

# **doctest**

Jenny JS Liang (jsliang)  
PyConTW 2013

# It all began with MoSQL...(thank you Mosky!)

- <http://mosql.mosky.tw/>
- 2013年的春節，我在GitHub上看到Mosky開始著手進行MoSQL.....
- 3月的Taipei.py聚會，我問 Mosky:「為什麼妳的MoSQL文件可以寫得這麼快？」
- Mosky:「因為我用**doctest** + Sphinx！」

# A simple sample...

**# add.py**

```
def add(a, b):
```

```
    """
```

```
    This is an addition function.
```

```
>>> add(1,2)
```

```
3
```

```
>>> add("Hello", "World")
```

```
'HelloWorld'
```

```
    """
```

```
    return a+b
```

**\$ python -m doctest -v add.py**

Trying:

```
    add(1,2)
```

Expecting:

```
    3
```

ok

...

1 items passed all tests:

```
    2 tests in add.add
```

2 tests in 2 items.

2 passed and 0 failed.

Test passed.

# Multilines

```
# multilines.py
def multilines(input_str):
    """
    >>> input_str = '''1st
    ... 2nd
    ... 3rd'''
    >>> multilines(input_str)
    1st
    2nd
    3rd
    """
    print input_str
```

```
$ python -m doctest -v \
    multilines.py
```

Trying:

```
input_str = '''1st
2nd
3rd'''
```

Expecting nothing  
ok

Trying:

```
prnt_multilines(input_str)
```

...

Test passed.

# Handling blank lines...

**# blank.py**

```
def blankline(line1, line2):
```

```
    """
```

```
    Print "line1\n\nline2".
```

```
    >>> blankline("Hello", "World")
```

```
    Hello
```

```
    <BLANKLINE>
```

```
    World
```

```
    """
```

```
    print line1
```

```
    print ""
```

```
    print line2
```

```
$ python -m doctest -v \
    blank.py
```

```
Trying:
```

```
    blankline("Hello", "World")
```

```
Expecting:
```

```
    Hello
```

```
    <BLANKLINE>
```

```
    World
```

```
ok
```

```
...
```

```
Test passed.
```

# Checking Examples in Docstrings

End your module with the following code:

```
if __name__ == "__main__":  
    import doctest  
    doctest.testmod()
```

and execute `python <module_name>.py` .

# **Autodocumenting with Sphinx**

## Table Of Contents

Welcome to mymodule's  
documentation!  
[Indices and tables](#)

## This Page

[Show Source](#)

## Quick search

Go

Enter search terms or a module,  
class or function name.

# Welcome to mymodule's documentation!

## add.py

mymodule.add. **add**(*a*, *b*)

This is an addition function. >>> add(1,2) 3 >>> add("Hello", "World")  
'HelloWorld'

## blank.py

mymodule.blank. **blankline**(*line1*, *line2*)

Print "line1

line2".

```
>>> blankline("Hello", "World")  
Hello  
  
World
```

## multilines.py

mymodule.multilines. **multilines**(*input\_str*)

```
>>> input_str = '''1st  
... 2nd  
... 3rd'''  
>>> multilines(input_str)  
1st  
2nd  
3rd
```

## Indices and tables



# Module Structure

- mymodule/
  - `__init__.py`
  - `add.py`
  - `blank.py`
  - `multiline.py`

```
$ export PYTHONPATH=$PYTHONPATH:  
/path/to/  
mymodule
```

# Modify index.rst (use ReST)

Welcome to mymodule's documentation!

=====

```
.. toctree::  
    :maxdepth: 2
```

```
.. automodule:: mymodule.add  
    :members:
```

```
.. automodule:: mymodule.blank  
    :members:
```

# Generating documentation with Sphinx

1. `$ pip install sphinx`
2. `$ sphinx-quickstart`
3. Please indicate if you want to use one of the following Sphinx extensions:  
**> autodoc: automatically insert docstrings from modules (y/N) [n]: y**
4. modify `index.rst`
5. `$ make html`

# References

- Python doctest
  - <http://docs.python.org/3/library/doctest.html>
- Documenting Your Project Using Sphinx
  - [http://pythonhosted.org/an\\_example\\_pypi\\_project/sphinx.html](http://pythonhosted.org/an_example_pypi_project/sphinx.html)
- Minimal Sphinx setup for autodocumenting Python modules
  - <http://scienceoss.com/minimal-sphinx-setup-for-autodocumenting-python-modules/>
- sphinx.ext.autodoc – Include documentation from docstrings
  - <http://sphinx-doc.org/ext/autodoc.html>