Website for study:

https://www.guru99.com/selenium-tutorial.html

Selenium webdriver:

it is not a tool

it is web driver API

it is collection of Jars file. this files contain packages , classes, functions, interfaces.

Setup selenium:

https://www.youtube.com/watch?v=IkK7bS6hLms

setup chomedriver:

https://www.youtube.com/watch?v=FSetmugrnmY

chrome driver:

https://chromedriver.storage.googleapis.com/index.html?path=114.0.5735.90/

firefox driver:

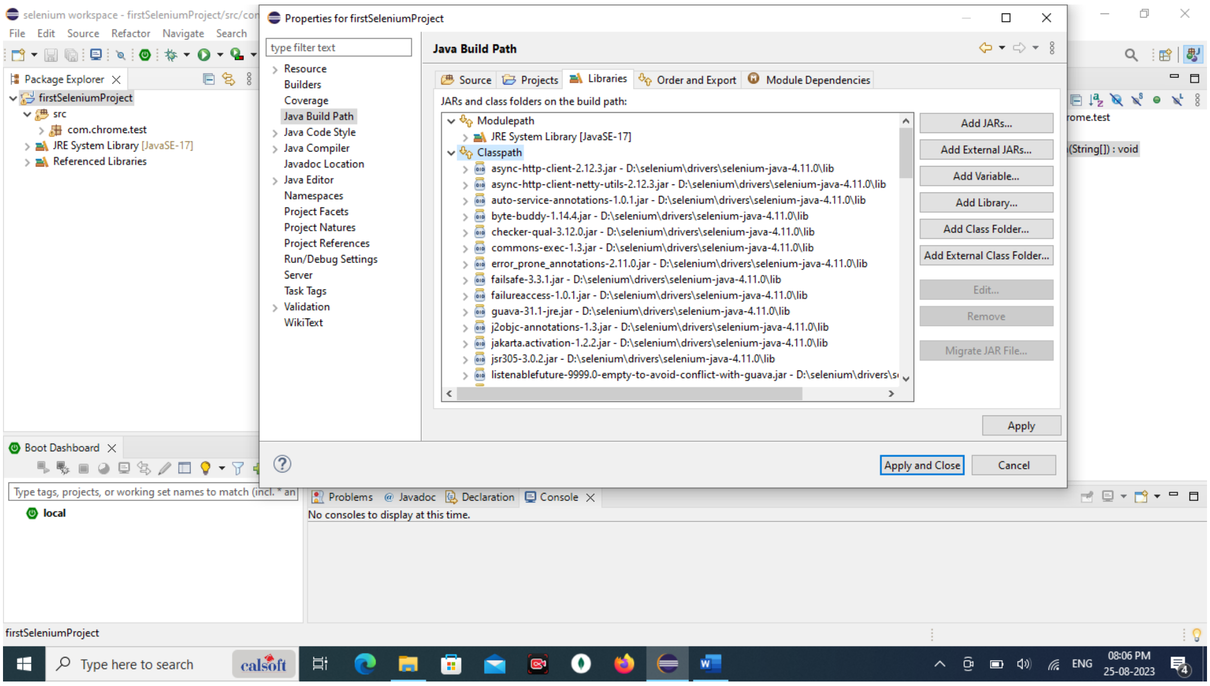
https://www.youtube.com/watch?v=avr68JmDLJQ

Demo webSite for test:

https://demo.guru99.com/test/newtours/

Selenium library (Jars) setup in eclipse project:

1) Create simple java project > Right click on java project > properties > select java build path > click on libraries > select class path(don’t select modulepath) > add external jars > select from selenium folder.



JARs and class folders on the build path:

```
Modulepath
  JRE System Library [JavaSE-17]
Classpath
  async-http-client-2.12.3.jar - D:\selenium\drivers\selenium-java-4.11.0\lib
  async-http-client-netty-utils-2.12.3.jar - D:\selenium\drivers\selenium-java-4.11.0\lib
  auto-service-annotations-1.0.1.jar - D:\selenium\drivers\selenium-java-4.11.0\lib
  byte-buddy-1.14.4.jar - D:\selenium\drivers\selenium-java-4.11.0\lib
  checker-qual-3.12.0.jar - D:\selenium\drivers\selenium-java-4.11.0\lib
  commons-exec-1.3.jar - D:\selenium\drivers\selenium-java-4.11.0\lib
  error_prone_annotations-2.11.0.jar - D:\selenium\drivers\selenium-java-4.11.0\lib
  failsafe-3.3.1.jar - D:\selenium\drivers\selenium-java-4.11.0\lib
  failureaccess-1.0.1.jar - D:\selenium\drivers\selenium-java-4.11.0\lib
  guava-31.1-jre.jar - D:\selenium\drivers\selenium-java-4.11.0\lib
  j2objc-annotations-1.3.jar - D:\selenium\drivers\selenium-java-4.11.0\lib
  jakarta.activation-1.2.2.jar - D:\selenium\drivers\selenium-java-4.11.0\lib
  jsr305-3.0.2.jar - D:\selenium\drivers\selenium-java-4.11.0\lib
  listenablefuture-9999.0-empty-to-avoid-conflict-with-guava.jar - D:\selenium\drivers\s
```

No consoles to display at this time.

firstSeleniumProject

08:06 PM
25-08-2023

List of websites to practice selenium:

- **Cleartrip: https://www.cleartrip.com/**
- **http://the-internet.herokuapp.com/** - This is not a full-fledged website, but it offers components of the website that you can test. If you are working on practicing a complex component, I would recommend this site.
- **https://www.saucedemo.com/** - This is a demo website from SauceLabs. It's a custom eCommerce site by SauceLabs. But, not with much functionality.
- **https://ultimateqa.com/automation** - This is also a good site. But, it does not have a complete user journey. If you want to practice just a part of a user journey, this is a good site.
- **http://opensource-demo.orangehrmlive.com/** - This website is not meant to practice test automation. But, you can write a test around this.
- **https://phptravels.org/** - Similarly, this website is not meant to practice. But, for a short time, you can write tests and execute them. Can't be used for demonstration as it may get changed anytime.
- **https://demo.openmrs.org/openmrs/** - This is another open-source site. But, can't expect to have it unchanged.

Web Elements: button, option, input box, submit, login ,checkbox, drop down box, images, list box, web table, drag and drop, calendar control, sate pickers, links, radio button, elements on website are considered as web elements.

(Everything on page considered as a web elements)

Check box : in check box we can select multiple option.

Radio button: only one we can select.

Note: selenium can not support window. Means while upload photo window is open to pick image. That window feature is not supported.

Overcome: to overcome above issue. We can use third party tool. i.e. Auto-IT

Locators: (we can say properties of web elements)

Every web elements has different properties like: name, id, cssSelector, xpath these are mainly known as locators.

(to identify the web elements available on web page. For that we use locators.)

1. **Name**
2. **Id**
3. **cssSelectors**
4. **xpath**
5. **class name**

Test case:

1. open browser
2. open website
3. enter username
4. enter password
5. submit
6. close

Login details properties:

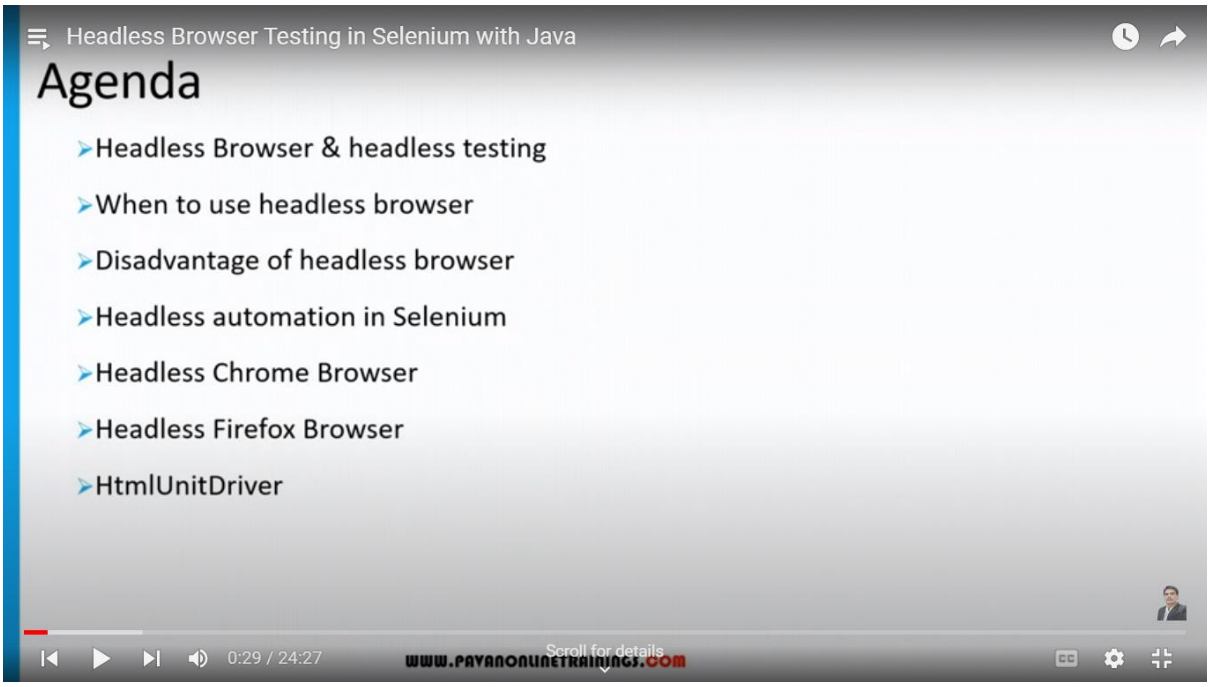
1. **Username:**
   copy element : <input type="text" name="userName" size="10">
   copy xpath:
   /html/body/div[2]/table/tbody/tr/td[2]/table/tbody/tr[4]/td/table/tbody/tr/td[2]/table/tbody/tr[2]/td[3]/form/table/tbody/tr[4]/td/table/tbody/tr[2]/td[2]/input
2. **Password:**
   copy element: <input type="password" name="password" size="10">
   copy xpath:
   /html/body/div[2]/table/tbody/tr/td[2]/table/tbody/tr[4]/td/table/tbody/tr/td[2]/table/tbody/tr[2]/td[3]/form/table/tbody/tr[4]/td/table/tbody/tr[3]/td[2]
3. **Submit:**
   Copy elemts: <input type="submit" name="submit" value="Submit" style="background-color: #FACB00;border: 1px solid black;font-family:MS Sans Serif;font-size:15px;">
   Copy xpath:
   /html/body/div[2]/table/tbody/tr/td[2]/table/tbody/tr[4]/td/table/tbody/tr/td[2]/table/tbody/tr[2]/td[3]/form/table/tbody/tr[4]/td/table/tbody/tr[4]/td[2]/div/input

