



## Python数据分析——第2周

**【声明】** 本视频和幻灯片为炼数成金网络课程的教学资料，所有资料只能在课程内使用，不得在课程以外范围散播，违者将可能被追究法律和经济责任。

课程详情访问炼数成金培训网站

<http://edu.dataguru.cn>

- 提供全面的数据价值资讯，涵盖商业智能与数据分析、大数据、企业信息化、数字化技术等，各种高性价比课程信息，赶紧掏出您的手机关注吧！



## ◆ 条件语句

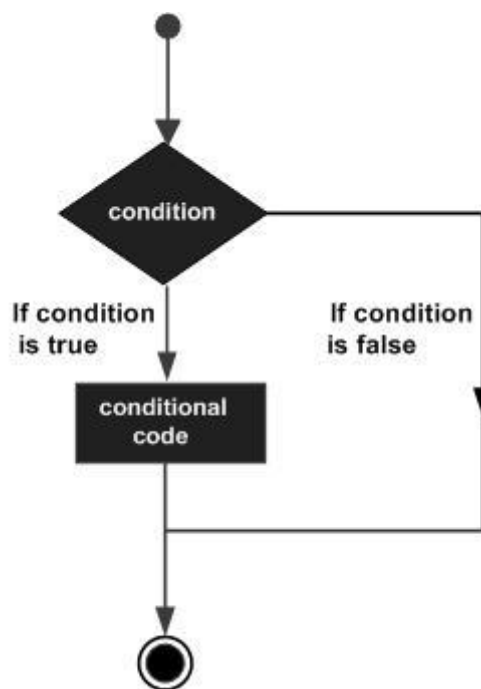
## ◆ 循环语句

- While语句
- For语句
- Break语句
- Continue语句

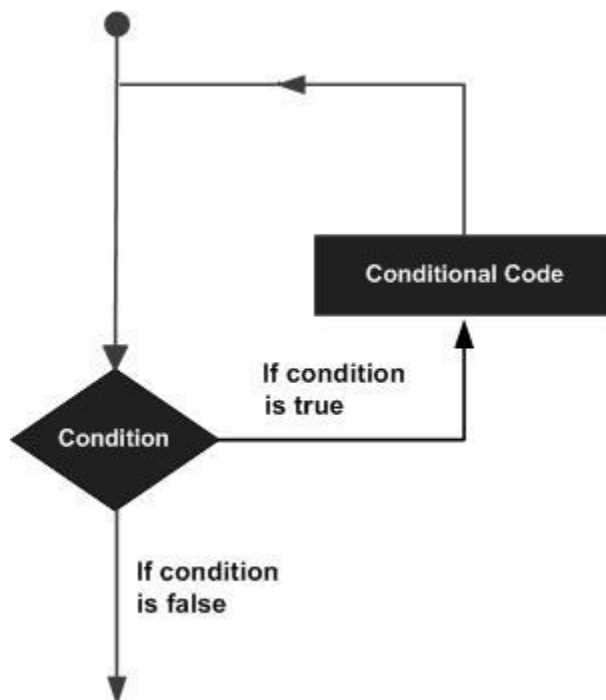
## ◆ 常用函数

## ◆ 自定义函数

- ◆ Python条件语句是通过一条或多条语句的执行结果（True或者False）来决定执行的代码块



- ◆ 循环语句允许我们执行一个语句或语句组多次

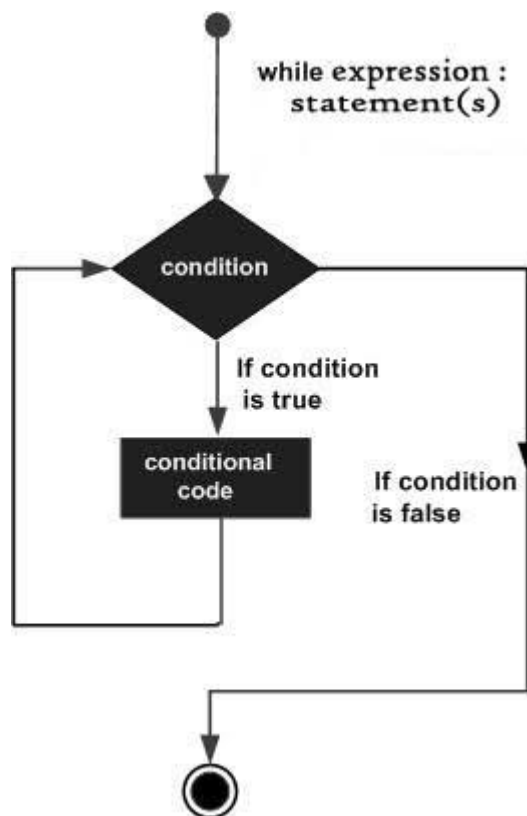


循环类型	描述
<a href="#">while 循环</a>	在给定的判断条件为 true 时执行循环体，否则退出循环体。
