**Section 1: Selenium Introduction & Basics (Q1–Q20)**

1. **Selenium भनेको के हो?**  
   वेब ब्राउजर Automation गर्ने Open-Source Tool Suite हो।
2. **Selenium का मुख्य Components के-के हुन्?**  
   Selenium IDE, Selenium RC (deprecated), Selenium WebDriver, Selenium Grid।
3. **Selenium WebDriver के हो?**  
   Web Browser Control गर्नको लागि API, जुन Direct Browser संग Communicate गर्छ।
4. **Selenium किन प्रयोग गर्ने?**  
   Cross-Browser Testing, Multiple Programming Language Support, Free/Open Source।
5. **Selenium कुन Programming Languages support गर्छ?**  
   Java, Python, C#, Ruby, JavaScript, Kotlin आदि।
6. **Selenium कुन Browsers support गर्छ?**  
   Chrome, Firefox, Edge, Safari, Opera आदि।
7. **Selenium कुन Operating Systems मा चल्छ?**  
   Windows, macOS, Linux, Unix।
8. **Selenium IDE के हो?**  
   Browser Extension जसले Record & Playback Testing गर्न मद्दत गर्छ।
9. **Selenium Grid के हो?**  
   Multiple Machines/Browsers मा Parallel Test Execution गर्न प्रयोग हुने Tool।
10. **Selenium WebDriver र IDE बीच फरक के हो?**  
    WebDriver – Code Based, Flexible; IDE – Record & Playback, Limited।
11. **Selenium को Limitation के हो?**  
    Desktop Application Automation गर्न सक्दैन।
12. **Selenium को Test Execution कुन Browser मा Default हुन्छ?**  
    Default Browser, प्रायः Firefox (Old Version मा)।
13. **WebDriver को Basic Architecture कस्तो छ?**  
    Test Script → Language Bindings → JSON Wire Protocol → Browser Driver → Browser।
14. **JSON Wire Protocol के हो?**  
    WebDriver र Browser बीच Communication Protocol।
15. **Browser Driver के हो?**  
    WebDriver बाट आउने Command Browser मा Execute गर्ने Middle Layer।
16. **ChromeDriver के हो?**  
    Chrome Browser Automation गर्न ChromeDriver executable।
17. **GeckoDriver के हो?**  
    Firefox Browser Automation गर्न GeckoDriver executable।
18. **WebDriver Manager के हो?**  
    Browser Drivers Automatically Manage गर्ने Library (Java मा io.github.bonigarcia)।
19. **Selenium Standalone Server कहिले प्रयोग गर्ने?**  
    Selenium Grid वा Remote Execution को लागि।
20. **Selenium र QTP/UFT बीच फरक?**  
    Selenium Open-Source र Web आधारित; QTP/UFT Paid र Web/Desktop दुवै।

**Section 2: Locators in Selenium (Q21–Q40)**

1. **Locator भनेको के हो?**  
   WebElement पहिचान गर्न प्रयोग हुने तरिका।
2. **Selenium मा Locators का प्रकार?**  
   ID, Name, Class Name, Tag Name, Link Text, Partial Link Text, CSS Selector, XPath।
3. **सबैभन्दा तेज Locator कुन हो?**  
   ID (यदि Available छ भने)।
4. **Name Locator कहिले प्रयोग गर्ने?**  
   जब Unique Name Attribute छ।
5. **Class Name Locator को Limit के हो?**  
   Multiple Classes भए Error आउन सक्छ।
6. **Tag Name Locator कहिले प्रयोग गर्ने?**  
   Same Type Elements (जस्तै: <input>) फेला पार्न।
7. **Link Text Locator कहिले प्रयोग गर्ने?**  
   Exact Visible Text भएको Link Locate गर्न।
8. **Partial Link Text Locator कहिले प्रयोग गर्ने?**  
   Link Text लामो हुँदा Partial Match गर्न।
9. **CSS Selector के हो?**  
   HTML Element Locate गर्न CSS Syntax प्रयोग गर्ने तरिका।
10. **CSS Selector vs XPath फरक?**  
    CSS तेज; XPath Flexible (DOM Traverse गर्न सकिन्छ)।
11. **CSS Selector मा ID कसरी प्रयोग गर्ने?**  
    #idValue
12. **CSS Selector मा Class कसरी प्रयोग गर्ने?**  
    .classValue
13. **CSS Selector मा Attribute कसरी प्रयोग गर्ने?**  
    [attribute='value']
14. **XPath के हो?**  
    XML Path Language, DOM Traverse गरेर Element Locate गर्ने तरिका।
15. **Absolute XPath के हो?**  
    Root Node बाट सुरु हुने Path (/html/body/...)।
16. **Relative XPath के हो?**  
    Current Node बाट सुरु हुने Path (//tagname[@attr='value'])।
17. **XPath मा OR Condition कसरी प्रयोग गर्ने?**  
    //tag[@attr='value1' or @attr='value2']
18. **XPath मा AND Condition कसरी प्रयोग गर्ने?**  
    //tag[@attr1='val1' and @attr2='val2']
19. **XPath मा text() कसरी प्रयोग गर्ने?**  
    //tag[text()='value']
20. **XPath मा contains() कसरी प्रयोग गर्ने?**  
    //tag[contains(text(),'partialText')]

