Katalon Keywords Explanation

This document explains the usage of keywords built into Katalon.

KMS Technology

123 Cong Hoa Street,
Ward 12,Tan Binh District,
Ho Chi Minh City

(84 8) 3811 9977

(84 8) 3811 9955

11/20/2014

Table of Content

1. Alert Class Action

1.1 acceptAlert

Description	Simulate users clicking on "OK" button of an alert popup (alert, confirmation popup, prompt popup)
Object	N/a
Input	N/a
Output	N/a

Example **Scenario** – Users want to click on "OK" button of an alert.

Item	Object	Input	Output
acceptAlert			

1.2 dismissAlert

Description	Simulate users clicking on "Cancel" button of an alert popup (alert, confirmation popup, prompt popup)
Object	N/a
Input	N/a
Output	N/a

Example **Scenario** – Users want to click on "Cancel" button of a confirmation popup.

Item	Object	Input	Output
dismissAlert			

1.3 getAlertText

Description	Get displayed text of an alert popup (alert, confirmation popup, prompt popup)

Object	N/a
Input	N/a
Output	String result – text of the alert

Example **Scenario** – Users want to get displayed text of an alert. The output is stored into variable var_result.

Item	Object	Input	Output
getAlertText			var_result

1.4 setAlertText

Description	Simulate users typing text into a prompt popup
Object	N/a
Input	1. String value – text to type into the prompt popup
Output	N/a

Example **Scenario** – Users want to type "KMS Technology" into a prompt popup.

Item	Object	Input	Output
setAlertText		"KMS Technology"	

1.5 verifyAlertNotPresent (not implemented yet)

Description	Verify if there is **NOT** an alert
Object	N/a
Input	N/a
Output	boolean result – **true** if there is **NOT** an alert; otherwise, **false**

Example **Scenario** – Users want to verify if there is **NOT** an alert. The output is stored into variable var_result.

Item	Object	Input	Output
verifyAlertNotPresent			var_result

1.6 verifyAlertPresent (not implemented yet)

Description	Verify if there is an alert
Object	N/a
Input	N/a
Output	boolean result – **true** if there is an alert; otherwise, **false**

Example **Scenario** – Users want to verify if there is an alert. The output is stored into variable var_result.

Item	Object	Input	Output
verifyAlertPresent			var_result

1.7 waitForAlertNotPresent (not implemented yet)

Description	Wait for an alert to not present (disappear) within the given time in second unit
Object	N/a
Input	1. int seconds – the amount of seconds to wait
Output	N/a

Example **Scenario** – Users want to wait for an alert to disappear within 10 seconds.

Item	Object	Input	Output
waitForAlertNotPresent		10	

1.8 waitForAlertPresent (not implemented yet)

Description	Wait for an alert to present (appear) within the given time in second unit
Object	N/a
Input	1. int seconds – the amount of seconds to wait
Output	N/a

Example **Scenario** – Users want to wait for an alert to appear within 10 seconds.

Item	Object	Input	Output
waitForAlertPresent		10	

2. Browser Action

2.1 authenticate

Description	Navigate to a page that requires authentication. System will enter username and password
Object	N/a
Input	1. String url – url of the page to navigate 2. String username

	3. String password 4. int waitingTime – time to wait since navigating to the page until entering username
Output	N/a

Example **Scenario** – Users want to navigate to page requiring authentication.

Item	Object	Input	Output
authenticate		"https://www.example.com", "testuser", "P@ssword", 2	

2.2 back

Description	Simulate users clicking "back" button on their browser
Object	N/a
Input	N/a
Output	N/a

Example **Scenario** – Users want to click on "back" button.

Item	Object	Input	Output
back			

2.3 closeBrowser

Description	Close the browser. This action will close all windows of the browser.
Object	N/a
Input	N/a
Output	N/a

Example **Scenario** – Users want to close browser.

Item	Object	Input	Output
closeBrowser			

2.4 closeWindowIndex

Description	Close window with the given index. Window index is counted by order of window's appearance and starts from 0. After closing a window, all windows appeared after that window will have index decreased by 1. For example: we have 4 windows (index: 0, 1, 2, 3); close window index 1, after closing, the window with old

	index 2 will become new index 1, the window with old index 3 will become new index 2. If users close current window, system will switch to the first window. If current window happens to be the first window, system will switch to the new first window. However we strongly recommend users switch to another window before closing current window to prevent any confusion.
Object	N/a
Input	1. int index – index of the window to close
Output	N/a

Example **Scenario** – Users want to close the second window (index 1).

Item	Object	Input	Output
closeWindowIndex		1	

2.5 closeWindowTitle

Description	Close window with the given title. If users close current window, system will switch to the first window. If current window happens to be the first window, system will switch to the new first window. However we strongly recommend users switch to another window before closing current window to prevent any confusion.
Object	N/a
Input	1. String title – title of the window to close
Output	N/a

Example **Scenario** – Users want to close the window with title "Contact Page".

Item	Object	Input	Output
closeWindowTitle		"Contact Page"	

2.6 closeWindowUrl

Description	Close window with the given url. If users close current window, system will switch to the first window. If current window happens to be the first window, system will switch to the new first window. However we strongly recommend users switch to another window before closing current window to prevent any confusion.
Object	N/a

Input	1. String url – url of the window to close
Output	N/a

Example **Scenario** – Users want to close the window with url "http://www.kms-technology.com/contact.html".

Item	Object	Input	Output
closeWindowUrl		"http://www.kms-technology.com/contact.html"	

2.7 deleteAllCookies

Description	Delete all cookies of all windows.
Object	N/a
Input	N/a
Output	N/a

Example **Scenario** – Users want to delete all cookies.

Item	Object	Input	Output
deleteAllCookies			

2.8 forward

Description	Simulate users clicking "forward" button on their browser
Object	N/a
Input	N/a
Output	N/a

Example **Scenario** – Users want to click "forward" button.

Item	Object	Input	Output
forward			

2.9 getUrl

Description	Get url of the current window
Object	N/a
Input	N/a
Output	1. String url – url of the current window

Example **Scenario** – Users want to get url of the current window. The output is stored into variable var_url.

Item	Object	Input	Output
getUrl			var_url

2.10 getWindowTitle

Description	Get title of the current window
Object	N/a
Input	N/a
Output	1. String title – title of the current window

Example **Scenario** – Users want to get title of the current window. The output is stored into variable var_title.

Item	Object	Input	Output
getWindowTitle			var_title

2.11 maximizeWindow

Description	Resize current window to take up the entire screen
Object	N/a
Input	N/a
Output	N/a

Example **Scenario** – Users want to maximize the window.

Item	Object	Input	Output
maximizeWindow			

2.12 navigateToUrl

Description	Navigate to the specified web page
Object	N/a
Input	1. String url – url of the web page to navigate to
Output	N/a

Example **Scenario** – Users want to navigate to the page "http://www.kms-technology.com/".

Item	Object	Input	Output
navigateToUrl		"http://www.kms-technology.com/"	

2.13 openBrowser

Description	Open browser and navigate to the specified url; if url is left empty, just open browser
Object	N/a
Input	1. String url – url of the web page to be opened, can be left empty
Output	N/a

Example **Scenario** – Users want to open browser and navigate to the page "http://www.kms-technology.com/".

Item	Object	Input	Output
openBrowser		"http://www.kms-technology.com/"	

2.14 refresh

Description	Simulate users clicking "refresh" button on their browser
Object	N/a
Input	N/a
Output	N/a

Example **Scenario** – Users want to refresh the page.

