|  |  |  |
| --- | --- | --- |
| **Command Name** | **Description** | **Syntax** |
| **Click command** | This command is used to click on a Webelement. For the element to be clickable, the element must be visible on the webpage. This command is used for checkbox and radio button operations as well. | IWebelement element = driver.FindElement(By.xpath("xpath of Webelement")); element.Click(); |
| **Clear command** | This command is specifically used for clearing the existing contents of textboxes. | IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));  element.Clear(); |
| **SendKeys command** | This command is used to input a value onto text boxes. The value to be entered must be passed as a parameter to | IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));  element.SendKeys("guru99"); |
| **Displayed command** | This command is used to identify if a specific element is displayed on the webpage. This command returns a Boolean value; true or false depending on the visibility of web element. | IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));  Boolean status = element.Displayed; |
| **Enabled command** | This command is used to identify if a particular web element is enabled on the web page. This command returns a Boolean value; true or false as a result. | IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));  Boolean status = element.Enabled; |
| **Selected command** | This command is used to identify if a particular web element is selected. This command is used for checkboxes,radio buttons, and select operations. | IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));  Boolean status = element.Selected; |
| **Submit command:** | This command is similar to click command, The difference lies in whether the HTML form has a button with the type Submit. While the click command clicks on any button, submit command clicks on the only the buttons with type submit. | IWebelement element = driver.FindElement(By.xpath("xpath of Webelement")); element.submit(); |
| **Text command** | This command returns the innertext of a Webelement. This command returns a string value as a result. | IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));  String text=element.Text; |
| **TagName command** | This command returns the HTML tag of a web element. It returns a string value as the result. | IWebelement element = driver.FindElement(By.xpath("xpath of Webelement")); String tagName = element.TagName; |
| **GetCSSValue Command:** | This method is used to return the color of a web element on the form of a rgba string (Red,Green,Blue, and Alpha). | IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));  String color = element.getCSSValue;  **Output**– If the color of element is red, output would be rgba(255,0,0,1) |

**Browser commands:**

|  |  |  |
| --- | --- | --- |
| **Command Name** | **Description** | **Syntax** |
| **Url Command** | This command is used to open a specified URL in the browser. | driver.Url = "https://www.guru99.com" |
| **Title Command** | This command is used to retrieve the page title of the web page that is currently open | String title = driver.Title |
| **PageSource Command** | This command is used to retrieve the source code of web page that is currently open. | String pageSource = driver.PageSource |
| **Close Command** | This command is used to close the recently opened browser instance. | driver.Close(); |
| **Quit Command** | This command is used to close all open browser instances | driver.Quit(); |
| **Back Command** | This command is used to navigate to the previous page of browser history. | driver.Navigate().Back(); |
| **Forward Command** | This command is used to navigate to the next page of browser history. | driver.Navigate().Forward() |
| **Refresh Command** | This command is used to perform browser refresh. | driver.Navigate().Refresh() |

Following is a list of browser commands available in C# with Selenium.

**Webelement Commands:**

A Webelement represents all the elements on a web page. They are represented by HTML tags. Each of the buttons, textboxes, links, images, tables, and frames fall under Webelements. Operations on web elements can be triggered using the IWebelement interface. To interact with a Webelement, we need to find the element on the webpage and then perform operations on it. Tools like Firebug and Firepath can be used to identify the [Xpath](https://www.guru99.com/xpath-selenium.html) of Webelement.

The following are the list of Webelement commands available in C#.

**Dropdown Commands:**

|  |  |  |
| --- | --- | --- |
| **Command Name** | **Description** | **Syntax** |
| **SelectByText Command** | This command selects an option of a dropdown based on the text of the option. | IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));  SelectElement select = new SelectElement(element);  select.SelectByText("Guru99"); |
| **SelectByIndex Command** | This command is used to select an option based on its index. Index of dropdown starts at 0. | IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));  SelectElement select = new SelectElement(element);  select.SelectByIndex("4"); |
| **SelectByValue Command** | This command is used to select an option based on its option value. | IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));  SelectElement select = new SelectElement(element);  select.SelectByValue("Guru99"); |
| **Options Command** | This command is used to retrieve the list of options displayed in a dropdown. | IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));  SelectElement select = new SelectElement(element);  List<IWebelement> options = select. Options;  int size = options.Count;  for(int i=0;i<options.size();i++)  {  String value = size.elementAt(i).Text;  Console.writeLine(value);  }  The above code prints all the options onto console within a dropdown. |
| **IsMultiple command** | This command is used to identify if a dropdown is a multi select dropdown; A multi select dropdown enables user to select more than one option in a dropdown at a time. This command returns a Boolean value. | IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));  SelectElement select = new SelectElement(element);  Boolean status = select.IsMultiple(); |
| **DeSelectAll command** | This command is used in multi select dropdowns. It clears the options that have already been selected. | IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));  SelectElement select = new SelectElement(element);  select.DeSelectAll(); |
| **DeSelectByIndex command** | This command deselects an already selected value using its index. | IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));  SelectElement select = new SelectElement(element);  select.DeSelectByIndex("4"); |
| **DeSelectByValue command** | This command deselects an already selected value using its value. | IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));  SelectElement select = new SelectElement(element);  select.DeSelectByValue("Guru99"); |
| **DeSelectByText command** | This command deselects an already selected value using its text. | IWebelement element = driver.FindElement(By.xpath("xpath of Webelement"));  SelectElement select = new SelectElement(element);  select.DeSelectByText("Guru99"); |

Dropdown operations in C# can be achieved using the SelectElement class.

The following are the various dropdown operations available in C#.