**Section 3: WebElement Methods (Q41–Q60)**

1. **WebElement भनेको के हो?**  
   HTML Element जसलाई Selenium ले Represent गर्छ।
2. **click() method के हो?**  
   Element क्लिक गर्ने।
3. **sendKeys() method के हो?**  
   Text Input Field मा Value टाइप गर्ने।
4. **clear() method के हो?**  
   Input Field को Value हटाउने।
5. **getText() method के हो?**  
   Element को Visible Text फिर्ता गर्ने।
6. **getAttribute() method के हो?**  
   HTML Attribute को Value फिर्ता गर्ने।
7. **isDisplayed() method के हो?**  
   Element Page मा देखिन्छ कि छैन Check गर्ने।
8. **isEnabled() method के हो?**  
   Element Enabled छ कि छैन Check गर्ने।
9. **isSelected() method के हो?**  
   Checkbox/Radio Button Selected छ कि छैन Check गर्ने।
10. **submit() method के हो?**  
    Form Submit गर्ने।
11. **getCssValue() method के हो?**  
    CSS Property को Value फिर्ता गर्ने।
12. **getTagName() method के हो?**  
    HTML Tag को नाम फिर्ता गर्ने।
13. **findElement() method के हो?**  
    Single Element Locate गर्ने।
14. **findElements() method के हो?**  
    Multiple Elements Locate गरेर List फिर्ता गर्ने।
15. **getSize() method के हो?**  
    Element को Height र Width फिर्ता गर्ने।
16. **getLocation() method के हो?**  
    Element को Position (X,Y Coordinates) फिर्ता गर्ने।
17. **getRect() method के हो?**  
    Size र Location दुवै फिर्ता गर्ने।
18. **getScreenshotAs() method के हो?**  
    Element को Screenshot लिने।
19. **Actions Class को clickAndHold() method के हो?**  
    Mouse Button Hold गर्ने।
20. **Actions Class को release() method के हो?**  
    Hold गरिएको Mouse Button छोड्ने।

**Section 4: Waits & Browser Handling (Q61–Q80)**

1. **Implicit Wait के हो?**  
   Driver ले Element Available हुने समयसम्म Wait गर्ने Default Time सेट गर्ने।
2. **Explicit Wait के हो?**  
   Specific Condition पुरा नभएसम्म Wait गर्ने।
3. **Fluent Wait के हो?**  
   Polling Frequency सेट गरेर Wait गर्ने।
4. **Thread.sleep() Selenium मा प्रयोग गर्न मिल्छ?**  
   मिल्छ तर Recommended छैन, किनकि Fixed Wait हुन्छ।
5. **WebDriverWait class के हो?**  
   Explicit Wait Implement गर्न प्रयोग हुने Class।
6. **ExpectedConditions class के हो?**  
   Common Wait Conditions Define गर्ने Helper Class।
7. **titleIs() ExpectedCondition के हो?**  
   Page Title Exact Match Check गर्ने।
8. **urlContains() ExpectedCondition के हो?**  
   URL मा Specific Text छ कि छैन Check गर्ने।
9. **elementToBeClickable() ExpectedCondition के हो?**  
   Element Visible र Enabled दुवै भए मात्र Return गर्ने।
10. **alertIsPresent() ExpectedCondition के हो?**  
    Alert Box Present छ कि छैन Check गर्ने।
11. **Browser Window कसरी Maximize गर्ने?**  
    driver.manage().window().maximize();
12. **Window Handle के हो?**  
    प्रत्येक Browser Window को Unique ID।
13. **getWindowHandle() method के हो?**  
    Current Window को Handle फिर्ता गर्ने।
14. **getWindowHandles() method के हो?**  
    सबै Open Windows को Handles फिर्ता गर्ने।
15. **switchTo().window() method के हो?**  
    अर्को Window मा Switch गर्ने।
16. **switchTo().frame() method के हो?**  
    iFrame मा Switch गर्ने।
17. **switchTo().defaultContent() method के हो?**  
    Main Document मा फर्किने।
18. **switchTo().parentFrame() method के हो?**  
    One Level Up Frame मा जाने।
19. **switchTo().alert() method के हो?**  
    Alert Box Handle गर्न।
20. **alert.accept() method के हो?**  
    Alert OK/Yes बटन क्लिक गर्ने।

