

特工身份确认



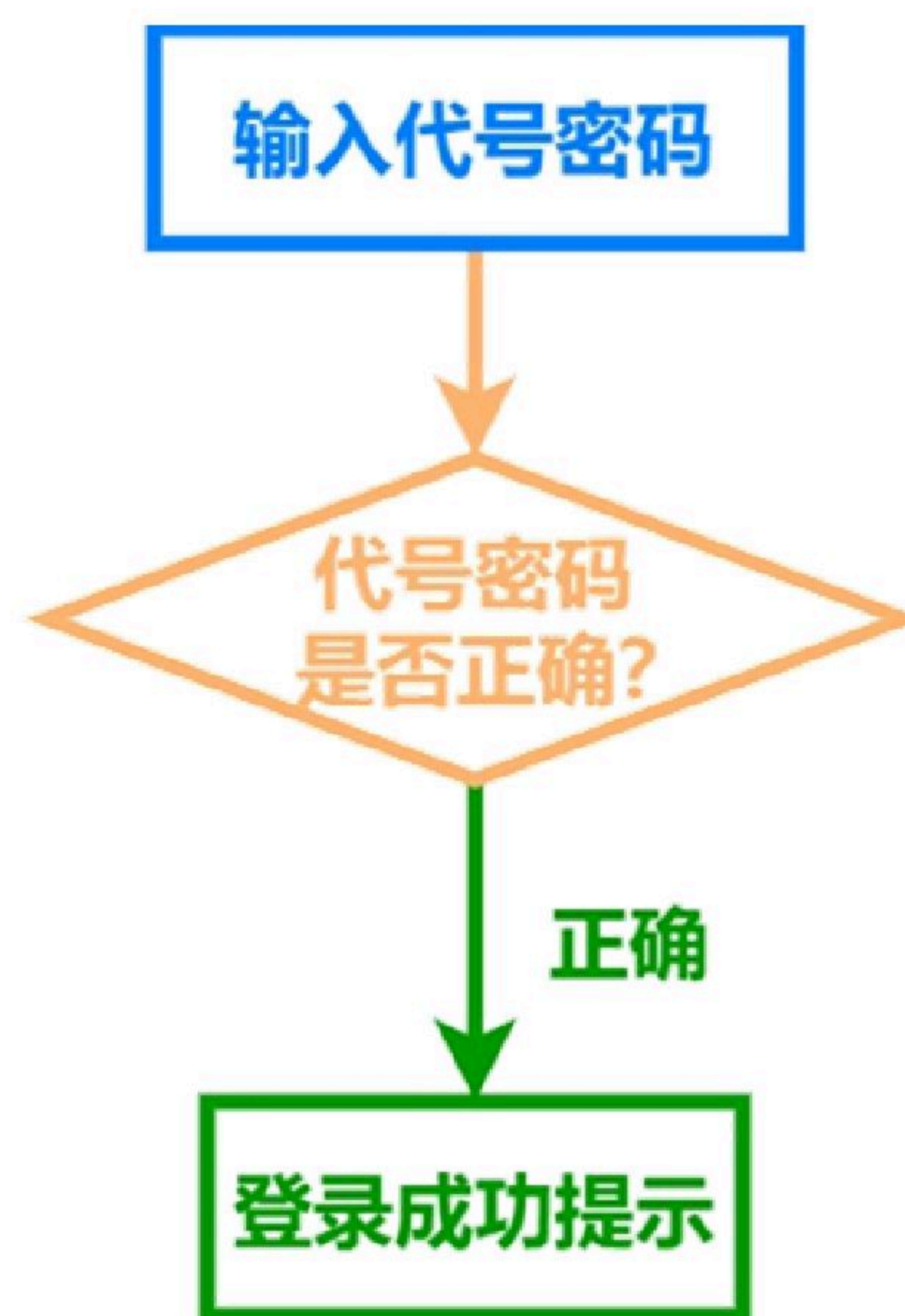


1. 探索新知

1.1

核对登录信息

```
username = input('请输入您的代号：')  
password = input('请输入您的密码：')  
if username == '007' and password == '妈咪妈咪哄':  
    print('登录成功！')
```



