

# 循环语句

■ ■ ■ CONTROL STATEMENTS  
3 ■ 控制语句



## 三天打鱼





## 二天晒网







## 3天打鱼





2天晒网

循环







## 2天晒网

## 循环

```
While(没有退休) {  
    3天打鱼 ;  
    2天晒网 ;  
}
```

## 循环语句

循环结构是程序设计不可缺少的一部分。

包括三种循环语句：

**for,**

**while,**

**do while**

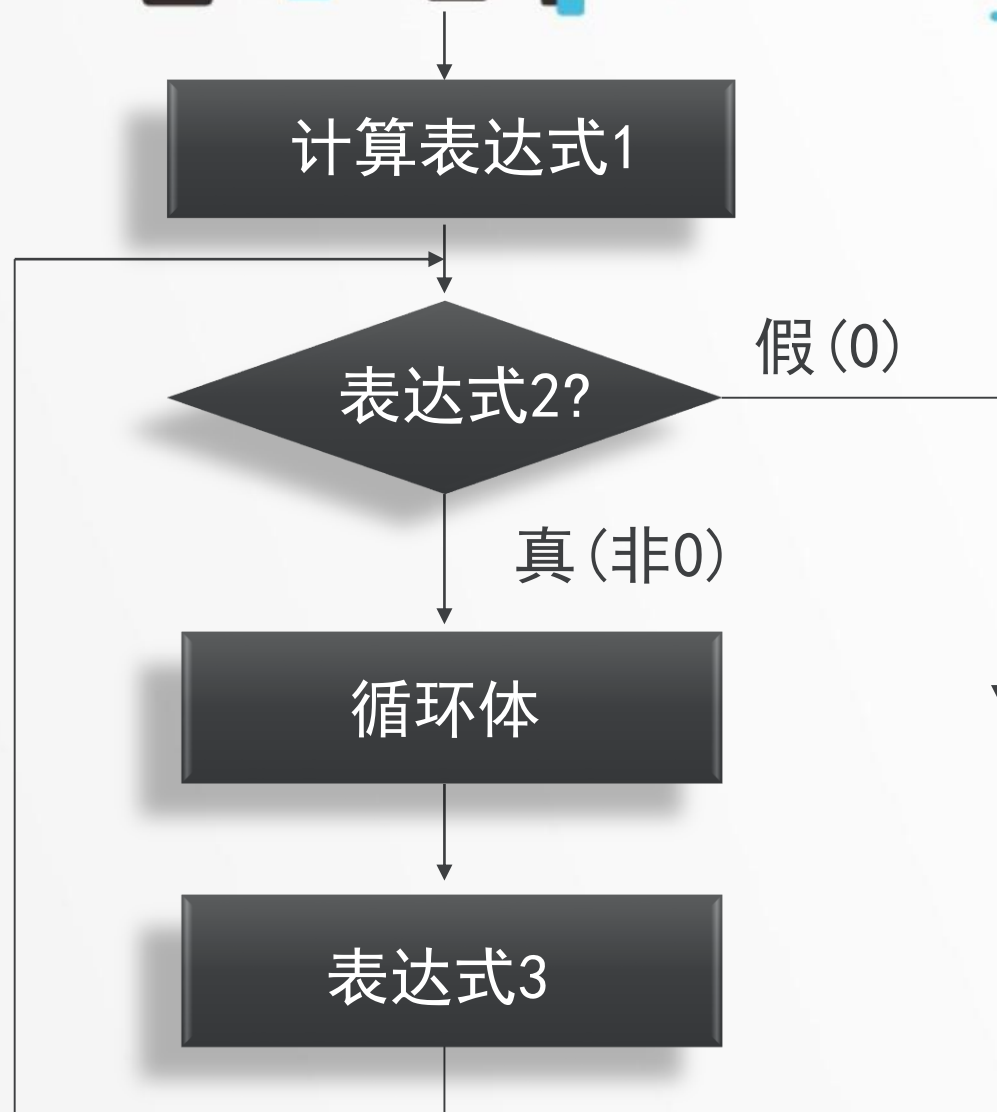


## for 循环语句

for循环的一般形式是：

**for(表达式1;表达式2;表达式3)**  
**循环体;**

语句流程如图所示. →