Item	Object	Input	Output
refresh			

2.15 switchToDefaultContent

Description	Use after an action on child object of an iframe.
Object	N/a
Input	N/a
Output	N/a

Example **Scenario** – Users want to set text into "txtbx_username" and "txtbx_password" textboxes. Those elements lie on an iframe.

Item	Object	Input	Output
setText	txtbx_username	"test"	
switchToDefaultContent			
setText	txtbx_password	"secret"	
switchToDefaultContent			

2.16 switchToWindowIndex

Description	Switch to the window with given index. Window index is counted by order of window's appearance and starts from 0. After switching, window index does NOT change, i.e. the first window still has index 0, the second window still has index 1 and so on.
Object	N/a
Input	1. int index – index of the window to switch to
Output	N/a

Example **Scenario** – Users want to switch to the third window (index 2) then switch back to the first window (index 0).

Item	Object	Input	Output
switchToWindowIndex		2	
switchToWindowIndex		0	

2.17 switchToWindowTitle

Description	Switch to the window with the given title
Object	N/a
Input	1. String title – title of the window to switch to
Output	N/a

Example **Scenario** – Users want to switch to the window with title "Home Page".

Item	Object	Input	Output
switchToWindowTitle		"Home Page"	

2.18 switchToWindowUrl

Description	Switch to the window with the given url
Object	N/a
Input	1. String url – title of the window to switch to
Output	N/a

Example **Scenario** – Users want to switch to the window with url "http://www.kms-technology.com/contact.html".

Item	Object	Input	Output
switchToWindowUrl		"http://www.kms-technology.com/contact.html"	

2.19 verifyTextNotPresent

Description	Verify if the given texts do **NOT** present anywhere in the page source
Object	N/a
Input	1. String text – text to be verified if **NOT** presenting anywhere in the page source 2. boolean isRegex – **true** if text is regular expression; otherwise, **false**
Output	boolean result – **true** if text does **NOT** present anywhere in the page source; otherwise, **false**

Example **Scenario** – Users want to verify if text "KMS Technology" does **NOT** present anywhere in the page source. Not use regular expression. The output is stored into variable var_result.

Item	Object	Input	Output
verifyTextNotPresent		"KMS Technology";false	var_result

2.20 verifyTextPresent

Description	Verify if the given texts present anywhere in the page source
Object	N/a
Input	1. String text – text to be verified if presenting anywhere in the page source 2. boolean isRegex – **true** if text is regular expression; otherwise, **false**
Output	boolean result – **true** if text presents anywhere in the page source; otherwise, **false**

Example **Scenario 1** – Users want to verify if text "KMS Technology" presents anywhere in the page source. Not use regular expression. The output is stored into variable var_result.

Item	Object	Input	Output
verifyTextPresent		"KMS Technology"; false	var_result

Scenario 2 – Users want to verify if texts "KMS" and "Technology" present anywhere in the page source. In addition, there could be some characters between those two texts. Use regular expression. The output is stored into variable var_result.

Item	Object	Input	Output
verifyTextPresent		".*KMS.*Technology.*"; true	var_result

2.21 waitForPageLoad

Description	Wait for the web page to load within the given time in second unit
Object	N/a
Input	1. int seconds – the amount of seconds to wait
Output	N/a

Example **Scenario** – Users want to wait for the web page to load within 10 seconds.

Item	Object	Input	Output
waitForPageLoad		10	

3. Keyboard Action

3.1 sendKeys

Description	Simulates keystroke events on the specified element, as though you typed the value key-by-key
Object	TestObject object – represent a web element
Input	1. value – the combination of keys to type
Output	N/a

Example **Scenario** – Users want to type "KMS" into the text-box "txtbx_Search".

Item	Object	Input	Output
sendKeys	txtbx_Search	"KMS"	

4. Mouse Action

4.1 click

Description	Click on the given web element
Object	TestObject object – represent a web element
Input	N/a
Output	N/a

Example **Scenario** – Users want to click on the object "btn_Login".

Item	Object	Input	Output
click	btn_Login		

4.2 doubleClick

Description	Double click on the given web element
Object	TestObject object – represent a web element
Input	N/a
Output	N/a

Example **Scenario** – Users want to double click on the object "btn_Login".

Item	Object	Input	Output
doubleClick	btn_Login		

4.3 dragAndDropByOffset

Description	Drags an object a certain distance and then drops it.
Object	TestObject object – represent a web element
Input	1. int xOffset - x offset in pixels from the current location to which the element should be moved, e.g., 70, -30 2. int yOffset - y offset in pixels from the current location to which the element should be moved, e.g., 70, -30
Output	N/a

Example **Scenario** – Users want to drag image "img_Pic1" and drop it at a location that is 30 pixels on the left and 70 pixels down.

Item	Object	Input	Output
dragAndDropByOffset	img_Pic1	-30;70	

4.4 dragAndDropToObject

Description	Drags an object and then drops it to another object.
Object	TestObject object – represent a web element
Input	1. TestObject destinationObject
Output	N/a

Example **Scenario** – Users want to drag image "img_Pic1" and drop onto image "img_Pic2". Currently, users must switch to Advanced mode to add destination object.

Item	Object	Input	Output
dragAndDropToObject	img_Pic1	img_Pic2	

4.5 mouseOver

Description	Simulate users hovering a mouse over the given element
Object	TestObject object – represent a web element
Input	N/a
Output	N/a

Example **Scenario** – Users want to hover over the object "btn_Login".

Item	Object	Input	Output
mouseOver	btn_Login		

4.6 rightClick

Description	Right click on the given web element
Object	TestObject object – represent a web element
Input	N/a
Output	N/a

Example **Scenario** – Users want to right click on the object "btn_Login".

Item	Object	Input	Output
rightClick	btn_Login		

5. Select Class Action

5.1 deselectAllOption

Description	Deselect all options
Object	TestObject object – represent a web element
Input	N/a
Output	N/a

Example **Scenario** – There is a select element with the following html code:

```
<select id="selenium_commands" class="input-
xlarge" name="selenium_commands" multiple="multiple">
        <option value="browser_commands">Browser
        Commands</option>
        <option value="navigation_commands">Navigation
        Commands</option>
        <option value="switch_commands">Switch
        Commands</option>
        <option value="wait_commands">Wait
        Commands</option>
```

```
        <option value="webelement_commands">WebElement
        Commands</option>
</select>
```

Users want to deselect all options.

Item	Object	Input	Output
deselectAllOption	sl_Commands		

5.2 deselectOptionByIndex

Description	Deselect the options at the given indices. Index starts from 0.
Object	TestObject object – represent a web element
Input	1. String index_range – index range of the options to be deselected Example: 2 - index 2 2,3 - index 2 and 3 2-5 - index 2 to 5 (2, 3, 4, 5)
Output	N/a

Example **Scenario** – There is a select element with the following html code:

```
<select id="selenium_commands" class="input-
xlarge" name="selenium_commands" multiple="multiple">
        <option value="browser_commands">Browser
        Commands</option>
        <option value="navigation_commands">Navigation
        Commands</option>
        <option value="switch_commands">Switch
        Commands</option>
        <option value="wait_commands">Wait
        Commands</option>
        <option value="webelement_commands">WebElement
        Commands</option>
</select>
```

Users want to deselect "Navigation Commands" option (index 1).

