**## DEBUGGING ##**

* Step over – F8
* Step into – F7
* Resume – F9

**## GIT ##**

* Git is an open-source distributed version control system.
* It is designed to handle minor to major projects with high speed and efficiency.
* It is developed to co-ordinate the work among the developers
* The version control allows us to track and work together with our team members at the same workspace.

**## GIT HUB ##**

* GitHub is a Git repository hosting service
* GitHub also facilitates with many of its features such as access control and collaboration.
* It provides a web-based graphical interface
* GitHub is an American company. It hosts source code of your project in the form of different programming language and keeps track of the various changes made by programmers.

**##INTELLIJ##**

* IntelliJ is an Integrated Development Environment written in Java for developing computer software.
* User friendly.
* Open browser>>Search for IntelliJ download>>Tap on first link>>Download Community Edition>>Open it>>Next>>Install and Finish.
* Open IntelliJ>>New project>>Advance settings>>Module name>>Untick checkbox>>Create.
* **Package name format**, for example, com.obs.sample (in small letters).

**## GIT USING COMMANDS (pushing project to Git) ##**

* Create a new repository in GitHub.
* Open command prompt >> **git - -version** (to know the version of the git).
* Open IntelliJ >> Open terminals>> Inside terminal, we can see PS (Power shell).
* To delete PS, Open File menu>>Settings>>Search for Terminals>>Configure Terminal>>Terminal>>Choose second option in Shed Path>>Apply>>OK.
* Type **git - -version** on Terminal.
* **To see folders**, right click on folder>>Open in>>Explorer.
* **To know the value of an equation,** right click on that line>>Evaluate expression.
* **To find changes we did**, File>>settings>>commit>>Untick first option>>Apply>>OK.
* **To we change,** Right click>>git>>show diff.

**Step 1: Git Initialization**

* Open Terminal>>**git init** (git initialization command).
* Goto project folder>>View>Show>>Hidden items (. git folder will be created).
* **To see folders**, right click on folder>>Open in>>Explorer.

**Step 2: Linking folder and repository**

* Open Repository>>Scroll down and copy second commands>>Paste it in terminal>>Refresh repository and terminal.

**Step 3: Status check**

* Open Terminal>>**git status**
* If it’s in red colour means not pushed.

**Step 4: Adding folders**

* To select all folders, Open Terminal>>**git add .**
* To add particular folder, **git add path**

**Step 5: To Commit**

* Open Terminal>>**git commit -m”initial commit”**

**Step 6: To Push**

* Open Terminal>>**git push**
* To which branch, **git push - -set-upstream origin main**

**## AUTOMATION TESTING ##**

* Testing an application using automation tool, such as programming languages without human intervention.
* In automation testing, automation tools are used to execute the test cases.

**## BENEFITS OF AUTOMATION TESTING ##**

* Cost saving
* Productivity increases
* Accuracy
* Saves time
* Test suit reusability
* Ability to test on various platforms (cross browser testing)
* Less human resources
* Early bug detection

**## IDEAL CANDIDATE FOR AUTOMATION ##**

* Depends upon the budget of the project
* Check repeated tc’s or not
* Suits
* Feasibility
* Tests that tend to cause human error
* Tests that are impossible to perform manually.
* Tests that run on several different hardware or software platforms and configurations.
* Depends upon no. of TC

**## WHY SELENIUM ##**

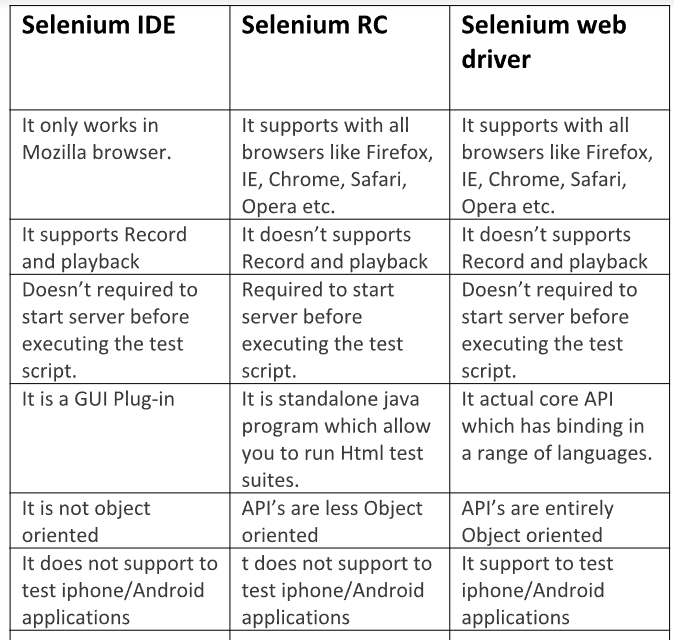
* Cross-browser testing possible
* Open-source tool
* Selenium is communicated with web html elements.
* Language preference
* Most of the web applications are in java

