# S M Square 1st Round Interview Questions

## W.A.P to arrange string in descending order and find odd index.

## W.A.P to find odd digits from the given string.

## Authenticate vs Authentication

## BDD vs TDD

## Diff b/w CSS and XPATH

## CI/CD

## Constructor

## W.A.P to switch frame

## Diff types of Alerts in Selenium

## Difference b/w Scenario and Scenario Outline?

A Scenario is a single, independent test case with a fixed set of inputs and expected outputs. A Scenario Outline, on the other hand, allows you to execute the same test scenario multiple times with different sets of inputs, effectively making it a template for parameterized testing.

**Scenario:**

A single test case that runs once, with hardcoded values for inputs and expected outputs.

**Scenario Outline:**

A template for a scenario that can be run multiple times with different data provided in an "Examples" table.

**Key Difference:**

The primary difference lies in how they handle data. Scenario uses fixed values, while Scenario Outline uses placeholders that are populated by the "Examples" table, allowing for data-driven testing.

**Example:**

Imagine a scenario to test a login functionality. A regular scenario might hardcode the username and password. A Scenario Outline, however, would use placeholders for username and password, and then the "Examples" table would provide multiple sets of valid and invalid credentials to test against.

**Purpose:**

Scenario Outlines are particularly useful when you need to test the same functionality with various inputs, ensuring the application behaves correctly under different conditions.

In essence, a Scenario Outline provides a more efficient and readable way to write test cases that need to be executed with different input combinations, reducing redundancy and making the code more maintainable.

## What is Bug Lifecycle?

A systematic process that tracks the stages a bug or defect goes through from its initial detection to its resolution or closure in software testing. It ensures that bugs are properly identified, managed, and resolved, ultimately leading to higher quality software.

Key Stages of the Bug Lifecycle:

1. **New:** A bug is initially discovered and reported.
2. **Assigned:** The bug report is reviewed, and a developer is assigned to investigate and potentially fix it.
3. **Open:** The developer starts working on the bug, reproducing the issue and analyzing its root cause.
4. **Fixed:** The developer makes the necessary code changes to resolve the bug.
5. **Retest/Pending Retest:** The QA team verifies if the bug is resolved, and if necessary, the developer makes further changes.
6. **Reopened:** If the bug is still present after retesting, it is reopened and returned to the developer for further investigation.
7. **Verified:** The QA team confirms that the bug is resolved and verified.
8. **Closed:** The bug is officially resolved and closed.

The Bug Lifecycle helps in:

* **Tracking:** Monitoring the progress and status of each bug.
* **Prioritization:** Determining the severity and urgency of bugs for efficient resolution.
* **Communication:** Facilitating communication and collaboration between testers, developers, and project managers.
* **Root Cause Analysis:** Identifying the underlying causes of bugs to prevent their recurrence.
* **Quality Assurance:** Ensuring that bugs are properly addressed and that the software delivers high-quality performance.

## Difference between const and readonly?

|  |  |
| --- | --- |
| **ReadOnly Keyword** | **Const Keyword** |
| **In C#, readonly fields can be created using readonly keyword** | **In C#, constant fields are created using const keyword.** |
| **ReadOnly is a runtime constant.** | **Const is a compile time constant.** |
| **The value of readonly field can be changed.** | **The value of the const field can not be changed.** |
| **It cannot be declared inside the method.** | **It can be declared inside the method.** |
| **In readonly fields, we can assign values in declaration and in the constructor part.** | **In const fields, we can only assign values in declaration part.** |
| **It can be used with static modifiers.** | **It cannot be used with static modifiers.** |

# Capgemini 1st Round Interview Questions

## What all different activities you have done in your last project?

## 2. BDD Framework

## 3. Selenium WebDriver is Interface or Class? Why it is interface?

WebDriver in Selenium is an interface, not a class. In the case of Selenium, the WebDriver interface defines a set of methods for interacting with web browsers. By using an interface, Selenium allows different browser vendors to provide their own implementations of the WebDriver interface.

For example, you can use the same Selenium script to interact with Chrome, Firefox, or other browsers, just by using the corresponding WebDriver implementation for each browser. The fact that WebDriver is an interface promotes flexibility and code reusability, as it allows for easy switching between different browser implementations without changing the core logic of your test scripts.

## 4. Where interface method will be implemented?

In Selenium, the interface methods of WebDriver are implemented by specific browser classes that serve as WebDriver implementations for different browsers. These classes are often provided by the browser vendors or the Selenium project itself.

Here are some examples of WebDriver implementations for popular browsers:

ChromeDriver: Implements the WebDriver interface for Google Chrome. You need to download the ChromeDriver executable and use it in your Selenium script.

**Example usage:**

**WebDriver driver = new ChromeDriver();**

The WebDriver interface in Selenium provides a set of methods that allow you to interact with web browsers. Here are some commonly used methods:

**Navigation methods:**

get(String url): Open a new browser window with the specified URL.

navigate().to(String url): Navigate to the specified URL.

navigate().back(): Navigate back in the browser history.

navigate().forward(): Navigate forward in the browser history.

navigate().refresh(): Refresh the current page.

**Window handling methods:**

getWindowHandle(): Get the handle of the current window.

getWindowHandles(): Get the handles of all open windows.

switchTo().window(String windowHandle): Switch to a different window.

**Element location methods:**

findElement(By by): Find the first element matching the given By strategy.

findElements(By by): Find all elements matching the given By strategy.

**Browser interaction methods:**

manage().window(): Methods related to managing the browser window, such as maximizing, minimizing, or setting the size.

manage().timeouts(): Set timeouts for different actions (page load timeout, script timeout, etc.).

manage().deleteAllCookies(): Delete all cookies in the current browser session.

**Element interaction methods:**

WebElement findElement(By by): Find the first element matching the given By strategy.

List<WebElement> findElements(By by): Find all elements matching the given By strategy.

getTitle(): Get the title of the current page.

getCurrentUrl(): Get the URL of the current page.

getPageSource(): Get the source code of the current page.

**Alert handling methods:**

switchTo().alert(): Switch to the currently active alert.

switchTo().frame(): Switch to the specified frame.

**Script execution methods:**

executeScript(String script, Object... args): Execute JavaScript in the context of the currently selected frame or window.

executeAsyncScript(String script, Object... args): Execute asynchronous JavaScript in the context of the currently selected frame or window.

## 5. How to get input box text? Difference methods?

To retrieve the text from an input box in Selenium, you can use the `getAttribute("value")` method, or you can use the `getText()` method. The appropriate method to use may depend on the type of input element you are working with.