**Section 5: Common Automation Tasks (Q81–Q100)**

1. **alert.dismiss() method के हो?**  
   Alert Cancel/No बटन क्लिक गर्ने।
2. **alert.getText() method के हो?**  
   Alert को Text फिर्ता गर्ने।
3. **alert.sendKeys() method के हो?**  
   Prompt Alert मा Text इनपुट गर्ने।
4. **Select Class के हो?**  
   Dropdown Handle गर्न प्रयोग हुने Class।
5. **selectByVisibleText() method के हो?**  
   Dropdown मा Visible Text अनुसार Option Select गर्ने।
6. **selectByValue() method के हो?**  
   Option को value attribute अनुसार Select गर्ने।
7. **selectByIndex() method के हो?**  
   Index अनुसार Dropdown Option Select गर्ने।
8. **getOptions() method (Select Class) के हो?**  
   सबै Dropdown Options फिर्ता गर्ने।
9. **getFirstSelectedOption() method के हो?**  
   पहिलो Selected Option फिर्ता गर्ने।
10. **deselectAll() method के हो?**  
    सबै Selected Options Unselect गर्ने (Multi-Select मा मात्र)।
11. **Robot Class Selenium मा किन प्रयोग गर्ने?**  
    Keyboard/Mouse Events Handle गर्न।
12. **TakesScreenshot interface के हो?**  
    Screenshot Capture गर्न प्रयोग हुने Interface।
13. **getScreenshotAs() (Driver) के हो?**  
    Browser को Screenshot Capture गर्ने।
14. **File Upload कसरी गर्ने?**  
    sendKeys() मार्फत File Path पठाएर।
15. **Drag and Drop कसरी गर्ने?**  
    Actions class को dragAndDrop() method प्रयोग गरेर।
16. **Mouse Hover कसरी गर्ने?**  
    moveToElement() method प्रयोग गरेर।
17. **Double Click कसरी गर्ने?**  
    doubleClick() method प्रयोग गरेर।
18. **Right Click कसरी गर्ने?**  
    contextClick() method प्रयोग गरेर।
19. **Page Scroll कसरी गर्ने?**  
    JavaScriptExecutor को executeScript() प्रयोग गरेर।
20. **JavaScriptExecutor Selenium मा के हो?**  
    Browser मा JavaScript Code Execute गर्न प्रयोग हुने Interface।

**Section 6: Page Object Model (POM) & Framework Basics (Q101–Q120)**

1. **Page Object Model (POM) भनेको के हो?**  
   एक Design Pattern जसमा प्रत्येक Web Page को लागि अलग Class बनाइन्छ जसले Elements र Actions Represent गर्छ।
2. **POM किन प्रयोग गर्ने?**  
   Code Reusability, Maintainability, र Readability बढाउन।
3. **POM मा Page Class के हुन्छ?**  
   Elements (Locators) र Actions (Methods) को संग्रह।
4. **PageFactory के हो?**  
   Selenium को Helper Class जसले @FindBy Annotation प्रयोग गरेर Elements Initialize गर्छ।
5. **@FindBy Annotation के हो?**  
   WebElement को Locator Define गर्ने Annotation।
6. **PageFactory.initElements() method के हो?**  
   Page Class का Elements Initialize गर्ने Method।
7. **POM र PageFactory बीच फरक?**  
   POM Concept हो; PageFactory Selenium को Implementation Helper हो।
8. **Single Responsibility Principle (POM) मा कसरी लागू हुन्छ?**  
   प्रत्येक Page Class ले केवल त्यस Page को Elements र Actions Handle गर्ने।
9. **Test Class मा Page Object कसरी प्रयोग गर्ने?**  
   Page Class को Object बनाएर Method Call गर्ने।
10. **POM मा Constructor किन प्रयोग गर्ने?**  
    Driver र Elements Initialize गर्न।
11. **POM बिना Test Script को कमजोरी के हो?**  
    Duplicate Code, Maintenance Problem।
12. **POM + TestNG/JUnit एकसाथ किन प्रयोग गर्ने?**  
    Structured Framework + Test Execution Control पाउन।
13. **Reusable Methods Framework मा कसरी बनाउने?**  
    Utility Classes मा Common Actions Define गर्ने।
14. **Base Class Framework मा के हो?**  
    Common Setup, Teardown, Configurations राख्ने Class।
15. **Config Properties File के हो?**  
    URL, Credentials, Timeout जस्ता Configurations External File मा राख्ने।
16. **Config Data कसरी Load गर्ने?**  
    Java Properties Class प्रयोग गरेर।
17. **POM मा Data Driven Testing कसरी गर्ने?**  
    Excel, CSV, JSON वा Database बाट Test Data Fetch गर्ने।
18. **Framework मा Constants Class किन बनाउने?**  
    Fixed Values Centralize गर्न।
19. **Framework Folder Structure मा के हुन्छ?**  
    Pages, Tests, Utils, Config, Reports, Drivers आदि।
20. **POM Framework को Maintenance कस्तो हुन्छ?**  
    Page मा Change भए Page Class मात्र Update गर्नु पर्छ, Test Script untouched रहन्छ।

