目录

[第一章 2](#_Toc528793049)

[第二章 2](#_Toc528793050)

[第三章 4](#_Toc528793051)

[3.1、练习题 4](#_Toc528793052)

[3.2、实践题 5](#_Toc528793053)

[第四章 6](#_Toc528793054)

[4.1、练习题 6](#_Toc528793055)

[4.2、实践题 7](#_Toc528793056)

[第五章 8](#_Toc528793057)

[5.1、练习题 8](#_Toc528793058)

[5.2、实践题 8](#_Toc528793059)

[第六章 9](#_Toc528793060)

[6.1、练习题 9](#_Toc528793061)

[6.2、实践题 10](#_Toc528793062)

# 一、python基础

## 一、基础题

1．下面哪些是操作符，哪些是值？

\* 'hello' -88.8 - / + 5

答：操作符（+、 -、 \*、 /、 //、 %和\*\*是数学操作符， +和\*是字符串操作符）

三种数据类型（整型、 浮点型和字符串）带引号的是字符串，不带引号的是数字

2．下面哪个是变量，哪个是字符串？

spam

'spam'

答：第一个是变量，第二个是字符串

3．说出 3 种数据类型

答：整型int 、字符串 str 、浮点型 float

4．表达式由什么构成？所有表达式都做什么事？

答: 一般有运算符和操作对象组成。根据表达式的值做出决策

5．本章介绍了赋值语句，如 spam = 10。表达式和语句有什么区别？

答：表达是某件事，语句：表达式做某件事

6．下列语句运行后，变量 bacon 的值是什么？

bacon = 20

bacon + 1

答：21

7．下面两个表达式求值的结果是什么？

'spam' + 'spamspam'

'spam' \* 3

答：都是 spamspamspam

8．为什么 eggs 是有效的变量名，而 100 是无效的？

答：变量名不能是数字开头

9．哪 3 个函数能分别取得一个值的整型、浮点型或字符串版本？

答：整型 int() 、 浮点型 float() 、字符串 str()

10．为什么这个表达式会导致错误？如何修复？

'I have eaten ' + 99 + ' burritos.'

答：字符串和数字不能相加。修复方法：'I have eaten ' +’99’ + ' burritos.'

附加题：在线查找 len()函数的 Python 文档。它在一个标题为“Built-in Functions”

的网页上。扫一眼 Python 的其他函数的列表，查看 round()函数的功能，在交互式

环境中使用它。

答：len() 函数时计算长度。round()返回浮点数四舍五入。

## 二、控制流

1．布尔数据类型的两个值是什么？如何拼写？

答：bool值是 True 和 False

2． 3 个布尔操作符是什么？

答： and or not

3．写出每个布尔操作符的真值表（也就是操作数的每种可能组合，以及操作的结果）。

答：and 都是真的时候才是True，只要有一个是假 就是 False.

or 只要一个是真就是True,都是假的时候就是False.

ont 为真的时候就是False ,为假的时候就是True.

4．以下表达式求值的结果是什么？

(5 > 4) and (3 == 5) 答：False

not (5 > 4) 答：False

(5 > 4) or (3 == 5) 答：True

not ((5 > 4) or (3 == 5)) 答：False

(True and True) and (True == False) 答：False

(not False) or (not True) 答：True

5． 6 个比较操作符是什么？

答：== != > < <= >=

6．等于操作符和赋值操作符的区别是什么？

答：等于是== 2个等号，是比较操作符。赋值是= 一个等号

7．解释什么是条件，可以在哪里使用条件。

答：条件就是 表达式。可以在控制流语句中。

8．识别这段代码中的 3 个语句块：

spam = 0

if spam == 10:

print('eggs')

if spam > 5:

print('bacon')

else:

print('ham')

print('spam')

print(spam)

答：结果是打印0 ， 解释：1.排断spam是否等于10，如果等于就打印eggs，否则就跳过，2.并进入排断spam是否大于5，如果打印打印bacon，不大于打印ham。3.然后打印spam，4.打印spam的值等于0

9．编写代码，如果变量 spam 中存放 1，就打印 Hello，如果变量中存放 2，就打印 Howdy，如果变量中存放其他值，就打印 Greetings!

spam = int(input('>:'))

*if* spam == 1:  
 print('Hello')  
*elif* spam == 2:  
 print('Howdy')  
*else*:  
 print('Greetings!')