**## HISTORY OF SELENIUM ##**

* Year – 2004
* Founder – Jason Huggins
* Company – ThoughtWorks in Chicago
* Latest Version – 4.10.0
* Official website – [www.selenium.dev](http://www.selenium.dev)

**## DIFFERENT FLAVOURS/COMPONENTS OF SELENIUM ##**

* + - 1. Selenium IDE:
* Integrated Development Environment
* Record and playback only
* Username for Demo web shop – [aswathyarabind@gmail.com](mailto:aswathyarabind@gmail.com)
* Open <https://selenium.obsqurazone.com/index.php> >> Input form>>right click anywhere >>Inspect >>Inside that we can see attributes and its types.
  + - 1. Selenium RC
* Remote control
  + - 1. Selenium Web driver
      2. Selenium Grid
* Hub-Node concept



**## HOW TO SETUP SELENIUM ##**

* Selenium is an external library.
* Create a maven project, File>>New>>Project>> Choose Build system as maven>> Enter Groupid (Format as org.obs.SeleniumCommands)>>Create.
* To add jar files, Open chrome>>Maven repository >>Search for selenium>> Choose first one>>Copy dependency >> Paste it inside pom.xml>>Refresh.

**## SELENIUM WEBDRIVER ##**

The features of Selenium web driver are:

* **Multiple browser support** - Selenium WebDriver supports a diverse range of web browsers such as Firefox, Chrome, Internet Explorer, Opera and many more.
* **Multiple Language support** - WebDriver also supports most of the commonly used programming languages like Java, C#, JavaScript, PHP, Ruby, Pearl and Python.
* **Speed** - WebDriver performs faster as compared to other tools of Selenium Suite.
* **Simple commands** - Most of the commands used in Selenium WebDriver are easy to implement.

**## BROWSER COMMANDS ##**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **SYNTAX** | **USAGE** | **EXAMPLE** |
| 1 | driver=new ChromeDriver (); | To open Chrome browser | driver=new ChromeDriver (); |
| 2 | driver=new EdgeDriver (); | To open Edge browser | driver=new EdgeDriver (); |
| 3 | driver=new FirefoxDriver (); | To open Firefox browser | driver=new FirefoxDriver (); |

**## WEBDRIVER COMMANDS ##**

* If web driver is an interface (or parent), then, so many classes are inside selenium (like chrome driver, fire fox driver, Edge driver, etc).
* These classes implements interface.
* To set alignement of code, ctrl+A then ctrl+Alt+l.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **COMMANDS** | **SYNTAX** | **USAGE** | **EXAMPLE** |
| 1 | get () | driver.get (URL); | To open URL. | driver.get(“https://selenium.obsqurazone.com/index.php”); |
| 2 | getTitle () | driver.getTitle (); | To get title of the website. | String actualTitle = driver.getTitle(); |
| 3 | getCurrentUrl () | driver.getCurrentUrl (); | To get the URL of the current web page being displayed in the browser. | String actualCurrenturl=driver.getCurrentUrl(); |
| 4 | getPageSourcecode () | driver. getPageSource(); | To return the source code of the page. | String actualSourcecode = driver.getPageSource(); |
| 5 | Close () | driver.close (); | To close the corresponding browser window. | driver.close (); |
| 6 | Maximize () | driver.manage().window().maximize(); | To maximize the browser window. | driver.manage().window().maximize(); |
| 7 | deleteAllCookies () | driver.manage().deleteAllCookies(); | To delete all the cookies. | driver.manage().deleteAllCookies(); |

**## WEBELEMENT COMMANDS ##**

* Web element is an interface.
* Each element in UI is called as web elements.
* Web elements represents HTML elements.
* For instance, a HTML element is written as: "<tagname> content </tagname>"
* To find out web elements, we have to locate particular elements.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **COMMANDS** | **SYNTAX** | **USAGE** | **EXAMPLE** |
| 1 | sendKeys() | element.sendKeys("text"); | Used to enter editable content in the text and password fields during test execution. | emailfield.sendKeys(“a@gmail.com”); |
| 2 | click() | element.click(); | Used to click Login button. | loginbutton.click(); |

**## LOCATORS ##**

* A locator is a way to identify elements on a page.
* 8 types: id, name, classname, xpath, linktext, partiallinktext, cssselect, tagname.
* Web element should be unique.
* classname can be repeat.
* id is unique.
* To get xpath, right click on page>>inspect>>right click on highlighted tag>>copy>>copyxpath.

| **NO** | **LOCATORS** | **DESCRIPTION** |
| --- | --- | --- |
| 1 | class name | Locates elements whose class name contains the search value (compound class names are not permitted) |
| 2 | cssselector | Locates elements matching a CSS selector |
| 3 | Id | Locates elements whose ID attribute matches the search value |
| 4 | name | Locates elements whose NAME attribute matches the search value |
| 5 | link text | Locates anchor elements whose visible text matches the search value |
| 6 | partial link text | Locates anchor elements whose visible text contains the search value. If multiple elements are matching, only the first one will be selected. |
| 7 | tag name | Locates elements whose tag name matches the search value |
| 8 | xpath | Locates elements matching an XPath expression |