

27-02-2017

First, it checks whether the condition is satisfied or not, if it is satisfied, it will go to next statement.

If the condition is not satisfied, then it checks, time over or not, If time is over, we get timeout exception. If time is not over, then it waits for 500 milliseconds, then it will continue, checking the condition.

```
public class Demo {
    public static void verifyTitle ( WebDriver driver, String eTitle )
    { WebDriverWait wait = new WebDriverWait ( driver, 5 );
        try {
            wait.until(ExpectedConditions.titleIs ( eTitle ));
            System.out.println ("PASS : Title is matching ");
        }
        catch ( TimeoutException ) {
            System.out.println ("FAIL : Title is NOT matching ");
        }
        System.out.println ("Expected Title : " + eTitle );
        String aTitle = driver.getTitle ();
        System.out.println ("Actual Title : " + aTitle );
    }
}

P.S.V.M ( String[] args ) {
    System.setProperty ("webdriver.gecko.driver",
        "./software/geckodriver.exe" );
    WebDriver driver = new FirefoxDriver ();
    driver.manage().timeouts().implicitlyWait ( 10, TimeUnit.SECONDS );
    driver.get ("http://localhost/login.do");
    driver.findElement ( By.id ("username") ).sendKeys ("admin");
    driver.findElement ( By.name ("pwd") ).sendKeys ("manager");
    driver.findElement ( By.xpath ("//div[text()='Login']" ) ).click ();
}
```

```

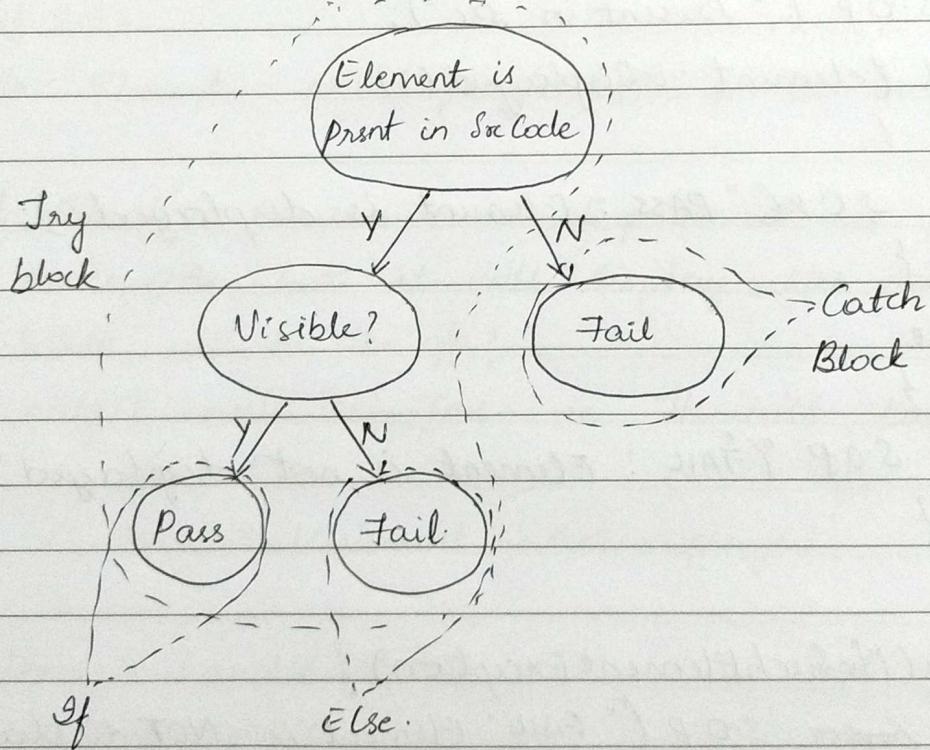
verifyTitle(driver, "actiTIME - Enter Time-Track");
driver.findElement(By.id("logoutlink")).click();
verifyTitle(driver, "actiTIME - Login");
driver.quit(); }

```

I.Q How do you verify whether a specified element is present or not on the Webpage?

→ Logic: Element is present in source code?

