



# Katalon

---

## Common Validation

**27–28 Apr. 2019**

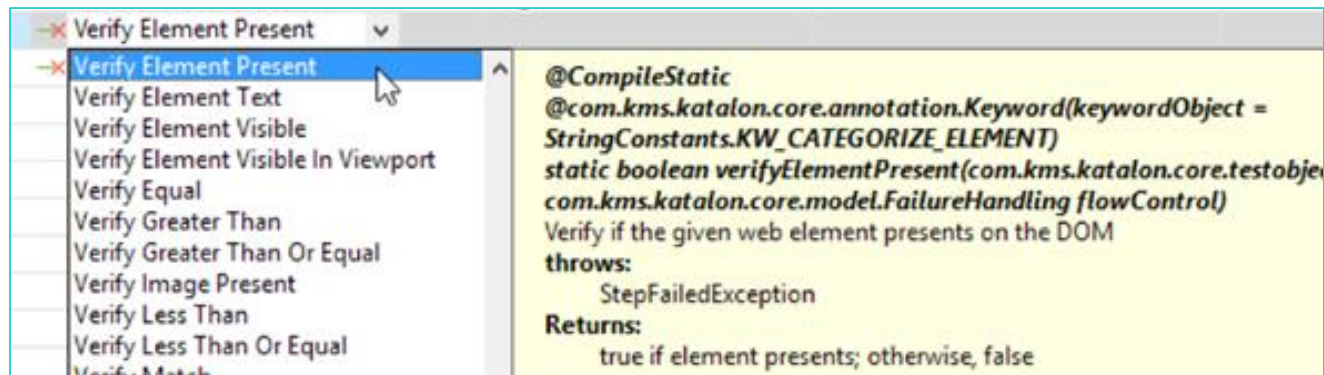
**4–5 May 2019 @IT MJU**

**Assoc. Rangsit Sirirangsi**

**[www.indythaitester.com](http://www.indythaitester.com)**

# Common Validation Statements

- Katalon Studio สนับสนุนการตรวจสอบความถูกต้องของส่วนประกอบต่าง ๆ ของ HTML ได้ 2 วิธี
  - วิธีการแรกเรียกใช้ built-in keywords ในการตรวจสอบสถานะของ TestObject ที่ถูกระบุ



- วิธีที่สองใช้ checkpoints ในการตรวจสอบกลุ่มของข้อมูลปัจจุบันที่ถูกระบุ เพื่อเปรียบเทียบความถูกต้องกับกลุ่มข้อมูลที่ถูกจัดเก็บไว้ก่อนหน้านี้

## Commonly used validation related keywords

---

- **Verify Element Present** : ตรวจสอบว่าส่วนประกอบของ HTML ถูกนำเสนออยู่ใน DOM หรือไม่
- **Verify Element Text** : ตรวจสอบว่าข้อความปรากฏบนส่วนประกอบของ HTML ที่ถูกระบุหรือไม่
- **Verify Element Visible** : ตรวจสอบว่าส่วนประกอบของ HTML ถูกนำเสนอภายใน DOM และแสดงผลบน UI หรือไม่
- **Verify Text Present** : ตรวจสอบว่าข้อความที่ถูกระบุปรากฏอยู่ในเพจหรือไม่
- **Verify Match** : ตรวจสอบว่าข้อความในรูปแบบของ String สองข้อความตรงกันหรือไม่

## Verify Element Present

- ตรวจสอบว่าส่วนประกอบของ HTML ถูกนำเสนอภายใน DOM หรือไม่  
คำสั่งหรือเมธอดนี้คืนค่า true ในกรณีที่เป็นอย่างจริงหรือ false ในกรณีที่เป็นอย่างเท็จ

**static boolean verifyElementPresent**(TestObject to, int timeout, FailureHandling flowControl)

actualText	String	Required	Represent the actual text.
to	TestObject	Required	Represent a web element.
timeout	int	Required	System will wait at most timeout (seconds) to return result

