



做真实的自己，用良心做教育

H5教学部

JS循环语句

# 目录

做真实的自己，用良心做教育

1

循环语句while

2

循环语句do-while

3

循环语句for

4

break和continue

# 循环语句

## 1. JS循环的概念

循环就是重复做一件事，在程序中指的是重复执行某段代码。

循环结构是程序中一种很重要的结构，其特点是在给定的条件成立时，反复执行某程序段，直到条件不成立为止。

如：大家每天早上7点起床，9点上课，12点吃饭，19点晚自习，21点下课，23点睡觉，第二天又重复7点起床，.....，直到毕业找到工作就可以跳出这个循环。

## 2.死循环的概念

死循环就是重复执行代码，不会停止。

死循环会造成程序卡死甚至崩溃等问题，我们不能写会造成死循环的代码。

## 3.循环的用处

循环的用处非常广泛，特别是对大量的数据进行操作的时候，

如：求若干个数之和；（重复做加法）

求某个班全部学生的总成绩；（重复做加法）

# 循环语句

## 4. 循环的分类

循环有很多种方式, 如: while, do-while, for, for-in, for-each等.

这里我们先介绍最常见的,也是很多其他计算机语言通用的三种方式:

1). while循环

2). do-while循环

3). for循环

# 循环语句while

## 1. while循环语法格式:

```
while(表达式)
{
    执行代码;
}
```

注意: 1, while循环必须按照上述语法格式来写, 只能更改表达式内容和执行代码.

2, 表达式可以是常量,变量或者其他表达式,该表达式会被强制转换为boolean类型, 可以理解表达式就是循环的条件, 条件成立则进入括号{}里面执行代码, 否则不进入;

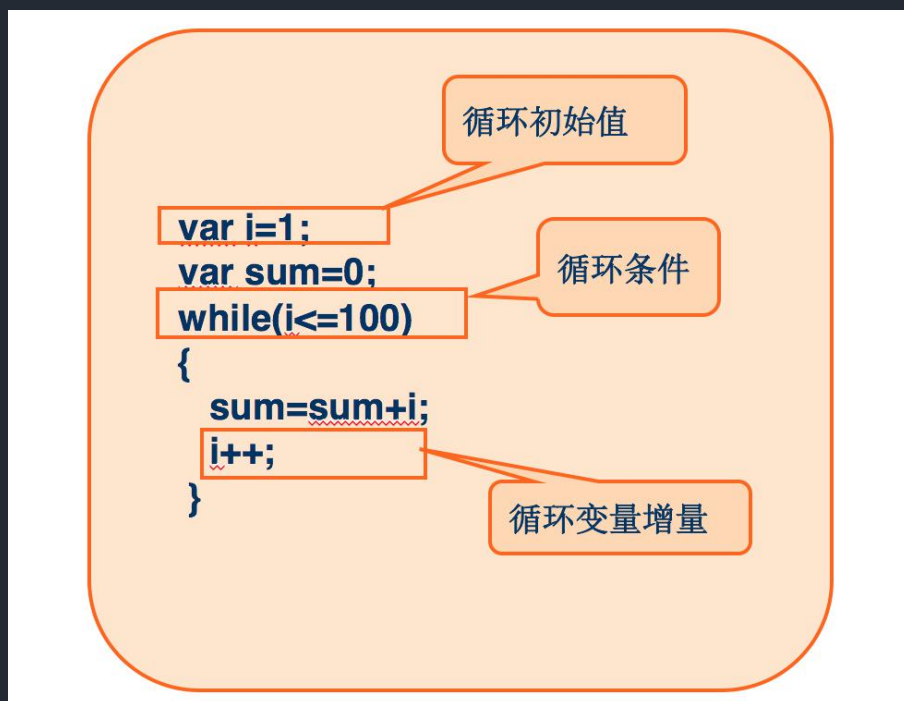
3, 执行代码可以有多行, {}所包含的称为循环体.

4, 表达式不要一直是true, 避免死循环.