10． 如果程序陷在一个无限循环中， 你可以按什么键？

答：ctrl + C

11． break 和 continue 之间的区别是什么？

答：break 终止整个循环，continue 跳出当次循环

12． 在 for 循环中， range(10)、 range(0, 10)和 range(0, 10, 1)之间的区别是什么？

答：结果是一样的。默认是从0开始计数。range(0,10,1),后面的1是表示隔1个数字

13．编写一小段程序，利用 for 循环，打印出从 1 到 10 的数字。然后利用 while循环，编写一个等价的程序，打印出从 1 到 10 的数字。

答：

cou = 1

*while* cou < 11:  
 print(cou)  
 cou +=1

*for* i *in* range(1,11):  
 print(i)

14． 如果在名为 spam 的模块中， 有一个名为 bacon()的函数，那么在导入 spam模块后， 如何调用它？

答：form spam import bacon

spam.bacon()

附加题：在因特网上查找 round()和 abs()函数，弄清楚它们的作用。在交互式环境中尝试使用它们。

答：round() 是四舍五入 abs() 返回绝对值

用法参考：http://www.runoob.com/python/func-number-abs.html

## 三、函数

### 3.1、练习题

1． 为什么在程序中加入函数会有好处？

答：节省代码，减少重复工作

2．函数中的代码何时执行：在函数被定义时，还是在函数被调用时？

答：函数被调用时

3．什么语句创建一个函数？

答：def 函数名():

4．一个函数和一次函数调用有什么区别？

答：一个函数是整个函数代码。一次函数调用时执行一次整个函数代码。

5． Python 程序中有多少全局作用域？有多少局部作用域？

答：在调用一个函数时，创建一个全局函数和一个局部作用域。

6．当函数调用返回时，局部作用域中的变量发生了什么？

答：函数返回时，局部作用域被销毁，所有的变量被遗忘。

7．什么是返回值？返回值可以作为表达式的一部分吗？

答：返回值就是函数调用求值的结果，像所有值一样返回值可以作用表达式的一部分。

8．如果函数没有返回语句，对它调用的返回值是什么？

答：没有返回语句，他的返回值就是None

9．如何强制函数中的一个变量指的是全局变量？

答：global 强制函数中的一个变量是全局变量

10． None 的数据类型是什么？

答：None有自己的数据类型NoneType。你可以将None复制给任何变量，但是你不能创建其他NoneType对象。

11． import areallyourpetsnamederic 语句做了什么？

答：导入模块

12．如果在名为 spam 的模块中，有一个名为 bacon()的函数，在引入 spam 后，如何调用它？

答：form spam import bacon导入该模块

spam.bacon() 调用该模块

13．如何防止程序在遇到错误时崩溃？

答：加入try

14． try 子句中发生了什么？ except 子句中发生了什么？

答:把可能错误的代码放入try子句中，发生错误时执行except里的代码

### 3.2、实践题

1、编写一个名为 collatz()的函数，它有一个名为 number 的参数。如果参数是偶数，那么 collatz()就打印出 number // 2， 并返回该值。如果 number 是奇数， collatz()就打印并返回 3 \* number + 1

*def* collatz(*number*):

a = *number* % 2  
 *if* a == 0:  
 *return number* // 2  
 *elif* a == 1:  
 *return* 3 \* *number*+1  
print(collatz(19))

2、编写一个程序， 让用户输入一个整数， 并不断对这个数调用 collatz()， 直到函数返回值１（令人惊奇的是， 这个序列对于任何整数都有效， 利用这个序列，你迟早会得到 1！ 既使数学家也不能确定为什么。 你的程序在研究所谓的“Collatz序列”，它有时候被称为“最简单的、 不可能的数学问题”）。第 3 章 函数记得将 input()的返回值用 int()函数转成一个整数，否则它会是一个字符串。

