

第6课 躲不过的重复—— for循环和while循环

课程目标

1. 掌握for...in...循环的格式与特点，了解其数据传递
2. 熟练掌握for...in...与range()函数的结合使用
3. 掌握while循环的格式与特点
4. 区别for...in...循环与while循环

课程难点

1. range()函数各个参数的含义
2. for...in...循环与while循环的变量更迭
3. while语句如何避免死循环

课程知识点总结

一、for...in...循环

1. i是变量名，可以根据需要取。
2. 遍历：逐一访问全部数据。
3. 用于处理已知循环次数或循环固定次数的问题。
4. 格式：
 - for...in...语句最后需要接冒号（一定要英文模式）
 - for...in...语句内部语句需要缩进，会被重复执行
1. in后可接列表、字典和字符串，但不允许接整数、浮点数。
 - 列表里的元素值依次赋值给变量 i，i 最终的值取决于最新一次的赋值。
 - 若是字典，则赋值给变量的是字典中的key，而不是value。即遍历的是key。
 - 对于字符串而言，会依次输出字符。

```
1 for i in [1,2,3]:
2     print(i*3)
3 #显示结果为:
4 #3
5 #6
6 #9
7
8 for i in '周杰伦':
9     print(i)
10 #显示结果为:
11 #周
```

```

12 #杰
13 #伦
14
15 city = {'北京':'天安门','上海':'东方明珠','广州':'珠江'}
16 for i in city:
17     print(i)
18 #显示结果为:
19 #北京
20 #上海
21 #广州
22
23 city = {'北京':'天安门','上海':'东方明珠','广州':'珠江'}
24 for i in city:
25     print(city[i])
26 #显示结果为:
27 #天安门
28 #东方明珠
29 #珠江

```

1. range()函数配合使用

可以用for...in...语句遍历range()函数生成的整数序列并打印。

```

1 for i in range(11,15):
2     print(i)
3 #显示结果为:
4 #11
5 #12
6 #13
7 #14

```

for i in range(n): , 表示循环n次。如下面例题中i=0, i=1, i=2时都执行一次语句内的代码, 也就是3次。

```

1 for i in range(0,3):
2     print('每天想你第'+str(i)+'遍')
3 #结果显示为:
4 #每天想你第0遍
5 #每天想你第1遍
6 #每天想你第2遍

```

二、range()函数

1. 使用range()函数, 可以生成一个整数序列。

2. 参数的意义:

- range(n)与range(0,n)一样, 都是生成一个从0到n-1的整数序列
- range(m,n), n>m, 生成从m到n-1的序列。取头不取尾。
- range(m,n,p), n>m, 生成m到n-1中间隔为p的整数序列。例如range(3,30,5)的意思就是从3开始每间隔5取一个数, 直到29, 结果为[3,8,13,18,23,28]。(只

有两个参数时，p默认为1。)

三、while循环

1. 与if条件判断类似，while后条件若满足，会进入语句内部循环直至条件不再满足或内部打断。
2. 用于处理未知循环次数或循环固定次数的问题。
3. 格式
 - 在while循环前要定义变量
 - 为避免陷入死循环，在循环内必须更新变量，如自加：`a=a+1`，也可以简写成：`a+=1`。
 - 循环体内部需要缩进！

```
1 x = 0
2 #非格式要求，但在while循环前必须要定义变量
3 while x < 6:
4     x = x+1
5     print(x)
6 #显示结果为：（不显示0是因为a先自加再输出，此时赋值已更新）
7 #1
8 #2
9 #3
10 #4
11 #5
12 #6
13 x = 0
14 while x < 6:
15     x = x+1
16 print(x)
17 #显示结果为：（因为print()在循环外，只输出最终x的赋值）
18 #6
```

四、while循环与for循环的差异

while循环与for循环的**最大区别在于**，while循环是基于条件判断的循环，而for循环则是基于容器的循环。对于while循环来说，当条件满足时，将一直处于循环状态，除非碰见break关键词；对于for循环来说，当容器内的元素没有迭代结束，则一直处于循环状态，同样碰见break关键词时也会退出循环。

所以，在做循环问题时，**首先自问循环过程中有没有明确的迭代对象（即容器）**，然后再根据判断结果，选择优先方案，即如果有迭代对象，则优先使用for循环，否则优先使用while循环。