Python基础 2

1. 字符串处理
2. 模块（导入，常用模块）
3. 列表进阶
4. 字典进阶
5. 函数

1. 字符串处理

python的字符串内建函数

常用方法	描述
string.count(str, beg=0, end=len(string))	返回 str 在 string 里面出现的次数，如果 begin 或者 end 指定则返回指定范围内 str 出现的次数
string.startswith(obj, beg=0,end=len(string))	检查字符串是否是以 obj 开头，是则返回 True，否则返回 False。如果beg 和 end 指定值，则在指定范围内检查.
string.endswith(obj, beg=0, end=len(string))	检查字符串是否以 obj 结束，如果beg 或者 end 指定则检查指定的范围内是否以 obj 结束，如果是，返回 True,否则返回 False.
string.find(str, beg=0, end=len(string))	检测 str 是否包含在 string 中，如果 beg 和 end 指定范围，则检查是否包含在指定范围内，如果是返回开始的索引值，否则返回-1
string.join(seq)	以 string 作为分隔符，将 seq 中所有的元素(的字符串表示)合并为一个新的字符串
string.replace(str1, str2, num=string.count(str1))	把 string 中的 str1 替换成 str2,如果 num 指定，则替换不超过 num 次.
string.split(str="", num=string.count(str))	以 str 为分隔符切片 string，如果 num 有指定值，则仅分隔 num+ 个子字符串
string.strip([obj])	在 string 上执行 lstrip()和 rstrip();string.rstrip()删除 string 字符串末尾的空格.string.lstrip()截掉 string 左边的空格
string.upper()	转换 string 中的小写字母为大写
string.lower()	转换 string 中所有大写字符为小写.

方法	描述
string.capitalize()	把字符串的第一个字符大写
string.center(width)	返回一个原字符串居中,并使用空格填充至长度 width 的新字符串
string.decode(encoding='UTF-8', errors='strict')	以 encoding 指定的编码格式解码 string，如果出错默认报一个 ValueError 的 异 常 ， 除非 errors 指 定 的 是 'ignore' 或 者'replace'
string.encode(encoding='UTF-8', errors='strict')	以 encoding 指定的编码格式编码 string，如果出错默认报一个ValueError 的异常，除非 errors 指定的是'ignore'或者'replace'
string.expandtabs(tabsize=8)	把字符串 string 中的 tab 符号转为空格，tab 符号默认的空格数是 8。
string.format()	格式化字符串
string.index(str, beg=0, end=len(string))	跟find()方法一样，只不过如果str不在 string中会报一个异常.
string.isalnum()	如果 string 至少有一个字符并且所有字符都是字母或数字则返回 True,否则返回 False
string.isalpha()	如果 string 至少有一个字符并且所有字符都是字母则返回 True,否则返回 False

方法	描述
string.isdecimal()	如果 string 只包含十进制数字则返回 True 否则返回 False.
string.isdigit()	如果 string 只包含数字则返回 True 否则返回 False.
string.islower()	如果 string 中包含至少一个区分大小写的字符，并且所有这些(区分大小写的)字符都是小写，则返回 True，否则返回 False
string.isnumeric()	如果 string 中只包含数字字符，则返回 True，否则返回 False
string.isspace()	如果 string 中只包含空格，则返回 True，否则返回 False.
string.istitle()	如果 string 是标题化的(见 title())则返回 True，否则返回 False
string.isupper()	如果 string 中包含至少一个区分大小写的字符，并且所有这些(区分大小写的)字符都是大写，则返回 True，否则返回 False
string.ljust(width)	返回一个原字符串左对齐,并使用空格填充至长度 width 的新字符串
string.maketrans(intab, outtab])	maketrans() 方法用于创建字符映射的转换表，对于接受两个参数的最简单的调用方式，第一个参数是字符串，表示需要转换的字符，第二个参数也是字符串表示转换的目标。
max(str)	返回字符串 str 中最大的字母。
min(str)	返回字符串 str 中最小的字母。
string.partition(str)	有点像 find()和 split()的结合体,从 str 出现的第一个位置起,把 字 符 串 string 分 成 一 个 3 元 素 的 元 组 (string_pre_str,str,string_post_str),如果 string 中不包含str 则 string_pre_str == string.
string.rfind(str, beg=0,end=len(string))	类似于 find()函数，不过是从右边开始查找.
string.rindex(str, beg=0,end=len(string))	类似于 index()，不过是从右边开始.
string.rjust(width)	返回一个原字符串右对齐,并使用空格填充至长度 width 的新字符串
string.rpartition(str)	类似于 partition()函数,不过是从右边开始查找
string.splitlines([keepends])	按照行('\r', '\r\n', \n')分隔，返回一个包含各行作为元素的列表，如果参数 keepends 为 False，不包含换行符，如果为 True，则保留换行符。
string.swapcase()	翻转 string 中的大小写
string.title()	返回"标题化"的 string,就是说所有单词都是以大写开始，其余字母均为小写(见 istitle())
string.translate(str, del="")	根据 str 给出的表(包含 256 个字符)转换 string 的字符,要过滤掉的字符放到 del 参数中
string.zfill(width)	返回长度为 width 的字符串，原字符串 string 右对齐，前面填充0