Item	Object	Input	Output
deselectOptionByIndex	sl_Commands	"1"	

Users want to deselect all options from index 0 to 3.

Item	Object	Input	Output
deselectOptionByIndex	sl_Commands	"0-3"	

Users want to deselect "Browser Commands" option (index 0) and "Wait Commands" option (index 3).

Item	Object	Input	Output
deselectOptionByIndex	sl_Commands	"0,3"	

5.3 deselectOptionByLabel

Description	Deselect the options with the given label (displayed text)
Object	TestObject object – represent a web element
Input	1. String label – displayed text of the options to be deselected 2. boolean isRegex – **true** if label is regular expression, **false** if not
Output	N/a

Example

Scenario – There is a select element with the following html code:

```
<select id="selenium_commands" class="input-
xlarge" name="selenium_commands" multiple="multiple">
        <option value="browser_commands">Browser
        Commands</option>
        <option value="navigation_commands">Navigation
        Commands</option>
        <option value="switch_commands">Switch
        Commands</option>
        <option value="wait_commands">Wait
        Commands</option>
        <option value="webelement_commands">WebElement
        Commands</option>
</select>
```

Users want to deselect "Browser Commands" option.

Item	Object	Input	Output
deselectOptionByLabel	sl_Commands	"Browser Commands", false	

Users want to deselect "Wait Commands" option and "WebElement Commands" option.

Item	Object	Input	Output
deselectOptionByLabel	sl_Commands	"^W(ai|ebElemen)t Commands$", true	

5.4 deselectOptionByValue

Description	Deselect the options with the given value
Object	TestObject object – represent a web element
Input	1. String value – value of the options to be deselected 2. boolean isRegex – **true** if value is regular expression, **false** if not
Output	N/a

Example

Scenario – There is a select element with the following html code:

```
<select id="selenium_commands" class="input-
xlarge" name="selenium_commands" multiple="multiple">
        <option value="browser_commands">Browser
        Commands</option>
        <option value="navigation_commands">Navigation
        Commands</option>
        <option value="switch_commands">Switch
        Commands</option>
        <option value="wait_commands">Wait
        Commands</option>
        <option value="webelement_commands">WebElement
        Commands</option>
</select>
```

Users want to deselect "Browser Commands" option (value "browser_commands").

Item	Object	Input	Output
deselectOptionByValue	sl_Commands	"browser_commands", false	

Users want to deselect "Wait Commands" option (value "wait_commands") and "WebElement Commands" option (value "webelement_commands").

Item	Object	Input	Output
deselectOptionByValue	sl_Commands	"^w(ai|ebelemen)t_commands$", true	

5.5 getNumberOfSelectedOption

Description	Count the number of options which are being selected
Object	TestObject object – represent a web element
Input	N/a
Output	int number – the number of selected options

Example

Scenario – There is a select element with the following html code:

```
<select id="selenium_commands" class="input-
xlarge" name="selenium_commands" multiple="multiple">
        <option value="browser_commands">Browser
        Commands</option>
        <option value="navigation_commands">Navigation
        Commands</option>
        <option value="switch_commands">Switch
        Commands</option>
        <option value="wait_commands">Wait
        Commands</option>
```

```
        <option value="webelement_commands">WebElement
        Commands</option>
</select>
```

Users want to count the number of selected options of the web element. Assume that there are 2 options being selected. The output (2) is stored into variable var_number.

Item	Object	Input	Output
getNumberOfSelectedOption	sl_Commands		var_number

5.6 getNumberOfTotalOption

Description	Count the total number of options the given web element has
Object	TestObject object – represent a web element
Input	N/a
Output	int number – the total number of options

Example

Scenario – There is a select element with the following html code:

```
<select id="selenium_commands" class="input-
xlarge" name="selenium_commands" multiple="multiple">
        <option value="browser_commands">Browser
        Commands</option>
        <option value="navigation_commands">Navigation
        Commands</option>
        <option value="switch_commands">Switch
        Commands</option>
        <option value="wait_commands">Wait
        Commands</option>
        <option value="webelement_commands">WebElement
        Commands</option>
</select>
```

Users want to count the total number of options of the web element. The output (5) is stored into variable var_number.

Item	Object	Input	Output
getNumberOfTotalOption	sl_Commands		var_number

5.7 selectAllOption

Description	Select all options
Object	TestObject object – represent a web element
Input	N/a
Output	N/a

Example

Scenario – There is a select element with the following html code:

```
<select id="selenium_commands" class="input-
xlarge" name="selenium_commands" multiple="multiple">
        <option value="browser_commands">Browser
        Commands</option>
        <option value="navigation_commands">Navigation
        Commands</option>
        <option value="switch_commands">Switch
        Commands</option>
        <option value="wait_commands">Wait
        Commands</option>
        <option value="webelement_commands">WebElement
        Commands</option>
</select>
```

Users want to select all options.

Item	Object	Input	Output
selectAllOption	sl_Commands		

5.8 selectOptionByIndex

Description	Select the options at the given indices. Index starts from 0.
Object	TestObject object – represent a web element
Input	1. String index_range – index range of the options to select. Index starts from 0. Example: 2 - index 2 2,3 - index 2 and 3 2-5 - index 2 to 5 (2, 3, 4, 5)
Output	N/a

Example

Scenario – There is a select element with the following html code:

```
<select id="selenium_commands" class="input-
xlarge" name="selenium_commands" multiple="multiple">
        <option value="browser_commands">Browser
        Commands</option>
        <option value="navigation_commands">Navigation
        Commands</option>
        <option value="switch_commands">Switch
        Commands</option>
        <option value="wait_commands">Wait
        Commands</option>
        <option value="webelement_commands">WebElement
        Commands</option>
</select>
```

Users want to select "Navigation Commands" option (index 1).

Item	Object	Input	Output
selectOptionByIndex	sl_Commands	"1"	

Users want to select all options from index 0 to index 3.

Item	Object	Input	Output
selectOptionByIndex	sl_Commands	"0-3"	

Users want to select "Browser Commands" option (index 0) and "Wait Commands" option (index 3).

Item	Object	Input	Output
selectOptionByIndex	sl_Commands	"0,3"	

5.9 selectOptionByLabel

Description	Select the options with the given label (displayed text)
Object	TestObject object – represent a web element
Input	1. String label – displayed text of the options to select 2. boolean isRegex – **true** if label is regular expression, **false** if not
Output	N/a

Example

Scenario – There is a select element with the following html code:

```
<select id="selenium_commands" class="input-
xlarge" name="selenium_commands" multiple="multiple">
        <option value="browser_commands">Browser
        Commands</option>
        <option value="navigation_commands">Navigation
        Commands</option>
        <option value="switch_commands">Switch
        Commands</option>
        <option value="wait_commands">Wait
        Commands</option>
        <option value="webelement_commands">WebElement
        Commands</option>
</select>
```

Users want to select "Browser Commands" option and "Wait Commands" option.

Item	Object	Input	Output
selectOptionByLabel	sl_Commands	"Browser Commands", false	

Users want to select "Wait Commands" option and "WebElement Commands" option.

