

马哥教育



Python流程控制

主讲：马永亮(马哥)

QQ:113228115

客服QQ: 2813150558, 1661815153

<http://www.magedu.com>

<http://mageedu.blog.51cto.com>

马哥教育

www.magedu.com





Python流程控制之 if测试

主讲：马永亮(马哥)

QQ:113228115

客服QQ: 2813150558, 1661815153

<http://www.magedu.com>

<http://mageedu.blog.51cto.com>

❖ 所有的Python对象都支持比较操作

- ➡ 可用于测试相等性、相对大小等
- ➡ 如果是复合对象，Python会检查其所有部分，包括自动遍历各级嵌套对象，直到可以得出最终结果

❖ 测试操作符

- ➡ “==”操作符测试值的相等性
- ➡ “is”表达式测试对象的一致性

马哥教育

www.magedu.com



❖ Python中不同类型的比较方法

- ➡ 数字：通过相对大小进行比较
- ➡ 字符串：按照字典次序逐字符进行比较
- ➡ 列表和元组：自左至右比较各部分内容
- ➡ 字典：对排序之后的(键、值)列表进行比较

❖ Python中真和假的含义

- ➡ 非零数字为真，否则为假
- ➡ 非空对象为真，否则为假
- ➡ **None**则始终为假

马哥教育

www.magedu.com



❖ 组合条件测试

➔ and

➔ or

➔ not: 返回True或False

❖ 注意: Python中, and和or运算会返回真或假的对象, 而不是True或False

➔ 对象在本质上不是“真”, 就是“假”

❖ and和or是短路操作符

马哥教育

www.magedu.com



if测试的语法结构

❖ **if** `boolean_expression1`:

 ➔ `suite1`

❖ **elif** `boolean_expression2`:

 ➔ `suite2`

❖ ...

❖ **else**:

 ➔ `else_suite`

❖ **elif**语句是可选的

马哥教育

❖ 仅用于占位，而后再填充相关语句时，可以使用**pass**



❖ 通常在为某变量设定默认值时通常用到的如下格式

➡ if X:

 ➡ A = Y

➡ else:

 ➡ A = Z

❖ 可以改写为如下简短格式

➡ A = Y if X else Z

❖ 其通用条件表达式语法格式为

➡ **expression1 if *boolean_expression* else expression2**

➡ 如果boolean_expression的值为True，则条件表达式的结果为expression1，否则为expression2



Python流程控制之 while和for循环

主讲：马永亮(马哥)

QQ:113228115

客服QQ: 2813150558, 1661815153

<http://www.magedu.com>

<http://mageedu.blog.51cto.com>

循环机制及应用场景

❖ while循环

- ➡ 用于编写通用迭代结构
- ➡ 顶端测试为真即会执行循环体，并会重复多次测试直到为假后执行循环后的其它语句

❖ for循环

- ➡ 一个通用的序列迭代器，用于遍历任何有序的序列对象内的元素
- ➡ 可用于字符串、元组、列表和其它的内置可迭代对象，以及通过类所创建的新对象

❖ Python也提供了一些能够进行隐性迭代的工具

- ➡ in成员关系测试
- ➡ 列表解析
- ➡ map、reduce和filter函数



❖ 语法格式

➡ **while** *boolean_expression*:

 ➡ *while_suite*

➡ **else**:

 ➡ *else_suite*

➡ **else**分支为可选部分

➡ 只要**boolean_expression**的结果为**True**，循环就会执行；

➡ **boolean_expression**的结果为**False**时终止循环，此时如果有**else**分支，则会执行

www.magedu.com



❖ break

- ➡ 跳出所处的最近层循环

❖ continue

- ➡ 跳到所处的最近层循环的开始处

❖ pass

- ➡ 占位语句
- ➡ 当语法需要语句但还没有任何实用语句可写时使用

❖ else代码块

- ➡ 只要循环是正常终止，**else**分支就会执行
- ➡ 在由于**break**语句、或由于返回语句(如果循环在函数或方法内)、或由于发生异常导致跳出循环，则**else**分支不会执行



❖ 语法格式

➡ **while** *boolean_expression1*:

 ➡ *while_suite*

 ➡ **if** *boolean_expression2*: **break**

 ➡ **if** *boolean_expression3*: **continue**

➡ **else**:

 ➡ *else_suite*

马哥教育

www.magedu.com



❖ 语法格式

➞ **for** *expression1* **in** *iterable*:

 ➞ *for_suite*

➞ **else**:

 ➞ *else_suite*

➞ 通常，**expression**或是一个单独的变量，或是一个变量序列，一般以元组的形式给出

➞ 如果以元组或列表用于**expression**，则其中的每个数据项都会拆分到表达式的项，例如

➞ **T**=[(1,2),(3,4),(5,6),(7,8)]

➞ **for** (a,b) **in** **T**:

 ● **print** a,b



❖ 语法格式

➡ *for expression in iterable:*

 ➡ *for_suite*

 ➡ *if boolean_expression2: break*

 ➡ *if boolean_expression3: continue*

➡ *else:*

 ➡ *else_suite*

马哥教育

www.magedu.com



编写循环的技巧

- ❖ **for**循环比**while**循环执行速度快
- ❖ **Python**提供了两个内置函数，用于在**for**循环中定制特殊的循环
 - ➔ **range**或**xrange**
 - **range**: 一次性地返回连续的整数列表
 - **xrange**: 一次产生一个数据元素，相较于**range**更节约空间
 - ➔ **zip**
 - 返回并行的元素元组的列表，常用于在**for**循环中遍历数个序列

马哥教育

www.magedu.com



❖ 非完备遍历

➡ 用于每隔一定的个数元素挑选一个元素

```
>>> S = 'How are you these days?'
>>> range(0, len(S), 2)
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22]
>>> for i in range(0, len(S), 2): print S[i],
...
H w a e y u t e e d y ?
```

❖ 修改列表

```
>>> L = [1, 2, 3, 4, 5]
>>> for i in range(len(L)): L[i] += 1
...
>>> L
[2, 3, 4, 5, 6]
```

马司教育

www.magedu.com



并行遍历

❖ zip

- ➡ 取得一个或多个序列为参数，将给定序列中的并排的元素配成元组，返回这些元组的列表
 - 当参数长度不同时，**zip**会以最短序列的长度为准
- ➡ 可在**for**循环中用于实现并行迭代

```
>>> L1 = [1, 2, 3, 4, 5, 6, 7]
>>> L2 = ['a', 'b', 'c', 'd', 'e', 'f', 'g']
>>> zip(L1, L2)
[(1, 'a'), (2, 'b'), (3, 'c'), (4, 'd'), (5, 'e'), (6, 'f'), (7, 'g')]
```

➡ zip也常用于动态构造字典

```
>>> keys = [1, 2, 3, 4, 5, 6, 7]
>>> values = ['Mon', 'Tus', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun']
>>> D = {}
>>> for (k, v) in zip(keys, values): D[k] = v
>>> D
{1: 'Mon', 2: 'Tus', 3: 'Wed', 4: 'Thu', 5: 'Fri', 6: 'Sat', 7: 'Sun'}
```



马哥教育



Python迭代器和列表解析

主讲：马永亮(马哥)

QQ:113228115

客服QQ: 2813150558, 1661815153

<http://www.magedu.com>

<http://mageedu.blog.51cto.com>

❖ 迭代：重复做一件事

❖ **iterable(可迭代)**对象

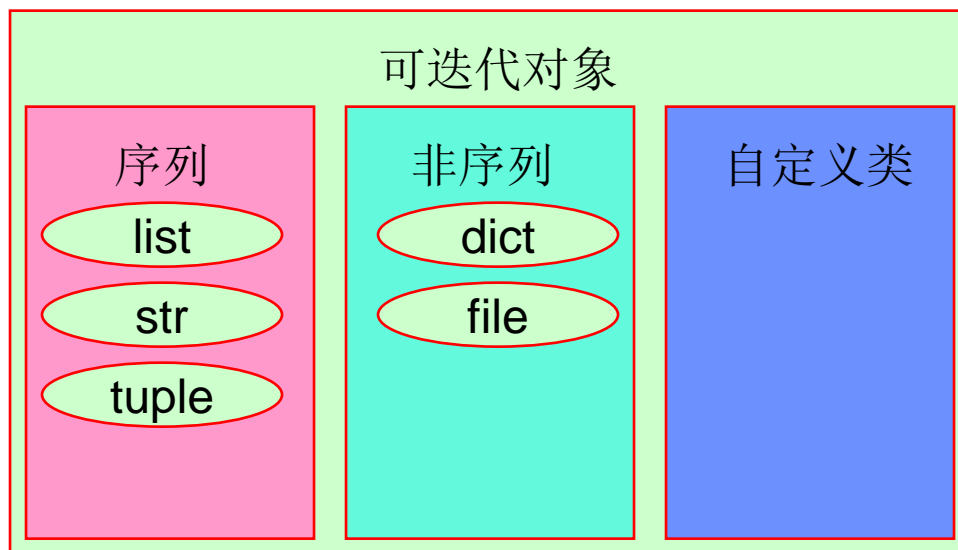
➡ 支持每次返回自己所包含的一个成员的对象

➡ 对象实现了**__iter__**方法

➤ 序列类型，如：**list, str, tuple**

➤ 非序列类型，如：**dict, file**

➤ 用户自定义的一些包含了**__iter__()** 或 **__getitem__()** 方法的类



- ❖ 迭代器(**iterator**)又称游标(**cursor**)，它是程序设计的软件设计模式，是一种可在容器物件(**container**，如列表等)上实现元素遍历的接口
- ❖ 迭代器是一种特殊的数据结构，当然在**Python**中，它也是以对象的形式存在的
 - ➔ 简单理解方式：对于一个集体中的每一个元素，想要执行遍历，那么针对这个集体的迭代器定义了遍历集体中每一个元素的顺序或者方法

马哥教育

www.magedu.com



- ❖ 在Python中，迭代器是遵循迭代协议的对象
- ❖ 使用`iter()`可从任何序列对象中得到迭代器
- ❖ 若要实现迭代器，需要在类中定义`next()`方法(Python 3中是`__next__()`)
- ❖ 要使得迭代器指向下一个元素，则使用成员函数`next()`
 - ➡ 在Python中，是函数`next()`，而非成员函数
- ❖ 当没有元素时，则引发 **StopIteration**异常
- ❖ **for**循环可用于任何可迭代对象
 - ➡ **for**循环开始时，会通过迭代协议传递给`iter()`内置函数，从而能够从可迭代对象中获得一个迭代器，返回的对象含有需要的`next`方法



❖ 列表解析是python迭代机制的一种应用，它常用于实现创建新的列表，因此要放置于[]中

❖ 语法：

➡ [expression for iter_var in iterable]

➡ [expression for iter_var in iterable if cond_expr]

```
>>> L = [x**2 for x in range(9)]
>>> L
[0, 1, 4, 9, 16, 25, 36, 49, 64]
```

```
>>> import os
>>> FILES=os.listdir('/var/log')
>>> [ filename for filename in FILES if filename.endswith('.log') ]
['anaconda.log', 'dracut.log', 'anaconda.program.log', 'yum.log', 'wpa_supplicant.log', 'anaconda.yum.log', 'spice-vdagent.log', 'boot.log', 'anaconda.storage.log', 'anaconda.ifcfg.log']
```



产生偏移和元素

❖ enumerate

- ➔ **range**可在非完备遍历中用于生成索引偏移，而非偏移处的元素
- ➔ 如果同时需要偏移索引和偏移元素，则可以使用**enumerate()**函数
- ➔ 此内置函数返回一个生成器对象

```
>>> S = 'hello world'
>>> E = enumerate(S)
>>> E.next()
(0, 'h')
>>> E.next()
(1, 'e')
```

马哥教育

www.magedu.com



生成器表达式

- ❖ 生成器表达式并不真正创建数字列表，而是返回一个生成器对象，此对象在每次计算出一个条目后，把这个条目“产生”(yield)出来
 - ➔ 生成器表达式使用了“惰性计算”或称作“延迟求值”的机制
- ❖ 序列过长，并且每次只需要获取一个元素时，应当考虑使用生成器表达式而不是列表解析
 - ➔ 生成器表达式于python 2.4引入
- ❖ 语法
 - ➔ (expr for iter_var in iterable)
 - ➔ (expr for iter_var in iterable if cond_expr)

马哥教育
www.magedu.com



- ❖ 博客: <http://magedu.blog.51cto.com>
- ❖ 主页: <http://www.magedu.com>
- ❖ QQ: 2813150558, 1661815153, 113228115
- ❖ QQ群: 203585050, 279599283



马哥教育

Thank You!