Driver methods:

1. **driver.findElement(By.name()).sendKeys();**
   this methods for : text box / input box
2. **driver.findElement(By.name()).click();**
   this methods for : radio button, check box, link.
3. **Driver.Close**
   Close browser
4. `driver.manage().window().maximize();`
   `for maximize browser window`

HeadLess Browser:

Agenda: Topics

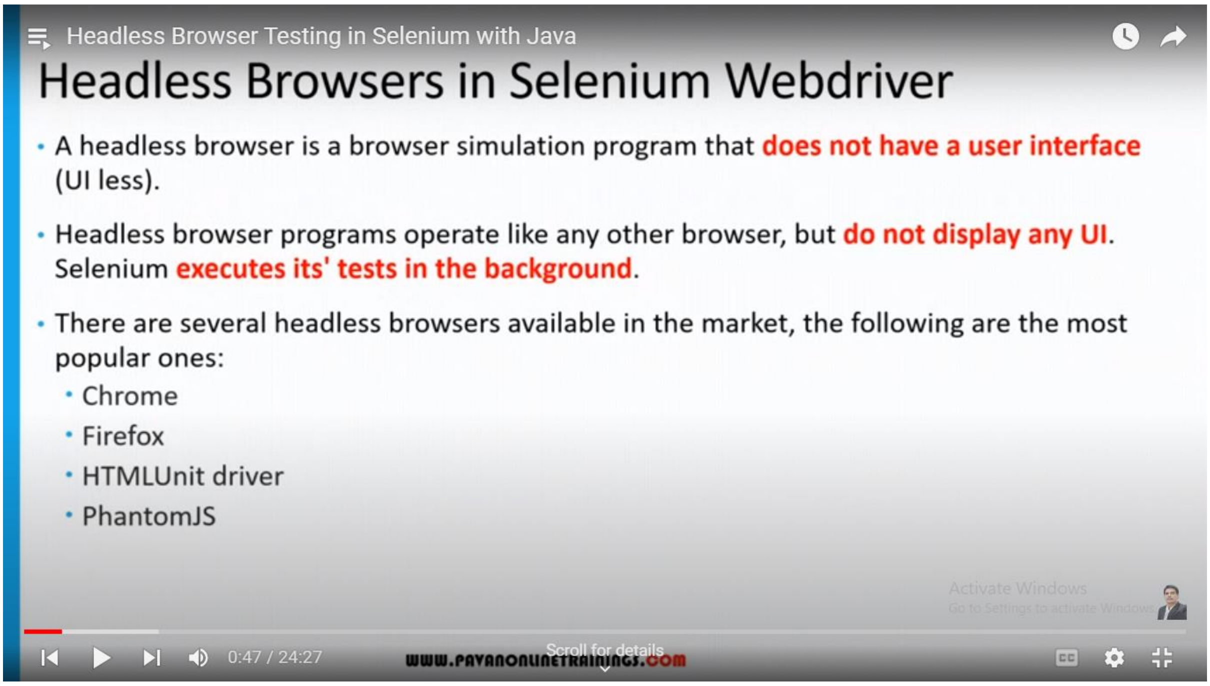


Headless Browser Testing in Selenium with Java

Agenda

- Headless Browser & headless testing
- When to use headless browser
- Disadvantage of headless browser
- Headless automation in Selenium
- Headless Chrome Browser
- Headless Firefox Browser
- HtmlUnitDriver

What is headless browser?



Headless Browser Testing in Selenium with Java

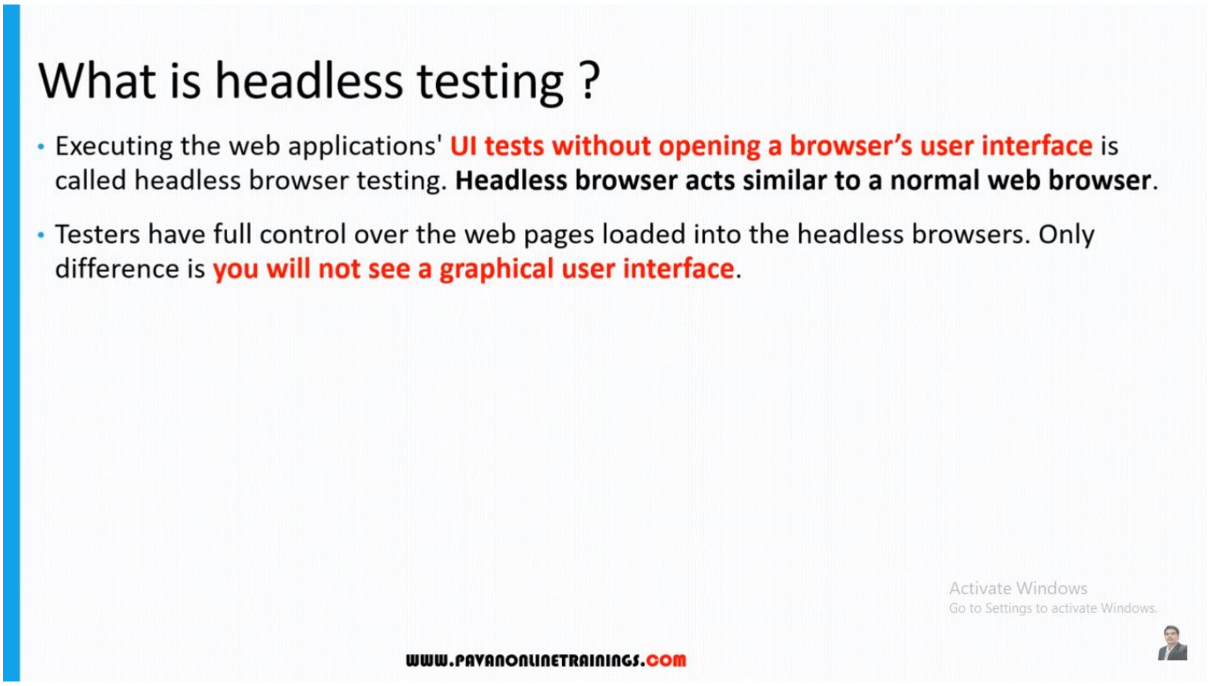
Headless Browsers in Selenium Webdriver

- A headless browser is a browser simulation program that **does not have a user interface** (UI less).
- Headless browser programs operate like any other browser, but **do not display any UI**. Selenium **executes its' tests in the background**.
- There are several headless browsers available in the market, the following are the most popular ones:
  - Chrome
  - Firefox
  - HTMLUnit driver
  - PhantomJS

0:47 / 24:27
Scroll for details
www.pavanonlinetrainings.com
Activate Windows
Go to Settings to activate Windows.

Browser does not have a user interface. Means it does not display any UI. Executes its tests in the background.

What is headless browser testing?

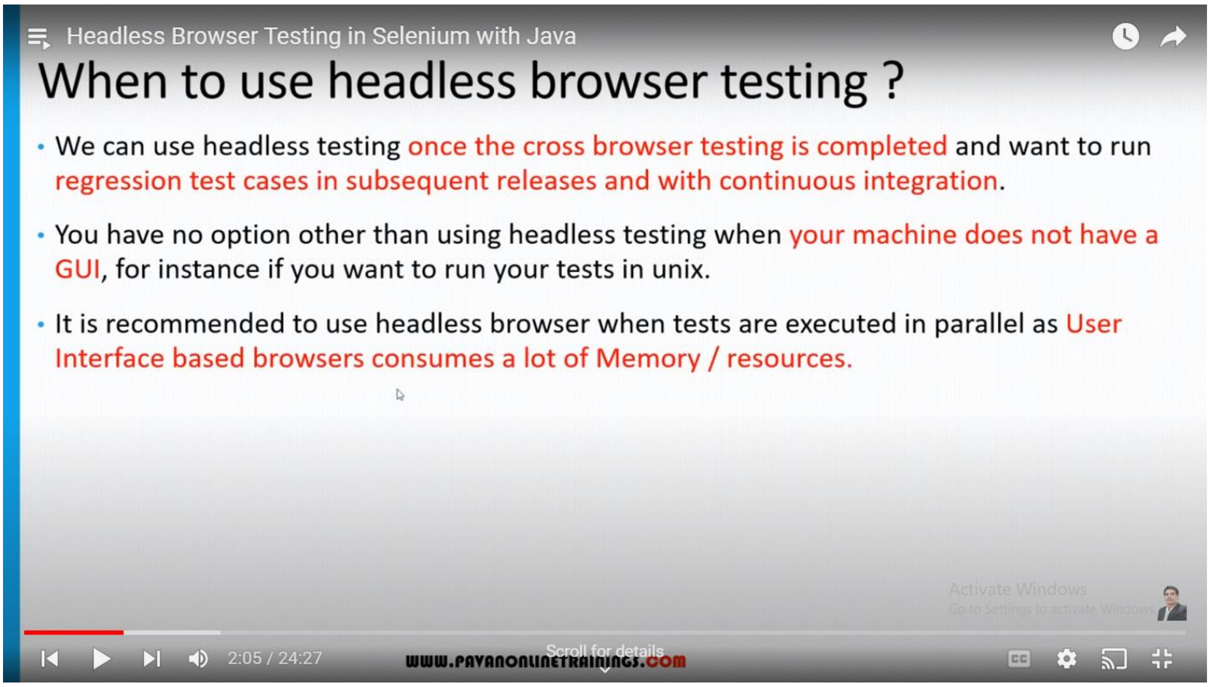


What is headless testing ?

