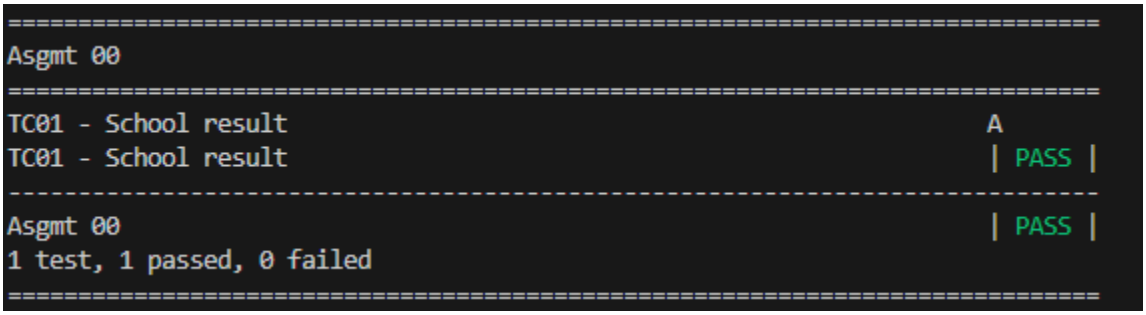
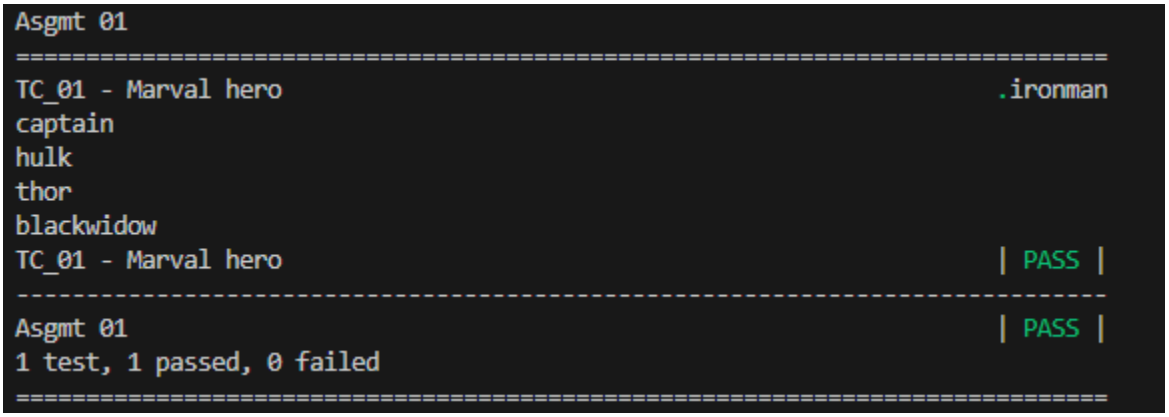
Assignment 00 (IF-Else)



Assignment 01 (For Loop)

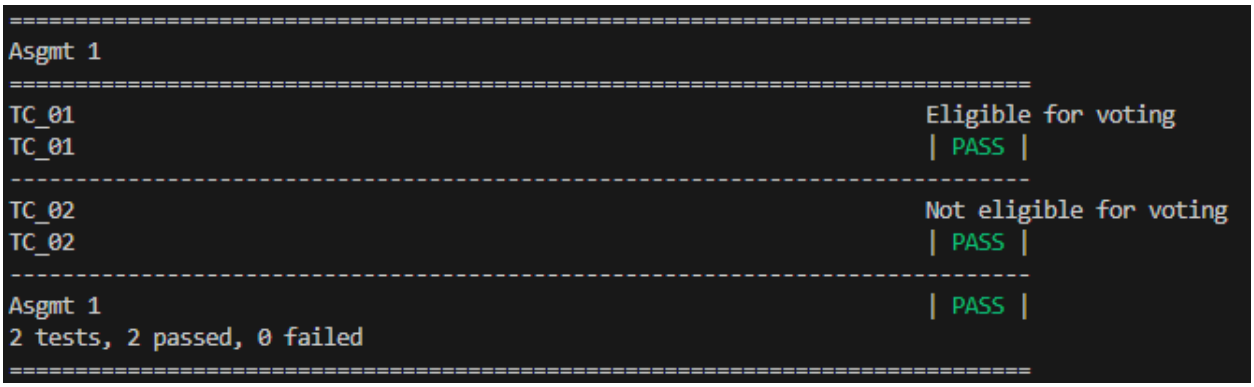


Assignment 1: การส่ง Arguments แบบ Scalar

วัตถุประสงค์: เรียนรู้วิธีการส่งและใช้ค่า scalar argument ใน Robot Framework

เป้าหมาย: - เข้าใจการส่งค่า scalar argument เข้าใน keyword - สามารถใช้ argument ที่ส่งเข้าไปในเงื่อนไข if-else ได้

วิธีการทำ: 1. สร้าง keyword ชื่อ `Validate Age` ที่รับค่า argument ชื่อ `${age}`. 2. ใน keyword นี้ ให้ใช้เงื่อนไข `Run Keyword If` เพื่อตรวจสอบว่า `${age}` มีค่ามากกว่า 18 หรือไม่: - ถ้ามากกว่า 18 ให้พิมพ์ว่า "Eligible for voting". - ถ้าไม่ใช่ ให้พิมพ์ว่า "Not eligible for voting". 3. สร้าง Test Case ที่ทดสอบ keyword `Validate Age` โดยส่งค่า 16 และ 20 เพื่อดูผลลัพธ์.