Item	Object	Input	Output
selectOptionByLabel	sl_Commands	"^W(ai|ebElemen)t Commands$", true	

5.10 selectOptionByValue

Description	Select the options with the given value
Object	TestObject object – represent a web element
Input	1. String value – value of the options to select 2. boolean isRegex – **true** if value is regular expression, **false** if not
Output	N/a

Example

Scenario – There is a select element with the following html code:

```
<select id="selenium_commands" class="input-
xlarge" name="selenium_commands" multiple="multiple">
        <option value="browser_commands">Browser
        Commands</option>
        <option value="navigation_commands">Navigation
        Commands</option>
        <option value="switch_commands">Switch
        Commands</option>
        <option value="wait_commands">Wait
        Commands</option>
        <option value="webelement_commands">WebElement
        Commands</option>
</select>
```

Users want to select "Browser Commands" option (value "browser_commands") and "Wait Commands" option (value "wait_commands").

Item	Object	Input	Output
selectOptionByValue	sl_Commands	"browser_commands", false	

Users want to select "Wait Commands" option (value "wait_commands") and "WebElement Commands" option (value "webelement_commands").

Item	Object	Input	Output
selectOptionByValue	sl_Commands	"^w(ai|ebelemen)t_commands", true	

5.11 verifyAnyOptionNotSelected (not implemented yet)

Description	Verify if no option is selected
Object	TestObject object – represent a web element
Input	1. int timeout – system will wait at most timeout (seconds) to return result
Output	result – **true** if no option is selected; otherwise, **false**

Example

Scenario – There is a select element with the following html code:

```
<select id="selenium_commands" class="input-
xlarge" name="selenium_commands" multiple="multiple">
        <option value="browser_commands">Browser
        Commands</option>
        <option value="navigation_commands">Navigation
        Commands</option>
        <option value="switch_commands">Switch
        Commands</option>
        <option value="wait_commands">Wait
        Commands</option>
        <option value="webelement_commands">WebElement
        Commands</option>
</select>
```

Users want to verify if no option is selected. The output is stored into variable var_result.

Item	Object	Input	Output
verifyAnyOptionNotSelected	sl_Commands	5	var_result

5.12 verifyAnyOptionSelected (not implemented yet)

Description	Verify if any option is selected
Object	TestObject object – represent a web element
Input	1. int timeout – system will wait at most timeout (seconds) to return result
Output	boolean result – **true** if any option is selected; otherwise, **false**

Example

Scenario – There is a select element with the following html code:

```
<select id="selenium_commands" class="input-
xlarge" name="selenium_commands" multiple="multiple">
        <option value="browser_commands">Browser
        Commands</option>
        <option value="navigation_commands">Navigation
        Commands</option>
        <option value="switch_commands">Switch
        Commands</option>
        <option value="wait_commands">Wait
        Commands</option>
        <option value="webelement_commands">WebElement
        Commands</option>
</select>
```

Users want to verify if any option is selected. The output is stored into variable var_result.

Item	Object	Input	Output
verifyAnyOptionSelected	sl_Commands	5	var_result

5.13 verifyOptionNotPresentByIndex (not implemented yet)

Description	Verify if the options at the given indices do **NOT** present. Index starts from 0.
Object	TestObject object – represent a web element
Input	1. String index_range – index range of options to verify if **NOT** presenting Example: 2 - index 2 2,3 - index 2 and 3 2-5 - index 2 to 5 (2, 3, 4, 5) 2. int timeout – system will wait at most timeout (seconds) to return result
Output	boolean result – **true** if all options at given indices do **NOT** present; otherwise, **false**

Example

Scenario 1 – There is a select element with the following html code:

```
<select id="selenium_commands" class="input-
xlarge" name="selenium_commands" multiple="multiple">
        <option value="browser_commands">Browser
        Commands</option>
        <option value="navigation_commands">Navigation
        Commands</option>
        <option value="switch_commands">Switch
        Commands</option>
        <option value="wait_commands">Wait
        Commands</option>
        <option value="webelement_commands">WebElement
        Commands</option>
</select>
```

Users want to verify if options at index 0 and index 2 do **NOT** present. The output (**false**) is stored into variable var_result.

Item	Object	Input	Output
verifyOptionNotPresentByIndex	sl_Commands	"0,2", 5	var_result

Scenario 2 – With the above select element, users want to verify if option at index 4 and index 5 do **NOT** present. The output (**false**) is stored into variable var_result.

Item	Object	Input	Output
verifyOptionNotPresentByIndex	sl_Commands	"4,5", 5	var_result

Scenario 3 – With the above select element, users want to verify if option at index 5 and index 6 do **NOT** present. The output (**true**) is stored into variable var_result.

Item	Object	Input	Output

verifyOptionNotPresentByIndex	sl_Commands	"5,6", 5	var_result

5.14 verifyOptionNotPresentByLabel

Description	Verify if the option with the given label (displayed text) does **NOT** present
Object	TestObject object – represent a web element
Input	1. String label – displayed text of the option to verify if **NOT** presenting 2. boolean isRegex – **true** if label is regular expression, **false** if not 3. int timeout – system will wait at most timeout (seconds) to return result
Output	boolean result – **true** if option with the given label does **NOT** present; otherwise, **false**

Example

Scenario 1 – There is a select element with the following html code:

```
<select id="selenium_commands" class="input-
xlarge" name="selenium_commands" multiple="multiple">
        <option value="browser_commands">Browser
        Commands</option>
        <option value="navigation_commands">Navigation
        Commands</option>
        <option value="switch_commands">Switch
        Commands</option>
        <option value="wait_commands">Wait
        Commands</option>
        <option value="webelement_commands">WebElement
        Commands</option>
</select>
```

Users want to verify if "Frame Commands" option does **NOT** present. The output is stored into variable var_result.

Item	Object	Input	Output
verifyOptionNotPresentByLabel	sl_Commands	"Frame Commands", false, 5	var_result

Scenario 2 – With the above select element, users want to verify if "Wait Commands" option and "WebElement Commands" option do **NOT** present. The output is stored into variable var_result.

Item	Object	Input	Output
verifyOptionNotPresentByLabel	sl_Commands	"^W(ai|ebelemn)t Commands", true, 5	var_result

5.15 verifyOptionNotPresentByValue

Description	Verify if the option with the given value does **NOT** present
Object	TestObject object – represent a web element
Input	1. String value – value of the option to verify if **NOT** presenting 2. boolean isRegex – **true** if value is regular expression, **false** if not 3. int timeout – system will wait at most timeout (seconds) to return result
Output	boolean result – **true** if option with the given value does **NOT** present; otherwise, **false**

Example

Scenario 1 – There is a select element with the following html code:

```
<select id="selenium_commands" class="input-
xlarge" name="selenium_commands" multiple="multiple">
        <option value="browser_commands">Browser
        Commands</option>
        <option value="navigation_commands">Navigation
        Commands</option>
        <option value="switch_commands">Switch
        Commands</option>
        <option value="wait_commands">Wait
        Commands</option>
        <option value="webelement_commands">WebElement
        Commands</option>
</select>
```

Users want to verify if option with value "browser_commands" does **NOT** present. The output is stored into variable var_result.

Item	Object	Input	Output
verifyOptionNotPresentByValue	sl_Commands	"browser_commands", false, 5	var_result

Users want to verify if options with values "table_commands" and "frame_commands" do **NOT** present. The output is stored into variable var_result.

