Mastering Python

الدرس # 9_3

Testing Python Code

DocTest

فحص واختبار برمجيات بايثون

By:
Hussam Hourani

V1.0 -NOV 2019

Agenda

- DocTest Example

What is doctest

doctest is a test framework that comes prepackaged with Python.

The doctest module searches for pieces of text that look like interactive Python sessions inside of the documentation parts of a module, and then executes (or reexecutes) the commands of those sessions to verify that they work exactly as shown, i.e. that the same results can be achieved

pip install doctest

DocTest Rules

'''

Module showing how doctests can be included with source code

Each '>>>' line is run as if in a python shell, and counts as a test.

The next line, if not '>>>' is the expected output of the previous line.

If anything doesn't match exactly (including trailing spaces), the test fails.

'''

Example

```
1  import doctest
2
3  def sum(a,b):
4      """
5      Calculates the sum of a and b
6
7      >>> sum(1,1)
8      2
9      >>> sum(1,-1)
10     0
11     >>> sum(-1,-1)
12     -2
13     >>> sum(0,-10)
14     -10
15     >>>
16
17     """
18
19     return a+b
20
21 if __name__ == "__main__":
22     doctest.testmod()
```

Doc Testing example

```python
import doctest

def sum(a,b):
    """
    Calculates the sum of a and b

    >>> sum(1,1)
    2
    >>> sum(1,-1)
    0
    >>> sum(-1,-1)
    -2
    >>> sum(0,-10)
    -10
    >>>

    """

    return a + b

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

Output

```
In [33]: runfile('C:/Python36/Mystuff/doctest_1.py', wdir='C:/Python36/Mystuff')

In [34]:
```

Doc Testing – Pass Results

```python
import doctest

def sum(a,b):
    """
    Calculates the sum of a and b

    >>> sum(1,1)
    2
    >>> sum(1,-1)
    0
    >>> sum(-1,-1)
    -2
    >>> sum(0,-10)
    -10
    >>>

    """