*def* collatz(*number*):

a = *number* % 2  
 *if number* == 1:  
 *return number  
 elif* a == 0:  
 number = *number* // 2  
 print(*number*)  
 collatz(*number*)  
 *elif* a == 1:  
 number = 3 \* *number*+1  
 print(*number*)  
 collatz(*number*)  
*while True*:  
 *try*:  
 num = int(input("请输入一个整数->:"))  
 collatz(num)  
 *break  
 except* ValueError:  
 print("please input a integer number")

3、在前面的项目中添加 try 和 except 语句，检测用户是否输入了一个非整数的字符串。正常情况下， int()函数在传入一个非整数字符串时，会产生 ValueError 错误，比如 int('puppy')。在 except 子句中，向用户输出一条信息，告诉他们必须输入一个整数。

检测是否整数方法1：

*try*:

num = int(input("请输入一个整数->:"))  
*except* ValueError:  
 print("please input a integer number")

方法2：

num = input("请输入一个整数->:")

*if* num.isdigit():  
 num = int(num)  
*else*:  
 print("please input a integer number")

## 四、列表

### 4.1、练习题

1．什么是[]？

答：列表，它包含多个字构成的序列

2．如何将'hello'赋给列表的第三个值，而列表保存在名为 spam 的变量中？（假定变量包含[2, 4, 6, 8, 10]）。

答：spam[2] = 'hello'

对接下来的 3 个问题，假定 spam 包含列表['a', 'b', 'c', 'd']。

3． spam[int('3' \* 2) / 11]求值为多少？ 答：d

4． spam[-1]求值为多少？ 答：d

5． spam[:2]求值为多少？ 答：a、b

对接下来的 3 个问题。假定 bacon 包含列表[3.14, 'cat', 11, 'cat', True]。

6． bacon.index('cat')求值为多少？ 答：1

7． bacon.append(99)让 bacon 中的列表值变成什么样？ 答：后面多个99

8． bacon.remove('cat')让 bacon 中的列表时变成什么样？ 答：删除了下标1的cat

9．列表连接和复制的操作符是什么？

答：和字符串一样 连接是+ 号 复制是 \* 号

10． append()和 insert()列表方法之间的区别是什么？

答：append() 是添加到末尾，insert()是插入任意位置

11．从列表中删除值有哪两种方法？

答：知道删除的值用remove()，知道删除的下标用del和pop()

用法： remove(‘cat’) del a[1] a.pop(1)

12．请说出列表值和字符串的几点相似之处。

答： 都可以用len()，都有下标和切片，都可以用for循环，都可以连接和复制

13．列表和元组之间的区别是什么？

答：列表创建是[]，是可变的，元组创建是()，是只读的。

14．如果元组中只有一个整数值 42，如何输入该元组？

答：元组只有一个值末尾必须有逗号，可以用type()查看类型

15．如何从列表值得到元组形式？如何从元组值得到列表形式？

答：tuple()可以吧列表变成元组和list()可以吧元组变成列表

16．“包含” 列表的变量，实际上并未真地直接包含列表。它们包含的是什么？

答：对列表值的引用

17． copy.copy()和 copy.deepcopy()之间的区别是什么？

答：浅拷贝和深拷贝。只有copy.deepcopy()会复制列表内的所有列表

### 4.2、实践题

1、逗号代码

假定有这样的列表：spam = ['apples', 'bananas', 'tofu', 'cats']

编写一个函数，它以一个列表值作为参数，返回一个字符串。该字符串包含所有表项，表项之间以逗号和空格分隔，并在最后一个表项之前插入 and。例如，将前面的 spam 列表传递给函数，将返回'apples, bananas, tofu, and cats'。但你的函数应该能够处理传递给它的任何列表。

spam = ['apples', 'bananans', 'tofu', 'cats']

