

第五章 条件和循环

一、条件语句

1. Python之if语句

计算机之所以能做很多自动化的任务，因为它可以自己做条件判断。

输入用户年龄，根据年龄打印不同的年龄，在Python中用if语句实现

```
age = 20
if age >= 18:
    print 'your age is', age
    print 'adult'
print 'END'
```

注意：Python代码的缩进规则：具有相同缩进的代码被视为代码块，上面的3, 4行 print 语句就构成一个代码块（但不包括第5行的print）。如果 if 语句判断为 True，就会执行这个代码块。

缩进严格按照Python的习惯：4个空格，不要使用Tab，更不要使用混合Tab和空格，否则会因为缩进问题引起的语法错误

注意：if语句后接表达式，然后用：表示代码块开始

如果在Python的交互环境下敲代码，还要特别留意缩进，并且退出缩进要多敲一行回车：

```
age = 20
if age >= 18:
    print 'your age is', age
    print 'adult'

    #your age is 20
    #adult
```

2. Python之if-else语句

if...else语句:

```
if age >= 18:
    print 'adult'
else:
    print 'teenager'
```

利用 if ... else ... 语句，我们可以根据条件表达式的值为 True 或者 False，分别执行 if 代码块或者 else 代码块。

注意：else后面有个“:”

3. Python之if-elif-else语句

一个if...else不够用：需要嵌套if...else语句，但是缩进会很难看

所以引入if...elif...else语句:

```
if age >= 18:
    print 'adult'
elif age >= 6:
    print 'teenager'
elif age >= 3:
    print 'kid'
else:
    print 'baby'
```

elif就是else if

特别注意：这一系列的条件判断会从上到下依次判断，如果某个判断为true，执行完对应的代码块，后面的条件判断就直接忽略，不再执行了

二、循环语句

1. Python之for循环

list或tuple可以表示一个有序集合。如果我们想依次访问一个list中的每一个元素呢？比如 list：但是元素多了就比较麻烦并且不切实际

Python的for循环就可以依次把list或Tuple的每个元素迭代出来

```
L = ['Adam', 'Lisa', 'Bart']
for name in L:
    print name
```

注意：name这个变量是在for循环中定义的，意思是，依次取出list的每一个元素，并把元素赋值给name，然后执行for循环（就是缩进的代码块）

2. Python之while循环

while循环不会迭代list或Tuple中的元素，而是根据表达式判断循环是否结束

```
N = 10
x = 0
while x < N:
    print x
    x = x + 1
```

while循环每次先判断 $x \leq N$ ，如果为true，则执行循环的代码块，否则，退出循环

在循环体内， $x=x+1$ 会让x不断增加，最终因为 $x \leq N$ 不成立而退出循环

如果没有这一个语句，while循环在判断 $x \leq N$ 时总是为True，就会无限循环下去，变成死循环，所以要特别留意while循环的退出条件。

3. Python之break退出循环

用 for 循环或者 while 循环时，如果要在循环体内直接退出循环，可以使用break 语句。

计算1至100的整数和：

```
sum = 0
x = 1
while True:
    sum = sum + x
    x = x + 1
    if x > 100:
        break
print sum
```

注意：while True 就是一个死循环，但是在循环体内，我们还判断了 $x \leq 100$ 条件成立时，用break语句退出循环，这样也可以实现循环的结束。

练习：利用 while True 无限循环配合 break 语句，计算 $1 + 2 + 4 + 8 + 16 + \dots$ 的前20项的和。

```
sum = 0
x = 1
n = 1
while True:
    sum=sum+x
    x=x*2
    n=n+1
    if n>20:
        break
print sum
```

4. Python之continue继续循环

continue跳过后续循环代码，继续下一次循环

利用for循环计算平均分的代码

```
L = [75, 98, 59, 81, 66, 43, 69, 85]
sum = 0.0
n = 0
for x in L:
    sum = sum + x
    n = n + 1
print sum / n
```

只想统计及格分数的平均分，就要把 $x \geq 60$ 的分数剔除掉，这时，利用 continue，可以做到当 $x \geq 60$ 的时候，不继续执行循环体的后续代码，直接进入下一次循环：

```
for x in L:
    if x < 60:
        continue
    sum = sum + x
    n = n + 1
```

5. Python之多重循环

在循环内部，还可以嵌套循环，例如：

```
for x in ['A', 'B', 'C']:
    for y in ['1', '2', '3']:
        print x + y
```

练习：对100以内的两位数，请使用一个两重循环打印出所有十位数字比个位数字小的数，例如，23 ($2 < 3$)

```
for x in [1,2,3,4,5,6,7,8,9]:  
    for y in [1,2,3,4,5,6,7,8,9]:  
        if x<y:  
            print x*10+y
```