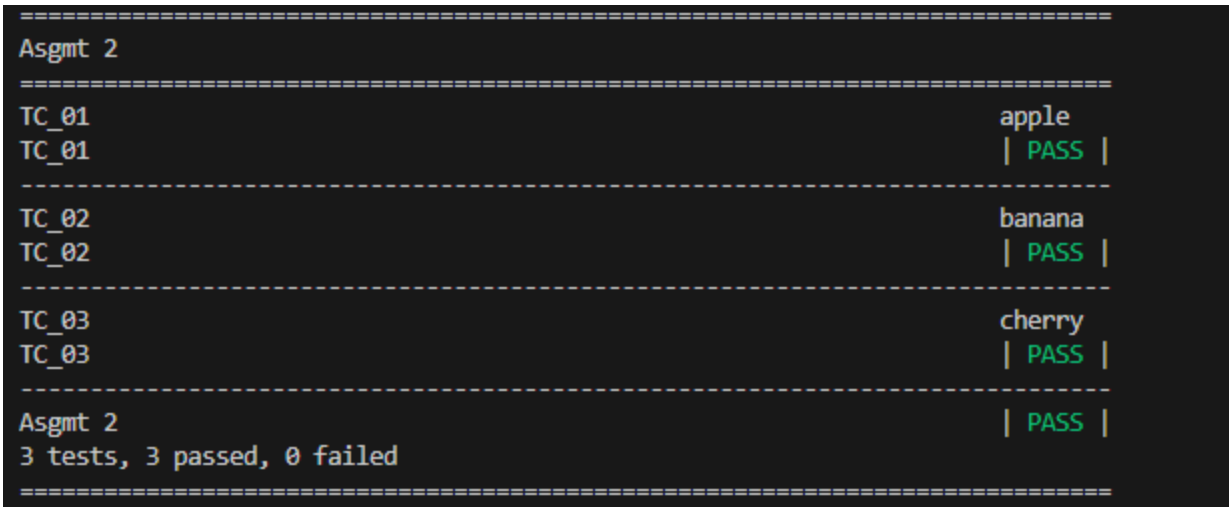


Assignment 2: การส่ง Arguments แบบ List

วัตถุประสงค์: เรียนรู้การส่งและจัดการค่า list argument

เป้าหมาย: - สามารถรับและวนลูป list argument ได้ - เข้าใจการใช้งาน `FOR` loop ใน Robot Framework

วิธีการทำ: 1. สร้าง keyword ชื่อ `Print Fruits` ที่รับค่า list argument ชื่อ `@{fruits}`. 2. ใน keyword นี้ ให้ใช้ `FOR` loop เพื่อนำค่าแต่ละตัวใน list มาพิมพ์: `robot FOR ${fruit} IN @{fruits} Log ${fruit} END` 3. สร้าง Test Case ที่ทดสอบ keyword `Print Fruits` โดยส่งค่าเป็น list เช่น `apple`, `banana`, `cherry`.

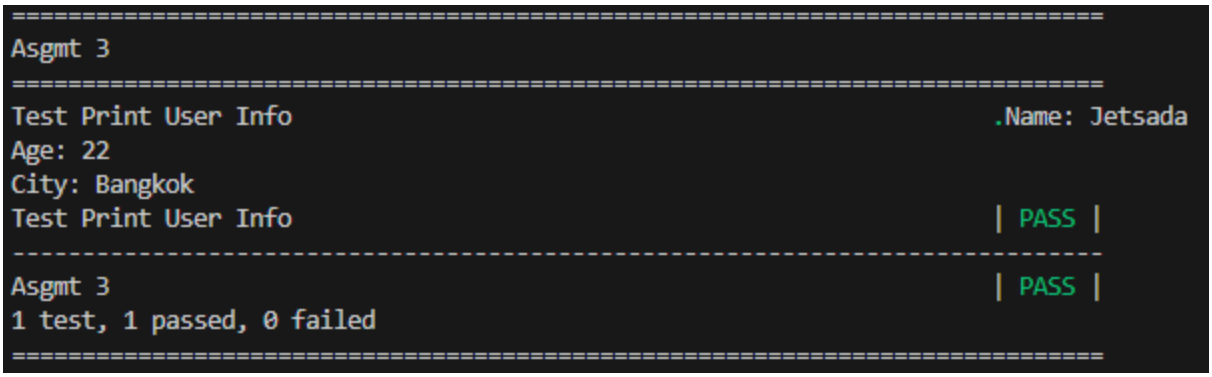


Assignment 3: การส่ง Arguments แบบ Dictionary

วัตถุประสงค์: เรียนรู้การส่งและจัดการค่า dictionary argument

เป้าหมาย: - เข้าใจการใช้งาน dictionary argument ใน Robot Framework - สามารถดึงค่าออกจาก dictionary และใช้ค่าเหล่านั้นใน keyword ได้

วิธีการทำ: 1. สร้าง keyword ชื่อ `Print User Info` ที่รับค่า dictionary argument ชื่อ `&{user_info}`. 2. ใน keyword นี้ ให้ดึงค่า `name`, `age`, และ `city` จาก dictionary และพิมพ์ข้อมูลนั้นออกมา: `robot Log Name: ${user_info["name"]} Log Age: ${user_info["age"]} Log City: ${user_info["city"]}` 3. สร้าง Test Case ที่ทดสอบ keyword `Print User Info` โดยส่ง dictionary ที่มี key เป็น `name`, `age`, และ `city`.