    return a+b

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

python -m doctest -v doctest_1.py

Output

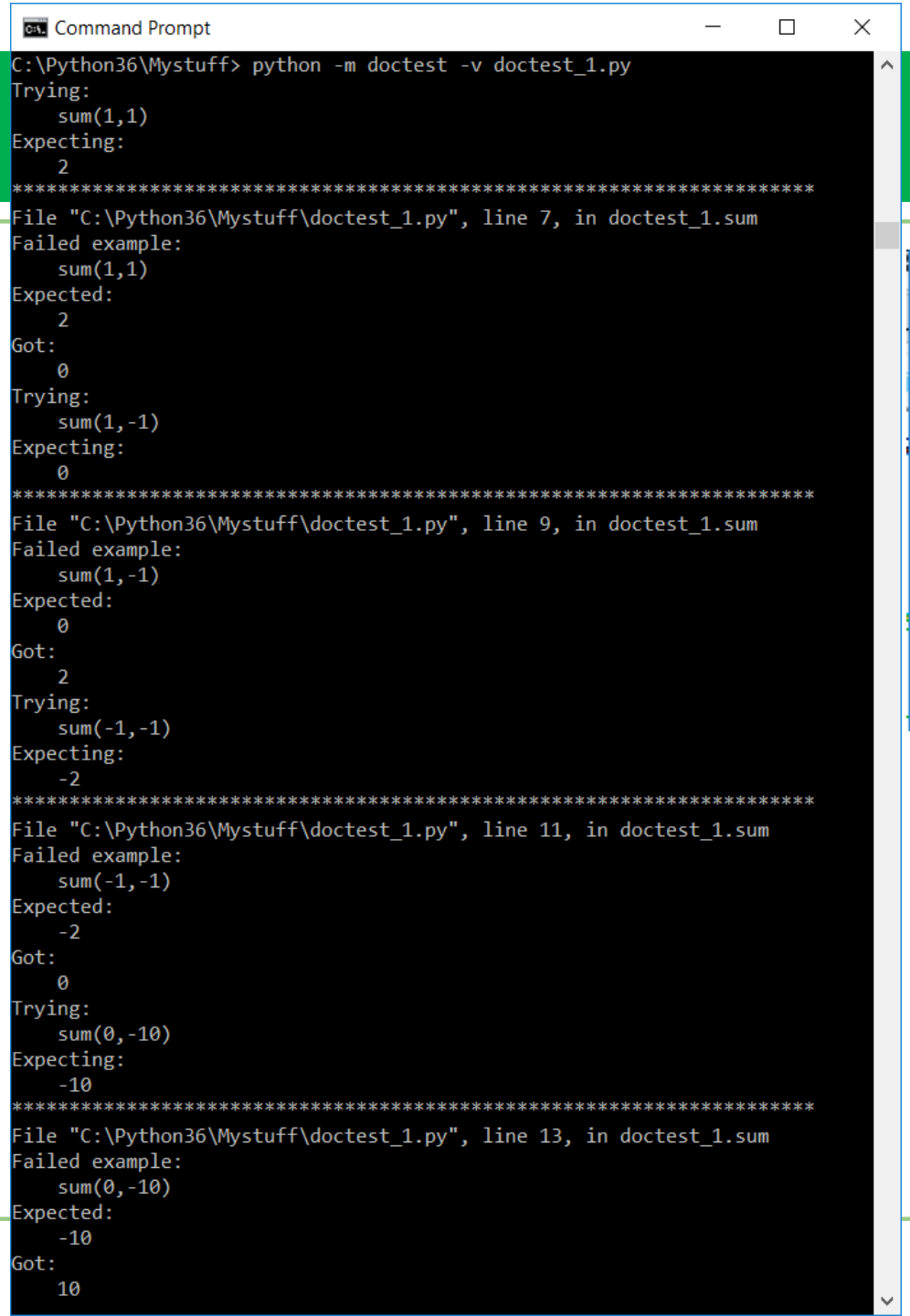
Command Prompt

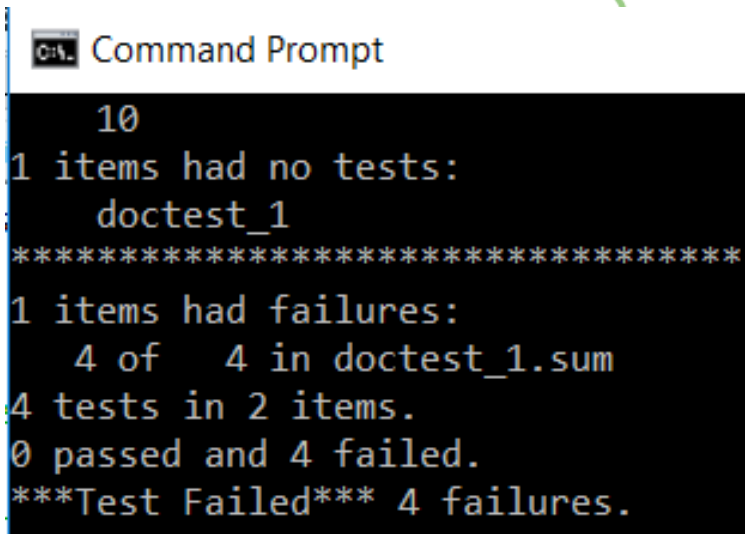
```
C:\Python36\Mystuff> python -m doctest -v doctest_0.py
Trying:
    sum(1,1)
Expecting:
    2
ok
Trying:
    sum(1,-1)
Expecting:
    0
ok
Trying:
    sum(-1,-1)
Expecting:
    -2
ok
Trying:
    sum(0,-10)
Expecting:
    -10
ok
1 items had no tests:
    doctest_0
1 items passed all tests:
   4 tests in doctest_0.sum
4 tests in 2 items.
4 passed and 0 failed.
Test passed.

C:\Python36\Mystuff>
```

Doctest fails

Output

```python
import doctest

def sum(a,b):
    """
    Calculates the sum of a and b

    >>> sum(1,1)
    2
    >>> sum(1,-1)
    0
    >>> sum(-1,-1)
    -2
    >>> sum(0,-10)
    -10
    >>>

    """

    return a - b

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





Doc Testing

```python
import doctest

def sum(a,b):
    """
    Calculates the sum of a and b

    >>> sum(1,1)
    2
    >>> sum(1,-1)
    0
    >>> sum(-1,-1)
    -2
    >>> sum(0,-10)
    -10
    >>>

    """

    return a - b

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

Output

```
**********************************************************************
File "C:/Python36/Mystuff/doctest_1.py", line 7, in __main__.sum
Failed example:
    sum(1,1)
Expected:
    2
Got:
    0
**********************************************************************
File "C:/Python36/Mystuff/doctest_1.py", line 9, in __main__.sum
Failed example:
    sum(1,-1)
Expected:
    0
Got:
    2
**********************************************************************
File "C:/Python36/Mystuff/doctest_1.py", line 11, in __main__.sum
Failed example:
    sum(-1,-1)
Expected:
    -2
Got:
    0
**********************************************************************
File "C:/Python36/Mystuff/doctest_1.py", line 13, in __main__.sum
Failed example:
    sum(0,-10)
Expected:
    -10
Got:
    10
**********************************************************************
1 items had failures:
   4 of   4 in __main__.sum
***Test Failed*** 4 failures.

In [33]:
```

By : Hussam Hourani

Doc Testing – Function Sample

```python
import doctest

def Largest(first,second):
    """
    Calculates the largest of first and second
    >>> Largest(4,5)
    5
    >>> Largest(5,4)
    5
    >>> Largest(5,5)
    5
    """
    if (first > second):
        return first
    return second

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

Output

```
In [10]: run

In [11]:
```

```python
import doctest

def Largest(first,second):
    """
    Calculates the largest of first and second
    >>> Largest(4,5)
    5
    >>> Largest(5,4)
    5
    >>> Largest(5,5)
    5
    """
    if (first < second):
        return first
    return second

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

```

Output

```
File "C:/Python36/Mystuff/doctest_2.py", line 6, in __main__.Largest
Failed example:
    Largest(4,5)
Expected:
    5
Got:
    4
**********************************************************************
File "C:/Python36/Mystuff/doctest_2.py", line 8, in __main__.Largest
Failed example:
    Largest(5,4)
Expected:
    5
Got:
    4
**********************************************************************
1 items had failures:
   2 of   3 in __main__.Largest
***Test Failed*** 2 failures.
```

Hussam has over 23 years of Organizations Transformation, VROs, PMO, Large Scale and Enterprise Programs Global Delivery, leadership, Business Development and Management. His client experience is wide ranging across many sectors but focuses on Performance Enhancement, Transformation, Program Management and Data Science.