

HANDLING MULTIPLE ELEMENTS

▷ In order to get multiple elements from the webpage or in order to perform operation on multiple elements, we use find elements method of webdriver.

2) In order to identify multiple elements, follow below mentioned process

i) See the html hierarchy of first element.

ii) See the html hierarchy for second element.

Like that check the html hierarchy of any random element if you found all are following the same hierarchy then see the common parent.

ii) Identify their common parent uniquely, then define the hierarchy in which all the required elements are present.

e.g. In google search if we need to print all the suggestions given by google then as per above steps, all the options are having below mentioned html hierarchy,

li ↳ div ↳ div ↳ span

iii) After checking the common html hierarchy, we need to reachout the common parent which has all the options -

ul (class = "crkuge")

↳ li ↳ div ↳ div ↳ span

Find elements method returns list of web elements.

List <webElement> options = driver.findElements(
By.cssSelector("ul[class = "crkuge"] li > div > div > span"))

After getting all the elements, we can get the count of identified element by using "size()".

e.g. options.size() → (size of list)

NO

DATE

In order to retrieve individual element from above list <webElement>. We use get method by specifying index in the loop.

```
for (int i=0; i<options.size(); i++)
{
    Thread.sleep(2000);
    System.out.println(options.get(i).getText());
}
give you a help you to retrieve the
xweb element text from web element
```

Write a script to open any browser & open google in the search input field type selenium, get the all suggestions of count & name.

```
→
public class HandlingMultipleElements
{
    public static void main (String [] args)
    {
        System.setProperty("webdriver.chrome.driver",
                           "Path");
        WebDriver d = new ChromeDriver();
    }
}
```

NO

DATE

```
/*+ Maximise a browser window */
driver.manage().window().maximize();

/* Implicit wait */
driver.manage().timeouts().implicitlyWait(20,
                                          TimeUnit.SECONDS);

/* Step 2 */
driver.get("https://google.com");

/* Identify search input Field & type selenium */
driver.findElement(By.name("q")).sendKeys
("Selenium");

/* Identify all the suggestion given by google */
List<WebElement> options = driver.findElement(
    By.cssSelector("ul[class='erkre']>li>div>div>
    span"));

/* get the op count of options */
System.out.println("Options count:" + options.size());

/* print all the options name */
for (int i=0; i<options.size(); i++)
{
    Thread.sleep(2000);
    System.out.println(options.get(i).getText());
}
```

NO _____
DATE _____

Q. Write a selenium script to get Samsung device count & their name from GSMArena.

→ class Test

```
p.s.v.m (String [] args)  
{
```

```
    system.setProperty ("webdriver.chrome.driver", Path);  
    webdriver driver = new ChromeDriver ();  
    /* Maximize window site  
    driver.manage().window().maximize();
```

/* Implicit wait //

```
driver.manage().timeouts().implicitlyWait(  
    30, TimeUnit.seconds());
```

/* get URL */

```
driver.get ("https://www.gsmarena.com/  
Samsung=phone"));
```

/* Identify device name */

```
List<webElement> deviceName = driver.findElements  
(By.cssSelector("div [class = "makers"]>ul>li>  
a>string>span"));
```

/* get the count */

```
s.o.p ("Count": "+ deviceName.size());
```

NO _____
DATE _____

```
for (int i=0 ; i<deviceName.size(); i++)  
{  
    s.o.p (deviceName.get(i).getText());  
}
```

Q. Write a script to print all the phone names from phone finder section of gsmarena application.

Q. Write a script to perform following actions-

- Open any browser & enter the url of gsmarena app & verify the landing page.
- select Samsung phone from phone finder section & validate the page is open or not.

- Verify whether this page has page initiation or not. If yes, then print all the page initiation link. ^{Size > 0 or not.}

HANDLING DROPODOWN

In order to handle an html dropdown which is developed using select tag of html e.g.

```
<select id='make' name='make'>
<option value="default">please select</option>
<option value="Auct">Audi </option>
    "BMW" > BMW </option>
</select>
```

If the dropdown is developed using select tag of html then we dont use sendkeys() or click(). Instead of this, we follow below mentioned process.

