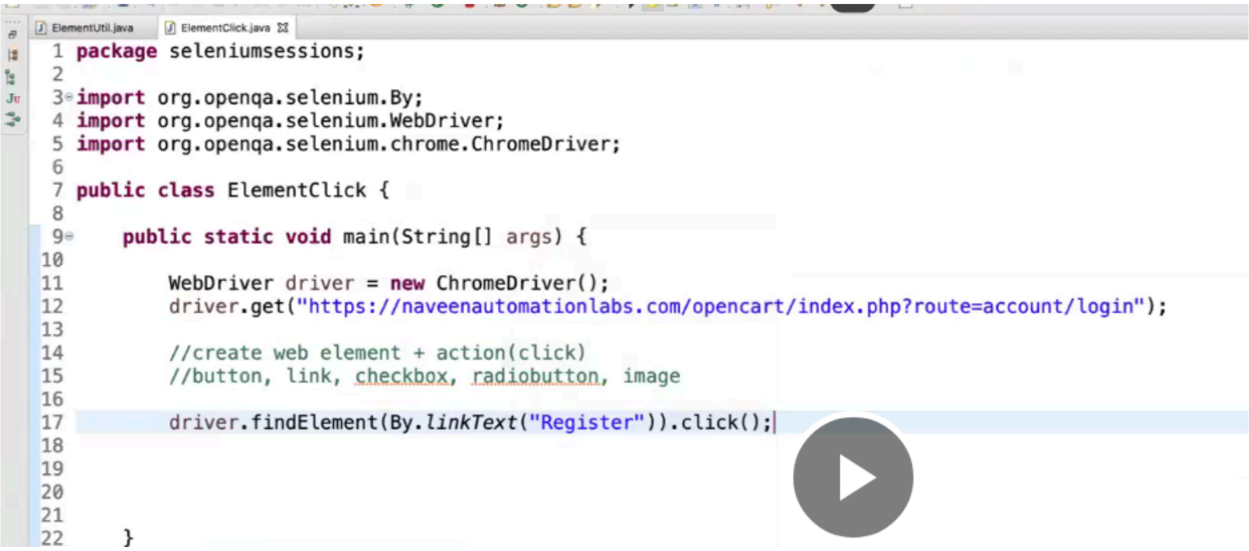
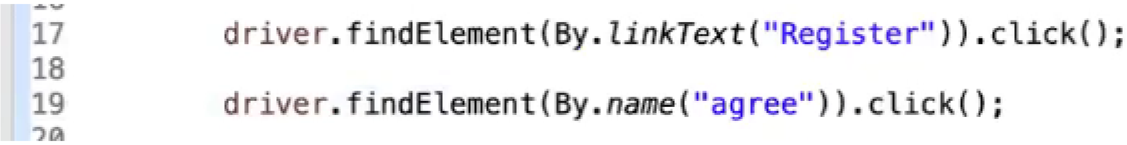
Simple click-

Click on register link



```
1  package seleniumsessions;
2
3  import org.openqa.selenium.By;
4  import org.openqa.selenium.WebDriver;
5  import org.openqa.selenium.chrome.ChromeDriver;
6
7  public class ElementClick {
8
9      public static void main(String[] args) {
10
11         WebDriver driver = new ChromeDriver();
12         driver.get("https://naveenautomationlabs.com/opencart/index.php?route=account/login");
13
14         //create web element + action(click)
15         //button, link, checkbox, radiobutton, image
16
17         driver.findElement(By.linkText("Register")).click();
18
19
20
21
22     }
```

