**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

**GitHub**

* GitHub is a Git repository hosting service
* GitHub also facilitates with many of it’s 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.

**History**

* Starts in 2004 at ThoughtWorks in Chicago
* Father - Jason Huggins
* Older name "JavaScriptTestRunner"
* Shinya Kasatani in Japan could wrap the core code into an IDE module into the Firefox browser
* Selenium latest version – 4.10.0
* Selenium official website – Selenium.dev

**Automation**

Automation testing refers to the automatic testing of the software in which developer or tester write the test script once with the help of testing tools and framework and run it on the software. The test script automatically test the software without human intervention and shows the result (either error, bugs are present or software is free from them).

**Advantages of automation:**

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

**Ideal candidates for automation**

* Repeated TC
* 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 on the budget of project
* Depends on no.of TC

**Why JAVA(selenium)**

* Most of the web applications are in java
* Open source
* Cross browser testing

**Different flavours (component)of selenium**

* selenium IDE
* selenium RC
* selenium web driver
* selenium grid

|  |  |  |
| --- | --- | --- |
| **Selenium IDE** | **Selenium RC** | **Selenium web driver** |
| It only works in Mozilla browser. | It supports with all browsers like Firefox, IE, Chrome, Safari, Opera etc. | It supports with all browsers like Firefox, IE, Chrome, Safari, Opera etc. |
| It supports Record and playback | It doesn’t supports Record and playback | It doesn’t supports Record and playback |
| Doesn’t required to start server before executing the test script. | Required to start server before executing the test script. | Doesn’t required to start server before executing the test script. |
| It is a GUI Plug-in | It is standalone java program which allow you to run Html test suites. | It actual core API which has binding in a range of languages. |
| It is not object oriented | API’s are less Object oriented | API’s are entirely Object oriented |
| It does not support to test iphone/Android applications | It does not support to test iphone/Android applications | It support to test iphone/Android applications |
|  |  |  |

**Selenium WebDriver**

* Selenium WebDriver was first introduced as a part of Selenium v2.0
* In WebDriver, test scripts can be developed using any of the supported programming languages and can be run directly in most modern web browsers. Languages supported by WebDriver include C#, Java, Perl, PHP, Python and Ruby.
* Selenium WebDriver performs much faster as compared to Selenium RC because it makes direct calls to the web browsers. RC on the other hand needs an RC server to interact with the browser.
* WebDriver has a built-in implementation of Firefox driver (Gecko Driver). For other browsers, you need to plug-in their browser specific drivers to communicate and run the test. Most commonly used WebDriver's include:
* Google Chrome Driver
* Internet Explorer Driver
* Opera Driver
* Safari Driver
* HTML Unit Driver (a special headless driver)

**Commands for launching browsers**

1. ChromeDriver

**Syntax:** WebDriver driver=new ChromeDriver

1. EdgeDriver

**Syntax:** WebDriver driver=new EdgeDriver

1. FirefoxDriver

**Syntax:** WebDriver driver=new FirefoxDriver

**Selenium commands**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Command** | **Syntax** | **Usage** | **Example** |
| 1 | maximize() | driver.manage().window().maximize() | To maximise the browser | driver.manage().window().maximize() |
| 2 | deleteAllCookies() | driver.manage().deleteAllCookies() | To delete cookies | driver.manage().deleteAllCookies() |
| 3 | get() | driver.get(url) | To load URL | Driver.get(<https://selenium.obsqurazone.com/index.php>) |
| 4 | getTitle() | driver.getTitle() | To Get the title of the website | String actualTitle=driver.getTitle() |
| 5 | getCurrentUrl() | driver.getCurrentUrl() | To get the URL of the website | String actualCurrentUrl=driver.getCurrentUrl() |
| 6 | getPageSource() | driver.getPageSource() | To get the source code of the current page | String actualPageSource=driver.getPageSource() |
| 7 | Driver.close() | Driver.close() | To close the browser | Driver.close() |
| 8 | findElement() | driver.findElement(By.partialLinkText("Text”) | uniquely identify a web element within the web page. | WebElement yourDestination=driver.findElement(By.partialLinkText("Register")); |
| 9 | findElements() | driver.findElements(By.tagName("a")); | uniquely identify the list of web elements within the web page. | List<WebElement> hyperLinks=driver.findElements(By.tagName("a")); |

