循环

廖雪峰
资深软件开发工程师，业余马拉松选手。

要计算1+2+3，我们可以直接写表达式：

```
>>> 1 + 2 + 3
6
```

要计算1+2+3+...+10，勉强也能写出来。

但是，要计算1+2+3+...+10000，直接写表达式就不可能了。

为了让计算机能计算成千上万次的重复运算，我们就需要循环语句。

Python的循环有两种，一种是for...in循环，依次把list或tuple中的每个元素迭代出来，看例子：

```
names = ['Michael', 'Bob', 'Tracy']
for name in names:
    print(name)
```

执行这段代码，会依次打印 `names` 的每一个元素：

```
Michael
Bob
Tracy
```

所以 `for x in ...` 循环就是把每个元素代入变量 `x` ，然后执行缩进块的语句。

再比如我们想计算1-10的整数之和，可以用一个 `sum` 变量做累加：

```
sum = 0
for x in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]:
    sum = sum + x
print(sum)
```

如果要计算1-100的整数之和，从1写到100有点困难，幸好Python提供一个 `range()` 函数，可以生成一个整数序列，再通过 `list()` 函数可以转换为list。比如 `range(5)` 生成的序列是从0开始小于5的整数：

```
>>> list(range(5))
[0, 1, 2, 3, 4]
```

`range(101)` 就可以生成0-100的整数序列，计算如下：

```
sum = 0
for x in range(101):
    sum = sum + x
print(sum)
```

请自行运行上述代码，看看结果是不是当年高斯同学心算出的5050。

第二种循环是while循环，只要条件满足，就不断循环，条件不满足时退出循环。比如我们要计算100以内所有奇数之和，可以用while循环实现：

```
sum = 0
n = 99
while n > 0:
    sum = sum + n
    n = n - 2
print(sum)
```

在循环内部变量 `n` 不断自减，直到变为 `-1` 时，不再满足while条件，循环退出。

练习

请利用循环依次对list中的每个名字打印出 `Hello, xxx!` ：

```
L = ['Bart', 'Lisa', 'Adam']
```

break

在循环中， `break` 语句可以提前退出循环。例如，本来要循环打印1～100的数字：

```
n = 1
while n <= 100:
    print(n)
    n = n + 1
print('END')
```

上面的代码可以打印出1~100。

如果要提前结束循环，可以用 `break` 语句：

```
n = 1
while n <= 100:
    if n > 10: # 当n = 11时，条件满足，执行break语句
        break # break语句会结束当前循环
    print(n)
    n = n + 1
print('END')
```

执行上面的代码可以看到，打印出1~10后，紧接着打印 `END` ，程序结束。

可见 `break` 的作用是提前结束循环。

continue

在循环过程中，也可以通过 `continue` 语句，跳过当前的这次循环，直接开始下一次循环。

```
n = 0
while n < 10:
    n = n + 1
    print(n)
```

上面的程序可以打印出1～10。但是，如果我们想只打印奇数，可以用 `continue` 语句跳过某些循环：

```
n = 0
while n < 10:
    n = n + 1
    if n % 2 == 0: # 如果n是偶数，执行continue语句
        continue # continue语句会直接继续下一轮循环，后续的print()语句不会执行
    print(n)
```

执行上面的代码可以看到，打印的不再是1～10，而是1，3，5，7，9。

可见 `continue` 的作用是提前结束本轮循环，并直接开始下一轮循环。

参考源码

do_for.py

do_while.py

小结

循环是让计算机做重复任务的有效的方法。

`break` 语句可以在循环过程中直接退出循环，而 `continue` 语句可以提前结束本轮循环，并直接开始下一轮循环。这两个语句通常都 ***必须*** 配合 `if` 语句使用。

要特别注意，不要滥用 `break` 和 `continue` 语句。 `break` 和 `continue` 会造成代码执行逻辑分叉过多，容易出错。大多数循环并不需要用到 `break` 和 `continue` 语句，上面的两个例子，都可以通过改写循环条件或者修改循环逻辑，去掉 `break` 和 `continue` 语句。

有些时候，如果代码写得有问题，会让程序陷入“死循环”，也就是永远循环下去。这时可以用 `Ctrl+C` 退出程序，或者强制结束Python进程。

请试写一个死循环程序。

« 模式匹配　　使用dict和set »