- Executing the web applications' **UI tests without opening a browser's user interface** is called headless browser testing. **Headless browser acts similar to a normal web browser.**
- Testers have full control over the web pages loaded into the headless browsers. Only difference is **you will not see a graphical user interface**.

Activate Windows
Go to Settings to activate Windows.
www.pavanonlinetrainings.com

When to use headless browser testing?



Headless Browser Testing in Selenium with Java
2:05 / 24:27
Scroll for details
WWW.PAVANONLINETRAININGS.COM
Activate Windows
Go to Settings to activate Windows.

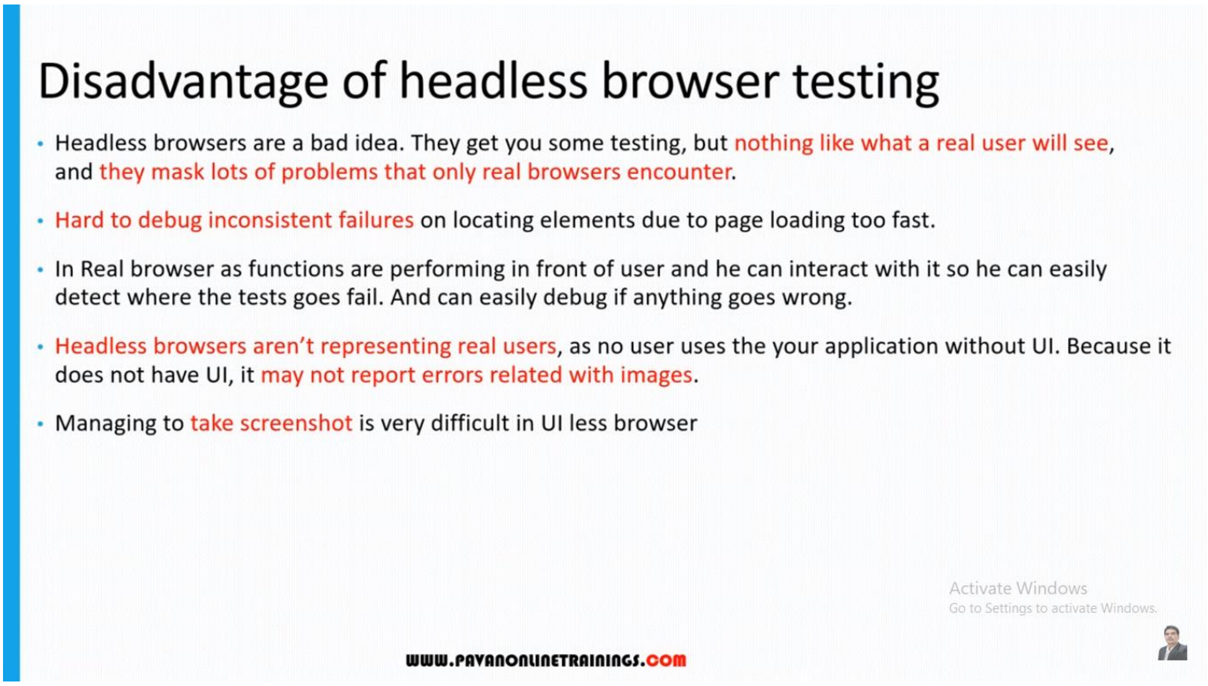
When to use headless browser testing ?

- We can use headless testing once the cross browser testing is completed and want to run regression test cases in subsequent releases and with continuous integration.
- You have no option other than using headless testing when your machine does not have a GUI, for instance if you want to run your tests in unix.
- It is recommended to use headless browser when tests are executed in parallel as User Interface based browsers consumes a lot of Memory / resources.

Advantages of headless browser testing?

Faster bcoz it doesn’t have UI

Disadvantages of headless browser testing?

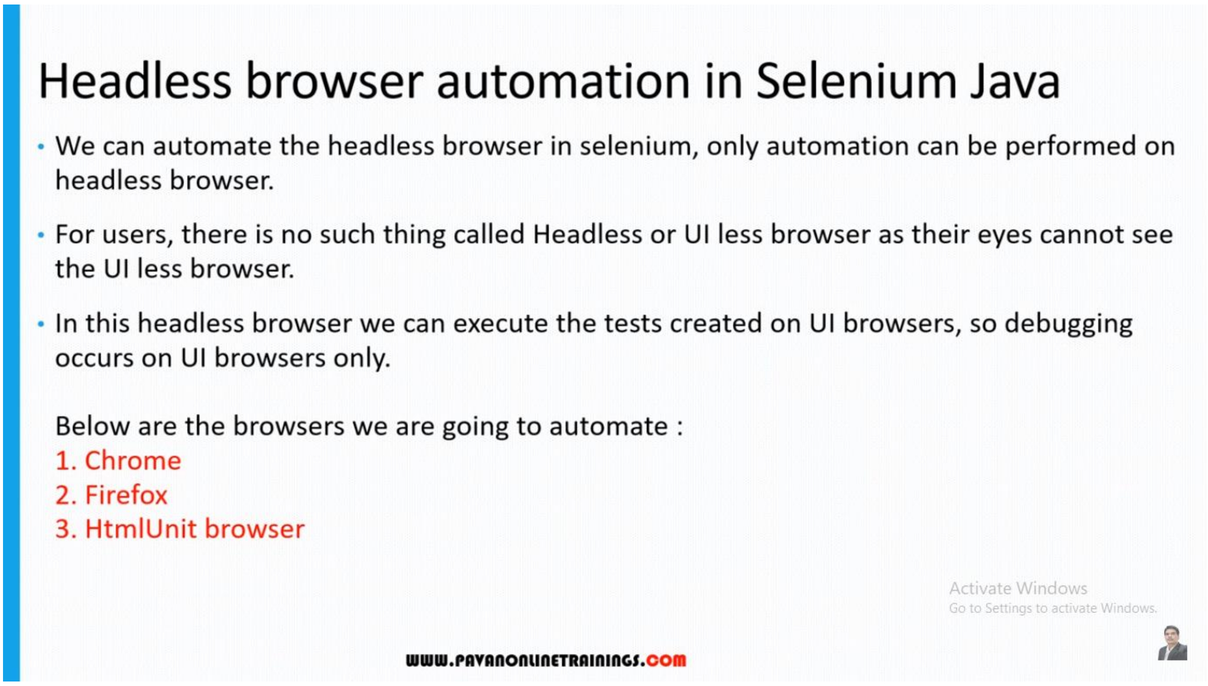


WWW.PAVANONLINETRAININGS.COM
Activate Windows
Go to Settings to activate Windows.

Disadvantage of headless browser testing

- Headless browsers are a bad idea. They get you some testing, but nothing like what a real user will see, and they mask lots of problems that only real browsers encounter.
- Hard to debug inconsistent failures on locating elements due to page loading too fast.
- In Real browser as functions are performing in front of user and he can interact with it so he can easily detect where the tests goes fail. And can easily debug if anything goes wrong.
- Headless browsers aren’t representing real users, as no user uses the your application without UI. Because it does not have UI, it may not report errors related with images.
- Managing to take screenshot is very difficult in UI less browser

Headless bowser automation in selenium?



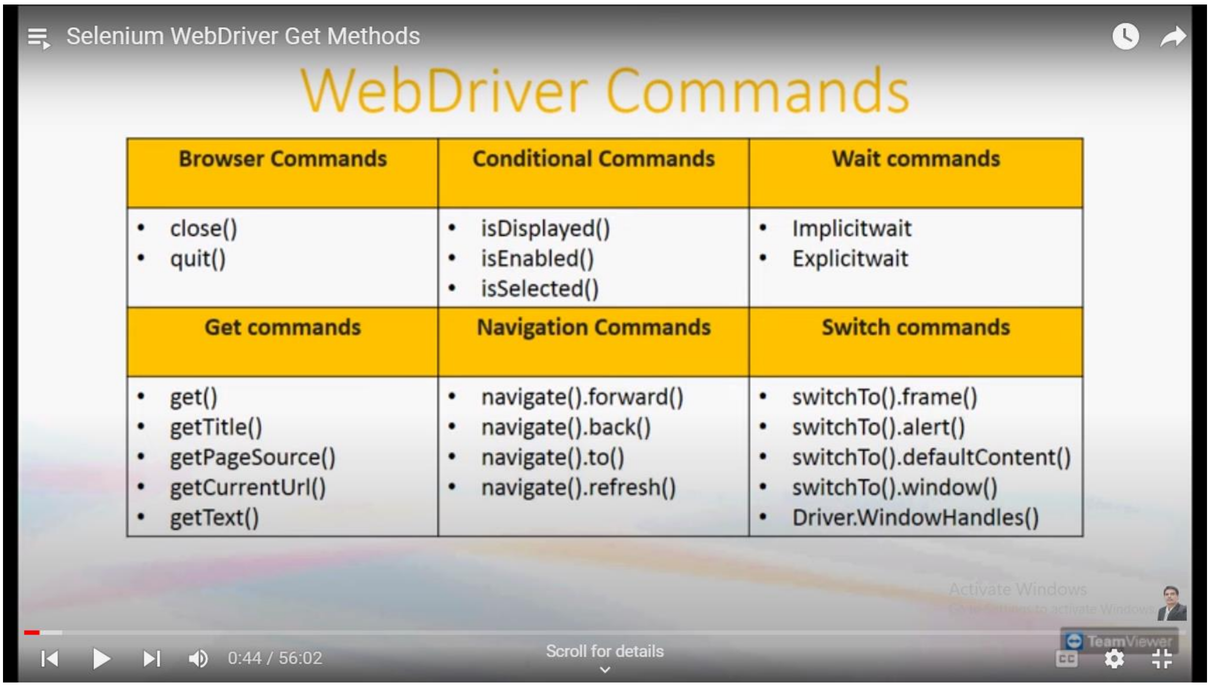
Headless browser automation in Selenium Java

- We can automate the headless browser in selenium, only automation can be performed on headless browser.
- For users, there is no such thing called Headless or UI less browser as their eyes cannot see the UI less browser.
- In this headless browser we can execute the tests created on UI browsers, so debugging occurs on UI browsers only.

