

顺序结构

In [9]:

```
print('-----程序开始-----')
print('    1. 把冰箱门打开')
print('    2. 把大象放冰箱里')
print('    3. 把冰箱门关上')
print('-----程序结束-----')
```

```
-----程序开始-----
    1. 把冰箱门打开
    2. 把大象放冰箱里
    3. 把冰箱门关上
-----程序结束-----
```

选择结构

单分支结构

In [6]:

```
from random import choice, random

name = choice('董孙李周赵钱王')
if random() > 0.5:      # random() 函数返回[0,1)区间上的随机数
    '如果大于0.5 则三字人名，小于则连个字'
    name += choice('付玉延邵子凯')
name += choice('国楠栋涵雪玲瑞')
print(name)
```

孙凯栋

In [1]:

```
money = 1000      #余额
s = int(input('请输入取款金额')) #取款金额

#判断余额是否充足
if money >= s:
    money = money - s
    print("取款成功，余额：为：", money)
```

请输入取款金额500

取款成功，余额：为： 500

In [2]:

```
#取款大于最大值 if 下一行 句子不运行

money = 1000 #余额
s = int(input('请输入取款金额')) #取款金额

#判断余额是否充足
if money >= s:
    money = money - s
    print("取款成功，余额：为：", money)
```

请输入取款金额15000

双分支结构

In [3]:

```
#举例如果if能被2整除则是偶数，print输出偶数
#否则输出奇数，这样进行选择输出

num = int(input("请输入一个整数"))

#判断条件
if num%2 == 0:
    print(num, "是偶数")
else:
    print(num, "是奇数")
```

请输入一个整数5

5 是奇数

多分枝结构

In [5]:

```
for score in range(1,7):
    score = int(input('请输入一个成绩: '))
#判断
    if score >=90 and score <= 100:
        print('A级')
    elif score >=80 and score <= 89:
        print('B级')
    elif score >=70 and score <= 79:
        print('C级')
    elif score >=60 and score <= 69:
        print('D级')
    elif score >=0 and score <= 59:
        print('E级')
    else:
        print('对不起, 成绩有误, 不在成绩的有效范围')
```

请输入一个成绩: 58

E级

请输入一个成绩: 77

C级

请输入一个成绩: 69

D级

请输入一个成绩: 87

B级

请输入一个成绩: 91

A级

请输入一个成绩: -2

对不起, 成绩有误, 不在成绩的有效范围

注意缩进

In [8]:

#本语句中的因为缩进问题 所以只运行一次

```
score = int(input('请输入一个成绩: '))
#判断
if score >=90 and score <= 100:
    print('A级')
elif score >=80 and score <= 89:
    print('B级')
elif score >=70 and score <= 79:
    print('C级')
elif score >=60 and score <= 69:
    print('D级')
elif score >=0 and score <= 59:
    print('E级')
else:
    print('对不起, 成绩有误, 不在成绩的有效范围')
```

请输入一个成绩: 52

E级

嵌套

In []:

```

会员>=200 8折
    >=100 9折
    不打折
非会员 >=200 9.5折
    否则不打折

```

In [7]:

```

for answer in range(1,3):
    answer = input('您是会员吗? y/n') #y=yes n=no
    money = float(input('请输入你的购物金额'))
#外层判断是否为会员
    if answer == 'y': #是会员
        for money in range(1,4):
            if money >=200:
                print('打八折, 付款金额为: ', money*0.8)
            elif money >= 100:
                print('打九折, 付款金额为: ', money*0.9)
            else:
                print('不打折, 付款金额为: ', money)
    else: #不是会员
        for money in range(1,3):
            if money >=0:
                print('打九五折, 付款金额为: ', money*0.95)
            else:
                print('不打折, 付款金额为: ', money)

#因为 float函数要有数字但是这个eqe不是数字类型