Click on privacy checkbox

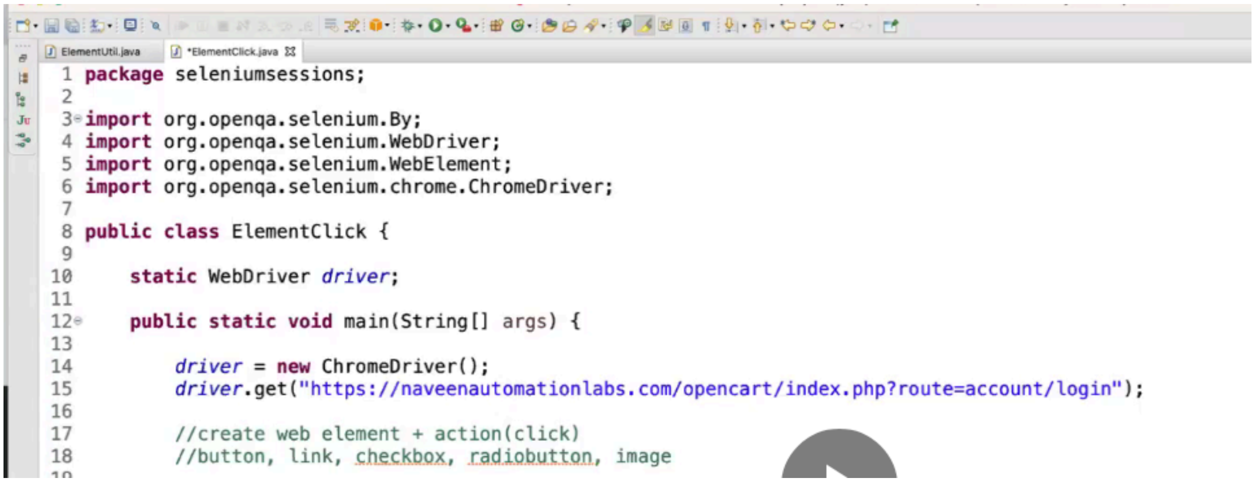


```
17         driver.findElement(By.linkText("Register")).click();
18
19         driver.findElement(By.name("agree")).click();
```

Lets make it without hard coding-

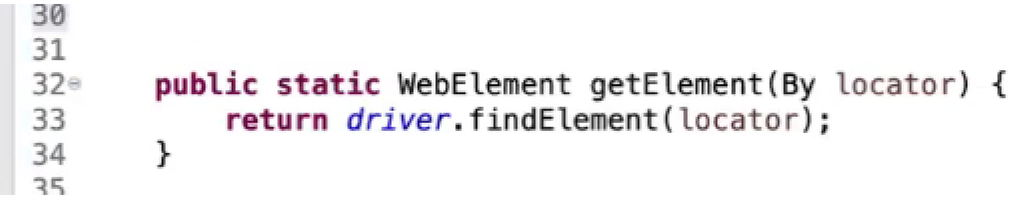
Changed webdriver to static.

Updated line 14.



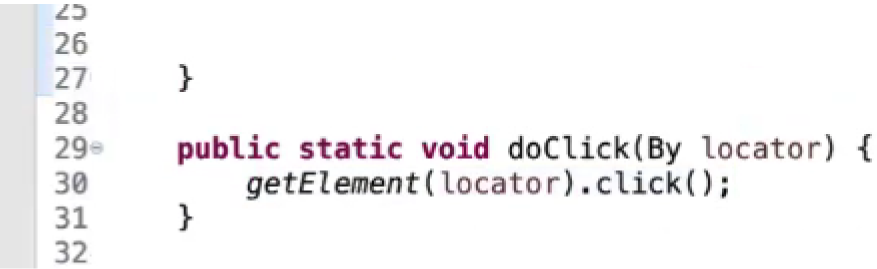
```java
package seleniumsessions;

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

public class ElementClick {

    static WebDriver driver;

    public static void main(String[] args) {

        driver = new ChromeDriver();
        driver.get("https://naveenautomationlabs.com/opencart/index.php?route=account/login");

        //create web element + action(click)
        //button, link, checkbox, radiobutton, image
```

Add the get element method



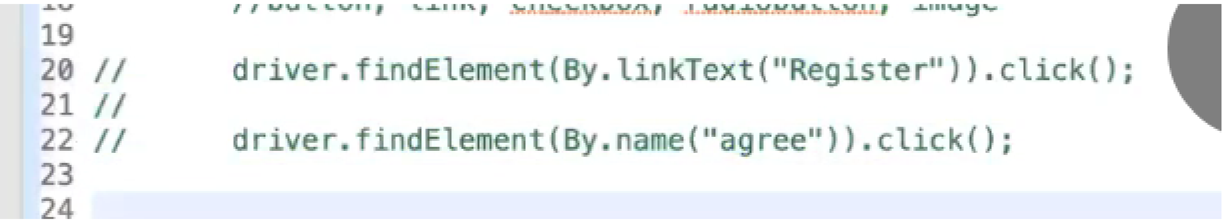
```java
    public static WebElement getElement(By locator) {
        return driver.findElement(locator);
    }
```