Yes? Visibility of Element? ← YES : PASS
 No: FAIL



Ex: HTML page .

```

<input type = "text" id = "t1" value = "admin">
<input type = "hidden" id = "t2" value = "admin">

```

```
P.S.V.M (String[] args) {  
    System.out.println("Enter the id of the Element (t1/t2/t3):");  
    Scanner sc = new Scanner (System.in);  
    String v = sc.nextLine();  
    sc.close();  
    System.setProperty ("webdriver.gecko.driver", "./software/geckodriver.exe");  
    WebDriver driver = new FirefoxDriver();  
    driver.get ("file:///E:/demo.html");  
    try {  
        WebElement element = driver.findElement(By.id(v));  
        System.out.println("Present in sec");  
        if (element.isDisplayed())  
            System.out.println("PASS : Element is displayed");  
        else  
            System.out.println("FAIL : Element is not displayed");  
    }  
    catch (NoSuchElementException) {  
        System.out.println("FAIL : Element is NOT displayed (not present in sec)");  
    }  
    driver.quit();  
}
```

Execute the above code with different inputs as shown below:

i/p: t1

o/p: Present in Src

PASS: Element is displayed

i/p: t2

o/p: Present in Src

FAIL: Element is NOT displayed

i/p: t3

o/p: Element is NOT displayed (Not present in src) : FAIL.

Q: In the above scenario, what if p element is present in the source code, but it will display after some time (Ex: 3secs). What will be the o/p?

→ output will be FAIL. ie. It will print

'present in Src'

FAIL: Fail: element is NOT displayed'

Q: How to handle it?

→ Using ExplicitWait():

Refer to the program in Next page ⇒

```

P.S.V.M (String[] args) {
    S.O.P ("Enter the id of the Element (t1/t2/t3):? ");
    Scanner sc = new Scanner (System.in);
    String v = sc.nextLine();
    sc.close();
    System.setProperty ("", "", "");
    WebDriver driver = new FirefoxDriver ();
    driver.get ("");
    WebDriverWait wait = new WebDriverWait (driver, 2);
    try {
        wait.until(ExpectedConditions.visibilityOfElementLocated
            (By.id (v)));
        S.O.P ("PASS: Element is visible");
    } catch (TimeoutException) {
        Sys.S.O.P ("FAIL: Element is not visible");
    }
    driver.quit();
}

```

Q: What are the differences between ImplicitWait() & ExplicitWait() :

To W

- 1) Do not specify the condition \Rightarrow Need to specify conditions
- 2) Timeout can be day, hour, etc \Rightarrow Only seconds.
- 3) We can handle synchronization findElement() & findElements()
- 4) After the timeout we get NoSuchElementException \Rightarrow Here we get TimeoutException

Eo W

- 3) We can handle synchronization of any method.

Q.9 How do you handle findElement() using explicit wait().

```
WebDriverWait wait = new WebDriverWait(driver, 10);
wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("logoutLink")));
driver.findElement(By.id("logoutLink")).click();
```

- * Handling multiple elements: To perform action on multiple elements, we use findElements() method + which returns list of WebElement.

List should be imported from java.util package.

If the specified locator is not matching with any of the element, then it returns empty list.

Ex: Google
 GMail

html

↳ body

 ↳ a
 ↳ a -

PSVM (String [] args) {

 System.setProperty(" ", " ");

 WebDriver driver = new ChromeDriver();

 driver.get("file:///E:/demo.html");

 List<WebElement> allLinks = driver.findElements(By.tagName("a"));

 int count = allLinks.size();

 S.O.P(count);

 WebElement link = allLinks.get(0);

 String text = link.getText();

 S.O.P(text);

 driver.quit();

