

nsd1905_py02_day03

备份练习

```
# mkdir /tmp/demo/  
# cp -r /etc/security/ /tmp/demo/  
# 要备份的目录是/tmp/demo/security，备份的目标是/tmp/demo/backup  
# mkdir /tmp/demo/backup
```

OOP：面向对象编程

术语

类：class。用来描述具有相同的属性和方法的对象的集合。

实例：对象。通过类创建出具体的实例。

方法：类中的函数。

组合

两个类明显不同，其中一个类是另一个类的组件，使用组合。

子类

当两个类有很多相同，只有部分不同时，采用子类。通过父类派生子类，子类继承于父类。

子类可以有多个父类。子类将会继承所有父类的方法。如果有同名方法，查找顺序是自下向上，自左向右。

特殊方法

在class中，有很多以双下划线开头、结尾的方法，这些方法称作magic，魔法方法。常用的有三个魔法方法。

正则表达式

示例，为mac地址添加冒号

```
192.168.1.1      00000C123456  
192.168.1.2      52541B9C23A3  
# 思路：1、定位mac地址  2、每两个数字分一组  3、在组之间添加冒号  
# 1.行尾、12个任意字符  
# 2.每两个字符用()分组  
# 3.组之间加冒号  
:%s\((..\)\((..\)\((..\)\((..\)\((..\)\((..\))$/\1:\2:\3:\4:\5:\6/
```

```
>>> import re
>>> re.match('f..', 'food') # 在food的开头匹配f..匹配到，返回匹配对象
<_sre.SRE_Match object; span=(0, 3), match='foo'>
>>> print(re.match('f..', 'seafood')) # 没匹配到，返回None
None

>>> re.search('f..', 'seafood') # 在字符串中匹配f..
<_sre.SRE_Match object; span=(3, 6), match='foo'>
>>> m = re.search('f..', 'seafood')
>>> m.group() # 取出匹配到的内容
'foo'

>>> re.findall('f..', 'seafood is food')
['foo', 'foo']
# finditer返回由匹配对象构成的生成器
>>> for m in re.finditer('f..', 'seafood is food'):
...     m.group()

# 以句点或-作为分隔符，切隔字符串
>>> re.split('\.|-', 'hello-world.tar.gz')
['hello', 'world', 'tar', 'gz']
# 把字符串中的X替换成zzg
>>> re.sub('X', 'zzg', 'Hi X. ni hao X.')
'Hi zzg. ni hao zzg.'

# 如果有大量匹配的时候，把正则表达式模式先编译，会有更好的效率
>>> patt = re.compile('f..')
>>> m = patt.search('seafood')
>>> m.group()
'foo'
>>> patt.findall('seafood is food')
['foo', 'foo']
```