Item	Object	Input	Output
verifyOptionNotPresentByValue	sl_Commands	"^(table|frame)_commands$", true, 5	var_result

5.16 verifyOptionNotSelectedByIndex

Description	Verify if the options at the given indices are **NOT** selected. Index starts from 0.
Object	TestObject object – represent a web element
Input	1. String index_range – index range of the options to be verify if **NOT** being selected Example: 2 - index 2

	2,3 - index 2 and 3 2-5 - index 2 to 5 (2, 3, 4, 5) 2. int timeout – system will wait at most timeout (seconds) to return result
Output	boolean result – **true** if all options in the given index range are **NOT** selected; otherwise, **false**

Example

Scenario – There is a select element with the following html code:

```
<select id="selenium_commands" class="input-
xlarge" name="selenium_commands" multiple="multiple">
        <option value="browser_commands">Browser
        Commands</option>
        <option value="navigation_commands">Navigation
        Commands</option>
        <option value="switch_commands">Switch
        Commands</option>
        <option value="wait_commands">Wait
        Commands</option>
        <option value="webelement_commands">WebElement
        Commands</option>
</select>
```

Users want to verify if "Navigation Commands" option (index 1) **NOT** selected. The output is stored into variable var_result.

Item	Object	Input	Output
verifyOptionNotSelectedByIndex	sl_Commands	"1", 5	var_result

Users want to verify if "Browser Commands" option (index 0) and "Wait Commands" option (index 3) are **NOT** selected. The output is stored into variable var_result.

Item	Object	Input	Output
verifyOptionNotSelectedByIndex	sl_Commands	"0,3", 5	var_result

Users want to verify if all options from index 0 to index 3 are **NOT** selected. The output is stored into variable var_result.

Item	Object	Input	Output
verifyOptionNotSelectedByIndex	sl_Commands	"0-3", 5	var_result

5.17 verifyOptionNotSelectedByLabel

Description	Verify if the option with the given label (displayed text) is **NOT** selected
Object	TestObject object – represent a web element
Input	1. String label – displayed text of the option to verify if **NOT** being selected 2. boolean isRegex – **true** if label is regular expression, **false** if not 3. int timeout – system will wait at most timeout (seconds) to

	return result
Output	boolean result – **true** if option with the given label is **NOT** selected; otherwise, **false**

Example

Scenario – There is a select element with the following html code:

```
<select id="selenium_commands" class="input-
xlarge" name="selenium_commands" multiple="multiple">
        <option value="browser_commands">Browser
        Commands</option>
        <option value="navigation_commands">Navigation
        Commands</option>
        <option value="switch_commands">Switch
        Commands</option>
        <option value="wait_commands">Wait
        Commands</option>
        <option value="webelement_commands">WebElement
        Commands</option>
</select>
```

Users want to verify if "Browser Commands" option is **NOT** selected. The output is stored into variable var_result.

Item	Object	Input	Output
verifyOptionNotSelectedByLabel	sl_Commands	"Browser Commands", false, 5	var_result

Users want to verify if "Wait Commands" and "WebElement Commands" options are **NOT** selected. The output is stored into variable var_result.

Item	Object	Input	Output
verifyOptionNotSelectedByLabel	sl_Commands	"^W(ai|ebElemen)t Commands", true, 5	var_result

5.18 verifyOptionNotSelectedByValue

Description	Verify if the option with the given value is **NOT** selected
Object	TestObject object – represent a web element
Input	1. String value – value of the option to verify if **NOT** being selected 2. boolean isRegex – **true** if value is regular expression, **false** if not 3. int timeout – system will wait at most timeout (seconds) to return result
Output	boolean result – **true** if the option with the given value is **NOT** selected; otherwise, **false**

Example

Scenario – There is a select element with the following html code:

```
<select id="selenium_commands" class="input-
xlarge" name="selenium_commands" multiple="multiple">
        <option value="browser_commands">Browser
        Commands</option>
        <option value="navigation_commands">Navigation
        Commands</option>
        <option value="switch_commands">Switch
        Commands</option>
        <option value="wait_commands">Wait
        Commands</option>
        <option value="webelement_commands">WebElement
        Commands</option>
</select>
```

Users want to verify if option with value "browser_commands" is **NOT** selected. The output is stored into variable var_result.

Item	Object	Input	Output
verifyOptionNotSelectedByValue	sl_Commands	"browser_commands", false, 5	var_result

Users want to verify if options with value "wait_commands" and "webelement_commands" are **NOT** selected. The output is stored into variable var_result.

Item	Object	Input	Output
verifyOptionNotSelectedByValue	sl_Commands	"^w(ai|ebelemen)t_commands", true, 5	var_result

5.19 verifyOptionPresentByIndex (not implemented yet)

Description	Verify if the options at the given indices present. Index starts from 0.
Object	TestObject object – represent a web element
Input	1. String index_range – index range of options to verify if presenting Example: 2 - index 2 2,3 - index 2 and 3 2-5 - index 2 to 5 (2, 3, 4, 5) 2. int timeout – system will wait at most timeout (seconds) to return result
Output	boolean result – **true** if all options at the given indices present; otherwise, **false**

Example

Scenario 1 – There is a select element with the following html code:

```
<select id="selenium_commands" class="input-
xlarge" name="selenium_commands" multiple="multiple">
        <option value="browser_commands">Browser
        Commands</option>
        <option value="navigation_commands">Navigation
        Commands</option>
        <option value="switch_commands">Switch
        Commands</option>
        <option value="wait_commands">Wait
        Commands</option>
        <option value="webelement_commands">WebElement
        Commands</option>
</select>
```

Users want to verify if options at index 0 and index 3 present. The output (**true**) is stored into variable var_result.

Item	Object	Input	Output
verifyOptionPresentByIndex	sl_Commands	"0,3", 5	var_result

Scenario 2 – With the above select element, users want to verify if option at index 0 and index 5 present. The output (**false**) is stored into variable var_result.

Item	Object	Input	Output
verifyOptionPresentByIndex	sl_Commands	"0,5", 5	var_result

5.20 verifyOptionPresentByLabel

Description	Verify if the option with the given label (displayed text) presents
Object	TestObject object – represent a web element
Input	1. String label – displayed text of the option to verify if presenting 2. boolean isRegex – **true** if label is regular expression, **false** if not 3. int timeout – system will wait at most timeout (seconds) to return result
Output	result – **true** if the option with the given label presents; otherwise, **false**

Example

Scenario 1 – There is a select element with the following html code:

```
<select id="selenium_commands" class="input-
xlarge" name="selenium_commands" multiple="multiple">
        <option value="browser_commands">Browser
        Commands</option>
        <option value="navigation_commands">Navigation
        Commands</option>
        <option value="switch_commands">Switch
        Commands</option>
```

```
<option value="wait_commands">Wait
Commands</option>
<option value="webelement_commands">WebElement
Commands</option>
</select>
```

Users want to verify if "Browser Commands" option presents. The output is stored into variable var_result.

Item	Object	Input	Output
verifyOptionPresentByLabel	sl_Commands	"Browser Commands", false, 5	var_result

Scenario 2 – With the above select element, users want to verify if "Wait Commands" option and "WebElement Commands" option present. The output is stored into variable var_result.

