

Locators:

- Required to fetch the address of any web element
- Locators are static methods belongs to the By class. This By is an abstract class.

Types of Locators:

- Id Locator
- name locator
- tagname Locator
- classNameLocator id locator
- linkText locator
- partialLinkText
- css selector Locator
- X path Locators -----5 syntaxes

id Locators: This id() method Locator is used to find the address of the web element using attribute name called **id** in html tree structure.

```
<input type="text" class="inputtext _55r1 _6luy"
      name="email" id="email" data-
      testid="royal_email" placeholder="Email address
      or phone number" autofocus="1" aria-label="Email
      address or phone number"
      fdprocessedid="i2bgk">
```

- `driver.findElement(By.id("email"));`

name Locator: This name() method Locator is used to find the address of the web element using attribute name called `name` in html tree structure.

```
<input type="text" class="inputtext _55r1_6luy"
name="email" id="email" data-
testid="royal_email" placeholder="Email address
or phone number" autofocus="1" aria-label="Email
address or phone number"
fdprocessedid="i2bgk">
```

- `driver.findElement(By.name("email"));`

linkText Locator:

- with the help of this linkText() method locator we can only fetch the address of web element called link.
- Links are always developed with a tag name called `a`
- Here we pick the text value to fetch the address.

```
<a href="https://www.facebook.com/recover/initiate/?privacy_mutation_token=eyJ0eXBIIjowLCJjcmVhdGlvbl90aW1IjoxNjg4OTc4NjA1LCJjYWxsc2l0ZV9pZCI6MzgxMjI5MDc5NTc1OTQ2fQ%3D%3D&ars=facebook_login" waprocessedanchor="true">Forgotten password?</a>
```

- driver.findElement(By.linkText("Forgotten password?"));

partialLinkText Locator:

- with the help of this partialLinkText() method locator we can only fetch the address of web element called link.
- Links are always developed with a tag name called **a**
- Whenever the text is very big and contains more spaces, then we will use this method
- Here we pick the partial text value to fetch the address of a web element.
- When ever text contains more spaces or large text then we will use this locator.

```
<a href="https://www.facebook.com/recover/initiate/?privacy_mutation_token=eyJ0eXBIIjowLCJjcmVhdGlvbl90aW1IIjoxNjg4OTc4NjA1LCJjYWxsc2I0ZV9pZCI6MzgxMjI5MDc5NTc1OTQ2fQ%3D%3D&ars=facebook_login" waprocessedanchor="true">Forgotten password?</a>
```

- driver.findElement(By.partialLinkText("password"));

tagName locator: Using this tagName() method locator, we can fetch the list of specified tag names that web elements contains.

```
driver.findElements(By.tagName("a"));
```

Script: Write a script to fetch all the links from the facebook login page.

```
package Locators;  
  
import java.util.List;  
  
import org.openqa.selenium.By;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.WebElement;
```

```

import org.openqa.selenium.chrome.ChromeDriver;
import io.github.bonigarcia.wdm.WebDriverManager;

public class TagNameLocator {

    public static void main(String[] args) throws Throwable {
        WebDriverManager.chromedriver().setup();
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://www.facebook.com/");
        Thread.sleep(5000);
        List<WebElement> links =
driver.findElements(By.tagName("a"));

        for (WebElement b : links) {
            System.out.println(b.getText());
        }

        driver.close();
    }
}

```

className Locator: using this className() method locator we can fetch the address of the web element with the help of attribute name called class.

```

<form class="_9vtf" data-testid="royal_login_form"
action="/login/?privacy_mutation_token=eyJ0eXBIIjow
LCJcmVhdGlvbl90aW1IIjoxNjg4OTc4NjA1LCJjYWxsc2I0
ZV9pZCI6MzgxMjI5MDc5NTc1OTQ2fQ%3D%3D"
method="post" onsubmit="" id="u_0_2_gT"><input
type="hidden" name="jazoest" value="2895"
autocomplete="off"><input type="hidden" name="lsd"
value="AVoRQIQ9IaY" autocomplete="off"><div><div>

