สยามชำนาญกิจ
SIAM CHAMNAN KIT

Create your own Library

Hello library

Create file hello.robot

```
*** Settings ***
Library    HelloLibrary.py


*** Testcases ***
Testcase 01
    Say Hi  somkiat

Testcase 02
    Say Hi  somkiat
    Result Should Be  Hi, somkiat
```

Run with robot

$pybot hello.robot

```
[ ERROR ] Error in file '/Users/somkiat/data/slide/robot-framework/advance-robot-
urse/workshop/hello/hello.robot': Test library 'HelloLibrary.py' does not exist.
==============================================================================
Hello
==============================================================================
Testcase 01                                                           | FAIL |
No keyword with name 'Say Hi' found.
------------------------------------------------------------------------------
Testcase 02                                                           | FAIL |
No keyword with name 'Say Hi' found.
------------------------------------------------------------------------------
Hello                                                                 | FAIL |
2 critical tests, 0 passed, 2 failed
2 tests total, 0 passed, 2 failed
==============================================================================
```

Create file HelloLibrary.py

```
class HelloLibrary:

    def __init__(self):
        self._result = ''

    def say_hi(self, name):
        print('Hi, %s' % name)
        self._result = 'Hi, %s' % name

    def result_should_be(self, expected):
        if self._result != expected:
            raise AssertionError('%s != %s' % (self._result,
expected))
```

Create file HelloLibrary.py

```
class HelloLibrary:

    def __init__(self):
        self._result = ''

    def say_hi(self, name):
        print('Hi, %s' % name)
        self._result = 'Hi, %s' % name

    def result_should_be(self, expected):
        if self._result != expected:
            raise AssertionError('%s != %s' % (self._result,
expected))
```

Run with robot

$pybot hello.robot

```
==============================================================================
Hello
==============================================================================
Testcase 01                                                           | PASS |
------------------------------------------------------------------------------
Testcase 02                                                           | PASS |
------------------------------------------------------------------------------
Hello                                                                 | PASS |
2 critical tests, 2 passed, 0 failed
2 tests total, 2 passed, 0 failed
==============================================================================
```

Show log message in console

Show log message in console

```
class HelloLibrary:

    def __init__(self):
        self._result = ''

    def say_hi(self, name):
        print('Hi, %s' % name)
        self._result = 'Hi, %s' % name

    def result_should_be(self, expected):
        if self._result != expected:
            raise AssertionError('%s != %s' % (self._result,
expected))
```

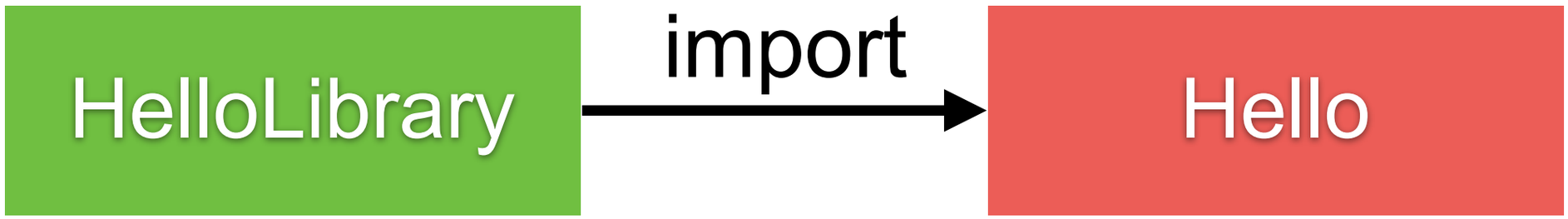
Show log message in console

```python
from robot.api import logger

class HelloLibrary:

    def say_hi(self, name):
        self._hello.set_name(name)
        logger.console('Say hi with %s' %(name))
```

https://github.com/robotframework/robotframework

Separate logic from library file

Separate logic from library file



Create file Hello.py

```
class Hello:
    def __init__(self):
        self._result = ''

    def set_name(self, name):
        self._name = name

    def get_result(self):
        return 'Hi, %s' %(self._name)
```

Update file HelloLibrary.py

```
from hello import Hello

class HelloLibrary:

    def __init__(self):
        self._hello = Hello()
        self._result = ''

    def say_hi(self, name):
        self._hello.set_name(name)

    def result_should_be(self, expected):
        if self._hello.get_result() != expected:
            raise AssertionError('%s != %s' %
(self._result, expected))
```

Update file HelloLibrary.py

```
from hello import Hello

class HelloLibrary:

    def __init__(self):
        self._hello = Hello()
        self._result = ''

    def say_hi(self, name):
        self._hello.set_name(name)

    def result_should_be(self, expected):
        if self._hello.get_result() != expected:
            raise AssertionError('%s != %s' %
(self._result, expected))
```

Update file HelloLibrary.py

```
from hello import Hello

class HelloLibrary:

    def __init__(self):
        self._hello = Hello()
        self._result = ''

    def say_hi(self, name):
        self._hello.set_name(name)

    def result_should_be(self, expected):
        if self._hello.get_result() != expected:
            raise AssertionError('%s != %s' %
(self._result, expected))
```

Run with robot

$pybot hello.robot

```
==============================================================================
Hello
==============================================================================
Testcase 01                                                           | PASS |
------------------------------------------------------------------------------
Testcase 02                                                           | PASS |
------------------------------------------------------------------------------
Hello                                                                 | PASS |
2 critical tests, 2 passed, 0 failed
2 tests total, 2 passed, 0 failed
==============================================================================
```

Improve name of library

Create file hello.robot

```
*** Settings ***
Library    HelloLibrary
```

```
*** Testcases ***
Testcase 01
    Say Hi  somkiat

Testcase 02
    Say Hi  somkiat
    Result Should Be  Hi, somkiat
```

Run with robot

$pybot hello.robot

```
[ ERROR ] Error in file '/Users/somkiat/data/slide/robot-framework/adva
nce-robot-course/workshop/hello/hello.robot': Importing test library 'H
elloLibrary' failed: ModuleNotFoundError: No module named 'HelloLibrary
'

Traceback (most recent call last):
  None
PYTHONPATH:
  /usr/local/Cellar/robot-framework/3.0.2_1/libexec/bin
  /usr/local/Cellar/python/3.6.4_3/Frameworks/Python.framework/Versions
/3.6/lib/python36.zip
  /usr/local/Cellar/python/3.6.4_3/Frameworks/Python.framework/Versions
/3.6/lib/python3.6
  /usr/local/Cellar/python/3.6.4_3/Frameworks/Python.framework/Versions
/3.6/lib/python3.6/lib-dynload
```

Run robot with PythonPath

$pybot -P . hello.robot

```
==============================================================================
Hello
==============================================================================
Testcase 01                                                           | PASS |
------------------------------------------------------------------------------
Testcase 02                                                           | PASS |
------------------------------------------------------------------------------
Hello                                                                 | PASS |
2 critical tests, 2 passed, 0 failed
2 tests total, 2 passed, 0 failed
==============================================================================
```

Custom name of keyword

Change name of keyword

```python
from robot.api.deco import keyword

class HelloLibrary:

    @keyword('Try to say hi with')
    def say_hi(self, name):
```

Use new keyword

```
*** Settings ***
Library    HelloLibrary.py


*** Testcases ***
Testcase 01
    Try to say hi with  somkiat

Testcase 02
    Try to say hi with  somkiat
    Result Should Be  Hi, somkiat
```

Run robot again

$pybot -P . hello.robot

```
==============================================================================
Hello
==============================================================================
Testcase 01                                                           | PASS |
------------------------------------------------------------------------------
Testcase 02                                                           | PASS |
------------------------------------------------------------------------------
Hello                                                                 | PASS |
2 critical tests, 2 passed, 0 failed
2 tests total, 2 passed, 0 failed
==============================================================================
```

Default value of keyword

Default value of keyword

```
def say_hi2(self, name='no name 1', name2='no name 2’):
    self._hello.set_name(name)
```

Default value of keyword

```
*** Settings ***
Library    HelloLibrary.py

*** Testcases ***

Testcase 03
    Say Hi2
    Say Hi2    name1
    Say Hi2    name1    name2
    Say Hi2    name2=name2
    Say Hi2    name=name1
    Say Hi2    name2=name2    name=name1
```

Free style keyword

Free style keyword

```
def say_hi_all(self, **names):

    for name, value in names.items():
        print('%s = %s' % (name, value))
```

Free style keyword

```
*** Settings ***
Library    HelloLibrary.py

*** Testcases ***

Testcase 03
    Say Hi All  key=value   name=somkiat  age=30
```

Embedding arguments into keyword names

http://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#embedding-arguments-into-keyword-name

Add arguments into keyword

```
*** Settings ***
Library   HelloLibrary2

*** Testcases ***
Testcase 01

    Hello somkiat with age 30 year(s)
```

${name} ${age}

Add arguments into keyword

```
from robot.api import logger

from robot.api.deco import keyword

class HelloLibrary2:

    @keyword('Hello ${name} with age ${age:\d+} year(s)')
    def say_hi(self, name, age):

        logger.console('Hello %s with age %s' %(name,
age))
```

More readable and understanding

```
*** Settings ***
Library   HelloLibrary2

*** Testcases ***
Testcase 01

    Hello "somkiat" with age "30" year(s)
```

${name} ${age}

More readable and understanding

```python
from robot.api import logger

from robot.api.deco import keyword

class HelloLibrary2:

    @keyword('Hello "${name}" with age "${age:\d+}" year(s)')
    def say_hi(self, name, age):

        logger.console('Hello %s with age %s' %(name, age))
```

Add document to library

Add document of library

```
from hello import Hello

class HelloLibrary:
    """ Hello Library to *Hello* with name

    Calling from ``set_name`` method
    """
```

Add document of methods

```
def say_hi(self, name):
        """ Say hi with name

        Examples:
        | Say hi | name 1 |
        | Say hi | name 2 |
        """
        self._hello.set_name(name)
```

Add document of methods

```
def result_should_be(self, expected):
        """ Verifies that the current result is ``expected``.

        Examples:
        | Result Should Be | Hi, name 1 |
        | Result Should Be | Hi, name 2 |
        """
        if self._hello.get_result() != expected:
            raise AssertionError('%s != %s' % (self._result, expected))
```

Generate document of library

http://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#libdoc

Generate document of library

$python **-m robot.libdoc** -P . HelloLibrary HelloLibrary.html

Generate document of library

HelloLibrary

Library scope: test case
Named arguments: supported

Introduction

Hello Library to **Hello** with name

Calling from `set_name` method

Shortcuts

Result Should Be · **S**ay Hi

Keywords

Keyword	Arguments	Documentation
Result Should Be	*expected*	Verifies that the current result is `expected`. Examples: Result Should Be Hi, name 1 Result Should Be Hi, name 2
Say Hi	*name*	Say hi with name Examples: Say hi name 1 Say hi name 2

Altogether 2 keywords.
Generated by Libdoc on 2018-08-28 23:12:19.

Need more Knowledges

Basic of Python
Object-Oriented Programming

Return value of keyword

Return value of keyword

Scalar variables
List variables
Dictionary variables
Environment variables