EU Fast Track Selenium Day 4

▼ iFrames

- Webpage inside another webpage
- Frequently used for ads, maps, interactive contents
- Starts with <iframe>
- We can locate the iframe by:
  - Name or Id
  - Index —- starting from 0
  - Web Element

```
http://practice.cybertekschool.com/iframe

WebElement iframe = driver.findElement(By.id("mce_0_ifr"));

driver.switchTo().frame(frame name or id);
driver.switchTo().frame(0);
driver.switchTo().frame(iframe);
driver.switchTo().parentFrame; // at the end, we have to switch back main frame
```

- For frames, we have to switch to main frame first, then switch to child frames.
- Differences between iframe and frames
  - A frame is used to divide a page into multiple sections, with new content on each section.
  - An iFrame is used to embed the content of the external websites into the web page, in order to avoid cross-site scripting issues

▼ Upload

- We can handle uploads using sendKeys() method
- Locate the upload button, then provide file path as parameter

```
WebElement chooseFile = driver.findElement(locator);
chooseFile.sendKeys("file path");
driver.findElement(locator).click();
```

Actions

- Allows us to do advanced mouse and keyboard events
- Comes from selenium
- Some mouse events:
  - Double click
  - Hover
  - Drag and drop
  - Right click / Double click
  - Move
- Some Keyboard events:
  - Send keys
  - Key up
  - Key down
  - Scroll up / down
- Syntax

```
Actions actions = new Actions(driver);
actions.moveToElement(Web Element).perform();
```

JS Executer

- Helps us execute JS codes in Selenium
- Only have two methods:
  - `executeScript`() — runs in the body of an anonymous function
    - click

        - type
        - scroll down/up

    - `executeAsynchScript()` — executes the asynchronous piece of JavaScript, increasing performance

```
JavascriptExecutor js = (JavascriptExecutor) driver;
js.executeScript(Script,Arguments);
```

▼ Page Object Model

What is Page Object Model?

> *Page Object Model, also known as POM, is a design pattern in Selenium that **creates an object repository for storing all web elements**.*

- Creating java class for each page of the web application.
- All of the related web elements or methods to current page will be stored to its own .java class.

Advantages:

- Reusability
- Easy to maintain
- Easy to explain to non-tech person (with BDD)

How to implement POM?

- Create a constructor and initialize the object and driver instance using PageFactory.initElements() method.

```
public LoginPage(){
        PageFactory.initElements(Driver.getDriver(), this);
```

```
    }
```

- Use @FindBy annotation to locate web elements, instead of findElement();

> 💡 StaleElementReferenceException is solved by POM Design pattern. Because every time we try to use the WebELement the reference of the Web Element will be refreshed. Therefore, no more StaleElementReferenceException

▼ BDD

> Behavior-Driven Development (BDD) is a process where team members use domain-specific language to express the expected behavior of an application in the form of scenarios.

How BDD structure looks like?

1. We have feature files written in Gherkin language (for non-tech people)
2. Step definitions are created according to feature files (for IT people)
3. All of our codes are in Step definitions
4. Then we run our tests using CukesRunner class

Gherkin language

- **Scenario**:
- **Given**: used for pre-condition
- **When**: used for when some actions happen
- **And**: used when repeating any keywords
- **Then**: used for verification part

- **Background**: for running before each scenario steps