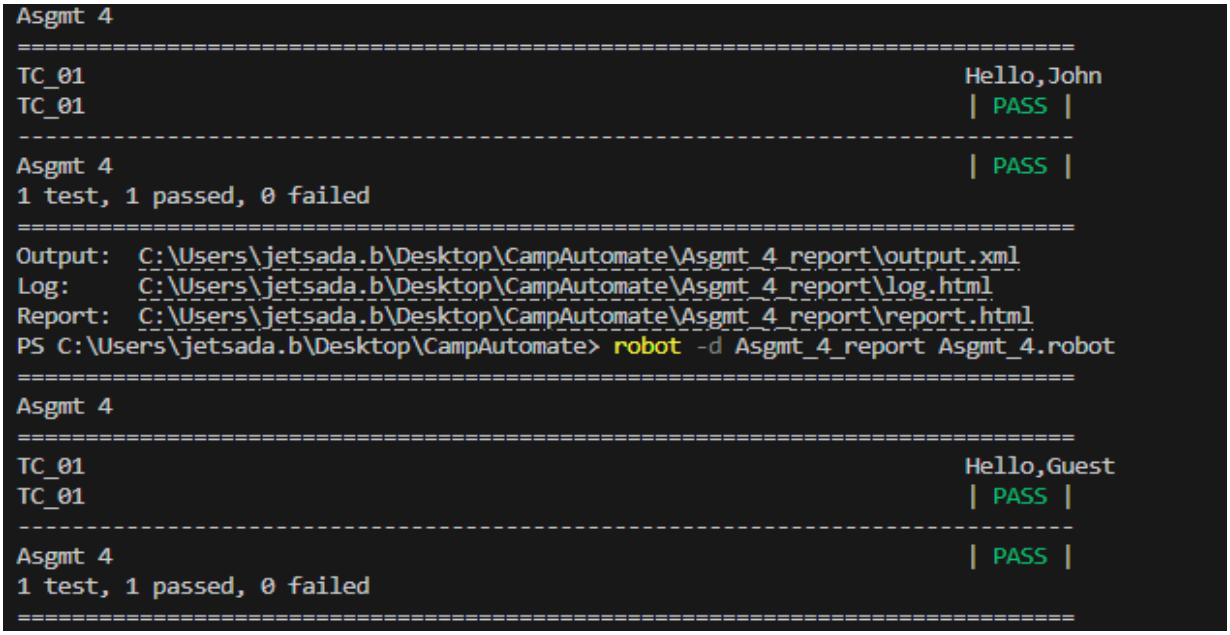


Assignment 4: การใช้ Default Value ใน Arguments

วัตถุประสงค์: เข้าใจการกำหนดค่า default ให้กับ argument

เป้าหมาย: - สามารถสร้าง keyword ที่มี default value ใน argument ได้ - เข้าใจการทำงานเมื่อส่งค่าและไม่ส่งค่าไปใน argument

วิธีการทำ: 1. สร้าง keyword ชื่อ `Greet User` ที่รับค่า argument ชื่อ `${name}` โดยกำหนด default value เป็น "Guest". 2. ใน keyword นี้ ให้พิมพ์ข้อความว่า "Hello, ${name}". 3. สร้าง Test Case ที่ทดสอบ keyword `Greet User` โดย: - เรียกใช้งานโดยไม่ส่งค่าไปใน `${name}` (ควรแสดง "Hello, Guest"). - เรียกใช้งานโดยส่งค่า `${name}` เป็น "John" (ควรแสดง "Hello, John").

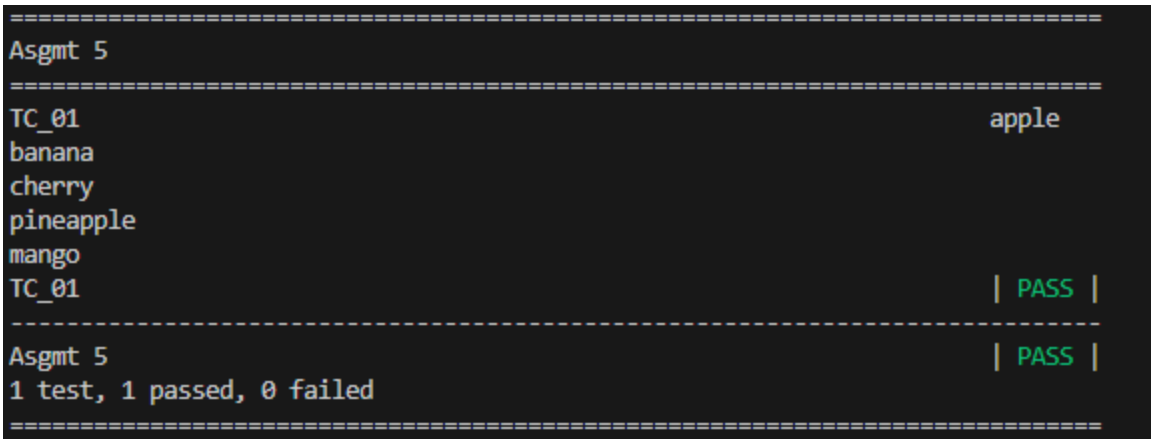


Assignment 5: Keyword ที่รับ Arguments ไม่จำกัดจำนวน (เป็น List)

วัตถุประสงค์: เรียนรู้การสร้าง keyword ที่รับ arguments ไม่จำกัดจำนวน

เป้าหมาย: - เข้าใจการใช้ `*args` เพื่อรับ arguments จำนวนไม่จำกัด - สามารถจัดการกับ arguments ที่ส่งเข้าไปได้

วิธีการทำ: 1. สร้าง keyword ชื่อ `Print All Items` ที่รับ arguments ไม่จำกัดจำนวนด้วย `*{items}`. 2. ใน keyword นี้ ให้ใช้ `FOR` loop เพื่อนำค่าแต่ละตัวที่รับมาใน `*{items}` มาพิมพ์: `robot FOR ${item} IN @{items} Log ${item} END` 3. สร้าง Test Case ที่ทดสอบ keyword `Print All Items` โดยส่ง arguments 3-5 ค่า เช่น "apple", "banana", "cherry".



Assignment 6: การจัดการกับ Variables (Global vs Local)

วัตถุประสงค์: เข้าใจความแตกต่างระหว่างตัวแปร global และ local

เป้าหมาย: - รู้วิธีการประกาศและใช้ตัวแปร global และ local - เข้าใจขอบเขตการทำงานของตัวแปรใน Robot Framework

วิธีการทำ: 1. สร้าง Test Case ที่กำหนดค่าให้ตัวแปร global เช่น: `robot Set Global Variable ${GLOBAL_VAR} "Global Value"` 2. สร้าง keyword ที่ประกาศและใช้ตัวแปร local ภายใน keyword นั้นเอง: `robot Set Local Variable ${LOCAL_VAR} "Local Value" Log ${LOCAL_VAR}` 3. ทดสอบการเข้าถึงค่าของตัวแปรทั้งสองประเภทใน Test Case.

```
Asgmt 6
==============================================================================
TC_01                                                          .Local Value
Global Value
TC_01                                                          | PASS |
------------------------------------------------------------------------------
Asgmt 6                                                        | PASS |
1 test, 1 passed, 0 failed
==============================================================================
```

Assignment 7: การดึงค่า Variables จากไฟล์ YAML

วัตถุประสงค์: เรียนรู้การดึงค่า variables จากไฟล์ YAML

เป้าหมาย: - เข้าใจวิธีการโหลดไฟล์ YAML และดึงค่า variables มาใช้งาน - สามารถใช้ค่าที่ดึงมาใน Test Case ได้

