



*** Settings ***

Documentation Robot Framework Workshop
... Werkspoorkathedraal 10-06-2020
... MeetUp by deTesters & TestCoders

Resource Rein van der Vegt

Resource Tim Lolkema

*** Variables ***

`${GITHUB}` <https://github.com/tlolkema/robot-workshop>



*** Keywords ***

Robot Framework Presentation

Introduction

Robot Framework basics 1. Settings 2. Variables 3. Test Cases

... 4. Keywords 5. Libraries

Running Tests

Setup & Teardown

SeleniumLibrary

Custom Python Library

Robot Framework Workshop

Exercises



*** Settings ***

*** Variables ***

*** Keywords ***

*** Test Cases ***



*** Settings ***

Documentation Under Settings you can:

... - Import Libraries

... - Import Resource Files

... - Configure Setup & Teardown Keywords

Library SeleniumLibrary

Resource resources.robot

Test Setup Do Something

Test Teardown Do Something Else



*** Variables ***

`${HOST}` localhost:7272

`${LOGIN URL}` http://`${HOST}`/

`${WELCOME URL}` http://`${HOST}`/welcome.html

`${BROWSER}` Firefox



```
*** Test Cases ***
```

```
Go To Homepage And Login  #<-- Test Case
```

```
    Go To Homepage          #<-- Keyword
```

```
    Login With Admin        #<-- Keyword
```

```
    Homepage Should Be Loaded  #<-- Keyword
```



*** Variables ***


`${LOGIN URL}` `http://localhost:5000`

*** Keywords ***

Go To Homepage

Open Browser

Navigate to `${LOGIN URL}`



```
*** Variables ***
```

```
${VAR1}      Lorum
```

```
${VAR2}      ipsum
```

```
${VAR3}      ${VAR1} ${VAR2}
```

```
${INT1}      3
```

```
${INT2}      9
```

```
${INT3} =    Evaluate    ${INT1} + ${INT2}
```

```
${BOOL}      true
```

```
${BOOL2}     ${INT1} > ${INT3}
```




```
*** Keywords ***
```

```
Documentation    What can keywords do:  
...             - Executing low-level keywords  
...             - Receive and return variables  
...             - BuiltIn keywords  
...             - Install libraries for more keywords  
...             - Making custom keywords
```



```
*** Keywords ***
```

```
Keyword Which Sets Test Variable
```

```
    ${RESPONSE} =      Keyword With Return Value      Workshop  
    Set Test Variable    ${RESPONSE}
```

```
Keyword With Return Value
```

```
    [Arguments]    ${ARG1}  
    Log To Console    ${ARG1}  
    [Return]        ${ARG1}
```

```
Validation Keyword
```

```
    Should Be Equal    ${RESPONSE}    Workshop
```



```
*** Test Cases ***
```

```
Testcase Gherkin Syntax
```

```
  Given Step 1      ### Starting situation
```

```
  When Step 2       ### Actions
```

```
  Then Step 3       ### Validations
```



```
*** Settings ***
```

```
Resource    Variables.resource
```

```
Resource    Keywords.resource
```

```
*** Test Cases ***
```

```
Test Case
```

```
Keyword From Resource File
```

```
Log         ${VAR_FROM_RESOURCE}
```



```
*** Test Cases ***
```

```
BuiltIn Keywords Validations
```

```
Should Be True
```

```
1 > 0
```

```
Should Be Equal
```

```
lorum ipsum
```

```
lorum ipsum
```

```
Should Be Equal As Integers
```

```
3
```

```
"3"
```



*** Settings ***

Library	SeleniumLibrary	# Selenium Browser Tests
Library	DebugLibrary	# Debug Robot Framework Tests Via Terminal
Library	OracleDB	# Test Oracle DB

*** Settings ***

Library SeleniumLibrary

*** Keywords ***

Example SeleniumLibrary Test Case

Open Browser chrome http://localhost:5000

Page Should Contain Text Robot Framework

Element Should Not Be Visible xpath=//*[@name="register"]

Input Text id=input_username Test Username

Click Element id=submit_button



```
robot@workshop:~$ robot Testscript.robot
```

```
robot@workshop:~$ robot Tests
```

```
robot@workshop:~$ robot -i smoketest Testscript.robot
```

```
robot@workshop:~$ robot -e smoketest Testscript.robot
```

```
robot@workshop:~$ robot -d test-results/ Testscript.robot
```

```
robot@workshop:~$ robot -v env:PROD Testscript.robot
```




```
*** Settings ***
```

```
Test Setup      KeywordBeforeEveryTest  
Test Teardown   KeywordAfterEveryTest  
Suite Setup     KeywordBeforeEverySuite  
Suite Teardown  KeywordAfterEverySuite
```



```
import requests
```

```
class ExampleLibrary:
```

```
    def example_keyword(self, arg):
```

```
        url = 'localhost:9999/workshop'
```

```
        params = {
```

```
            'param': arg,
```

```
        }
```

```
        response = requests.get(url=url, params=params)
```

```
        return response.text
```



```
*** Settings ***
```

```
Library    ExampleLibrary.py    # Import Your Own Keywords
```

```
*** Keywords ***
```

```
Easily Use Your Own Keywords
```

```
    ${RESPONSE} =    Example Keyword    Argument
```

```
    Should Be Equal As Strings    ${RESPONSE}    Robot!
```



*** Settings ***

Resource resources.resource

Test Template Test Template Keyword

*** Test Cases ***

	FIRST_ARG	SECOND_ARG	THIRD_ARG
Testcase 1	Test	User	Robot
Testcase 2	User	Robot	Test
Testcase 3	Robot	Test	User
Testcase 4	Test	Test	Test
Testcase 5	User	User	User
Testcase 6	Robot	Robot	Robot



```
*** Settings ***
```

```
Documentation      Break & Workshop Time!!!! :)
```

```
...
```

```
...
```

```
...
```

```
...
```

```
...
```

```
...
```

```
...
```

```
( (  
 ) )
```

```
.....  
|       |]  
\       /  
'-----'
```



```
# Create python env
```

```
robot@workshop:~$ python -m venv env
```

```
# Activate env Linux / Mac
```

```
robot@workshop:~$ source env/bin/activate
```

```
# Activate env Windows
```

```
robot@workshop:~$ env\Scripts\activate
```

```
# Install Requirements
```

```
robot@workshop:~$ pip install -r requirements.txt
```

```
# Run Workshop Application
```

```
robot@workshop:~$ python app/app.py
```



Test if you installed python and robot framework correctly:

```
robot@workshop:~$ robot --version
```

You should see output:

Robot Framework x (Python x)

Test if the demo application is running:

Go to <http://localhost:5000/>