<a href="#">for 循环</a>	重复执行语句
<a href="#">嵌套循环</a>	你可以在while循环体中嵌套for循环

控制语句	描述
<a href="#">break 语句</a>	在语句块执行过程中终止循环，并且跳出整个循环
<a href="#">continue 语句</a>	在语句块执行过程中终止当前循环，跳出该次循环，执行下一次循环。
<a href="#">pass 语句</a>	pass是空语句，是为了保持程序结构的完整性。

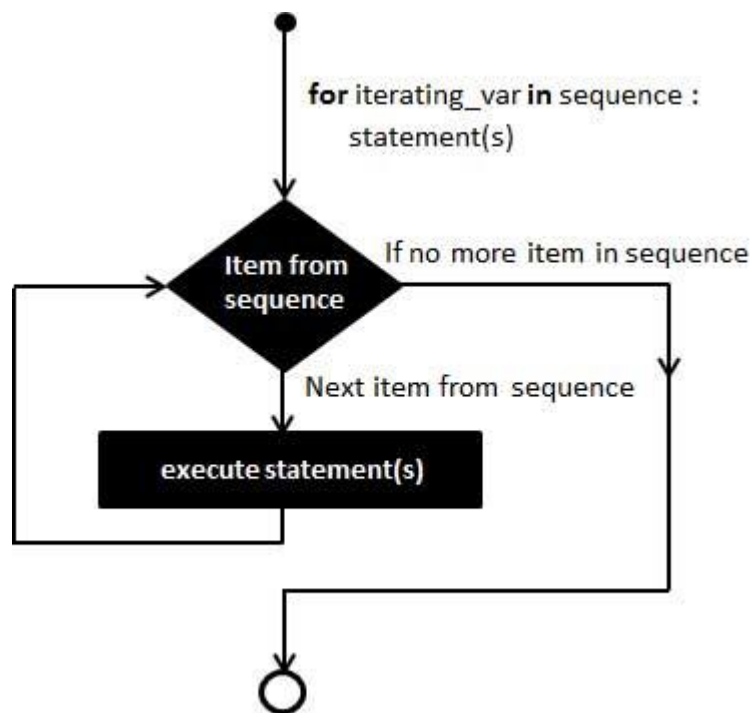
# While语句

- ◆ while 语句用于循环执行程序，即在某条件下，循环执行某段程序，以处理需要重复处理的相同任务





- ◆ for循环可以遍历任何序列的项目，如一个列表或者一个字符串



## ◆ Python for 循环嵌套语法

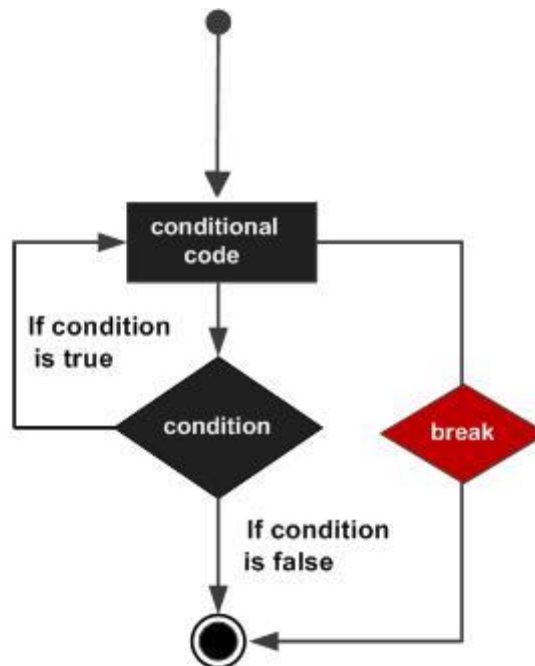
```
for iterating_var in sequence:
    for iterating_var in sequence:
        statements(s)
    statements(s)
```