**Section 7: TestNG & JUnit Integration (Q121–Q140)**

1. **TestNG के हो?**  
   Testing Framework जसले Test Configuration, Grouping, Parallel Execution Support गर्छ।
2. **TestNG र JUnit बीच फरक?**  
   TestNG – Advanced Features (DataProvider, Dependency), JUnit – Basic Unit Testing।
3. **TestNG मा @Test Annotation के हो?**  
   Method लाई Test Method Define गर्ने Annotation।
4. **TestNG मा @BeforeMethod के हो?**  
   प्रत्येक Test Method अघि Run हुने Method।
5. **TestNG मा @AfterMethod के हो?**  
   प्रत्येक Test Method पछि Run हुने Method।
6. **TestNG मा @BeforeClass के हो?**  
   Class को Test सुरु हुनु अघि एकपटक चल्ने।
7. **TestNG मा @AfterClass के हो?**  
   Class को Test सकिएपछि एकपटक चल्ने।
8. **TestNG मा @BeforeTest के हो?**  
   <test> Tag भित्रको सबै Test सुरु हुनु अघि चल्ने।
9. **TestNG मा @AfterTest के हो?**  
   <test> Tag भित्रको सबै Test सकिएपछि चल्ने।
10. **TestNG मा @BeforeSuite के हो?**  
    Test Suite सुरु हुनु अघि एकपटक चल्ने।
11. **TestNG मा @AfterSuite के हो?**  
    Test Suite सकिएपछि एकपटक चल्ने।
12. **TestNG मा testng.xml के हो?**  
    Test Configuration XML File जसले Test Classes, Groups, Parameters Define गर्छ।
13. **TestNG मा Groups के हो?**  
    Similar Test Cases लाई Categorize गर्ने।
14. **TestNG मा Dependency कसरी Define गर्ने?**  
    dependsOnMethods वा dependsOnGroups प्रयोग गरेर।
15. **TestNG मा DataProvider के हो?**  
    Test Method लाई Multiple Sets of Data Provide गर्ने।
16. **DataProvider को Return Type के हुन्छ?**  
    Object[][] Array।
17. **TestNG मा Parameters कसरी Pass गर्ने?**  
    testng.xml मा Define गरेर वा @Parameters Annotation प्रयोग गरेर।
18. **TestNG मा Retry Analyzer के हो?**  
    Failed Test पुन: Run गर्ने Mechanism।
19. **TestNG मा Parallel Execution कसरी Enable गर्ने?**  
    testng.xml मा parallel="methods/tests/classes" Set गर्ने।
20. **TestNG मा Priority कसरी Set गर्ने?**  
    @Test(priority = 1) जस्तो Define गरेर।