Add click method



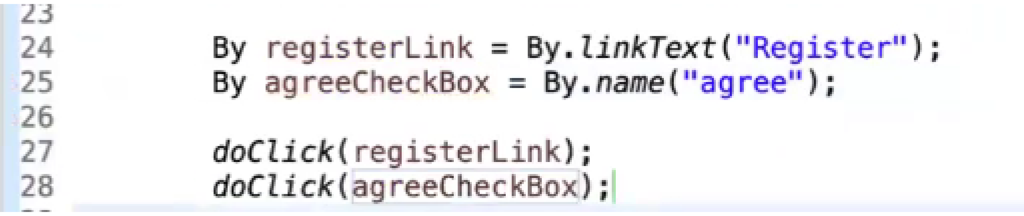
```java
    }

    public static void doClick(By locator) {
        getElement(locator).click();
    }
```

Comment out the hardcoded line



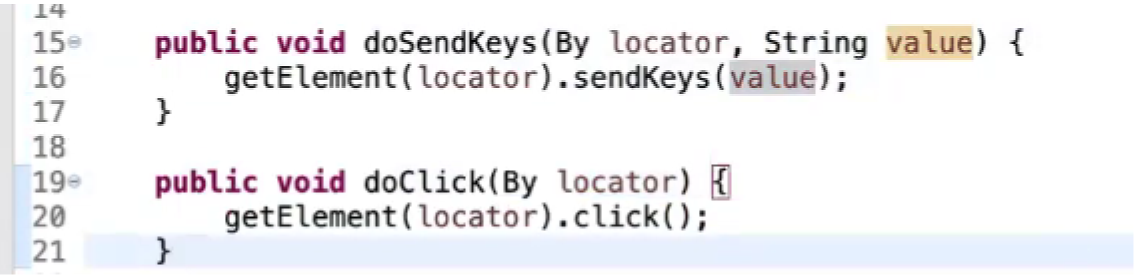
```java
        //button, link, checkbox, radiobutton, image

//      driver.findElement(By.linkText("Register")).click();
//
//      driver.findElement(By.name("agree")).click();

```

Call methods



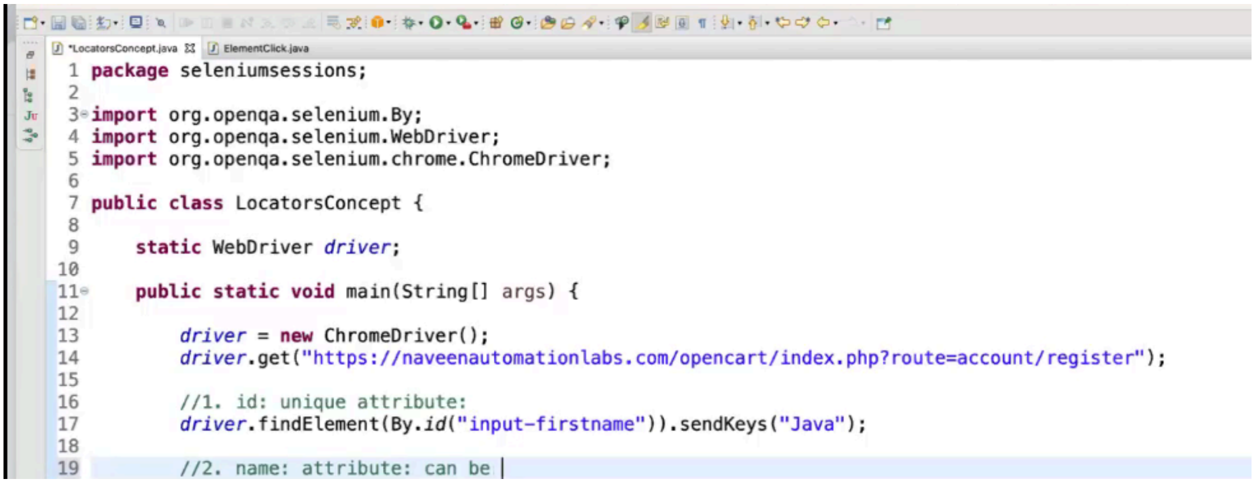
```
        By registerLink = By.linkText("Register");
        By agreeCheckBox = By.name("agree");

        doClick(registerLink);
        doClick(agreeCheckBox);
```