*def* list\_str(*lst*):  
 *return* ','.join(*lst*[:-1]+['and,'+*lst*[-1]])  
print(list\_str(spam))

特别说明：join()方法将序列中的元素以指定的字符连接生成一个新的字符串

参考：http://www.runoob.com/python/att-string-join.html

2、字符图网格

假定有一个列表的列表， 内层列表的每个值都是包含一个字符的字符串， 像这样：

grid = [['.', '.', '.', '.', '.', '.'],

['.', 'O', 'O', '.', '.', '.'],

['O', 'O', 'O', 'O', '.', '.'],

['O', 'O', 'O', 'O', 'O', '.'],

['.', 'O', 'O', 'O', 'O', 'O'],

['O', 'O', 'O', 'O', 'O', '.'],

['O', 'O', 'O', 'O', '.', '.'],

['.', 'O', 'O', '.', '.', '.'],

['.', '.', '.', '.', '.', '.']]

你可以认为 grid[x][y]是一幅“图” 在 x、 y 坐标处的字符， 该图由文本字符组成。 原点(0, 0)在左上角， 向右 x 坐标增加， 向下 y 坐标增加。复制前面的网格值， 编写代码用它打印出图像。

..OO.OO..

.OOOOOOO.

.OOOOOOO.

..OOOOO..

...OOO...

....O....

*for* i *in* range(len(grid[0])):

*for* j *in* range(len(grid)):  
 print(grid[j][i],end='')  
 print()

特别说明：[end](https://www.baidu.com/s?wd=end&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)=' '意思是末尾不换行

## 五、字典

### 5.1、练习题

1．空字典的代码是怎样的？

答：{} 大括号是字典,[]是列表 ()是元组，集合是set()

2．一个字典包含键'fow'和值 42， 看起来是怎样的？

答：a = {‘fow’:42}

3．字典和列表的主要区别是什么？

答：字典是{}，包含键和值,是无序的，列表是[]，只包含值是有序的。

4．如果 spam 是{'bar': 100}， 你试图访问 spam['foo']， 会发生什么？

答：报错，提示foo键不存在

5．如果一个字典保存在 spam 中， 表达式'cat' in spam 和'cat' in spam.keys()之间的区别是什么？

答：结果是一样的，都是查询 cat 是否 是spam里的key

6．如果一个字典保存在变量中， 表达式'cat' in spam 和'cat' in spam.values()之间的区别是什么？

答：一个是查键是否存在，一个是查值是否存在

7．下面代码的简洁写法是什么？

if 'color' not in spam:

spam['color'] = 'black'

答：spam.setdefault(‘color’,’black’)

8．什么模块和函数可以用于“漂亮打印”字典值？

答：模块需要导入,导入方法 import pprint，打印的时候 需要pprint.pprint()

### 5.2、实践题

**5.6.1、好玩游戏的物品清单**

你在创建一个好玩的视频游戏。用于对玩家物品清单建模的数据结构是一个字典。其中键是字符串，描述清单中的物品，值是一个整型值，说明玩家有多少该物品。例如，字典值{'rope': 1, 'torch': 6, 'gold coin': 42, 'dagger': 1, 'arrow': 12}意味着玩家有 1 条绳索、 6 个火把、 42 枚金币等。

写一个名为 displayInventory()的函数，它接受任何可能的物品清单， 并显示如下：

Inventory:

12 arrow

42 gold coin

1 rope

6 torch

1 dagger

Total number of items: 62

*def* disp(*spa*):

coun = 0  
 *for* k,v *in spa*.items():  
 print(v,k)  
 coun +=v  
 print('Total number of items:' + str(coun))

特别说明：函数items() 返回可遍历的(键, 值) 元组数组。

参考：http://www.runoob.com/python/att-dictionary-items.html

**5.6.2 列表到字典的函数，针对好玩游戏物品清单**

假设征服一条龙的战利品表示为这样的字符串列表：

dragonLoot = ['gold coin', 'dagger', 'gold coin', 'gold coin', 'ruby']