Below are the browsers we are going to automate :
1. Chrome
2. Firefox
3. HtmlUnit browser

Note: HtmlUnit browser has no UI. Key value settings are not required.

WebDriver commands(methods):



WebDriver Commands

Browser Commands	Conditional Commands	Wait commands
• close() • quit()	• isDisplayed() • isEnabled() • isSelected()	• Implicitwait • Explicitwait
Get commands	**Navigation Commands**	**Switch commands**
• get() • getTitle() • getPageSource() • getCurrentUrl() • getText()	• navigate().forward() • navigate().back() • navigate().to() • navigate().refresh()	• switchTo().frame() • switchTo().alert() • switchTo().defaultContent() • switchTo().window() • Driver.WindowHandles()

Close vs quit:

```
driver.close();// it will close one window/tab of browser
driver.quit();// it will close multiple windows of browser
```

Navigate Commands:

1. driver.get() => first open any url by using this method. eg. www.gmail.com
2. navigate.To() => replace gmail url with new url. eg. new url. www.facebook.com
3. navigate.back() => again u want to go back to gmail page, we use this method. it perform back arrow operation.
4. navigate.forword() => again u want to go facebook page. we use this method. it perform forward arrow operation.
5. navigate.refresh() => refresh the page.

Conditional Commands: this commands always returns true or false.

isEnabled() : used to verify status of text boxes. Some times text boxes are disabled, we cant enter any value.

isDisplayed(): use to check particular web-element present on page or not.

If present return true. If not present return false.

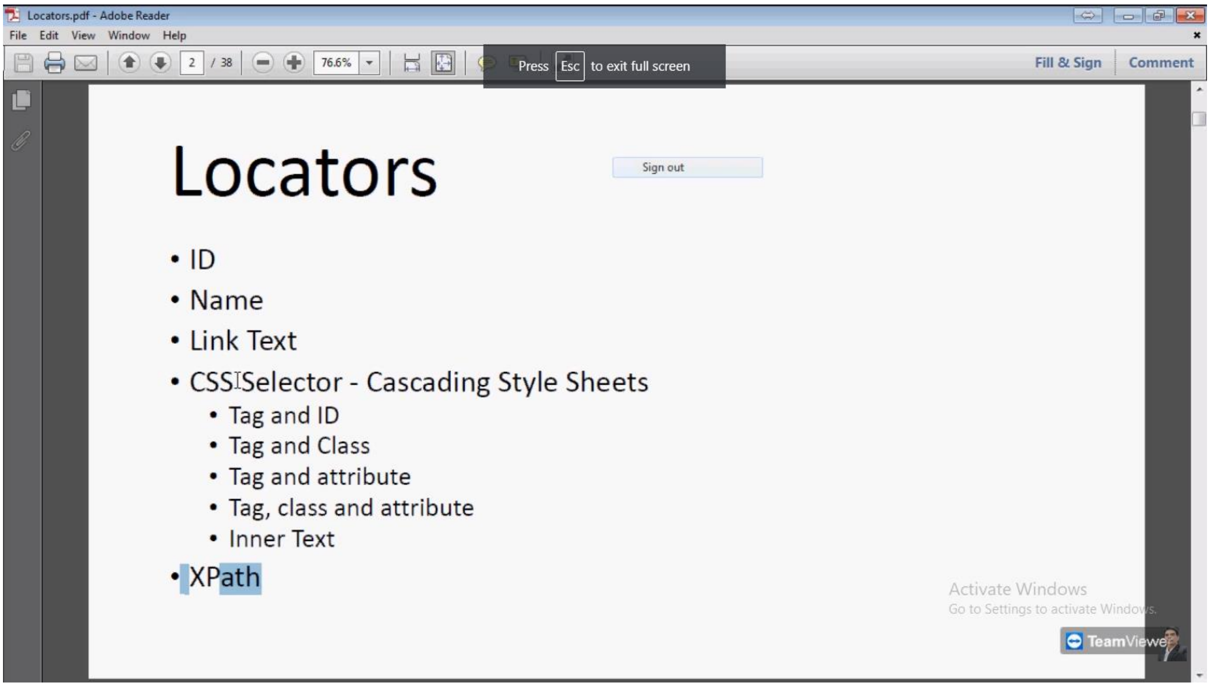
isSelected(): used for radio buttons or check boxes.

Locators Part1: (locators are used to fine web elements on web page)

Client: whatever user can see that is basically client.

3 tier application:
web server(UI), application server(backend), database server(db).

Different types of Locators:


Locators.pdf - Adobe Reader
File Edit View Window Help
2 / 38
76.6%
Press Esc to exit full screen
Fill & Sign
Comment
Sign out
Activate Windows
Go to Settings to activate Windows.
TeamViewer

Locators

- ID
- Name
- Link Text
- CSSSelector - Cascading Style Sheets
  - Tag and ID
  - Tag and Class
  - Tag and attribute
  - Tag, class and attribute
  - Inner Text
- XPath

Mainly we use xpath locator.

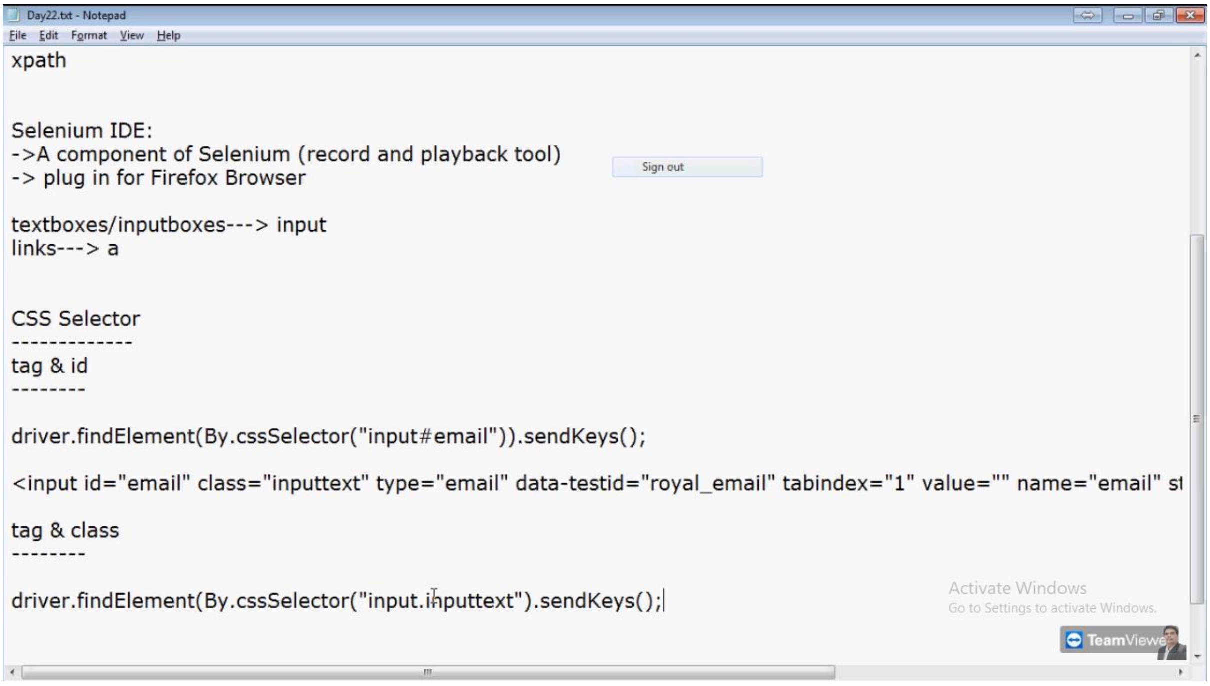
Selenium IDE: it is a component of selenium. It is a plugin only for firefox browser.(record and playback). It cannot perform complex operation. It is simple type of recoder.