## Verify Element Text

---

- ตรวจสอบข้อความบนส่วนประกอบ HTML คืนค่า **true** ในกรณีที่มีข้อความปรากฏหรือมิฉะนั้นคืนค่า **false**

**static boolean verifyElementText**(TestObject to, String expectedText)

actualText	String	Required	Represent the actual text.
to	TestObject	Required	Represent a web element.
expectedText	String	Required	System will wait at most timeout (seconds) to return result

## Verify Text Present

- ตรวจสอบว่าข้อความที่ถูกระบุปรากฏอยู่ในเพจหรือไม่ คืนค่า True ในกรณีที่ปรากฏอยู่ในตำแหน่งใด ๆ ภายในเพจ Source หรือมิฉะนั้น False

**static boolean verifyTextPresent(String text, boolean isRegex)**

actualText	String	Required	Represent the actual text
text	String	Required	Text to be verified if presenting anywhere in the page source.
isRegex	Boolean	Required	Indicate whether the expected text is a regular expression.



# Script Mode

---

'Open browser and navigate to AUT'

```
WebUI.openBrowser(GlobalVariable.G_SiteURL)
```

'Input username'

```
WebUI.setText(findTestObject('Page_Login/txt_UserName'), Username)
```

'Input password'

```
WebUI.setText(findTestObject('Page_Login/txt_Password'), Password)
```

'Click on \'Login\' button'

```
WebUI.click(findTestObject('Page_Login/btn_Login'))
```

'Verify message after logging in'

```
WebUI.verifyTextPresent("Login successfully", false)
```

'Close browser'

```
WebUI.closeBrowser()
```

## Verify Match

- ตรวจสอบว่าข้อความในรูปของ String สองข้อความตรงกันหรือไม่ ในกรณีที่ตรงกันคืนค่า True มิฉะนั้นคืนค่า False
- ข้อความที่สองสามารถอยู่ในรูปของ Regular Expression

```
static boolean verifyMatch(String actualText, String expectedText,  
boolean isRegex)
```

actualText	String	Required	Represent the actual text
expectedText	String	Required	Represent the expected text (can be a regular expression).
isRegex	Boolean	Required	Indicate whether the expected text is a regular expression.



# Script Mode

---

'Open browser and navigate to AUT'

```
WebUI.openBrowser(GlobalVariable.G_SiteURL)
```

'Input username'

```
WebUI.setText(findTestObject('Page_Login/txt_UserName'), Username)
```

'Input password'

```
WebUI.setText(findTestObject('Page_Login/txt_Password'), Password)
```

'Click on \'Login\' button'

```
WebUI.click(findTestObject('Page_Login/btn_Login'))
```

'Get actual message after logging in'

```
actualMessage = WebUI.getText(findTestObject('lbl_message'))
```

'Verify message after logging in'

```
WebUI.verifyMatch(actualMessage, "Login successfully.", false)
```

'Close browser'

```
WebUI.closeBrowser()
```

## Demo : Gmail Login

---

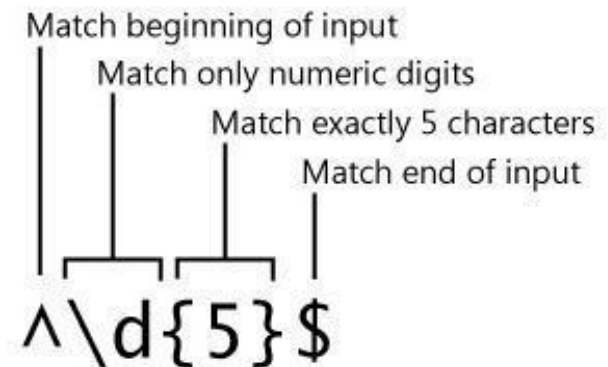
- สร้างสคริปต์ด้วยการ Add Property โดยใช้ ChroPath จากรายละเอียดของ Test Case ดังต่อไปนี้