写一个名为 addToInventory(inventory, addedItems)的函数， 其中 inventory 参数是一个字典， 表示玩家的物品清单（像前面项目一样）， addedItems 参数是一个列表，就像 dragonLoot。

addToInventory()函数应该返回一个字典， 表示更新过的物品清单。请注意， 列表可以包含多个同样的项。

dragonLoot = ['gold coin', 'dagger', 'gold coin', 'gold coin', 'ruby']

stuff = {'rope': 1, 'torch': 6, 'gold coin': 42, 'dagger': 1, 'arrow': 12}  
*def* add\_dis(*inv*,*add*):  
 a = {}  
 coun = 0  
 print('inventory')  
 *for* i *in add*:  
 a.setdefault(i,0)  
 a[i] += 1  
 *for* j *in* a.keys():  
 *if* j *in inv*:  
 a[j] += *inv*[j]  
 *for* k *in inv*.keys():  
 *if* k *not in* a.keys():  
 a.setdefault(k,0)  
 a[k] += *inv*[k]  
 *for* b,c *in* a.items():  
 print(str(c) + ' : ' +str(b))  
 coun += c  
 print('Total number of items:' + str(coun))  
add\_dis(stuff,dragonLoot)

## 六、字符串操作

### 6.1、练习题

1．什么是转义字符？

答：让你输入一些字符，它们用其他方式是不可能放在字符串里的

2．转义字符\n 和\t 代表什么？

答：\n是换行，\t是tab键

3．如何在字符串中放入一个倒斜杠字符\？

答： \\ 两个倒斜杠

4．字符串"Howl's Moving Castle"是有效字符串。为什么单词中的单引号没有转义， 却没有问题？

答：因为引号不一样。双引号里可以放一个单引号，单引号里可以放一个双引号

5．如果你不希望在字符串中加入\n， 怎样写一个带有换行的字符串？

答：三引号可以换行不用\n

6．下面的表达式求值为什么？

• 'Hello world!'[1] 答： e

• 'Hello world!'[0:5] 答：Hello

• 'Hello world!'[:5] 答：Hello

• 'Hello world!'[3:] 答：lo world!

7．下面的表达式求值为什么？

• 'Hello'.upper() 答：全部变成大写 HELLO, upper()是变成大写

• 'Hello'.upper().isupper() 答：返回True，检测是不是全部大写,islower()检测是不是全部小写

• 'Hello'.upper().lower() 答：全部变成小写 hello ，lower()是变成小写

8．下面的表达式求值为什么？

• 'Remember, remember, the fifth of November.'.split()

答：把字符串变成列表,以空格分割['Remember,', 'remember,', 'the', 'fifth', 'of', 'November.']

• '-'.join('There can be only one.'.split())

答：把空格换成- 返回There-can-be-only-one.

9．什么字符串方法能用于字符串右对齐、 左对齐和居中？

答：右对齐：rjust() 左对齐：ljust() 居中：center()

10．如何去掉字符串开始或末尾的空白字符？

答：删除左边空白lstrip()，右边rstrip()，中间strip()

### 6.2、实践题

**1、表格打印**

编写一个名为 printTable()的函数， 它接受字符串的列表的列表，将它显示在组织良好的表格中， 每列右对齐。假定所有内层列表都包含同样数目的字符串。例如，

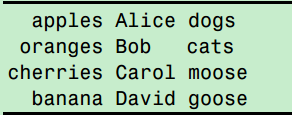
该值可能看起来像这样：

tableData = [['apples', 'oranges', 'cherries', 'banana'],

['Alice', 'Bob', 'Carol', 'David'],

['dogs', 'cats', 'moose', 'goose']]

你的 printTable()函数将打印出：



a = [0,0,0]

*for* i,j *in* enumerate(tableData):  
 *for* k *in* j:  
 *if* len(k) > a[i]:  
 a[i] = len(k)  
*for* l *in* range(len(tableData[0])):  
 print(tableData[0][l].rjust(a[0]),tableData[1][l].ljust(a[1]),tableData[2][l].rjust(a[2]))