1.2

重复判断登录

实际操作时，可能会因为忘记或者失误输入错误，所以我们需要**不停**重新输入并判断是否正确



☹️ 想要不停做某件事情，这里需要用到循环结构，还记得我们学过的循环知识吗？

1.2

重复判断登录

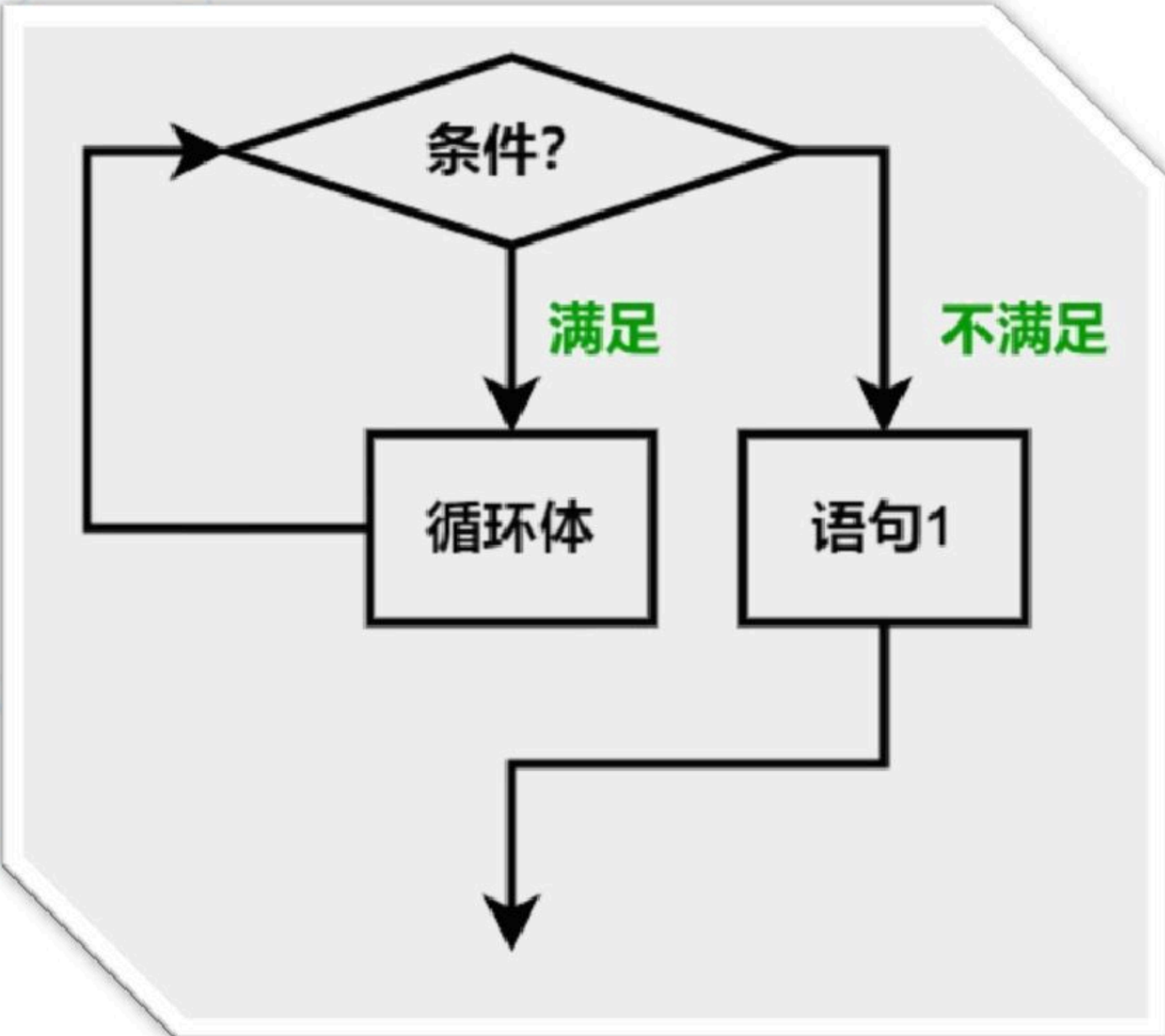
- 回顾：for循环

```
for i in range():  
    <循环体>
```



- 今天我们来学一种循环结构——while循环

```
while <条件? > :  
    <循环体>  
    <语句1>
```



想要不停执行循环体，也就是条件始终满足，该如何表示呢？

```
while True :  
    <循环体>  
    <语句1>
```

1.2

重复判断登录

while True:

```
username = input('请输入您的代号：')  
password = input('请输入您的密码：')  
if username == '007' and password == '妈咪妈咪哄':  
    print('登录成功！')
```