TC #	Test Step	Test Data
1	Open Browser	Chrome
2	Go to the Application URL	<a href="http://www.gmail.com">www.gmail.com</a>
3	Enter Username	xxx
4	Click next	Yes
5	Enter Password	yyyy
6	Click next	Yes
7	Wait for Element Present	Inbox
8	Get Text from Inbox	noOfEmail
9	Verify Match	Inbox , ?, false

verifyMatchGmail

# What Regular Expressions ?

- เรียกสั้น ๆ ว่า RegEx เป็นแพตเทิร์นที่ถูกกำหนดขึ้นเพื่อค้นหาข้อความที่ต้องการ
- Regular Expression มาจากแนวคิดพื้นฐานที่เรียกว่า Finite Automata Theory
- ค้นหาจากการ "match" บางส่วนของข้อความของข้อมูลต้นทางที่เป็น Text หรืออยู่ในรูปของลำดับของไบต์หรืออักขระต่าง ๆ
- Engine ของ Regular Expression ที่ต่างกันอาจให้ผลลัพธ์ต่างกันด้วย



# RegEx : Literal Characters

---

*regular expression* → 

c	k	s
---	---	---

Test Tools ro**cks**.

↑  
*match*

Test Tools su**cks**.

↑  
*match*

Test Tools is okay.

# RegEx : Multiple Matches

---

- Regular Expression สามารถค้นหา String ได้มากกว่าหนึ่งตำแหน่ง

*regular expression* → 

a	p	p	l	e
---	---	---	---	---

Scrapple from the apple.

↑ match 1

↑ match 2

# Special Characters

---

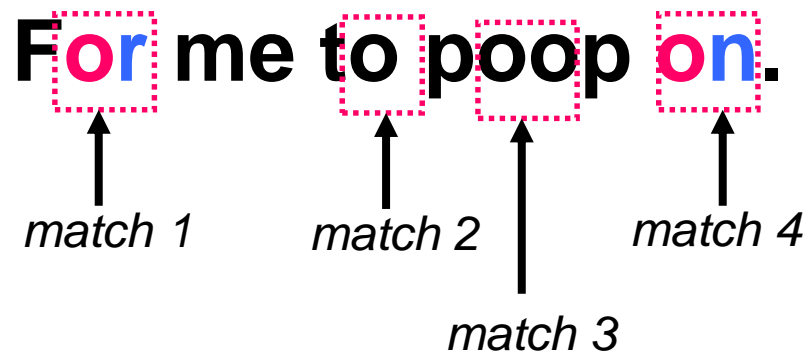
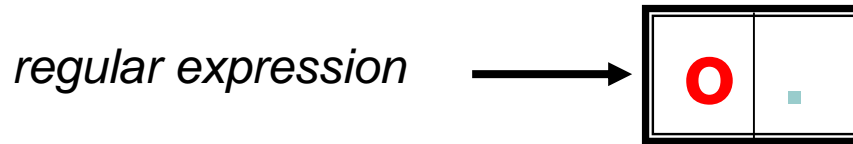
- บางครั้งเรียกว่า Metacharacters หรืออักขระที่มีความหมายพิเศษ ซึ่งประกอบไปด้วยอักขระจำนวน 11 ชนิด ได้แก่:

[ , \ , ^ , \$ , . , | , ? , \* , + , ( , )

- ในกรณีที่ต้องการ match  $1+1=2$  ต้องระบุแพตเทิร์นเป็น  $1\backslash+1=2$ 
  - เนื่องจากอักขระ + มีความหมายพิเศษ
- ส่วนอักขระอื่น ๆ ที่ไม่ใช่ Metacharacter ไม่ต้องใช้ escaped ด้วย backslash
- อักขระ backslash เมื่อใช้ร่วมกับ literal character จะมีความหมายพิเศษ เช่น
  - $\backslash d$  match ค่าตัวเลขหนึ่ง ๆ จาก 0 ถึง 9

