EU Fast Track Selenium Day 2

▼ Locators

xpath

```
- like css, we can create custom locators
- there are 2 types of xpath locator, absolute and relative
  - absolute xpath starts with /, starts from the first tag(html tag),  step by step to locate the element
    /html/body/div/div[2]/div/div/div/h1/span
  - not recomended

  - relative xpath starts with //, starts from wherever we want
    //span[@class = "h1y"]
  - syntax: //tagName[@attribute='value']
    <div id="page-footer" class="row">
    //div[@id="page-footer"]
    //div[@class="row"]

- frequently used syntaxes:
  //tagName[@attribute='value’]
  //tagName[contains(@attribute, 'value’)]
  //tagName[.='text'] (same as //tagName[text()='text’], . represents text)
  //*[@attribute='value’] ( use * when we don't want use tag name)

- use / to go from parent to child
- use /.. to go from child to parent
- use /following-sibling::tagname  to go to next sibling
- use /preceding-sibling::tagname to go to previous sibling
  <select id="state">
  <option value="" selected="selected">Select a State</option>
  <option value="AL">Alabama</option>
  <option value="AK">Alaska</option>
  <option value="AZ">Arizona</option>
  <option value="AR">Arkansas</option>
  <option value="CA">California</option>
  <option value="CO">Colorado</option>
  <option value="CT">Connecticut</option>

//option[@value='AL']/following-sibling::option
//option[@value='AL']/preceding-sibling::option

- Dynamic web elements
  contains         //tagName[contains(@attribute, 'value')]
  starts-with      //tagName[starts-with(@attribute, 'value')]
  ends-with        //tagName[ends-with(@attribute , 'value')]
```

css vs xpath

- Css cannot locate only using text

  ```
  xpath://*[.="Don't Click!"]
  css: NA
  ```

  - Css cannot use index to locate

  ```
  xpath:--> (xpathFormula)[indexNumber]
  css: NA
  ```

- Css cannot go from child to parent, but xpath can

▼ Practice

```
1. Go to goolge, locate gmail link using linetext / partial linktext
2. Locate text area using css, xpath
3. Go to https://www.w3schools.com/, locate "Learn to code" using css and xpath
4. Locate search bar using id
5. Locate "learn html" link using parent to child ( both in css and xpath)
6. Locate R language link using child to parent (xpath)
7. Go to https://www.ebay.com/globaldeals, locate first "deal's" price using contains, starts-with, methods.
9. Locate all future deals and print their name and price
```

▼ Utilities

- add configuration.properties file
- add ConfigurationReader
- add Driver class

▼ Methods

1. findElement & findElements

   receives locator type and located element as parameter

   ```
   WebElement elementName = driver.findElement(By.LocatorStrategy("LocatorValue"));
   WebElement submitButton = driver.findElement(By.xpath("//input[@class='btn btn-primary']")
   ```

- differences: (famous interview question)
  - findElement returns a single web element, findElements returns a list of web element
  - if there is no mathching element, findElement will return **NoSuchElementException**, but findElements will return **empty list.**

2. click()
   - will click the web element
   - syntax: driver.findElement(locator).click;
3. sendKeys()
   - accept String data type as parameter, provide to that web element

   ```
   Webelement email = driver.findElement(By.xpath("*******")
   email.sendKeys("abc@gmail.com")
   ```

   - also except keyboard keys method

   ```
   element.sendKeys(Keys.ENTER)
   We also have other kes, like
   BACK_SPACE
   SHIFT
   CONTROL
   SPACE
   ...
   ...
   ```

4. getText();

- return the text between opening and closing tag as String

```
<h3>Autocomplete</h3>
System.out.println(driver.findElement(By.xpath("//h3")).getText());
Result: Autocomplete
```

5. getAttribute();

```
<div id="content" class="large-12 columns">
System.out.println(driver.findElement(By.xpath("//div[@id=\"content\"]")).getAttribute("id")); --> content
System.out.println(driver.findElement(By.xpath("//div[@id=\"content\"]")).getAttribute("class")); --> large-12 columns
```

6. isDisplayed();

- return true if the element is displayed, otherwise, return false

```
syntax: driver.findElement(locator).isDisplayed(); --> true, if displayed
syntax: driver.findElement(locator).isDisplayed(); --> false, if not displayed
```