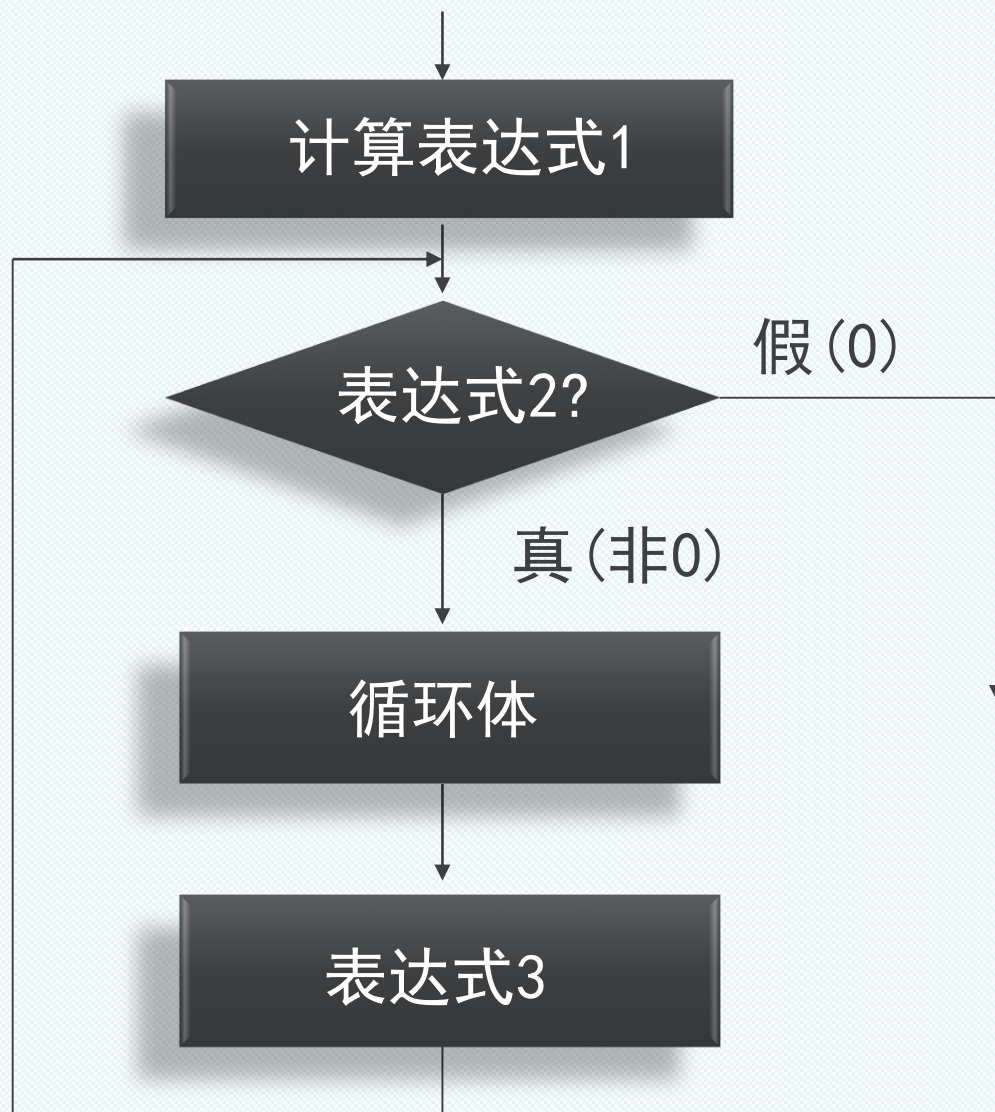


## for 循环语句

for循环的一般形式是：

**for(表达式1;表达式2;表达式3)**  
**循环体;**

语句流程如图所示. →



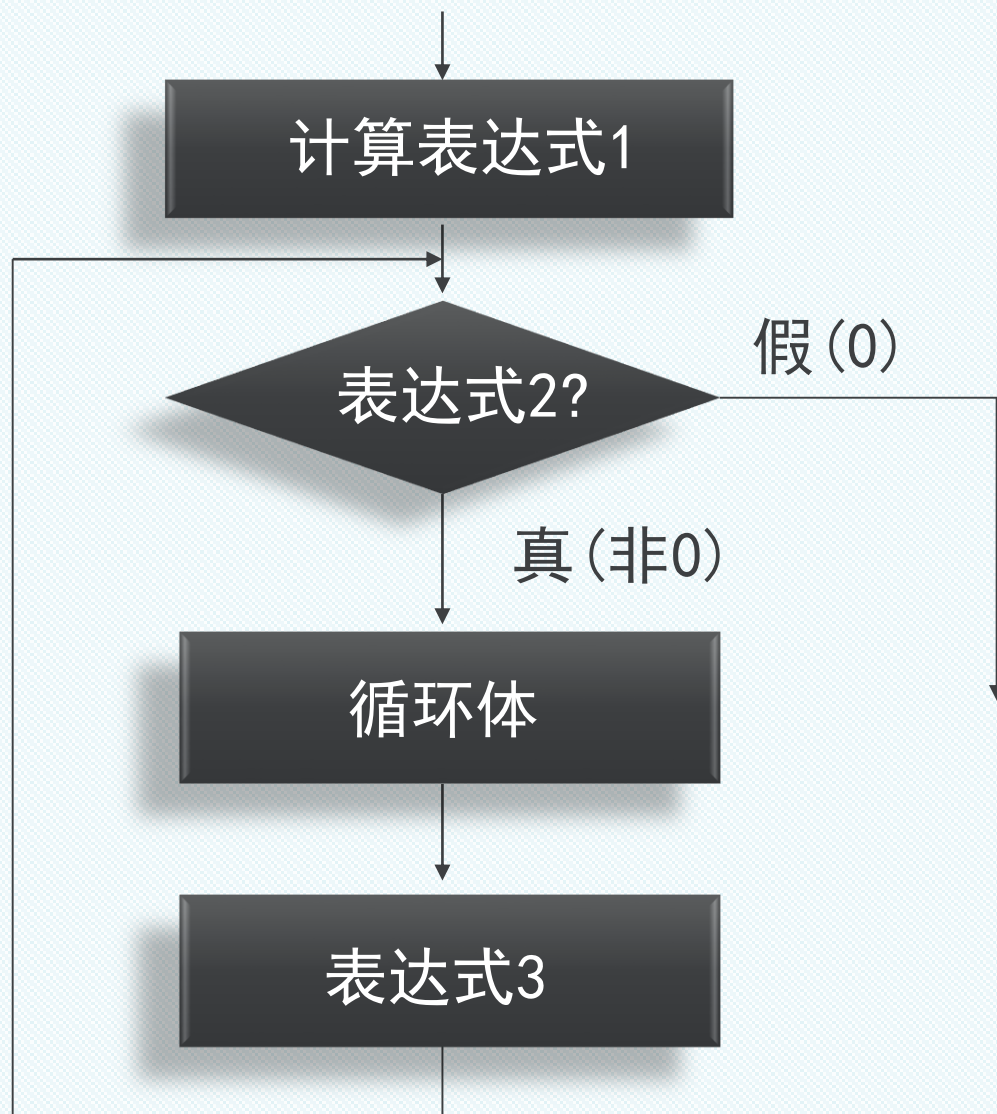
## for 循环语句

for循环的一般形式是：

**for(表达式1;表达式2;表达式3)**

**循环体;**

语句流程如图所示. →



## for 循环语句

## 例子：求n的1-5次方

输入整数n，然后输出n的1-5次方(空格区分)

例如输入-1，输出-1 1 -1 1 -1

### 算法描述：

输入：整数n

输出：n的1次方，2次方,...,5次方

算法思想：用i表示第i次方，从1递增到5；

$result = result * n$ ; 则每乘积1次n，则求出 $n^i$ 结果





## for 循环语句

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int n;
    long result=1;
    scanf_s("%d", &n);
    for (int i = 1; i <= 5; ++i)
    {
        result *= n;
        printf("%ld ", result);
    }
    printf("\n");
    system("pause");
    return 0;
}
```



## 注意: 使用for语句的灵活性

### 1 使用逗号运算符