**Section 8: Selenium Grid & Parallel Execution (Q141–Q160)**

1. **Selenium Grid के हो?**  
   Multiple Machines/Browsers मा Tests Parallel चलाउने Tool।
2. **Selenium Grid मा Hub के हो?**  
   Central Machine जसले Tests Manage गर्छ।
3. **Selenium Grid मा Node के हो?**  
   Hub संग Connect भएको Machine जसमा Tests Run हुन्छ।
4. **Hub र Node बीच Communication कसरी हुन्छ?**  
   JSON Wire Protocol मार्फत।
5. **Selenium Grid का Types के हुन्?**  
   Standalone, Hub-Node, Fully Distributed।
6. **Grid मा Cross-Browser Testing कसरी गर्ने?**  
   DesiredCapabilities मा Browser Name/Version Define गर्ने।
7. **Grid मा Parallel Execution कसरी Set गर्ने?**  
   TestNG Parallel Config वा Framework Code प्रयोग गरेर।
8. **Docker-Selenium के हो?**  
   Selenium Grid Docker Containers मा चलाउने Setup।
9. **Docker मा Chrome र Firefox Nodes कसरी Run गर्ने?**  
   docker-compose.yml मा Define गरेर।
10. **Selenium Grid को फाइदा?**  
    Time Save, Multi-Environment Testing, Scalability।
11. **Grid मा Version Mismatch समस्या के हो?**  
    Browser र Driver Version नमिल्दा Error आउनु।
12. **RemoteWebDriver के हो?**  
    Selenium Grid वा Cloud Execution मा Browser Control गर्ने Class।
13. **RemoteWebDriver को Constructor Parameters के हुन्?**  
    URL (Hub Address) र Capabilities।
14. **Capabilities र Options बीच फरक?**  
    Capabilities = Old Approach; Options = New Browser-Specific।
15. **BrowserStack/Sauce Labs के हुन्?**  
    Cloud Selenium Grid Providers।
16. **Cloud Grid को फाइदा?**  
    Real Device/Browser Access, No Local Setup।
17. **Grid मा Session Timeout के हो?**  
    Inactive Session Automatically Close हुने समय।
18. **Grid मा Node Timeout के हो?**  
    Node Idle हुँदा Disconnect हुने समय।
19. **Grid मा MaxSession के हो?**  
    Node मा एकपटक चल्ने Max Tests संख्या।
20. **Parallel Execution को Risk के हो?**  
    Shared Resources ले Data Conflict ल्याउन सक्छ।

**Section 9: Reporting & Logging (Q161–Q180)**

1. **Extent Reports के हो?**  
   Beautiful HTML Reports Generate गर्ने Reporting Library।
2. **Extent Reports Selenium मा कसरी Integrate गर्ने?**  
   ExtentReports, ExtentTest Objects Create गरेर।
3. **ExtentTest.log() के हो?**  
   Step-wise Logs Add गर्ने।
4. **ExtentReports.flush() के हो?**  
   Report File मा Data Write गर्ने।
5. **Screenshot Report मा कसरी Add गर्ने?**  
   getScreenshotAs() ले Capture गरेर Report मा Embed गर्ने।
6. **Allure Report के हो?**  
   Lightweight Reporting Framework।
7. **Allure Report Selenium मा कसरी Integrate गर्ने?**  
   TestNG Listeners वा Annotations प्रयोग गरेर।
8. **Log4j के हो?**  
   Logging Framework Java Projects मा प्रयोग हुने।
9. **Log4j Config File के हो?**  
   Logging Level, Appender, Pattern Define गर्ने File।
10. **Logging Level का Types के हुन्?**  
    TRACE, DEBUG, INFO, WARN, ERROR, FATAL।
11. **Why use logging in automation?**  
    Debugging र Execution Flow Tracking को लागि।
12. **Slf4j के हो?**  
    Logging Abstraction Layer।
13. **Reporting vs Logging फरक?**  
    Reporting – Execution Summary; Logging – Detailed Step Info।
14. **Custom HTML Report कसरी Generate गर्ने?**  
    Listener वा Reporter Interface Implement गरेर।
15. **Listener Interface के हो?**  
    Test Events Capture गर्ने TestNG Interface।
16. **ITestListener मा onTestSuccess() के हो?**  
    Test Pass हुँदा Trigger हुने Method।
17. **ITestListener मा onTestFailure() के हो?**  
    Test Fail हुँदा Trigger हुने Method।
18. **Reporter.log() के हो?**  
    TestNG Default Reporter मा Message Add गर्ने।
19. **Custom Screenshot Name कसरी राख्ने?**  
    Test Method Name + Timestamp प्रयोग गरेर।
20. **Reports CI/CD मा कसरी Integrate गर्ने?**  
    Build Pipeline मा Report Generate Path Add गर्ने।

