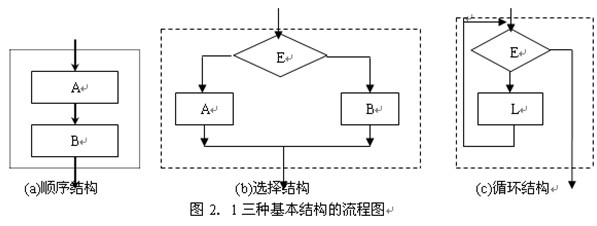
# Python程序流程控制

## 三种程序流程控制

计算机程序在解决某个具体问题时，包括三种情形，即顺序执行所有的语句、选择执行部分的语句和循环执行部分语句，这正好对应着程序设计中的三种程序执行结构流程：**顺序结构、选择结构和循环结构**。



## 顺序结构

程序的顺序结构流程不需要单独的关键字来控制，就是一行行的执行，从前往后按先后顺序执行。

a = 10

b = 20

c = a + b

Print(c)

## 选择结构

### if分支语句

**（1）、单分支**

**语法：**

if 判断条件:

代码块

实例：判断指定的uid是不是root用户

uid = 0

if uid == 0:

print("root")

也可以这样写：

uid = 0

if uid == 0: print("root")

**（2）、双分支**

****语法：****

if 判断条件:

代码块

else:

代码块

实例：

uid = 100

if uid == 0:

print("root")

else:

print("Common user")

**（3）、多分支**

****语法：****

if 判断条件1:

代码块1

elif 判断条件2:

代码块2

...

elif 判断条件n:

代码块n

else:

默认代码块

实例：根据学生分数输出等级

score = 88.8

level = int(score % 10)

if level >= 10:

print('Level A+')

elif level == 9:

print('Level A')

elif level == 8:

print('Level B')

elif level == 7:

print('Level C')

elif level == 6:

print('Level D')

else:

print('Level E')

## 循环结构

当我们需要多次执行一个代码语句或代码块时，可以使用循环语句。Python中提供的循环语句有：while循环和for循环。

### while循环语句

语法：

while 判断条件:

代码块

实例：

count = 0

while count <= 9:

print(count, end=' ')

count = count + 1

while死循环

while True:

print("这是一个死循环")

编写代码通常要避免死循环，除非是业务需要，让程序一直运行在一个循环里。

while..else

语法：

while 判断条件:

代码块

else:

代码块

else中的代码块会在while循环正常执行完的情况下执行，如果while循环被break中断，else中的代码块不会执行。

实例：

count = 0

while count <=9:

print(count, end=' ')

count += 1

else:

print('end')

执行结果为：0 1 2 3 4 5 6 7 8 9 end

### for循环语句

语法：

for 临时变量 in 可迭代对象:

代码块

迭代对象如果是一个值范围，其原则是前闭后开。闭表示包含该值，开表示不包含该值。

for i in range(0, 10):

print(i)

for i in range(10, 0, -1):

print(i)

会打印出10,9,....1

for循环还常用于遍历序列(如list、tuple、range、str)、集合(如 set)和映射对象(如dict)。

实例：

names = ['Tom', 'Peter', 'Jerry', 'Jack']

for name in names:

print(name)

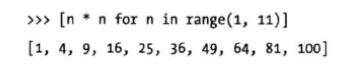
对于序列，也通过索引进行迭代：

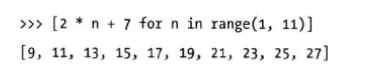
names = ['Tom', 'Peter', 'Jerry', 'Jack']

for i in range(len(names)):

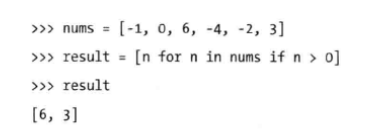
print(names[i])

利用for循环快速创建列表





使用列表进行筛选：



for...else

与while..else基本一致，不再赘述。

### 循环控制语句

可以更改循环体中程序的执行过程，如中断循环、跳过本次循环。

| **循环控制语句** | **说明** |
| --- | --- |
| break | 终止整个循环 |
| contine | 跳过本次循环，执行下一次循环 |
| pass | pass语句是个空语句，只是为了保持程序结构的完整性，没有什么特殊含义。pass语句并不是只能用于循环语句中，也可以用于分支语句中。 |

（1）、break中断循环

count = 0

while count <=9:

print(count, end=' ')

if count == 5:

break

count += 1

else:

print('end')

print(‘other code…’)

运行结果：0 1 2 3 4 5 other code…

（2）、continue继续下次循环

for i in range(10):

if i % 2 == 0:

continue

print(i, end=' ')

输出结果：1 3 5 7 9

（3）、pass空语句

pass语句什么也不做，一般作为占位符或者创建占位程序，pass语句不会执行任何操作，比如：

i = 0

while i < 10:

if i % 2 == 0:

pass

else:

print(str(i) + '是奇数')

i = i + 1

（4）、循环嵌套

j = 1

while j <= 9:

i = 1

while i <= j:

print('%d\*%d=%d' % (i, j, i\*j), end='\t')

i += 1

print()

j += 1