**Python学习笔记**

**第一部分：基础编程（9.28-10.7）**

9.28

1、转义字符  
转义字符 描述  
\(在行尾时) 续行符  
\\ 反斜杠符号  
\’ 单引号  
\” 双引号  
\a 响铃  
\b 退格(Backspace)  
\e 转义  
\000 空  
\n 换行  
\v 纵向制表符  
\t 横向制表符  
\r 回车  
\f 换页  
\oyy 八进制数yy代表的字符，例如：\o12代表换行  
\xyy 十进制数yy代表的字符，例如：\x0a代表换行  
\other 其它的字符以普通格式输出  
2、常用函数  
str.replace(old, new[, max])替换函数  
len(x)字符串长度计算  
str.split(' ')字符串分割  
3、格式化输出:  
%s,表示格式化一个对象为字符1. 打印字符串;  
%d,打印整数;  
%f,打印浮点数;%.nf打印浮点数（指定保留小数点位数）  
%10s/%8d/%8.2f:指定占位符宽度,%-10s指定占位符宽度（左对齐）  
%08d指定占位符（只能用0当占位符？）  
科学计数法:format(0.0015,'.2e')  
4、编码解码  
encode(以指定的编码格式编码字符串，默认编码为 'utf-8'；  
decode解码  
# -\*- coding: utf-8 -\*-:  
为了告诉Python解释器，按照UTF-8编码读取源代码，否则，你在源代码中写的中文输出可能会有乱码  
5、几种常用的Python内置数据结构  
list: 有序集合。可以添加、删除、插入数据(可以改变数据资料）  
('.append:方法用于在列表末尾添加新的对象')  
(insert() :函数用于将指定对象插入列表的指定位置)  
(pop函数用于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)  
tuple:tuple:和list类似，不同的是，tuple的元素不可改变！更安全  
dict:很多书上又称为健值对（key:value） pairs,具有极快的查询速度  
get() 函数返回指定键的值，如果值不在字典中返回默认值。  
keys() 函数以列表返回一个字典所有的键  
values() 函数以列表返回字典中的所有值。  
set:可以看成数学意义上的无序和无重复元素的集合，因此，两个set可以做数学意义上的交集、并集等操作：  
add:add() 方法用于给集合添加元素，如果添加的元素在集合中已存在，则不执行任何操作  
remove:remove() 函数用于移除列表中某个值的第一个匹配项。  
请务必注意，dict内部存放的顺序和key放入的顺序是没有关系的。  
##和list比较，dict有以下几个特点：  
查找和插入的速度极快，不会随着key的增加而变慢；  
需要占用大量的内存，内存浪费多。  
而list相反：  
查找和插入的时间随着元素的增加而增加；  
占用空间小，浪费内存很少。  
所以，dict是用空间来换取时间的一种方法。  
dict可以用在需要高速查找的很多地方，在Python代码中几乎无处不在，正确使用dict非常重要，需要牢记的第一条就是dict的key必须是不可变对象  
6、可变对象和不可变对象  
对于可变对象，比如list，对list进行操作，list内部的内容是会变化的，  
sort() 函数用于对原列表进行排序，如果指定参数，则使用比较函数指定的比较函数

7、条件循环-控制流（**特别注意冒号**）

**条件：**

if <条件判断1>:

<执行1>

elif <条件判断2>:

<执行2>

elif <条件判断3>:

<执行3>

else:

<执行4>

注意：if语句执行有个特点，它是从上往下判断，如果在某个判断上是True，把该判断对应的语句执行后，就忽略掉剩下的elif和else，所以，请测试并解释为什么下面的程序打印的是teenager：

**循环**：

For：for iterating\_var in sequence:

statements(s)

While：while 判断条件：

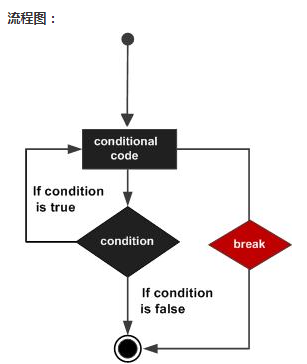
执行语句……

Break：

break语句用来终止循环语句，即循环条件没有False条件或者序列还没被完全递归完，也会停止执行循环语句。

break语句用在while和for循环中。

如果您使用嵌套循环，break语句将停止执行最深层的循环，并开始执行下一行代码。



Continue: 语句跳出本次循环，而break跳出整个循环。

continue 语句用来告诉Python跳过当前循环的剩余语句，然后继续进行下一轮循环。

continue语句用在while和for循环中。

8、函数

python 内置了很多函数，正确调用函数可以高效地完成任务

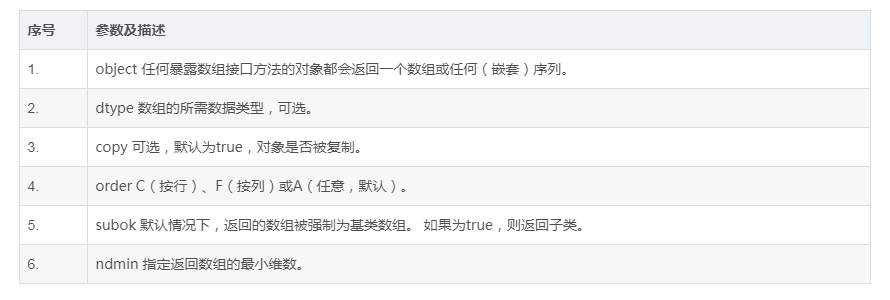
函数名是可以改变的，和变量名意义类似，函数名是用来存放函数的，所以它指向一个函数