In [12]:

```
s="""
   The villagers of Little Hangleron still called it "the Riddle House," even though it had been
many years since the Riddle family had lived there. It stood on a hill overlooking the village,
some of its windows boarded, tiles missing from its roof, and ivy spreading unchecked over its
face. Once a fine-looking manor, and easily the largest and grandest building for miles around,
the Riddle House was now damp, derelict, and unoccupied.
"""
s.count('r')#该方法区分大小写
s.lower().count('r')#该方法区分大小写
```

Out[12]:

In [15]:

```
s.startswith("T")
s.strip().startswith("T")
```

Out[15]:

```
True
```

In [20]:

```
s.split('\n')
' '.join(s.split('\n')).strip()
```

Out[20]:

```
'The villagers of Little Hangleron still called it "the Riddle House," even though i
t had been many years since the Riddle family had lived there. It stood on a hill ov
erlooking the village, some of its windows boarded, tiles missing from its roof, and
ivy spreading unchecked over its face. Once a fine-looking manor, and easily the lar
gest and grandest building for miles around, the Riddle House was now damp, derelic
t, and unoccupied.'
```

思考题：

- 根据上节课讲的题目，如何统计harry potter文章中，出现的harry potter名字的频率，不区分大小写，不会受错误换行影响

In [ ]:

```

```

2. 模块（导入，常用模块）

2.1 什么是模块

- Python 模块(Module)，是一个 Python 文件，以 .py 结尾，包含了 Python 对象定义和Python语句。
- 模块能定义函数，类和变量，模块里也能包含可执行的代码。
- 已经写好的工具包

2.2 模块导入

- 模块定义好后，我们可以使用 import 语句来引入模块，语法如下：

```
import module1[, module2[,... moduleN]]
```

- 比如要引用模块 math，就可以在文件最开始的地方用 import math 来引入。在调用 math 模块中的函数时，必须这样引用：

```
模块名.函数名 math.sqrt(4)
```

- from…import 语句

- Python 的 from 语句让你从模块中导入一个指定的部分到当前命名空间中。语法如下：

> from modname import name1[, name2[, ... nameN]]

- from…import* 语句
  - 把一个模块的所有内容全都导入到当前的命名空间也是可行的，只需使用如下声明

> from modname import *

2.3 常用模块

- datetime,date,time模块
- random模块
- string模块
- re模块
- math模块
- Numpy
- Matplotlib
- Pandas
- scikit-learn
- request

In [46]:

```
import math
math.sqrt(4)
```

Out[46]:

2.0

In [47]:

```
from collections import defaultdict
s=defaultdict()
```

3. 列表

- 更新列表
  - 你可以对列表的数据项进行修改或更新，你也可以使用append()方法来添加列表项，如下所示：
- 删除列表元素
  - 可以使用 del 语句来删除列表的元素
- 操作符
  - 列表对 + 和 * 的操作符与字符串相似。+ 号用于组合列表，* 号用于重复列表。

Python表达式	结果	描述
len([1, 2, 3])	3	长度
[1, 2, 3] + [4, 5, 6]	[1, 2, 3, 4, 5, 6]	组合
['Hi!'] * 4	['Hi!', 'Hi!', 'Hi!', 'Hi!']	重复
3 in [1, 2, 3]	True	元素是否存在于列表中

Python表达式	结果	描述
for x in [1, 2, 3]: print(x)	1 2 3	迭代

- Python列表截取
  - 列表根据index下标访问元素
- 列表函数&方法 -Python包含以下函数和方法:

序号	函数	含义
1	cmp(list1, list2)	比较两个列表的元素
2	len(list)	列表元素个数
3	max(list)	返回列表元素最大值
4	min(list)	返回列表元素最小值
5	list(seq)	将元组转换为列表

序号	方法	含义
1	list.append(obj)	在列表末尾添加新的对象
2	list.count(obj)	统计某个元素在列表中出现的次数
3	list.extend(seq)	在列表末尾一次性追加另一个序列中的多个值（用新列表扩展原来的列表）
4	list.index(obj)	从列表中找出某个值第一个匹配项的索引位置
5	list.insert(index, obj)	将对象插入列表
6	list.pop([index=-1])	移除列表中的一个元素（默认最后一个元素），并且返回该元素的值
7	list.remove(obj)	移除列表中某个值的第一个匹配项
8	list.reverse()	反向列表中元素
9	list.sort(cmp=None, key=None, reverse=False)	对原列表进行排序

In [48]:

```
#更新列表
list = []          ## 空列表
list.append('Google')   ## 使用 append() 添加元素
list.append('Runoob')
print(list)
#删除元素
list1 = ['physics', 'chemistry', 1997, 2000]
print(list1)
del list1[2]
print("After deleting value at index 2 : ")
print(list1)
```

```
['Google', 'Runoob']
['physics', 'chemistry', 1997, 2000]
After deleting value at index 2 :
['physics', 'chemistry', 2000]
```