1.3

break跳出循环



那么问题来了！

运行程序发现，就算代号和密码正确，系统还在不停地提示输入，该怎么跳出循环呢？

break [breɪk]

打破；中断

while True:

if <条件1>:

break

<语句>

1.3

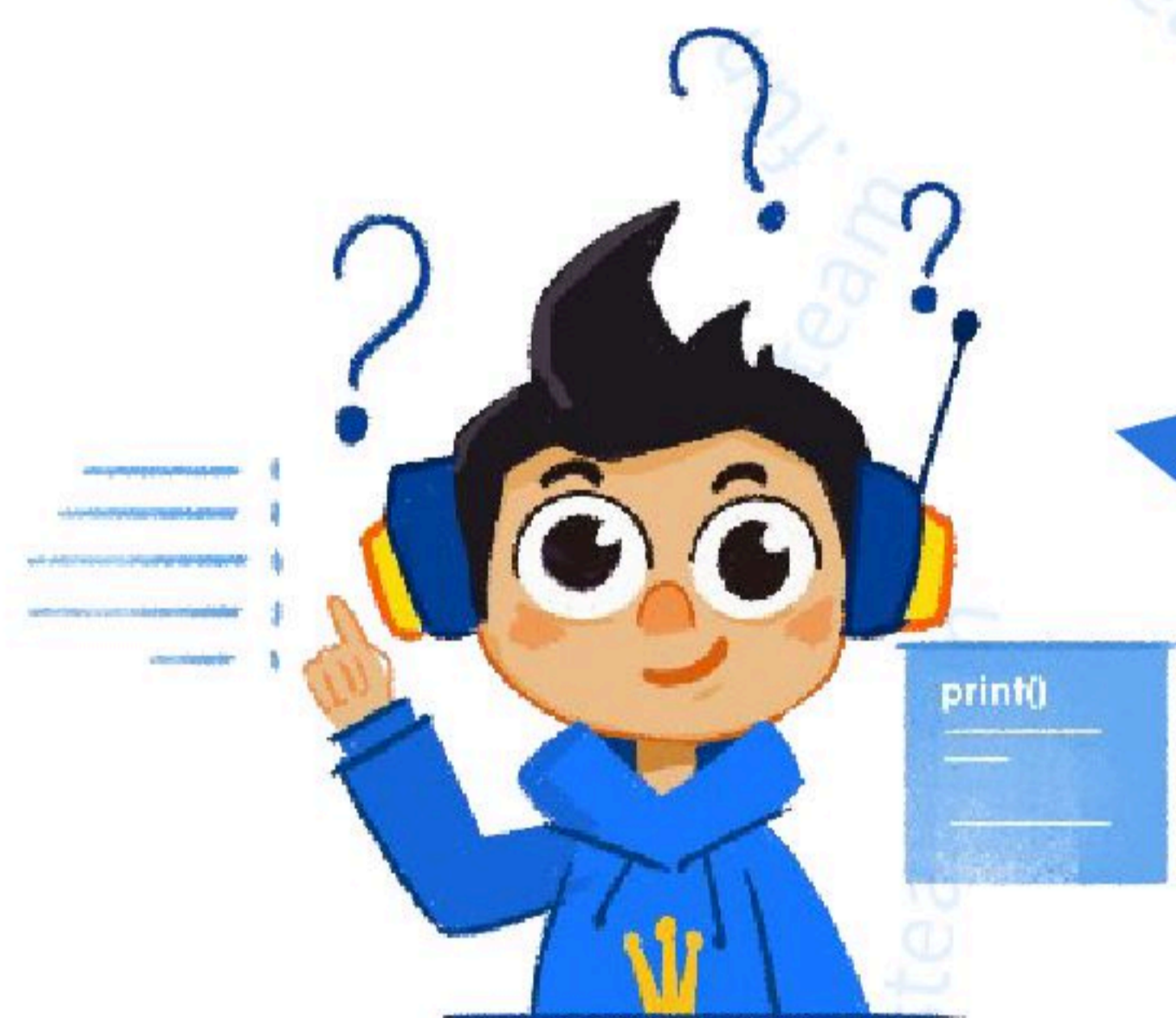
break跳出循环

```
while True:  
    username = input('请输入您的代号: ')  
    password = input('请输入您的密码: ')  
    if username == '007' and password == '妈咪妈咪哄':  
        print('登录成功!')  
        break  
    print('特工007, 欢迎回家')
```

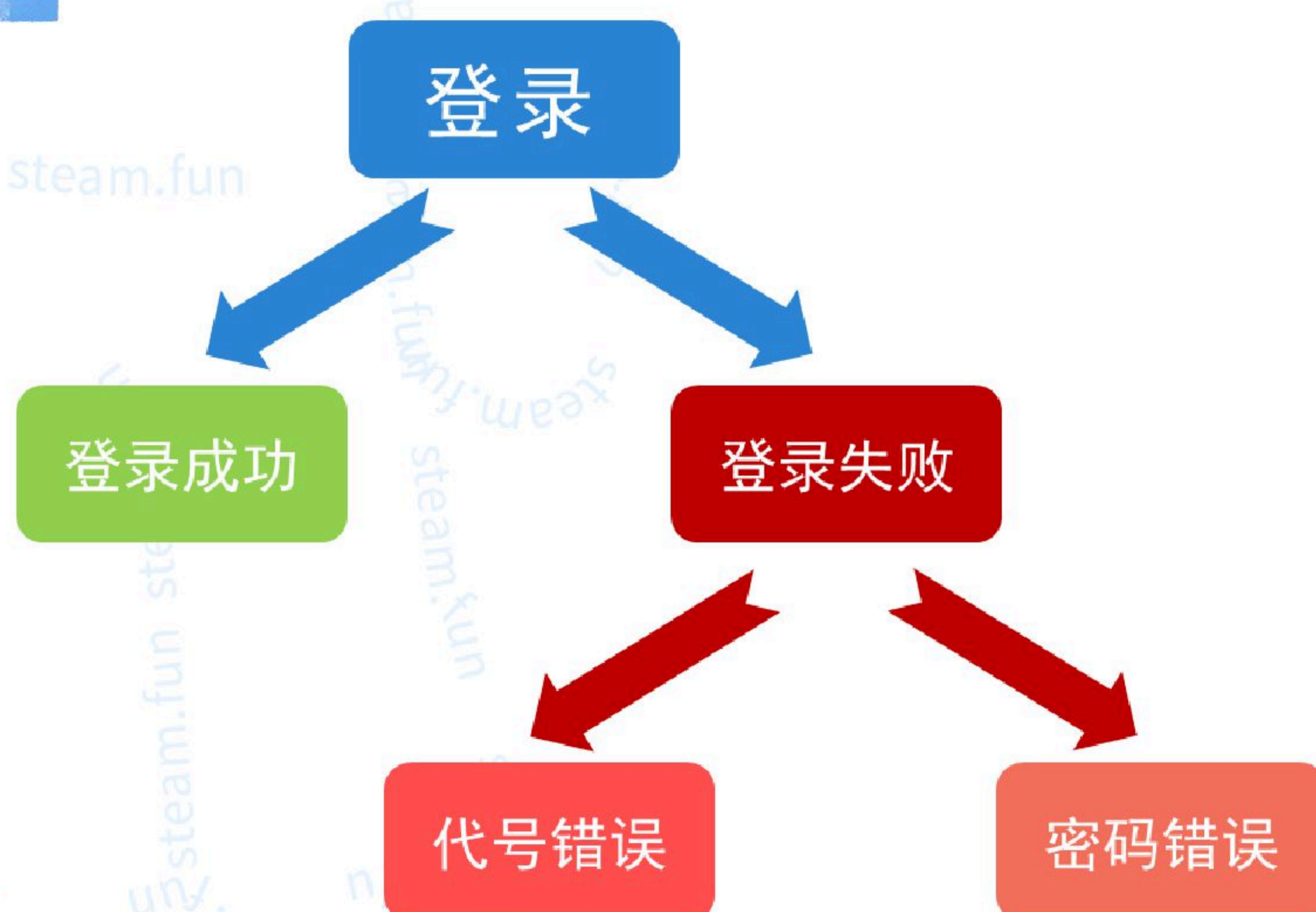
跳出
当前层循环

1.4

代号、密码错误



怎么表示输入的代号密码与正确的代号密码不相等呢?



