Selenium JavaScriptExecutor

Sure! Let's dive deep into **JavaScriptExecutor in Selenium**, covering it from **beginner to expert level**, with **examples, use cases, and interview relevance**.

🔸 What is JavaScriptExecutor in Selenium?

JavaScriptExecutor is an interface provided by Selenium WebDriver to run **JavaScript code** directly within the browser from your Java code.

🔸 Why Use JavaScriptExecutor?

Some tasks cannot be handled easily using standard WebDriver methods, such as:

- Clicking hidden elements
- Scrolling into view
- Getting values from dynamic JavaScript variables
- Setting input field values

In such cases, **JavaScriptExecutor** is powerful.

🔸 How to Use JavaScriptExecutor?

✅ Import and Cast WebDriver

```
java

import org.openqa.selenium.JavascriptExecutor;

// Cast WebDriver to JavascriptExecutor
JavascriptExecutor js = (JavascriptExecutor) driver;
```

✅ Syntax

```
java

js.executeScript("JavaScript code here");
js.executeScript("JavaScript code", args...);
```

🔸 Common Use Cases with Examples

1️⃣ Scroll Down / Up the Page

```java
java

js.executeScript("window.scrollBy(0,1000)"); // Scroll down by 1000 pixels
```

2️⃣ Scroll to Specific Element

```java
java

WebElement element = driver.findElement(By.id("footer"));
js.executeScript("arguments[0].scrollIntoView(true);", element);
```

3️⃣ Click Hidden Element

```java
java

WebElement button = driver.findElement(By.id("hiddenBtn"));
js.executeScript("arguments[0].click();", button);
```

4️⃣ Set Value to Input Field

```java
java

WebElement input = driver.findElement(By.id("username"));
js.executeScript("arguments[0].value='admin';", input);
```

5️⃣ Get Page Title or Any JS-returned Value

```java
java

String title = (String) js.executeScript("return document.title;");
System.out.println("Page title: " + title);
```

🔸 Full Example

```java
java

import org.openqa.selenium.*;
import org.openqa.selenium.chrome.ChromeDriver;

public class JSExecutorExample {
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("https://example.com");

        JavascriptExecutor js = (JavascriptExecutor) driver;

        // Scroll to bottom
        js.executeScript("window.scrollTo(0, document.body.scrollHeight)");

        // Set value to a field
        WebElement input = driver.findElement(By.id("name"));
        js.executeScript("arguments[0].value='Koti';", input);

        // Click using JS
        WebElement btn = driver.findElement(By.id("submitBtn"));
```

```
        js.executeScript("arguments[0].click();", btn);

        driver.quit();
    }
}
```

🔸 Interview Questions on JavaScriptExecutor

1. **What is JavaScriptExecutor and why is it used in Selenium?**
2. **How do you scroll to an element using JavaScriptExecutor?**
3. **Can you click an element using JavaScriptExecutor?**
4. **When would you prefer JavaScriptExecutor over WebDriver methods?**
5. **How do you retrieve data using JavaScriptExecutor?**

🔸 Tips and Best Practices

- ✅ Use JavaScriptExecutor **only when WebDriver fails**.
- ✅ Always check element visibility before using JS click.
- ✅ Avoid overuse; it may make tests brittle or dependent on UI JS behavior.

🔸 Advanced Use (Expert Level)

✅ Highlight Element (for debugging)

```
java

js.executeScript("arguments[0].style.border='3px solid red'", element);
```

✅ Disable Element

```
java

js.executeScript("arguments[0].setAttribute('disabled','true')", element);
```

✅ Simulate Mouse Over

```
java

js.executeScript("var evObj = document.createEvent('MouseEvents');" +
  "evObj.initMouseEvent('mouseover', true, true, window);" +
  "arguments[0].dispatchEvent(evObj);", element);
```

✅ Summary Table

Action	JavaScript Code Example
Scroll down	`window.scrollBy(0,1000);`
Scroll to element	`arguments[0].scrollIntoView(true);`
Click using JS	`arguments[0].click();`
Set value	`arguments[0].value='test';`
Get title	`return document.title;`
Highlight element	`arguments[0].style.border='3px solid red';`

Would you like me to give you **JavaScriptExecutor hands-on assignments or practice tasks**?