**Using getAttribute("value"):** This method is commonly used for input elements such as text boxes. WebElement inputElement = driver.findElement(By.id("yourInputId"));

String inputValue = inputElement.getAttribute("value");

System.out.println("Text from input box: " + inputValue);

**Using getText():** While getText() is commonly used for regular text elements (e.g., `<div>`, `<span>`), it might work for some input elements as well, particularly if they are read-only or if the text is available as a visible label.

WebElement inputElement = driver.findElement(By.id("yourInputId"));

String inputValue = inputElement.getText();

System.out.println("Text from input box: " + inputValue);

## 6. ChromeOptions

ChromeOptions is a class in Selenium that allows you to customize and configure various options when working with the Chrome browser. It provides a way to set additional capabilities and preferences that control the behavior of the ChromeDriver during test execution.

Here are some common use cases for ChromeOptions

**Setting Browser Preferences:**

ChromeOptions options = new ChromeOptions();

options.addArguments("--start-maximized"); // Maximize the browser window on startup

options.addArguments("--disable-popup-blocking"); // Disable popup blocking

**Headless Mode:**

ChromeOptions options = new ChromeOptions();

options.setHeadless(true); // Run Chrome in headless mode (no GUI)

**Setting Browser Language:**

ChromeOptions options = new ChromeOptions();

options.addArguments("--lang=en-US"); // Set browser language to English (United States)

**Ignoring Certificate Errors:**

ChromeOptions options = new ChromeOptions();

options.addArguments("--ignore-certificate-errors"); // Ignore SSL certificate errors

**Adding Extensions:**

ChromeOptions options = new ChromeOptions();

options.addExtensions(new File("/path/to/extension.crx")); // Add a Chrome extension

**Disabling GPU:**

ChromeOptions options = new ChromeOptions();

options.addArguments("--disable-gpu"); // Disable GPU hardware acceleration

**Setting a Proxy:**

ChromeOptions options = new ChromeOptions();

Proxy proxy = new Proxy();

proxy.setHttpProxy("proxy.example.com:8080");

options.setProxy(proxy);

**Running Chrome in Incognito Mode:**

ChromeOptions options = new ChromeOptions();

options.addArguments("--incognito"); // Run Chrome in incognito (private browsing) mode

## 7. Alert is interface or class? What all are methods available in Alerts?

In Selenium, the Alert interface represents JavaScript pop-up alerts, confirmations, and prompts. It is not a class but an interface. The Alert interface provides methods to interact with these pop-ups.

Here are some common methods available in the Alert interface:

**accept():** Clicks the "OK" button in the alert.

Alert alert = driver.switchTo().alert();

alert.accept();

**dismiss():** Clicks the "Cancel" button in the alert.

Alert alert = driver.switchTo().alert();

alert.dismiss();

**getText():** Retrieves the text present in the alert

Alert alert = driver.switchTo().alert();

String alertText = alert.getText();

**sendKeys(String keysToSend):** Sends some text to the alert. Useful when dealing with prompts.

Alert alert = driver.switchTo().alert();

alert.sendKeys("Text to send");

**sendKeys(Keys keysToSend):** Sends special keys to the alert. This can be used for handling certain keyboard actions.

Alert alert = driver.switchTo().alert();

alert.sendKeys(Keys.ENTER);

**authenticateUsing(Credentials credentials): Authenticates the user with the given credentials when a basic authentication pop-up is present.**

Alert alert = driver.switchTo().alert();

alert.authenticateUsing(new UserAndPassword("username", "password"));

**setCredentials(Credentials credentials):** Sets the user credentials for handling HTTP authentication pop-ups.

Alert alert = driver.switchTo().alert();

alert.setCredentials(new UserAndPassword("username", "password"));

## 8. How to navigate to child window and again back to parent window?

To navigate between the parent window and child windows in Selenium, you can use the `getWindowHandles()` method to get handles of all currently open windows, and then switch between them using the `switchTo().window()` method. Here's an example:

// Assuming you are working with a WebDriver instance called 'driver'

// Get the handle of the parent window

String parentWindowHandle = driver.getWindowHandle();

// Perform an action that opens a new window (e.g., click a link that opens in a new window)

// ...

// Get all window handles

Set<String> allWindowHandles = driver.getWindowHandles();

// Iterate through the handles to find the handle of the child window

for (String windowHandle : allWindowHandles) {

if (!windowHandle.equals(parentWindowHandle)) {

// Switch to the child window

driver.switchTo().window(windowHandle);

// Perform actions in the child window if needed

// ...

// Close or do other operations in the child window

// Close the child window (optional)

driver.close();

// Switch back to the parent window

driver.switchTo().window(parentWindowHandle);

// Perform actions in the parent window if needed

// ...

// Break out of the loop after handling the child window

break;

}

}

In this example:

1. `driver.getWindowHandle()` gets the handle of the parent window.

2. `driver.getWindowHandles()` gets a set of all open window handles.

3. A loop iterates through the handles to find the handle of the child window.

4. `driver.switchTo().window(windowHandle)` switches to the child window.

5. Actions can be performed in the child window if needed.

6. Optionally, the child window can be closed using `driver.close()`.

7. `driver.switchTo().window(parentWindowHandle)` switches back to the parent window.

8. Actions can be performed in the parent window if needed.

## 9. Exception? What action will u perform under catch block?

In Selenium, exceptions are used to handle unexpected situations or errors that may occur during the execution of your test scripts. These exceptions provide a mechanism to gracefully handle errors and allow you to take appropriate actions, such as logging the error, capturing screenshots for debugging, or implementing recovery strategies.

Here are a few common uses and actions performed under the catch block in Selenium:

**Logging and Reporting:**

* You can log information about the exception to provide insights into what went wrong during the test execution.
* You may use a logging framework (e.g., Log4j) to log detailed information about the exception.

catch (Exception e) {

// Log the exception details

log.error("Exception occurred: " + e.getMessage());

}

**Taking Screenshots:**

* Capture screenshots when an exception occurs to visually inspect the state of the application at the point of failure.

catch (Exception e) {

// Take a screenshot for debugging

File screenshotFile = ((TakesScreenshot) driver).getScreenshotAs(OutputType.FILE);

FileUtils.copyFile(screenshotFile, new File("screenshot.png"));

}

**Recovery Mechanisms:**

* Implement recovery strategies to bring the application back to a stable state after encountering an exception.

catch (Exception e) {

// Attempt to refresh the page or navigate back to a known state

driver.navigate().refresh();

}

## 10. How to select 2nd last dropdown value from dropdown?

To select the second-to-last (penultimate) option from a dropdown in Selenium, you can use the Select class along with the getOptions() method to get a list of all options, and then choose the one at the desired index. The index of options is zero-based.