# RegEx : Matching Any Character

- สัญลักษณ์ . สามารถใช้ในการค้นหาอักขระใด ๆ



# Character Classes or Character Sets

---

- “Character Class“ หรือบางครั้งเรียกว่า “Character Set“ ใช้สั่งให้ regex ค้นหาเฉพาะอักขระหนึ่ง ๆ จากกลุ่มของอักขระที่กำหนดไว้
- Character Classes เป็นการจัดกลุ่มอักขระต่าง ๆ ไว้ด้วยกัน เช่น เครื่องหมายก้ามปู :
  - `[aeiou]` ใช้ค้นหาอักขระใด ๆ จาก a, e, i, o, หรือ u
- Character Classes ใช้กำหนดค่าพิสัย
  - `[1-9]` ใช้ในความหมายเดียวกับ `[123456789]`
  - `[abcde]` ใช้ในความหมายเดียวกับ `[a-e]`
- ผู้ใช้สามารถกำหนดพิสัยได้มากกว่าหนึ่ง
  - `[abcde123456789]` ใช้ในความหมายเดียวกับ `[a-e1-9]`



# RegEx : Alternate Character Classes

- เครื่องหมายก้ามปู [] ใช้ในการค้นหากลุ่มของอักขระที่ถูกระบุ

*regular expression* → 

g	r	[ae]	y
---	---	------	---

grey



*match 1*

geay greay

gray



*match 2*

# RegEx : Anchors

- อักขระ ^ (caret) ใช้ค้นหาข้อความที่ขึ้นต้นด้วยอักขระที่ถูกระบุในบรรทัดนั้น

regular expression → 

^	c	a
---	---	---

cat

↑  
match

tomcat

cattle

↑  
match

regular expression →

^	b	[eor]	a	t
---	---	-------	---	---

beat

↑  
match

a brat

↑  
match

on a boat

↑  
match

# RegEx : Anchors

- ^ เมื่อใช้ร่วมกับ Character Classes จะมีความหมายเป็น Negate
- เช่น `q[^A-Z]` หมายถึง “อักขระ q ต่อท้ายด้วยอักขระที่ไม่ใช่ A-Z”

regular expression → `q[ ^ A - Z ]`

`q1` ← match

`qA`

`qc` ← match

# The Vertical Bar or Pipe Symbol

---

- Regex สามารถใช้อักขระ | (OR) สำหรับ matching นิพจน์หรือนิพจน์ย่อยได้ เช่น
  - **(T|F)an** match **Tan** หรือ **Flan**
  - **^(From|Subject):** match บรรทัดที่คำขึ้นต้นด้วย **From** หรือ **Subject**  
ต่อท้ายด้วยเครื่องหมาย : ภายในไฟล์ข้อมูลที่ถูกระบุ
- นอกจากนั้นยังสามารถใช้กับนิพจน์ย่อยเพื่อจำกัดขอบเขตการทำงานที่เป็นทางเลือกได้ เช่น
  - **At(ten|nine)tion** match **Attention** หรือ **Atnintion**
  - **Atten|nintion** match **Atten** หรือ **nintion** (ไม่มีการกำหนดวงเล็บไว้)

# Quantifier : Limiting Repetition

---

- รูปแบบเครื่องหมายปีก :  $\{min,max\}$
- Ranges สามารถระบุขนาดได้ดังนี้
  - $\{n,m\}$  ระบุค่าต่ำสุด  $n$  และค่าสูงสุดเป็น  $m$
  - $\{n\}$  หมายถึงจำนวน  $n$  ครั้งที่เกิดขึ้น
  - $\{n,\}$  หมายถึงอย่างน้อยที่สุดเป็นจำนวน  $n$
- ตัวอย่างเช่น
  - $\{0,\}$  ใช้ในความหมายเดียวกับ  $\{0,\}$
  - $a\{2,\}$  ใช้ในความหมายเดียวกับ  $aa*$
- $[1-9]\{3\}$  match ค่าตัวเลขระหว่าง 111 และ 999