Add the click method in element util class



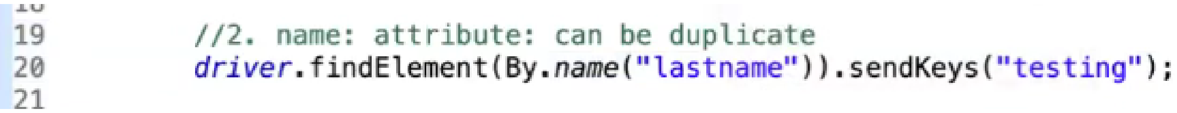
```
    public void doSendKeys(By locator, String value) {
        getElement(locator).sendKeys(value);
    }

    public void doClick(By locator) {
        getElement(locator).click();
    }
```

Different locators



```
package seleniumsessions;

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

public class LocatorsConcept {

    static WebDriver driver;

    public static void main(String[] args) {

        driver = new ChromeDriver();
        driver.get("https://naveenautomationlabs.com/opencart/index.php?route=account/register");

        //1. id: unique attribute:
        driver.findElement(By.id("input-firstname")).sendKeys("Java");

        //2. name: attribute: can be 
```

Name locator



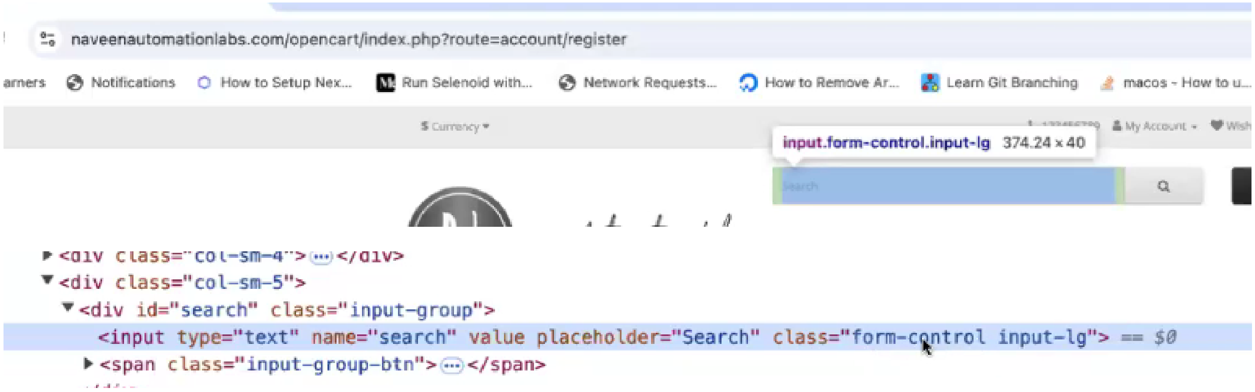
```
        //2. name: attribute: can be duplicate
        driver.findElement(By.name("lastname")).sendKeys("testing");
```

How many classes for search-

Two.

Separated by space.

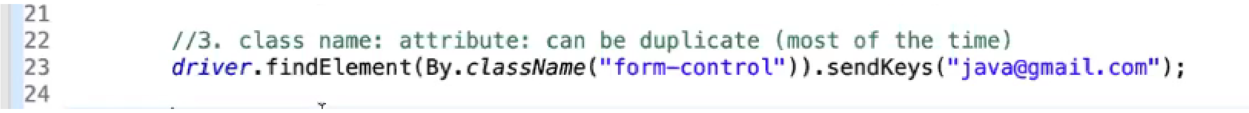
Class is mainly for css setup for the page.



```
▶<div class="col-sm-4">⋯</div>
▼<div class="col-sm-5">
  ▼<div id="search" class="input-group">
     <input type="text" name="search" value placeholder="Search" class="form-control input-lg"> == $0
    ▶<span class="input-group-btn">⋯</span>
```

