

循环结构

一日三省且温故知新



为什么要使用循环

如果想要执行类似下面的程序：

```
console.log( 1 + 'hello' );  
console.log( 2 + 'hello' );  
console.log( 3 + 'hello' );  
...  
console.log( 100 + 'hello' );
```

上述功能可以使用循环结构来代替
循环，就是一遍又一遍的重复执行相同或者相似的代码。

1、什么是循环结构

反复一遍又一遍的做着相同(相似)的事情

什么是循环

生活中的循环：



打圈5的单车



4000米赛跑



1、什么是循环结构

反复一遍又一遍的做着相同(相似)的事情

什么是循环

循环中的两大要素：

1、循环条件

什么时候开始，什么时候结束

2、循环操作

循环体，循环过程中 干了什么



2、循环结构-while循环

2.1、语法

```
while( boolean表达式 ){  
    循环体语句;  
}
```

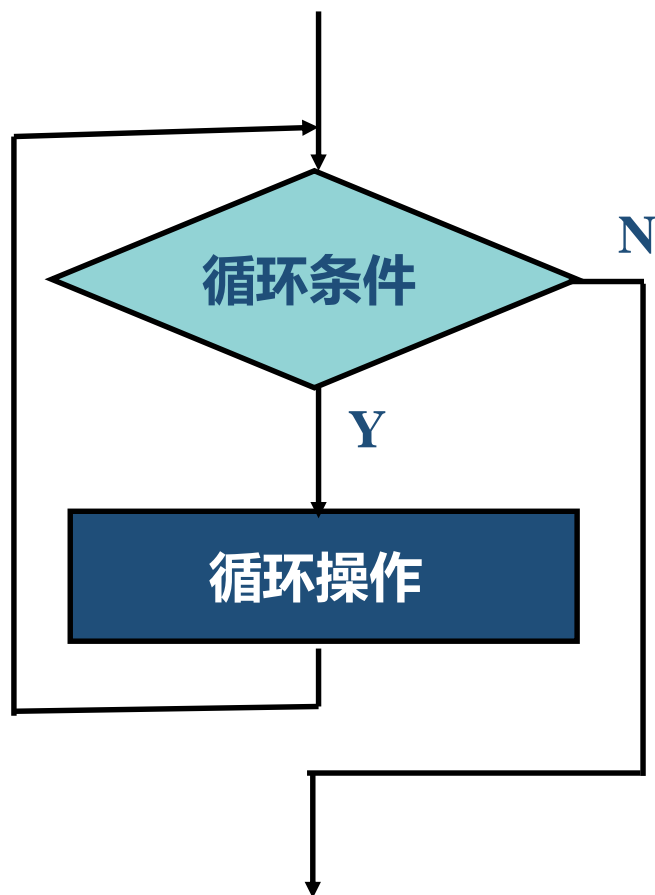
流程:

- 1、判断 条件
- 2、如果条件为真，则执行循环操作
- 3、再判断条件，如果为真，继续执行循环操作
- ...
- n、判断循环条件，如果条件为假，则退出循环结构

注：如果 循环条件一直为真的话，那么循环就是个死循环，
尽量避免死循环的产生。一定要记得更新循环变量(条件)



While语句流程图



需要注意:

一般情况下，循环操作中会存在使得循环终止的可能性，否则将成为“死循环”。



While语句用于处理循环逻辑

```
var i = 0 ;  
  
while ( i < 10 ) {  
    console.log( ' 你好世界 ' );  
    i ++ ;  
}
```

循环条件每执行一次就改变一次，使得循环结束成为可能



2、循环结构-while循环

2.2、循环的流程控制

1、continue 关键字

作用：用在循环中，用于终止本次循环，继续执行下次循环

2、break 关键字

作用：用在循环中的话，终止整个循环结构!!!

```
var i = 0 ;  
while ( i < 100 ) {  
    console.log( i );  
    if( i == 5 ){  
        break;  
    }  
    i ++ ;  
}
```

输出的结果是：

0 1 2 3 4 5, 当 i==5时, 退出循环。



3、循环结构- do-while循环

3.1、语法

```
do{  
    可执行语句;  
} while( boolean表达式 );
```

无论循环条件是
否满足，循环体
至少执行一遍

流程：

- 1、先执行循环操作
- 2、再判断循环条件
如果条件为真，则继续执行循环操作
如果条件为假，则终止循环即退出循环

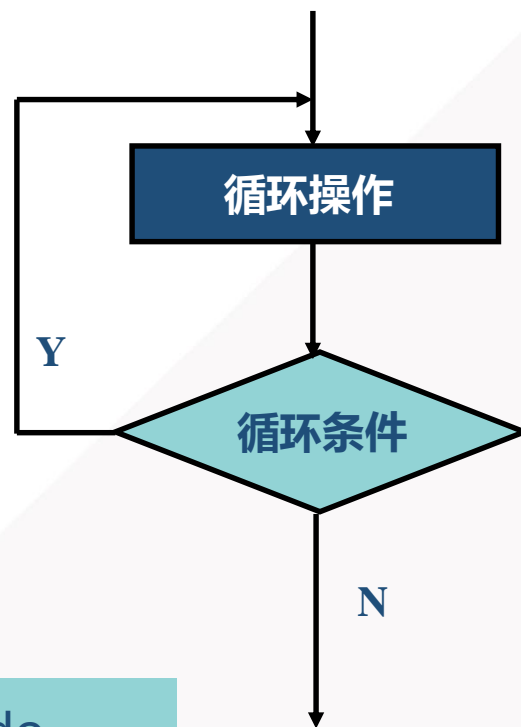
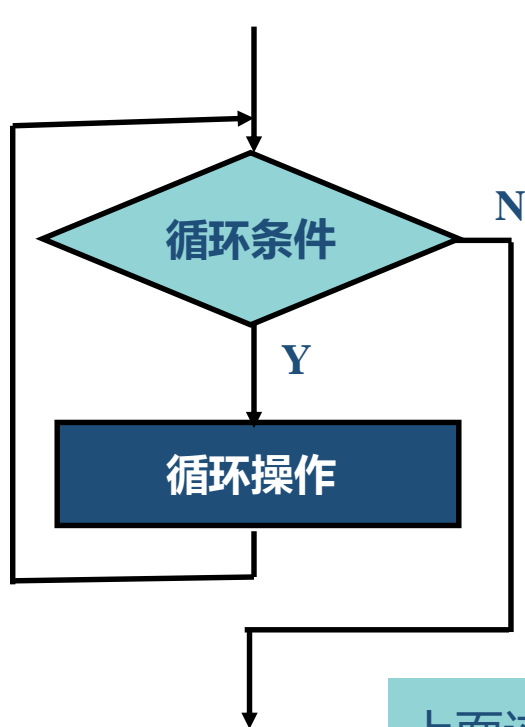
特点：

无论条件满足与否，至少要执行一次循环操作



3、循环结构- do-while循环

3.2、do-while语句流程图



上面这两幅流程图哪一副是在描述do-while? 哪一副是在描述while呢?



3、循环结构- do-while循环

3.3、while 与 do...while 区别

1、while

先判断，后执行

最少一次循环都不执行(第一次条件就不满足)

2、do...while

先执行，后判断

最少要执行一次循环操作(第一次条件就不满足)