วิธีการทำ: 1. สร้างไฟล์ YAML ชื่อ `variables.yaml` ที่มีค่าตัวแปรต่างๆ เช่น: `yaml user: name: John age: 30 city: Bangkok` 2. สร้าง Test Case ที่โหลดค่า variables จากไฟล์ YAML: `robot Variables variables.yaml` 3. ทดสอบการดึงค่าตัวแปรใน Test Case เช่น `${user.name}`, `${user.age}`, `${user.city}`.

```
==============================================================================
Asgmt 7
==============================================================================
TC_01 - Load Variables from YAML                               Name: John
.Age: 30
.City: Bangkok
TC_01 - Load Variables from YAML                               | PASS |
------------------------------------------------------------------------------
Asgmt 7                                                        | PASS |
1 test, 1 passed, 0 failed
==============================================================================
```

Assignment 8: การ Preload Variables จากไฟล์ YAML ก่อนการทดสอบ

วัตถุประสงค์: เรียนรู้การ preload ตัวแปรจากไฟล์ YAML ก่อนการรัน Test Case

เป้าหมาย: - เข้าใจวิธีการโหลดตัวแปรจากไฟล์ YAML และใช้ค่าที่โหลดมาใน Test Case - รู้วิธีการตั้งค่าให้ตัวแปรถูกโหลดอัตโนมัติเมื่อเริ่มการทดสอบ

วิธีการทำ: 1. สร้างไฟล์ YAML ชื่อ `config.yaml` ที่มีค่าตัวแปรต่างๆ เช่น: `yaml app_url: http://example.com credentials: username: admin password: password123` 2. ใน Test Suite, กำหนดให้โหลดไฟล์ YAML โดยใช้ `Variables: robot Variables config.yaml` 3. สร้าง Test Case ที่ใช้ค่าตัวแปรจาก YAML ในการทดสอบ เช่น: `robot Open Browser ${app_url} chrome Input Text username_field ${credentials.username} Input Text password_field ${credentials.password}`

```
==============================================================================
Asgmt 8
==============================================================================
TC_01
DevTools listening on ws://127.0.0.1:59253/devtools/browser/db972093-5b0d-46b2-8058-0816ee43278e
TC_01                                                                 | PASS |
------------------------------------------------------------------------------
Asgmt 8                                                               | PASS |
1 test, 1 passed, 0 failed
==============================================================================
```

Assignment 9: การ Return ค่ากลับเป็น Scalar

วัตถุประสงค์: เรียนรู้การสร้าง keyword ที่ return ค่ากลับเป็น scalar

เป้าหมาย: - เข้าใจการใช้ `Return From Keyword` ในการส่งค่ากลับจาก keyword - สามารถสร้าง keyword ที่ return ค่าผลลัพธ์กลับมาได้

วิธีการทำ: 1. สร้าง keyword ชื่อ `Calculate Square` ที่รับค่า argument และ return ค่ากลับเป็นผลคูณของตัวเอง: `robot *** Keywords *** Calculate Square [Arguments] ${number} ${result} Set Variable ${number} * ${number} Return From Keyword ${result}` 2. สร้าง Test Case ที่เรียกใช้ keyword `Calculate Square` และตรวจสอบผลลัพธ์: `robot ${square} Calculate Square 4 Should Be Equal ${square} 16`

```
Asgmt 9
==============================================================================
TC_01                                                                 .4 * 4
TC_01                                                                | PASS |
------------------------------------------------------------------------------
Asgmt 9                                                              | PASS |
1 test, 1 passed, 0 failed
==============================================================================
```

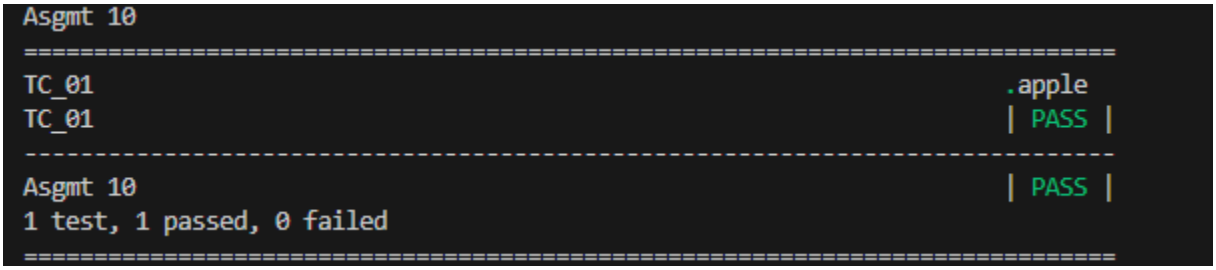
Assignment 10: การ Return ค่ากลับเป็น List

วัตถุประสงค์: เรียนรู้การสร้าง keyword ที่ return ค่ากลับเป็น list

เป้าหมาย: - เข้าใจการสร้างและ return list จาก keyword - สามารถใช้ค่าที่ return กลับมาใน Test Case ได้

วิธีการทำ: 1. สร้าง keyword ชื่อ `Create Fruit List` ที่ return ค่ากลับเป็น list: `robot *** Keywords *** Create Fruit List ${fruits} Create List apple banana cherry Return From Keyword ${fruits}` 2. สร้าง Test Case ที่เรียกใช้ keyword `Create Fruit List`

และทดสอบการใช้งาน list ที่ return กลับมา: `robot @{my_fruits} Create Fruit List Log ${my_fruits[0]} # apple`