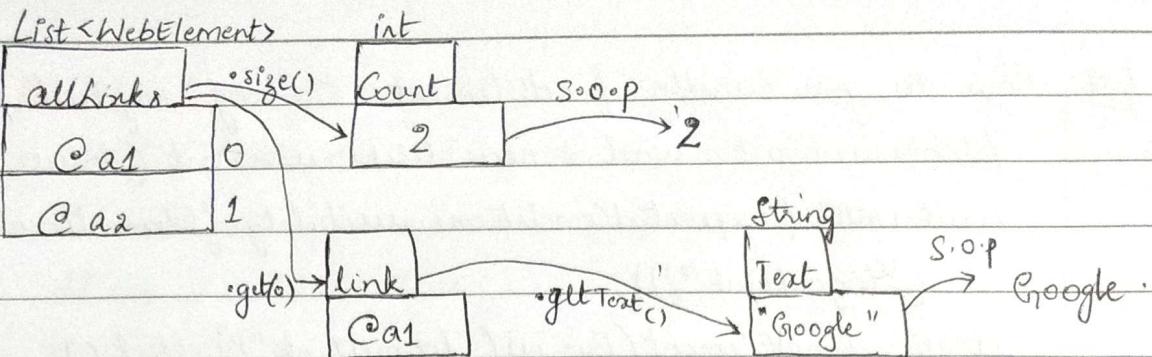
}

html

```

  / body
  |   |
  |   |   ↗ a
  |   |   ↗ a
  |   |
  |   ↗ allLinks
  |   ↗ size()
  |   ↗ Count
  |   ↗ 2
  |   ↗ S.O.P
  |   ↗ 2
  |   ↗ link
  |   ↗ get()
  |   ↗ @a1
  |   ↗ Text
  |   ↗ "Google"
  |   ↗ String
  |   ↗ S.O.P
  |   ↗ Google
  ↗ @a1
  ↗ getText()
  ↗ "Google"
  ↗ String
  ↗ S.O.P
  ↗ Google

```



T.Q: Write a script to count no of links in flipkart.com.

T.Q Write a script to print text of all the links present in google.com.

T.Q Write a script to count no of checkboxes present on the page and select all of them using enhanced for loop & deselect all of them in reverse order.

```

→ public void (String [] args) {
    WebDriver driver = new WebDriver FirefoxDriver();
    driver.get ("file:///E:/demo.html");
    String xp = "// input[@ type = 'checkbox']";
    List<WebElement> allCB = driver.findElements (By.xpath (xp));
    int count = allCB.size();
    System.out.println (count);
    for (WebElement cb : allCB)
        { cb.click(); }
    for (int i = count - 1; i >= 0; i--)
        { allCB.get (i).click(); }
    allCB.get (0).click(); //select 1st CB
    allCB.get (count - 1).click(); //select last CB
}

```

I.Q. Write a script to select alternative checkboxes present on the page?

I.Q. Write a script to select first & last checkbox, using ^{without} findElements.

→ By using findElement & xpath & groupIndex.

* Handling standard ^{AUTO} suggestions:

We handle auto suggestions by using findElements().

Ex: → Open Browser

2) Enter Google.com

→ Type java

→ Get all the auto suggestions

5) Print the count.

6) Print text of all the suggestions.

7) Select the last one.

P. S. V. M (String[] args) {

System.setProperty(" ", " ");

WebDriver driver = new FirefoxDriver();

driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

driver.get("http://www.google.com");

driver.findElement(By.id("lst-ib")).sendKeys("java");

String xp = "//div[@class='sbqs-c']";

List<WebElement> allAST = driver.findElements(By.xpath(xp));

int count = allAST.size(); S.O.P(count);

for(int i=0; i<count; i++) { WebElement ast = allAST.get(i);

String text = ast.getText(); S.O.P(text); }

allAST.get(count-1).click(); }

Handling WebTables

E.g.

```

<table id="t1" border="1">
  <tr>
    <td> SL No </td>
    <td> Sub </td>
    <td> Cost </td>
  </tr>
  <tr>
    <td> 1 </td>
    <td> QC </td>
    <td> 200 </td>
  </tr>
  <tr>
    <td> 2 </td>
    <td> QA </td>
    <td> 350 </td>
  </tr>
</table>
```

Q: Write a script/code to print content of the table?

```

⇒ P.S.V.M {String[] args) {
  System.setProperty(" ", " ");
  WebDriver driver = new FirefoxDriver();
  driver.get("file:///D:/table.html");
  WebElement table = driver.findElement(By.id("t1"));
  List<WebElement> allTD = table.findElements(By.tagName("td"));
  for (WebElement td : allTD) { System.out.println(td.getText()); }
  driver.quit();
}
```

<u>Method</u>	<u>Scope</u>	<u>Returns</u>
1) driver.findElement	Complete Page	1st Matching Element
2) driver.findElements	Complete Page	All the matching Elements
3) element.findElement	Inside current Element	1st Matching Element
4) element.findElements	Inside current Elements	All the matching Elements