```

```

您是会员吗? y/ny
请输入你的购物金额343414
不打折, 付款金额为: 1
不打折, 付款金额为: 2
不打折, 付款金额为: 3
您是会员吗? y/nn
请输入你的购物金额eqe

```

```

-----
ValueError                                Traceback (most recent call last)
<ipython-input-7-2acc79c1ce58> in <module>
      1 for answer in range(1,3):
      2     answer = input('您是会员吗? y/n') #y=yes n=no
----> 3     money = float(input('请输入你的购物金额'))
      4 #外层判断是否为会员
      5     if answer == 'y': #是会员

```

```

ValueError: could not convert string to float: 'eqe'

```

In [6]:

```
for answer in range(1,3):
    answer = input('您是会员吗? y/n') #y=yes n=no
    money = float(input('请输入你的购物金额'))
#外层判断是否为会员
    if answer == 'y': #是会员
        if money >=200:
            print('打八折, 付款金额为: ', money*0.8)
        elif money >= 100:
            print('打九折, 付款金额为: ', money*0.9)
        else:
            print('不打折, 付款金额为: ', money)
    else: #不是会员
        if money >=200:
            print('打九五折, 付款金额为: ', money*0.95)
        else:
            print('不打折, 付款金额为: ', money)
```

您是会员吗? y/ny
请输入你的购物金额232
打八折, 付款金额为: 185.60000000000002
您是会员吗? y/nn
请输入你的购物金额123213
打九五折, 付款金额为: 117052.34999999999

In [3]:

```
num_a = int(input('请输入一个整数'))
num_b = int(input('请输入一个整数'))
#比较大小
if num_a >= num_b:
    print(num_a, '大于等于', num_b)
else:
    print(num_a, '小于', num_b)
```

请输入一个整数10
请输入一个整数20
10 小于 20

条件表达式

if.....else简写

In []:

- 语法结构

```
x if 判断条件 else y
```

- 运算规则

如果判断条件的布尔值为True, 条件表达式的返回值为x, 否则条件表达式的返回值为False

In [12]:

```
num_a = int(input('请输入一个整数'))
num_b = int(input('请输入一个整数'))
print('使用条件表达式进行比较')
print((num_a, '大于等于', num_b) if num_a >= num_b else (num_a, '小于', num_b) )

# if num_a >= num_b 条件代码为True 执行左侧 为 False 执行右侧
print( str(num_a)+'大于等于'+str(num_b) if num_a >= num_b else str(num_a)+'小于'+str(num_b))

print( num_a, '大于等于', num_b if num_a >= num_b else num_a, '小于', num_b )

#最后一句因为没有括号 所以没有进行选择
```

请输入一个整数10
请输入一个整数2
使用条件表达式进行比较
(10, '大于等于', 2)
10大于等于2
10 大于等于 2 小于 2

pass 语句

大部分时候其实不推荐用,它表示不运行 减少报错 但大部分情况用不到,用到其实就是自己错了

In [1]:

```
answer = input('您是会员吗? y/n') #y=yes n=no
if answer == 'yes':
    pass
else:
    pass
#因为pass了 所以后面都不运行 获取了一个函数后不再运行
```

您是会员吗? y/ny

while 循环结构

In []:

选择结构的if与循环结构while的区别

- if是判断 '一次', 条件为 True 执行一行
- while是判断 'N+1次', 条件为 True 执行N次

In [17]:

```
a = 1
#判断条件执行体
if a < 10:

    #执行条件循环执行体
    print(a)
    a += 1    #执行一次
```

1

In [19]:

```
a = 1
#判断条件执行体
while a < 10:

    #执行条件循环执行体
    print(a)
    a += 1    #执行多次
```

1
2
3
4
5
6
7
8
9

In []:

`while`循环的执行流程

#四步循环法

- 初始化变量
- 条件判断
- 条件执行体(循环体)
- 改变变量

总结：初始化的变量与条件判断的变量与改变的变量为同一个

In [2]:

```

sum = 0 #用于累加和
''' 初始化变量为0'''
a = 0
''' 判断条件'''

while a < 5:

    ''' 条件执行体（循环体）'''
    sum += a

    ''' 改变变量'''
    a += 1
print('和为', sum)
''' 过程
a      a <5      sum      sum+=a
0      0<5 True   0        0+0
1      1<5 True   1        0+1
2      2<5 True   3        1+2
3      3<5 True   6        3+3
4      4<5 True   10       6+4
5      5=5 False  '''

```

和为 10

Out[2]:

```

' 过程\
na      a <5      sum      sum+=a\n
0      0<5 True   0        0+0\n
1      0+1\n      2<5 True   3        1+2\n
2      3<5 True   6        3+3\n
3      4<5 True   10       6+4\n
4      5=5 False  '

```

In [28]:

```

''' 计算1到100的偶数和'''
sum = 0 #用于累加偶数和
''' 初始化变量为0'''
a = 0

''' 判断条件'''
while a < 101:      #这里我们也可以这样写 a <=100

    ''' 条件执行体（循环体）'''
    if a % 2 == 0 :
        sum += a

    ''' 改变变量'''
    a += 1
print('和为', sum)

```

和为 2550

In [3]:

```
''' 计算1到100的偶数和'''
sum = 0 #用于累加偶数和

''' 初始化变量为0'''
a = 0

''' 判断条件'''
while a < 101:

    ''' 条件执行体'''
    #判断条件是否为
    if a % 2: #这里变成bool判断, 余数为1 为 Ture, 余数为0 为 False, Ture输出 导致成了奇数相加
        sum += a
    ''' 改变变量'''

    a += 1
print(' 和为', sum)
```

和为 2500

In [33]:

```
''' 计算1到100的偶数和'''
sum = 0 #用于累加偶数和
''' 初始化变量为0'''
a = 0

''' 判断条件'''
while a < 101:

    ''' 条件执行体'''
    #判断条件是否为
    if not bool(a % 2): #not bool 取反 变成了偶数和
        sum += a

    ''' 改变变量'''
    a += 1
print(' 和为', sum)
```

和为 2550

for-in循环

In []:

for-in循环

- in表达从 (字符串、序列等) 中依次取值, 又称为遍历
- for-in遍历的对象必须是可迭代对象

for-in的语法结构

- for 自定义的变量 in 可迭代对象:
 循环体

for-in的执行图

- 循环体内不需要访问自定义变量, 可以将自定义变量替代为下划线

In [37]:

```
for item in 'Python':#第一次取出来的是P ,将P赋值给item,将item的值输出
    print(item)

#range() 产生一个整数序列，--》也是一个可迭代对象
for i in range(10):
    print(i)

#如果在循环体中不需要使用到自定义变量，可将自定义变量写为“_”
for _ in range(5):
    print('人生苦短，我用python')
print('计算1到100的偶数和')
sum = 0 #用于储存偶数和

for item in range(1,101): # 这里的101不包含101 左闭右开
    if item % 2 ==0:
        sum += item
print('1到100之间的整数和为:',sum)
```

```
P
y
t
h
o
n
0
1
2
3
4
5
6
7
8
9
人生苦短，我用python
人生苦短，我用python
人生苦短，我用python
人生苦短，我用python
人生苦短，我用python
计算1到100的偶数和
1到100之间的整数和为: 2550
```

In [4]:

```
''' 输出100到999之间的水仙花数
举例:
153 = 3*3*3+5*5*5+1*1*1'''

for item in range(100,1000):
    ge = int(item %10)      #个位
    shi = int(item % 100/10) #十位
    bai = int(item/100)      #百位

    if ge*ge*ge + shi*shi*shi + bai*bai*bai == item:
        print(item)
```