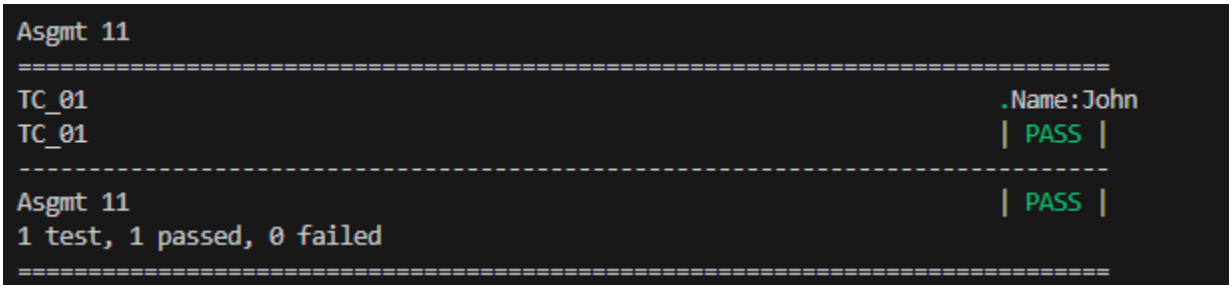


Assignment 11: การ Return ค่ากลับเป็น Dictionary

วัตถุประสงค์: เรียนรู้การสร้าง keyword ที่ return ค่ากลับเป็น dictionary

เป้าหมาย: - เข้าใจการสร้างและ return dictionary จาก keyword - สามารถดึงค่า key-value จาก dictionary ที่ return กลับมาได้

วิธีการทำ: 1. สร้าง keyword ชื่อ `Create User Info` ที่ return ค่ากลับเป็น dictionary: `robot *** Keywords *** Create User Info ${user_info} Create Dictionary name=John age=30 city=Bangkok Return From Keyword ${user_info}` 2. สร้าง Test Case ที่เรียกใช้ keyword `Create User Info` และทดสอบการดึงค่า key-value: `robot &{user} Create User Info Log Name: ${user.name} # John`



Assignment 12: การใช้ If/If-Else Statement

วัตถุประสงค์: เรียนรู้การสร้างเงื่อนไข if-else ใน Robot Framework

เป้าหมาย: - เข้าใจการใช้ `Run Keyword If` ในการสร้างเงื่อนไข if-else - สามารถจัดการกับเงื่อนไขที่ซับซ้อนได้

วิธีการทำ: 1. สร้าง Test Case ที่ตรวจสอบอายุและพิมพ์ข้อความตามเงื่อนไข: `robot *** Test Cases *** Age Verification ${age} Set Variable 20 Run Keyword If ${age} > 18 Log You are an adult ... ELSE IF ${age} == 18 Log You just became an adult ... ELSE Log You are underage`

```
Asgmt 12
==============================================================================
TC_01                                                         .You are an adult
TC_01                                                         | PASS |
------------------------------------------------------------------------------
Asgmt 12                                                      | PASS |
1 test, 1 passed, 0 failed
==============================================================================
```

Assignment 13: การใช้ For Loop

วัตถุประสงค์: เรียนรู้การใช้ FOR loop ใน Robot Framework

เป้าหมาย: - เข้าใจการใช้ FOR loop เพื่อวนลูปรายการ - สามารถประยุกต์ใช้ loop ในสถานการณ์ต่างๆ ได้

วิธีการทำ: 1. สร้าง Test Case ที่วนลูปพิมพ์รายการผลไม้: `robot *** Test Cases *** Print Fruit List @{fruits} Create List apple banana cherry FOR ${fruit} IN @{fruits} Log ${fruit} END`

```
Asgmt 13
==============================================================================
Print Fruit List                                                      .apple banana cherry
Print Fruit List                                                      | PASS |
------------------------------------------------------------------------------
Asgmt 13                                                              | PASS |
1 test, 1 passed, 0 failed
==============================================================================
```

Assignment 14: การใช้ Break และ Continue ใน Loop

วัตถุประสงค์: เรียนรู้การใช้ break และ continue ใน Robot Framework

เป้าหมาย: - เข้าใจวิธีการใช้ `Exit For Loop` และ `Continue For Loop` - สามารถจัดการกับการหยุดและข้ามการวนลูปได้

วิธีการทำ: 1. สร้าง Test Case ที่วนลูปตัวเลขและใช้ break และ continue ตามเงื่อนไข: `robot *** Test Cases *** Loop Control Example FOR ${number} IN RANGE 1 10 Run Keyword If ${number} == 5 Exit For Loop Run Keyword If ${number} == 3 Continue For Loop Log ${number} END`

```
Asgmt 14
==============================================================================
TC_01                                                                 1
2
4
TC_01                                                                 | PASS |
------------------------------------------------------------------------------
Asgmt 14                                                              | PASS |
1 test, 1 passed, 0 failed
==============================================================================
```

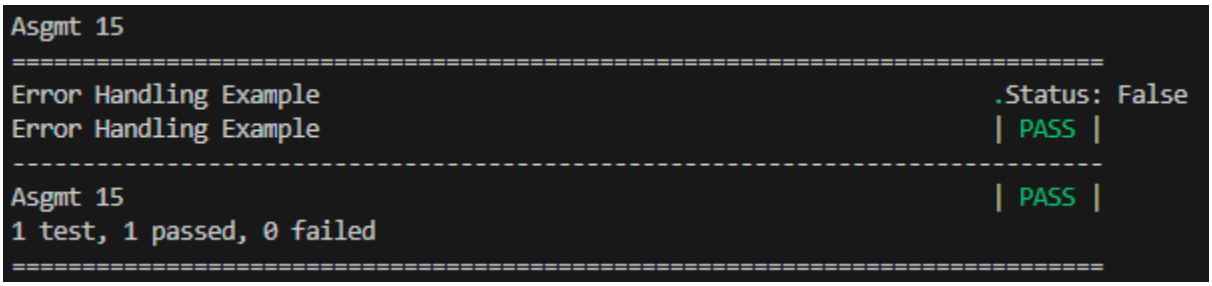
Assignment 15: การจัดการข้อผิดพลาดด้วย `Run Keyword And Return Status` และ `Run Keyword And Ignore Error`

วัตถุประสงค์: เรียนรู้การจัดการข้อผิดพลาดใน Robot Framework

เป้าหมาย: - เข้าใจการใช้ `Run Keyword And Return Status` และ `Run Keyword And Ignore Error` - สามารถจัดการการทำงานเมื่อเกิดข้อผิดพลาดได้

