

Selenium WebDriver :-

what are web Elements:-

1. Selenium Sees everything on the page like textbox, button, link, dropdown as a web Element.
2. For simple ones or most of them, web Element type is used.

eg:- textbox, button, link.

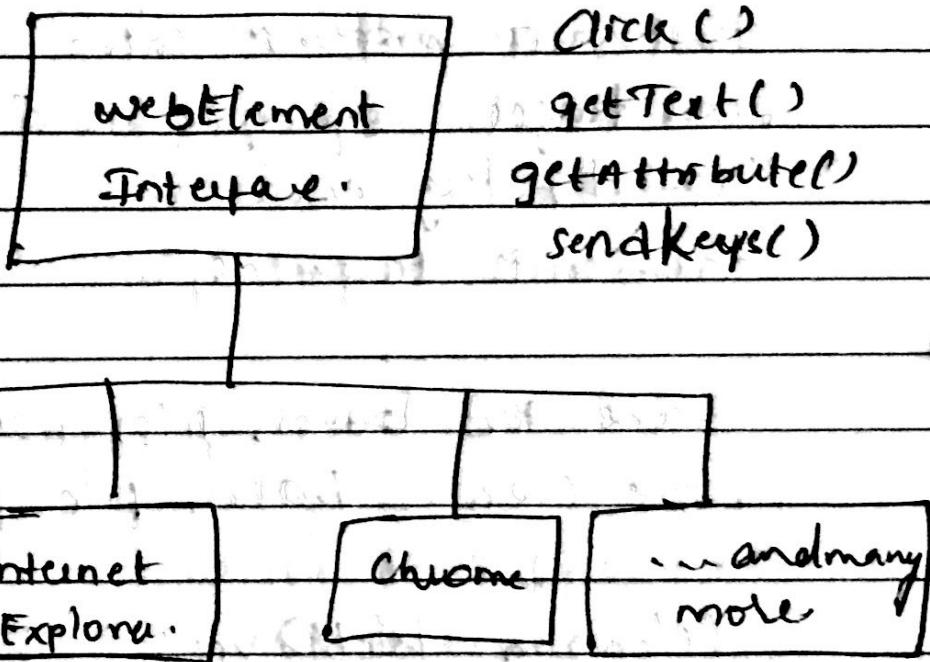
download geckodriver exe file from seleniumhq.org download
System.setProperty("webdriver.gecko.driver", "C:\\Users\\

→ siva\\Downloads\\compressedgeckodriver-V0.23.0-
win64\\geckodriver.exe");

3. for complex ones, separate classes are used:

e.g.- dropdown, alerts/pop-ups, frames-
frames - we have Select class to deal with
dropdown.

webElement Interface



we can use same method names for web
elements with different browsers.

this line is written driver.manage().window().maximize();
to maximize the output window.
public static void main(String[] args) {
 WebDriver driver = new FirefoxDriver();
 driver.get("http://test1.absolutesqa.com/");

download → Mozilla Gecko driver → ...

1111701021814089790

// driver.findElement(By.id("xyz"));
or

System.out.println("Login button displayed:" +
driver.findElement(By.name("login")).
isDisplayed());

System.out.println("Login button enabled:" +
driver.findElement(By.name("login")).
isEnabled());

System.out.println("Login button text:" +
driver.findElement(By.name("login")).
getAttribute("value"));

System.out.println("Login button width:" +
driver.findElement(By.name("login")).
getSize().width);

System.out.println("Login button height:" +
driver.findElement(By.name("login")).
getSize().height);

driver.findElement(By.name("login")).click();
} (Checking whether a element exist or not
private static boolean isElementPresent(By by) {
try {
driver.findElement(by);
return true;
} catch (NoSuchElementException e) {
return false;

Scanned by CamScanner

drill
comple
the
escape
display

standard form (by name ("logon"));
System.out.println("By-name Standard Form (Logon)");
3"

Ques - 5 login button displayed true
login button enabled : true
login button font : login
login button width : 105
login button height : 40
true
false

Handling different input types:-

Create one class by same package.

1. package com.abc.hello;

import java.awt.*;

import java.awt.event.*;

public class Main

public static void main (String [] args)

Abdullah obj = new Abdullah();

obj.manageWindow();

rdm.get("http://www.abdullah.com/hello.html");

Handling buttons we have
looked in earlier. → on-login.

```
System.out.println ("Default text: " +  
driver.findElement(By.id("username")).getAttribute("value"));  
driver.findElement(By.id("username")).sendKeys(@  
("baksbu"));  
System.out.println ("value after typing: " + driver.  
findElement(By.id("username")).getAttribute("value"));  
driver.findElement(By.id("username")).clear();  
System.out.println ("value after clear : " + driver.  
findElement(By.id("username")).getAttribute("value");  
})
```

Handling checkboxes:-

typical actions are

1. getting default state like whether it is checked or unchecked.
2. checking the checkbox
3. unchecking the checkbox.
4. getting the current status.
5. getting a attribute. → we get attribute method

Get to elipsis.

Create new class for same package.

Class name Checkboxes.

Main method ✓

1-8 lines are same as previous code with little bit changes like changing class name.

9. `System.out.println ("Default state: " + driver.findElement(By.id ("rememberme")).isSelected());`
to check or uncheck we can use if statement but it is not recommended to use because it is hardcoded. In webpages we can see many checkboxes and hence using hard code is not recommended.

for this we can use checkbox helper class.
↳ Below is having equal one. we can write code for this :-

```
package com.test1.tests;  
import org.openqa.selenium.By;  
public class Checkbox {  
    private WebElement checkbox;  
    public Checkbox(WebElement checkbox)  
    {  
        this.checkbox = checkbox;  
    }  
    public WebElement getCheckbox()  
    {  
        return checkbox;  
    }  
}
```

```
public void check()
{
    if (!(-checkbox.isSelected()))
    {
        -checkbox.click();
    }
}
```

```
public void uncheck()
{
    if (-checkbox.isSelected())
    {
        -checkbox.click();
    }
}
```

Now go back to checkboxes class.

```
10. Checkbox cb = new Checkbox("dnv");
    findElement(By.id("rememberme"));
    cb.click();
    System.out.println("State after click :" + dnv);
    findElement(By.id("rememberme"));
    isSelected();
    Thread.sleep(1000);
    cb.uncheck();
```

```
System.out.println ("State after uncheck :" +  
driver.findElement(By.id("rememberme")).  
isSelected());
```

}

}

Output - default state : false

State after check : true

State after uncheck : false.

Handling radio buttons:-

Create new class with RadioButton as
class name

main method ✓

```
WebDriver driver = new FirefoxDriver();
```

```
driver.manage().window().maximize();
```

```
driver.get ("http://test.automationtesting.com/product/
```

apple-phone");

```
driver.findElement(By.id("addToCart")).click();
```

```
driver.findElement(By.linkText("Viewcart")).  
click();
```

```
driver.findElement(By.name("proceed")).click();
```

```
System.out.println ("Direct Bank Transfer :" + driver  
.findElement(By.id("payment-method-bci")).  
.isSelected());
```

```
S.O.PIn ("cheque payment:" + driver.findElement  
        (By.id ("payment-method-cheque"))  
        .isDisplayed () && driver.findElement (By.id  
        ("payment-method-cheque")).isSelected());  
  
S.O.PIn ("paypal :" + driver.findElement (By.id  
        ("payment-method-paypal")).isDisplayed());
```

Thread.sleep (5000);
if (driver.findElement (By.id ("payment method paypal"))
 .isDisplayed ()) {
 driver.click ();
}

S.O.PIn ("----- After click -----");

S.O.PIn ("Direct Bank Transfer:" +

frame

same

O/P Direct Bank transfer: true.

cheque payment: false

paypal: false

----- After click -----

Direct Bank transfer: false

cheque payment: false

paypal: true.

Handling Links

```
package com.test1.tests;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class links {
    public static void main (String [] args) {
        WebDriver driver = new FirefoxDriver();
        driver.manage().window().maximize();
        driver.get ("http://test1.automationtrainings.com/shop");
        System.out.println (driver.findElement(By.partialLinkText ("Samsung phone")).getAttribute ("href"));
        System.out.println (driver.findElement(By.partialLinkText ("Samsung phone")).getText ());
        driver.findElement(By.partialLinkText ("Samsung phone")).click ();
    }
}
```

OIP:- Link URL :- http://test1.automationtrainings.com / product / Samsung - phone
link Text : Samsung phone
£ 400.00

Extracting WebElements :-

Create class ExtractingWebElements class with main method.

```
public static void main() {  
    WebDriver driver = new FirefoxDriver();  
    driver.manage().window().maximize();  
    driver.get("http://test1.automationtrivya.com/product/  
    Samsung-phone/");  
}
```

WebElement cart = driver.findElement(By.id("cart"));

cart.findElement(By.name("quantity")).sendKeys("3");

```
driver.findElement(By.xpath("//form[@id='cart']  
    /div/input[2]")).sendKeys("8");
```

}

}

Code for dropdown lists is in phone - 3 pic.

column side
gadget \downarrow $\text{">//nav[@id='top-menu-nav']//ul//a}$

This is the xpath which represents
any link in the navigation menu.

go to eclipse write code.

Create new class.

```
package com.test1.tests;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.By;  
public class MultipleWebElements {  
    public static void main (String [] args) {  
        WebDriver driver = new FirefoxDriver();  
        driver.manage().window().maximize();  
        driver.get ("http://test1.absopftraining.com/shop/");  
        driver.findElements (By.xpath ("//nav[@id='top-  
        menu-nav']//ul//a"));  
        System.out.println (menu.getText () + ":"  
        + menu.getAttribute ("href"));  
    }  
}
```

create for each loop for menu list to use
all navigation links.

```
for (WebElement menu : menus)
```

```
System.out.println (menu.getText () + ":"  
+ menu.getAttribute ("href"));
```

}

Handling Static and Dynamic Web tables.

Generally xpath locating strategies are used for table rows or table cells

go to selenium IDE and in target field write
xpath //table[@id='testaccounts']/tbody/tr,

It will identify the first row

if we give number after tr

like tr[2] → it will identify 2nd row in the table

If we write td after tr like tr[3]/td,
it will identify 3rd row first cell.

It is always recommended to use helper class for webtables to do all require automations

helper class is

```
package com.testl.tests;
```

```
import java.util.List;
```

```
public class WebTable {
```

```
private WebElement webTable;
```

```
public WebTable(WebElement webTable)
    // take a web element parameter representing the table
    - webTable = webTable; } and assign that to - webTable.
```

```
public WebElement getWebTable()
```

```
{
```

```
return webTable;
```

```
}
```

```
public int getCount()
```

```
{
```

```
return webTable.findElements(By.xpath("//tbody/
```

```
tr")).size();
```

```
{
```

Counted ----- in website.

return count of all web elements by tag name. tr (tr-represents table row)

webTable Actions -> code is in phone.

Here we are using all the actions of web table class.

Dim...