153
370
371
407

In [10]:

```
''' folat 和 不转成int的都无法实现, 需要转换成整形'''

for item in range(100,1000):
    ge = float(item %10)      #个位
    shi = float(item % 100/10) #十位
    bai = float(item/100)      #百位
    if ge*ge*ge + shi*shi*shi + bai*bai*bai == item:
        print(item)

for item in range(100,1000):
    ge = item %10      #个位
    shi = item % 100/10 #十位
    bai = item/100      #百位
    if ge*ge*ge + shi*shi*shi + bai*bai*bai == item:
        print(item)
```

break语句

·用于结束循环结构,通常与分支结构if一起使用

for xxx in range(x) x表示次数

In [11]:

```
'''从键盘录入密码，最多录入3次，如果正确就结束循环'''
```

```
for item in range(3):  
    pwd = input('请输入密码: ')  
    if pwd == '8888':  
        print('密码正确:~')  
        break  
    else:  
        print("密码错误，请再次输入")
```

请输入密码: 97945
密码错误，请再次输入
请输入密码: 151545
密码错误，请再次输入
请输入密码: 115454
密码错误，请再次输入

for xxx in range(m,n+1) m表示初始次数， n+1表示结尾次数但不包括 n+1 所以只能遍历到n

In [3]:

```
'''从键盘录入密码，最多录入3次，如果正确就结束循环'''
```

```
for item in range(1,4):  
    pwd = input("请输入第"+str(item)+"次猜测的数字:")  
    if pwd == '8888':  
        print('密码正确:~')  
        break  
    else:  
        print("密码错误，请再次输入")  
    if item == 3:  
        print("很遗憾，三次机会已用尽，请 5 分钟后再尝试")
```

请输入第1次猜测的数字:545
密码错误，请再次输入
请输入第2次猜测的数字:116
密码错误，请再次输入
请输入第3次猜测的数字:21316
密码错误，请再次输入
很遗憾，三次机会已用尽，请 5 分钟后再尝试

continue语句

用于结束当前循环，进入下一次循环，通常与分支结构中的if连用

In [11]:

```
''' 要求输出1到50之间所有5的倍数，5, 10, 15, 20, 25... ..
    5的倍数的共同点：一和5的余数为0的数都是5的倍数
    什么样的数不是5的倍数，-1, 2, 3, 4, 6, 7, 8, 9... 与5的余数不为0的数都不是5的倍数
'''

print('----不使用continue 的 == ----')
for item in range (1,51):
    if item % 5 == 0:
        print(item)    #输出的是5的整数

print('----不使用continue 的 != ----')
for item in range (1,51):
    if item % 5 != 0:
        print(item)    #输出的是非5的整数

print('-----使用continue 的 == -----')
for item in range (1,51):
    if item % 5 == 0:
        continue
    print(item)        #输出的非是5的整数，因为continue 跳出了这个循环

print('-----使用continue 的 != -----')
for item in range (1,51):
    if item % 5 != 0:
        continue
    print(item)        #输出的是五的整数
```

----不使用continue 的 == ----

5
10
15
20
25
30
35
40
45
50

----不使用continue 的 != ----

1
2
3
4
6
7
8
9
11
12
13
14
16
17
18
19
21
22
23
24

26
27
28
29
31
32
33
34
36
37
38
39
41
42
43
44
46
47
48
49

-----使用continue 的 == -----

1
2
3
4
6
7
8
9
11
12
13
14
16
17
18
19
21
22
23
24
26
27
28
29
31
32
33
34
36
37
38
39
41
42
43
44
46
47
48
49