วิธีการทำ: 1. สร้าง Test Case ที่ใช้ `Run Keyword And Return Status` และ `Run Keyword And Ignore Error` ในการจัดการข้อผิดพลาด: ```robot *** Test Cases *** Error Handling Example ${status} Run Keyword And Return Status Click Element invalid_locator Log Status: ${status}

```
   Run Keyword And Ignore Error    Input Text    invalid_locator
value
    Log    Ignored the error and continued
```
```




# Assignment 16: การสร้าง Recursive Keyword

**วัตถุประสงค์:** เรียนรู้การสร้าง recursive keyword ใน Robot Framework
```

เป้าหมาย: - เข้าใจการทำงานของ recursive function ในรูปแบบของ keyword - สามารถสร้าง keyword ที่เรียกใช้งานตัวเองซ้ำ ๆ ได้

วิธีการทำ: 1. สร้าง keyword ชื่อ `Calculate Factorial` ที่รับค่า argument ชื่อ `${number}` และคำนวณค่าของ factorial โดยใช้การเรียกตัวเองซ้ำ: `robot *** Keywords *** Calculate Factorial [Arguments] ${number} Run Keyword If ${number} == 1 Return From Keyword 1 ${previous} Calculate Factorial ${number - 1} ${result} Set Variable ${number} * ${previous} Return From Keyword ${result}` 2. สร้าง Test Case ที่ทดสอบการคำนวณ factorial ของตัวเลข เช่น 5 และตรวจสอบผลลัพธ์ที่ได้: `robot *** Test Cases *** Test Factorial Calculation ${result} Calculate Factorial 5 Should Be Equal ${result} 120`

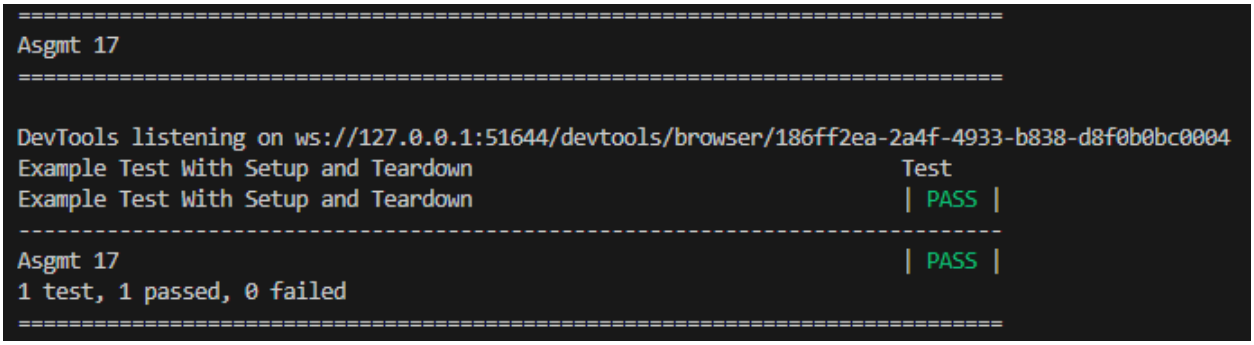
```
Asgmt 16
==============================================================================
TC_01                                                                 ...120
TC_01                                                                 | PASS |
------------------------------------------------------------------------------
Asgmt 16                                                              | PASS |
1 test, 1 passed, 0 failed
==============================================================================
```

Assignment 17: การตั้งค่า Setup และ Teardown

วัตถุประสงค์: เรียนรู้การใช้ setup และ teardown ใน Test Case

เป้าหมาย: - เข้าใจการตั้งค่า setup และ teardown เพื่อจัดเตรียมและทำความสะอาดสภาพแวดล้อมการทดสอบ - สามารถประยุกต์ใช้ setup และ teardown ในสถานการณ์ต่าง ๆ ได้

วิธีการทำ: 1. สร้าง Test Case ที่มีการตั้งค่า setup และ teardown: `robot *** Test Cases *** Example Test With Setup And Teardown [Setup] Open Browser http://example.com chrome [Teardown] Close Browser Log Test is running` 2. ทดสอบการทำงานของ Test Case โดยสังเกตว่าการเปิดและปิดเบราว์เซอร์ทำงานอย่างถูกต้อง



Assignment 18: การตั้งค่า Suite Setup และ Suite Teardown

วัตถุประสงค์: เข้าใจการใช้ Suite Setup และ Suite Teardown ในการเตรียมและสรุปการทดสอบ

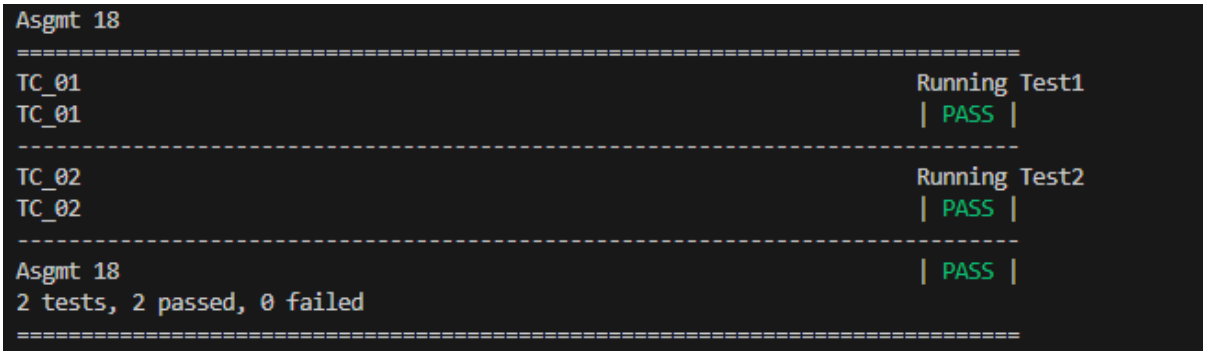
เป้าหมาย: - สามารถตั้งค่า Suite Setup และ Suite Teardown ได้ - เข้าใจการทำงานของ Setup/Teardown ในระดับ Suite

วิธีการทำ: 1. สร้าง Test Suite ที่มี Suite Setup และ Suite Teardown: ```robot *** Settings *** Suite Setup Log Suite is starting Suite Teardown Log Suite is ending

```
*** Test Cases ***
Test 1
    Log    Running Test 1
```

```
Test 2
    Log    Running Test 2
```