**WebElements**

* WebElements are HTML element on a website
* To findout WebElements we have to locate particular elements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Commands** | **Syntax** | **Usage** | **Example** |
| 1 | sendKeys() | Elements.sendKeys(“text”) | allows the user to type content automatically into an editable field while executing tests | emailField.sendKeys("geethuaravind1994@gmail.com"); |
| 2 | Click() | Element.click() | Used to click action on a button, link, radio button or checkbox. | loginButton.click(); |

**Locators**

* A locator is a way to identify elements on a page. It is the argument passed to the [Finding element](https://www.selenium.dev/documentation/webdriver/elements/finders/) methods.
* 8 types of locators available

id

name

classname

xpath

linktext

partiallinktext

cssselector

tagname

| **Locator** | **Description** |
| --- | --- |
| class name | Locates elements whose class name contains the search value (compound class names are not permitted) |
| css selector | Locates elements matching a CSS selector |
| id | Locates elements whose ID attribute matches the search value |
| name | Locates elements whose NAME attribute matches the search value |
| link text | Locates anchor elements whose visible text matches the search value |
| 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. |
| tag name | Locates elements whose tag name matches the search value |
| xpath | Locates elements matching an XPath expression |

|  |  |  |
| --- | --- | --- |
| No | Locators | Example |
| 1 | id | WebDriver driver = new ChromeDriver();  driver.findElement(By.id("lname")); |
| 2 | name | WebDriver driver = new ChromeDriver();  driver.findElement(By.name("newsletter")); |
| 3 | classname | WebDriver driver = new ChromeDriver();  driver.findElement(By.className("information")); |
| 4 | xpath | WebDriver driver = new ChromeDriver();  driver.findElement(By.xpath("//input[@value='f']")); |
| 5 | linktext | WebDriver driver = new ChromeDriver();  driver.findElement(By.linkText("Selenium Official Page")); |
| 6 | partiallinktext | WebDriver driver = new ChromeDriver();  driver.findElement(By.partialLinkText("Official Page")); |
| 7 | cssselector | WebDriver driver = new ChromeDriver();  driver.findElement(By.cssSelector("#fname")); |
| 8 | tagname | WebDriver driver = new ChromeDriver();  driver.findElement(By.tagName("a")); |

**Selenium exceptions**

|  |  |  |
| --- | --- | --- |
| **No** | **Exception** | **Occurance** |
| 1 | NoSuchElementException | Occurs when the locator is unable to to find the web element on the web page |
|  |  |  |

**Differene between findElement and findElements methods**

 findElement: This command is used to uniquely identify a web element within the web page.

 findElements: This command is used to uniquely identify the list of web elements within the web page.

XPath

* Xpath is one of the most important locators used in Selenium for identifying web elements.
* XPath can be used as a substitute when you don't have a suitable id or name attribute for the element you want to locate.
* XPath allows you to select individual elements, attributes, and some other part of XML documents for specifying location of a particular web element.

**Syntax:** findElement(By.xpath("XPath"));

* Two types

1. Absolute XPath - Here xpath traverses direct from parent to child in DOM. Thus in absolute xpath we have to travel from the root node to the target.

**Syntax −** driver.findElement(By.xpath("/html/body/div/input")).

1. Relative XPath - Here xpath finds the matching element in every corner of DOM. It doesn't have a particular beginning point.

**Syntax −** driver.findElement(By.xpath("//input[@name=’Tutorial’]")).

* It is always recommended to use relative xpath rather than absolute xpath. In absolute xpath, we need to specify from the root to the desired element so if any of the attributes and its value get changed in between, then our xpath will no longer be correct.
* Syntax of relative xpath

1. Using tag name,attribute and value

Syntax : //tagName[@attribute='value']

eg : //input[@id="single-input-field"]

//button[@class='navbar-toggler']

1. Using contains

Syntax: //tagName[conatins(@attribute,'value')]

eg : //input[contains(@id,'gender')]

//input[contains (@id,'register')]

1. Using 'and'

Syntax: //tagName[@attribute1='value1' and @attribute2='value2']

eg : //input[@id='gender-male' and @name="Gender"]

1. Using 'or'

Syntax://tagName[@attribute1='value1' or @attribute2='value2']

eg : //input[@id='gender-male' or @name="Gender"]