1. ID locator:
```
driver.findElement(By.id("email")).sendKeys("digvijayptemp@gmail.com");
```
2. Name Locator:
```
driver.findElement(By.name("pass")).sendKeys("123456");
```
3. Link Text locator:
```
//link text used for only href link.
driver.findElement(By.linkText("Forgotten password?")).click();
```
4. Css selector combinations:

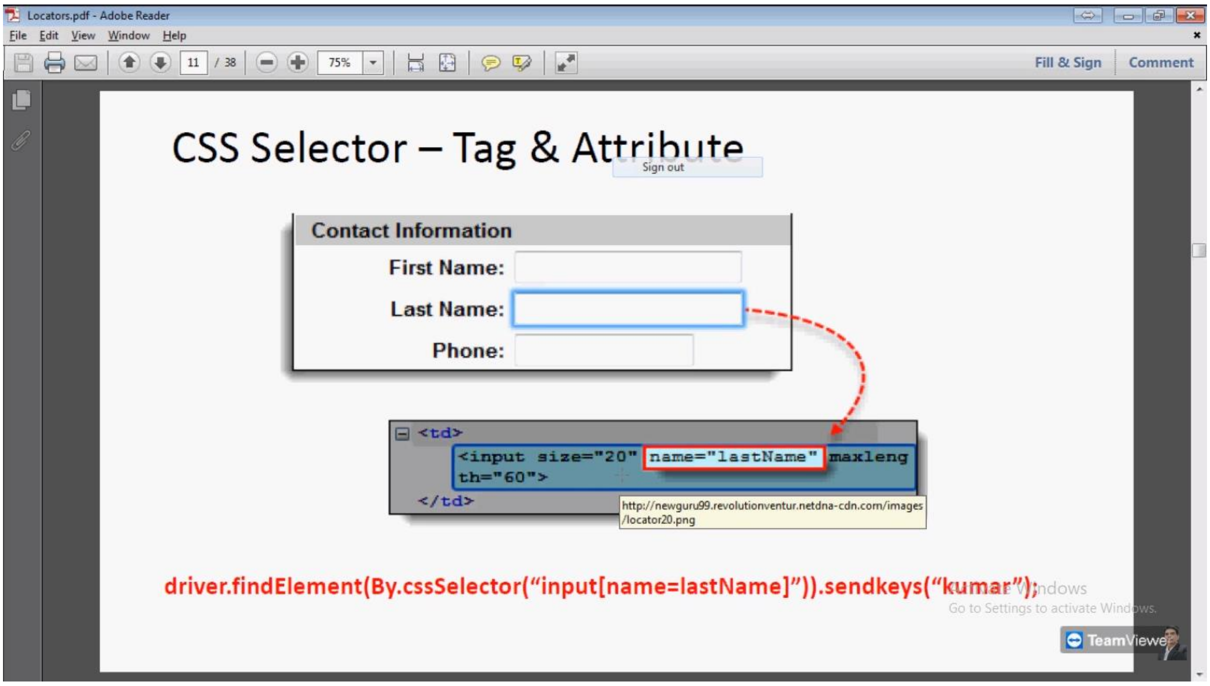
```
//in cssSelector diff combinations like:
Tag & id,
Tag & name
Tag & class,
Tag & attribute,
```

Tag & class & attribute
Inner text

```
// below we used tag & id here tag is "input" and id is "email"
 use # between them
driver.findElement(By.cssSelector("input#email")).sendKeys("digvijay@gmail.com");

//tag & class here tag is "a" and class is "_97w5" use dot between them
driver.findElement(By.cssSelector("a._97w5")).click();
```


Day22.txt - Notepad
File Edit Format View Help
Sign out
Activate Windows
Go to Settings to activate Windows.
TeamViewer

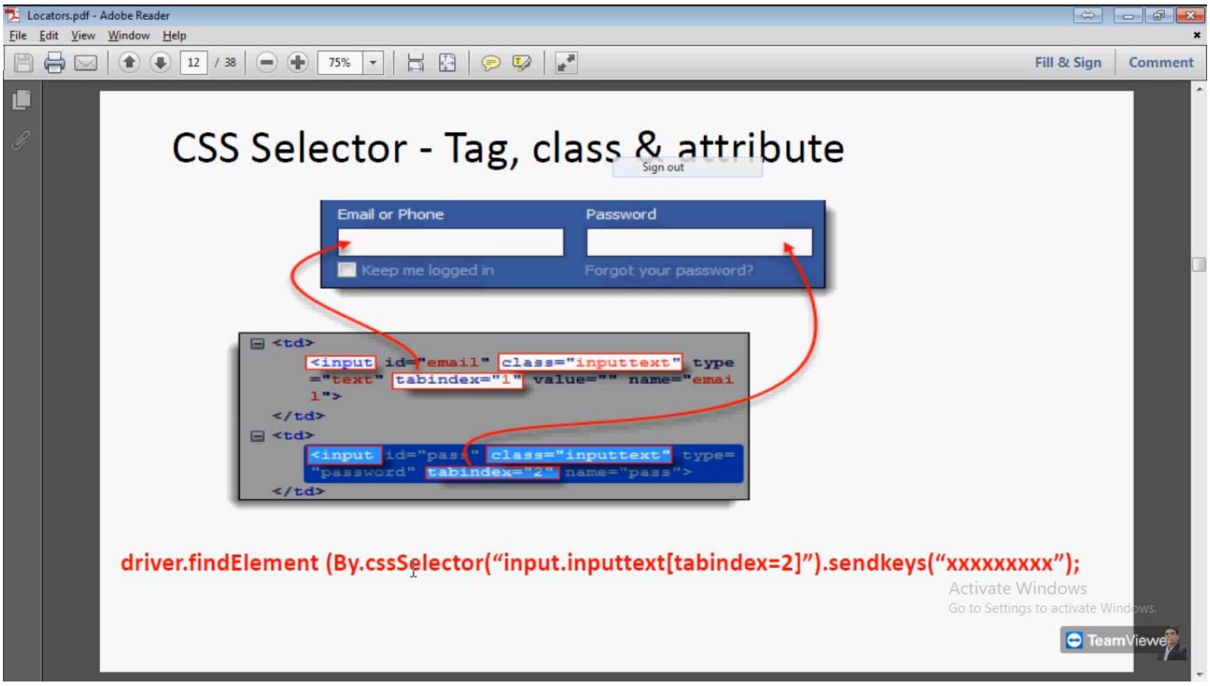
```
xpath


Selenium IDE:
->A component of Selenium (record and playback tool)
-> plug in for Firefox Browser

textboxes/inputboxes---> input
links---> a


CSS Selector
-------------
tag & id
--------

driver.findElement(By.cssSelector("input#email")).sendKeys();

<input id="email" class="inputtext" type="email" data-testid="royal_email" tabindex="1" value="" name="email" st

tag & class
--------

driver.findElement(By.cssSelector("input.inputtext").sendKeys();
```


Locators.pdf - Adobe Reader
File Edit View Window Help
11 / 38
75%
Fill & Sign
Comment
Sign out
Contact Information
First Name:
Last Name:
Phone:
http://newguru99.revolutionventur.netdna-cdn.com/images
/locator20.png
Activate Windows
Go to Settings to activate Windows.
TeamViewer

CSS Selector – Tag & Attribute

```
<td>
  <input size="20" name="lastName" maxleng
  th="60">
</td>
```

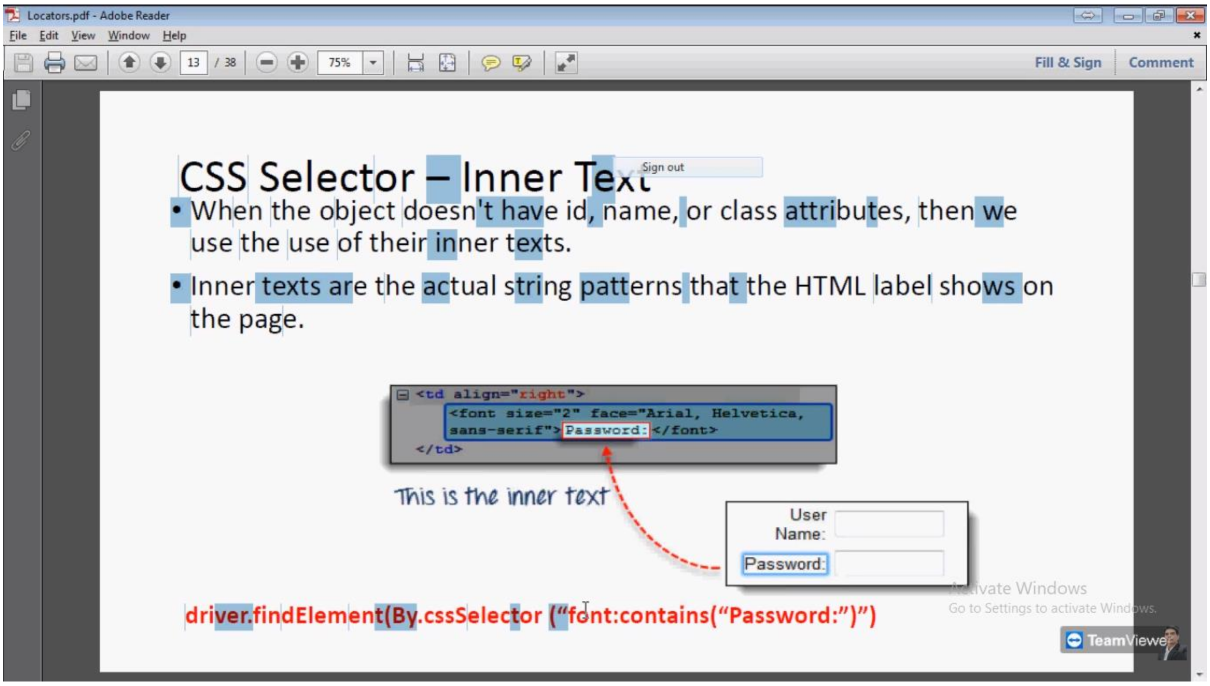
driver.findElement(By.cssSelector("input[name=lastName]")).sendkeys("kumar");



CSS Selector - Tag, class & attribute

```
<td>
    <input id="email" class="inputtext" type
    ="text" tabindex="1" value="" name="emai
    l">
</td>
<td>
    <input id="pass" class="inputtext" type=
    "password" tabindex="2" name="pass">
</td>
```

driver.findElement (By.cssSelector(“input.inputtext[tabindex=2]”).sendkeys(“xxxxxxxxx”);