```

- ```
class="_6lux">><input type="text" class="inputtext _55r1 _6luy"
```
- driver.findElement(By.className("inputtext \_55r1 \_6luy"));

```
package Locators;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

import io.github.bonigarcia.wdm.WebDriverManager;

public class ClassNameLocator {

 public static void main(String[] args) throws
InterruptedException {
 WebDriverManager.chromedriver().setup();
 WebDriver driver = new ChromeDriver();
 driver.manage().window().maximize();
 driver.get("https://www.facebook.com/");
 Thread.sleep(5000);
 WebElement value =
driver.findElement(By.className("_6luy _55r1 _1kbt")));
 value.click();

 }

}
```

## CSS (Cascading style sheet) Locator:

- With this locator we can fetch the address of an web element using Attributes.

- Drawback of this it will not support text
- It has a syntax

**tagname[Attribute name='Attribute value']**

**Below is the html code:**

```
<input type="text" class="inputtext _55r1 _6luy"
name="email" id="email" data-testid="royal_email"
placeholder="Email address or phone number"
autofocus="1" aria-label="Email address or phone
number" fdprocessedid="ajgo2">
```

**How to put this code in Syntax format:**

**Input[type='text']**

**driver.findElement(By.cssSelector("Input[type='text']"))**

**NOTE:** count should be 1 of 1

**X-path:** It is used to find unknown paths in html tree structure.

2 types of x path locators

1. Absolute X path locator
2. Relative X path Locator

## **Absolute X path Locator:**

- Here we have to write complete path beginning from the root element to the element which we wanted to identify.
- The Expression will becomes very big which is the draw back of this locator
- To overcome this draw back we will go for Relative x path.
- Here we use / (single forward slash) to traverse from one element to another.
- Eg:

```
<!DOCTYPE html>

<html>
<head>
<title>Sample Page</title>
</head>
<body>
<div class="container">
<h1>Sample Heading</h1>

Item 1
```

```
Item 2
Item 3

<form>
 <input type="text" id="username"
name="username" />
 <input type="password" id="password"
name="password" />
 <button type="submit">Login</button>
</form>
</div>
</body>
</html>
```

**Absolute XPath:** /html/body/div/form/button

**Draw back:** In real time application, we will not be using this because the length of expression is too big.

**Relative X path Locators:**

---

- It will search anywhere on the web page.
- We use // (double forwarding slash) to traverse from parent to any child element.

We have 5 types of Relative X path Locators:

- 1. X path by Attribute**
- 2. X path by Text**
- 3. X path by contains**
- 4. X path by traversing**
- 5. X path by group index**

## **1. X path by Attribute:**

---

**Syntax:**

**//tagname[@Attribute name='Attribute value']**

```
<input type="text" class="inputtext _55r1 _6luy"
name="email" id="email" data-testid="royal_email"
placeholder="Email address or phone number"
autofocus="1" aria-label="Email address or phone
number" fdprocessedid="r13i4">
```

**How to write this in script:**

```
driver.findElement(By.xpath("//input[@id='email']"));
```

## Drawbacks:

- Attributes are mandatory.
- It does not support text.

## 2. X path by Text:

---

### Syntax:

**//tagname[text()='text value']**

### Drawbacks:

- It will support only text.
- If the text is too big and contains more spaces difficult to handle.

```
<a
href="https://www.facebook.com/recover/initiate/?p
rivacy_mutation_token=eyJ0eXBIIjowLCJjcmVhdGlvbl
90aW1IjoxNjg5MTUxODcyLCJjYWxsc2l0ZV9pZCI6Mzg
xMjI5MDc5NTc1OTQ2fQ%3D%3D&ars=facebook
```

[\\_login" waprocessedanchor="true">Forgotten password?</a>](#)

**How to write this in script:**

```
driver.findElement(By.xpath("//input[text() = 'Forgotten password?']));
```

### **3. X path by contains:**

---

**It has two syntaxs**

**1. X path contains with respect to Attributes:**

**//tagname[contains(@Attributename,'Attribute value')]**

**2. X path contains with respect to text:**

**//tagname[contains(text(),'text value')]**

```
Forgotten password?
```

**How to write this in script:**

```
driver.findElement(By.xpath("//input[contains(@wapro
cessedanchor,'true')]"));
```

```
driver.findElement(By.xpath("//input[contains(text(),Fo
rgotten')]"));
```

**Advantages:**

- Easy to handle lengthy text and also text that contains spaces.
- Supports both text and attributes.

