Selenium WebDriver

Selenium automates browsers. That’s it. Primarily, it is for automating web applications for testing purposes

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

- Initializing WebDriver. Each browser has own implementation of WebDriver interface (different browser reads and represents web applications differently) so they need different approaches to perform automation
- It drives the browser
- Good example of Polymorphism

```
driver.get("website url");
```

- Navigates to given url
- **driver.get("url")** has build in wait. It will wait until web page fully loaded
- **driver.navigate().to("url")** does same thing as **driver.get("url")**

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

- This method returns WebElement object which will represent specific html element in the web page
- Using WebElement object you can perform some actions on this element in the web page (click, read text, send keys)

html element – building blocks in the webpage. It usually consists of a **start** tag and an **end** tag, with the content inserted in between:

HTML Element:

```
<div class=“b-1”>Meaningful content here..</div>
```

How many html elements in this page?

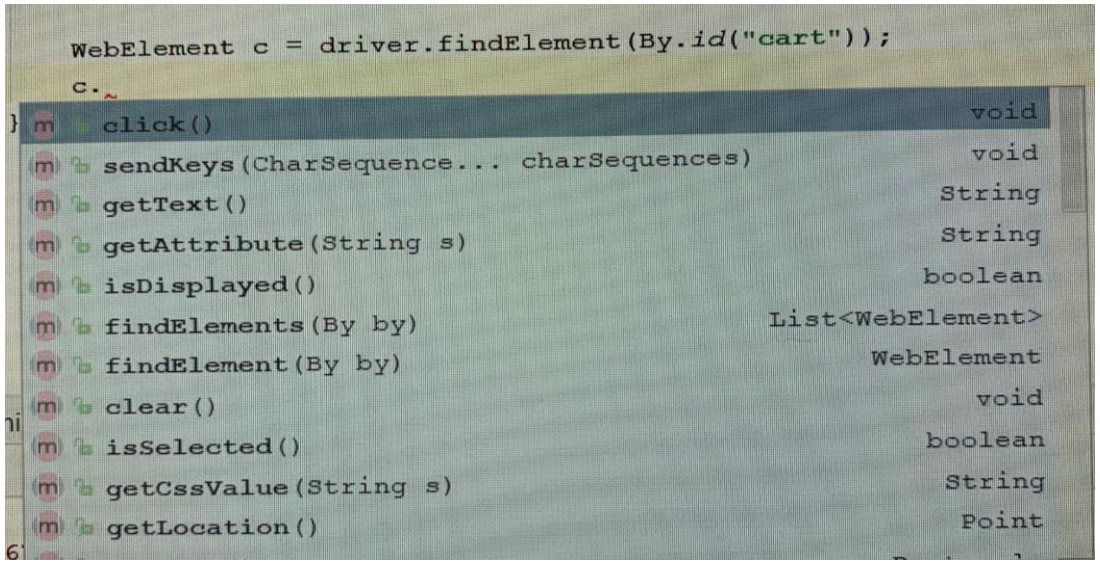
```
<!DOCTYPE html>
<html>
  <body>
    <h1>My First Heading</h1>
    <p>My first paragraph.</p>
  </body>
</html>
```

WebElement (Selenium)

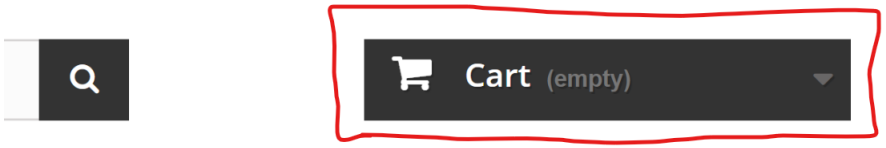
- Represents an HTML element. Generally, all interesting operations to do with interacting with a page will be performed through this interface.
- https://selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebElement.html

Selenium:

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



Browser:




```
<a id="cart" href="">
  <b class="cart_q">Cart</b>
  <span class="c_p">Product</span>
  <span>(empty)</span>
</a>
```


WebElement
Locator
html Element

Finding / Locating WebElements

- **By** class is used to find elements
- Main locators:

By.id(“id”)	Id should be unique in the page so it’s the best option to use
By.name(“name”)	If element has unique name attribute
By.className(“class”)	If element has unique class attribute
By.cssSelector(“css-l”)	Write your own ccs locator
By.xpath(“xpath”)	Write your own xpath locator

Example 1: find/locate '**Home**' Link

```
<div class="container">
    <div class="header">
      <div>
        <a href="" id="home-link" class="link-prime">Home</a>
      </div>
      <div>
        <a href="" name="contact-link" class="link-prime">Contact Us</a>
      </div>
    </div>
</div>
```

```
WebElement homeLink = driver.findElement(By.id("home-link"));
```

Example 2: find/locate '**Contact Us**' link

```
<div class="container">
    <div class="header">
      <div>
        <a href="" id="home-link" class="link-prime">Home</a>
      </div>
      <div>
        <a href="" name="contact-link" class="link-prime">Contact Us</a>
      </div>
    </div>
</div>
```

```
WebElement contLink = driver.findElement(By.name("contact"));
```

Example 3: find/locate **‘Python’** heading

```
<div class="container">
    <div class="main">
      <div>
        <h1 title="java-prog">Java</h1>
      </div>
      <div>
        <h1 title="python-prog">Python</h1>
      </div>
      <div>
        <h1 title="js-prog">JS</h1>
      </div>
    </div>
</div>
```

ccsSelector(“css”)

tag[attribute=‘value’]

ccs locator for Example 3:

```
h1[title='python-prog']
```

Example 4: find/locate **'Java'** heading

```
<div class="container">
    <div class="main">
      <div title="java-prog">
        <h1>Java</h1>
      </div>
      <div title="python-prog">
        <h1>Python</h1>
      </div>
      <div title="js-prog">
        <h1>JS</h1>
      </div>
    </div>
</div>
```

ccs locator for **Submit** button

```
<html>
  <body>
      <div>
        <button>Submit</button>
      </div>
  </body>
</html>

```

Html > body > div > button

> is used to find child element/elements

Example 4: find/locate **'Java'** heading

```
<div class="container">
    <div class="main">
      <div title="java-prog">
        <h1>Java</h1>
      </div>
      <div title="python-prog">
        <h1>Python</h1>
      </div>
      <div title="js-prog">
        <h1>JS</h1>
      </div>
    </div>
</div>
```

ccs locator for Example 4:

div[title=‘java-prog’] > h1

Example 6: find/locate **'Last Name'** input box

```
<div class="container">
    <div class="main">
     <form>
      <div id="first-name">
        <div>
          <span>First Name:</span>
          <input type="text">
        </div>
      </div>
      <div id="last-name">
        <div>
          <span>Last Name:</span>
          <input type="text">
        </div>
      </div>
      <div id="submit-btn">
        <div>
          <input type="submit" value="Save">
        </div>
      </div>
    </form>
    </div>
</div>
```

ccs locator for Example 6:

#last-name > div > input

Example 7: find/locate **‘Orange Page’** link

```
<div class="apple">
  <a href="">Apple Page</a>
</div>
<div class="banana">
  <a href="">Banana Page</a>
</div>
<div class="orange">
  <a href="">Orange Page</a>
</div>
<div class="kiwi">
  <a href="">Kiwi Page</a>
</div>
```

ccs locator for Example 7:

.orange > a


WebElement
Locator
html Element

WebElement Methods:

https://selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebElement.html

Method	Description
click()	Click this element
getText()	Get the visible value Hello
sendKeys(String str)	Use this method to simulate typing into element.
isDisplayed()	Is this element displayed or not?
isSelected()	Determine whether or not this element is selected (for check boxes)