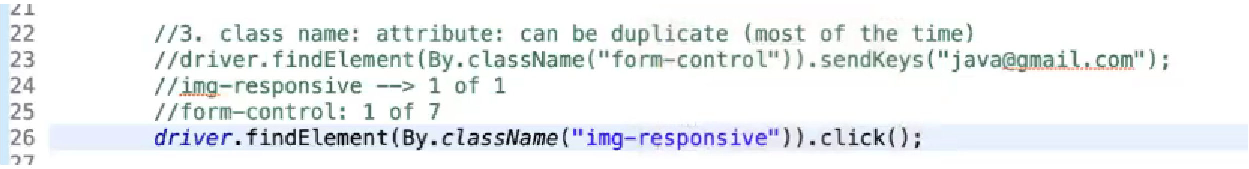
Classname-

Will start scanning from top left. If duplicates then data entered in first element. In our case in search.



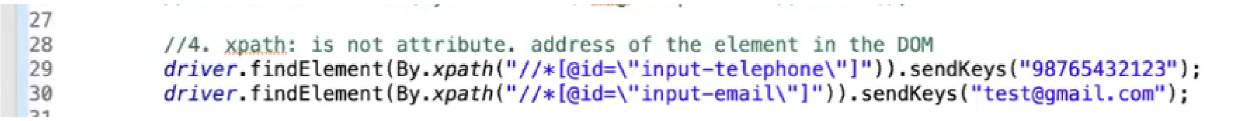
```
        //3. class name: attribute: can be duplicate (most of the time)
        driver.findElement(By.className("form-control")).sendKeys("java@gmail.com");
```

Click on image in the webpage-



```
        //3. class name: attribute: can be duplicate (most of the time)
        //driver.findElement(By.className("form-control")).sendKeys("java@gmail.com");
        //img-responsive --> 1 of 1
        //form-control: 1 of 7
        driver.findElement(By.className("img-responsive")).click();
```

Enter email and phone using xpath

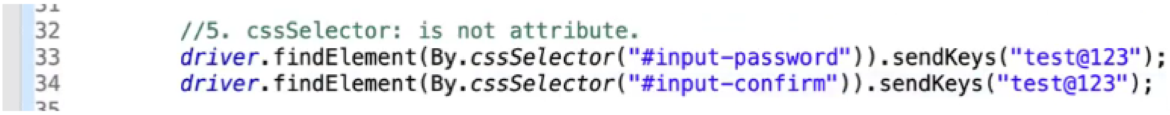


```
        //4. xpath: is not attribute. address of the element in the DOM
        driver.findElement(By.xpath("//*[@id=\"input-telephone\"]")).sendKeys("98765432123");
        driver.findElement(By.xpath("//*[@id=\"input-email\"]")).sendKeys("test@gmail.com");
```

Every by returns BY.

Enter password using css selector

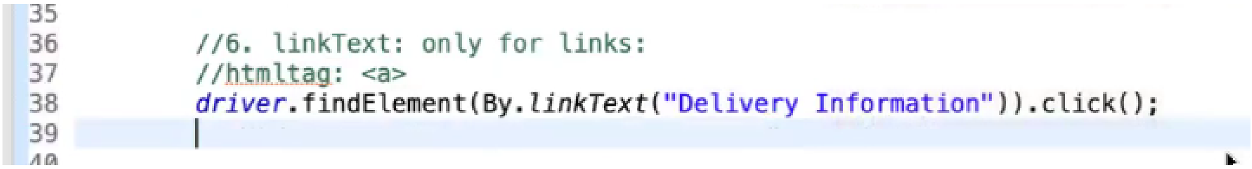
Enter confirm password using css selector



```
        //5. cssSelector: is not attribute.
        driver.findElement(By.cssSelector("#input-password")).sendKeys("test@123");
        driver.findElement(By.cssSelector("#input-confirm")).sendKeys("test@123");
```

