Selenium JavaScriptExecutor – Complete Guide

JavaScriptExecutor is used in Selenium to execute JavaScript directly in the browser when standard Selenium actions fail or are insufficient.

1. Highlight Element

Used for debugging and demo purposes.

```
JavascriptExecutor js = (JavascriptExecutor) driver;
WebElement element = driver.findElement(By.id("login"));

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

2. Scroll Operations

1 Scroll by pixels
2 Scroll to bottom of page
3 Scroll to a specific element

```
// Scroll by pixels
js.executeScript("window.scrollBy(0,500)");

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

// Scroll to element
js.executeScript(
  "arguments[0].scrollIntoView(true);",
  element
);
```

3. Click Using JavaScript

Used when ElementClickInterceptedException occurs.

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

4. Set Value Using JavaScript

```
js.executeScript(
  "arguments[0].value='admin';",
  element
);
```

5. Get Page Information

```
String title = js.executeScript(
  "return document.title;").toString();

String url = js.executeScript(
  "return document.URL;").toString();
```

Best Practices

1. Avoid overusing JavaScriptExecutor
2. Prefer Selenium actions + waits first
3. Use JS mainly for scroll, highlight, and edge cases

Interview Tip: JavaScriptExecutor is commonly used to bypass UI synchronization issues and interact directly with the DOM.