## ◆ Python while 循环嵌套语法

```
while expression:
    while expression:
        statement(s)
    statement(s)
```

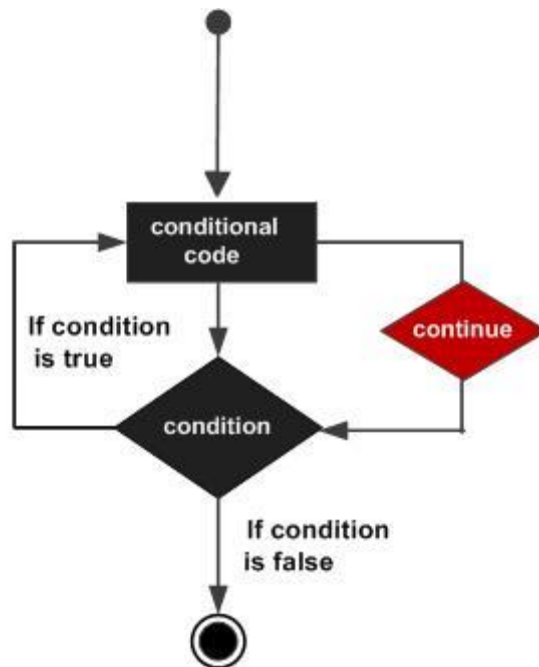
# break 语句

- ◆ Python break语句，就像在C语言中，打破了最小封闭for或while循环



# continue 语句

- ◆ Python continue 语句跳出本次循环，而break跳出整个循环。



# pass 语句

- ◆ Python pass是空语句，是为了保持程序结构的完整性。
- ◆ pass 不做任何事情，一般用做占位语句。

函数	返回值 (描述)
<a href="#"><code>abs(x)</code></a>	返回数字的绝对值, 如 <code>abs(-10)</code> 返回 10
<a href="#"><code>ceil(x)</code></a>	返回数字的上入整数, 如 <code>math.ceil(4.1)</code> 返回 5
<a href="#"><code>cmp(x, y)</code></a>	如果 $x < y$ 返回 -1, 如果 $x == y$ 返回 0, 如果 $x > y$ 返回 1
<a href="#"><code>exp(x)</code></a>	返回e的x次幂( $e^x$ ), 如 <code>math.exp(1)</code> 返回2.718281828459045
<a href="#"><code>fabs(x)</code></a>	返回数字的绝对值, 如 <code>math.fabs(-10)</code> 返回10.0
<a href="#"><code>floor(x)</code></a>	返回数字的下舍整数, 如 <code>math.floor(4.9)</code> 返回 4
<a href="#"><code>log(x)</code></a>	如 <code>math.log(math.e)</code> 返回1.0, <code>math.log(100,10)</code> 返回2.0
<a href="#"><code>log10(x)</code></a>	返回以10为基数的x的对数, 如 <code>math.log10(100)</code> 返回 2.0
<a href="#"><code>max(x1, x2,...)</code></a>	返回给定参数的最大值, 参数可以为序列。
<a href="#"><code>min(x1, x2,...)</code></a>	返回给定参数的最小值, 参数可以为序列。
<a href="#"><code>modf(x)</code></a>	返回x的整数部分与小数部分, 两部分的数值符号与x相同, 整数部分以浮点型表示。
<a href="#"><code>pow(x, y)</code></a>	$x^y$ 运算后的值。
<a href="#"><code>round(x [,n])</code></a>	返回浮点数x的四舍五入值, 如给出n值, 则代表舍入到小数点后的位数。
<a href="#"><code>sqrt(x)</code></a>	返回数字x的平方根, 数字可以为负数, 返回类型为实数, 如 <code>math.sqrt(4)</code> 返回 2+0j

函数	描述
<a href="#"><code>choice(seq)</code></a>	从序列的元素中随机挑选一个元素，比如 <code>random.choice(range(10))</code> ，从0到9中随机挑选一个整数。
<a href="#"><code>randrange ([start,] stop [,step])</code></a>	从指定范围内，按指定基数递增的集合中获取一个随机数，基数缺省值为1
<a href="#"><code>random()</code></a>	随机生成下一个实数，它在[0,1)范围内。
<a href="#"><code>seed([x])</code></a>	改变随机数生成器的种子seed。如果你不了解其原理，你不必特别去设定seed，Python会帮你选择seed。
<a href="#"><code>shuffle(lst)</code></a>	将序列的所有元素随机排序
<a href="#"><code>uniform(x, y)</code></a>	随机生成下一个实数，它在[x,y]范围内。

函数	描述
<a href="#"><code>acos(x)</code></a>	返回x的反余弦弧度值。
<a href="#"><code>asin(x)</code></a>	返回x的反正弦弧度值。
<a href="#"><code>atan(x)</code></a>	返回x的反正切弧度值。
<a href="#"><code>atan2(y, x)</code></a>	返回给定的 X 及 Y 坐标值的反正切值。
<a href="#"><code>cos(x)</code></a>	返回x的弧度的余弦值。
<a href="#"><code>hypot(x, y)</code></a>	返回欧几里德范数 $\sqrt{x^2 + y^2}$ 。
<a href="#"><code>sin(x)</code></a>	返回的x弧度的正弦值。
<a href="#"><code>tan(x)</code></a>	返回x弧度的正切值。
<a href="#"><code>degrees(x)</code></a>	将弧度转换为角度,如 <code>degrees(math.pi/2)</code> , 返回 90.0
<a href="#"><code>radians(x)</code></a>	将角度转换为弧度



常量	描述
pi	数学常量 pi ( 圆周率 , 一般以 $\pi$ 来表示 )
e	数学常量 e , e即自然常数 ( 自然常数 ) 。

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	十六进制数，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

操作符	描述	实例
+	字符串连接	a + b 输出结果：HelloPython
*	重复输出字符串	a*2 输出结果：HelloHello
[]	通过索引获取字符串中字符	a[1] 输出结果 e
[ : ]	截取字符串中的一部分	a[1:4] 输出结果ell
in	成员运算符 - 如果字符串中包含给定的字符返回 True	<b>H in a</b> 输出结果 1
not in	成员运算符 - 如果字符串中不包含给定的字符返回 True	<b>M not in a</b> 输出结果 1
r/R	原始字符串 - 原始字符串：所有的字符串都是直接按照字面的意思来使用，没有转义特殊或不能打印的字符。原始字符串除在字符串的第一个引号前加上字母"r"（可以大小写）以外，与普通字符串有着几乎完全相同的语法。	<b>print r'\n'</b> prints \n 和 <b>print R'\n'</b> prints \n
%	格式字符串	

符 号	描述
%c	格式化字符及其ASCII码
%s	格式化字符串
%d	格式化整数
%u	格式化无符号整型
%o	格式化无符号八进制数
%x	格式化无符号十六进制数
%X	格式化无符号十六进制数（大写）
%f	格式化浮点数字，可指定小数点后的精度
%e	用科学计数法格式化浮点数
%E	作用同%e，用科学计数法格式化浮点数
%g	%f和%e的简写
%G	%f 和 %E 的简写
%p	用十六进制数格式化变量的地址

符号	功能
*	定义宽度或者小数点精度
-	用做左对齐
+	在正数前面显示加号( + )
<sp>	在正数前面显示空格
#	在八进制数前面显示零('0')，在十六进制前面显示'0x'或者'0X'(取决于用的是'x'还是'X')
0	显示的数字前面填充'0'而不是默认的空格
%	'%%'输出一个单一的'%'
(var)	映射变量(字典参数)
m.n.	m 是显示的最小总宽度,n 是小数点后的位数(如果可用的话)

## ◆ 系统时间的记录方法

## ◆ 时间元组

序号	字段	值
0	4位数年	2008
1	月	1 到 12
2	日	1到31
3	小时	0到23
4	分钟	
5	秒	0到61 (60或61 是闰秒)
6	一周的第几日	0到6 (0是周一)
7	一年的第几日	1到366 (儒略历)
8	夏令时	-1, 0, 1, -1是决定是否为夏令时的旗帜

