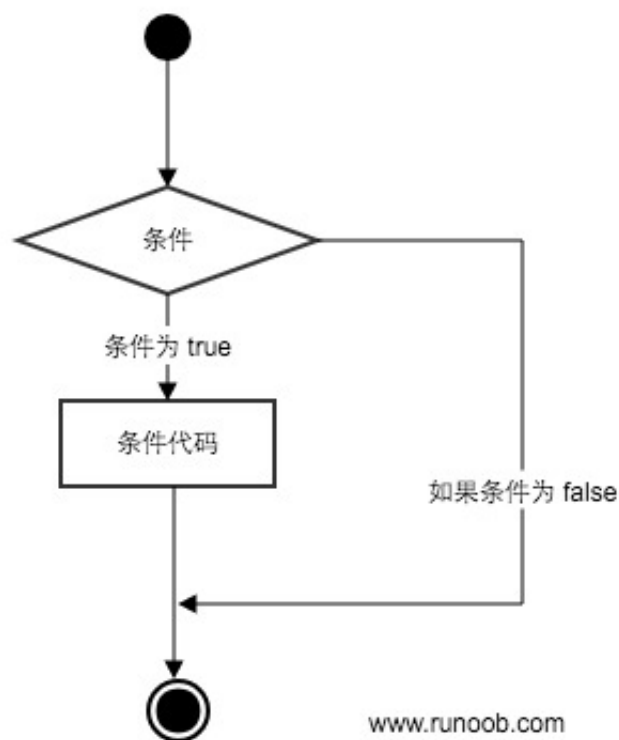


# 条件语句

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：



---

## if语句

Python中if语句的一般形式如下所示：

```
if condition_1:
    statement_block_1
elif condition_2:
    statement_block_2
else:
    statement_block_3
```

- 如果 "condition1" 为 *True* 将执行 "statementblock\_1" 块语句
- 如果 "condition1" 为 *False*, 将判断 "condition2"
- 如果 "condition2" 为 *True* 将执行 "statementblock\_2" 块语句

- 如果 "condition2" 为 *False*, 将执行 "statementblock\_3" 块语句

Python 中用 `elif` 代替了 `else if`, 所以 `if` 语句的关键字为: `if – elif – else`。

注意:

- 1、每个条件后面要使用冒号 `:`, 表示接下来是满足条件后要执行的语句块。
- 2、使用缩进来划分语句块, 相同缩进数的语句在一起组成一个语句块。

## 3、在Python中没有switch – case语句。

---

### if 嵌套

---

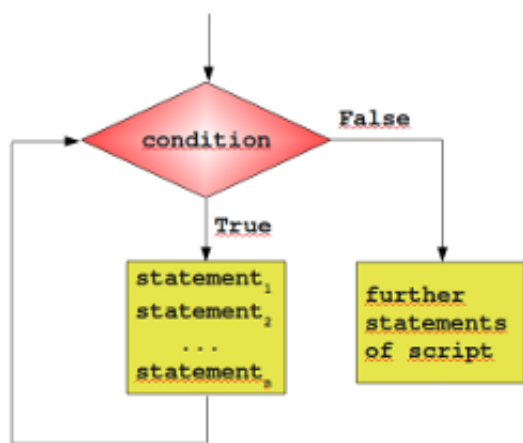
在嵌套 `if` 语句中, 可以把 `if...elif...else` 结构放在另外一个 `if...elif...else` 结构中。

```
if 表达式1:
    语句
    if 表达式2:
        语句
    elif 表达式3:
        语句
    else:
        语句
elif 表达式4:
    语句
else:
    语句
```

---

## Python3 循环语句

本章节将为大家介绍Python循环语句的使用。Python中的循环语句有 `for` 和 `while`。Python循环语句的控制结构图如下所示:



---

## while 循环

---

Python中while语句的一般形式：

```
while 判断条件:
    语句
```

同样需要注意冒号和缩进。另外，在Python中没有do..while循环。以下实例使用了 while 来计算 1 到 100 的总和：

```
n = 100

sum = 0
counter = 1
while counter <= n:
    sum = sum + counter
    counter += 1

print("1 到 %d 之和为: %d" % (n,sum))
```

我们可以通过设置条件表达式永远不为 false 来实现无限循环，实例如下：

```
var = 1
while var == 1 : # 表达式永远为 true
    num = int(input("输入一个数字  :"))
    print ("你输入的数字是: ", num)

print ("Good bye!")
```

你可以使用 `ctrl+c` 来退出当前的无限循环。无限循环在服务器上客户端的实时请求非常有用。

## while 循环使用 else 语句

---

在 while ... else 在条件语句为 false 时执行 else 的语句块：

```
count = 0
while count < 5:
    print (count, " 小于 5")
    count = count + 1
else:
    print (count, " 大于或等于 5")
```

## for 语句

---

Python 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。for循环的一般格式如下：

```
for <variable> in <sequence>:
    <statements>
else:
    <statements>
```

## range()函数

---

如果你需要遍历数字序列，可以使用内置range()函数。它会生成数列，例如：

```
for i in range(5):
    print(i)
```

你也可以使用range指定区间的值：

```
for i in range(5,9) :
    print(i)
```

也可以使range以指定数字开始并指定不同的增量(甚至可以是负数，有时这也叫做'步长')：

```
for i in range(0, 10, 3) :
    print(i)
```

## break和continue语句及循环中的else子句

---

break 语句可以跳出 for 和 while 的循环体。如果你从 for 或 while 循环中终止，任何对应的循环 else 块将不执行。实例如下：

```

for letter in 'Runoob':      # 第一个实例
    if letter == 'b':
        break
    print ('当前字母为 :', letter)

var = 10                     # 第二个实例
while var > 0:
    print ('当期变量值为 :', var)
    var = var -1
    if var == 5:
        break

print ("Good bye!")

```

continue语句被用来告诉Python跳过当前循环块中的剩余语句，然后继续进行下一轮循环。

for letter in 'Runoob': # 第一个实例 if letter == 'o': # 字母为 o 时跳过输出 continue print ('当前字母 :', letter)

```

var = 10                     # 第二个实例
while var > 0:
    var = var -1
    if var == 5:             # 变量为 5 时跳过输出
        continue
    print ('当前变量值 :', var)
print ("Good bye!")

```

循环语句可以有 else 子句，它在穷尽列表(以for循环)或条件变为 false (以while循环)导致循环终止时被执行，但循环被break终止时不执行。如下实例用于查询质数的循环例子：

```

for n in range(2, 10):
    for x in range(2, n):
        if n % x == 0:
            print(n, '等于', x, '*', n//x)
            break
    else:
        # 循环中没有找到元素
        print(n, ' 是质数')

```

## 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，如下实例