1.4

代号、密码错误

文字

不等于

数学中

≠

python中

!=

```
if username != '007':  
    print('代号错误！请重新输入')  
if password != '妈咪妈咪哄':  
    print('密码记错了吧！请重新输入')
```

- 此时，程序是这样的

```
while True:  
    username = input('请输入您的代号：')  
    password = input('请输入您的密码：')  
    if username == '007' and password == '妈咪妈咪哄':  
        print('登录成功！')  
        break  
    if username != '007':  
        print('代号错误！请重新输入')  
    if password != '妈咪妈咪哄':  
        print('密码记错了吧！请重新输入')  
print('特工007，欢迎回家')
```



自己运行试试，看看有没有bug或者不合理的地方呢？

1.4

代号、密码错误

控制台

```

请输入您的代号: 008
请输入您的密码: 啦啦啦
代号错误! 请重新输入
密码记错了吧! 请重新输入
请输入您的代号:

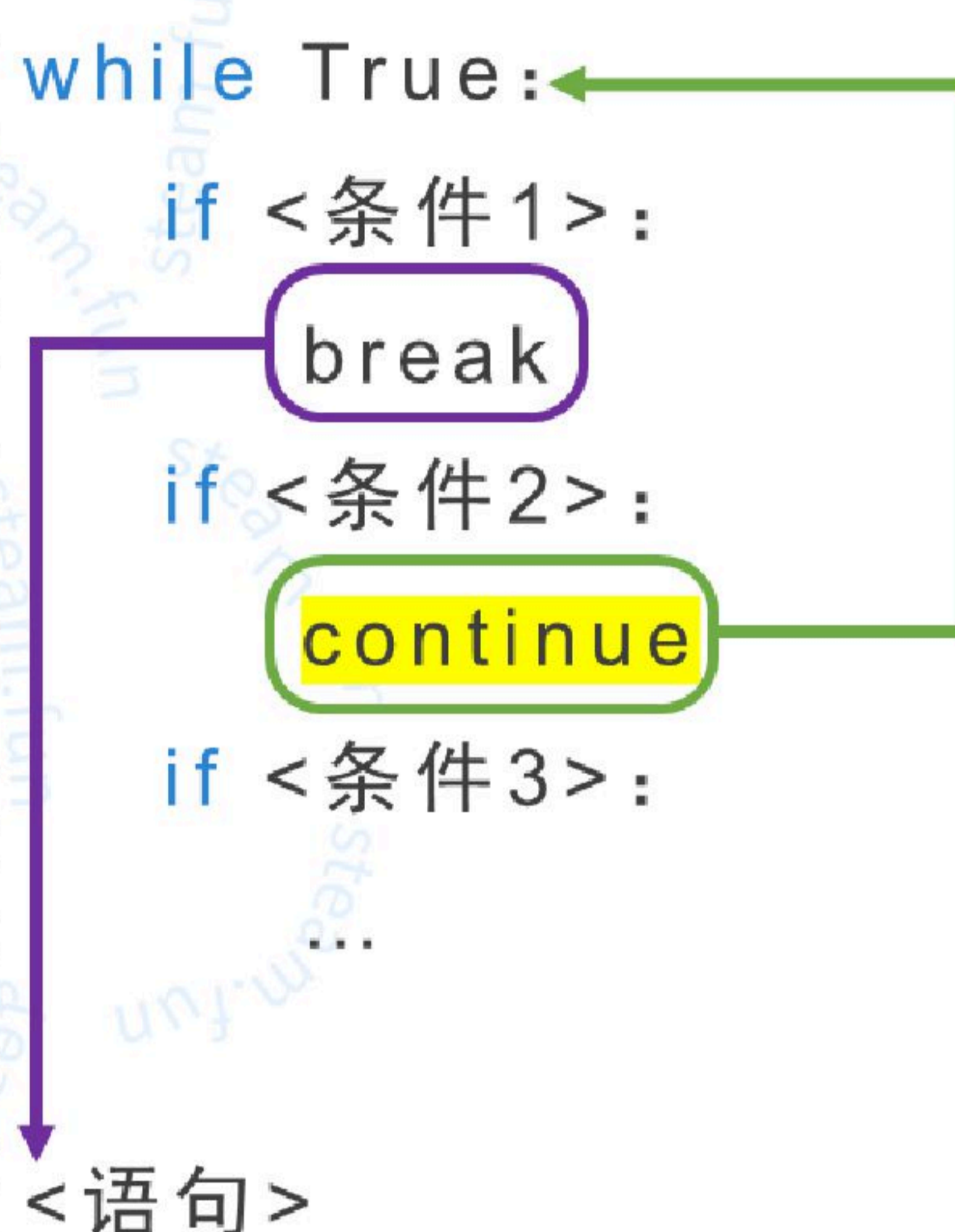
```