# Quantifier : Repetition with \*, +, ?

---

- \* หมายถึงอักขระหรือกลุ่มของอักขระที่อยู่ก่อนหน้าปรากฏขึ้นเป็นจำนวนครั้งตั้งแต่ 0 หรือมากกว่า
  - \* ใช้ในความหมายเดียวกับ {0,}
- + หมายถึงอักขระหรือกลุ่มของอักขระที่อยู่ก่อนหน้าปรากฏขึ้นเป็นจำนวนครั้งตั้งแต่ 1 หรือมากกว่า
  - + ใช้ในความหมายเดียวกับ {1,}
- ? ใช้ระบุ **optional character** ของอักขระที่อยู่ก่อนหน้าปรากฏขึ้นเป็นจำนวนครั้งตั้งแต่ 0 หรือ 1 ครั้ง
  - ? ใช้ในความหมายเดียวกับ {0,1}

# RegEx : Repetitions

- อักขระ \* ใช้ในการระบุว่ามีการปรากฏขึ้นเป็นจำนวนครั้งตั้งแต่ 0 หรือมากกว่าในนิพจน์ที่กำหนดไว้ก่อนหน้าอักขระ

regular expression → 

y	a	*	y
---	---	---	---

I got mail, **yaaaaaaaaaay!**

↑  
match

# Extended RegEx : Repetition Shorthands

---

- **abc+d**
  - match **abcd**, **abccd**, หรือ **abcccccd** แต่ไม่ match กับ **abd**
- **abc?d**
  - match **abd** และ **abcd** แต่ไม่ match กับ **abccd**
- **July?** match **Jul** หรือ **July** (ปรากฏขึ้นตั้งแต่ 0 จนถึง 1 ครั้ง)
  - ใช้ในความหมายเดียวกับ **{0,1}**
  - ใช้ในความหมายเดียวกับ **(Jul|July)**
- **abc\***
  - matches **ab**, **abc**, **abcc**, **abccc**
- **(abc)\***
  - match **abc**, **abcabc** แต่ไม่ match กับ **ab** หรือ **abcd**



# Regular Expression Quick Guide

---

- abc** **Matches** ข้อความที่มีคำว่า **abc**
- ^** **Matches** อักขระหรือกลุ่มอักขระของบรรทัดที่ขึ้นต้นด้วย
- \$** **Matches** อักขระหรือกลุ่มอักขระของบรรทัดที่ลงท้ายด้วย
- .** **Matches** อักขระใด ๆ
- \s** **Matches** ช่องว่างหรือ **whitespace**
- [aeiou]** **Matches** อักขระหนึ่ง ๆ ภายใน **set**
- {n, m}** จำกัดจำนวนต่ำสุด **n** และสูงสุด **m** ของอักขระหรือกลุ่มอักขระ
- [a-z0-9]** กำหนดกลุ่มของอักขระที่สามารถรวมกันเป็นช่วงพิสัย
- \*** **Repeats** อักขระหนึ่ง ๆ ที่มีจำนวนครั้งตั้งแต่ 0 หรือมากกว่า
- +** **Repeats** อักขระหนึ่ง ๆ ที่มีจำนวนครั้งตั้งแต่ 1 หรือมากกว่า
- ?** **Repeats** อักขระหนึ่ง ๆ ที่มีจำนวนครั้งตั้งแต่ 0 หรือ 1 ครั้ง

# Verify Match with Regex

---

```
WebUI.openBrowser("")
```

```
WebUI.navigateToUrl('http://demoaut.katalon.com/')
```

```
String text = WebUI.getText(findTestObject('Page_CURA  
Healthcare Service/a_Make Appointment'))
```

```
WebUI.comment("text = "+text)
```

```
WebUI.verifyMatch(text, 'Make Appointment', true)
```

```
WebUI.verifyMatch(text, '.*ment$', true)
```

```
WebUI.verifyNotMatch(text, 'ment$', true)
```

```
WebUI.verifyMatch(text, '^M.*', true)
```

```
WebUI.verifyNotMatch(text, '^M', true)
```

```
WebUI.verifyMatch(text, '.*', true)
```

```
WebUI.closeBrowser()
```