一、arange()类似于内置函数range()，通过指定开始值、终值和步长创建表示等差数列的一维数组，注意得到的结果数组不包含终值。

二、linspace()通过指定开始值、终值和元素个数创建表示等差数列的一维数组，可以通过endpoint参数指定是否包含终值，默认值为True，即包含终值

三、numpy：它是一个由多维数组对象和用于处理数组的例程集合组成的库。

* 数组的算数和逻辑运算。
* 傅立叶变换和用于图形操作的例程。
* 与线性代数有关的操作。 NumPy 拥有线性代数和随机数生成的内置函数



四、matplotlib：生成 绘图、直方图、功率谱、柱状图、误差图、散点图等

matplotlib.pyplot：提供一个类似matlab的绘图框架。

9、文件操作

 open() 函数：用于打开一个文件，创建一个 **file** 对象，相关的方法才可以调用它进行读写

open(name[, mode[, buffering]])

参数说明：

* name : 一个包含了你要访问的文件名称的字符串值。
* mode : mode 决定了打开文件的模式：只读，写入，追加等。所有可取值见如下的完全列表。这个参数是非强制的，默认文件访问模式为只读(r)。
* buffering : 如果 buffering 的值被设为 0，就不会有寄存。如果 buffering 的值取 1，访问文件时会寄存行。如果将 buffering 的值设为大于 1 的整数，表明了这就是的寄存区的缓冲大小。如果取负值，寄存区的缓冲大小则为系统默认。

**file 对象方法**

* **file.read([size])**：size 未指定则返回整个文件，如果文件大小 >2 倍内存则有问题，f.read()读到文件尾时返回""(空字串)。
* **file.readline()**：返回一行。
* **file.readlines([size])**：返回包含size行的列表, size 未指定则返回全部行。
* **for line in f: print line**：通过迭代器访问。
* **f.write("hello\n")**：如果要写入字符串以外的数据,先将他转换为字符串。
* **f.tell()**：返回一个整数,表示当前文件指针的位置(就是到文件头的比特数)。
* **f.seek(偏移量,[起始位置])**：用来移动文件指针。
  + 偏移量: 单位为比特，可正可负
  + 起始位置: 0 - 文件头, 默认值; 1 - 当前位置; 2 - 文件尾
* **f.close()** 关闭文件
* 不同模式打开文件的完全列表：

|  |  |
| --- | --- |
| **模式** | **描述** |
| r | 以只读方式打开文件。文件的指针将会放在文件的开头。这是默认模式。 |
| rb | 以二进制格式打开一个文件用于只读。文件指针将会放在文件的开头。这是默认模式。 |
| r+ | 打开一个文件用于读写。文件指针将会放在文件的开头。 |
| rb+ | 以二进制格式打开一个文件用于读写。文件指针将会放在文件的开头。 |
| w | 打开一个文件只用于写入。如果该文件已存在则打开文件，并从开头开始编辑，即原有内容会被删除。如果该文件不存在，创建新文件。 |
| wb | 以二进制格式打开一个文件只用于写入。如果该文件已存在则打开文件，并从开头开始编辑，即原有内容会被删除。如果该文件不存在，创建新文件。 |
| w+ | 打开一个文件用于读写。如果该文件已存在则打开文件，并从开头开始编辑，即原有内容会被删除。如果该文件不存在，创建新文件。 |
| wb+ | 以二进制格式打开一个文件用于读写。如果该文件已存在则打开文件，并从开头开始编辑，即原有内容会被删除。如果该文件不存在，创建新文件。 |
| a | 打开一个文件用于追加。如果该文件已存在，文件指针将会放在文件的结尾。也就是说，新的内容将会被写入到已有内容之后。如果该文件不存在，创建新文件进行写入。 |
| ab | 以二进制格式打开一个文件用于追加。如果该文件已存在，文件指针将会放在文件的结尾。也就是说，新的内容将会被写入到已有内容之后。如果该文件不存在，创建新文件进行写入。 |
| a+ | 打开一个文件用于读写。如果该文件已存在，文件指针将会放在文件的结尾。文件打开时会是追加模式。如果该文件不存在，创建新文件用于读写。 |
| ab+ | 以二进制格式打开一个文件用于追加。如果该文件已存在，文件指针将会放在文件的结尾。如果该文件不存在，创建新文件用于读写。 |

**10.4**

1. itertools提供了非常有用的用于操作迭代对象的函数。

2. ZeroDivisionError异常处理。