

Python语言基础与应用

计算和控制流 / 控制流：迭代循环 (for)

陈斌 北京大学 gischen@pku.edu.cn



控制流：迭代循环 (for)

- › 什么是迭代循环
- › range函数
- › 跳出和继续

什么是迭代循环

› 迭代循环语句

- Python语言中的for语句实现了循环结构的
第一种循环前提

for <循环变量> in <可迭代对象>:

 <语句块1>

 break #跳出循环

 continue #略过余下循环语句

else: #迭代完毕，则执行

 <语句块2>

什么是迭代循环

› for循环语法

- for、in和else都是“保留字”
- 可迭代对象表示从这个数据对象中可以逐个取出数据项赋值给“循环变量”
- 可迭代对象有很多类型，如字符串、列表、元组、字典、集合等，也可以有后面提到的生成器、迭代器等

什么是迭代循环

```
>>> for n in range(5):  
    print (n)
```

```
0  
1  
2  
3  
4
```

```
>>> alist = ['a', 123, True]  
>>> for v in alist:  
    print (v)
```

```
a  
123  
True
```

```
>>> adic = {'name':'Tom', 'age':18, 'gender':'Male'}  
>>> for k in adic:  
    print (k, adic[k])
```

```
name Tom  
age 18  
gender Male
```

```
>>> for k, v in adic.items():  
    print (k, v)
```

```
name Tom  
age 18  
gender Male
```

range函数

› range函数

`range(<终点>)`

- 返回一个从0开始到终点的数列

`range(<起点>,<终点>)`

- 从0以外的任何整数开始构建数列

`range(<起点>,<终点>,<步长>)`

- 修改数列的步长，通过将步长设置为负数能够实现反向数列

› range构建的数列，包含起点整数，而不包含终点整数

range函数

› range类型的对象

直接当做序列

转换为list或者tuple等容器类型

```
>>> list(range(10))  
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
>>> list(range(5, 10))  
[5, 6, 7, 8, 9]  
>>> list(range(1, 10, 2))  
[1, 3, 5, 7, 9]  
>>> list(range(10, 1, -2))  
[10, 8, 6, 4, 2]  
>>> range(10)  
range(0, 10)  
>>> tuple(range(10))  
(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
```

range函数实例

› 验证 $3x+1$ 问题

```
for i in range(1, 10000):
    n = i
    step = 0
    while n != 1:
        if n % 2 == 0:
            n = n // 2
        else:
            n = n * 3 + 1
            step += 1
    else:
        print(i, "Traped!", step, "steps")
```

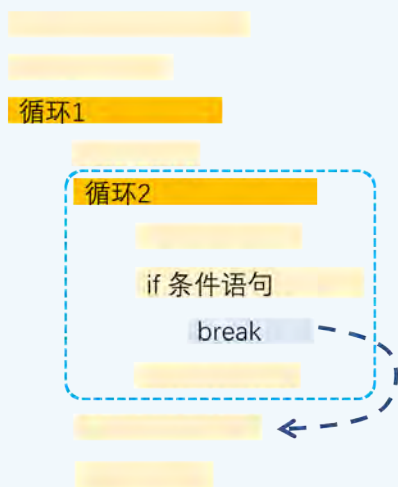
1167 Traped! 31 steps
1168 Traped! 119 steps
1169 Traped! 57 steps
1170 Traped! 57 steps
1171 Traped! 57 steps
1172 Traped! 119 steps
1173 Traped! 119 steps
1174 Traped! 119 steps
1175 Traped! 119 steps
1176 Traped! 119 steps
1177 Traped! 31 steps
1178 Traped! 119 steps

- 中断程序运行：CTRL+C
- 通过修改程序中range函数的参数，还可以验证其它范围的连续整数

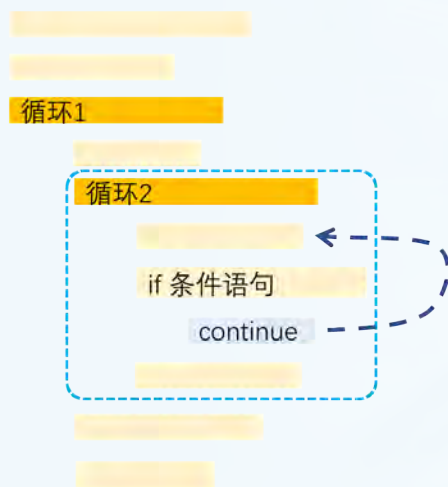
嵌套循环中的跳出和继续

› continue语句和break语句

- 都只能在循环内部使用
- 均作用于离它们最近的一层循环
- break是跳出当前循环并结束本层循环
- continue是略过余下循环语句并接着下一次循环



break语句跳出最近一层循环



continue语句忽略同一层循环剩余语句