结合生活实际，如果代号输错了，就不需要判断密码是否正确。该怎么修正这个bug呢？



continue [kən'tɪnju:]

继续；持续；再开始



跳过当次该语句后的循环内容，进入下次循环

```

while True:
    username = input('请输入您的代号: ')
    password = input('请输入您的密码: ')
    if username == '007' and password == '妈咪妈咪哄':
        print('登录成功! ')
        break
    if username != '007':
        print('用户名错误! 请重新输入')
        continue
    if password != '妈咪妈咪哄':
        print('密码记错了吧! 请重新输入')
print('特工007, 欢迎回家')

```

完整代码

```
while True:
    username = input('请输入您的代号: ')
    password = input('请输入您的密码: ')

    # 检查用户名和密码是否正确
    if username == '007' and password == '妈咪妈咪哄':
        print('登录成功! ')
        break

    if username != '007':
        print('代号错误! 请重新输入')
        continue

    if password != '妈咪妈咪哄':
        print('密码记错了吧! 请重新输入')

print('特工007, 欢迎回家')
```

小彩蛋

break 和 continue 对比理解

```
i = 0
while i < 6:
    i = i + 1
    if i == 3:
        break
    print(i)
print("完成啦")
```

控制台：

```
1
2
完成啦
```

```
i = 0
while i < 6:
    i = i + 1
    if i == 3:
        continue
    print(i)
print("完成啦")
```

控制台：

```
1
2
4
5
6
完成啦
```



2. 强化练习

1. 下列说法错误的是？（ ）

- A. while语句通常用于重复执行某一段程序
- B. break语句可以跳出for和while的循环体
- C. 在Python中可以使用do...while循环
- D. if语句通常用于执行条件判断

2. 不能输出0~10之间所有奇数的程序是？（ ）

- A.

```
a = 0
while a < 10:
    if a % 2 == 0:
        a = a + 1
        continue
    print(a, end=' ')
    a = a + 1
```
- B.

```
for a in range(0, 11):
    if a % 2 != 1:
        continue
    print(a, end=' ')
```
- C.

```
for a in range(0, 11):
    if a % 2 == 0:
        continue
    print(a, end=' ')
```
- D.

```
a = 0
while a < 10:
    if a % 2 == 1:
        break
    a = a + 1
    print(a)
```

3. 以下代码运行后t的值是？（ ）

```
n, t = 10, 0
while n != 0:
    n = n - 1
    t = t + n
```

💡提示:

【n, t = 10, 0】

表示给n、t同时赋值，相当于n = 10, t = 0

- A. 36
- B. 45
- C. 50
- D. 55



2. 强化练习

4. 分别运行下列两段代码，输出的结果是？（ ）

A. 代码1: 2

代码2: 0

B. 代码1: 1

代码2: 0

C. 代码1: 1

代码2: 1

D. 代码1: 1

代码2: 2

```
# 代码1
```

```
n = 5
```

```
while n > 0:
```

```
    n = n - 1
```

```
    if n == 2:
```

```
        n = n - 1
```

```
        break
```

```
print("代码1: ", n)
```

```
# 代码2
```

```
n = 5
```

```
while n > 0:
```

```
    n = n - 1
```

```
    if n == 2:
```

```
        n = n - 1
```

```
        continue
```

```
print("代码2: ", n)
```

5. 关于 **break** 的作用，以下说法正确的是？（ ）

A. 按照缩进跳出当前层语句块

B. 暂停该语句后面的循环内容，跳转到循环开始位置开始下一轮循环

C. 跳出当前层for/while循环

D. 跳出所有for/while循环



3. 术语箱

while 当...的时候

username 用户名

break 打破, 中断

password 密码

continue 继续, 持续, 再开始

4. 课后挑战

猜猜我心里的数字

要求如下:

(1)确定一个目标数字 (1-100之间)

(2)询问并要求输入数字

(3)判断输入数字是否等于目标数字, 如果等于, 则跳出循环;

如果不等于, 则重新询问并要求输入

(3)找到数字跳出循环后, 说“终于找到你了!”