CSS Selector – Inner Text

- When the object doesn't have id, name, or class attributes, then we use the use of their inner texts.
- Inner texts are the actual string patterns that the HTML label shows on the page.

```
<td align="right">
    <font size="2" face="Arial, Helvetica,
    sans-serif">Password:</font>
</td>
```

driver.findElement(By.cssSelector (“font:contains(“Password:”)”)

5. **Xpath locator:** this locator is very important and widely used.
   It gives exact address of web elements on web page. As compared to others locators.
   Two types of xpaths are there:
   1. Absolute Xpath: (chrome copy fullXpath)
   - complete/full address of web element in the web page.
   - In which we need to provide Entire path or entire address from source to destination this is called absolute Xpath.
     Eg. Our home address > country> state>district> sub district> town> village
     Eg. Html page there diff tags. <A tag > under <input tag > under another tag.
     Like wise
   - www.facebook.com
     1. copy full Xpath for email addres::
        /html/body/div[1]/div[1]/div[1]/div/div/div/div[2]/div/div[1]/form/div[1]/div[1]/input

   2. Relative Xpath: (chrome copy Xpath)
   - Partial address of web elements in web page.
   - no need to provide entire address from source to destination.
   - We need to provide only whichever is familiar with surrounding of destination and then we need to provide destination.
   - www.facebook.com
     2. Only copy Xpath for email addres:
        //*[@id="email"]

How to differentiate Absolute Xpath and Relative Xpath:

Absolute Xpath:

- /html/body/div[1]/div[1]/div[1]/div/div/div/div[2]/div/div[1]/form/div[1]/div[1]/input
- **In Absolute Xpath you will find multiple forward slashes (/) and all are single slashes**
- **Always start with single forward slash.**

Relative Xpath:

- //*[@id="email"]
- **Relative Xpath you will find double slash(//)**
- **Always starts with double forward slash.**
- **// * indicates that regular expression. this points to all tag names. means Skip all tags and comes to actual web element.**

Which one is better to use?

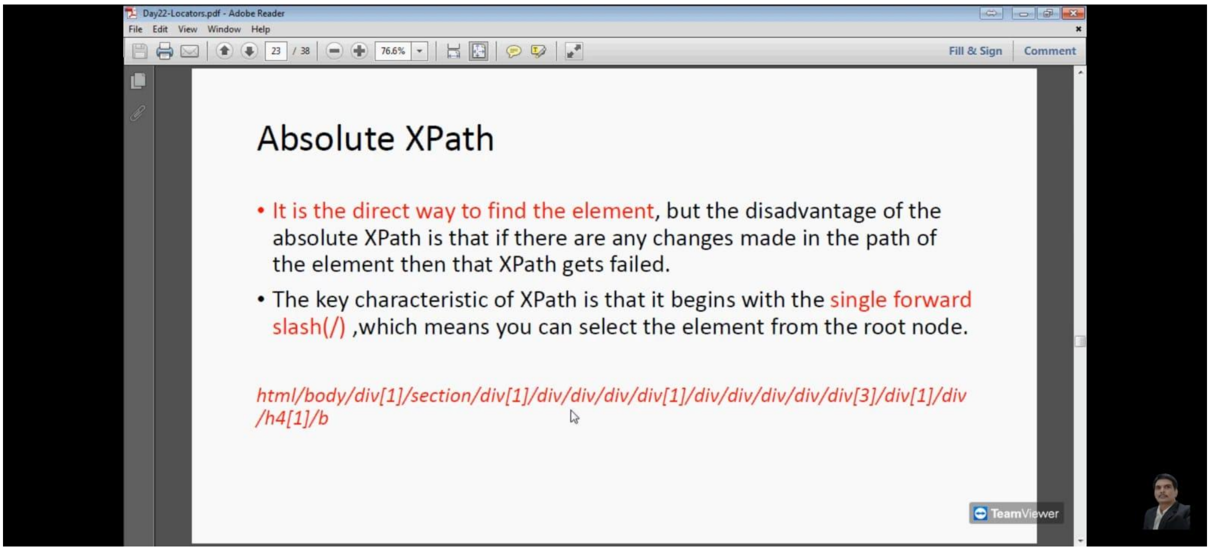
- We always prefer Relative Xpath.

Why not better Absolute path?

- /**html**/body/div[1]/div[1]/div[1]/div/div/div/div[2]/div/div/div[1]/form/div[1]/div[1]/**input** => under this input tag, actual element is present.
- Html: we know Html is start point of page. In above absolute path, path start from Html tag to Input tag.
- suppose developer inserted or deleted any feature on page. Then tag structure will change. Or entire hierarchy will change. So can't reach to our expected web element. Finally Absolute path will failed.
- Relative Xpath will not provide entire structure.
- **// * indicates that regular expression. this points to all tag names. means Skip all tags and comes to actual web element.**
- Relative Xpath not prefer entire structure. That is reason we mostly use Relative Xpath.

How to write Xpath manually?

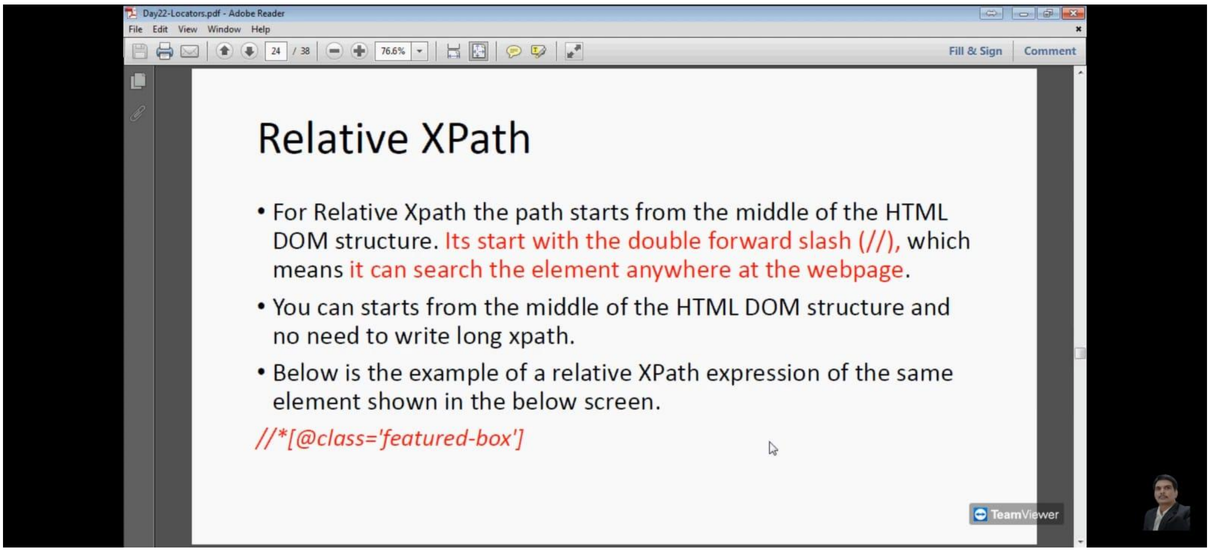
- Inspect the web page
- Make sure u r in element tab after inspect
- Open Html tag.
- Select pointer or firebug and move to Web element(eg. any input box)
- Location of that web element get highlighted in html script.
- Once location get highlighted open all tag from start
- Start with first tag i.e. /Html
- There is body tag . path will be /html/body
- Suppose there are multiple <div tags under <body tag. And ur web element is under first <div tag. Then ur path will be /html/body/div[1]
- Suppose under <div tag again there are three <div tag and ur web elemelnt in under second <div tag. Then ur path will be /html/body/div[1]/div[2]
- Then there is <form <tag and under form tag there is <table tag. Then path will be /html/body/div[1]/div[2]/form/table/
- Under <table tag there is <tbody tag and under <tbody tag there are multiple <tr tag and ur we element under third <tr tag. Path will be
- /html/body/div[1]/div[2]/form/table/tbody/tr[3]
- Under <tr tag there is multiple <td tag and ur web elememt under fourth <td tag . then path will be:
- /html/body/div[1]/div[2]/form/table/tbody/tr[3]/td[4]
- And there is input tag and under that input tag ur web web element is present. Path will be
- /html/body/div[1]/div[2]/form/table/tbody/tr[3]/td[4]/input
- **This way we can write path manually. In real practice we don't use. But for interview purpose we can prepare.**



Absolute XPath

- It is the direct way to find the element, but the disadvantage of the absolute XPath is that if there are any changes made in the path of the element then that XPath gets failed.
- The key characteristic of XPath is that it begins with the single forward slash(/) ,which means you can select the element from the root node.