**Section 10: Best Practices & Troubleshooting (Q181–Q200)**

1. **Selenium मा Test Stability कसरी बढाउने?**  
   Explicit Wait प्रयोग गर्ने, Hard Wait हटाउने।
2. **ElementNotVisibleException किन आउँछ?**  
   Element DOM मा भए पनि Visible नभए।
3. **NoSuchElementException किन आउँछ?**  
   Locator गलत वा Element Page मा छैन।
4. **StaleElementReferenceException किन आउँछ?**  
   DOM Refresh हुँदा पुरानो Reference Invalid हुनु।
5. **ElementClickInterceptedException किन आउँछ?**  
   Element माथि अर्को Element छ।
6. **TimeoutException किन आउँछ?**  
   Wait Condition समयमै पुरा नभए।
7. **WebDriverException किन आउँछ?**  
   Driver र Browser बीच Communication Error।
8. **Parallel Execution मा Data Conflict कसरी रोक्ने?**  
   ThreadLocal Driver Instance प्रयोग गर्ने।
9. **Page Load Timeout कसरी Set गर्ने?**  
   driver.manage().timeouts().pageLoadTimeout()।
10. **Script Timeout कसरी Set गर्ने?**  
    driver.manage().timeouts().setScriptTimeout()।
11. **Browser Cache कसरी Clear गर्ने Automation मा?**  
    Delete Cookies वा Incognito Mode प्रयोग गर्ने।
12. **Maximize Browser हर Test मा कसरी गर्ने?**  
    Base Class मा maximize code राख्ने।
13. **Headless Browser Mode के हो?**  
    GUI बिना Browser Run गर्ने Mode।
14. **Headless Mode किन प्रयोग गर्ने?**  
    Faster Execution, CI/CD मा Resource Save गर्न।
15. **ChromeOptions के हो?**  
    Chrome Browser Customization को लागि Class।
16. **FirefoxOptions के हो?**  
    Firefox Browser Customization को लागि Class।
17. **Browser Extensions Automation मा कसरी Load गर्ने?**  
    Options Class मा Extension Path Add गर्ने।
18. **Proxy Settings Selenium मा कसरी Set गर्ने?**  
    Proxy Object Create गरेर Capabilities मा Add गर्ने।
19. **Driver.quit() र driver.close() फरक?**  
    quit() – सबै Windows बन्द; close() – Current Window मात्र।
20. **Selenium Automation Best Practices के हुन्?**  
    POM प्रयोग गर्ने, Explicit Wait प्रयोग गर्ने, Reusable Methods बनाउने, Data Externalize गर्ने, Logs/Reports Maintain गर्ने।

**Section 11: Selenium Configuration & Environment (Q201–Q220)**

1. **Selenium Standalone JAR के हो?**  
   Selenium Server चलाउन प्रयोग हुने JAR, Grid वा Remote Execution का लागि।
2. **Selenium Language Bindings के हो?**  
   Selenium API को Programming Language Specific Library।
3. **Selenium मा Driver executable path कसरी Set गर्ने?**  
   System.setProperty("webdriver.chrome.driver", "path");
4. **WebDriver Manager किन प्रयोग गर्ने?**  
   Driver Version स्वतः Manage गर्न, Manual Download नहोस् भनेर।
5. **Selenium Java Project मा Maven किन प्रयोग गर्ने?**  
   Dependencies र Build Process Manage गर्न।
6. **pom.xml के हो?**  
   Maven Project Configuration File जसमा Dependencies, Plugins Define हुन्छन्।
7. **Selenium Maven Dependency कसरी Add गर्ने?**  
   pom.xml मा <dependency> tag भित्र Selenium Coordinates राखेर।
8. **TestNG Maven मा कसरी Add गर्ने?**  
   pom.xml मा TestNG Dependency राखेर।
9. **Selenium को Latest Version कसरी Check गर्ने?**  
   Maven Central Repository वा Official Website हेरेर।
10. **Driver Path Set नगरी Selenium कसरी Run गर्ने?**  
    WebDriver Manager वा PATH Environment Variable प्रयोग गरेर।
11. **PATH Variable मा Driver कसरी Add गर्ने?**  
    OS Environment Variables मा Driver को Folder Path राखेर।
12. **Maven Clean Command के हो?**  
    mvn clean – Target Folder Remove गर्छ।
13. **Maven Test Command के हो?**  
    mvn test – Tests Run गर्छ।
14. **Selenium Project मा src/test/java र src/main/java बीच फरक?**  
    src/main/java – Main Code, src/test/java – Test Code।
15. **Configurable Timeouts कसरी बनाउने?**  
    Properties File वा Constants Class बाट।
16. **Test Data Externalize किन गर्ने?**  
    Code बिना बदल्न, Maintainability बढाउन।
17. **Test Data Excel बाट कसरी पढ्ने?**  
    Apache POI वा JExcel API प्रयोग गरेर।
18. **JSON Test Data कसरी पढ्ने?**  
    JSON Simple वा Jackson Library प्रयोग गरेर।
19. **Properties File कसरी Read गर्ने?**  
    Java Properties Class प्रयोग गरेर।
20. **Configurable Browser Execution कसरी गर्ने?**  
    Browser Name Parameter Properties वा XML मा राखेर।