In [49]:

```
L = ['Google', 'Runoob', 'Taobao']
L[2]
L[-2]
L[1:]
```

Out[49]:

['Runoob', 'Taobao']

4. 字典

- 访问字典里的值
  - 把相应的键放入熟悉的方括弧
  - 如果用字典里没有的键访问数据
- 修改字典
  - 向字典添加新内容的方法是增加新的键/值对
- 删除字典元素(不常用)
  - 能删单一的元素也能清空字典，清空只需一项操作。
  - 显示删除一个字典用del命令
- 键的特性
  - 不允许同一个键出现两次。
  - 键必须不可变，所以可以用数字，字符串或元组充当，所以用列表就不行
- 内置函数&方法

序号	函数	描述
1	cmp(dict1, dict2)	比较两个字典元素。
2	len(dict)	计算字典元素个数，即键的总数。
3	str(dict)	输出字典可打印的字符串表示。
4	type(variable)	返回输入的变量类型，如果变量是字典就返回字典类型。

序号	方法	描述
1	dict.clear()	删除字典内所有元素
2	dict.copy()	返回一个字典的浅复制
3	dict.fromkeys(seq[, val])	创建一个新字典，以序列 seq 中元素做字典的键，val 为字典所有键对应的初始值
4	dict.get(key, default=None)	返回指定键的值，如果值不在字典中返回default值
5	dict.has_key(key)	如果键在字典dict里返回true，否则返回false
6	dict.items()	以列表返回可遍历的(键, 值) 元组数组
7	dict.keys()	以列表返回一个字典所有的键
8	dict.setdefault(key, default=None)	和get()类似, 但如果键不存在于字典中，将会添加键并将值设为default
9	dict.update(dict2)	把字典dict2的键/值对更新到dict里
10	dict.values()	以列表返回字典中的所有值
11	pop(key[,default])	删除字典给定键 key 所对应的值，返回值为被删除的值。key值必须给出。 否则，返回default值。

序号	方法	描述
12	popitem()	随机返回并删除字典中的一对键和值。

In [50]:

```
dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}
del dict['Name']  # 删除键是'Name'的条目
dict.clear()      # 清空字典所有条目
del dict          # 删除字典
```

5. Python 函数

- 函数是组织好的，可重复使用的，用来实现单一，或相关联功能的代码段。
- 函数能提高应用的模块性，和代码的重复利用率。你已经知道Python提供了许多内建函数，比如print()。但你也可以自己创建函数，这被叫做用户自定义函数。

5.1 定义一个函数

- 你可以定义一个由自己想要功能的函数，以下是简单的规则：
  - 函数代码块以 def 关键词开头，后接函数标识符名称和圆括号()。
  - 任何传入参数和自变量必须放在圆括号中间。圆括号之间可以用于定义参数。
  - 函数的第一行语句可以选择性地使用文档字符串—用于存放函数说明。
  - 函数内容以冒号起始，并且缩进。
  - return [表达式] 结束函数，选择性地返回一个值给调用方。不带表达式的return相当于返回 None。

5.2 函数调用

- 定义一个函数只给了函数一个名称，指定了函数里包含的参数，和代码块结构。
- 这个函数的基本结构完成以后，你可以通过另一个函数调用执行，也可以直接从Python提示符执行。

5.3 参数传递

- 传递不可变对象和传可变对象
- strings, tuples, 和 numbers 是不可更改的对象，而 list,dict 等则是可以修改的对象。

5.4 return 语句

- return语句[表达式]退出函数，选择性地向调用方返回一个表达式
- 不带参数值的return语句返回None。

In [51]:

```
#以下为一个简单的Python函数，它将一个字符串作为传入参数，再打印到标准显示设备上。

def printme( str ):
   print(str)

# 调用函数
printme("我要调用用户自定义函数!");
printme("再次调用同一函数");
```

```
我要调用用户自定义函数!
再次调用同一函数
```

In [52]:

```
#传不可变对象实例
def ChangeInt( a ):
    a = 10
b = 2
ChangeInt(b)
print(b) # 结果是 2

#传可变对象实例
# 可写函数说明
def changeme( mylist ):
   mylist.append([1,2,3,4]);
   print("函数内取值: ", mylist)

# 调用changeme函数
mylist = [10,20,30];
changeme( mylist );
print("函数外取值: ", mylist)
```

```
2
函数内取值:  [10, 20, 30, [1, 2, 3, 4]]
函数外取值:  [10, 20, 30, [1, 2, 3, 4]]
```

In [53]:

```
# 可写函数说明
def sum( arg1, arg2 ):
   # 返回2个参数的和."
   total = arg1 + arg2
   print("函数内 : ", total)
   return total;

# 调用sum函数
total = sum( 10, 20 );
```

```
函数内 :  30
```

In [ ]: