**Selenium Class**

# Chapter#13 – Scrolling in Selenium Webdriver

1. **Scroll Bottom**
2. **Scroll Top**
3. **Scroll Left**
4. **Scroll Right**
5. **Scroll To Element**

Scrolling a webpage is required in the automation when the application requires scrolling down or up to display additional information e.g. most of the e-commerce sites display only 10-20 products at a time and then load more products as the user scrolls down. In this tutorial, we’ll take an example of an e-commerce website – Flipkart and automate the scrolling in Selenium in order to fetch more results.

In this script, first, we will launch filpkart.com, write a search term and then scroll down to fetch more results corresponding to that search term. Automating page scrolling will make use of “scrollBy” method of javascript. For executing the javascript method we will use Javascript executor. The scrollBy method takes two parameters one each for horizontal and vertical scroll in terms of pixels.

## Scroll Bottom

**window.scrollTo(0, document.body.scrollHeight)**

This code is scrolling the browser window to the bottom of the page. Here's a breakdown of the scrollTo function parameters:

**0:** This is the x-coordinate. In this case, it's set to 0, meaning it scrolls horizontally to the left (since it's often not necessary to scroll horizontally when reaching the bottom of a page).

**document.body.scrollHeight:** This represents the y-coordinate. It's set to the total height of the body element, effectively scrolling to the bottom of the page.

## Scroll Top

**window.scrollTo(0, -document.body.scrollHeight)**

This code is scrolling the browser window to the top of the page but with a vertical offset equal to the negative value of the total height of the body element. Here's a breakdown of the parameters:

**0:** This is the x-coordinate, indicating no horizontal scrolling (staying at the leftmost position).

**-document.body.scrollHeight:** This represents the y-coordinate. It's set to the negative total height of the body element.

## Scroll Right

**window.scrollTo(document.body.scrollWidth, 0)**

Utilizes the JavaScript executeScript method, and it's responsible for scrolling the web page horizontally. Let's break it down:

This line of code scrolls the web page horizontally to the rightmost edge, ensuring that content on the right side of the page becomes visible. It's often used in scenarios where you need to interact with or capture elements that are initially not visible because they are located to the right of the initial viewport.

**document.body.scrollWidth:** This represents the target x-coordinate for scrolling. It is set to the total width of the body element, effectively scrolling to the rightmost edge of the page.

**0:** This is the y-coordinate and indicates no vertical scrolling (staying at the topmost position).

## Scroll Left

**window.scrollTo(-document.body.scrollWidth, 0)**

Utilizes the JavaScript executeScript method, and it's responsible for scrolling the web page horizontally. Let's break it down:

This line of code scrolls the web page horizontally to the leftmost edge, ensuring that content on the left side of the page becomes visible. It's used in scenarios similar to the previous case, where horizontal visibility or interaction with elements on the left side of the page is required.

**-document.body.scrollWidth:** This represents the target x-coordinate for scrolling. It is set to the negative total width of the body element, effectively scrolling to the leftmost edge of the page.

**0:** This is the y-coordinate, indicating no vertical scrolling (staying at the topmost position).

## Scroll To Element

**arguments[0].scrollIntoView()**

The jse.executeScript("arguments[0].scrollIntoView()", element) JavaScript code is typically used in Selenium WebDriver to scroll the webpage to bring a specific element into the view.

**arguments[0].scrollIntoView()**: This is the JavaScript code being executed. It uses the scrollIntoView() method, which is a standard JavaScript method for scrolling an element into the visible area of the browser window.

element: This is the parameter passed to the JavaScript code. It represents the web element you want to bring into view. The arguments[0] syntax is a way to pass the element as an argument to the JavaScript code.