Item	Object	Input	Output
verifyOptionPresentByLabel	sl_Commands	"^W(ai|ebelemen)t Commands$", true, 5	var_result

5.21 verifyOptionPresentByValue

Description	Verify if the option with the given value presents
Object	TestObject object – represent a web element
Input	1. String value – value of the option to verify if presenting 2. boolean isRegex – **true** if value is regular expression, **false** if not 3. int timeout – system will wait at most timeout (seconds) to return result
Output	boolean result – **true** if the option with the given value presents; otherwise, **false**

Example

Scenario 1 – There is a select element with the following html code:

```
<select id="selenium_commands" class="input-
xlarge" name="selenium_commands" multiple="multiple">
        <option value="browser_commands">Browser
        Commands</option>
        <option value="navigation_commands">Navigation
        Commands</option>
        <option value="switch_commands">Switch
        Commands</option>
        <option value="wait_commands">Wait
        Commands</option>
        <option value="webelement_commands">WebElement
        Commands</option>
</select>
```

Users want to verify if options with value "browser_commands" presents. The output is stored into variable var_result.

Item	Object	Input	Output
verifyOptionPresentByValue	sl_Commands	"browser_commands", false, 5	var_result

Scenario 2 – With the above select element, users want to verify if option with value "wait_commands" and "webelement_commands" present. The output is stored into variable var_result.

Item	Object	Input	Output
verifyOptionPresentByValue	sl_Commands	"^w(ai|ebelemen)t_commands $", true, 5	var_result

5.22 verifyOptionSelectedByIndex

Description	Verify if the options in the given index range are selected. Index starts from 0.
Object	TestObject object – represent a web element
Input	1. String index_range – index range of the options to verify if being selected Example: 2 - index 2 2,3 - index 2 and 3 2-5 - index 2 to 5 (2, 3, 4, 5) 2. int timeout – system will wait at most timeout (seconds) to return result
Output	boolean result – **true** if all options in the given index range are selected; otherwise, **false**

Example

Scenario – There is a select element with the following html code:

```
<select id="selenium_commands" class="input-
xlarge" name="selenium_commands" multiple="multiple">
        <option value="browser_commands">Browser
        Commands</option>
        <option value="navigation_commands">Navigation
        Commands</option>
        <option value="switch_commands">Switch
        Commands</option>
        <option value="wait_commands">Wait
        Commands</option>
        <option value="webelement_commands">WebElement
        Commands</option>
</select>
```

Users want to verify if "Navigation Commands" option (index 1) is selected. The output is stored into variable var_result.

Item	Object	Input	Output

Item	Object	Input	Output
verifyOptionSelectedByIndex	sl_Commands	"1", 5	var_result

Users want to verify if "Browser Commands" option (index 0) and "Wait Commands" option (index 3) are selected. The output is stored into variable var_result.

Item	Object	Input	Output
verifyOptionSelectedByIndex	sl_Commands	"0,3", 5	var_result

Users want to verify all options from index 0 to index 3 are selected. The output is stored into variable var_result.

Item	Object	Input	Output
verifyOptionSelectedByIndex	sl_Commands	"0-3", 5	var_result

5.23 verifyOptionSelectedByLabel

Description	Verify if the option with the given label (displayed text) is selected
Object	TestObject object – represent a web element
Input	1. String label – displayed text of the option to verify if being selected 2. boolean isRegex – **true** if label is regular expression, **false** if not 3. int timeout – system will wait at most timeout (seconds) to return result
Output	boolean result – **true** if the option with the given label is selected; otherwise, **false**

Example

Scenario – There is a select element with the following html code:

```
<select id="selenium_commands" class="input-
xlarge" name="selenium_commands" multiple="multiple">
        <option value="browser_commands">Browser
        Commands</option>
        <option value="navigation_commands">Navigation
        Commands</option>
        <option value="switch_commands">Switch
        Commands</option>
        <option value="wait_commands">Wait
        Commands</option>
        <option value="webelement_commands">WebElement
        Commands</option>
</select>
```

Users want to verify if "Browser Commands" option is selected. The output is stored into variable var_result.

Item	Object	Input	Output
verifyOptionSelectedByLabel	sl_Commands	"Browser Commands", false, 5	var_result

Users want to verify if "Wait Commands" option and "WebElement Commands" option are selected. The output is stored into variable var_result.

Item	Object	Input	Output
verifyOptionSelectedByLabel	sl_Commands	"^W(ai|ebElemen)t Commands$", true, 5	var_result

5.24 verifyOptionSelectedByValue

Description	Verify if the option with the given value is selected
Object	TestObject object – represent a web element
Input	1. String value – value of the option to verify if being selected 2. boolean isRegex – **true** if value is regular expression, **false** if not 3. int timeout – system will wait at most timeout (seconds) to return result
Output	boolean result – **true** if the option with the given value is selected; otherwise, **false**

Example

Scenario – There is a select element with the following html code:

```
<select id="selenium_commands" class="input-
xlarge" name="selenium_commands" multiple="multiple">
        <option value="browser_commands">Browser
        Commands</option>
        <option value="navigation_commands">Navigation
        Commands</option>
        <option value="switch_commands">Switch
        Commands</option>
        <option value="wait_commands">Wait
        Commands</option>
        <option value="webelement_commands">WebElement
        Commands</option>
</select>
```

Users want to verify if option with value "browser_commands" is selected. The output is stored into variable var_result.

Item	Object	Input	Output
verifyOptionSelectedByValue	sl_Commands	"browser_commands", false, 5	var_result

Users want to verify if options with value "wait_commands" and value "webelement_commands" are selected. The output is stored into variable var_result.

Item	Object	Input	Output
verifyOptionSelectedByValue	sl_Commands	"^w(ai|ebelemen)t_commands", true, 5	var_result

6. Utility Action

6.1 callTestCase

Description	Call another test case
Object	N/a
Input	1. tc_id – ID of the test case to be called
Output	N/a

Example **Scenario** – Users want to call "Login" test case. This test case resides inside folder "Common Test Cases".

Item	Object	Input	Output
callTestCase		"Test Cases/Common Test Cases/Login"	

6.2 concatenate

Description	Concatenate all strings end-to-end into one string
Object	N/a
Input	1. String[] string_list – list of strings to concatenate
Output	String result – the end-to-end joined string of the input

Example **Scenario** – Users want to concatenate two strings "KMS" and " Technology" into one string "KMS Technology". The output is stored into variable var_result.

Item	Object	Input	Output
concatenate		["KMS", " Technology"]	var_result

6.3 delay

Description	Delay the execution for the specified seconds
Object	N/a
Input	1. int seconds – the amount of seconds to delay
Output	N/a

Example **Scenario** – Users want to delay the execution for 10 seconds.

Item	Object	Input	Output
delay		10	

6.4 modifyObjectProperty

Description	Modify existing property or add new property to test object. Use when test object has attributes changing in runtime. This keyword does not modify the object saved in Object Repository, instead, it creates another test object, modifies and returns this test object. Hence, users must use a variable to store the returned object.
Object	TestObject – represent a web element
Input	1. String propertyName - name of the property, for example, xpath, id, name, ... If the property already exists in the object, the keyword will modify its related artifacts; if not, the keyword will add new property. 2. String matchCondition - condition to match property name with property value, for example, equals, not equals, ... In case the property already exists, input **null** to this argument to keep the old value of match condition. 3. String propertyValue - value of the property. In case the property already exists, input **null** to this argument to keep the old property value. 4. boolean isActive - **true** if the property is checked (used to find the test object); otherwise, **false.** In case the property already exists, input **null** to this argument to keep the old value.
Output	TestObject – the modified object

Example **Scenario** – Users want to modify property of an object and click on this modified object.

