

文档测试

403次阅读

如果你经常阅读Python的官方文档，可以看到很多文档都有示例代码。比如[re模块](#)就带了很多示例代码：

```
>>> import re
>>> m = re.search('(?<=abc)def', 'abcdef')
>>> m.group(0)
'def'
```

可以把这些示例代码在Python的交互式环境下输入并执行，结果与文档中的示例代码显示的一致。

这些代码与其他说明可以写在注释中，然后，由一些工具来自动生成文档。既然这些代码本身就可以粘贴出来直接运行，那么，可不可以自动执行写在注释中的这些代码呢？

答案是肯定的。

当我们编写注释时，如果写上这样的注释：

```
def abs(n):
    """
    Function to get absolute value of number.

    Example:

    >>> abs(1)
    1
    >>> abs(-1)
    1
    >>> abs(0)
    0
    """
    return n if n >= 0 else (-n)
```

无疑更明确地告诉函数的调用者该函数的期望输入和输出。

并且，Python内置的“文档测试”（doctest）模块可以直接提取注释中的代码并执行测试。

doctest严格按照Python交互式命令行的输入和输出来判断测试结果是否正确。只有测试异常的时候，可以用...表示中间一大段烦人的输出。

让我们用doctest来测试上次编写的Dict类：

```
class Dict(dict):
    """
    Simple dict but also support access as x.y style.

    >>> d1 = Dict()
    >>> d1['x'] = 100
    >>> d1.x
    100
    >>> d1.y = 200
    >>> d1['y']
    200
    >>> d2 = Dict(a=1, b=2, c='3')
    >>> d2.c
```

```

'3'
>>> d2['empty']
Traceback (most recent call last):
...
KeyError: 'empty'
>>> d2.empty
Traceback (most recent call last):
...
AttributeError: 'Dict' object has no attribute 'empty'
,,,
def __init__(self, **kw):
    super(Dict, self).__init__(**kw)

def __getattr__(self, key):
    try:
        return self[key]
    except KeyError:
        raise AttributeError(r"'Dict' object has no attribute '%s'" % key)

def __setattr__(self, key, value):
    self[key] = value

if __name__ == '__main__':
    import doctest
    doctest.testmod()

```

运行python mydict.py:

```
$ python mydict.py
```

什么输出也没有。这说明我们编写的doctest运行都是正确的。如果程序有问题，比如把__getattr__()方法注释掉，再运行就会报错：

```

$ python mydict.py
*****
File "mydict.py", line 7, in __main__.Dict
Failed example:
    d1.x
Exception raised:
    Traceback (most recent call last):
        ...
    AttributeError: 'Dict' object has no attribute 'x'
*****
File "mydict.py", line 13, in __main__.Dict
Failed example:
    d2.c
Exception raised:
    Traceback (most recent call last):
        ...
    AttributeError: 'Dict' object has no attribute 'c'
*****

```

注意到最后两行代码。当模块正常导入时，doctest不会被执行。只有在命令行运行时，才执行doctest。所以，不必担心doctest会在非测试环境下执行。

小结

doctest非常有用，不但可以用来测试，还可以直接作为示例代码。通过某些文档生成工具，就可以自动把包含doctest的注释提取出来。用户看文档的时候，同时也看到了doctest。