S.Q. To print only the numbers present inside the table?
and also print sum of that.

```

P.S.V.M (String[] args)
{
    System.setProperty(" ", " ");
    WebDriver driver = new FirefoxDriver();
    driver.get("file:///D:/table.html");
    WebElement table = driver.findElement(By.id("t1"));
    List<WebElement> allTD = table.findElements(By.tagName("td"));
    int sum = 0;
    for (WebElement td : allTD)
    {
        String text = td.getText();
        try {
            int v = Integer.parseInt(text);
            S.O.P(v);
            sum = sum + v;
        }
        catch (NumberFormatException e) {}
    }
    S.O.P(sum);
    driver.quit();
}
  
```

Assign Q. Write a script to print content of the table in table format. (By using nested for loop)

Assign Q. Write a script to print sum of the numbers present in the specified column.

-x-x HANDLING LIST BOX

There are two types of list box

- ↳ Multi Select list box
- ↳ Single Select list box (Dropdown list / Compo box)

Sample page :

<html>

<body>

```
<select id="mtr" multiple>
    <option value="a"> Idly </option>
    <option value="b"> Vada </option>
    <option value="c"> Poori </option>
    <option value="d"> Poori </option>
</select> <br> <br> <br>
<select id="slv" >
    <option value="a"> Tea </option>
    <option value="b"> Coffee </option>
    <option value="c"> Juice </option>
    <option value="d"> Juice </option>
</select>
```

</body>

</html>

To handle the list box, we use select class, it should be imported from org.openqa.selenium.support.ui

Methods of Select Class:

- 1) Select By Index
- 2) Select By Value
- 3) Select By Visible Text
- 4) deselect By Index
- 5) deselect By Value
- 6) deselect By Visible Text
- 7) deselect All
- 8) is Multiple
- 9) get options
- 10) get all Selected options
- 11) get first Selected options

NOTE ➔ Value & Text are case sensitive.

- ➔ If specified index value or text is invalid, we get NSEE for both select & deselect methods.
- ➔ Deselect method only works on
 - ➔ Multi Select list box
 - ➔ If we try to use it on single select list box, we get unsupported Operation exception.
- ➔ To check the listbox type, we use isMultiple() method

Ex:

```
P. S V M (String[] args) {  
    System.setProperty ("", "");  
    WebDriver driver = new FirefoxDriver();  
    driver.get ("file:///D:/1 hotel.html");
```

```

WebElement listbox = driver.findElement(By.id("slv"));
Select select = new Select(listbox);
select.selectByIndex(2);
select.selectByValue("b");
select.selectByVisibleText("Tea");
if (select.isMultiple())
    {
        select.deselectByIndex(2);
        select.deselectByValue("b");
        select.deselectByVisibleText("Tea");
        select.deselectAll();
    }
else
{
    System.out.println("Sorry");
}
}

```

09/03/2017

- * In multiselect listbox, if specified option (value/text), is duplicate, then it selects all the matching options
- * In single select listbox, if the specified option is duplicate, it will select 1st matching option.
- * To handle the duplicate option, we can use index.

I-Q: Write a script to count the no of options in listbox and print all the options present in the listbox.

P.S.V.M (String[] args){

System.setProperty(" ", " ");

WebDriver driver = new FirefoxDriver();
driver.get("file:///D:/hotel.html");

```
 WebElement listBox = driver.findElement(By.id("mtc"));
 Select select = new Select(listBox);
 List<WebElement> allOptions = select.getOptions();
 int count = allOptions.size();
 System.out.println(count);
 for(int i=0 ; i < count ; i++) {
 WebElement option = allOptions.get(i);
 String text = option.getText();
 System.out.println(text);
 }
 }
```

Q. Write a script to print the content of listBox in reverse order.

Q. Write a script to select and deselect all the options in multi select listBox in reverse order.

Q. Write a script to print content of the listbox in sorted order

//generic method to get listbox content

```
public static ArrayList<String> getAllTextFromListBox(WebElement listBox)
```

```
Select select = new Select(listBox);
List<WebElement> allOptions = select.getOptions();
ArrayList<String> allText = new ArrayList<String>();
for(WebElement option : allOptions) {
 String text = option.getText();
 allText.add(text);
}
return allText;
```

```

P.S.V.M (String[] args) {
    System.setProperty("","");
    WebDriver driver = new ChromeDriver();
    driver.get("http://www.facebook.com");
    WebElement listBox = driver.findElement(By.id("months"));
    ArrayList<String> allText = getAllTextFromListBox(listBox);
    Collections.sort(allText);
    for (String text : allText) {
        System.out.println(text);
    }
}

```

<u>O/P</u>	Apr	Jun
	Aug	Mar
	Dec	May
	Feb	Month
	Jan	Nov
	Jul	Oct
		Sept.

