

3. 控制流

3.1 if判断语句

Python中的条件控制是通过条件语句的执行结果（True或者False）来决定执行的代码块。

Python中使用if来控制程序的执行，如果有多个条件的判断，可以使用if – elif – else的形式

In [1]:

```
a = input("socer: ")
a = float(a)
if 100 >= a >= 60:
    print("及格")
elif 0 <= a < 60:
    print("不及格")
else:
    print("分数错误")
```

及格

Python中的判断条件也可以为隐式的True False：在Python中0、None、空值默认为False，其他的都为True

In [2]:

```
if 0:
    print("F")
if 1:
    print("T")
if "python":
    print("python")
if None:
    print("None")
```

T
python

3.2 循环语句for

Python中的for语句区别于其他语言中的for循环，它接受可迭代对象（例如序列）作为其参数，每次迭代其中的一个元素。

In [3]:

```
for i in [1,2,3]:
    print("循环第%d次"%(i))
```

循环第1次
循环第2次
循环第3次

for 循环后面同样可以加入else，在循环结束后（正常结束）会执行。

In [5]:

```
for i in [1,2,3]:
    print("循环第%d次"%(i))
```

```
else:
    print("循环完毕")
```

循环第1次
循环第2次
循环第3次
循环完毕

嵌套循环

In [14]:

```
# 使用 for 循环完成乘法口诀
for i in range(1,10):
    for j in range(1,i+1):
        print("%dX%d=%-2d"%(j,i,j*i), end=" ")
    print()
```

```
1X1=1
1X2=2  2X2=4
1X3=3  2X3=6  3X3=9
1X4=4  2X4=8  3X4=12  4X4=16
1X5=5  2X5=10  3X5=15  4X5=20  5X5=25
1X6=6  2X6=12  3X6=18  4X6=24  5X6=30  6X6=36
1X7=7  2X7=14  3X7=21  4X7=28  5X7=35  6X7=42  7X7=49
1X8=8  2X8=16  3X8=24  4X8=32  5X8=40  6X8=48  7X8=56  8X8=64
1X9=9  2X9=18  3X9=27  4X9=36  5X9=45  6X9=54  7X9=63  8X9=72  9X9=81
```

使用嵌套循环时注意不宜过多的嵌套

3.3 while 循环

Python中 while 语句用于执行循环程序，在某条件下，循环执行某段程序，以处理需要重复处理的相同任务。

Python中没有do-while

In [7]:

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

5
4
3
2
1

while循环也可以添加else

In [8]:

```
i = 5
while i>0:
    print(i)
    i -= 1
else:
    print("循环结束")
```

5
4
3

```
.
2
1
循环结束
```

在使用while 循环时需要注意死循环和空循环。

```
In [ ]:
```

```
while True: # 死循环
    pass     # 空循环
```

3.3 break和continue

```
In [16]:
```

```
# 使用continue跳过当次循环
for i in range(10):
    if i % 2:
        continue
    print(i)
```

```
0
2
4
6
8
```

```
In [17]:
```

```
# 使用break跳出当前循环
for i in range(10):
    if i == 2:
        break
    print(i)
```

```
0
1
```

```
In [1]:
```

```
# 嵌套循环中的break
for i in range(3):
    for j in range(i):
        if i == 2:
            break
    print(i)
```

```
1
```

```
In [ ]:
```