5. while循环的顺序是: 先判断表达式是否为true, 若true则执行执行代码, 然后再继续判断表达式是否为true, .... 直到判断表达式为false才会跳出循环, 执行while后面的代码

# 循环语句while

2. 示例：计算 $1+2+3+\dots+100$ 之和。



while循环一般都会有: 循环初始值, 循环条件 和 循环变量增量(或者减量)

# 循环语句while

## 3. 练习

1. 打印100以内 7的倍数
2. 打印100以内的奇数
3. 打印100以内所有偶数的和
4. 打印图形

\* \* \* \* \*

\* \* \* \* \*

\* \* \* \* \*

\* \* \* \* \*

# 循环语句do-while

## 1. do-while循环语法格式:

```
do {  
    执行代码;  
} while(表达式)
```

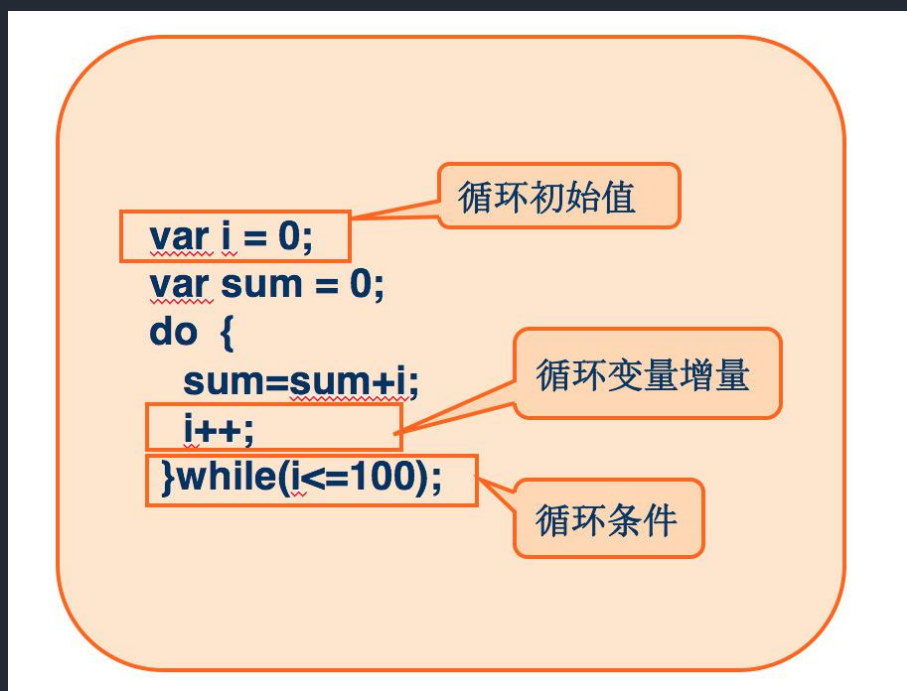
注意: 1, do-while和while比较类似, 也有表达式和执行代码.

2, do-while和while的区别在于判断条件的先后不一样, while是先判断条件, 而do-while是先执行再判断; 所以不管do-while表达式的条件是否成立, 都至少会执行一次循环体.



# 循环语句do-while

2. 示例：计算 $1+2+3+\dots+100$ 之和。



do-while循环一般也会有: 循环初始值, 循环条件 和 循环变量增量(或者减量)

# 循环语句do-while

## 3. 和while的区别

执行顺序的问题，while是先判断，do-while是先执行循环体的代码，再判断。当条件第一次就不满足时，更能体现出区别。

在开发中，while用的情况多。如果while处理逻辑比较别扭时，就需要使用do-while。

示例：举一个第一次条件判断为false的例子

# 循环语句for

## 1. for循环语法格式:

```
for (表达式1; 表达式2; 表达式3) {  
    执行代码;  
}
```

注意: 1, 表达式1: 可写循环初始值, (可以不填)