T.Q Write a script to check whether the listBox content is in sorted order or NOT.

```

P.S.V.M (String[] args) {
    System.setProperty("","");
    WebDriver driver = new ChromeDriver();
    driver.get("file:///D:/hotel.html");
    WebElement listBox = driver.findElement(By.id("slv"));
    ArrayList<String> allText = getAllTextFromListBox(listBox);
    ArrayList<String> clone = new ArrayList<String>(allText);
    Collections.sort(clone);
}

```

```

if (allText.equals(clone)) {
    S.O.P("ListBox is Sorted");
} else {
    S.O.P("ListBox is NOT sorted");
}

```

I.Q: Write a script to check whether specified option is present in list or not.

```

WebElement
ArrayList listBox = driver.findElement(By.id("slv"));
ArrayList<String> allText = getAllTextFromListBox(listBox);
String value = "Tea";
if (allText.contains(value)) {
    S.O.P("Present");
} else {
    S.O.P("Not Present");
}

```

03-03-2017

Q Write a script to check whether listbox has any duplicate option or not

Q Print the content of the listbox without duplicates.

```

WebElement listBox = driver.findElement(By.id("inti"));
ArrayList<String> allText = getAllTextFromListBox(listBox);
int count1 = allText.size();
HashSet<String> clone = new HashSet<String>(allText);
int count2 = clone.size();
if (count1 == count2) {
    S.O.P("ListBox has No Duplicates");
} else {
    S.O.P("ListBox has Duplicates");
}

```

```
//printing the listbox content without duplicates  
for (String text : clone) {  
    System.out.println(text); }  
driver.quit();  
}
```

I.Q: Print only the duplicate options present in the listBox.

```
public void main(String[] args) {  
    WebElement listBox = driver.findElement(By.id("nts"));  
    ArrayList<String> allText = getAListFromListBox(listBox);  
    HashSet<String> clone = new HashSet<String>();  
    for (String v : allText) {  
        if (!clone.add(v)) {  
            System.out.println(v); }  
    }  
}
```

I.Q How do you handle dropdown menu?

⇒ dropdown menu is element on which if we move the mouse pointer, it will display list of options.
⇒ Moving the mouse on the element is called as mouseovers.
This can be done using moveToElement() method of Actions class. It has parameterized constructor which takes an argument of type WebDriver. (app^{and} - actionind)

Ex: String xp = "//span[text()='About Company']";
WebElement menu = driver.findElement(By.xpath(xp));

Actions actions = new Actions(driver);

//mouse hovers on menu.

actions.moveToElement(menu).perform();

driver.findElement(By.linkText("Basic Facts")).click(); }

- x. Note: Whenever we call any method of Actions class, we must call perform() method at the end.

To Try → Appn: OrangeHRM

Asgi Write a script to select online enrollment present in ISQB.in path → foundation → Enrollment → Corporate Enrollment → online enrollment.

IQ How to handle context menu?

- ⇒ Right clicking on the element is called as context click.
- ⇒ When we right click on any element, Browser will display list of options which is called as context menu.
- ⇒ To right click, we use contextClick() method of Actions class. and to select the options present on the context menu, we press the shortcut such as t (New Tab), w (New Window) etc.
- ⇒ To press the shortcut we use sendKeys method of Actions class.
- ⇒ If it is not working, then we can use Robot class of Java.awt package. (AWT = Abstract Window Toolkit)

Ex:

```
Actions actions = new Actions (driver);  
actions.contextClick (element).perform ();  
//actions.sendKeys ("t").perform ();  
Robot r = new Robot ();  
r.keyPress (KeyEvent.VK_T);  
r.keyRelease (KeyEvent.VK_T);
```

I.Q.: How do you perform drag and drop in Selenium?
⇒ using action class.

site : igoogle drag & drop. (i-google-like-drag-drop)

```
String xp1 = "//h1[\" = 'Block 1\"]";  
WebElement b1 = driver.findElement(By.xpath(xp1));  
String xp2 = "//h1[\" = 'Block 3\"]";  
WebElement b2 = driver.findElement(By.xpath(xp2));  
Actions actions = new Actions(driver);  
actions.dragAndDrop(b1, b3).perform();
```

Using Action class, we can also perform doubleClick();
 ⇒ actions.doubleClick(element).perform();

* Handling Pop-Ups :

In selenium pop-ups are categorized into following types:

- 1) Alert & Confirmation popup.
- 2) Hidden Division popup.
- 3) File upload Popup
- 4) File download popup.
- 5) Child Browser popup.
- 6) Window popup.

* Handling ALERT & CONFIRMATION Pop-UP

Characteristics:

- If the firefox browser is opened by Selenium, then the alert popup displayed will be having following characteristics:
 - o We cannot inspect the pop-up, But we can move it.
 - o ~~We cannot~~ It will have 'OK' button with warning symbol !
 - o If the popup has 'OK' & 'CANCEL' button with confirmation symbol ?, then it is called as confirmation pop-up.
 - o Both alert & confirmation popup are also called as javascript popup.
 - o If the Browser is opened by the user, then pop-up will have whitebackground without any border, which cannot be moved.
 - o To handle this pop-up, we use switchTo.alert(), to transfer control to the pop-up. In order to get the message present on pop-up. we use getText(); to

click 'OK' we use accept() & to click on 'CANCEL' we use 'dismiss()'. All these methods are present in Alert Interface. Same methods works on confirmation pop-up.

g) Once the popup closes, the control is automatically transferred to webpage.

```
Ex P S V M (String[] args) {  
    System.setProperty(" ", " ");  
    WebDriver driver = new FirefoxDriver();  
    driver.get("http://www.actimind.com/contact.php");  
    String xp = "//input[@value = 'Submit Request']";  
    driver.findElement(By.xpath(xp)).click();  
    try {  
        Alert alert = driver.switchTo().alert();  
        String msg = alert.getText();  
        S.O.P(msg);  
        alert.accept(); }  
        { alert.dismiss(); }  
        used for cancelling  
    catch (NoAlertPresentException e) {  
        S.O.P("Alert is not displayed");  
    }  
}
```

* Handling HIDDEN DIVISION Popup:

Characteristics:

a) We can inspect the popup.

b) We cannot move the popup.

c) Colourfull.

d) We can handle the popup using findElement method.

Ex. Selecting the DOB in calendar.

```
PSVM (String[] args) {  
    System.setProperty(" ", " ");  
    WebDriver driver = new FirefoxDriver();  
    driver.get("https://retail.starhealth.in/renewal");  
    driver.findElement(By.id('dtDOB')).click();  
    String xp = "(//div[@id='date-picker']//a[text()='3'])[1]";  
    driver.findElement(By.xpath(xp)).click();
```

Assign: Write a script to perform following steps:

→ Open the browser & go to cleartrip.com.

→ click on 'pick a date'.

→ Select a date.

//span[.= 'March']//..//a[.= '5'] → Specific

//span[.= " + month + "]//..//a[.= " + day "] → Dynamic

Assign Write a script to select todays date in the calendar.

* Convert the date into required format.

```
PSVM (String[] args)
```

```
SimpleDateFormat s = new SimpleDateFormat("dd/MMM/yy");
```

```
Date d = new Date();
```

```
String res = s.format(d);
```

```
s.o.p(res);
```

* Handling FILE UPLOAD popup:

Characteristics:

a) Clicking on browse button will display a popup with the title file upload.

by we can move the popup, but we cannot inspect
c) This popup is used to select the file to be uploaded.

To handle this popup, please specify absolute path of the file using double backward slash (\ \) as argument for sendKeys() method.

```
Ex: P S V M (String[] args){  
    System.setProperty ("", "");  
    WebDriver driver = new FirefoxDriver();  
    driver.get ("http://www.actimind.com/contact.php");  
    driver.findElement (By.name("file")).sendKeys ("d:\\My.doc");  
}
```

SendKeys() method works only if 'Browse' button is present. If it is other type of buttons or icons, like attachment icon (✉) in gmail → (U), then we write a code to click on attachment icon using findElement() and then we use AutoIT to handle the popup.