1. ทดสอบการรัน Suite และสังเกตการทำงานของ Suite Setup และ Suite Teardown



Assignment 19: การใช้ Teardown ใน Keyword, Test Case, และ Suite

วัตถุประสงค์: เข้าใจความแตกต่างระหว่าง Teardown ในระดับ keyword, Test Case, และ Suite

เป้าหมาย: - รู้วิธีการตั้งค่า Teardown ในระดับต่าง ๆ - เข้าใจการทำงานและลำดับของ Teardown ในแต่ละระดับ

วิธีการทำ: 1. สร้าง Test Suite ที่มี Teardown ในระดับต่าง ๆ: ```robot *** Settings *** Suite Teardown Log Suite Teardown is running

```
*** Test Cases ***
Test Case With Teardown
    [Teardown]    Log    Test Case Teardown is running
    Log    Running the test

*** Keywords ***
```

```
Keyword With Teardown
    [Teardown]    Log    Keyword Teardown is running
    Log    Running the keyword
```

1. ทดสอบการรันและสังเกตลำดับการทำงานของ Teardown ในแต่ละระดับ

```
Asgmt 19
==============================================================================
TC_01                                                 Keyword Teardown is running
Running the keyword
.Test Case Teardown is running
.Running the test
TC_01                                                 | PASS |
------------------------------------------------------------------------------
Asgmt 19                                              | PASS |
1 test, 1 passed, 0 failed
==============================================================================
```

Assignment 20: การใช้งาน Tags ในการทดสอบ

วัตถุประสงค์: เรียนรู้การใช้ tags ในการจัดกลุ่มและควบคุมการรัน Test Case

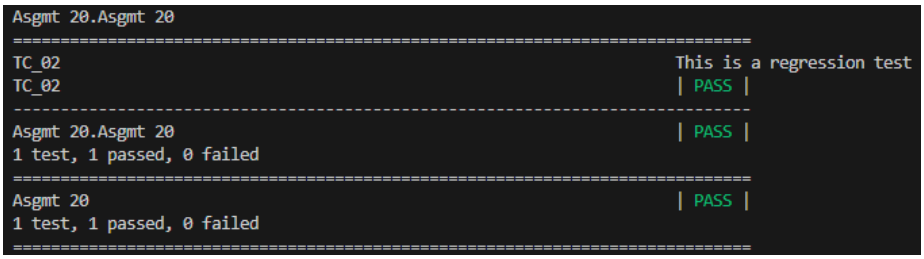
เป้าหมาย: - เข้าใจการติด tag ให้กับ Test Case - รู้วิธีการรันและควบคุมการรัน Test Case ตาม tag ที่กำหนด

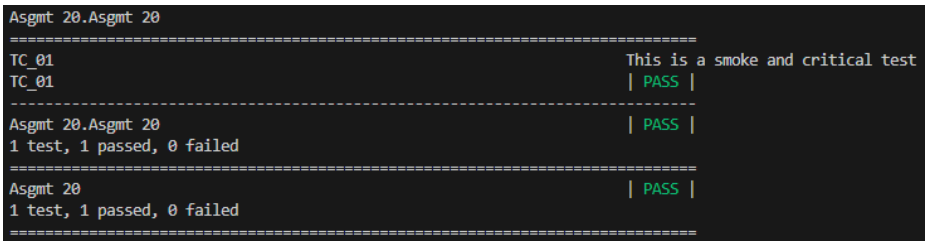
วิธีการทำ: 1. สร้าง Test Suite ที่มีการติด tag ใน Test Case: ```robot *** Test Cases *** Test With Tag 1 [Tags] smoke Log This is a smoke test

```
Test With Tag 2
    [Tags]    regression
    Log    This is a regression test

Test With Multiple Tags
    [Tags]    smoke    critical
    Log    This is a smoke and critical test
```

1. ทดสอบการรัน Test Case โดยใช้คำสั่งรันเฉพาะ tag ที่กำหนด เช่น:`robot --include smoke .`



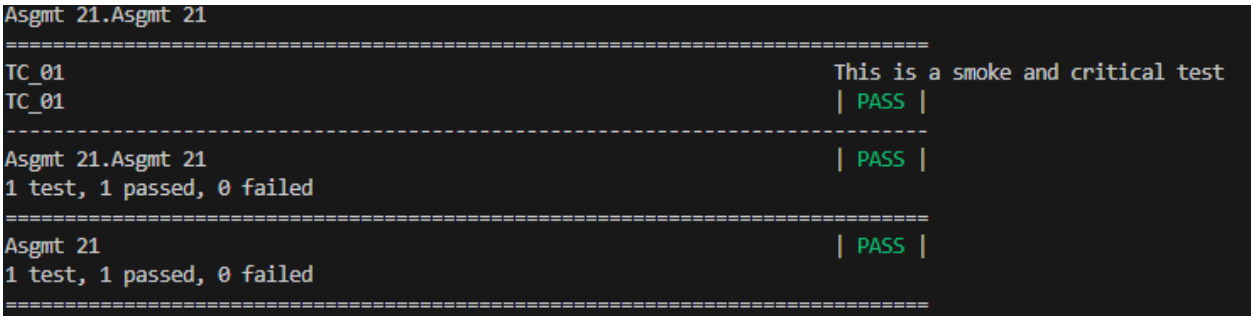


Assignment 21: การรัน Test Case โดยรวมเฉพาะบาง Tag

วัตถุประสงค์: เข้าใจการรัน Test Case โดยเลือกเฉพาะ tag ที่ต้องการ

เป้าหมาย: - สามารถรัน Test Case โดยรวมเฉพาะบาง tag ที่กำหนดได้

วิธีการทำ: 1. สร้าง Test Suite ที่มีหลาย tag ติดอยู่ใน Test Case ดังตัวอย่างใน Assignment 20 2. ทดสอบการรัน Test Case โดยรวมเฉพาะ tag ที่กำหนด เช่น: `bash robot --include critical .`