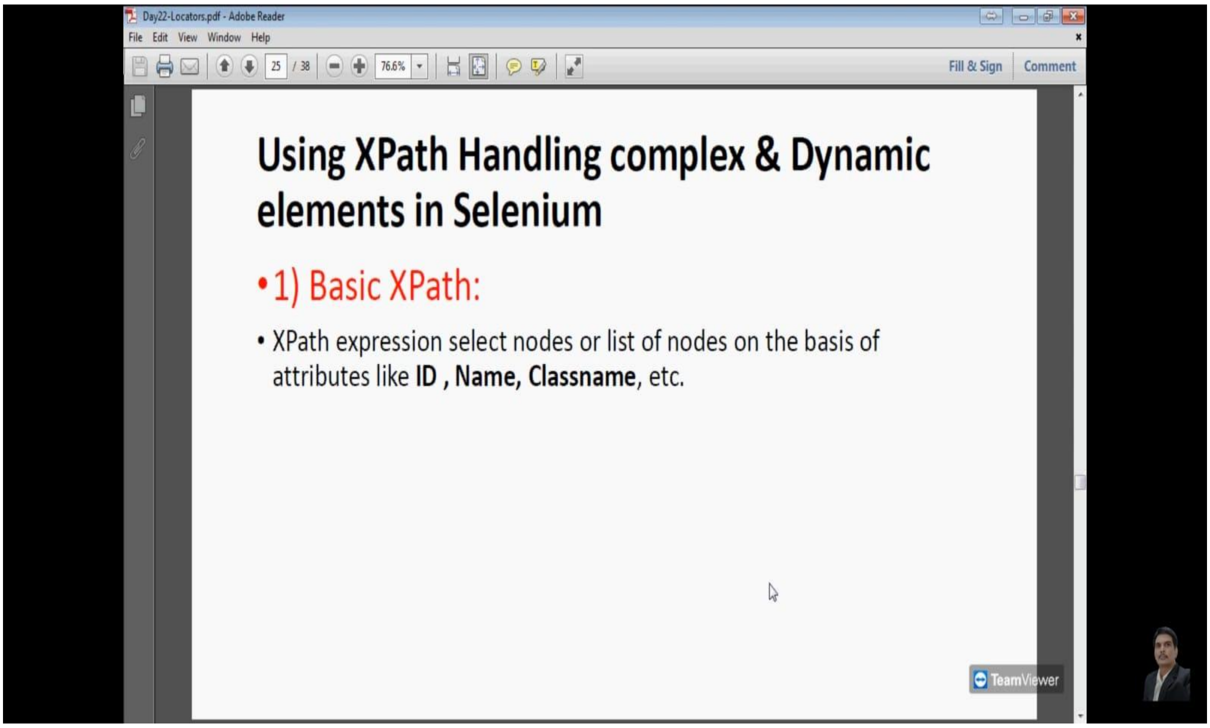
html/body/div[1]/section/div[1]/div/div/div/div[1]/div/div/div/div/div[3]/div[1]/div/h4[1]/b



Relative XPath

- For Relative Xpath the path starts from the middle of the HTML DOM structure. Its start with the double forward slash (//), which means it can search the element anywhere at the webpage.
- You can starts from the middle of the HTML DOM structure and no need to write long xpath.
- Below is the example of a relative XPath expression of the same element shown in the below screen.

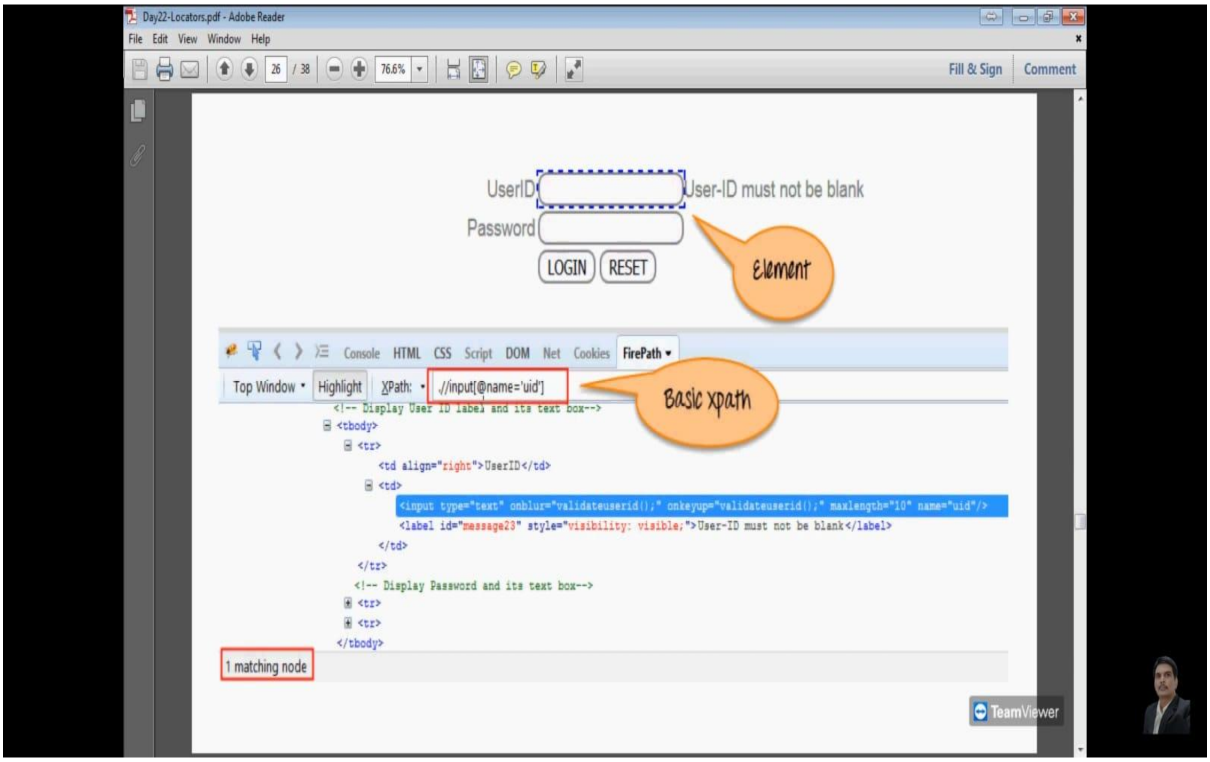
//[@class='featured-box']*



Using XPath Handling complex & Dynamic elements in Selenium

• 1) Basic XPath:

- XPath expression select nodes or list of nodes on the basis of attributes like **ID , Name, Classname**, etc.


Day22-Locators.pdf - Adobe Reader
File Edit View Window Help
26 / 38
76.6%
Fill & Sign
Comment
UserID
User-ID must not be blank
Password
LOGIN
RESET
Element
Console HTML CSS Script DOM Net Cookies FirePath
Top Window
Highlight
XPath:
Basic xpath
TeamViewer

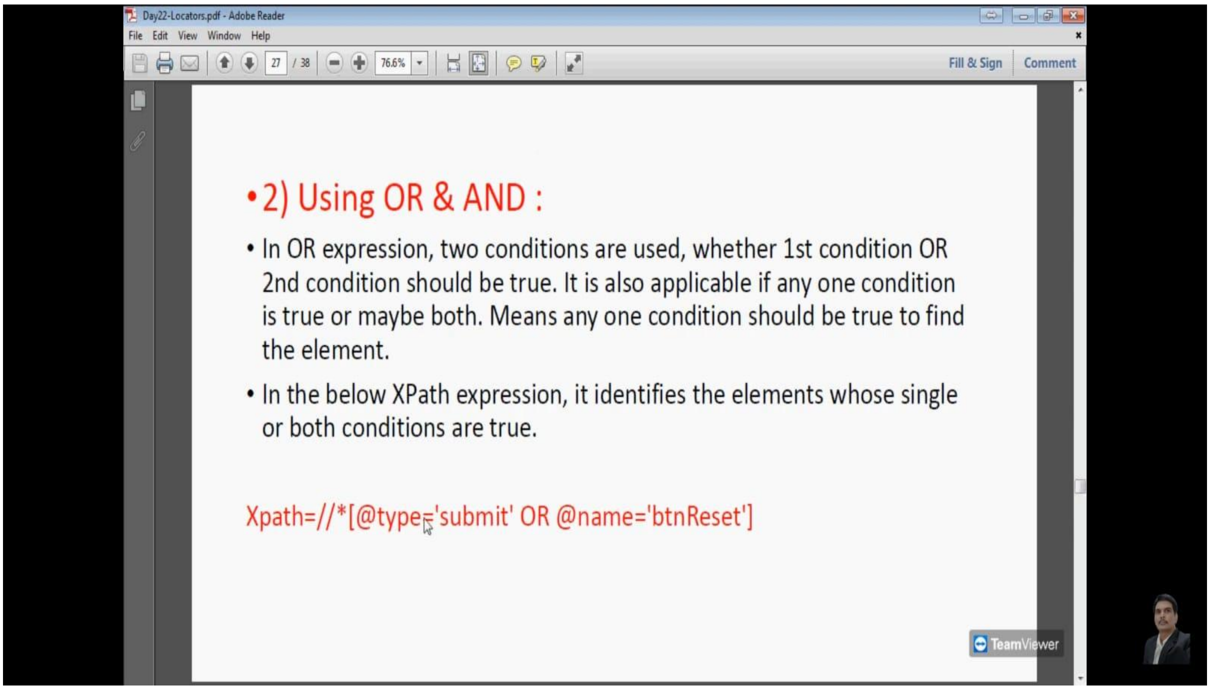
XPath: .//input[@name='uid']

```
<!-- Display User ID label and its text box-->
⊟ <tbody>
   ⊟ <tr>
        <td align="right">UserID</td>
      ⊟ <td>
           <input type="text" onblur="validateuserid();" onkeyup="validateuserid();" maxlength="10" name="uid"/>
           <label id="message23" style="visibility: visible;">User-ID must not be blank</label>
        </td>
     </tr>
     <!-- Display Password and its text box-->
   ⊞ <tr>
   ⊞ <tr>
  </tbody>
```

1 matching node

AND OR Operators in web elements:

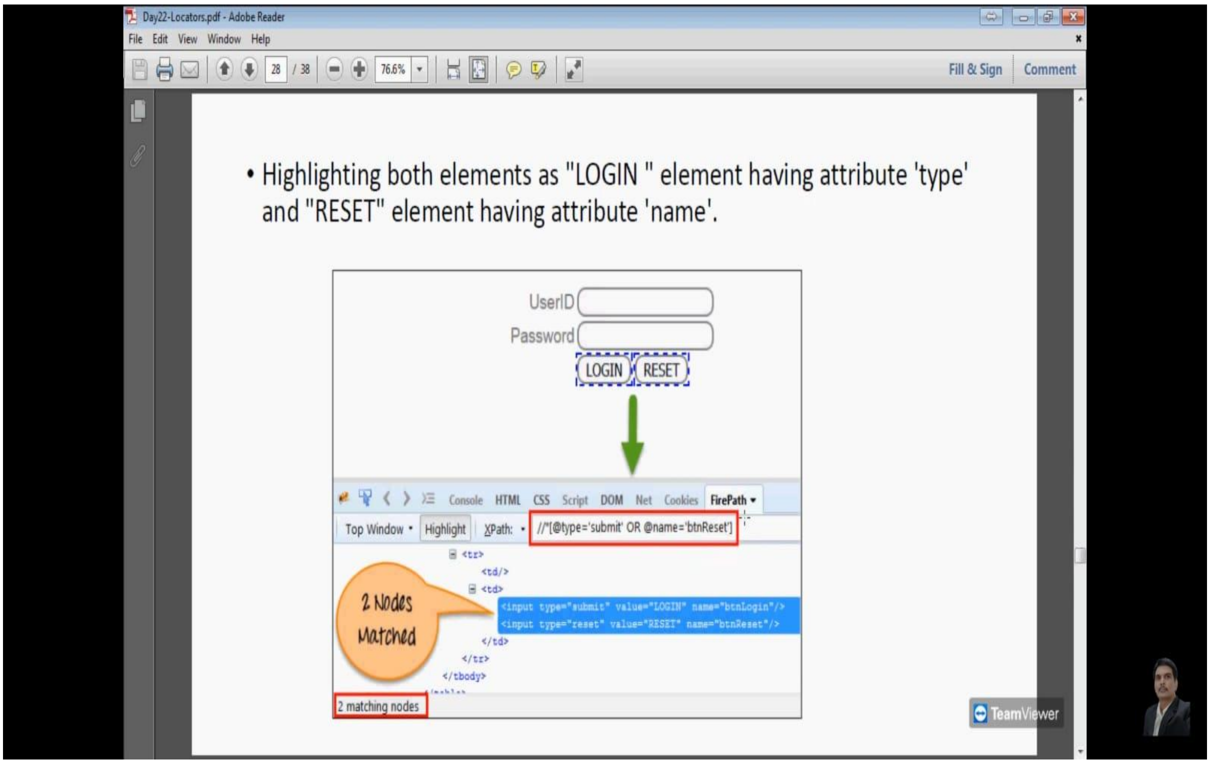
If web elements properties are changing dynamically .in that case we can use these operators.



•2) Using OR & AND :

- In OR expression, two conditions are used, whether 1st condition OR 2nd condition should be true. It is also applicable if any one condition is true or maybe both. Means any one condition should be true to find the element.
- In the below XPath expression, it identifies the elements whose single or both conditions are true.

Xpath=//*[@type='submit' OR @name='btnReset']

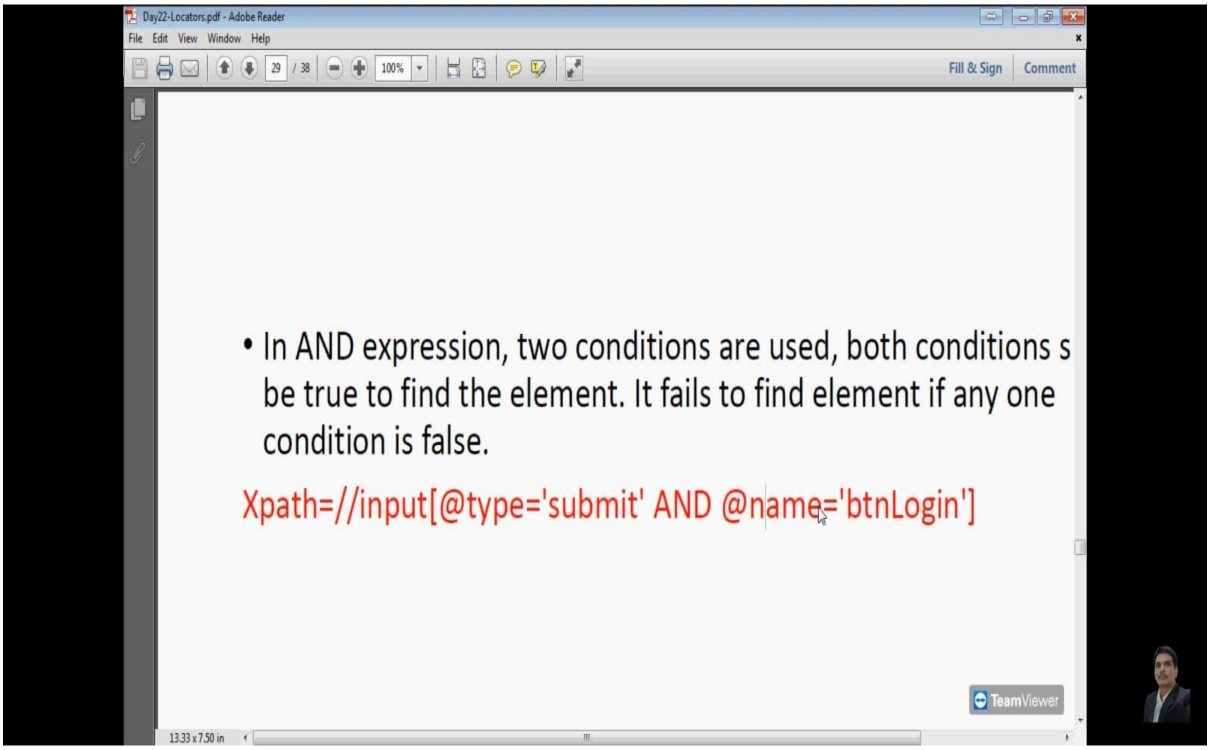


- Highlighting both elements as "LOGIN " element having attribute 'type' and "RESET" element having attribute 'name'.

//*[@type='submit' OR @name='btnReset']

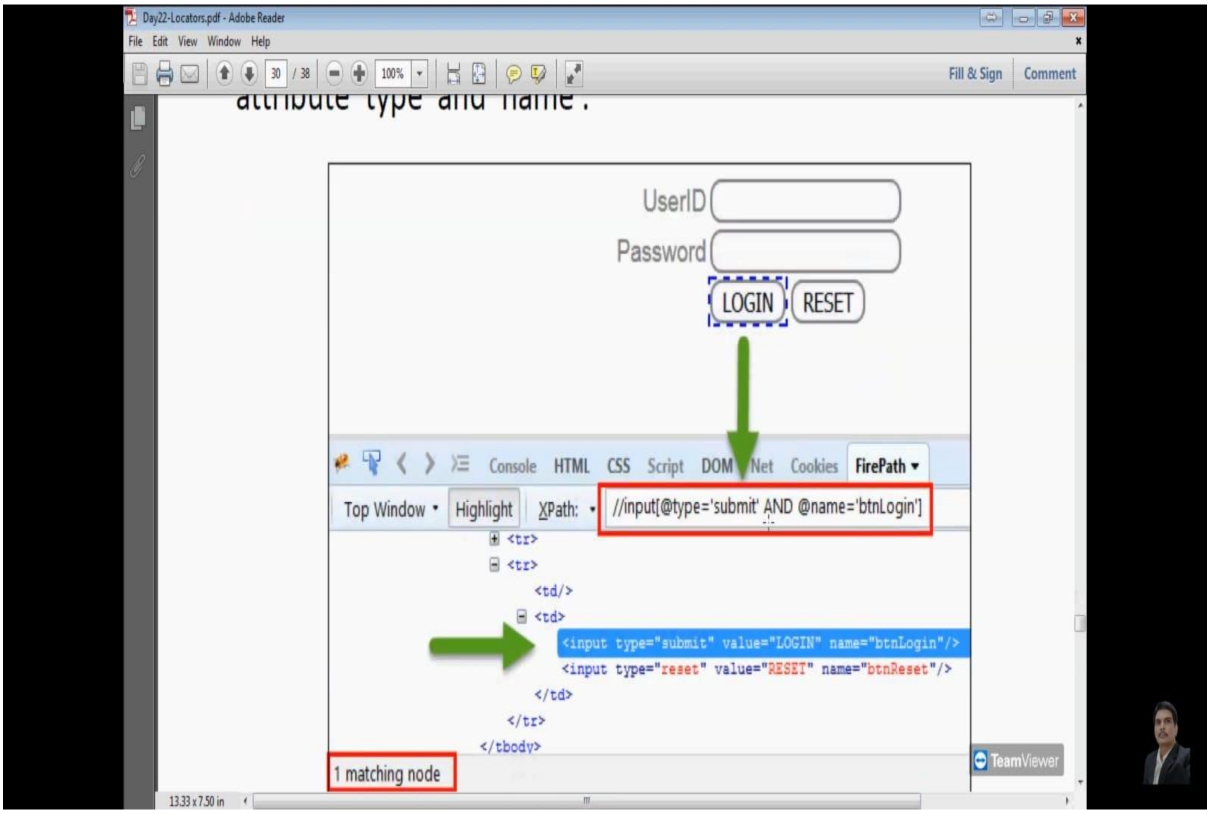
```
<tr>
  <td/>
  <td>
    <input type="submit" value="LOGIN" name="btnLogin"/>
    <input type="reset" value="RESET" name="btnReset"/>
  </td>
</tr>
</tbody>
```

2 matching nodes



- In AND expression, two conditions are used, both conditions s be true to find the element. It fails to find element if any one condition is false.

Xpath=//input[@type='submit' AND @name='btnLogin']



attribute 'type' and 'name'.

//input[@type='submit' AND @name='btnLogin']

```
<tr>
<tr>
    <td/>
    <td>
        <input type="submit" value="LOGIN" name="btnLogin"/>
        <input type="reset" value="RESET" name="btnReset"/>
    </td>
</tr>
</tbody>
```

1 matching node