

Here `mul` function accept the first argument and return anonymous function which take the second parameter and return anonymous function which take the third parameter and return multiplication of arguments which is being passed in successive

In JavaScript function defined inside has access to outer function variable and function is the first class object so it can be returned by function as well and passed as argument in another function.

- A function is an instance of the Object type
- A function can have properties and has a link back to its constructor method

- Function can be stored as variable
- Function can be pass as a parameter to another function
- Function can be returned from function

Source: *github.com/ganqqwerty*

Q35: Explain event bubbling and how one may prevent it ⭐⭐

Answer: Event bubbling is the concept in which an event triggers at the deepest possible element, and triggers on parent elements in nesting order. As a result, when clicking on a child element one may exhibit the handler of the parent activating.

One way to prevent event bubbling is using `event.stopPropagation()` or `event.cancelBubble` on IE < 9.

Source: *https://github.com/kennymkchan*

Q36: Provide some examples of non-bulean value coercion to a boolean one ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q37: How to compare two objects in JavaScript? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q38: What will be the output of the following code? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q39: What is the drawback of creating true private in JavaScript? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q40: Write a recursive function that returns the binary string of a given decimal number ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q41: What will be the output of the following code? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q42: What will be the output of the following code? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q43: When should we use generators in ES6? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q44: What is IIFEs (Immediately Invoked Function Expressions)? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q45: Given two strings, return true if they are anagrams of one another ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q46: Find the intersection of two arrays. An intersection would be the common elements that exists within both arrays. In this case, these elements should be unique! ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q47: What will the following code output? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q48: Write a function that would allow you to do this ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q49: Given an array of integers, find the largest difference between two elements such that the element of lesser value must come before the greater element ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q50: Given an array of integers, find the largest product yielded from three of the integers ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q51: Check if a given string is a palindrome. Case sensitivity should be taken into account. ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q52: What's the difference between using “let” and “var” to declare a variable in ES6? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q53: When should I use Arrow functions in ES6? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q54: What is the motivation for bringing Symbols to ES6? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q55: What is generator in JS? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q56: What are the benefits of using spread syntax in ES6 and how is it different from rest syntax? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q57: Explain the difference between "undefined" and "not defined" in JavaScript ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q58: What is 'Currying'? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q59: What is the definition of a higher-order function? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q60: What are the differences between ES6 class and ES5 function constructors? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q61: Explain the differences on the usage of `foo` between `function foo() {}` and `var foo = function() {}` ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q62: Could you explain the difference between ES5 and ES6 ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q63: What is the difference between a shim and a polyfill? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q64: What are the advantages and disadvantages of using "use strict"? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q65: Describe closure concept in JavaScript as best as you could ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q66: What is the difference between anonymous and named functions? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q67: What is the difference between document `load` event and document `DOMContentLoaded` event? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q68: Why is extending built-in JavaScript objects not a good idea? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q69: Explain `Function.prototype.bind`. ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q70: What's the difference between `.call` and `.apply`? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q71: What's a typical use case for anonymous functions? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q72: What is a closure, and how/why would you use one? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q73: What do you think of AMD vs CommonJS? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q74: Suggest one simple way of removing duplicates from an array using ES6 ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q75: Why should we use ES6 classes? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q76: What is the preferred syntax for defining enums in JavaScript? ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q77: Explain the difference between Object.freeze() vs const ⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q78: Can you give an example for destructuring an object or an array in ES6? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q79: Explain what is hoisting in Javascript ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q80: Explain prototype inheritance in JavaScript? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q81: What will the following code output? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q82: What does the term "Transpiling" stand for? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q83: How would you create a private variable in JavaScript? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q84: What is the "new" keyword in JavaScript? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q85: Explain the Prototype Design Pattern ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q86: How does the “this” keyword work? Provide some code examples. ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q87: What is the Temporal Dead Zone in ES6? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q88: Create a function that will evaluate if a given expression has balanced parentheses using stacks ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q89: Check if a given string is a isomorphic ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q90: When should you NOT use arrow functions in ES6? Name three or more cases. ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q91: Explain how JSONP works (and how it's not really Ajax) ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q92: What will be the output of the following code? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q93: What are the actual uses of ES6 WeakMap? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q94: Explain difference between: `function Person(){}`, `var person = Person()`, and `var person = new Person()`? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q95: What is Hoisting in JavaScript? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q96: What is “closure” in javascript? Provide an example? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q97: Can you describe the main difference between a `.forEach` loop and a `.map()` loop and why you would pick one versus the other? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q98: What's the difference between a variable that is: `null`, `undefined` or undeclared? How would you go about checking for any of these states? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q99: How can you share code between files? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q100: Explain why the following doesn't work as an IIFE. What needs to be changed to properly make it an IIFE? ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q101: Write a recursive function that performs a binary search ⭐⭐⭐⭐

Read answer on 👉 FullStack.Cafe

Q102: What will be the output of the following code? ⭐⭐⭐⭐

Read answer on 👉 <a hre

Releases

No releases published

Sponsor this project

🔗 https://www.fullstack.cafe

🔗 https://www.mlstack.cafe

Packages

No packages published