**Section 12: Common WebDriver Commands (Q221–Q240)**

1. **driver.get() र driver.navigate().to() फरक?**  
   get() Simple Navigation; navigate().to() Browser History Maintain गर्छ।
2. **driver.navigate().back() के हो?**  
   Previous Page मा फर्किने।
3. **driver.navigate().forward() के हो?**  
   Next Page मा जाने।
4. **driver.navigate().refresh() के हो?**  
   Current Page Reload गर्ने।
5. **driver.getTitle() के हो?**  
   Page Title फिर्ता गर्ने।
6. **driver.getCurrentUrl() के हो?**  
   Current URL फिर्ता गर्ने।
7. **driver.getPageSource() के हो?**  
   HTML Source Code String फिर्ता गर्ने।
8. **driver.manage().window().setSize() के हो?**  
   Browser Size Change गर्ने।
9. **driver.manage().window().setPosition() के हो?**  
   Browser Position Change गर्ने।
10. **driver.manage().deleteAllCookies() के हो?**  
    सबै Cookies हटाउने।
11. **driver.manage().addCookie() के हो?**  
    New Cookie Add गर्ने।
12. **driver.manage().getCookies() के हो?**  
    सबै Cookies List फिर्ता गर्ने।
13. **driver.manage().timeouts().implicitlyWait() के हो?**  
    Default Implicit Wait Set गर्ने।
14. **driver.manage().timeouts().pageLoadTimeout() के हो?**  
    Page Load हुने Maximum Time Set गर्ने।
15. **driver.manage().timeouts().setScriptTimeout() के हो?**  
    Async Script Execute हुने Maximum Time Set गर्ने।
16. **driver.close() के हो?**  
    Current Browser Window बन्द गर्ने।
17. **driver.quit() के हो?**  
    सबै Browser Windows बन्द गर्ने।
18. **driver.switchTo().window() के हो?**  
    अर्को Window मा Switch गर्ने।
19. **driver.switchTo().frame() के हो?**  
    iFrame मा Switch गर्ने।
20. **driver.switchTo().defaultContent() के हो?**  
    Main Document मा फर्किने।

**Section 13: Element Handling & Actions (Q241–Q260)**

1. **isDisplayed() कहिले False हुन्छ?**  
   Element Hidden वा CSS display:none भए।
2. **isEnabled() कहिले False हुन्छ?**  
   Element Disabled भए (disabled attribute)।
3. **isSelected() कहिले True हुन्छ?**  
   Checkbox/Radio Button Checked भए।
4. **Actions Class किन प्रयोग गर्ने?**  
   Advanced Mouse र Keyboard Actions Handle गर्न।
5. **moveToElement() के हो?**  
   Mouse Hover गर्ने।
6. **clickAndHold() के हो?**  
   Mouse Button Hold गर्ने।
7. **release() के हो?**  
   Mouse Button छोड्ने।
8. **contextClick() के हो?**  
   Right Click गर्ने।
9. **doubleClick() के हो?**  
   Double Click गर्ने।
10. **dragAndDrop() के हो?**  
    Element एक स्थानबाट अर्कोमा सार्ने।
11. **Keys Class के हो?**  
    Keyboard Keys Constants (ENTER, TAB आदि) को Collection।
12. **sendKeys(Keys.TAB) के गर्छ?**  
    Tab Key Press गर्ने।
13. **Robot Class कसरी Initialize गर्ने?**  
    Robot robot = new Robot();
14. **robot.keyPress() के हो?**  
    Keyboard Key Press गर्ने।
15. **robot.keyRelease() के हो?**  
    Keyboard Key Release गर्ने।
16. **Select Class को isMultiple() के हो?**  
    Dropdown Multiple Select Support गर्छ कि छैन Check गर्ने।
17. **Select मा deselectByValue() के हो?**  
    Specific Value Unselect गर्ने।
18. **Select मा deselectByIndex() के हो?**  
    Specific Index Unselect गर्ने।
19. **Select मा deselectByVisibleText() के हो?**  
    Specific Visible Text Unselect गर्ने।
20. **Select मा getAllSelectedOptions() के हो?**  
    सबै Selected Options फिर्ता गर्ने।