Item	Object	Input	Output
modifyObjectProperty	btn_Temp	"xpath"; "equals"; "//*[@type='button']"; true	var_ModifiedBtn
click	var_ModifiedBtn		

6.5 removeObjectProperty

Description	Remove existing property of test object. Use when test object has attributes changing in runtime. This keyword does not modify the object saved in Object Repository, instead, it creates another test object, modify and return this test object. Hence, users must use a variable to store the returned object.
Object	TestObject – represent a web element
Input	1. String propertyName - name of the property, for example, xpath, id, name, ...

Output	TestObject – the modified object

Example **Scenario** – Users want to remove property of an object and click on this modified object.

Item	Object	Input	Output
removeObjectProperty	btn_Temp	"class"	var_ModifiedBtn
click	var_ModifiedBtn		

6.6 takeScreenshot

Description	Take screenshot of the browser
Object	N/a
Input	N/a
Output	N/a

Example **Scenario** – Users want to take a screenshot of the browser at current state

Item	Object	Input	Output
takeScreenshot			

6.7 verifyEqual

Description	Verify if two numbers are equal
Object	N/a
Input	1. number1 – first number 2. number2 – second number
Output	boolean result – **true** if number1 equals number2; otherwise, **false**

Example **Scenario 1** – Users want to verify if number 10 equals number 10, the result (**true**) is stored into variable var_result.

Item	Object	Input	Output
verifyEqual		10; 10	var_result

Scenario 2 – Users want to verify if number 10 equals number 11, the result (**false**) is stored into variable var_result.

Item	Object	Input	Output
verifyEqual		10; 11	var_result

6.8 verifyGreaterThan

Description	Verify if the first number is greater than the second number

Object	N/a
Input	1. number1 – first number 2. number2 – second number
Output	boolean result – **true** if number1 is greater than number2; otherwise, **false**

Example **Scenario 1** – Users want to verify if number 10 is greater than number 9, the result (**true**) is stored into variable var_result.

Item	Object	Input	Output
verifyGreaterThan		10; 9	var_result

Scenario 2 – Users want to verify if number 10 is greater than number 11, the result (**false**) is stored into variable var_result.

Item	Object	Input	Output
verifyGreaterThan		10; 11	var_result

Scenario 3 – Users want to verify if number 10 is greater than number 10, the result (**false**) is stored into variable var_result.

Item	Object	Input	Output
verifyGreaterThan		10; 10	var_result

6.9 verifyGreaterThanOrEqual

Description	Verify if the first number is greater than or equal the second number
Object	N/a
Input	1. number1 – first number 2. number2 – second number
Output	boolean result – **true** if number1 is greater than or equal number2; otherwise, **false**

Example **Scenario 1** – Users want to verify if number 10 is greater than or equal number 9, the result (**true**) is stored into variable var_result.

Item	Object	Input	Output
verifyGreaterThanOrEqual		10; 9	var_result

Scenario 2 – Users want to verify if number 10 is greater than or equal number 10, the result (**true**) is stored into variable var_result.

Item	Object	Input	Output
verifyGreaterThanOrEqual		10; 10	var_result

Scenario 3 – Users want to verify if number 10 is greater than or equal number 11, the result (**false**) is stored into variable var_result.

Item	Object	Input	Output
verifyGreaterThanOrEqual		10; 11	var_result

6.10 verifyLessThan

Description	Verify if the first number is less than the second number
Object	N/a
Input	1. number1 – first number 2. number2 – second number
Output	boolean result – **true** if number1 is less than number2; otherwise, **false**

Example

Scenario 1 – Users want to verify if number 10 is less than number 9, the result (**false**) is stored into variable var_result.

Item	Object	Input	Output
verifyLessThan		10; 9	var_result

Scenario 2 – Users want to verify if number 10 is less than number 10, the result (**false**) is stored into variable var_result.

Item	Object	Input	Output
verifyLessThan		10; 10	var_result

Scenario 3 – Users want to verify if number 10 is less than number 11, the result (**true**) is stored into variable var_result.

Item	Object	Input	Output
verifyLessThan		10; 11	var_result

6.11 verifyLessThanOrEqual

Description	Verify if the first number is less than or equal the second number
Object	N/a
Input	1. number1 – first number 2. number2 – second number
Output	boolean result – **true** if number1 is less than or equal number2; otherwise, **false**

Example

Scenario 1 – Users want to verify if number 10 is less than or equal number 9, the result (**false**) is stored into variable var_result.

Item	Object	Input	Output
verifyLessThanOrEqual		10; 9	var_result

Scenario 2 – Users want to verify if number 10 is less than or equal number 10, the result (**true**) is stored into variable var_result.

Item	Object	Input	Output
verifyLessThanOrEqual		10; 10	var_result

Scenario 3 – Users want to verify if number 10 is less than or equal number 11, the result (**true**) is stored into variable var_result.

Item	Object	Input	Output
verifyLessThanOrEqual		10; 11	var_result

6.12 verifyMatch

Description	Verify if two strings match each other, the second string can be a regular expression
Object	N/a
Input	1. String string1 – first string 2. String string2 – second string 3. boolean isRegex – **true** if string2 is regular expression; otherwise, **false**
Output	boolean result – **true** if string1 matches string2; otherwise, **false**

Example

Scenario 1 – Users want to verify if string "KMS Technology" matches string "KMS Tech.*", not using regular expression. The output (**false**) is stored into variable var_result.

Item	Object	Input	Output
verifyMatch		"KMS Technology"; "KMS Tech.*";false	var_result

Scenario 2 – Users want to verify if string "KMS Technology" matches string "KMS Tech.*", using regular expression. The output (**true**) is stored into variable var_result.

Item	Object	Input	Output
verifyMatch		"KMS Technology"; "KMS Tech.*"; true	var_result

6.13 verifyNotEqual

Description	Verify if two numbers are **NOT** equal
Object	N/a
Input	1. number1 – first number 2. number2 – second number

Output	boolean result – **true** if number1 is **NOT** equal number2; otherwise, **false**

Example

Scenario 1 – Users want to verify if number 10 is **NOT** equal number 10, the result (**false**) is stored into variable var_result.

Item	Object	Input	Output
verifyNotEqual		10; 10	var_result

Scenario 2 – Users want to verify if number 10 is **NOT** equal number 11, the result (**true**) is stored into variable var_result.

Item	Object	Input	Output
verifyNotEqual		10; 11	var_result

6.14 verifyNotMatch

Description	Verify if two strings do **NOT** match each other, the second string can be a regular expression
Object	N/a
Input	1. String string1 – first string 2. String string2 – second string 3. boolean isRegex – **true** if string2 is regular expression; otherwise, **false**
Output	result – **true** if string1 does **NOT** match string2; otherwise, **false**

Example

Scenario 1 – Users want to verify if string "KMS Technology" does not match "KMS Tech.*", not using regular expression. The output (**true**) is stored into variable var_result.

Item	Object	Input	Output
verifyNotMatch		"KMS Technology"; "KMS Tech.*";false	var_result

Scenario 2 – Users want to verify if string "KMS Technology" does not match "KMS Tech.*", using regular expression. The output (**false**) is stored into variable var_result.

Item	Object	Input	Output
verifyNotMatch		"KMS Technology"; "KMS Tech.*"; true	var_result

7. Web Element Action

7.1 check

Description	Check a toggle-button (check-box/radio-button)
Object	TestObject object – represent a web element
Input	N/a
Output	N/a

Example **Scenario** – Users want to check the check-box "chkbx_KeepLogin".