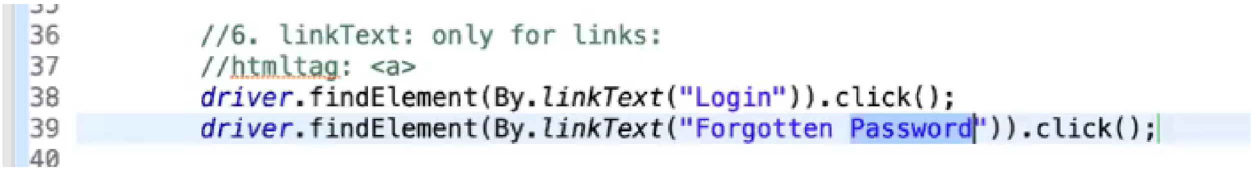
Link text

Click delivery information in footer



```

        //6. linkText: only for links:
        //htmltag: <a>
        driver.findElement(By.linkText("Delivery Information")).click();

```

Click login and forgot password



```
        //6. linkText: only for links:
        //htmltag: <a>
        driver.findElement(By.linkText("Login")).click();
        driver.findElement(By.linkText("Forgotten Password")).click();

```

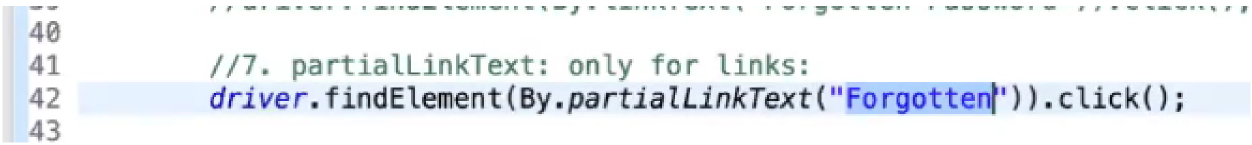
When you see <a> tag then we can use link text else no.

Partial link text

Only for links

With <a> tag

Forgot password element



```

        //7. partialLinkText: only for links:
        driver.findElement(By.partialLinkText("Forgotten")).click();

```

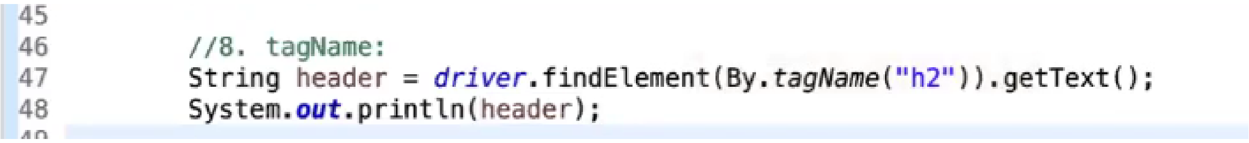
Tagname-

Pass html tag

New customer element

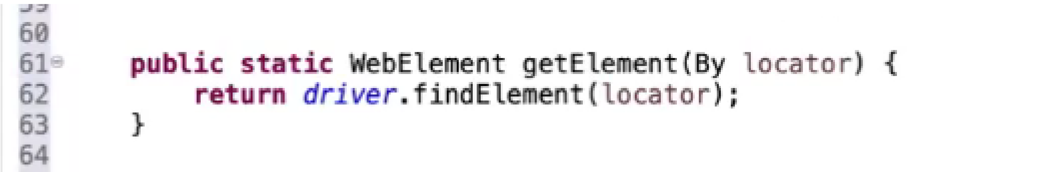
Capture text using get text method

Returns string

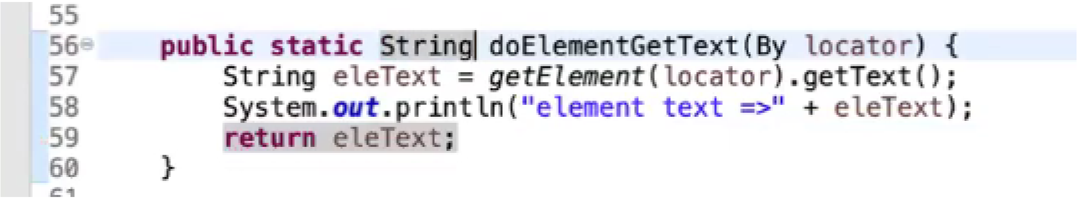


```
        //8. tagName:
        String header = driver.findElement(By.tagName("h2")).getText();
        System.out.println(header);
```