Here's an example using Java and the Select class:

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.ui.Select;

import org.openqa.selenium.chrome.ChromeDriver;

public class SelectPenultimateOption {

public static void main(String[] args) {

// Set the path to the chromedriver executable

System.setProperty("webdriver.chrome.driver", "path/to/chromedriver");

// Launch the Chrome browser

WebDriver driver = new ChromeDriver();

// Navigate to your webpage

driver.get("your\_website\_url");

// Identify the dropdown element

WebElement dropdownElement = driver.findElement(By.id("your\_dropdown\_id"));

// Create a Select object

Select dropdown = new Select(dropdownElement);

// Get all options

java.util.List<WebElement> options = dropdown.getOptions();

// Calculate the index of the second-to-last option

int penultimateIndex = options.size() - 2;

// Select the second-to-last option

dropdown.selectByIndex(penultimateIndex);

// Close the browser

driver.quit();

}

}

## 11. LINQ

* It is a query language designed by Microsoft as .Net framework using which we can write query on wide data source like Arrays, Collections, Database Table, Dataset’s, Xml Data.

Int[] arr = {12,44, 46, 87, 84, 27, 96, 21, 90};

**SQL:** Select <column\_list> From <table> as <alias> [<clauses>]

Clauses - Where -> Group By, Having, Order By

**Collection || Array :** From <alias> in <Collection || array> [<clauses>] Select <alias>

Ex: var brr = From I in arr Select I;

**Select value > 40**

var brr = From I in arr Where I > 40 Select I;

**Select value in Ordered Array**

var brr = From I in arr Where I > 40 Orderby I Select I;

**Select value in Descending Order Array**

var brr = From I in arr Where I > 40 Orderby I Descending Select I;

## 12. What is Sealed class in C#?

In C#, a sealed class is a class that cannot be inherited or used as a base class for other classes. When a class is marked as sealed, it means that it is complete and cannot have any subclasses.

Here is an example of a sealed class in C#:

sealed class SealedClass

{

// Class members

}

* In this example, SealedClass is marked as sealed, and no other class can inherit from it:

## 13. Spectflow?

Don’t Know

## 14. What is Dictionary in C#?

In C#, Dictionary is a collection type provided by the .NET Framework that represents a generic collection of key-value pairs. It is part of the System.Collections.Generic namespace. A Dictionary is used to store data in pairs where each pair consists of a unique key and its corresponding value. The key is used to access its associated value quickly.

Here's a simple example of using a Dictionary:

using System;

using System.Collections.Generic;

class Program

{

static void Main()

{

// Creating a Dictionary with string keys and int values

Dictionary<string, int> myDictionary = new Dictionary<string, int>();

// Adding key-value pairs

myDictionary.Add("one", 1);

myDictionary.Add("two", 2);

myDictionary.Add("three", 3);

// Accessing values using keys

Console.WriteLine("Value for key 'two': " + myDictionary["two"]);

// Iterating through key-value pairs

foreach (var pair in myDictionary)

{

Console.WriteLine($"Key: {pair.Key}, Value: {pair.Value}");

}

}

}

Output:

Value for key 'two': 2

Key: one, Value: 1

Key: two, Value: 2

Key: three, Value: 3

## 15. What will happen if we create class and method as static?

When you declare a class or method as static in C#, it means that the class or method belongs to the type itself rather than an instance of the type. Here's what happens when you create a static class or method:

**Static Class:**

* A static class is a class that cannot be instantiated. It cannot have instance members (fields, properties, or methods that require an instance of the class).
* All members (fields, properties, and methods) of a static class must be declared as static.
* You cannot create an instance of a static class using the new keyword.

Example of a static class:

public static class StaticClass

{

public static void StaticMethod()

{

// Static method implementation

}

public static int StaticProperty { get; set; }

}

**Static Method:**

* A static method is a method that belongs to the class itself rather than an instance of the class. It can be called without creating an instance of the class.
* Inside a static method, you cannot access instance members directly (non-static members). You can only access other static members or local variables.

Example of a class with a static method:

public class MyClass

{

public static void StaticMethod()

{

// Static method implementation

}

public void InstanceMethod()

{

// Instance method implementation

}

}

In summary, when you create a class or method as static:

* A static class cannot be instantiated, and all its members must be static.
* A static method belongs to the class itself and can be called without creating an instance of the class. Inside a static method, you cannot access instance members directly.

## 16. How will you handle the Excel?

Handling Excel files in C# can be accomplished using various libraries. One commonly used library for working with Excel files in C# is the DocumentFormat.OpenXml.

To handle Excel files using the DocumentFormat.OpenXml library, you can perform tasks such as reading data from an existing Excel file or creating and writing data to a new Excel file. Below are examples for both scenarios:

**Reading Data from Excel:**

using DocumentFormat.OpenXml.Packaging;

using DocumentFormat.OpenXml.Spreadsheet;

// Specify the path to the Excel file

string filePath = "path/to/your/excel/file.xlsx";

// Open the Excel file for reading

using (SpreadsheetDocument spreadsheetDocument = SpreadsheetDocument.Open(filePath, false))

{

WorkbookPart workbookPart = spreadsheetDocument.WorkbookPart;

Sheet sheet = workbookPart.Workbook.Descendants<Sheet>().FirstOrDefault();

if (sheet != null)

{

WorksheetPart worksheetPart = (WorksheetPart)workbookPart.GetPartById(sheet.Id);

SheetData sheetData = worksheetPart.Worksheet.Elements<SheetData>().First();

// Iterate through rows and cells

foreach (Row row in sheetData.Elements<Row>())

{

foreach (Cell cell in row.Elements<Cell>())

{

string cellValue = GetCellValue(workbookPart, cell);

Console.Write(cellValue + "\t");

}

Console.WriteLine();

}

}

}

// Helper method to get cell value

private static string GetCellValue(WorkbookPart workbookPart, Cell cell)

{

SharedStringTablePart sharedStringPart = workbookPart.SharedStringTablePart;

if (cell.DataType != null && cell.DataType.Value == CellValues.SharedString)

{

int sharedStringIndex = int.Parse(cell.InnerText);

return sharedStringPart.SharedStringTable.Elements<SharedStringItem>().ElementAt(sharedStringIndex).InnerText;

}

else

{

return cell.InnerText;

}

}

## 17. Difference b/w == and Equals method?

In C#, the == operator and the Equals method serve different purposes and behave differently based on the type of the objects being compared.

**== Operator:**

* The == operator is used for equality comparison between two objects.
* For value types (e.g., primitive types like int, double, struct), == compares the values.
* For reference types, == compares the references, not the actual content of the objects.
* For strings, == compares the content of the strings, but it can be overloaded for custom types to provide a specific behavior.