-----使用continue 的 != -----

5
10
15
20
25
30
35
40
45
50

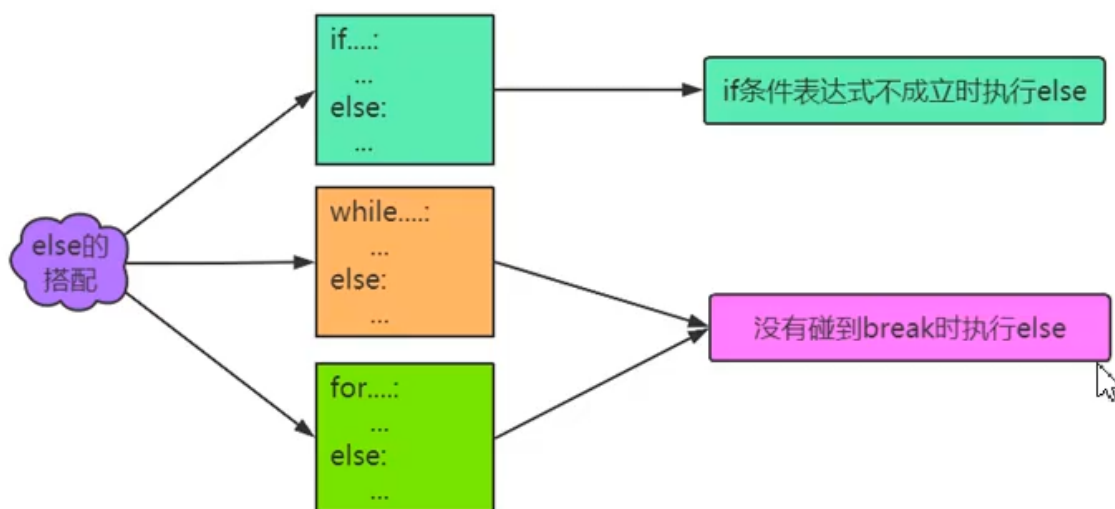
In [15]:

```
for item in range (1,51):  
    if item % 5 != 0:  
        continue  
    print(item)
```

#因为格式问题 print 无法输出 ‘ ’ 缩进 ‘ ’

else 语句

- else语句
 - 与else语句配合使用的三种情况



In [16]:

```
'''从键盘录入密码，最多录入3次，如果正确就结束循环'''  
for item in range(1,4):  
    pwd = input("请输入第"+str(item)+"次猜测的数字:")  
    if pwd == '8888':  
        print('密码正确:')  
        break  
    else:  
        print("密码错误，请再次输入")  
else:  
    #这里的else 是对for  
    print("很遗憾，三次机会已用尽，请 5 分钟后再尝试")
```

请输入第1次猜测的数字:

密码错误，请再次输入

请输入第2次猜测的数字:

密码错误，请再次输入

请输入第3次猜测的数字:33

密码错误，请再次输入

很遗憾，三次机会已用尽，请 5 分钟后再尝试

In [8]:

```
a = 0  
while a < 3:  
    pwd = input('请输入密码')  
    if pwd == '8888':  
        print('密码正确')  
        break  
    else:  
        print('输出错误')  
        '''改变变量'''  
        a += 1  
else:  
    print('对不起，三次密码均输入错误')
```

请输入密码64

输出错误

请输入密码6199

输出错误

请输入密码6149

输出错误

对不起，三次密码均输入错误

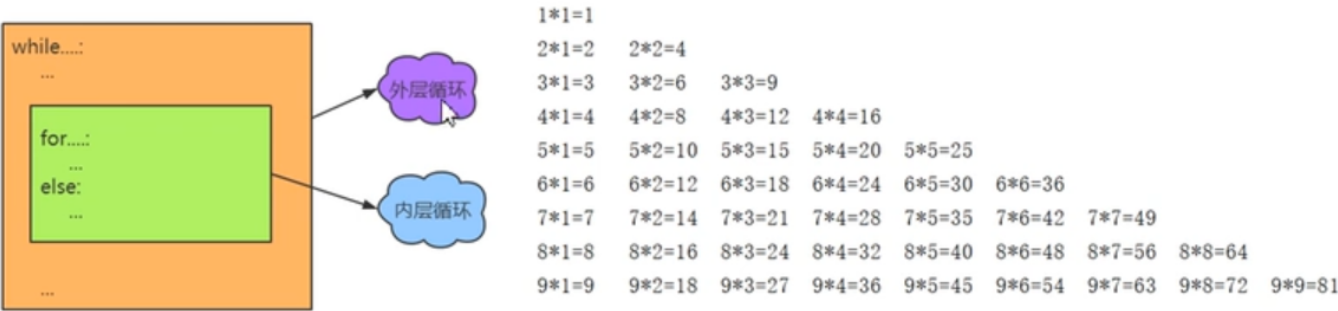
In []:

```
a = 0
while a < 3:
    pwd = input(' 请输入密码')
    if pwd == '8888':
        print(' 密码正确')
        break
    else:
        print(' 输出错误')
#上面都是条件执行体
else:
    #没有条件改变量
    print(' 对不起，三次密码均输入错误')
```

