

# if,for,while相关

## if语句

if 表达式:

语句块

如果表达式为真，那么则运行语句块，否则跳过

```
number = int(input("请输入您的QQ号: "))
if number == 182133566:
    print("恭喜中奖了!")
if number != 182133566:
    print("很遗憾没有中奖333")
```

```
In [1]: runfile('F:/笔记相关/Pj
二次分享/day06')
```

```
请输入您的QQ号: 182133566
恭喜中奖了!
```

```
In [2]: runfile('F:/笔记相关/Pj
二次分享/day06')
```

```
请输入您的QQ号: 3333
很遗憾没有中奖
```

```
In [3]:
```

使用if语句时，如果只有一条语句可以简写成这样： if 表达式:语句块

```
if number == 182133566:print("你也中奖了!")
```

## if..else语句

语法:

if 表达式:

语句块

else:

语句块

可以将上面的代码换一个写法:

```
number = int(input("请输入您的QQ号: "))
if number == 182133566:
    print("恭喜中奖了!")
else:
    print("很遗憾没有中奖")
```

也可以简化成下面的形式:

```
msg = "恭喜中奖了!" if number == 182133566 else "很遗憾没中奖"
print(msg)
```

## if..elif..else语句

```
age = int(input("请输入你的年龄: "))
if age < 18:
    print("你好啊同学")
elif age < 30:
    print("你好啊, 老铁")
else:
    print("兄弟你好啊56")
```

```
请输入你的年龄: 12
你好啊同学
```

```
In [8]: runfile('F:/笔记相关/Python基础第
二次分享/day06')
```

```
请输入你的年龄: 23
你好啊, 老铁
```

```
In [9]: runfile('F:/笔记相关/Python基础第
二次分享/day06')
```

```
请输入你的年龄: 56
兄弟你好啊
```

## if语句嵌套

可以将多个判断嵌套在一起, 但要注意不同级别的代码的缩进, 搞错了可能会出问题。所以不建议嵌套过多。

```
number = int(input("请输入去年的销量:"))

if number >= 1000:
    print("销量不错")
else:
    if number >= 500:
        print("销量还过得去")
    else:
        print("还需要努力啊")
```

```
请输入去年的销量:10000
销量不错
```

```
In [11]: runfile('F:/笔记相关/Python/
第二次分享/day06')
```

```
请输入去年的销量:600
销量还过得去
```

```
In [12]: runfile('F:/笔记相关/Python/
第二次分享/day06')
```

```
请输入去年的销量:200
还需要努力啊
```

## and , or , not连接

```
age = int(input("请输入你的年龄: "))
if age >=18 and age <=70:
    print("你可以考驾照了。")
```

等价于下面这张嵌套写法:

```
if age >= 18:
    if age <= 70:
        print("你可以考驾照了。")
```

or语句

```
# or
sales = int(input("请输入该商品的日销量: "))
if sales < 20 or sales > 100:
    print("该商品销售异常, 需要注意")
```

```
请输入该商品的日销量: 7
该商品销售异常, 需要注意
```

```
In [18]: runfile('F:/笔记相关/Python/
第二次分享/day06')
```

```
请输入你的年龄: 90
-----
```

```
请输入该商品的日销量: 3000
该商品销售异常, 需要注意
```

```
In [19]:
```

not 语句

```
a = input("请输入一位数的密码: ")
b = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']
if a not in b:
    print("非法输入, 请重新输入!")
```

请输入一位数的密码: a  
非法输入, 请重新输入!

In [20]:

## for循环

前面我们其实已经结果过了for循环, 比如变量列表:

```
# 遍历列表
for i in ['Python', 'C#', 'Java', 'VBA']:
    print(i)
```

```
Python
C#
Java
VBA
In [21]:
```

也可以遍历range()

计算1~100的和:

```
# 计算1~100的和
print("1~10相加的和:")
result = 0
for i in range(101):#等价于 range(1, 101),以为不包括右边的数, 所以是101, 而不是100
    result += i # 等价于 result = result + i
print(result) # 5050
```

## while循环

语法:

while 表达式:

循环体

循环体是一组被循环重复执行的语句

```
i = 1
print("三井寿: ")
while i <= 3:
    print("教练! 我想打篮球")
    i = i+1
```

```
三井寿：
教练！我想打篮球
教练！我想打篮球
教练！我想打篮球
```

注意表达式一定是一个可以终止的语句，否则会产生死循环(无限循环)

这里的`i = i + 1`就是这个功能。

## 循环嵌套

和if嵌套一样。大家可以试试。

同样不推荐嵌套太深。

## 跳转语句

- break语句

但执行到break语句时就跳出循环

```
flag = input("每次输入点东西，输入q时退出：")
while True: #一个死循环
    print(flag)
    flag = input("每次输入点东西，输入q时退出：")
    if flag == 'q':
        break #结束循环
```

```
每次输入点东西，输入q时退出： dassd
dassd

每次输入点东西，输入q时退出： dsadsa
dsadsa

每次输入点东西，输入q时退出： dasd
dasd

每次输入点东西，输入q时退出： q

In [27]:
```

- continue语句

跳出当前循环，继续下一个循环

演示逢7拍腿的游戏：就是所有人从1开始数，但数到7，或者包含7，或者是7的倍数时，则不说出这个数改为拍腿。

```
total = 99 #记录怕退的次数
mylist = [] #保存拍腿的数字
for number in range(1, 101): #从1数到100
    if number % 7 == 0: #7的倍数
        print("拍腿")
        mylist.append(number)
        continue
```

```

else:
    string = str(number) #转成字符串，是为了判断是否包含7
    if string.find('7') > -1:
        print("拍腿")
        mylist.append(number)
        continue
    else:
        print(number) #说出该数字
# 这个并不是每次都会执行到的。如果没有continue最终为-1，而
# 有continue的情况，但执行continue时，该运算不会被执行
total -= 1

print("从1数到100共拍腿{}次\n其中拍腿的数字有:\n{}".format(total,mylist))

```

```

100
从1数到100共拍腿29次
其中拍腿的数字有：
[7, 14, 17, 21, 27, 28, 35, 37, 42, 47, 49, 56, 57, 63, 67, 70, 71, 72, 73, 74, 75, 76, 77,
78, 79, 84, 87, 91, 97, 98]
In [38]:

```