**Example:**

int a = 5;

int b = 5;

Console.WriteLine(a == b); // true (values are equal)

string str1 = "Hello";

string str2 = "Hello";

Console.WriteLine(str1 == str2); // true (content is equal)

**Equals Method:**

* The Equals method is a method provided by the System.Object class, and it is overridden by many types to provide custom equality comparison.
* The default implementation of Equals in the Object class compares references for reference types.
* For value types, the default implementation compares the values.
* For strings, Equals can be overridden to provide custom comparison logic.

**Example:**

object obj1 = new object();

object obj2 = new object();

Console.WriteLine(obj1.Equals(obj2)); // false (references are different)

string str3 = "Hello";

string str4 = "Hello";

Console.WriteLine(str3.Equals(str4)); // true (content is equal)

# AlgoShack 1st Round Interview Questions

## Can you please introduce yourself?

## Can you please explain me your framework?

* I have an experience on Data Driven Framework. It is easy to handle, all my scripts to execute and generate los, screenshots and reports by this framework.
* The Data Driven test automation framework is focused on separating the test logics and the test data from each other.
* It allows us to create test automations scripts by passing different sets of test data.
* The test data set kept in the external files pr resources such as MS Excel sheet, MS access tables, SQL database, XML files etc.
* The test scripts connect to the external resources to get the test data.
* By using this framework we could easily make the scripts work properly for different sets of test data.
* This framework significantly reduces the number of test scripts compared to the module based framework.
* It is reliable in terms of no impact on tests by chaining the test data but it has its own drawback such as testers who work on this framework needs to have the hands on programming knowledge to the test scripts.

## Explain the Page Objet Model?

As per the Page Object Model, we have maintained a class for every webpage.

Each webpage has separate class that holds the functionality and members of the web page. Separate test class for individual tests.

## Write a program to perform login? (username, password, click)

using OpenQA.Selenium;

using OpenQA.Selenium.Chrome;

class Program

{

static void Main()

{

// Set the path to the ChromeDriver executable

string chromeDriverPath = "path\_to\_chromedriver.exe"; // Download from https://sites.google.com/chromium.org/driver/

// Create a new instance of the ChromeDriver

IWebDriver driver = new ChromeDriver(chromeDriverPath);

// Navigate to the login page

driver.Navigate().GoToUrl("https://example.com/login");

// Find the username and password input elements by their IDs

IWebElement usernameInput = driver.FindElement(By.Id("username"));

IWebElement passwordInput = driver.FindElement(By.Id("password"));

IWebElement loginButton = driver.FindElement(By.Id("loginButton"));

// Enter your login credentials

usernameInput.SendKeys("your\_username");

passwordInput.SendKeys("your\_password");

// Click the login button

loginButton.Click();

// You may want to add some verification or error handling here

// For example, check if the login was successful by looking for elements on the next page

// Close the browser

driver.Quit();

}

}

## W.A.P to reverse a string? (//Vikash - //hsakiv

Refer 1st question under Logical Questions List?

## Write logic for Extent Report?

using System;

using AventStack.ExtentReports;

using AventStack.ExtentReports.Reporter;

using NUnit.Framework;

[TestFixture]

public class ExtentReportExample

{

private ExtentReports extent;

private ExtentTest test;

[OneTimeSetUp]

public void Setup()

{

// Specify the report file location

var htmlReporter = new ExtentHtmlReporter(@"C:\Reports\ExtentReport.html");

// Create an ExtentReports object and attach the reporter

extent = new ExtentReports();

extent.AttachReporter(htmlReporter);

}

[Test]

public void Test1()

{

// Start a test

test = extent.CreateTest("Test 1", "This is a sample test");

// Log information

test.Log(Status.Info, "Test step information");

// Log a failure

test.Fail("Test step failure");

// Log a screenshot (you can replace "screenshot.png" with an actual screenshot file)

test.AddScreenCaptureFromPath(@"C:\Screenshots\screenshot.png");

// Mark the test as pass or fail

Assert.Fail("Failing the test intentionally for demo");

}

[OneTimeTearDown]

public void TearDown()

{

// Flush the ExtentReports instance to generate the report

extent.Flush();

}

}

## Type of waits?

The different waits in selenium are –

**Implicit Wait:** An implicit wait is a type of wait which wait for a specified time while locating an element before throwing NoSuchElementException. By default, selenium tries to find elements immediately when required without any wait. So, it is good to use implicit wait. This wait is applied to all the element the current driver instance.

driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);

**Explicit Wait:** An implicit wait is a type of wait which is applied to a particular web element until the expected condition specified is met.

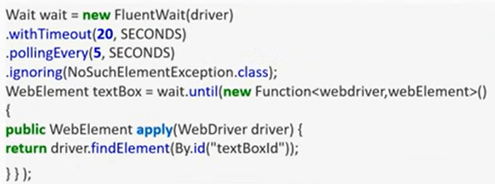
WebDriverWait wait = new WebDriverWait(driver,10);

WebElement element = wail.until(ExpectedCondition.elementToBeClickable(By.id(“elementId”));

Some of the commonly used expected condition of an element that can be used with explicit waits are –

* elementToBeClickable(WebElement element or By locator)
* visibilityOfElementLocated(By locator)
* attributeContains (WebElement element, String attribute, String value)
* alertIsPresent()
* titleContains(String title)
* titleIs(String title)
* textToBePresentInElementLocated

**Fluent Wait:** A fluent wait is a type of wait in which we can also specify polling interval (intervals after which driver will try to find the element) along with the maximum timeout value.



## Locators in Selenium?

The different locators in selenium are –

* ID
* XPath
* cssSelector
* className
* tagname
* name
* linkText
* partialLinkText

## Types of XPath?

**Absolute**: An absolute XPath is a way of locating an element using an XML expression beginning from root node i.e. htmlnode in case of web pages. The main disadvantage of absolute xpath is that even with slightest change in the UI or any element the whole absolute XPath fails.

Example – html/body/div/div[2]/div/div/div/div[1]/div/input

**Relative**: A relative XPath is a way of locating an element using an XML expression beginning from anywhere in the HTML document. There are different ways of creating relative XPath which are used for creating robust XPaths (unaffected by changes in other UI elements).

Example - //input[@id=’username’]

## Difference b/w method overloading and method overriding?

Method overloading and method overriding are two concepts in object-oriented programming that involve the use of methods in classes. Here are the key differences between method overloading and

**method overriding:**

1. Definition:

* Method Overloading:
  + Method overloading occurs when a class has multiple methods with the same name but different parameter lists (number or types of parameters).
  + Overloaded methods are distinguished by the number or types of parameters they accept.
* Method Overriding:
* Method overriding occurs when a subclass provides a specific implementation for a method that is already defined in its superclass.
* The method in the subclass must have the same signature (name, return type, and parameter types) as the method in the superclass.

