

循环结构

应用场景

如果在程序中我们需要重复的执行某条或某些指令，例如用程序控制机器人踢足球，如果机器人持球而且还没有进入射门范围，那么我们就需要一直发出让机器人向球门方向奔跑的指令。当然你可能已经注意到了，刚才的描述中不仅仅有需要重复的动作，还需要用到上一章讲的分支结构。再举一个简单的例子，我们要实现一个每隔1秒中在屏幕上打印一次"hello, world"并持续打印一个小时的程序，我们肯定不能够直接把 `print('hello, world')` 这句代码写3600遍，如果真的要这样做，那么编程的工作就太无聊乏味了。因此，我们还需要了解一下循环结构，有了循环结构我们就可以轻松的控制某件事或者某些事重复、重复、再重复的去执行。

在Python中构造循环结构有两种做法，一种是 `for-in` 循环，一种是 `while` 循环。

for-in循环

如果明确的知道循环执行的次数或者要对一个容器进行迭代（后面会讲到），那么我们推荐使用 `for-in` 循环，例如下面代码中计算1~100求和的结果（ $\sum_{n=1}^{100} n$ ）。

```
"""
用for循环实现1~100求和

Version: 0.1
Author: 骆昊
"""

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

需要说明的是上面代码中的 `range(101)` 可以用来构造一个从0到100的取值范围，这样就可以构造出一个整数的序列并用于循环中，例如：

- `range(101)` 可以产生一个0到100的整数序列。
- `range(1, 100)` 可以产生一个1到99的整数序列。
- `range(1, 100, 2)` 可以产生一个1到99的奇数序列，其中2是步长，即数值序列的增量。

知道了这一点，我们可以用下面的代码来实现1~100之间的偶数求和。

```
"""
用for循环实现1~100之间的偶数求和

Version: 0.1
Author: 骆昊
"""

sum = 0
for x in range(2, 101, 2):
    sum += x
print(sum)
```

也可以通过在循环中使用分支结构的方式来实现相同的功能，代码如下所示。

```
"""
用for循环实现1~100之间的偶数求和

Version: 0.1
Author: 骆昊
"""

sum = 0
for x in range(1, 101):
    if x % 2 == 0:
        sum += x
print(sum)
```

while循环

如果要构造不知道具体循环次数的循环结构，我们推荐使用 `while` 循环。`while` 循环通过一个能够产生或转换出 `bool` 值的表达式来控制循环，表达式的值为 `True` 循环继续，表达式的值为 `False` 循环结束。下面我们通过一个“猜数字”的小游戏（计算机出一个1~100之间的随机数，人输入自己猜的数字，计算机给出对应的提示信息，直到人猜出计算机出的数字）来看看如何使用 `while` 循环。

```
"""
猜数字游戏
计算机出一个1~100之间的随机数由人来猜
计算机根据人猜的数字分别给出提示大一点/小一点/猜对了

Version: 0.1
Author: 骆昊
"""

import random

answer = random.randint(1, 100)
counter = 0
while True:
    counter += 1
    number = int(input('请输入: '))
    if number < answer:
        print('大一点')
    elif number > answer:
        print('小一点')
    else:
        print('恭喜你猜对了!')
        break
print('你总共猜了%d次' % counter)
if counter > 7:
    print('你的智商余额明显不足')
```

上面的代码中使用了 `break` 关键字来提前终止循环，需要注意的是 `break` 只能终止它所在的那个循环，这一点在使用嵌套的循环结构（下面会讲到）需要引起注意。除了 `break` 之外，还有另一个关键字是 `continue`，它可以用来放弃本次循环后续的代码直接让循环进入下一轮。

和分支结构一样，循环结构也是可以嵌套的，也就是说在循环中还可以构造循环结构。下面的例子演示了如何通过嵌套的循环来输出一个九九乘法表。

```

"""
输出乘法口诀表(九九表)

Version: 0.1
Author: 骆昊
"""

for i in range(1, 10):
    for j in range(1, i + 1):
        print('%d*%d=%d' % (i, j, i * j), end='\t')
    print()

```

练习

练习1：输入一个正整数判断是不是素数。

提示：素数指的是只能被1和自身整除的大于1的整数。

参考答案：

```

"""
输入一个正整数判断它是不是素数

Version: 0.1
Author: 骆昊
Date: 2018-03-01
"""

from math import sqrt

num = int(input('请输入一个正整数: '))
end = int(sqrt(num))
is_prime = True
for x in range(2, end + 1):
    if num % x == 0:
        is_prime = False
        break
if is_prime and num != 1:
    print('%d是素数' % num)
else:
    print('%d不是素数' % num)

```

练习2：输入两个正整数，计算它们的最大公约数和最小公倍数。

参考答案：

```

"""
输入两个正整数计算它们的最大公约数和最小公倍数

Version: 0.1
Author: 骆昊
Date: 2018-03-01
"""

x = int(input('x = '))
y = int(input('y = '))
# 如果x大于y就交换x和y的值

```

```

if x > y:
    # 通过下面的操作将y的值赋给x，将x的值赋给y
    x, y = y, x
# 从两个数中较的数开始做递减的循环
for factor in range(x, 0, -1):
    if x % factor == 0 and y % factor == 0:
        print('%d和%d的最大公约数是%d' % (x, y, factor))
        print('%d和%d的最小公倍数是%d' % (x, y, x * y // factor))
        break

```

练习3：打印如下所示的三角形图案。

```

*
**
***
****
*****

```

```

    *
   **
  ***
 ****
*****

```

```

      *
     ***
    *****
   ********
  *********
 *****

```

参考答案：

```

"""
打印三角形图案

Version: 0.1
Author: 骆昊
"""

row = int(input('请输入行数: '))
for i in range(row):
    for _ in range(i + 1):
        print('*', end='')
    print()

for i in range(row):
    for j in range(row):
        if j < row - i - 1:
            print(' ', end='')
        else:
            print('*', end='')
    print()

for i in range(row):

```

```
for _ in range(row - i - 1):  
    print(' ', end='')  
for _ in range(2 * i + 1):  
    print('*', end='')  
print()
```