Assignment 22: การรัน Test Case โดยยกเว้นบาง Tag

วัตถุประสงค์: เรียนรู้วิธีการยกเว้นบาง tag เมื่อรัน Test Case

เป้าหมาย: - สามารถรัน Test Case โดยยกเว้น tag ที่ไม่ต้องการได้

วิธีการทำ: 1. ใช้ Test Suite จาก Assignment 20 2. ทดสอบการรัน Test Case โดยยกเว้น tag ที่กำหนด เช่น: `bash robot --exclude regression .`

```
Asgmt 22.Asgmt 22
==============================================================================
TC_01                                                         This is a smoke and critical test
TC_01                                                         | PASS |
------------------------------------------------------------------------------
Asgmt 22.Asgmt 22                                             | PASS |
1 test, 1 passed, 0 failed
==============================================================================
Asgmt 22                                                      | PASS |
1 test, 1 passed, 0 failed
==============================================================================
```

Assignment 23: การใช้ Pabot Command ในการรัน Test Case

แบบขนาน

วัตถุประสงค์: เรียนรู้การใช้ Pabot ในการรัน Test Case แบบขนาน

เป้าหมาย: - เข้าใจวิธีการใช้ Pabot เพื่อรัน Test Case แบบ parallel - สามารถควบคุมการรัน Test Case แบบขนานได้

วิธีการทำ: 1. สร้าง Test Suite ที่มีหลาย Test Case และไม่มีการพึ่งพากัน: ```robot *** Test Cases *** Parallel Test 1 Log Running Test 1

```
Parallel Test 2
    Log    Running Test 2

Parallel Test 3
    Log    Running Test 3
```

1. รัน Test Suite โดยใช้ Pabot:`pabot --processes 3 .`

```
Robot Framework remote server at 127.0.0.1:8270 started.
Storing .pabotsuitenames file
2025-02-05 09:08:17.912051 [PID:2672] [0] [ID:0] EXECUTING Asgmt 23.Asgmt 23
2025-02-05 09:08:18.458123 [PID:2672] [0] [ID:0] PASSED Asgmt 23.Asgmt 23 in 0.5 seconds
5 tests, 5 passed, 0 failed, 0 skipped.
==================================================
```

```
Robot Framework remote server at 127.0.0.1:8270 stopped.
PabotLib process stopped
Total testing: 0.50 seconds
Elapsed time:  3.10 seconds
```

Assignment 24: การควบคุม Pabot Flow ด้วย `-splittestlevel`

วัตถุประสงค์: เรียนรู้การควบคุม flow ของการรัน Test Case ด้วย `-splittestlevel`

เป้าหมาย: - เข้าใจการใช้ `-splittestlevel` เพื่อควบคุมการแบ่งการรัน Test Case - สามารถควบคุมการทำงานของ Pabot ได้อย่างมีประสิทธิภาพ

วิธีการทำ: 1. สร้าง Test Suite ที่มีหลาย Test Case ดังตัวอย่างใน Assignment 23 2. รัน Test Suite โดยใช้ `-splittestlevel` เพื่อควบคุมการแบ่งการรัน เช่น: `bash pabot --splittestlevel 2 .`

```
Robot Framework remote server at 127.0.0.1:8270 started.
Storing .pabotsuitenames file
2025-02-05 09:11:00.018105 [PID:7080] [0] [ID:1] EXECUTING Asgmt 24.Asgmt 24.TC_02
2025-02-05 09:11:00.018414 [PID:10420] [1] [ID:2] EXECUTING Asgmt 24.Asgmt 24.TC_03
2025-02-05 09:11:00.018513 [PID:14852] [4] [ID:3] EXECUTING Asgmt 24.Asgmt 24.TC_04
2025-02-05 09:11:00.018513 [PID:6300] [3] [ID:0] EXECUTING Asgmt 24.Asgmt 24.TC_01
2025-02-05 09:11:00.018513 [PID:6664] [2] [ID:4] EXECUTING Asgmt 24.Asgmt 24.TC_05
2025-02-05 09:11:00.689454 [PID:10420] [1] [ID:2] PASSED Asgmt 24.Asgmt 24.TC_03 in 0.6 seconds
2025-02-05 09:11:00.689454 [PID:14852] [4] [ID:3] PASSED Asgmt 24.Asgmt 24.TC_04 in 0.6 seconds
2025-02-05 09:11:00.689454 [PID:7080] [0] [ID:1] PASSED Asgmt 24.Asgmt 24.TC_02 in 0.6 seconds
2025-02-05 09:11:00.708359 [PID:6300] [3] [ID:0] PASSED Asgmt 24.Asgmt 24.TC_01 in 0.6 seconds
2025-02-05 09:11:00.725037 [PID:6664] [2] [ID:4] PASSED Asgmt 24.Asgmt 24.TC_05 in 0.6 seconds
5 tests, 5 passed, 0 failed, 0 skipped.
==================================================
```

```
Robot Framework remote server at 127.0.0.1:8270 stopped.
PabotLib process stopped
Total testing: 3.0 seconds
Elapsed time:  1.50 seconds
```

Assignment 25: การควบคุมจำนวน Threads ใน Pabot ด้วย `--processes`

วัตถุประสงค์: เรียนรู้การควบคุมจำนวน threads ที่ใช้ในการรัน Test Case แบบขนาน

เป้าหมาย: - เข้าใจวิธีการใช้ `--processes` ในการกำหนดจำนวน threads - สามารถปรับแต่งการรัน Test Case แบบ parallel ได้ตามความต้องการ

วิธีการทำ: 1. ใช้ Test Suite จาก Assignment 23 2. รัน Test Suite โดยกำหนดจำนวน threads ที่ต้องการ เช่น: `bash pabot --processes 4 .`

```
Robot Framework remote server at 127.0.0.1:8270 started.
Storing .pabotsuitenames file
2025-02-05 09:19:40.237136 [PID:13808] [0] [ID:0] EXECUTING Asgmt 25.Asgmt 25
2025-02-05 09:19:40.781350 [PID:13808] [0] [ID:0] PASSED Asgmt 25.Asgmt 25 in 0.5 seconds
5 tests, 5 passed, 0 failed, 0 skipped.
==================================================
```

```
Robot Framework remote server at 127.0.0.1:8270 stopped.
PabotLib process stopped
Total testing: 0.50 seconds
Elapsed time:  3.19 seconds
```