nsd_1908_py02_day03

标识符命名网站（支持汉字）：https://unbug.github.io/codelf/

hashlib模块

- 用于计算数据的hash值

```
>>> import hashlib
# 计算123456的md5值
>>> m = hashlib.md5(b'123456')
>>> m.hexdigest()
'e10adc3949ba59abbe56e057f20f883e'

>>> m1 = hashlib.md5()
>>> m1.update(b'12')
>>> m1.update(b'34')
>>> m1.update(b'56')
>>> m1.hexdigest()
'e10adc3949ba59abbe56e057f20f883e'
```

tarfile模块

- 用于压缩、解压缩

```
>>> import tarfile
# 压缩
>>> tar = tarfile.open('/tmp/nsd1908/demo/a.tar.gz', 'w:gz')
>>> tar.add('/etc/hosts')
>>> tar.add('/etc/security')
>>> tar.close()

# 解压
>>> tar = tarfile.open('/tmp/nsd1908/demo/a.tar.gz')
>>> tar.extractall(path='/tmp/nsd1908/demo')
>>> tar.close()
```

通过os.walk拼接路径

```
>>> for path, folders, files in os.walk('/tmp/nsd1908/security'):
...   for file in files:
...     os.path.join(path, file)
```

OOP:面向对象编程

- 在python中，一切皆对象
- 不同的对象有不同的属性
    - 静态的属性，表现为变量
    - 动态的属性，表现为方法（函数）

组合

- 当两个类明显不同，其中一个类是另一个类的组件的时候，使用组合

子类继承

- 两个类有很多一致的地方，只有少部分不同，可以使用继承。经常说，父类派生子类，子类继承于父类

多重继承

- 子类可以有多个父类
- 子类可以继承所有父类的方法
- 查找方法的顺序是自下向上，自左向右。

正则表达式

```
# 给mac地址加冒号
192.168.1.1     000C29123456
192.168.1.2     5254A382C80B

# 1. 定位、取出mac地址； 2. 每两个数分一组； 3. 各组间加冒号
:%s/\(..\)\(..\)\(..\)\(..\)\(..\)\(..\)$/\1:\2:\3:\4:\5:\6/
```

re模块

```
>>> import re
# match从字符串开头匹配，如果匹配到，返回匹配对象；否则返回None
>>> re.match('f..', 'food')
<_sre.SRE_Match object; span=(0, 3), match='foo'>
>>> print(re.match('f..', 'seafood'))
None

>>> re.search('f..', 'seafood')
<_sre.SRE_Match object; span=(3, 6), match='foo'>
>>> m = re.search('f..', 'seafood')
>>> m.group()  # 返回匹配到的内容
'foo'

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

```
'foo'
'foo'

# re.split用于切割字符串
>>> re.split('-|\.', 'hello-world-how-are-you.tar.gz')
['hello', 'world', 'how', 'are', 'you', 'tar', 'gz']

# re.sub用于查找替换
>>> re.sub('X', 'tom', 'Hi X. Nice to meet you, X')
'Hi tom. Nice to meet you, tom'

# 在有大量匹配的情况下，先把正则表达式模式进行编译，会有更好的执行效率
>>> patt = re.compile('f..')
>>> m = patt.search('seafood')
>>> m.group()
'foo'
>>> patt.findall('seafood is food')
['foo', 'foo']
```