

python基础

- python基础
 - 语言基础
 - 变量
 - 条件语句
 - try / except 语句
 - 函数
 - 循环和迭代
 - 字符串
 - 文件
 - 数据结构
 - 列表(list)
 - 字典(dict)
 - 元组(tuple)
 - 正则表达式

语言基础

变量

- 变量、表达式、赋值语句
- 可以使用**函数type() **询问所有对象的类型
- 数值类型、类型转换
- 输入：input()函数

条件语句

在if 语句或for 语句后需要缩进 • 保持缩进指明if/for语句块的范围 • 取消缩进回到与if 语句或for 语句对齐，表明if/for语句块结束 • 空白行不影响缩进 • 注释语句也不考虑缩进

try / except 语句

- 使用try和 except把有危险的代码包围起来 • 如果try模块的代码正常运行，except将被忽略 • 如果try模块的代码运行失败，会跳转到except模块执行

函数

1、Python有两类函数：

- 系统函数: 作为Python的一部分提供给用户的 - print(), input(), type(), float(), int() ...
- 用户自定义函数: 定义之后使用，把用户自定义函数名看作“新”的保留关键字（避免变量与自定义函数同名）

2、使用关键字def创建新的函数，在括号中输入参数。

- 实参是当调用函数时传递给函数的输入值，不同的实参可以让函数在不同的时候做不同的工作 • 实参列表置于函数名之后的括号中。
- 形参是在函数定义时使用的变量。又类似于“句柄”，在特定的函数调用中使得函数中的代码访问实参。
- 在函数定义中可以有多个形参，在调用函数时，只需添加多个实参即可。形参和实参的数量一样、顺序一致。

循环和迭代

- break 申明结束当前循环，跳到循环体后面的那条语句去执行。
- continue申明结束当前循环，跳转到循环顶端，开始下一次循环。
- python中有操作符is和is not，可用在逻辑表达式中，意思是与...相同，但又比==更强烈。

字符串

- 可以在方括号中指明索引来获得字符串里的任意字符，索引值是整型，从0开始。
- 系统函数len可以得到字符串的长度。
- 字符串分片：s[:] 如果第一个数或第二个数空白，分别表示从字符串第一个字符开始和到字符串最后一个字符。
- 操作符+ 用于字符串时表示“链接”。

- 在string库中，Python有一些字符串操作函数。如dir(string)；find() 函数找出第一个出现的子串，如果没有找到子串， find() 返回 -1；upper()大写字符串；lower()小写字符串；replace()函数有“查找和 替代”的功能；lstrip() 和 rstrip() 删除左边和右边的空白；strip() 删除字符串前后的空白。
ine.startswith('Please')返回布尔值，判断前缀。

文件

- 文本文件可以看作行序列。打开文件使用**open()**函数，open() 返回“文件句柄” - 即一个变量，用于在文件上执行操作。

```
handle = open(filename, mode)
```

```
fhand = open('mbox.txt', 'r')
```

- 返回句柄，用于操作文件 • 文件名是一个字符串,mode选项可选择，如果只需要读文件可取值为‘r’，如果需要写文件则取值为‘w’。

- 文件句柄打开后，可以视为字符串序列，文件的每一行是序列中的一个字符串，使用for语句来遍历这个序列 注意:序列是一个有序的集合,计算文件的行数。

```
xfile = open('mbox.txt') for cheese in xfile: print(cheese)
```

- 可以读取整个文件(包括换行符和所有字符)到一个字符串中。

```
fhand = open('mbox-short.txt') inp = fhand.read()
```

数据结构

数据结构，在计算机中组织数据的特定方法。

列表(list)

- 列表和for循环 - 绝配
- 字符串是不可变的-不能改变字符串 的内容-要改变字符串只能创建新串。 列表是可变的-可以使用索引操作符 改变列表的元素值。
- len()函数使用列表为参数，返回列表 中元素的个数。range函数返回包含一组数的列表，这组数从0开始， 到参数-1。使用+链接列表。使用:符号进行列表分片。可以创建一个空列表，然后使用append方法，添加元素。可以给列表排序，sort方法意思是“给自己 排序”。Split 方法把字符串分成几个组成部分，生产一个字符串列表。可以把他看作单 词，可以访问特定单词或使用loop遍历

所有单词。

- 线性集合的值有序存放。

字典(dict)

- 许多值，每一个值都有自己的标签
- 字典是Python最有效的数据集合,字典允许在Python中像数据库一样进行快速操作。
- 在字典中使用“查询标签”索引元素。字典使用keys来查找值，而列表使用索引来查找元素。字典常量使用大括号，有key : value组合对列表 • 可以使用空的大括号创建空字典。
- 字典最常见的用法就是计算某个事物出现的次数。
- 如果指出的key不在字典中，则会出错。可以使用in操作符查看某个key是否在字典中。
- 检查某个key是否在字典中，如果不在则赋予缺省值，这个功能非常常见，因此有专门的get() 方法完成这个功能。x = counts.get(name, 0)。

计算一行字符串中单词出现的频率通常的模式是先把字符串分割成单词，然后在单词列表中循环，使用字典记录每一个单词出现的次数。

- 可以从字典中获得 keys, values, 或者 items (两者) 列表。

元组(tuple)

- 元组是另一种序列，功能上类似于列表-他们都可以用索引访问元素，索引从0开始。不像列表，一旦创建了元组，就不能改变它的内容，-类似于字符串。
- 查看序列的方法，dir。
- 由于Python创建元组数据结构为不可改变的，那么在内存使用 和执行上就比列表效率高。因此在程序中使用“临时变量”时，更多选择使用元组。
- 字典的 items()方法 返回(key, value)元组 列表。
- 比较运算符可以用于元组间或其他序列间。比较时，Python把 对应位置的元素进行比较，直到找到不同的元素。可以充分利用元组列表排序的功能，获得有序的字典 • 使用items()方法和sorted()函数根据字典的key排序（因为元组可以进行比较）。

正则表达式

常见的正则表达式:

Item	meaning
^	Matches the beginning of a line
\$	Matches the end of the line.
.	Matches any character
\s	Matches whitespace
\S	Matches any non-whitespace character
*	Repeats a character zero or more times.
*?	Repeats a character zero or more times (non-greedy)
+	Repeats a character one or more times
+?	Repeats a character one or more times (non-greedy)
[aeiou]	Matches a single character in the listed set
[^XYZ]	Matches a single character not in the listed set.
[a-z0-9]	The set of characters can include a range
(Indicates where string extraction is to start.
)	Indicates where string extraction is to end

- 在程序中使用正则表达式之前必须先使用“import re”引入库
- 可以使用re.search() 来查看一个字符串是否匹配某个正则表达式，，类似于在字符串使用使用find() 函数。可以使用re.findall() 来提取出匹配正则表达式的字符串部分。****re.search()** 根据字符串是否匹配正则表达式，返回真/假结果 • 如果想把匹配的字符串摘取出来可以使用re.findall()。当使用re.findall()时，会返回零个或多个匹配正则表达式的子串。重复字符 (和 +) 在两个方向尽可能地匹配字符串，并不是所有正则表达式的重复代码都是贪婪的!如果你增加?号， 和 +则变为非贪婪。可以精炼摘取函数re.findall()，使用括号摘取匹配的部分。
- 点字符匹配任意字符，如果使用星字符，则表示该字符出现任意次。
- 如果需要正则表达式的保留字符是普通字符含义，可以在前面加 "。