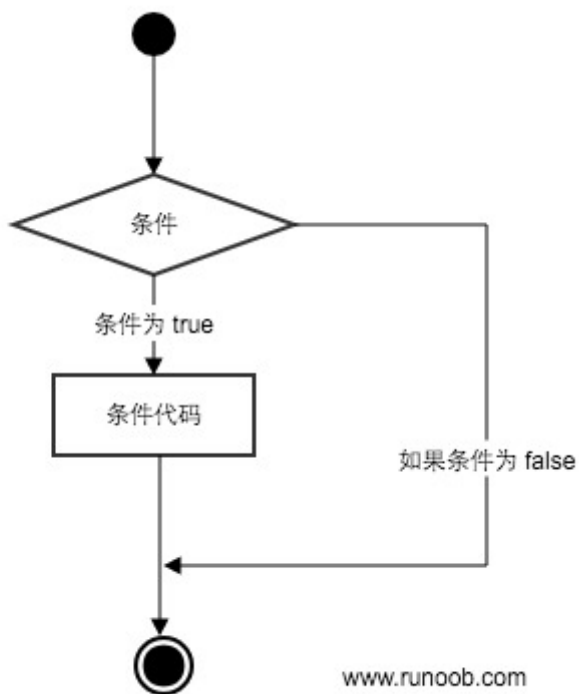


Python 条件语句

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。

可以通过下图来简单了解条件语句的执行过程：



Python程序语言指定任何非0和非空（ null ）值为true，0 或者 null为false。

Python 编程中 if 语句用于控制程序的执行，基本形式为：

```
if 判断条件:
    执行语句.....
else:
    执行语句.....
```

其中"判断条件"成立时（非零），则执行后面的语句，而执行内容可以多行，以缩进来区分表示同一范围。

else 为可选语句，当需要在条件不成立时执行内容则可以执行相关语句，具体例子如下：

实例

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
# 例1: if 基本用法
flag = False
name = 'luren'
if name == 'python': # 判断变量否为'python'
    flag = True # 条件成立时设置标志为真
```

```
print 'welcome boss' # 并输出欢迎信息
else:
print name # 条件不成立时输出变量名称
```

输出结果为：

```
luren          # 输出结果
```

if 语句的判断条件可以用>（大于）、<（小于）、==（等于）、>=（大于等于）、<=（小于等于）来表示其关系。

当判断条件为多个值时，可以使用以下形式：

```
if 判断条件1:
    执行语句1.....
elif 判断条件2:
    执行语句2.....
elif 判断条件3:
    执行语句3.....
else:
    执行语句4.....
```

实例如下：

实例

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
# 例2: elif用法
num = 5
if num == 3: # 判断num的值
print 'boss'
elif num == 2:
print 'user'
elif num == 1:
print 'worker'
elif num < 0: # 值小于零时输出
print 'error'
else:
print 'roadman' # 条件均不成立时输出
```

输出结果为：

```
roadman          # 输出结果
```

由于 python 并不支持 switch 语句，所以多个条件判断，只能用 elif 来实现，如果判断需要多个条件需同时判断时，可以使用 or（或），表示两个条件有一个成立时判断条件成功；使用 and（与）时，表示只有两个条件同时成立的情况下，判断条件才成功。

实例

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
# 例3: if语句多个条件
num = 9
if num >= 0 and num <= 10: # 判断值是否在0~10之间
print 'hello'
# 输出结果: hello
num = 10
if num < 0 or num > 10: # 判断值是否在小于0或大于10
print 'hello'
else:
print 'undefine'
# 输出结果: undefine
num = 8
# 判断值是否在0~5或者10~15之间
if (num >= 0 and num <= 5) or (num >= 10 and num <= 15):
print 'hello'
else:
print 'undefine'
# 输出结果: undefine
```

当if有多个条件时可使用括号来区分判断的先后顺序，括号中的判断优先执行，此外 and 和 or 的优先级低于>（大于）、<（小于）等判断符号，即大于和小于在没有括号的情况下会比与或要优先判断。

简单的语句组

你也可以在同一行的位置上使用if条件判断语句，如下实例：

实例

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
var = 100
if ( var == 100 ) : print "变量 var 的值为100"
print "Good bye!"
```

以上代码执行输出结果如下：

```
变量 var 的值为100
Good bye!
```

← Python 运算符

Python While 循环语句 →



4 篇笔记

写笔记