Comments

Comments loaded. To post a comment, please Sign In

殒-_-落 @ 2025/12/22 02:50:24

想要获取当前遍历值和索引使用enumerate(列表)-》键值对，index，value

```
names = ['Adam', 'Bart', 'Bob']
for index, name in enumerate(names):
    print(f'index:{index}, name:{name}')
```

index:0, name:Adam index:1, name:Bart index:2, name:Bob

me、Tranquility @ 2025/12/15 07:51:16

for x in L: name=(f'Hello,{x}!') print(name)

寄雨秋风 @ 2025/12/12 09:09:58

```
#4种写法
L = ['Bart','Lisa','Adam']
for name in L:
    # print("Hello,",name,"!")  # 用逗号隔开不限各种类型
    # print("Hello," + name + "!") # 用+号只能用字符串
    # print(f"Hello,{name}!")  # 用f-string Python 3.6+
    print("Hello,{}!".format(name)) # 用format()
```

北方烟火 @ 2025/12/10 04:52:59

无意中写了一个死循环

```
#定义正确密码
user_password = "123456"
#获取用户输入的密码
user_input = input('请输入密码:')
if user_input == user_password:
    print('✅密码正确!开始计算1-100内偶数的和...')
    sum=0
    for num in range(1,101):
        if num % 2 == 0:
            sum += num
    print(f'\n1-100内所有偶数的和为:{sum}')
else:
    print('❎您的密码不正确，请重新输入')
    user_input = input('请输入密码:')
```

Read More ▼

北方烟火 @ 2025/12/10 03:07:35

3行代码，print命令缩进后，会循环打印出所有list里的字符

```
L=['Bart','Lisa','Adam']
for x in L:
    print('hello', x,'!')
```

某不科学动物 @ 2025/12/6 07:43:52

```
L = ['Bart', 'Lisa', 'Adam']
for name in L:
    print('Hello! ','%s' % name)
```

啊，你以为我傻 @ 2025/11/28 04:21:33

L = ['Bart','Lisa','Adam'] for x in L: print('Hello,',x,'!')

【毅】行天下 @ 2025/11/27 02:23:27

> *n=0 sum=0 while n<101: ... n=n+1 ... if n%2==0: ... continue ... sum=sum+n ... print(n) ... print(sum)*

Floris @ 2025/11/26 01:55:41

```
L = ['Bart', 'Lisa', 'Adam']
for index, elem in enumerate(L):
    print(f"{index}: {elem}")
```

卿绛 @ 2025/11/24 03:16:35

```
L = ['Bart', 'Lisa', 'Adam']
for x in L:
    print(f'Hello,{x}!:')
    print('Hello,%s!:'% x)
    print('Hello,{}!:'.format(x))
```

Hypersomnia @ 2025/11/21 02:28:11

```
L = ['Bart', 'Lisa', 'Adam']
for name in L:
    print('Hello,%s!' % name)
```

小蓝 @ 2025/11/20 07:17:27

L = ['Bart', 'Lisa', 'Adam'] for name in L: print('Hello,%s!' % name)

YU| ▷)) @ 2025/11/17 01:43:51

L = ['Bart', 'Lisa', 'Adam'] for L in L: print(f'hello {L}')

77 @ 2025/11/9 04:43:08

打卡

炬火 @ 2025/11/5 19:16:33

打卡第四天

Ciel Zero @ 2025/10/29 06:17:53

n=0 while n <= 0: if n < 10: print('bingo!'), else: print('chishi!') print('END')

红红火火恍恍惚惚全是 bingobingo hhhhhhh

夜卜 @ 2025/10/28 06:11:07

L = ['Bart', 'Lisa', 'Adam'] for x in L: print(f"Hello,{x}")

圆圈⭕ @ 2025/10/28 04:05:58

n = 11 while n > 10: if n%2 != 0:
print(f'{n}为奇数') n = n + 1

头发不要乱了 @ 2025/10/28 03:57:33

L = ['Bart', 'Lisa', 'Adam'] for i in L: print("hello,",i,"!")

Bacium @ 2025/10/24 21:43:39

```
# 依次打印"Hello XXX"
L = ['Bart', 'Lisa', 'Adam']


for name in L:
    print(f"Hello,{name}")
```