**Section 14: Practical Scenarios (Q261–Q280)**

1. **Dynamic Element कसरी Handle गर्ने?**  
   XPath contains() वा starts-with() प्रयोग गर्ने।
2. **Hidden Element मा Click कसरी गर्ने?**  
   JavaScriptExecutor प्रयोग गर्ने।
3. **Disabled Element मा Text कसरी Send गर्ने?**  
   JavaScriptExecutor प्रयोग गर्ने।
4. **Broken Links कसरी Find गर्ने?**  
   प्रत्येक <a> को href URL HTTP Status Check गर्ने।
5. **Image Load भएको छैन कसरी Check गर्ने?**  
   NaturalWidth Property Check गर्ने JavaScript मार्फत।
6. **Multiple Windows Handle गर्दा क्रम कसरी Find गर्ने?**  
   Window Handles List Iterate गर्ने।
7. **Table Data Read कसरी गर्ने?**  
   Table Rows र Columns Loop गर्ने।
8. **Pagination Table मा Data कसरी Search गर्ने?**  
   Loop + Next Button Click गरेर Search गर्ने।
9. **File Download कसरी Verify गर्ने?**  
   Download Folder मा File Exist Check गर्ने।
10. **Upload Confirmation कसरी Verify गर्ने?**  
    Upload Success Message वा Uploaded File Name Verify गर्ने।
11. **Auto-Suggestion Dropdown Handle कसरी गर्ने?**  
    Suggestions List Fetch गरेर Desired Text Select गर्ने।
12. **Calendar Date Picker Handle कसरी गर्ने?**  
    Month/Year Navigate गरेर Date Element Click गर्ने।
13. **Slider Handle कसरी गर्ने?**  
    dragAndDropBy() प्रयोग गर्ने।
14. **Tooltip Text कसरी Fetch गर्ने?**  
    getAttribute("title") वा Actions hover पछि getText()।
15. **Modal Dialog Handle कसरी गर्ने?**  
    Visible Text Verify गरेर Close Button Click गर्ने।
16. **Infinite Scroll Page मा Data कसरी Fetch गर्ने?**  
    Scroll Loop गरेर Load Check गर्ने।
17. **Session Cookie कसरी Add गर्ने?**  
    driver.manage().addCookie() प्रयोग गर्ने।
18. **Session Cookie कसरी Delete गर्ने?**  
    driver.manage().deleteCookieNamed() प्रयोग गर्ने।
19. **Mobile Emulation कसरी Enable गर्ने?**  
    ChromeOptions मा Device Metrics Set गर्ने।
20. **Network Conditions Automation मा कसरी Set गर्ने?**  
    Chrome DevTools Protocol प्रयोग गरेर।

**Section 15: Integration, CI/CD & Troubleshooting (Q281–Q300)**

1. **Selenium Jenkins मा कसरी Run गर्ने?**  
   Maven Project Build Step Add गरेर।
2. **Jenkins मा Report कसरी Show गर्ने?**  
   HTML Publisher Plugin प्रयोग गरेर।
3. **Git मा Selenium Project कसरी Manage गर्ने?**  
   Source Code Push गरेर Version Control Maintain गर्ने।
4. **Pull Request मा Selenium Tests कसरी Run गर्ने?**  
   CI Pipeline मा Test Stage Add गरेर।
5. **Docker मा Selenium कसरी चलाउने?**  
   docker-compose वा Selenium Hub-Node Images प्रयोग गरेर।
6. **Cloud Selenium Providers कुन-कुन?**  
   BrowserStack, Sauce Labs, LambdaTest।
7. **Cloud Testing को फाइदा?**  
   Real Browser/Device Testing, Infrastructure Setup नगरी।
8. **Parallel Execution CI मा किन प्रयोग गर्ने?**  
   Execution Time कम गर्न।
9. **Cross-Browser Testing CI मा कसरी गर्ने?**  
   Different Browser Configurations Pipeline मा Add गर्ने।
10. **Common Selenium Failures CI मा कुन-कुन?**  
    ElementNotFound, Timeout, Version Mismatch।
11. **Driver Version Mismatch कसरी Fix गर्ने?**  
    Browser Driver Update गर्ने वा Browser Downgrade गर्ने।
12. **Headless Mode CI मा किन प्रयोग गर्ने?**  
    GUI बिना Faster Execution गर्न।
13. **Element Overlapping किन हुन्छ?**  
    CSS Layout Issues वा Dynamic Popups।
14. **Implicit र Explicit Wait Mixed गर्दा Risk के हो?**  
    Unexpected Delays, Timeout Issues।
15. **Flaky Test के हो?**  
    कहिले Pass, कहिले Fail हुने Test।
16. **Flaky Test कसरी Fix गर्ने?**  
    Stable Wait, Proper Synchronization, Data Isolation।
17. **Test Retry CI मा किन प्रयोग गर्ने?**  
    Temporary Issues को कारण Fail हुने Tests पुन: Run गर्न।
18. **Video Recording CI मा कसरी Enable गर्ने?**  
    Cloud Provider वा Custom Recorder Library प्रयोग गरेर।
19. **Failure Screenshot CI मा कसरी Save गर्ने?**  
    onTestFailure() मा Screenshot Code Add गर्ने।
20. **Selenium Test Best CI/CD Practice के हो?**  
    Parallel Execution, Proper Waits, Version Control, Reporting, Retry Logic।