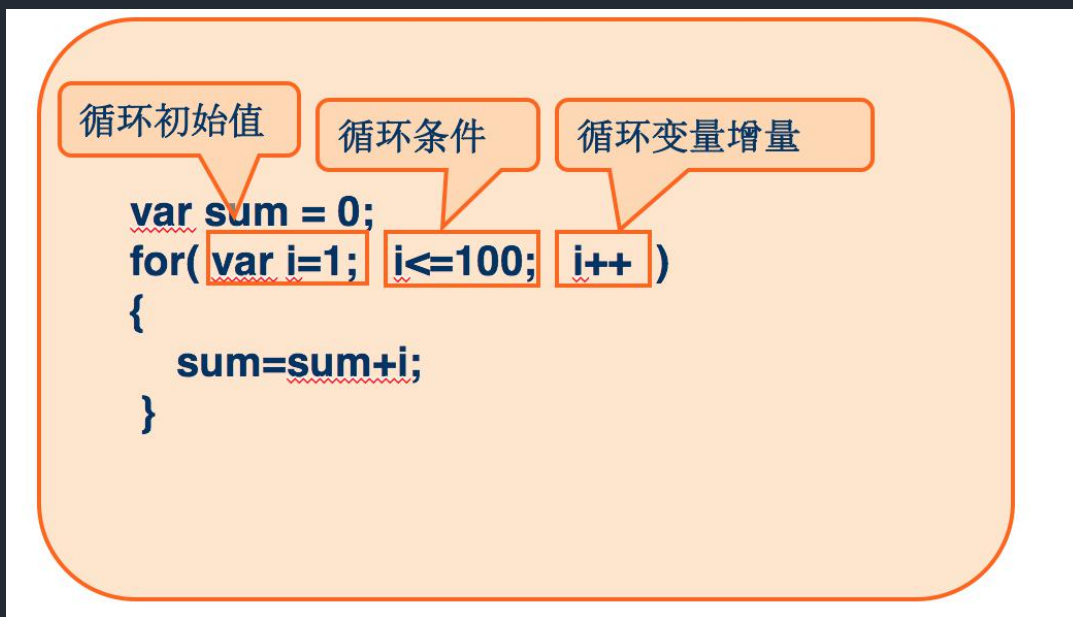
2, 表达式2: 循环条件

3, 表达式3: 循环增量(减量)

4, 两个分号必须写.

# 循环语句for

2. 示例：计算 $1+2+3+\dots+100$ 之和。



for循环将 循环初始值, 循环条件, 循环变量增量写在了一起

注意: 三个表达式都是可选填写, 但是如果后两个不写,会造成死循环; 而第一个为初始化值,所以我们三个一般都要写上.

# 循环语句for

## 3. 循环的嵌套

一个循环语句内又包含另一个完整的循环语句;

三种循环都可以相互嵌套; 而for循环嵌套for循环的使用会更频繁, 以下的循环嵌套都是合法的.

```
while()  
{  
    while()  
    {...}  
}
```

```
do  
{  
    do  
    {...}while();  
}while();
```

```
for(;;)  
{  
    for(;;)  
    {...}  
}
```

```
do{  
    while()  
    {...}  
}while();
```

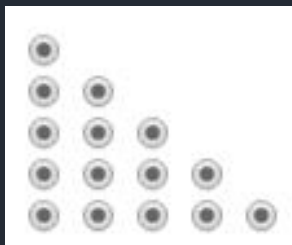
```
while()  
{  
    do{  
    }while();  
}
```

```
for(;;)  
{  
    while()  
    {...}  
}
```

# 循环语句for

## 4. 示例

打印三角形



# break和continue

## break关键字

break的功能:

- 1,在switch语句中使流程跳出switch结构。
- 2,在循环语句中使流程跳出当前循环

注意:

- 1, 如果已执行break语句，就不会执行循环体中位于break后的语句。
- 2, 在多层循环中，一个break语句只向外跳一层

示例:

- 1, 判断一个数是不是合数。(指自然数中除了能被1和本身整除外，还能被其他的数整除（不包括0)的数。)
- 2, 判断一个数是不是素数。(除了1和它本身以外不再其他的除数整除。)

# break和continue

## continue关键字

continue的功能:

只能在循环语句中使用，使本次循环结束，即跳过循环体中下面尚未执行的语句，接着进行下次是否执行循环的判断。

注意:

1, continue语句只能用在循环里。

2, 对于 while 和 do-while 循环，continue 语句执行之后的动作是条件判断；对于 for 循环，随后的动作是变量更新。

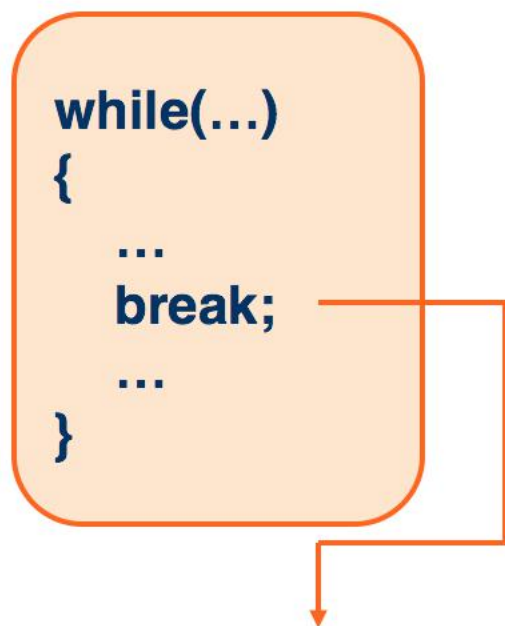
示例:

1, 求整数1 ~ 100的累加值，但要求跳过所有个位为3的数。

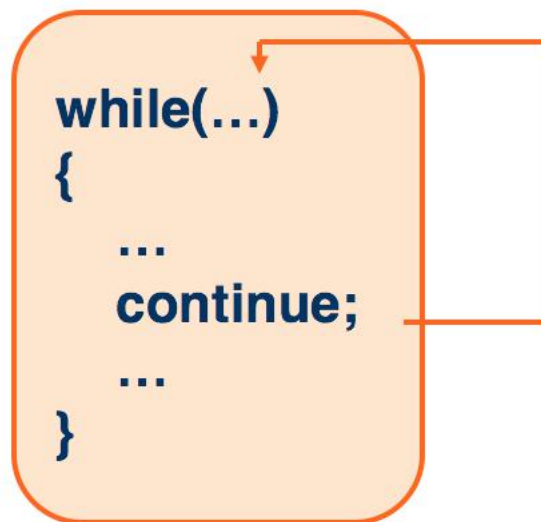


# break和continue

## break和continue对比



跳出整个循环



继续下一次循环

## 练习

- 1, 一个新入职，月工资为2000元的员工，每年涨当年工资5%，20年后的月工资是多少？
- 2, 山上有一口缸可以装50升水，现在有15升水。老和尚叫小和尚下山挑水，每次可以挑5升。问：小和尚要挑几次水才可以把水缸挑满？通过编程解决这个问题。
- 3, 打印100–200之间所有能被3或者7整除的数
- 4, 计算10的阶乘 ( $1*2*3*4*5*6*7*8*9*10$  n的阶乘 $1*2*...*n$ )
- 5, 计算 $1+3+5+...+99$ 的和
- 6, 99乘法表
- 7, 输出20~80之间能被3整除的整数, 每行5个
- 8, 打印1000~2000年中所有的闰年, 每行4个
- 9, 求:  $1-1/2+1/3-1/4 \dots 1/100$ 的和

## 练习

9, 输入两个数, 求两个数的最小公倍数 (\*\*\*\*)

如: 9和6的最小公倍数是18,

1, 先找出两个数中的最大数

2, 最大数++, 找出能被两个数整除的数(退出循环)

10, 输入两个数n, a, 如果n==3, a == 2;

输出  $2 + 22 + 222$  的值。(不用输出式子) (\*\*\*\*)

如: n == 4, a == 3;

输出  $3 + 33 + 333 + 3333$  的值。

提示: 1、n = 3, 相加三次, 每次相加比前一次相加的数, 多一位

2、每次多的这个位数的值为a,  $3, 3*10+3(33), 33*10+3(333), \dots$

11, 五位数中, 对称的数称为回文数, 找出所有的回文数。

如: 12321

## 练习

12, 宰相的麦子：相传古印度宰相达依尔，是国际象棋的发明者。有一次，国王因为他的贡献要奖励他，问他想要什么。达依尔说：“只要在国际象棋棋盘上（共64格）摆上这么些麦子就行了：第一格一粒，第二格两粒，……，后面一格的麦子总是前一格麦子数的两倍，摆满整个棋盘，我就感恩不尽了。”国王一想，这还不容易，刚想答应，如果你这时在国王旁边站着，你会不会劝国王别答应，为什么？

**THANK YOU**



做真实的自己，用良心做教育