

## 一.while和do while

### 一,是什么?

是指编程循环语句,当满足条件时进入循环,循环判断,直到不满足条件跳出循环。

### 二,有什么用?

可以根据条件反复执行一段代码

### 三,怎么用?

`while(<条件>) {<语句>;    do {<语句>} while(<条件>;`

### 四,什么时候用?

当需要重复执行一段代码或满足一定条件重复执行的一短代码实现相应的功能.

### 五,有什么特点?

- 1.只有满足条件循环才会执行.
- 2.注意死循环的出现.会一直循环执行下去,除非程序强制退出.
- 3.如果碰到**continue**,则跳出本次循环继续下一次判断,如果条件成立继续执行循环.
- 4如果碰到**break**无论条件满足与否,循环立刻结束.
- 5.**continue**与**break**跳出的都是当前所在的循环作用域.如果外层有循环,不受影响.
- 6.**do while**至少执行一次.而**while**不一定.

```

4  /*while的用法
5  while(条件){
6      需要反复执行的代码
7  }
8  */
9  int a=0;
10 while (a<10) {
11     a++;|
12     printf("hello world!\n");
13 }
14 return 0;
15 }
16

```

## 运行

[illegible]

```

22  */
23  int a=0;
24  do{
25      a++;
26      printf("hello world!\n");
27
28  }while (a<10);
29  return 0;
30  }
31

```

```

localhost:ios_workspace wengfada$ cc 24循环结构while与dowhile.c
localhost:ios_workspace wengfada$ ./a.out
hello world!
hello world!
hello world!
hello world!
hello world!
hello world!
hello world!
hello world!
hello world!
hello world!
localhost:ios_workspace wengfada$

```

do-while循环无论如何,循环体的代码都会执行一次.执行完才会进行判断.

```

}while (a<0):
/*特点
  注意死循环的出现.会一直循环执行下去,除非程序强制退出.
  在终端下必须安control+c才能强制退出程序.终止死循环
*/

```

```

41  */
42  int a=0;
43  while (a<10) {
44      if(5==a){
45          continue;
46      }
47      printf("第%d次 ",a);
48      printf("----hello world!\n");
49      a++;
50  }

```

```

localhost:ios_workspace wengfada$ cc 24循环结构while与dowhile.c
localhost:ios_workspace wengfada$ ./a.out
第0次 ----hello world!
第1次 ----hello world!
第2次 ----hello world!
第3次 ----hello world!
第4次 ----hello world!

```

在第五次打印后程序卡住了,原因是 $a==5$ 满足之后跳出本次循环,进行下一次循环, $a$ 仍然是5,又跳出循环,进入下一次循环, $a$ 仍然是5  
解决


```

42  int a=0;
43  while (a<10) {
44      if(5==a){
45          a++;
46          continue;
47      }
48      printf("第%d次 ",a);
49      printf("----hello world!\n");
50      a++;

```

此时跳入死循环

## 注意a++书写的位置

```
42     int a=0;
43     while (a<10) {
44         if(5==a){
45
46             continue;
47             a++; 
48         }
49         printf("第%d次 ",a);
50         printf("----hello world!\n");
```

```
localhost:ios_workspace wengfada$ cc 24循环结构wh
localhost:ios_workspace wengfada$ ./a.out
第0次 ----hello world!
第1次 ----hello world!
第2次 ----hello world!
第3次 ----hello world!
第4次 ----hello world!
```



continue与break跳出的都是当前所在的循环作用域.如果外层有循环,不受影响.

```
41  */
42  int a=0;
43  int b=1;
44  while(b<3){
45      while (a<10) {
46          if(5==a){
47              a++;
48              continue;
49              // a++;//在同一个作用域下continue下面不能写代码.因为不
              会执行.
50          }
51          //如果碰到break无论条件满足与否,循环立刻结束.
52          if (8==a) {
53              break;
54          }
55          printf("第%d次 ",a);
56          printf("----hello world!\n");
57          a++;
58      }
59      printf("外循环打印了\n");
60      b++;
61  }
62  return 0;
```

```
localhost:ios_workspace wengfada$ cc 24循环结构while与dowhile.c
localhost:ios_workspace wengfada$ ./a.out
第0次 ----hello world!
第1次 ----hello world!
第2次 ----hello world!
第3次 ----hello world!
第4次 ----hello world!
第6次 ----hello world!
第7次 ----hello world!
外循环打印了
外循环打印了
localhost:ios_workspace wengfada$
```

## 二.for循环

一,是什么?

是指编程循环语句,当满足条件时开始循环判断执行,直到不满足条件跳出循环.

二,有什么用?

可以根据条件反复执行一段代码

三,怎么用?

for(;;){}里面是3个语句,两个分号。第一个语句是开始前执行,第二个语句是判断真假,如果真,就执行后面(大括号内)的代码。第三个语句是每次执行完毕后执行

四,什么时候用?

当需要重复执行一段代码或满足一定条件重复执行的一短代码实现相应的功能.

五,有什么特点?

1.只有满足条件循环才会执行.

2.注意死循环的出现.会一直循环执行下去,除非程序强制退出.

3.如果碰到continue,则跳出本次循环继续下一次判断,如果条件成立继续执行循环.

4如果碰到break无论条件满足与否,循环立刻结束.

5.continue与break跳出的都是当前所在的循环作用域.如果外层有循环,不受影响.

## for循环执行顺序

注意正确的执行顺序是1-2-3-4 - 2-3-4 - 2-3-4....

下图表示的是最常见的错误理解:

```
5  int main()  
6  {  
7      1      2      4  
8      for(int i = 0; i < 10; i++)  
9      {  
10         3  
11         printf("第%d次Hello world\n", i);  
12     }  
13  
14 }
```

```
}
```

执行顺序:语句1-->判断语句2是否成立--成立-->循环体-->语句3-->判断语句2是否成立--成立-->循环体-->语句3-->再判断语句2...直到语句2不成立.for循环终止.

1.for循环第一个语句:只会执行一次:一般用于初始化变量.且如果是定义同一类型的变量,可以定义多个.方法:int a=1,b=2;

```
for (int a=0, b=2; a<10; a++) {  
printf("当前是第%d行Hello world!\n",a);  
}
```

2.for循环第二个语句:是一个条件语句,控制for循环是否要继续执行.如果这个条件不满足,for循环结束了.

注意:这个必须要有一个判断结果.如果为空,表示默认为真.则是死循环

```
for (int a=0; ; a++) {  
printf("当前是第%d行Hello world!\n",a);  
}
```

3.for循环第三个语句不一定会执行.一般用于控制循环增量或变化.

注意:不能放变量定义语句.

```
for (int a=0; a<10; int b=0) {  
printf("当前是第%d行Hello world!\n",a);  
}
```

4.for循环的循环体:没有限制,可以放任意你想执行的语句.

```
*/  
for (int a=0; a<10; a++) {  
    printf("当前是第%d行Hello world!\n",a);  
}  
//for循环中定义的变量,只在for循环中有效  
//printf("%d",a);  
return 0;
```

```
*/  
for (int a=0; a<10; a++) {  
    printf("当前是第%d行Hello world!\n",a);  
}  
return 0;  
}
```



```

3  */
4  //循环嵌套.如果子循环本身定义了变量,用自己的.如果本身作用域没有定义变量,
5  则会去上一层的作用域寻找,依次类推.(就近原则)
6  for (int a=0; a<2; a++) {
7      for (; a<10; a++) {
8          if(5==a){
9              continue;
10         }
11         if (8==a) {
12             break;
13         }
14         printf("当前是第%d行Hello world!\n",a);
15     }
16     printf("外循环当前是第%d行Hello world!\n",a);
17 }

```

```

/*
1.请问break与continue有什么区别?
  1.break:可以用于swtich和while,do while和for循环
  但continue不能用于switch.
  2.break是结束当前switch或循环语句.
  continue是结束本次循环,继续判断执行下一次循环.
相同点:在break与continue相同作用域下.二者下面都不能放代码.因为不会执
行.还有二者都是关键字.

```

```

*/
/*2.for语句与while语句可以互换吗?二者的相同点及区别是什么?
  1.可以互换.
  2.二者都是去根据条件反复执行相同的代码.
  3.区别是:二者结构用法不一样.
  while(条件判断){
  循环体...
  }
  do{
      循环体...
  }while(条件判断);
  for(语句1;语句2;语句3){
      循环体..
  }

```

## 三种循环结构死循环的最简单实现

```
while (1) {  
    ;  
    return 0;  
}
```

```
do;while (1);  
    return 0;
```

```
for (; ; )  
    ;  
return 0;
```