# 二、自动化任务

## 一、正则表达式

### 1、练习题

1．创建 Regex 对象的函数是什么？

答：re.compile() 所有的正则表达式的函数都在re模块中，导入 import re

2．在创建 Regex 对象时，为什么常用原始字符串？

答：使用原始字符串是为了反斜杠不必转义

3．search()方法返回什么？

答：没找到返回 None， 找到返回Match对象

4．通过 Match 对象，如何得到匹配该模式的实际字符串？

答：group()

5．用 r'(\d\d\d)-(\d\d\d-\d\d\d\d)'创建的正则表达式中，分组 0 表示什么？分组 1呢？分组 2 呢？

答：0或者不传入参数返回整个匹配的文件,分组1表示第一对括号，分组2表示第二对括号

6．括号和句点在正则表达式语法中有特殊的含义。如何指定正则表达式匹配真正的括号和句点字符？

答：句号和括号可以反斜杠转义：\. \( \)

7．findall()方法返回一个字符串的列表，或字符串元组的列表。是什么决定它提供哪种返回？

答：调用的正则表达式,没有分组的正则表达式返回列表,有分组的正则表达式返回元组

8．在正则表达式中， |字符表示什么意思？

答：| 表示匹配两组中的任何一个。

9．在正则表达式中， ?字符有哪两种含义？

答：匹配0次或一次前面的分组 或用于表示非贪心匹配

10．在正则表达式中， +和\*字符之间的区别是什么？

答：+匹配一次或多次，\*匹配0次或多次

11．在正则表达式中， {3}和{3,5}之间的区别是什么？

答：{3}匹配前面分组的精确3次实例，{3,5}匹配3至5次实例

12．在正则表达式中， \d、 \w 和\s 缩写字符类是什么意思？

答：\d 是数字 \w是单词 \s是空格(空白字符)

13．在正则表达式中， \D、 \W 和\S 缩写字符类是什么意思？

答：\D是数字外的字符 \W是单词外的字符 \S是空格外的字符。可参考：(120页)

14．如何让正则表达式不区分大小写？

答：re.compile传入第二个参数 re.I 或 re.IGNORECASE

15．字符 . 通常匹配什么？如果 re.DOTALL 作为第二个参数传递给 re.compile()，它会匹配什么？

答：. 匹配任意字符 换行除外。如果将re.DOTALL作为第二个参数传入re.compile()那么点也好匹配换行符

16．.\*和\*?之间的区别是什么？

答：.\* 贪心匹配， .？非贪心匹配

17．匹配所有数字和小写字母的字符分类语法是什么？

答: [0-9a-z] [a-z0-9]

18．如果 numRegex = re.compile(r'\d+')，那么 numRegex.sub('X', '12 drummers, 11 pipers, five rings, 3 hens’)返回什么？

答：表示查到的数字 替换成X 返回结果：X drummers, X pipers, five rings, X hens

19．将 re.VERBOSE 作为第二个参数传递给 re.compile()，让你能做什么？

答：re.VERBOSE 参数允许为传入 re.compile() 的字符串添加空格和注释。

20．如何写一个正则表达式， 匹配每 3 位就有一个逗号的数字？它必须匹配以下数字：

 '42'

 '1,234'

 '6,368,745'

但不会匹配：

 '12,34,567' （逗号之间只有两位数字）

 '1234' （缺少逗号）

答：re.compile（r'^\d{1,3}(,{3})\*$'）将创建这个正则表达式，但其他正则表达式字符串可以生成类似的正则表达式

21．如何写一个正则表达式， 匹配姓 Nakamoto 的完整姓名？ 你可以假定名字总是出现在姓前面， 是一个大写字母开头的单词。该正则表达式必须匹配：

 'Satoshi Nakamoto'

 'Alice Nakamoto'

 'RoboCop Nakamoto'

但不匹配：

 'satoshi Nakamoto'（名字没有大写首字母）

 'Mr. Nakamoto'（前面的单词包含非字母字符）

 'Nakamoto' （没有名字）

 'Satoshi nakamoto'（姓没有首字母大写）

答：re.compile(r'[A-Z][a-z]\*\sNakamoto')

22．如何编写一个正则表达式匹配一个句子，它的第一个词是 Alice、 Bob 或Carol， 第二个词是 eats、 pets 或 throws， 第三个词是 apples、 cats 或 baseballs。该句子以句点结束。 这个正则表达式应该不区分大小写。它必须匹配：

 'Alice eats apples.'