1) Identify the dropdown list as mentioned below,

```
webElement drop=driver.findElement(By.id("make"))
```

2) Create an instance of select class by passing above webElement as an argument to select class constructor -

```
Select $1 = new Select(drop);
```

3) In order to get default selected option or selected option from a dropdown, we will use get first selected option method of select class -

```
$1.getFirstSelectedOption().getText();
```

4) As get first selected option method returns web-elements hence in order to retrieve value, we use get text() method.

5) In order to select a value from dropdown, we can use any of the below mentioned method of select class.

\$1.selectByIndex(2); // select BMW
or

\$1.selectByValue ("BMW");
or

\$1.selectByVisibleText ("BMW");

In order to retrieve all the options of dropdown we will use getoptions() method of select class which will return List<webElement>.

In order to get count of options, we use size()

```
List<WebElement> op = s1.getOptions();  
S.O.P (op.size());
```

In order to ~~select~~^{retrieve} individual element from above list, we can use get() method by specifying index in a loop.

```
for (int i=0; i< op.size(); i++)  
{  
    S.O.P (op.get(i).getText());  
}
```

Q. Write a selenium script to perform following operations -

- 1) Open any browser & enter following application URL
<http://sampleapp.tricentis.com>
- 2) Click on car automobile link
- 3) Validate all the dropdowns
- 4) Default text, option count, print all the options & select any option.

FindElement(By) → Identify single element from the web page
→ returns web Element
→ Exception: No such element exception.

FindElements(By) → Identify multiple elements from the webpage
→ returns: List< web Element >
→ index
→ duplicate
→ Null
→ No exception will be thrown as list can be empty

...add(S1.getOptions()).get(i).getText()) }
List< webElements
get individual Element
retrieve text

NO

DATE

NO

DATE

Q. How you will verify drop down options are sorted in ascending order?

→ /** identify the dropdown list */

```
WebElement drop = driver.findElement(By.id("make"));
```

/** Create an instance of select class & pass the above element to its constructor */
 Select s1 = new Select(drop);

/** create a list to store the actual values */

```
List<String> actualList = new ArrayList<String>();
```

/* get the options from dropdown & store in list */

```
for (int i=1; i < s1.getOptions().size(); i++)  

{
```

```
actualList.add(s1.getOptions().get(i).getText());
```

}

s.o.p ("Actual list is: " + actualList);

/** create a duplicate list from actual list */

```
List<String> dupList = new ArrayList<String>();
```

```
dupList.addAll(actualList);
```

/** Sort the duplicate list in ascending order */

```
Collection.sort(dupList);
```

s.o.p ("After sorting dupList: " + dupList);

/** compare both the lists */

```
s.o.p (dupList.equals(actualList));
```

}

HANDLING MULTISELECT DROPDOWN -

Multiselect dropdown is also handled in the same way as single select dropdown is getting handled.

Step 1: Identify the dropdown

Step 2: Create an instance of select class by passing above dropdown WebElement to select class constructor.

Step 3: For selecting value from dropdown, we can use any of the below mentioned methods -

select by visible text, select by

1) SelectByVisibleText (String)

2) SelectByValue (String)

3) SelectByIndex (int)

We dont have any method to select all the values from dropdown. Instead of that we can use loops.

NO

DATE

In multiselect dropdown, we have some additional methods to deselect already selected options.

- 1) deselectByVisibleText (String)
- 2) deselectByValue (String)
- 3) deselectByIndex (int)
- 4) deselectAll ()

For multiselect dropdown, in order to retrieve only selected options, we should use "get [↑]All Selected Options ()" of select class.

This method returns list of web elements hence in order to get the count of selected options, we can use size method, in order to get individual elements, we use get method by specifying index in a loop.

NO

DATE

WORKING WITH PROPERTY FILE

- There are various ways available to read the text data from text file, excel file, property file, xml file, database.
- Property file is one of the easiest concept to deal with text data.
- Property file always has an extension ".properties"
- In property file, data is stored in form of key value pair.
- Steps to create property file -
 - 1) In eclipse → go to project & right click on required location, go to new & select file, specify the file name with ".properties" extension & click on OK button.
 - For e.g. property file contains
 1. username = admin
 2. pwd = manager
 3. title = Demo
- 2) Steps to read data from property file →
 - 1) Create an instance of file inputStream class & specify location of properties file in its constructor.
 - 2) Create an object of properties class.
 - 3) Call the load method of properties class by specifying file inputStream