请输入密码4
输出错误
请输入密码5
输出错误
请输入密码6
输出错误
请输入密码4
输出错误

嵌套

- 嵌套循环
 - 循环结构中又嵌套了另外的完整的循环结构，其中内层循环做为外层循环的循环体执行



一开始我们很难实现这个，我们可尝试慢慢来

In [2]:

```
''' 输出三行四列的矩阵'''

for i in range(1,4):          #行表，执行三次，一次一行 i, j
    for j in range(1,5):
        print('*',end ='\t' ) #不换行输出    |t = tab键 四个空格
    print()#打行

''' 流程过程
    先 i=1 j=1 输出* 再 i=1 j=2 输出*; i=1 j=3 输出* ; i=1 j=4 输出*
    换行 再 i=2 j=1    ...
'''

for i in range(1,4):          #行表，执行三次，一次一行 i, j
    for j in range(1,5):
        print('*',end ='') ) #不换行输出
    print()#打行

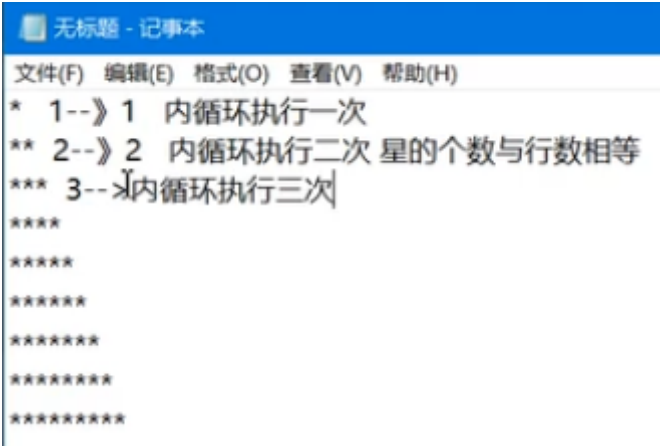
for i in range(1,4):          #行表，执行三次，一次一行 i, j
    for j in range(1,5):
        print('*','\t' )      #换行输出 这里 \t 有无无关
    print()
```

```
*      *      *      *
*      *      *      *
*      *      *      *
****
****
****
*
*
*
*

*
*
*
*

*
*
*
*
.
```

这样的过程



In [15]:

```
"""先尝试三角行的"""

for i in range(1,10):
    for j in range(1,i + 1):
        print('*',end='')
    print()
```

```
*
**
***
****
*****
*****
*****
*****
*****
*****
```