1. Inheritance:

* Method Overloading:
* Overloading can occur in the same class or in a class hierarchy.
* It is not dependent on inheritance.
* Method Overriding:
* Overriding is specifically associated with inheritance and involves a relationship between a superclass and a subclass.

1. Scope:

* Method Overloading:
* Overloaded methods are within the same class, and the selection of the method is determined at compile-time based on the method signature.
* Method Overriding:
* Overriding involves a relationship between a superclass and a subclass, and the selection of the method is determined at runtime based on the actual type of the object.

1. Return Type:

* Method Overloading:
* Return type alone is not considered when overloading methods.
* Method Overriding:
* Return type must be the same (or covariant) in both the superclass and subclass for method overriding.

1. Keyword Usage:

* Method Overloading:
* No specific keyword is used. It is achieved by defining multiple methods with the same name but different parameter lists.
* Method Overriding:
* The override keyword is used in the subclass to indicate that a method is intended to override a method in the superclass.

1. Use Case:

* Method Overloading:
* Used to provide multiple ways to perform the same operation within a class.
* Method Overriding:
* Used to provide a specific implementation of a method in a subclass, allowing for polymorphism and dynamic dispatch.

## Difference b/w Abstract class and Interface?

Abstract classes and interfaces are both mechanisms in object-oriented programming that allow for abstraction and the definition of common behavior. However, there are several key differences between abstract classes and interfaces:

1. Definition:

* Abstract Class:
* An abstract class is a class that cannot be instantiated on its own and may contain both abstract and non-abstract (concrete) methods.
* It may have fields, constructors, and other members.
* Interface:
* An interface is a contract for a class to implement. It contains only method signatures (abstract methods), properties, events, and indexers.
* It does not have any implementation; the implementing class provides the actual code.

1. Instantiation:

* Abstract Class:
* Cannot be instantiated directly. It serves as a base class for other classes.
* Interface:
* Cannot be instantiated on its own. It is implemented by classes that provide the implementation for its members.

1. Inheritance:

* Abstract Class:
* Supports single-class inheritance, meaning a class can inherit from only one abstract class.
* Interface:
* Supports multiple interface inheritance, meaning a class can implement multiple interfaces.

1. Constructor

* Abstract Class:
* Can have a constructor, and the constructor of an abstract class is called when an instance of a derived class is created.

1. Interface:

* Cannot have a constructor. Interfaces do not define any instance-level state.

1. Accessibility:

* Abstract Class:
* Can have access modifiers for its members (public, private, protected, etc.).
* Interface:
* All members are implicitly public. The access modifiers are determined by the implementing class.

1. Partial Implementation:

* Abstract Class:
* Can have both abstract and non-abstract methods. It may provide some default implementation for certain methods.
* Interface:
* Contains only method signatures, and the implementing class must provide the implementation for all the methods.

1. Fields:

* Abstract Class:
* Can have fields (instance variables).
* Interface:
* Cannot have fields; only method signatures, properties, events, and indexers.

1. Use Cases:

* Abstract Class:
* Used when there is a need for a common base class with shared code and state among derived classes.
* Allows for a mix of abstract and non-abstract methods.
* Interface:
* Used when defining a contract that multiple unrelated classes can implement.
* Supports achieving multiple inheritance through interfaces.

## Difference b/w final, finally and finalize?

**final:**

* Java and C#: In Java and C#, the final keyword is used to declare a variable, method, or class. Its meaning depends on where it is applied:
* For variables, it indicates that the variable cannot be reassigned (constant).
* For methods, it signifies that the method cannot be overridden by subclasses.
* For classes, it indicates that the class cannot be subclassed.

final int constantValue = 10;

class FinalMethod {

final void doSomething() {

// Method implementation

}

}

final class FinalClass {

// Class members

}

**finally:**

* The finally block is used in exception handling to specify a block of code that will be executed regardless of whether an exception is thrown or not. It is part of the try-catch-finally structure. The code in the finally block is guaranteed to execute, providing a way to clean up resources or perform necessary actions.

try {

// Code that may throw an exception

} catch (Exception e) {

// Exception handling

} finally {

// Code in this block will always execute

// regardless of whether an exception occurred or not

}

**finalize:**

* Java: The finalize method is a method in the Object class in Java. It is called by the garbage collector before an object is reclaimed for memory. However, it's generally not recommended to rely on the finalize method for resource cleanup, as its execution is not guaranteed and may lead to resource leaks.

Java:

public class MyClass {

// Other class members

@Override

protected void finalize() throws Throwable {

// Cleanup code

super.finalize();

}

}

C#:

public class MyClass : IDisposable {

// Other class members

public void Dispose() {

// Cleanup code

}

}

using (var myObject = new MyClass()) {

// Code that uses myObject

} // Dispose method is automatically called when exiting the using block

## Use of Sealed class in C#?

In C#, the sealed keyword is used to restrict the inheritance of a class or the override of a method. It can be applied to classes, methods, or properties. Here are two common uses of the sealed keyword:

**Sealing a Class:** When the sealed keyword is applied to a class, it means that the class cannot be used as a base class for other classes. It prevents other classes from inheriting from it.

sealed class SealedClass

{

// Class members and implementation

}

**Sealing a Method:** When the sealed keyword is applied to a method within a class, it means that the method cannot be overridden by derived classes. This is applicable when the base class provides a specific implementation that should not be changed by subclasses.

class BaseClass

{

// Regular method

public void NormalMethod()

{

// Implementation

}

// Sealed method

public sealed void SealedMethod()

{

// Implementation

}

}

class DerivedClass : BaseClass

{

// This is not allowed due to sealing

// public override void SealedMethod() { }

}

# State Streat 1st Round Interview Questions

## Reverse the below string word wise

## String s = “Vikash Kumar,Bangalore”;

## Create a Dictionary and store Vikash as key 1, Kumar as key 2 and Bangalore as key 3. Take these all key value from consoles and print value bases on key.