Item	Object	Input	Output
check	chkbx_KeepLogin		

7.2 focus

Description	Move the focus to the specified element; for example, if the element is an input field, move the cursor to that field
Object	TestObject object – represent a web element
Input	N/a
Output	N/a

Example **Scenario** – Users want to focus on text-box "txtbx_Username".

Item	Object	Input	Output
focus	txtbx_Username		

7.3 getAttribute

Description	Get attribute value of a web element
Object	TestObject object – represent a web element
Input	1. String attribute – name of the attribute
Output	String value – value of the attribute

Example **Scenario** – Users want to get value of the attribute "type" of the text-box "txtbx_Username" which has the following html code:

<input type="text" value="" name="user">. The output ("text") is stored into variable var_output.

Item	Object	Input	Output
getAttribute	txtbx_Username	"type"	var_output

7.4 getText

Description	Get the visible (i.e. not hidden by CSS) innerText of the web element, including sub-elements, without any leading or trailing

	whitespace.
Object	TestObject object – represent a web element
Input	N/a
Output	String text – innerText of the web element

Example **Scenario** – Users want to get text value of the link "lnk_Register" which has the following html code:

Register. The output ("Register") is stored into variable var_output.

Item	Object	Input	Output
getText	lnk_Register		var_output

7.5 setText

Description	Set the value of an input field, as though you type it in. It also clears the previous value of the input field
Object	TestObject object – represent a web element
Input	1. String value – the text to type
Output	N/a

Example **Scenario** – The input filed "txtbx_Username" already has text "user1". Users want to set text value to "user2". After the action is executed, input field will have text "user2", not "user1user2".

Item	Object	Input	Output
setText	txtbx_Username	"user2"	

7.6 uncheck

Description	Uncheck a toggle-button (check-box/radio-button)
Object	TestObject object – represent a web element
Input	N/a
Output	N/a

Example: **Scenario** – Users want to uncheck the check-box "chkbx_KeepLogin".

Item	Object	Input	Output
uncheck	chkbx_KeepLogin		

7.7 verifyElementChecked

Description	Verify if the given web element is checked
Object	TestObject object – represent a web element
Input	1. int timeout – system will wait at most timeout (seconds) to return result
Output	boolean result – **true** if the element is checked; otherwise, **false**

Example **Scenario** – Users want to verify if the check-box "chkbx_KeepLogin" is checked. The output is stored into variable var_result.

Item	Object	Input	Output
verifyElementChecked	chkbx_KeepLogin	5	var_result

7.8 verifyElementEnabled (not implemented yet)

Description	Verify if the given web element is enabled
Object	TestObject object – represent a web element
Input	1. int timeout – system will wait at most timeout (seconds) to return result
Output	boolean result – **true** if the element is enabled; otherwise, **false**

Example **Scenario** – Users want to verify if the check-box "chkbx_KeepLogin" is enabled. The output is stored into variable var_result.

Item	Object	Input	Output
verifyElementEnabled	chkbx_KeepLogin	5	var_result

7.9 verifyElementNotChecked

Description	Verify if the given web element is **NOT** checked
Object	TestObject object – represent a web element
Input	1. int timeout – system will wait at most timeout (seconds) to return result
Output	boolean result – **true** if the element is **NOT** checked; otherwise, **false**

Example **Scenario** – Users want to verify if the check-box "chkbx_KeepLogin" is **NOT** checked. The output is stored into variable var_result.

Item	Object	Input	Output
verifyElementNotChecked	chkbx_KeepLogin	5	var_result

7.10 verifyElementNotEnabled (not implemented yet)

Description	Verify if the given web element is NOT enabled
Object	TestObject object – represent a web element
Input	1. int timeout – system will wait at most timeout (seconds) to return result
Output	result – **true** if the element is NOT enabled; otherwise, **false**

Example **Scenario** – Users want to verify if the check-box "chkbx_KeepLogin" is NOT enabled. The output is stored into variable var_result.

Item	Object	Input	Output
verifyElementNotEnabled	chkbx_KeepLogin	5	var_result

7.11 verifyElementNotPresent

Description	Verify if the given web element does **NOT** present.
Object	TestObject object – represent a web element
Input	1. int timeout – system will wait at most timeout (seconds) to return result
Output	boolean result – **true** if the element does **NOT** present; otherwise, **false**

Example **Scenario** – Users want to verify if the check-box "chkbx_KeepLogin" does **NOT** present. The output is stored into variable var_result.

Item	Object	Input	Output
verifyElementNotPresent	chkbx_KeepLogin	5	var_result

7.12 verifyElementNotVisible (not implemented yet)

Description	Verify if the given web element is **NOT** visible
Object	TestObject object – represent a web element
Input	1. int timeout – system will wait at most timeout (seconds) to return result
Output	boolean result – **true** if the element is **NOT** visible; otherwise, **false**

Example **Scenario** – Users want to verify if the check-box "chkbx_KeepLogin" is **NOT** visible. The output is stored into variable var_result.

Item	Object	Input	Output

verifyElementNotVisible	chkbx_KeepLogin	5	var_result

7.13 verifyElementPresent

Description	Verify if the given web element presents
Object	TestObject object – represent a web element
Input	1. int timeout – system will wait at most timeout (seconds) to return result
Output	boolean result – **true** if the element presents; otherwise, **false**

Example **Scenario** – Users want to verify if the check-box "chkbx_KeepLogin" presents. The output is stored into variable var_result.

Item	Object	Input	Output
verifyElementPresent	chkbx_KeepLogin	5	var_result

7.14 waitForElementClickable

Description	Wait until the given web element is clickable within timeout.
Object	TestObject object – represent a web element
Input	1. int timeout – how many seconds to wait (maximum)
Output	boolean result – **true** if the element is clickable; otherwise, **false**

Example **Scenario** – Users want to wait until the check-box "chkbx_KeepLogin" is clickable within 5 seconds. The output is stored into variable var_result.

Item	Object	Input	Output
waitForElementClickable	chkbx_KeepLogin	5	var_result

7.15 waitForElementNotPresent

Description	Wait for the given element to **NOT** present (disappear) within the given time in second unit
Object	TestObject object – represent a web element
Input	1. int timeout – system will wait at most timeout (seconds) to return result
Output	N/a

Example **Scenario** – Users want to wait for the check-box "chkbx_KeepLogin" to **NOT** present (disappear) within 10 seconds.

Item	Object	Input	Output
waitForElementNotPresent	chkbx_KeepLogin	10	

7.16 waitForElementPresent

Description	Wait for the given element to present (appear) within the given time in second unit
Object	TestObject object – represent a web element
Input	1. int timeout – system will wait at most timeout (seconds) to return result
Output	N/a

Example **Scenario** – Users want to wait for the check-box "chkbx_KeepLogin" to present (appear) within 10 seconds.

Item	Object	Input	Output
waitForElementPresent	chkbx_KeepLogin	10	

7.17 waitForElementVisible

Description	Wait until the given web element is visible within timeout.
Object	TestObject object – represent a web element
Input	2. int timeout – how many seconds to wait (maximum)
Output	boolean result – **true** if the element is visible; otherwise, **false**

Example **Scenario** – Users want to wait until the check-box "chkbx_KeepLogin" is visible within 5 seconds. The output is stored into variable var_result.

Item	Object	Input	Output
waitForElementVisible	chkbx_KeepLogin	5	var_result