



# SELENIUM

ELŻBIETA SADEL

GRZEGORZ WITEK

# KLASA ACTIONS

- Do symulowania skomplikowanych akcji/gestów użytkownika.
- Dokumentacja:  
<https://seleniumhq.github.io/selenium/docs/api/java/org/openqa/selenium/interactions/Actions.html>
- Na przykład do akcji typu drag and drop, najeżdżania myszką na jakiś element (np. żeby podejrzeć tooltipa), do testowania skrótów klawiaturowych, do menu kontekstowego.

```
Actions actions = new Actions(driver);  
actions.moveToElement(sleeveProduct).build().perform();
```

# INTERFEJS JAVASCRIPTEXECUTOR

- ❑ Pozwala na wykonywanie skryptów JavaScript w kontekście ramki albo okna.
- ❑ Dokumentacja: <https://seleniumhq.github.io/selenium/docs/api/java/org/openqa/selenium/JavascriptExecutor.html>
- ❑ Na przykład do wyciągania czegoś z ciasteczek, local albo session storage, ale także to symulowania akcji, które nie zadziałały z użyciem klasy Actions (np. dragAndDrop).
- ❑ Obiekty implementujące ten interfejs mają dostępne dwie metody: executeScript i executeAsyncScript.

```
JavascriptExecutor js = (JavascriptExecutor) driver;  
js.executeScript("script");
```

# WARTOŚCI ZWRACANE PRZEZ JAVASCRIPTEXECUTOR

“If the script has a return value (i.e. if the script contains a return statement), then the following steps will be taken:

- For an HTML element, this method returns a WebElement
- For a decimal, a Double is returned
- For a non-decimal number, a Long is returned
- For a boolean, a Boolean is returned
- For all other cases, a String is returned.
- For an array, return a List<Object> with each object following the rules above. We support nested lists.
- For a map, return a Map<String, Object> with values following the rules above.
- Unless the value is null or there is no return value, in which null is returned”

<https://seleniumhq.github.io/selenium/docs/api/java/org/openqa/selenium/JavascriptExecutor.html>

# EXECUTESCRIPT I EXECUTEASYNCSCRIPT

- ❑ `executeScript` - wywoływana jak zwykła funkcja, po jej zakończeniu WebDriver przejdzie do kolejnej instrukcji;
- ❑ `executeAsyncScript` - wykonywana jest w przeglądarce, nie blokuje kolejnych akcji WebDrivera

# SWITCH TO

## **WebDriver driver.switchTo().frame(WebElement/int/String)**

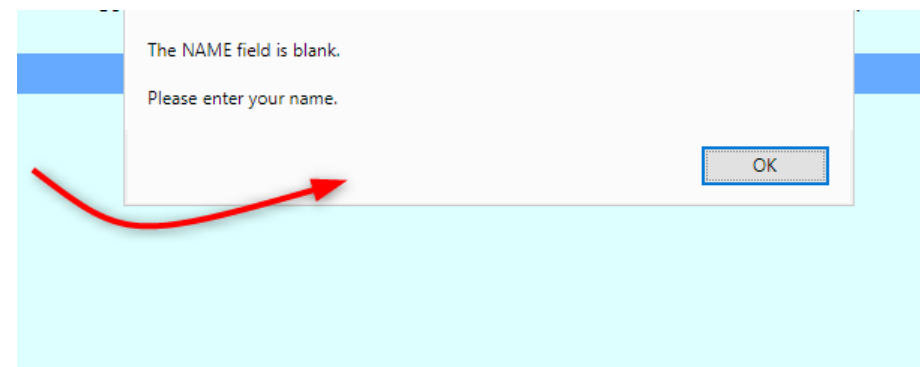
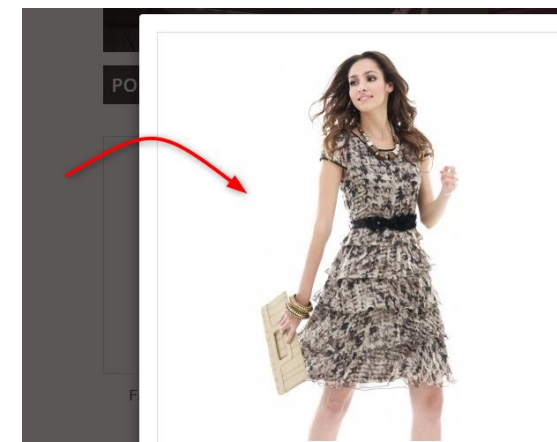
- przejście do frame lub iframe w celu używania elementów wewnętrznych

## **WebDriver driver.switchTo().defaultContent()**

- wyjście z frame lub iframe do głównej strony

## **Alert driver.switchTo().alert()**

- przełączenie się do alertu (popup), w celu zamknięcia lub pobrania tekstu)



# SWITCH TO

**WebDriver driver.switchTo().window(String) -**

- Przełączenie się do innego okna lub taba



**WebElement driver.switchTo().activeElement()**

- przełączenie się obecnego aktywnego elementu

ALREADY REGISTERED?

Email address

mail@domain.com ✓

Password

[Forgot your password?](#)



Sign in

DO DZIEŁA!!







DZIĘKUJEMY

