python in action

Lesson 4 - 条件判断和循环控制

凤凰山 github.com/gxtrobot/pyinaction

2019-11

Lesson4 - 条件判断和循环控制

课程内容

今天的课程将介绍 Python 的比较操作符和循环控制. 知道可以使用比较操作符产生一个 bool 值, 并且使用 bool 变量控制一个 while 循环语句块, 以及使用 for 循环对一个容器对象的所有元素进行操作.

课程目标

- 了解并使用 Python 的比较操作符
- 了解使用 while 循环控制语句
- 了解使用 for 循环控制语句
- 了解 range 函数的用法
- 编写一个小程序来接收并多个参数, 并使用循环来求和

比较操作符

比较大小的操作符

- >，<，==，!=，>=，<=
- 比较两个值或变量，并产生一个 bool 值
- 不支持的比较操作会报错，比如 int 和 str 比较

IDLE 玩一玩

在 IDLE 试着使用比较操作符，并查看结果

```
5  > 4
1 >= 1
1 > 1
1 != 1
0 == 0
True == True
True != False
'a' > 'b'
1 > 'a'
```

使用 While 循环语句

有限循环

```
a = 5
while a > 0:
    print(a)
    a = a - 1
```

- while 后接一个条件判断 bool 值表达式
- 当条件为真时，语句块就执行，完成一轮执行后回到条件判断位置
- 当条件为假时，循环结束，继续执行循环后语句
- while 语句第一行以':' 结束，后面跟缩进语句块
- 一般语句块中会对条件判断的变量的值进行修改，让它在某个时刻条件判断为假，然后退出循环
- 语句块中所有语句必须缩进对齐，所有对齐语句视为同一语句块，当语句取消缩进，语句块就结束了

使用 While 循环语句

无限循环

```
while True:
    print(1)
```

- 条件判断使用一个常数值，一般为 True
- 循环无限执行，一般需要人工结束
- 多用在编写一个后台程序，如 web 服务器程序，网络程序

IDLE 玩一玩

```
a = 10
while a > 0:
    print(a)
    a = a - 1

b = 0
while b < 10:
    print(b)
    b = b + 1


while True:
    print(1)
```

for 循环语句

```
a = [1,2,3,4]
for i in a:
    print(i)
```

- for ... in ... 循环遍历一个容器对象，对其所有元素进行操作
- for 后跟一个缩进语句块
- 使用一个变量可以引用容器对象里的每一个元素
- 所有元素遍历过一遍后自动退出循环

range 函数

- range 函数提供一个便利方式创建一个数字容器对象
- range(10), 产生 0-9, 共 10 个数字
- range(1,10), 产生 1-9 , 共 9 个数字
- range(2,10,2), 产生 2,4,6,8, 共 4 个数字

IDLE 玩一玩

```
a = [1,2,3,4]
for i in a:
    print(i)

for i in range(10):
    print(i)

for i in range(1,5):
    print(i)

for i in range(2,10,2):
    print(i)
```

for 和 while 比较

- for 一般用于对一个容器对象进行遍历操作
- while 用途更为广泛
- 可以将 for 循环转为 while 循环，但反之不行
- for 循环用起来更方便，且不易出错
- while 循环需要更精细的控制
- for 循环都为有限循环
- 一般优先考虑 for 循环，不行才采用 while 循环

IDLE 玩一玩

使用 for 和 while 循环完成相同任务

```
a = [1,2,3,4,5]
i = 0
while i < len(a):
    print(a[i])
    i = i + 1

for i in a:
    print(i)
```

break 和 continue 语句

这两个语句可用于 for 或 while 循环，用法一致

break 语句

- 提前退出循环，继续循环后语句执行
- 一般用于完成某任务时，提前结束循环，例如查询一个值，找到后退出

continue 语句

- 跳过当前这轮执行剩余语句，回到循环开始
- 一般用于需要跳过某个特殊值

IDLE 玩一玩

```
a = [1, 10, 2, 8 ,3]
b = 2
for i in a:
    print(i)
    if i == b:
        print('find b, quit now')
        break

for i in a:
    if i % 2 != 0:
        continue
    print(i)
```

求和小程序

将以下保存为 sum.py

```
import sys
print(sys.argv[1:])
nums = sys.argv[1:]
total = 0
for i in nums:
    total = total + int(i)
print('total=', total)
```

课后练习

- 修改 sum.py, 使用 while 循环进行求和计算，并思考那种方式更好
- 写个小程序打印出 1-100 的所有偶数，分别使用 for 和 while 编写

问题