## Find Second Duplicate From Array

## Identify missing elements in sequential array

# Logical Questions

## W.A.P to reverse a String?

static string Reverse(string str)

{

string reverse\_str = "";

for(int i = str.Length - 1; i >= 0; i--)

{

reverse\_str = reverse\_str + str[i];

}

return reverse\_str;

}

## W.A.P to check a String is palindrome or not?

static string Reverse(string str)

{

string reverse\_str = "";

for(int i = str.Length - 1; i >= 0; i--)

{

reverse\_str = reverse\_str + str[i];

}

return reverse\_str;

}

static string CheckPalindromString(string str1, string str2)

{

if (str1.Equals(str2))

{

return str1 + " : is a Palindrome String";

}

else

return str1 + " : is not a Palindrome String";

}

## W.A.P to check a String is Anagram or not?

public static bool areAnagram(string str1, string str2)

{

if (str1.Length == str2.Length)

{

char[] charArr1 = str1.ToCharArray();

char[] charArr2 = str2.ToCharArray();

System.Array.Sort(charArr1);

System.Array.Sort(charArr2);

for (int i = 0; i < charArr1.Length; i++)

{

if (charArr1[i] != charArr2[i])

{

Console.WriteLine(str1 + " : is not anagram");

return false;

}

}

return true;

}

else

return false;

}

## W.A.P to count of the string from given string?

static void Main()

{

string s1 = " Find number of words in given string ";

s1 = s1.Trim();

int count = 0;

for(int i = 0; i < s1.Length; i++)

{

if (s1[i]==' ' && Char.IsLetter(s1[i+1]))

count++;

}

Console.WriteLine(++count);

}

## W.A.P to Swap to Number and String with and without 3rd variable ?

static void SwapTwoNumbersWithoutTempVar()

{

int x = 10; int y = 20;

Console.WriteLine("X : value before swapping " + x);

Console.WriteLine("Y : value before swapping " + y);

x = y - x; //10

y = y - x;

x = x + y;

Console.WriteLine("X : value after swapping " + x);

Console.WriteLine("Y : value after swapping " + y);

}

static void SwapTwoNumbersWithTempVar()

{

int x = 20; int y = 10;

int temp = 0;

Console.WriteLine("X : value before swapping " + x);

Console.WriteLine("Y : value before swapping " + y);

temp = x;

x = y;

y = temp;

Console.WriteLine("X : value after swapping " + x);

Console.WriteLine("Y : value after swapping " + y);

}

static void SwapTwoStringsWithoutTempVar()

{

string s1 = "Vikash";

string s2 = "Kumar";

Console.WriteLine("s1 : value before swapping " + s1);

Console.WriteLine("s2 : value before swapping " + s2);

s1 = s1 + s2; //HelloWorld

s2 = s1.Substring(0,s1.Length - s2.Length); //Hello

s1 = s1.Substring(s2.Length);

Console.WriteLine("s1 : value after swapping " + s1);

Console.WriteLine("s2 : value after swapping " + s2);

}

static void SwapTwoStringsWithTempVar()

{

string s1 = "Hello";

string s2 = "World";

string temp = "";

Console.WriteLine("s1 : value before swapping " + s1);

Console.WriteLine("s2 : value before swapping " + s2);

temp = s1;

s1 = s2;

s2 = temp;

Console.WriteLine("s1 : value after swapping " + s1);

Console.WriteLine("s2 : value after swapping " + s2);

}

## W.A.P to multiple 2 number without \* operator?

int x = Convert.ToInt32 (Console.ReadLine());

int y = Convert.ToInt32 (Console.ReadLine());

int sum = 0;

for(int i = 1; i <= y; i++)

{

sum+= x;

}

Console.WriteLine(sum);

## W.A.P to get prime number?

static void Main()

{

//Write script to get prime numbers

int n = Convert.ToInt32 (Console.ReadLine());

for(int i = 0; i < n; i++)

{

if(isPrime(i))

Console.WriteLine(i);

}

}

static bool isPrime(int n)

{

//since 0 and 1 is not prime return false.

if (n == 1 || n == 0) return false;

//Run a loop from 2 to n-1

for (int i = 2; i < n; i++)

{

// if the number is divisible by i, then n is not a prime number.

if (n % i == 0) return false;

}

//otherwise, n is prime number.

return true;

}

## W.A.P to reverse a number?

static void Main()

{

int num = 123;

int rem,sum=0;

while(num > 0)

{

rem = num%10;

sum = (sum \* 10) + rem;

num /= 10;

}

Console.WriteLine(sum);

}

## W.A.P to check string is numeric or not?

## W.A.P to count the vowels and consonants from a given string?

## W.A.P to reverse the case of each characters of a string?

## W.A.P to generate random string?

## W.A.P to capitalize the first latter of each word in a string?

## W.A.P to find occurrences of characters in a String

string s1 = "Java is awesome";

char c = 'a';

int frequency = 0;

for (int i=0;i<s1.Length;i++)

{

if (s1[i]==c)

frequency++;

}

Console.WriteLine(frequency);

## W.A.P to find the count of Capital and small latter in a string?

string str1 = "Welcome";

int smallCount = 0;

int capitalCount = 0;

for(int i=0; i<str1.Length; i++) {

if (str1[i] >= 'A' && str1[i] <= 'Z')

{

capitalCount++;

}

else if (str1[i] >= 'a' && str1[i] <= 'z')

smallCount++;

}

Console.WriteLine(capitalCount);

Console.WriteLine(smallCount);

## W.A.P to remove duplicate character from given string?

static string RemoveDuplicates(string input)

{

// Create a HashSet to store unique characters

HashSet<char> uniqueChars = new HashSet<char>();

// Create a string builder to build the result string

StringBuilder resultStringBuilder = new StringBuilder();

// Iterate through each character in the input string

foreach (char character in input)

{

// If the character is not in the HashSet, add it to the HashSet and result string

if (uniqueChars.Add(character))

{

resultStringBuilder.Append(character);

}

}

return resultStringBuilder.ToString();

## W.A.P to print the factorial of a given number?

static void Main()

{

int num = 7; //1\*2\*3 = 6

int sum = 1;

while(num>0) {

sum = sum \* num;

num--;

}

Console.WriteLine(sum);

}

## W.A.P to print a Fibonacci series of a given number?

## W.A.P to count the number, alphabet and special characters?

## W.A.P to find 2nd large element in array?

# Agile Interview Questions

## ****What is Agile Testing?****

Agile testing is a software testing practice that follows the principles of agile software development. It is an iterative software development methodology where requirements keep changing as per the customer needs.  Testing is done in parallel to the development of an iterative model. Test team receives frequent code changes from the development team for testing an application.

## ****What is Agile Manifesto?****

Agile manifesto defines 4 key points:

* Individuals and interactions over process and tools.
* Working software over comprehensive documentation.
* Customer collaboration over contract negotiation.
* Responding to change over following a plan.

