

python 基础 06：循环

循环用于重复执行一些程序块。从上一讲的选择结构，我们已经看到了如何用缩进来表示程序块的隶属关系。循环也会用到类似的写法。

1、for 循环

for 循环需要预先设定好循环的次数(n)，然后执行隶属于 for 的语句 n 次。

基本构造是

```
for 元素 in 序列:    statement
```

举例来说，我们编辑一个叫 forDemo.py 的文件

```
for a in [3,4.4,'life']:    print a
```

这个循环就是每次从表[3,4.4,'life'] 中取出一个元素（回忆：表是一种序列），然后将这个元素赋值给 a，之后执行隶属于 for 的操作(print)。

介绍一个新的 Python 函数 range()，来帮助你建立表。

```
idx = range(5)print idx
```

可以看到 idx 是[0,1,2,3,4]

这个函数的功能是新建一个表。这个表的元素都是整数，从 0 开始，下一个元素比前一个大 1，直到函数中所写的上限（不包括该上限本身）

（关于 range()，还有丰富用法，有兴趣可以查阅，Python 3 中，range()用法有变化，见评论区）

举例

```
for a in range(10):    print a**2
```

while 循环

while 的用法是

```
while 条件:    statement
```

while 会不停地循环执行隶属于它的语句，直到条件为假(False)

举例

```
while i < 10:    print i    i = i + 1
```

2、中断循环

`continue` # 在循环的某一次执行中，如果遇到 `continue`, 那么跳过这一次执行，进行下一次的执行。

```
break    # 停止执行整个循环
```

```
for i in range(10):    if i == 2:
```

```
    continue    print i
```

当循环执行到 `i = 2` 的时候，`if` 条件成立，触发 `continue`, 跳过本次执行(不执行 `print`)，继续进行下一次执行(`i = 3`)。

```
for i in range(10):    if i == 2:                break    print i
```

当循环执行到 `i = 2` 的时候，`if` 条件成立，触发 `break`, 整个循环停止。

3、总结

`range()`

`for` 元素 `in` 序列:

while 条件:

continue

break