4、for循环

4.1、问题

打印 1 - 100 之间所有的数字

var i=1;//1、声明循环条件

while(i<=100){//2、循环条件判断

 console.log(i);//3、循环体-打印输出

 i++;//4、更新循环条件

}

结构不清晰



4、for循环

```
for (表达式1; 表达式2; 表达式3) {  
    循环体语句;  
}
```

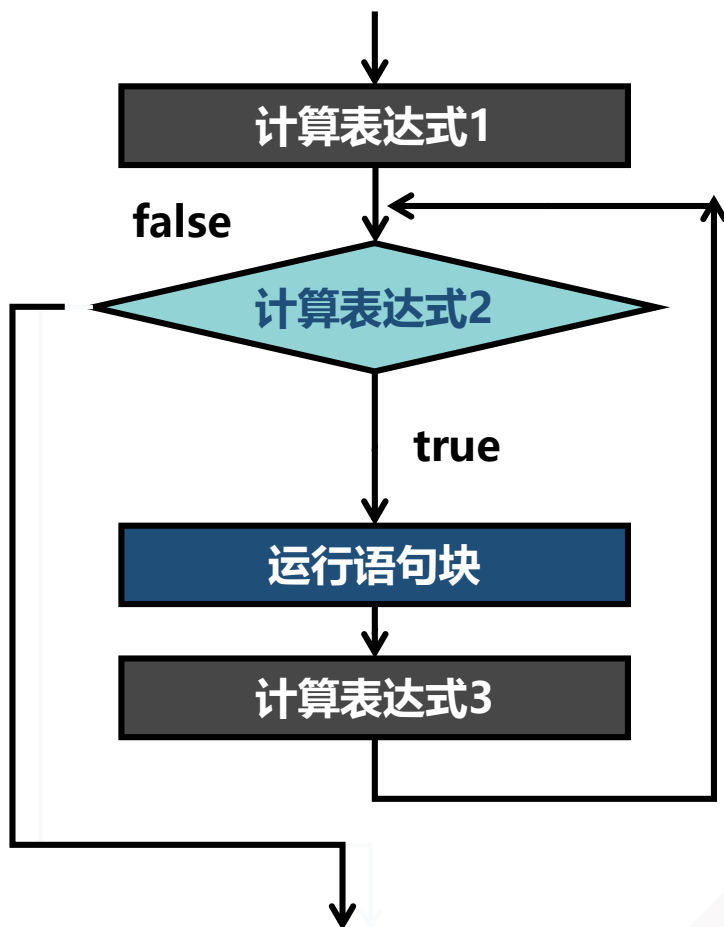
for循环执行过程:

1. 计算表达式1的值;
2. 计算表达式2 (boolean表达式) 的值, 如果为true则执行循环体, 否则退出循环;
3. 执行循环体;
4. 执行表达式3;
5. 计算表达式2, 如果为true则执行循环体, 否则退出循环;
6. 如此循环往复, 直到表达式2的值为false。



4、for循环

4.2、流程图



```
for ( var i = 0 ; i < 10 ; i ++ ) {  
    console.log( i );  
}
```

// 程序输出结果为:

// 0 1 2 3 4 5 6 7 8 9



4、for循环

4.3、for循环用于实现固定次数的循环

累加

求 $1 + 2 + 3 + \dots + 100 = ?$

```
var sum = 0 ;  
for ( var i = 1 ; i <= 100 ; i ++ ) {  
    sum += i ;  
}  
console.log("1到100的和为: " + sum ) ;
```

阶乘

求 $1 * 2 * 3 * \dots * 10 = ?$



4、for循环

4.4、for, while, do-while 之间的区别

1、for, while

最少执行 0 次

while:多数用于不确定循环次数时使用

for:多数用于确定循环次数时使用

2、do-while

最少执行 1 次

适合用于不确定循环次数时使用

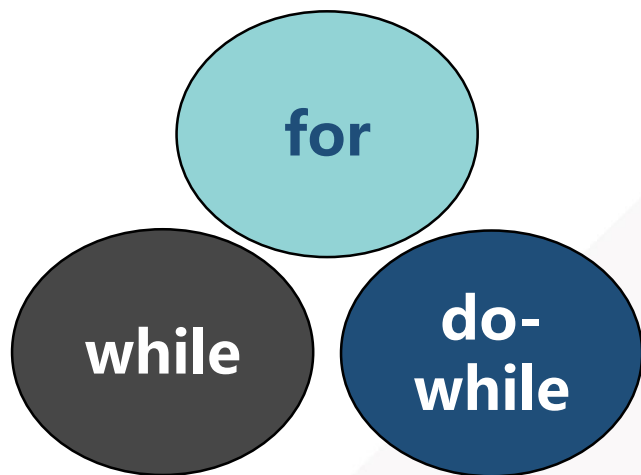


4、for循环

4.5、循环的嵌套

在一个循环体内再出现一个循环

for / while / do-while 三种循环在很多情况下是可以互换的；
一般情况下，for循环使用最多





小结

循环结构

为什么要使用循环

什么是循环结构

循环结构-while循环

循环结构- do-while循环

for循环