## ****What are the principles of Agile Software Development?****

* Highest priority is to satisfy the customer through early and continuous delivery of business valuable software
* Welcome changing requirements, even late in development
* Deliver working software frequently
* Business people and developers must work together with transparency on daily basis throughout the project
* Build projects around motivated individuals
* The best form of communication is to do face-to-face conversation
* Working software is the primary measure of progress
* Able to maintain a constant pace
* Continuous attention to technical excellence
* Simplicity – the art of maximizing the amount of work not done – is essential
* Self-organizing teams
* At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly

## What are the main roles in Scrum?

Scrum consists of three main roles:

**Product Owner**: Product Owner usually represents the Client and acts as a point of contact from the Client side. The one who prioritizes the list of Product Backlogs which Scrum Team should finish and release.

**Scrum** **Master**: Scrum Master acts as a facilitator to the Scrum Development Team. Clarifies the queries and organizes the team from distractions and teach the team how to use scrum and also concentrates on Return on Investment (ROI). Responsible for managing the sprint.

**Scrum** **Development** Team: Developer’s, QA’s. Who develops the product. Scrum development team decides the effort estimation to complete a Product Backlog Item.

**Scrum** **Team**: A cross-functional, self-organizing group of dedicated people (Group of Product Owner, Business Analyst, Developer’s and QA’s). Recommended size of a scrum team is 7 plus or minus 2 (i.e, between 5 to 9 members in a team).

## What approach do you follow when requirements change continuously?

In Agile methodology, change in requirement is possible. It’s not like other traditional methodologies where the requirements are locked down at the requirement phase. Every team member should be ready to handle the changes in the project.

The team should work closely with the Product Owner to understand the scope of requirement change and to negotiate to keep the requirement changes to a minimum or to adopt those changes in the next sprint. Based on the requirement changes Test Team could update the Test Plan and Test Cases to achieve the deadlines. The team should understand the risk in the requirement change and prepare a contingency plan. It is a best practice not to go for the automation process until requirements are finalized.

## How is Agile Testing different from other traditional Software Development Models?

It is one of the common Agile Testing Interview Questions.

In Agile Methodology, testing is not a phase like other traditional models. It is an activity parallel to development in the Agile. The time slot for the testing is less in the Agile compared to the traditional models. The testing team works on small features in Agile whereas the test team works on a complete application after development in the traditional models.

## What is a Sprint?

In Scrum, the project is divided into Sprints. Each Sprint has a specified timeline (2 weeks to 1 month). This timeline will be agreed by a Scrum Team during the Sprint Planning Meeting. Here, User Stories are split into different modules. The end result of every Sprint should be a potentially shippable product.

## What are Product Backlog and Sprint Backlog?

**Product** **Backlog**: Product Backlog is a repository where the list of Product Backlog Items stored and maintained by the Product Owner. The list of Product Backlog Items are prioritized by the Product Owner as high and low and also could re-prioritize the product backlog constantly.

**Sprint** **Backlog**: Group of user stories which scrum development team agreed to do during the current sprint (Committed Product Backlog items). It is a subset of the product backlog.

## What is a Daily Stand-up Meeting?

Daily Stand-up Meeting is a daily routine meeting. It brings everyone up to date on the information and helps the team to stay organized.

Each team member reports to the peers the following:

What did you complete yesterday?

Any impediments in your way?

What do you commit to today?

When do you think you will be done with that?

## W.A.P to print vowel character from given string?

## What is a Sprint Planning Meeting?

The first step of Scrum is the Sprint Planning Meeting where the entire Scrum Team attends. Here the Product Owner selects the Product Backlog Items (User Stories) from the Product Backlog.

Most important User Stories at the top of the list and least important User Stories at the bottom. Scrum Development Team decides and provides effort estimation.

## What is a Sprint Review Meeting?

In the Sprint Review Meeting, Scrum Development Team presents a demonstration of a potentially shippable product. Product Owner declares which items are completed and not completed. Product Owner adds the additional items to the product backlog based on the stakeholder’s feedback.

## What is a Sprint Retrospective Meeting?

Scrum Team meets again after the Sprint Review Meeting and documents the lessons learned in the earlier sprint such as “What went well”, “What could be improved”. It helps the Scrum Team to avoid the mistakes in the next Sprints.

## What is a Task Board?

A task board is a dashboard which illustrates the progress that an agile team is making in achieving their sprint goals.

In general, the columns used in a task board are as follows

* User Story: Actual Business Requirement (Description)
* To Do: All the tasks of current sprint
* In Progress: Any task being worked on
* To Verify: Tasks pending for verification
* Done: Tasks which are completed

# Automation Interview Questions

## Selenium WebDriver:

### What are the different types of locators in Selenium?

### How do you use XPath over CSS locator?

### How to get a specific value from a dropdown and reuse it in verification?

### How do we use JavaScript Executors?

### It is possible to validate Captcha using Selenium. If yes, how?

### What should be the ideal way to store data using Selenium WebDriver only?

### It is possible to use XPath like parent/child/node/. ? If yes, provide an example where to use it?

### What happens if you receive browser notification during test automation execution?

### Why does State Element Exception occur, and how to handle?

### What is Invalid Certificate Exception?

## APIS:

### What are the component of an HTTP request?

Below are the components of HTTP Request:

* Request URL, Request method, Request status code.
* Zero or more Request Headers.
* Request Body (optional)

**Request URL, Request method, Request status code.**

Let us open chrome browser and launch  
<https://opensource-demo.orangehrmlive.com/index.php/auth/login>



Press F12 key and click ‘Network’ section as shown below

A screen shot of a computer

AI-generated content may be incorrect.

Right now you would NOT see any HTTP requests (see yellow section in above figure).

 Refresh the website or press Ctrl+R. You would now see lot of HTTP requests in the Network section as shown below

A screenshot of a computer

AI-generated content may be incorrect.

Select ‘login’ request

A screenshot of a computer

AI-generated content may be incorrect.

On the right hand side, you would see ‘Headers’ section comprising of

* General
* Request Headers
* Response Headers

Let us expand the ‘General’ section

A screenshot of a computer

AI-generated content may be incorrect.

The ‘General’ section has information about

* Request URL (this matches the url in the address bar), this is the same url that client has requested;
* Request Method is of type GET (since the client is requesting a read-only information from server, client is NOT posting any data on server);
* Request status code is 200 since the server was successful in serving the client request.

**Request Header**

Let us now expand the ‘Request Headers’ as shown below

A screenshot of a computer

AI-generated content may be incorrect.

The header information can also be sent behind the scenes (automatically) along with the HTTP request. For example, as can be seen above, the cookie information is sent to the server (as part of http request header), similarly User-Agent information is sent and so on.

**Request Body**

Request body is optional that may or may not be submitted as part of request. The body could be either in the json or xml format.

As of now we haven’t sent any login credentials (Username and Password) along with the HTTP request

A screenshot of a login screen

AI-generated content may be incorrect.

