Selenium WebDriver CSS Selectors

Press Space for next page →

Agenda

CSS Selectors

> Primarily designed to style elements on a webpage using Cascading Style Sheets (CSS). They can also be used for selecting elements in web scraping and automation contexts.

Syntax

> Generally simpler and more readable syntax. They use a combination of **element tags, IDs, classes, attributes, and pseudo-classes** to target elements.

XPath

> Designed for navigating and querying XML documents, including HTML. XPath offers more powerful and complex selection criteria.

Capabilities

- **CSS Selectors:** Generally faster because they use a simpler approach to find elements.
- **XPath:** Can be slower due to the complexity of parsing expressions and potentially needing to traverse the entire document.

Recommendations: Use CSS selectors whenever possible, and only use XPath when CSS selectors cannot handle the selection criteria.

Difference between CSS selector and XPath

FEATURE	CSS SELECTORS	XPATH
Purpose	Styling and selecting elements in HTML	Navigating and querying XML documents (including HTML)
Syntax	Simpler, more readable	More complex, harder to learn
Capabilities	Limited to attributes (ID, class, etc.)	Powerful, can target based on position, content, etc.
Performance	Faster	Slower
Use Cases	Simpler selection tasks, well-structured HTML	Complex scenarios, specific content/position targeting

CSS Selectors Symbols

ATTRIBUTE	SYMBOL USED
Using id	`#` symbol
Using class name	`.` symbol
Using attribute	`tagname[attribute='value']`
Using multiple attribute	`tagname[attribute1='value1'][attribute2='value2']`
Contains	`*` symbol
Starts with	`^` symbol
Ends with	`$` symbol

CSS Selector: Using ID

Syntax: `tagname#id`

Example: `input#password`

CSS Selector: Using Class Name

Syntax: `tagname.classname`

Example: `input.form-control`

CSS Selector: Using Attribute

Syntax: `tagname[attribute='value']`

Example: `input[name='username']`

CSS Selector: Using Multiple Attributes

Syntax: `tagname[attribute1='value1'][attribute2='value2']`

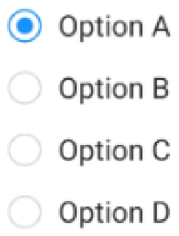
Example: `input[name='username'][type='text']`

CSS Selector: Contains, Starts With, Ends With

- **Contains:** `*` symbol
  - `input[id*='user']` matches `input` elements with an `id` attribute containing the text `user`.
- **Starts With:** `^` symbol
  - `input[id^='user']` matches `input` elements with an `id` attribute starting with the text `user`.
- **Ends With:** `$` symbol
  - `input[id$='name']` matches `input` elements with an `id` attribute ending with the text `name`.

Learn more about CSS Selectors: CSS Selectors

Code Example: Radio Button



[✓ Option A]

Option B

Option C

Option D

```java
class RadioButtons {
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("https://qbek.github.io/selenium-exercises/en/");
        driver.manage().window().maximize();

        driver.findElement(By.xpath("//a[@href=\"radio_buttons.html\"]")).click();
        List<WebElement> radio = driver.findElements(By.xpath("//input[@type='radio']"));

        for (WebElement local_radio : radio) {
            String value = local_radio.getAttribute("value");
            System.out.println("Values from radio buttons are  ==>" + value);
            if (value.equalsIgnoreCase("radiozet")) {
                local_radio.click();
            }
        }
    }
}
```

Code Example: Checkbox

```java
class CheckBox {
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("https://qbek.github.io/selenium-exercises/en/");
        driver.manage().window().maximize();

        driver.findElement(By.xpath("//a[@href=\"check_boxes.html\"]")).click();

        List<WebElement> check = driver.findElements(By.xpath("//input[@type='checkbox']"));

        for (WebElement local_check : check) {
            String name = local_check.getAttribute("name");
            System.out.println("Values from check boxes are  ⟹" + name);
            if (name.equalsIgnoreCase("blue")) {
                local_check.click();
            }
        }
    }
}
```

Thank you ❤️

qa-june-2024-automation-with-java-slides