 'Bob pets cats.'

 'Carol throws baseballs.'

 'Alice throws Apples.'

 'BOB EATS CATS.'

但不匹配：

 'RoboCop eats apples.'

 'ALICE THROWS FOOTBALLS.'

 'Carol eats 7 cats.'

答：re.compile(r'(Alice|Bob|Carol)\s(eats|pets|throws)\ s(apples|cats|baseballs)\.',re.IGNORECASE)

### 2、实践题

**7.1、强口令检测**

写一个函数，它使用正则表达式， 确保传入的口令字符串是强口令。 强口令的定义是： 长度不少于 8 个字符， 同时包含大写和小写字符， 至少有一位数字。你可能需要用多个正则表达式来测试该字符串， 以保证它的强度。

*import* re,pyperclip

password = pyperclip.paste()  
*def* Passwd(*password*):  
 *if* len(*password*) >=8:  
 num = re.compile(r'\d+')  
 *if* num.search(*password*) == *None*:  
 *return* "至少有一位数字"  
 number2 = re.compile(r'[A-Z]+')  
 *if* number2.search(*password*) == *None*:  
 *return* "至少包含一位大写字母"  
 number3 = re.compile(r'[a-z]+')  
 *if* number3.search(*password*) == *None*:  
 *return* "至少包含一位小写字母"  
 *return* "密码符合要求"  
 *else*:  
 *return* "密码不满足要求"

**7.2、strip()的正则表达式版本**

写一个函数，它接受一个字符串， 做的事情和 strip()字符串方法一样。如果只传入了要去除的字符串， 没有其他参数， 那么就从该字符串首尾去除空白字符。否则， 函数第二个参数指定的字符将从该字符串中去除。

*import* re

*def* re\_strip(*s*, *t*='\\s'):  
 tb\_format = '^' + *t* + '+'  
 tt\_format = *t* + '+' + '$'  
 sb\_re = re.compile(tb\_format)  
 st\_re = re.compile(tt\_format)  
 s = sb\_re.sub('', *s*)  
 s = st\_re.sub('', *s*)  
 *return s*

## 二、读写文件

### 1、练习题

1．相对路径是相对于什么？

答：相当于当前目录

2．绝对路径从什么开始？

答：从根目录开始的路径

3． os.getcwd()和 os.chdir()函数做什么事？

答：os.gercwd()返回当前工作目录，os.chdir()改变当前工作目录

4． .和..文件夹是什么？

答 .表示当前文件夹 ..表示上级文件夹

5． 在 C:\bacon\eggs\spam.txt 中， 哪一部分是目录名称， 哪一部分是基本名称？

答：C:\bacon\eggs\是目录 spam.txt是文件名称

6．可以传递给 open()函数的 3 种“模式”参数是什么？

答：只读模式 r 只写模式 w 追加模式 a

7．如果已有的文件以写模式打开，会发生什么？

答：原来的文件内容会被删除

8． read()和 readlines()方法之间的区别是什么？

答：read()读所有内容，readlines 读一行内容

9． shelf 值与什么数据结构相似？

答：shelf 值类似字典值，它有键和值，以及 keys() 和 values() 方法，类似于同名的字典方法。

### 2、实践题

**8.1、扩展多重剪贴板**

扩展本章中的多重剪贴板程序，增加一个 delete <keyword>命令行参数，它将从 shelf 中删除一个关键字。然后添加一个 delete 命令行参数，它将删除所有关键字。

**8.2、疯狂填词**

创建一个疯狂填词（Mad Libs）程序，它将读入文本文件， 并让用户在该文本文件中出现 ADJECTIVE、 NOUN、 ADVERB 或 VERB 等单词的地方， 加上他们自己的文本。例如，一个文本文件可能看起来像这样：

The ADJECTIVE panda walked to the NOUN and then VERB. A nearby NOUN was unaffected by these events.