# Assert Statement

---

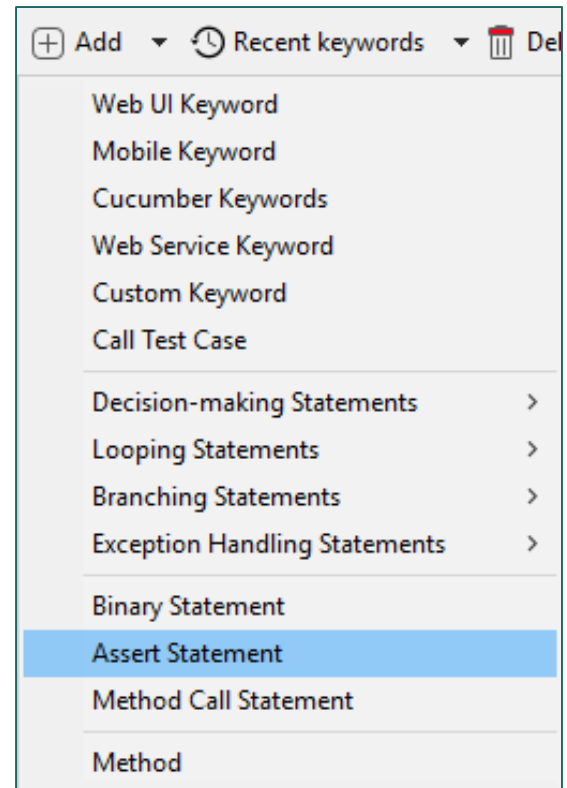
- Assert statement ประกอบด้วยนิพจน์ boolean ที่มีการทำงานต่อเนื่องเมื่อการประเมินผลเงื่อนไขเป็น แต่หากเป็นเท็จจะมีการรายงานความผิดพลาดทันที
- ความแตกต่างระหว่างการทำงานของคำสั่ง Assert และ Verify.. ใน Katalon Studio จะแตกต่างกันตรงที่เมื่อมีความผิดพลาดเกิดขึ้น
  - Assert จะแสดง Error และสิ้นสุดการทำงานทันที
  - Verify จะแสดง Error และทำงานต่อในคำสั่งถัดไป



# Assert Statement

- เนื่องจาก Assert ไม่ได้เป็นส่วนหนึ่งของคีย์เวิร์ดที่กำหนดไว้ใน WebUI
- ดังนั้นการเรียกใช้ Assert Statement สามารถทำได้โดยคลิกเลือก Add ภายใน Manual Mode โดยมีขั้นตอนดังนี้
  - Assert Statement จะถูกกำหนดลงใน Test Step ที่มีการเรียกใช้
  - จากนั้นดับเบิลคลิกที่ Input เพื่อกำหนดเงื่อนไข assertion
  - ตัวอย่างเช่น

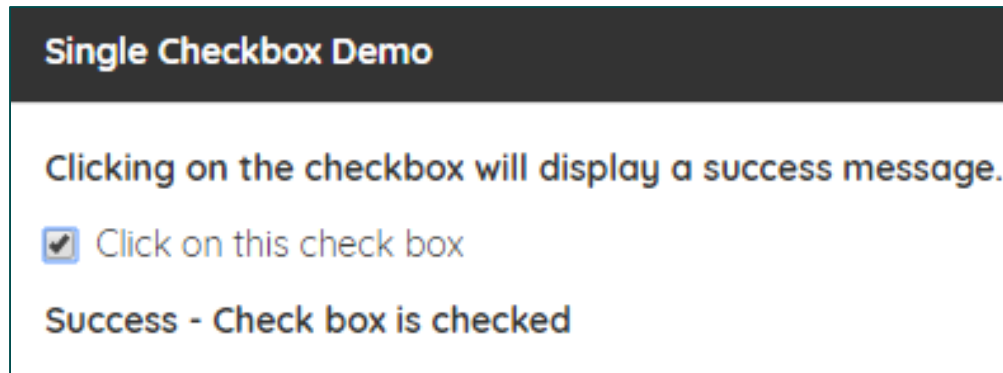
**`assert WebUI.getWindowTitle() == 'My Title'`**