序号	属性	值
0	tm_year	2008
1	tm_mon	1 到 12
2	tm_mday	1 到 31
3	tm_hour	0 到 23
4	tm_min	0 到 59
5	tm_sec	0 到 61 (60或61 是闰秒)
6	tm_wday	0到6 (0是周一)
7	tm_yday	1 到 366(儒略历)
8	tm_isdst	-1, 0, 1, -1是决定是否为夏令时的旗帜

## ◆ 自定义函数

- 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[expression]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。

## ◆ 参数

- 必备参数
- 命名参数
- 缺省参数
- 不定长参数

## ◆ 变量

- 全局变量
- 局部变量



- ◆ 模块让你能够有逻辑地组织你的Python代码段。
- ◆ 把相关的代码分配到一个 模块里能让你的代码更好用，更易懂。
- ◆ 模块也是Python对象，具有随机的名字属性用来绑定或引用。
- ◆ 简单地说，模块就是一个保存了Python代码的文件。模块能定义函数，类和变量。模块里也能包含可执行的代码。

- ◆ 包是一个分层次的文件目录结构，它定义了一个由模块及子包，和子包下的子包等组成的Python的应用环境。
- ◆ 考虑一个在Phone目录下的pots.py文件。这个文件有如下源代码：

```
def Pots():  
    print "I'm Pots Phone"
```

- ◆ 同样地，我们有另外两个保存了不同函数的文件：
  - Phone/Isdn.py 含有函数Isdn()
  - Phone/G3.py 含有函数G3()
- ◆ 现在，在Phone目录下创建file \_\_init\_\_.py：
  - Phone/\_\_init\_\_.py

```
from Pots import Pots  
from Isdn import Isdn  
from G3 import G3
```

- ◆ 打印到屏幕——print
- ◆ 读取键盘输入——input、raw\_input
- ◆ 打开文件——open

```
file object = open(file_name [, access_mode][, buffering])
```

- ◆ 关闭文件——close

```
fileObject.close();
```

- ◆ 写入文件——write

```
fileObject.write(string);
```

- ◆ 读取字符串——read

```
fileObject.read([count]);
```

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将其覆盖。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将其覆盖。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将其覆盖。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将其覆盖。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。

属性	描述
file.closed	返回true如果文件已被关闭，否则返回false。
file.mode	返回被打开文件的访问模式。
file.name	返回文件的名称。
file.softspace	如果用print输出后，必须跟一个空格符，则返回false。否则返回true。

◆ 异常即是一个事件，该事件会在程序执行过程中发生，影响了程序的正常执行

◆ 捕捉异常可以使用try/except语句。

```
try:
    <语句>          #运行别的代码
except <名字>:
    <语句>          #如果在 try 部份引发了'name'异常
except <名字>, <数据>:
    <语句>          #如果引发了'name'异常，获得附加的数据
else:
    <语句>          #如果没有异常发生
```

```
try:
    You do your operations here;
    .....
except (Exception1[, Exception2[,...ExceptionN]]):
    If there is any exception from the given exception list,
    then execute this block.
    .....
else:
    If there is no exception then execute this block.
```

```
try:
    <语句>
finally:
    <语句>          #退出 try 时总会执行
raise
```

```
try:
    You do your operations here;
    .....
except ExceptionType, Argument:
    You can print value of Argument here...
```

- ◆ Dataguru（炼数成金）是专业数据分析网站，提供教育，媒体，内容，社区，出版，数据分析业务等服务。我们的课程采用新兴的互联网教育形式，独创地发展了逆向收费式网络培训课程模式。既继承传统教育重学习氛围，重竞争压力的特点，同时又发挥互联网的威力打破时空限制，把天南地北志同道合的朋友组织在一起交流学习，使到原先孤立的学习个体组合成有组织的探索力量。并且把原先动辄成千上万的学习成本，直线下降至百元范围，造福大众。我们的目标是：低成本传播高价值知识，构架中国第一的网上知识流转阵地。
- ◆ 关于逆向收费式网络的详情，请看我们的培训网站 <http://edu.dataguru.cn>



# Thanks

## FAQ时间