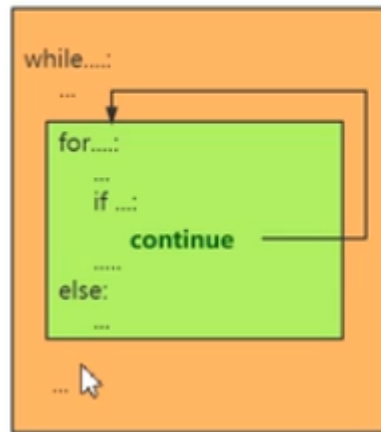
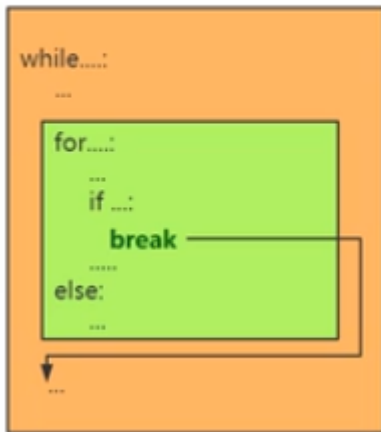
In [3]:

```
''' 然后实现'''
for i in range(1,10):
    for j in range(1,i + 1):
        print(i,'*',j,'=',i*j,end='\t')
    print()
```

```
1 * 1 = 1
2 * 1 = 2      2 * 2 = 4
3 * 1 = 3      3 * 2 = 6      3 * 3 = 9
4 * 1 = 4      4 * 2 = 8      4 * 3 = 12      4 * 4 = 16
5 * 1 = 5      5 * 2 = 10     5 * 3 = 15     5 * 4 = 20     5 * 5 = 25
6 * 1 = 6      6 * 2 = 12     6 * 3 = 18     6 * 4 = 24     6 * 5 = 30     6 *
6 = 36
7 * 1 = 7      7 * 2 = 14     7 * 3 = 21     7 * 4 = 28     7 * 5 = 35     7 *
6 = 42  7 * 7 = 49
8 * 1 = 8      8 * 2 = 16     8 * 3 = 24     8 * 4 = 32     8 * 5 = 40     8 *
6 = 48  8 * 7 = 56      8 * 8 = 64
9 * 1 = 9      9 * 2 = 18     9 * 3 = 27     9 * 4 = 36     9 * 5 = 45     9 *
6 = 54  9 * 7 = 63      9 * 8 = 72      9 * 9 = 81
```

·二重循环break和continue

• 二重循环中的break和continue用于控制本层循环



二重循环break

In [35]:

```
for i in range(5):
    for j in range(1,11):
        if j%2 == 0:
            break
        print(j)
''' i=0, j=1, 1%2!=0, 输出1   i=0, j=2, 2%2=0 break
    i=1, j=1, 1%2!=0, 输出1   i=1, j=2, 2%2=0 break
; ; ;
    #这里的break 只影响内层循环，不影响外层
    i=4, j=1, 1%2!=0, 输出1   i=4, j=2, 2%2=0 break
,,,
```

1
1
1
1
1
1

Out[35]:

```
' i=0, j=1, 1%2!=0, 输出1   i=0, j=2, 2%2=0 break\n    i=1, j=1, 1%2!=0, 输出1   i=1, j=2, 2%2=0 break\n; ; ; \n    #这里的break 只影响内层循环，不影响外层\n    i=4, j=1, 1%2!=0, 输出1   i=4, j=2, 2%2=0 break\n'
```

In [36]:

```
for i in range(5):
    for j in range(1,11):
        if j%2 == 0:
            break
        print(j)
    print()      # 这里输出空值达到空一行目的
```

1

1

1

1

1

In [4]:

```
for i in range(5):
    for j in range(1,11):
        if j%2 == 0:
            break
        print(j)  #这里内层没有值  因为是if的print if没值
```

二重循环continue

In [30]:

```
for i in range(5):
    for j in range(1,11):
        if j%2 == 0:
            continue
        print(j, end=' \t')  #这里内层没有值
    print()
```

1	3	5	7	9
1	3	5	7	9
1	3	5	7	9
1	3	5	7	9
1	3	5	7	9

In [37]:

```
for i in range(5):
    for j in range(1,11):
        if j%2 == 0:
            continue
        print(j, end=' \t')
#没print ()  所以不输出空行
```

1	3	5	7	9	1	3	5	7	9	1
3	5	7	9	1	3	5	7	9	1	3
5	7	9								

知识点总结

