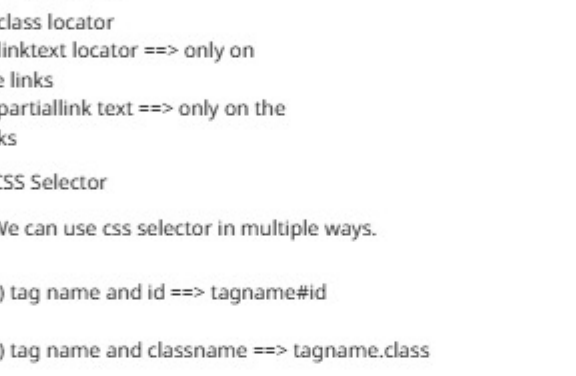
Selenium
1) It is a automation tool use to test
web applications(browser)
2) In selenium we have 4 types
=> Selenium RC(Deprecated) OLDEST
=> Selenium WebDriver(LATEST ==> THIS)
=> Selenium GRID(execute the test cases in mutiple
device)
=> Selenium IDE(Record and Playback Tool)

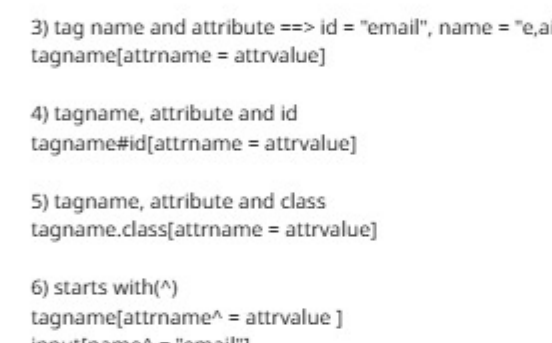
Open the browser
Naviagte to the website
Enter the valid username
Enter the valid password
Click on Login

Inbox page should
get displayed

Dont require coding skills,
and want to reduce
manual work

Selenium WebDriver

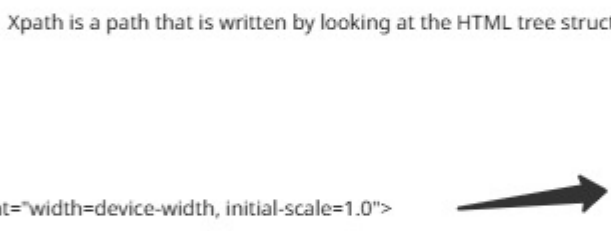




1) Open the Browser
2) Navigate to Facebook website
3) Enter a invalid email
4) Enter a invalid password
5) Click on Login

Error message should
be displayed

In selenium WebDriver, there is a built in class ==> ChromeDriver ==>
Launch the chrome browse / open



Selenium WebDriver is available in the form of JAR file(contains
so many built classes, interfaces, methods, ...).

Anything that we see on the website, in selenium we call it as WebElement
apart from text.

Locators

=> Helps to identify various webelements present in a website.

1) id locator
2) name locator
3) class locator
4) linktext locator ==> only on
the links
5) partiallink text ==> only on the
links
6) CSS Selector

We can use css selector in multiple ways.

1) tag name and id ==> tagname#id

2) tag name and classname ==> tagname.class

3) tag name and attribute ==> id = "email", name = "e,ail"
tagname[attrname = attrvalue]

4) tagname, attribute and id
tagname#id[attrname = attrvalue]

5) tagname, attribute and class
tagname.class[attrname = attrvalue]

6) starts with(^)
tagname[attrname^ = attrvalue]
input[name^ = "email"]

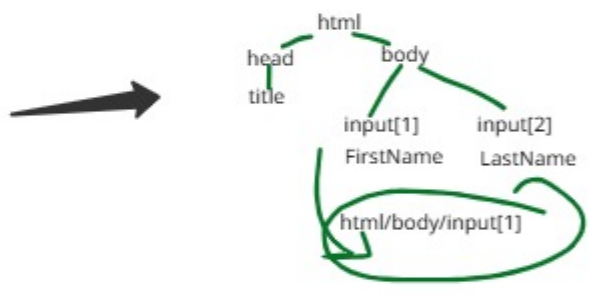
7) starts with($)
tagname[attrname$ = attrvalue]
input[name$ = "email"]

Xpath

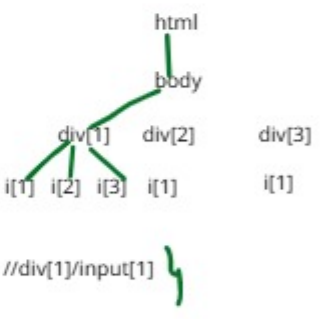
=> If we are unable to identify a webelement using any of
the aboe lcoators, then we can identify using XPATH.

Xpath is a path that is written by looking at the HTML tree structure

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
   Enter FirstName:<input type="text">
   Enter LastName:<input type="text">
</body>
</html>
```



/html/body/input[1] ==> Absolute Xpath(from root element(html))
//input[1] ==> Relative Xpath(from the HTML tag closer to the
WebElement)



1) Xpath with Absolute
2) Xpath with Relative
3) Xpath with text() method
=> If we have a text on a webpage we can identify
using text().

Thread.sleep() makes the selenium to wait, but internally
some time would get wasted if elements is found early.
to overcome this problem we can use implicit wait,
Implicit Wait is the wait given to all the web elements
that selenium would wait until element is found.

input[role='searchbutton']

1) Xpath with Absolute
2) Xpath with Relative
3) Xpath with text() method
=> If we have a text on a webpage we can identify
using text().
//tagname[text() = '----']
4) Xpath with tagname and attirbute
//tagname[@attrname=attrvalue]

(//input[@role='searchbox'])[1]

text() will identify the text but if faily to identify if the text has
any space in starting or ending.

contains()
this will identify text regardless tghere is space before or
after the string
//tagname[contains(text(), "-----")]

Explicit wait => It is a wait that is applied to a aprticular web element.
We can implement explicit wait using a class WebDriverWait.

Fluent Wait ==> Old one before Expkcit Wait came
99% same as Explciit Wait
In FluentWait we having polling duration

50 seconds of explicit wait
50 seconds of fluent wait, polling => 2sec

How to handle dropdown's in selenium:-

1) Drop down with options from the beginning
2) **Autosuggestive Dropdown**
text(), inplicitWait()

NOTE: If we want to handle non suggestive dropdowbns,
then we need to use a class in selenium SELECT class.

close() will close all the parent tabs,
quit() will close all the tabs i.e entire browser

Go to amazon website count how many links are available?

findElement()	findElements()
1) targets only oen element	1) targets multiple elemenys
2) return type is WebElement	2) return type is List<WebElement>
3) This gives NoSuchElement exception if element is not found3	3) If element is not found, it doesnt give any exception