That’s the reason, we do not yet see the ‘Payload’ section (see below). Payload is same as the body section



Let us now enter the Username/Password as Admin/admin123

A screenshot of a login page

AI-generated content may be incorrect.

Click LOGIN. After successful login, look for the validateCredentials http request as shown below

A screenshot of a computer

AI-generated content may be incorrect. On the right hand side, you will now see the body section ‘Payload’ as shown above. This section shows us the information (username, password) that we had sent as part of request body.

**Components of HTTP response**

Once client hits the request, the server processes the request and will send the HTTP response back to the client. So the HTTP response contains the data or information requested by the client.

HTTP Response also has the similar set of elements:

* Response Header
* Response Body (optional)

Let us understand these components one by one.

**Response Header**

Let us hit the url <https://jsonplaceholder.typicode.com/posts/4>

We will get the below response

A screenshot of a computer

AI-generated content may be incorrect.

Press F12 and refresh the page. Expand ‘Response Headers’. Notice below that we can see the server side header information like the server name etc

A screenshot of a computer

AI-generated content may be incorrect.

**Response Body**

See below. We have the ‘Response’ section that shows the exact same information which we get when we hit the url

A screenshot of a computer

AI-generated content may be incorrect.

Let us change the request url [https://jsonplaceholder.typicode.com/posts/**9**](https://jsonplaceholder.typicode.com/posts/9)

Notice below that the body of the response changes

A screenshot of a computer

AI-generated content may be incorrect.

**HTTP Methods**

The client hits the request URL to perform some operation on the server. The HTTP methods or operations that can be performed on the server are:

* Create new data on the server (HTTP POST method)
* Update the data on the server  (HTTP PUT method)
* Read data from server (HTTP GET method)
* Delete the data from the server (HTTP DELETE method)

These 4 methods are also known as CURD operations (Create, Update, Read, and Delete).

Let us discuss about Read and Create operations.

**HTTP GET method (Read)**

The GET request is used to access some information from the server. The GET request does not modify any data on the server. If the requested data is found on the server, the server returns ‘200 OK’ status code as part of response body

A screenshot of a computer

AI-generated content may be incorrect.

If the requested data is not found on the server, the server returns ‘404 (Not Found)’ status code as part of response body, see below. Try to access <https://jsonplaceholder.typicode.com/posts/444>

A screenshot of a computer

AI-generated content may be incorrect.

**HTTP POST method (Create)**

Whenever we create/post a new comment on twitter/facebook/linkedin, we are basically making an HTTP POST request that creates new data on the server.

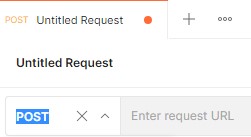
If the data is successfully created on the server, the server returns ‘201 Created’ status code as part of response body. To simulate the POST request, download, install and launch postman (postman is used to manually test the APIs, we will discuss in detail about the postman later).

Click + to create a new api request as shown below

A screenshot of a computer

AI-generated content may be incorrect.

Select POST from the dropdown since we want to post new data to the server



Launch <https://jsonplaceholder.typicode.com/posts/>

Scroll down to the page end, notice that this page contains data up to id:100 as shown below

A blue rectangle with white text

AI-generated content may be incorrect.

We will post and create a new id.

Copy Request URL and paste it in ‘Enter request URL’ field



 Next, select Body > raw > JSON as shown below

A screenshot of a computer

AI-generated content may be incorrect.

Next, copy the below json section

A blue rectangle with white text

AI-generated content may be incorrect.

Paste it in the body section

A screenshot of a chat

AI-generated content may be incorrect.

Change id to 101 as shown below

A screenshot of a web page

AI-generated content may be incorrect.

A screenshot of a phone

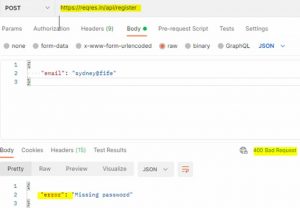
AI-generated content may be incorrect.

Click SEND

Notice the response body status ‘201 Created’



If the post request fails/unsuccessful, than the status ‘400 Bad Request’ is shown. See below, we get ‘Missing password’ error (the request url is different)



So, this was quick glimpse of GET/POST requests.

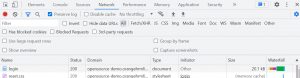
**Just a side note:** The network timelines section takes the extra space on our webpage. Since we don’t need the network timelines section, we can disable the same. To do that, click the ‘Network Settings’ gear icon



By default, ‘Show overview’ checkbox is checked



Just uncheck ‘Show overview’ checkbox. The timeline would disappear as shown below



Click ‘Network Settings’ gear icon again to hide the greyed-out checkboxes that you see above

A screenshot of a computer

AI-generated content may be incorrect.

### What is difference between API and unit testing?

### What is an HTTP response?

### How can we add validation points in Postman?

### What do you understand by server-side validation?

### What is 3-tier architecture?

### What is difference between Web services and Web APIS?

### What is REST, SOAP and GraphQL in APIs?

### What do you test in standalone APIs?

### What do you test in 3rd-party integrated APIs?

## Postman:

### What is use of collection, environment, and global variables?

### How to execute collection end-to-end?

### How to validate that API response has the correct status code?

### What happens when an API response return from Data instead of JSON, and how to validate it?

### How to set an Basic Auth in Postman?

### How do you store environment credentials?

### How to save a demo response for an API request?

### How will you validate an API request If VPN is required for it to work?

### How do you filter results in an API request using postman?

### How do you setup custom headers in postman?

## Git:

### What are the different stages in committing the code to GitHub?

### It is possible to revert changes in remote repository? If yes, how?

### When do you commit your code? After committing, how di you validate that everyone has the update?

### How to merge stashed changes in local repository?

### Why do we need a gitignore file? How do you add files to it?

## TestNG:

### What is the execution format of tests in TestNG?

### Can priority be negative for methods? Yes, what is the execution flow as per priority?

### What is the difference between dependsOnMethods and dependsOnGroups?

### What are the difference ways to exclude tests in TestNG?

### What does threadPoolSize mean in TestNG, and how does it work?

### Why do we need @BeforeSuite and @AfterSuite annotations?

## Difference between Asser and Verify

Asserts halt execution upon failure, while verify statements allow the test to continue, even if a verification condition is not met.

Assert (Hard Assert):

* If an assert statement fails, the test execution stops immediately, and the test case is marked as failed.
* The remaining steps in the test are not executed.
* Asserts are used for critical validations where the test case should fail if the condition is not met.

Verify (Soft Assert):

* If a verify statement fails, the test execution continues, and the test case is marked as failed, but the remaining steps are still executed.
* Verify statements are used for non-critical checks or when it's important to execute all tests, even if some conditions are not met.
* In TestNG, soft assertions are used to implement verify functionality.