python 第四部分

模块与包

什么是模块?

模块就是python程序，一段python代码，一个.py文件

简单的使用

import ...

```
>>> import sys
>>> sys.path
['', '/Users/zhangle/pyvirtualenv/python2.7/lib/python27.zip', '/Users/zhangle/pyvirtualenv/python2.7/lib/python2.7', '/Users/zhangle/pyvirt
ualenv/python2.7/lib/python2.7/plat-darwin', '/Users/zhangle/pyvirtualenv/python2.7/lib/python2.7/plat-mac', '/Users/zhangle/pyvirtualenv/py
thon2.7/lib/python2.7/plat-mac/lib-scriptpackages', '/Users/zhangle/pyvirtualenv/python2.7/Extras/lib/python', '/Users/zhangle/pyvirtualenv/
python2.7/lib/python2.7/lib-tk', '/Users/zhangle/pyvirtualenv/python2.7/lib/python2.7/lib-old', '/Users/zhangle/pyvirtualenv/python2.7/lib/p
ython2.7/lib-dynload', '/System/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7', '/System/Library/Frameworks/Python.framewor
k/Versions/2.7/lib/python2.7/plat-darwin', '/System/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/lib-tk', '/System/Library
/Frameworks/Python.framework/Versions/2.7/lib/python2.7/plat-mac', '/System/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/p
lat-mac/lib-scriptpackages', '/Users/zhangle/pyvirtualenv/python2.7/lib/python2.7/site-packages']
```

from ... import ...

```
>>> from sys import path
>>> path
['', '/Users/zhangle/pyvirtualenv/python2.7/lib/python27.zip', '/Users/zhangle/pyvirtualenv/python2.7/lib/python2.7', '/Users/zhangle/pyvirt
ualenv/python2.7/lib/python2.7/plat-darwin', '/Users/zhangle/pyvirtualenv/python2.7/lib/python2.7/plat-mac', '/Users/zhangle/pyvirtualenv/py
thon2.7/lib/python2.7/plat-mac/lib-scriptpackages', '/Users/zhangle/pyvirtualenv/python2.7/Extras/lib/python', '/Users/zhangle/pyvirtualenv/
python2.7/lib/python2.7/lib-tk', '/Users/zhangle/pyvirtualenv/python2.7/lib/python2.7/lib-old', '/Users/zhangle/pyvirtualenv/python2.7/lib/p
ython2.7/lib-dynload', '/System/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7', '/System/Library/Frameworks/Python.framewor
k/Versions/2.7/lib/python2.7/plat-darwin', '/System/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/lib-tk', '/System/Library
/Frameworks/Python.framework/Versions/2.7/lib/python2.7/plat-mac', '/System/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/p
lat-mac/lib-scriptpackages', '/Users/zhangle/pyvirtualenv/python2.7/lib/python2.7/site-packages']
```

自己的模块

demo.py

```
1  #encoding:utf-8
2
3  '''
4  这是python代码，将这个文件保存为demo.py
5  '''
6  print '我是一个模块'
```

```
>>> import demo
我是一个模块
```

查看模块文档

```
>>> print demo.__doc__
这是python代码，将这个文件保存为demo.py
```

使用模块中的属性

```python
#encoding:utf-8

'''
这是python代码，将这个文件保存为demo.py
'''
a = 'anjuke'
```

```
>>> import demo
>>> demo.a
'anjuke'
>>>
```

使用模块中的函数

```
1  #encoding:utf-8
2
3  '''
4  这是python代码，将这个文件保存为demo.py
5  '''
6  def func():
7      print 'this is anjuke'
```

```
>>> reload(demo)
<module 'demo' from 'demo.py'>
>>> demo.func()
this is anjuke
```

使用模块中的类

```python
#encoding:utf-8

'''
这是python代码，将这个文件保存为demo.py
'''
class DemoClass:
    def __init__(self):
        print 'I am a class'
```

```
>>> reload(demo)
<module 'demo' from 'demo.pyc'>
>>> d = demo.DemoClass()
I am a class
```