程序将找到这些出现的单词， 并提示用户取代它们。

Enter an adjective:

silly

Enter a noun:

chandelier

Enter a verb:

screamed

Enter a noun:

pickup truck

以下的文本文件将被创建：

The silly panda walked to the chandelier and then screamed. A nearby pickup truck was unaffected by these events.

结果应该打印到屏幕上， 并保存为一个新的文本文件。

**8.3、正则表达式查找**

编写一个程序， 打开文件夹中所有的.txt 文件，查找匹配用户提供的正则表达式的所有行。结果应该打印到屏幕上。

## 三、组织模块

### 1、重要模块及方法

### os模块：作系统的功能接口的模块

os.path.abspath( path ) ----------------- 返回参数的绝对路径，可以将相对路径转换为绝对路径

os.path.isabs( path ) -------------如果参数是绝对路径就返回True,如果是相对路径就返回False

os.path.relpath( path start )-------将返回从start路径到path的相对路径的字符串。如果没有提供start，就使用当前工作目录作为开始路径。

os.path.dirname(path) --------- 将返回一个字符串，它包含path参数中最后一个斜杠之前的所有内容。

os.path.basename(path) -------- 将返回一个字符串，它包含path参数中最后一个斜杠之后的所有内容。

os.path.split(path) ------------------- 返回2个字符串元组，路径的目录名称和基本名称，

os.getcwd()-----------------------------取得当前工作路径

os.chdir(c:\\windows) -----------------------------改变当前工作路径，变成c:\windows

os.path.join('fan','fan')---------------------------------拼接文件路径,返回 fan\fan

os.makedirs('E:\\System\\PyCharm\\fan')---------------------------创建文件夹fan

os.path.getsize(path)----------------------------将返回 path 参数中文件的大小

os.listdir(path)-------------------------------------将返回文件名字符串的列表

os.path.exists(path)------------path 参数所指的文件或文件夹存在将返回 True，否则返回 False。

os.path.isfile(path)---------------如果 path 参数存在并且是一个文件,将返回 True， 否则返回 False。

os.path.isdir(path)--------------如果 path 参数存在并且是一个文件夹，将返回 True，否则返回 False。

os.unlink(path)-----------------------------将删除 path 处的文件

os.rmdir(path)-------------------------将删除 path 处的文件夹。该文件夹必须为空，其中没有任何文件和文件夹。

os.walk()----------------遍历目录树

*import* os

*for* folderName, subfolders, filenames *in* os.walk('E:\\System\\PyCharm\\自动化练习\\其他\\时尚'):  
 print(folderName)  
 *for* sub *in* subfolders:  
 print(folderName + '\\'+ sub)  
 *for* file *in* filenames:  
 print(folderName + '\\' + file)  
 print('')

### shutil模块: Python 程序中复制\移动\改名和删除文件

shutil.copy(source, destination)------------------------将路径 source 处的文件复制到路径 destination处的文件夹

shutil.move('C:\\bacon.txt', 'C:\\eggs') ------------------------将 C:\bacon.txt 移动到文件夹 C:\eggs 中

shutil.rmtree(path)------------------将删除 path 处的文件夹，它包含的所有文件和文件夹都会被删除。

### zipfile模块:压缩文件

zip = zipfile.ZipFile('gx.zip')

print(zip.namelist())-------------------读取ZIP压缩包文件

info = zip.getinfo(’spam.txt’) ----------------------file\_size表示原来文件大小和compress\_size表示压缩后文件大小

print(info. file\_size) print(info. compress\_size)

zip.extractall()------------------------------解压缩

newZip = zipfile.ZipFile('new.zip', 'w')

newZip.write('22', compress\_type=zipfile.ZIP\_DEFLATED)------------------新加压缩包，添加的话吧w换成a，