```
for(i=0,j=100;(j-i)>10;i++,j-=5)
{ 循环体 }
```

### 2

### 允许在循环体内改变循环控制变量的值

```
sum=0;
for(count=1;count<=100;count++)
{
    scanf(" %d" ,&intnumber);
    sum+=intnumber;
    if(sum>=3000)
        count=100;
}
```



## for 循环语句

3

### 三个表达式均可缺省

(1) `for(i=0;i!=100;)  
    scanf(" %d" ,&i);`

省略表达式3

(2) `i=(a+b)/2  
for(; i>20;i++)  
    { 循环体 }`

省略表达式1

(3) `for(;;) /*死循环*/  
    { 循环体 }`

省略所有表达式

(4) `for(i=1,j=1;i<1000;i++,j*=1);`

省略循环体

`for(i=1,j=1;i<1000;i++)j*=1;`

没有省略循环体的等价表示

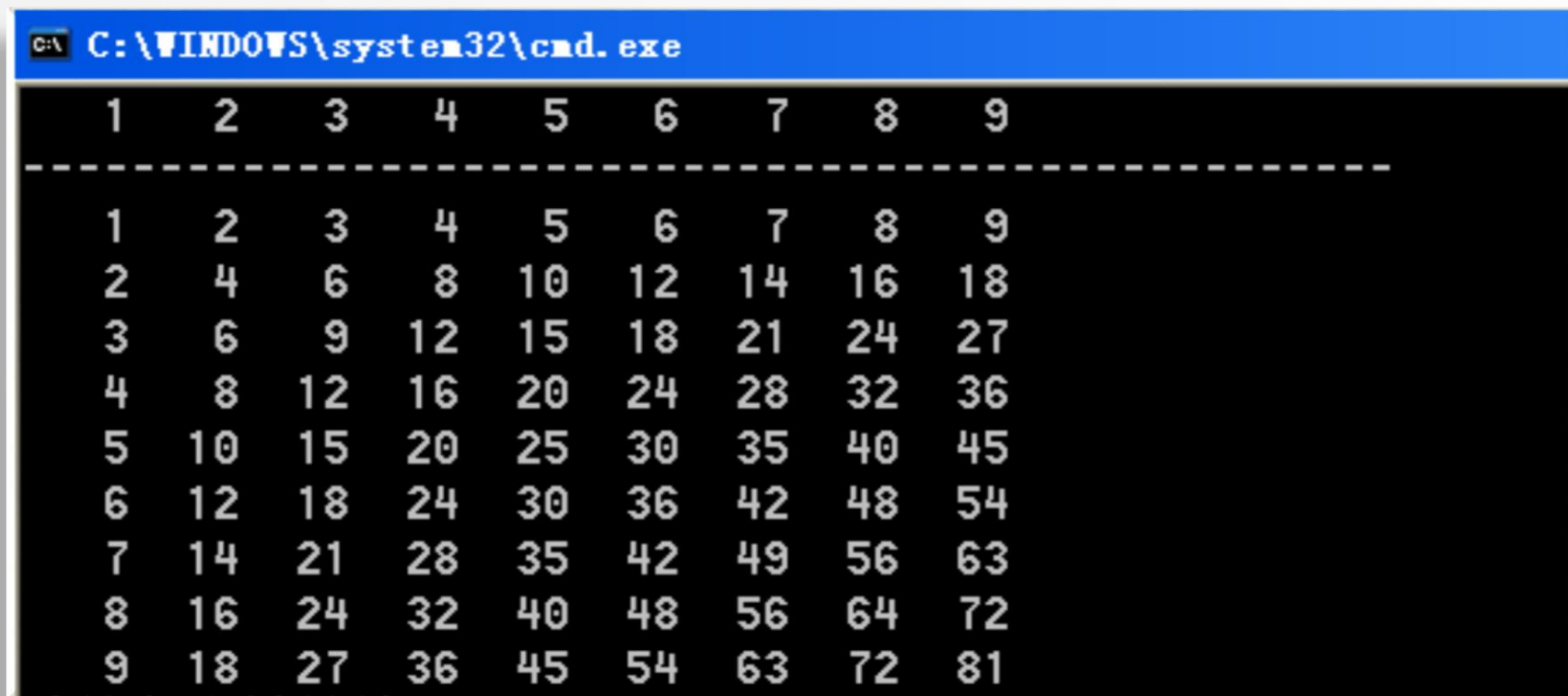




## for 循环语句

例：输出一个九九表

输出情况如图所示



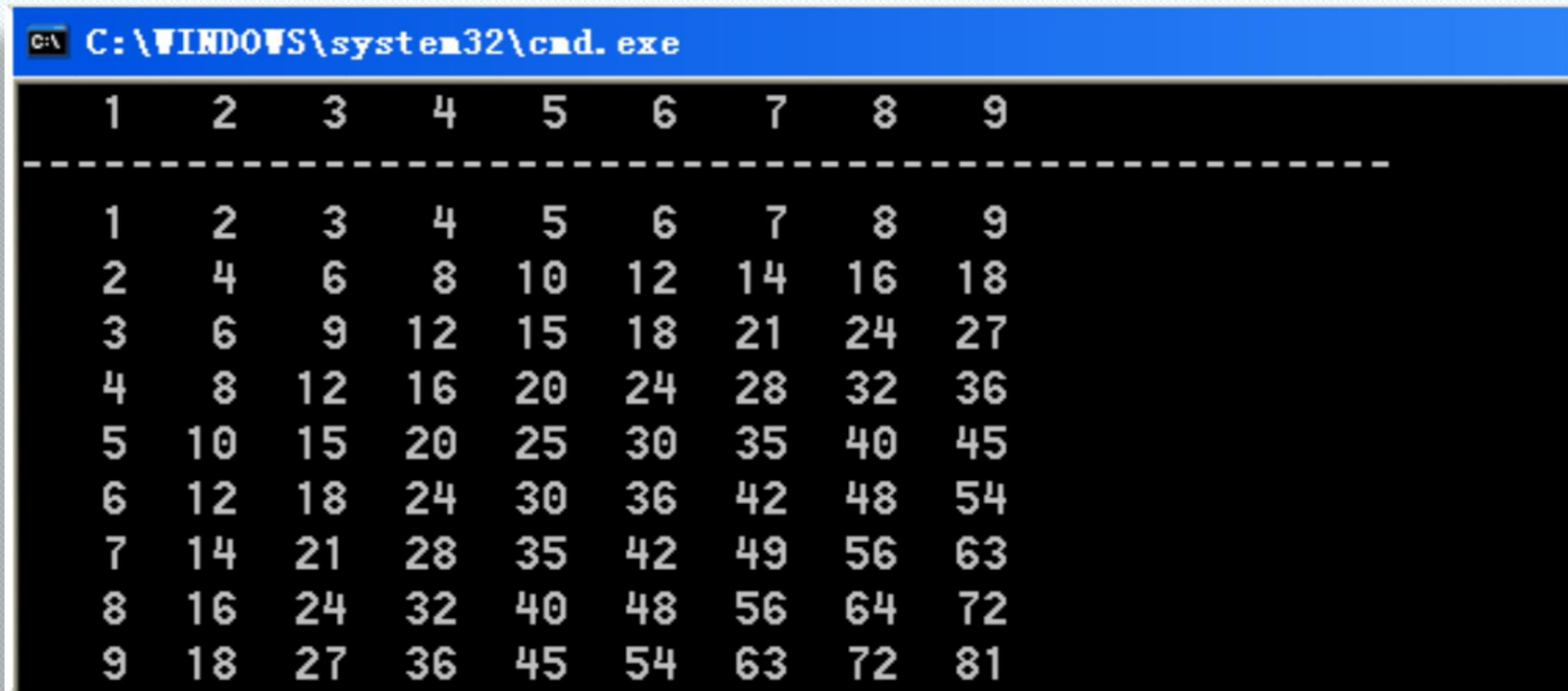
A screenshot of a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The window displays a 9x9 multiplication table (九九表) with numbers 1 through 9 in both rows and columns. The output is as follows:

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

分析：(1) 第一行是1到9递增输出

(2) 然后输出分割线-----

(3) 从第三行开始：用变量i从1到9递增，控制行的变化  
变量j从1到9递增，控制列的变化  
输出的结果是i\*j的值



The screenshot shows a Windows command prompt window with the title bar "C:\WINDOWS\system32\cmd.exe". The window displays a 9x9 multiplication table. The first row contains numbers 1 through 9. A dashed line separates the header from the body of the table. The body of the table contains the products of the row and column indices, ranging from 1 to 81.

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