**NOTE:** count should be 1 of 1

**X path syntaxes:**

**Xpath Attribute:** //tagname[@AN='AV']

**Xpath text:** //tagname[text()='text value']

**X path contains Attribute:**

//tagname[contains(@AN,'AV')]

**Xpath contains text:**

//tagname[contains(text(),'text value')]

### **3. X path by traversing:**

---

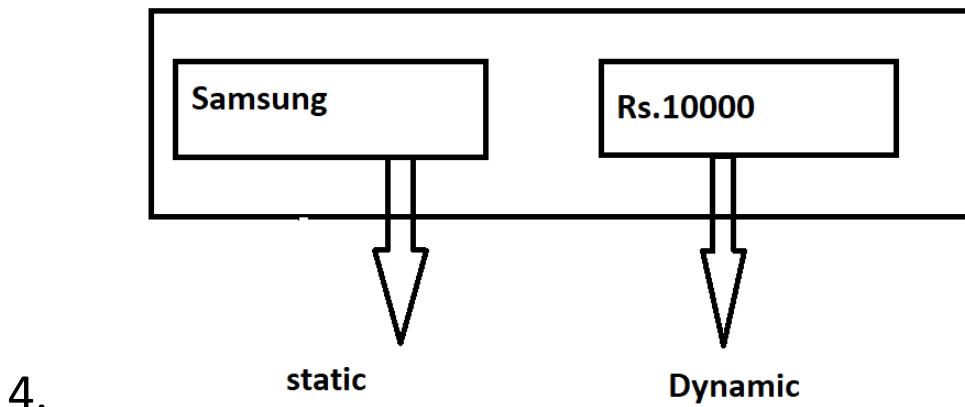
- It is used to handle dynamically changing elements.
- We don't have syntax.

#### **Steps to be followed:**

1. Identify the static element and write the x path expression.
2. Identify the common parent (with the help of /..)
3. Write the tag name or xpath for dynamically changing elements.

//span[contains(text(),'Samsung Galaxy M34 5G (Prism Silver, 8GB, 128GB Storage')]/../../../../../span[@class='a-price-whole']

Eg: //span[text()='itel A60s']/../../../../span[@class='a-price-whole']



**Static:** The element which is fixed and does not change.

**Dynamic:** The element which changes frequently,

```
//span[contains(text(),'Samsung Galaxy M34 5G
(Prism Silver, 8GB, 128GB
Storage')]/../../../../span[@class='a-price-whole']
```

**Script:**

```
package Locators;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

import io.github.bonigarcia.wdm.WebDriverManager;

public class xpath_Traversing {
```

```

public static void main(String[] args) {
 WebDriverManager.chromedriver().setup();
 WebDriver driver = new ChromeDriver();
 driver.manage().window().maximize();

 driver.get("https://www.amazon.in/");
 driver.findElement(By.xpath("//input[@id='twotabsearchtextbox']")).sendKeys("samsung phone");
 driver.findElement(By.xpath("//input[@id='nav-search-submit-button']")).click();
 WebElement value =
 driver.findElement(By.xpath("//span[contains(text(),'Samsung Galaxy M13 (Aqua Green, 4GB, 64GB Storage)])../../../../span[@class='a-price-whole']]"));
 System.out.println(value.getText());
}

}

```

## **NOTE:**

**/ --->used for Traversing from parent to immediate child**

**//---> used for Traversing from parent to any child**

**/.. --> used for Traversing from child to parent.**

---

## **5.X path by group Index:**

When ever multiple elements are matching we use group index to fetch the address of an element.

**Syntax: (X path expression)[position value]**

[1]

**Script:open the browser and enter google url type your name and fetch the 4<sup>th</sup> value into console from the suggestion**

**Note:**

- index always start from 0.
- Position value always starts from 1.

5	4	2	6	8	11	1
1	2	3	4	5	6	7

Position value of 6 is 4.

Eg: **(//input[@type='radio'])[4]**