# Checkbox : click()

---

- การเลือกค่า checkBox ใน Katalon Studio สามารถทำได้โดยการเรียกใช้คำสั่ง check หรือ WebUI.check()
- ในทำนองเดียวกันการไม่เลือก checkbox สามารถทำได้โดยการเรียกใช้คำสั่ง uncheck หรือ WebUI.uncheck()



## Verify Element Check /uncheck

---

- ในกรณีที่ต้องการตรวจสอบ Checkbox ว่ามีการเลือกไว้แล้วหรือไม่ ผู้ใช้สามารถใช้คำสั่ง Verify Element Check หรือ Verify Element Not Check ได้ โดยมีรูปแบบดังนี้

**static boolean verifyElementNotChecked(TestObject to, int timeout)**

Param	Param Type	Mandatory	Description
to	TestObject	Required	Represent a web element.
timeout	int	Required	System will wait at most timeout (seconds) to return result

# Dropdown List

---

- Katalon Studio สอนการทำงานของ Dropdown คำผ่านคีย์เวิร์ดต่าง ๆ ดังนี้
  - **getNumberOfSelectedOption:** คืนค่าจำนวน Option ที่ถูกเลือกใน Dropdown
  - **getNumberOfTotalOption:** คืนค่าจำนวน Option ที่อยู่ภายใน Dropdown
  - **Select All Option:** ใช้สำหรับเลือกทุก ๆ Options ภายใน Dropdown
  - **Select Option By Index:** ใช้สำหรับเลือก Option จากค่าตัวเลขที่ถูกระบุโดยปกติค่า Option เริ่มต้นจาก 0 เสมอ
  - **Select Option By Label:** ใช้สำหรับเลือก Option จากค่าที่ถูกนำเสนอภายใน Dropdown
  - **Select Option by Value:** ใช้เลือก Option จากค่าแอททริบิวต์ “value”

# Dropdown List

- ตัวอย่างการเรียกใช้สคริปต์เพื่อเลือกค่า Option ภายใน Dropdown List

**`WebUI.selectOptionByLabel(findTestObject('select'), 'Wednesday', false)`**

**`WebUI.selectOptionByIndex(findTestObject('select'), 4)`**

**`WebUI.selectOptionByValue(findTestObject('select'), 'Wednesday', false)`**

**Select List Demo**

Selected value from the list will display below the dropdown

Select a day (select one):

Please select ▼

Please select

Sunday

Monday

Tuesday

**Wednesday**

Thursday

Friday

Saturday



# RadioButton

- Katalon กำหนดค่า Radio Button โดยการเรียกใช้เมธอด `click()` ร่วมกับ `TestObject` ที่ถูกระบุดังตัวอย่างโค้ดต่อไปนี้

```
WebUI.openBrowser("")
```

```
WebUI.navigateToUrl('https://www.seleniumeasy.com/test/basic-radiobutton-demo.html')
```

```
WebUI.click(findTestObject('Female'))
```

```
WebUI.click(findTestObject('btnCheck'))
```

```
msgRadio = WebUI.getText(findTestObject('txtRadio'))
```

```
WebUI.comment('Text is ' + msgRadio)
```

```
WebUI.delay(GlobalVariable.timeOut)
```

```
WebUI.closeBrowser()
```

**Radio Button Demo**

Click on button to get the selected value

☐ Male ☒ Female

Radio button 'Female' is checked

Param Type	Param	Value Type	Value
String	"var"	String	"Female"

## Ex : Register

---

- สร้าง Test Script ด้วยมือ โดยอาศัย ChroPath ช่วยในการ Add Property ของ Test Object และผ่านค่าพารามิเตอร์ Test Object จาก Test Case ดังต่อไปนี้

TC #	Test Step	Test Data
1	Open Browser	Chrome
2	Go to the Application URL	<a href="http://demo.nopcommerce.com/login">http://demo.nopcommerce.com/login</a>
3	Enter username with Param	Yes
4	Enter password with Param	Yes
5	Click Login Button	Yes
6	Verify Element Present	Welcome Page
7	Close Browser	Yes