Day 2: Extra Exercise

# Exercise 1: Random Wiki

In this exercise we will navigate to wikipedia.org, click on the 'Random article' link, and print out the title of that random article.

## 1. Instantiate WebDriver

To start, WebDriver needs to be instantiated. WebDriver is the interface that runs & manages the browser.

To do this, start your main() function with:

//For Chrome

WebDriver driver = new ChromeDriver();

## 2. Launch the browser and navigate to Wikipedia Main Page

To navigate to a particular URL you can use

driver.get(URL);

or

driver.navigate().to(URL);

Both of those do the same thing, which is launch the browser specified by the WebDriver and then navigate to the specified URL.

Consider saving the URL into its own variable. Having hard-coded strings inside the code is usually bad as it can lead to having to dredge through pages of code later when the target object changes and needs to have its references updated.

## 3. Validate the Wikipedia Homepage

Before mindlessly moving forward with the next step, it's a good idea to validate the results of your actions. For the Wikipedia homepage, a good spot to validate might be the "From today's featured article" banner, since that's an item that only exists on the Wikipedia main page and probably won't move.

To do this, you will need to first identify that web element. You can do this by finding the XPath of the element.

Right-click on the "From today's featured article" headline and click Inspect Element. This will show the source code of the page.

Using what you know about XPaths, determine the full XPath of that object.

####//body/div[2]/div[2]/div[3]/table[1]/tbody/tr/td/table/tbody/tr/td/h2/span[@id="From\_today.27s\_featured\_article"]####

You can validate this web element's presence on the page by doing:

//Search for the 'Featured Article' banner.

//If it's not found, end the test and return an error message.

String featuredArticleXpath = ###THE XPATH###

driver.findElement(By.xpath(featuredArticleXpath));

Note that driver.findElement() throws a NoSuchElementException. You can use a catch block to handle this if you want to specify an error message; otherwise, Eclipse will return a standard one for that exception.

You could also search for the web element by its unique "id" value:

...

driver.findElement(By.id("From\_today.27s\_featured\_article"));

...

In both cases, the XPath and the ID, you should use a value that is unlikely to change. For example, the XPath would probably change significantly whenever Wikipedia changed the format of the main page and put the 'Featured Article' banner somewhere different.

## 4. Click on the 'Random article' link

To get to a random page we will need to click the link. We can find the web element with its XPath.

###//body/div[3]/div[1]/div/ul/li[@id="n-randompage"]/a[href="wiki/Special:Random"]###

###Note that we can determine XPath elements by certain attributes, such as id, instead of indexes.###

To click on the link, we just need to use the click() function:

//Find the 'Random page' link and click on it.

String randomPageXpath = ###THE XPATH###

WebElement randPageLink = driver.findElement(By.xpath(randomPageXpath));

randPageLink.click();

## 5. Output the random article title.

Finally, we need to find the the text of the title of the random page. The function getText() will return the "innerText" string that a WebElement contains, if it does contain text.

Search for this web element by ID:

//Find the page title and print it.

String pageTitleID = ###THE ID### "firstHeading";

WebElement pageTitle = driver.findElement(By.id(pageTitleID));

String title = pageTitle.getText();

System.out.println("The random page is: " + title + "!\n");

return;