```
① main( )
② {
③     int i,j;
④     for(i=1;i<10;i++)
⑤         printf(" %4d" ,i);
⑥     printf(" \n-----\n" );
⑦     for(i=1;i<10;i++)//控制行
⑧         for(j=1;j<10;j++)//控制列
⑨             if(j!=9)
⑩                 printf( "%4d" ,i*j);
⑪             else
⑫                 printf( "%4d\n" ,i*j);

⑬ }
```



## while和do while语句

for语句通常可以预知循环次数;若不能,则可使用while 或do while语句.



## while语句

一般形式:

```
while(表达式)  
{  
    循环体;  
}
```

## do while语句

后判定循环控制条件,至少需要执行一次循环体.

一般形式为:

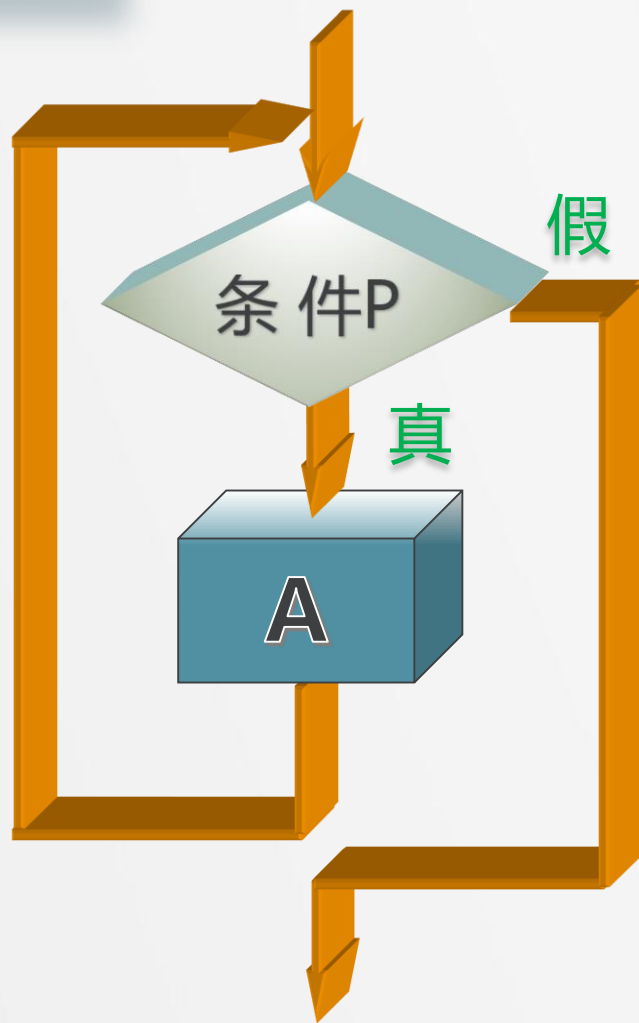
```
do{  
    循环体;  
}while(表达式);
```



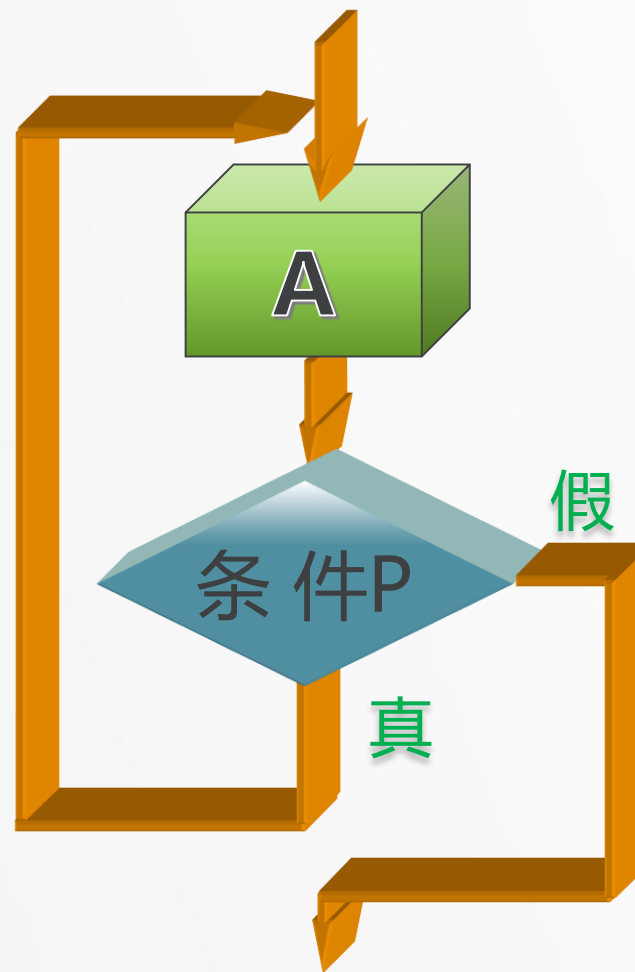
## 循环的类型

当型循环和直到型循环

当型循环



直到型循环





例: 读入字符并回显,直到读入 '\*' 字符为止的程序段.

```
① char c;  
② c=getchar( );  
③ while(c!= '*' )  
④ {  
⑤   putchar( c );  
⑥   c=getchar( );  
⑦ }
```



简写为:

```
① char c;  
② while((c=getchar( ))!= '*' )  
③   putchar( c );
```

假设, 输入:

aBCde?f\*

则, 输出:

a B C d e ? f



## 循环结构小结

## 循环 语句

功 能：

只要判断条件为真，就一直做循环语句，直到条件为假才退出循环。

三种结构：

for、while、do while