

Selenium

WebDriver - WebElement

WebElement

WebElement jest klasa, która reprezentuje element DOM.

DOM - Document Object Model, model reprezentujący strukturę dokumentu XML (i HTML) w sposób obiektowy.

Wyszukiwanie elementów - FindElement, FindElements

Elementy w dokumencie HTML możemy wyszukiwać w całym dokumencie (używając bezpośrednio WebDrivera) lub w części będącej potomkami innego elementu, wykonując operacje na obiekcie klasy WebElement.

- FindElement - znajduje element wskazany przez lokator, zwraca pojedynczy element. Jeżeli element nie może zostać znaleziony, rzuca wyjątek.
- FindElements - znajduje listę elementów pasujących do lokatora. Jeżeli nie ma takich elementów na stronie, zwracana jest pusta lista.

Aktywny element

Zwraca element, który w danym momencie posiada 'focus'. Może być przydatne w formularzach oraz do sprawdzenia, czy po wykonaniu akcji użytkownika focus jest przekazywany do właściwego elementu.

Java:

```
// Get attribute of current active element  
String attr = driver.switchTo().activeElement().getAttribute("title");
```

Python:

```
# Get attribute of current active element  
attr = driver.switch_to.active_element.get_attribute("title")
```

Czy element jest włączony

Sprawdza, czy element jest włączony i aktywny na interakcje.

Java:

```
//returns true if element is enabled else returns false  
boolean value = driver.findElement(By.name("btnK")).isEnabled();
```

Python:

```
# Returns true if element is enabled else returns false  
value = driver.find_element(By.NAME, 'btnK').is_enabled()
```

Czy element jest wybrany

Zwraca informację, czy element jest wybrany, czy nie. Przydatne w elementach typu: Check box, Radio Button, Input, Option.

Java:

```
//returns true if element is checked else returns false  
boolean value =  
driver.findElement(By.cssSelector("input[type='checkbox']:first-of-type")).isSelected();
```

Python:

```
# Returns true if element is checked else returns false  
value = driver.find_element(By.CSS_SELECTOR, "input[type='checkbox']:first-of-type").is_selected()
```

Jaki jest tag elementu

Zwraca nazwę taga HTML dla znalezionej elementu

Java:

//returns TagName of the element

```
String value = driver.findElement(By.cssSelector("h1")).getTagName();
```

Python:

Returns TagName of the element

```
attr = driver.find_element(By.CSS_SELECTOR, "h1").tag_name
```

Współrzędne elementu

Zwraca współrzędne ekranowe i rozmiary elementu

Java:

```
// Returns height, width, x and y coordinates referenced element  
Rectangle res = driver.findElement(By.cssSelector("h1")).getRect();  
// Rectangle class provides getX,getY, getWidth, getHeight methods  
System.out.println(res.getX());
```

Python:

```
# Navigate to url  
driver.get("https://www.example.com")  
# Returns height, width, x and y coordinates referenced element  
res = driver.find_element(By.CSS_SELECTOR, "h1").rect
```


Wartość CSS dla elementu

Zwraca wartość CSS dla wskazanej właściwości

Java:

```
// Retrieves the computed style property 'color' of linktext  
String cssValue = driver.findElement(By.linkText("More information...")).getCssValue("color");
```

Python:

```
# Retrieves the computed style property 'color' of linktext  
cssValue = driver.findElement(By.LINK_TEXT, "More information...").value_of_css_property('color')
```

Tekst danego elementu

Zwraca tekst zawarty pomiędzy tagiem otwierającym i zamykającym (również znajdujący się z zagnieżdżonych elementach).

Java:

```
// Retrieves the text of the element  
String text = driver.findElement(By.cssSelector("h1")).getText();
```

Python:

```
# Retrieves the text of the element  
  
text = driver.find_element(By.CSS_SELECTOR, "h1").text
```