导入指定属性

```python
#encoding:utf-8

'''
这是python代码，将这个文件保存为demo.py
'''
def func1():
    print 'I am func1'

def func2():
    print 'I am func2'
```

```
>>> from demo import func1
>>> func1()
I am func1
>>> func2()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'func2' is not defined
```

```
>>> from demo import func1,func2
>>> func1()
I am func1
>>> func2()
I am func2
```

别名

```
#encoding:utf-8

'''
这是python代码，将这个文件保存为demo.py
'''
def func1():
    print 'I am func1'

def func2():
    print 'I am func2'
```

```
>>> from demo import func1 as f
>>> f()
I am func1
```

模块中属性的封装

```python
#encoding:utf-8

'''
这是python代码，将这个文件保存为demo.py
'''
def func1():
    print 'I am func1'

def func2():
    print 'I am func2'

def func3():
    print 'I am func3'

__all__ = ['func1']
```

```
>>> from demo import *
>>> func1()
I am func1
>>> func2()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'func2' is not defined
>>> func3()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'func3' is not defined
>>>
Type "help", "copyri
>>> import demo
>>> demo.func2()
I am func2
>>>
```

__all__，_或__开头的属性只在使用from ... import *时起作用

单例模式的模块实现

python程序运行过程中，模块只会导入一次，由此可知，模块本身就实现了单例模式

demo.py

```
#encoding:utf-8

'''
这是python代码，将这个文件保存为demo.py
'''
__a = 11

def get_attr():
    return __a

def set_attr(value):
    global __a
    __a = value
```

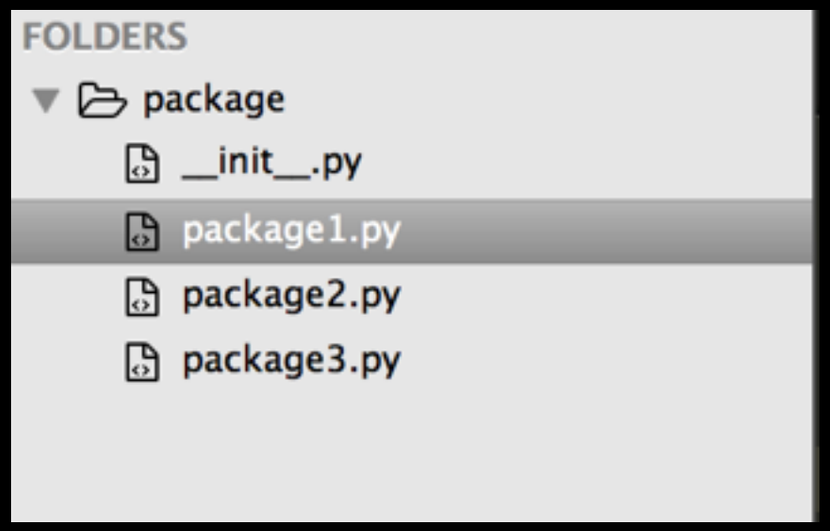
use1.py

```
#encoding:utf-8
import demo
import use2

class Use1:
    def __init__(self):
        demo.set_attr(20)
        self.u = use2.Use2()
        print demo.get_attr()

if __name__ == '__main__':
    u = Use1()
```

use2.py

```
#encoding:utf-8
import demo

class Use2:
    def __init__(slef):
        demo.set_attr(50)
```

包——由多个模块



package1.py

```
#edcoding:utf-8

package = 'package1'

def sayName():
    print 'I am %s'%package
```

package1.py | package2.py

```
#edcoding:utf-8

package = 'package2'

def sayName():
    print 'I am %s'%package
```

package1.py | package2.py | package3.py

```
#edcoding:utf-8

package = 'package3'

def sayName():
    print 'I am %s'%package

```

```
>>> from demo_package import package1
>>> package1.sayName()
I am package1
>>> from demo_package.package2 import sayName
>>> sayName()
I am package2
```