Great! Here are **10 hands-on JavaScriptExecutor assignments** in **Selenium with Java**, ranging from beginner to advanced level.

🔰 Prerequisites

Before starting:

- Ensure you're using **Selenium WebDriver** with **Java**.
- Use **ChromeDriver** or FirefoxDriver.
- Make sure `JavascriptExecutor` is cast properly:

```
java

JavascriptExecutor js = (JavascriptExecutor) driver;
```

🔪 Hands-on Assignments

✅ Assignment 1: Scroll Down the Page

Objective: Open a long webpage and scroll down by 1000 pixels.

Steps:

1. Open `https://en.wikipedia.org/wiki/Main_Page`
2. Scroll down using JS by 1000 pixels
3. Wait 2 seconds and scroll back up

```
java

js.executeScript("window.scrollBy(0, 1000);");
Thread.sleep(2000);
js.executeScript("window.scrollBy(0, -1000);");
```

✅ Assignment 2: Scroll to a Specific Element

Objective: Scroll to the footer section on a page

Steps:

1. Go to `https://www.selenium.dev/`
2. Scroll to the footer element

```
java

WebElement footer = driver.findElement(By.tagName("footer"));
js.executeScript("arguments[0].scrollIntoView(true);", footer);
```

✅ Assignment 3: Click a Hidden Element

Objective: Click a button using JavaScriptExecutor

Steps:

1. Navigate to **`https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_submit_get`**
2. Switch to iframe and click **"Try it"** button using JS

```
java

driver.switchTo().frame("iframeResult");
WebElement btn = driver.findElement(By.xpath("//input[@type='submit']"));
js.executeScript("arguments[0].click();", btn);
```

✅ Assignment 4: Set Input Field Value

Objective: Fill a disabled text field using JS

Steps:

1. Create a local HTML file with a disabled input
2. Set the value using JS

```
java

js.executeScript("document.getElementById('myInput').value='admin';");
```

✅ Assignment 5: Get Page Title Using JS

Objective: Get and print page title using JavaScriptExecutor

```
java

String title = (String) js.executeScript("return document.title;");
System.out.println("Page Title: " + title);
```

✅ Assignment 6: Highlight an Element

Objective: Visually highlight the Google Search input box

Steps:

1. Go to **`https://www.google.com`**
2. Highlight the search box

```
java

WebElement input = driver.findElement(By.name("q"));
```

```
js.executeScript("arguments[0].style.border='3px solid red'", input);
```

✅ Assignment 7: Simulate Mouse Hover (Advanced)

Objective: Hover over a menu using JS

Steps:

1. Go to **`https://demoqa.com/menu`**
2. Hover over “Main Item 2”

```
java

WebElement item = driver.findElement(By.xpath("//a[text()='Main Item 2']"));
js.executeScript(
    "var evObj = document.createEvent('MouseEvents');" +
    "evObj.initEvent('mouseover', true, false);" +
    "arguments[0].dispatchEvent(evObj);", item);
```

✅ Assignment 8: Disable a Button Using JS

Objective: Disable a submit button using JS

```
java

WebElement btn = driver.findElement(By.id("submit"));
js.executeScript("arguments[0].setAttribute('disabled', 'true');", btn);
```

✅ Assignment 9: Scroll Through Entire Page

Objective: Scroll from top to bottom automatically

```
java

long scrollHeight = (long) js.executeScript("return document.body.scrollHeight");

for (int i = 0; i < scrollHeight; i += 500) {
    js.executeScript("window.scrollBy(0, 500);");
    Thread.sleep(1000);
}
```

✅ Assignment 10: Flash Element for Debugging (Expert)

Objective: Create a flash effect by changing background color of an element repeatedly

```
java

WebElement element = driver.findElement(By.id("myElement"));
for (int i = 0; i < 5; i++) {
    js.executeScript("arguments[0].style.backgroundColor='yellow'", element);
    Thread.sleep(300);
```

```
    js.executeScript("arguments[0].style.backgroundColor=''", element);
}
```

🎯 Bonus: Create Your Own Mini Project

Build a Selenium test that:

- Opens a form
- Fills it using **`JavaScriptExecutor`**
- Highlights all fields
- Clicks submit with JS
- Prints a message after submission

Would you like me to send you the HTML form code for local testing or help you build a full mini framework around JavaScriptExecutor use cases?