Lets make get text method generic in same class-

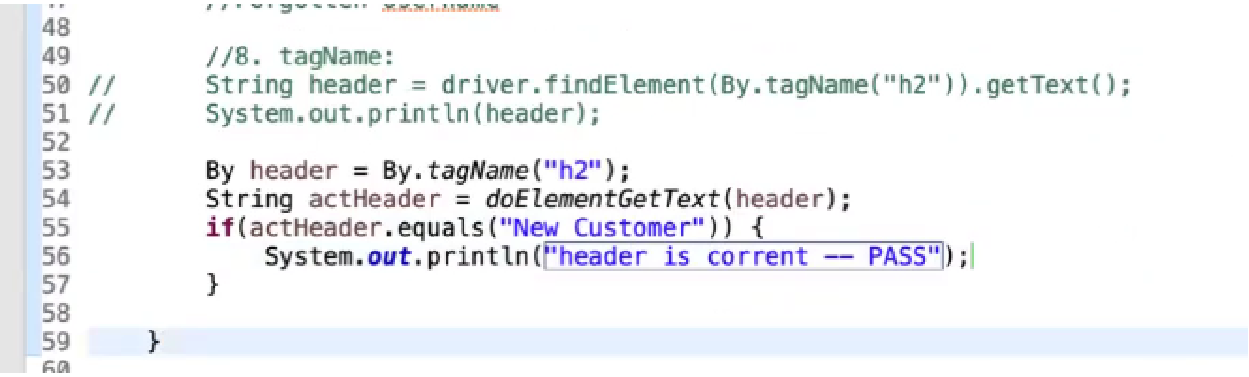


```
    public static WebElement getElement(By locator) {
        return driver.findElement(locator);
    }
```



```
    public static String doElementGetText(By locator) {
        String eleText = getElement(locator).getText();
        System.out.println("element text =>" + eleText);
        return eleText;
    }
```

Call generic method within same class

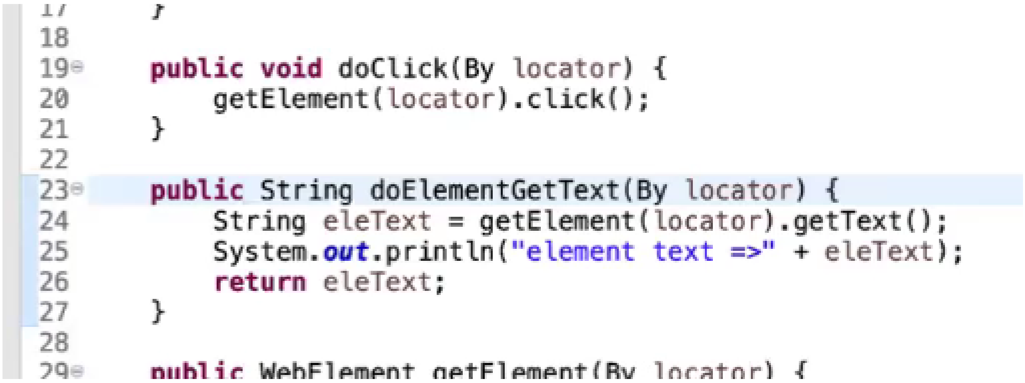


```
        //8. tagName:
//      String header = driver.findElement(By.tagName("h2")).getText();
//      System.out.println(header);

        By header = By.tagName("h2");
        String actHeader = doElementGetText(header);
        if(actHeader.equals("New Customer")) {
            System.out.println("header is corrent -- PASS");
        }

    }
```

Element util class

Get text generic method

Move the function here.



```
    }

    public void doClick(By locator) {
        getElement(locator).click();
    }

    public String doElementGetText(By locator) {
        String eleText = getElement(locator).getText();
        System.out.println("element text =>" + eleText);
        return eleText;
    }

    public WebElement getElement(By locator) {
```