

千锋HTML5学院

第二阶段javascript课程课件



1

循环的概念

2

While循环

3

do While循环

4

for循环

5

break和countinue



循环的概念

□ 循环就是重复做一件事

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

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）

.....

□ 分类

while 循环

do...while 循环

for 循环



while循环

□ while循环

语法格式:

```
while (表达式)
```

循环条件

```
{
```

```
    语句;
```

循环体

```
}
```

注意:

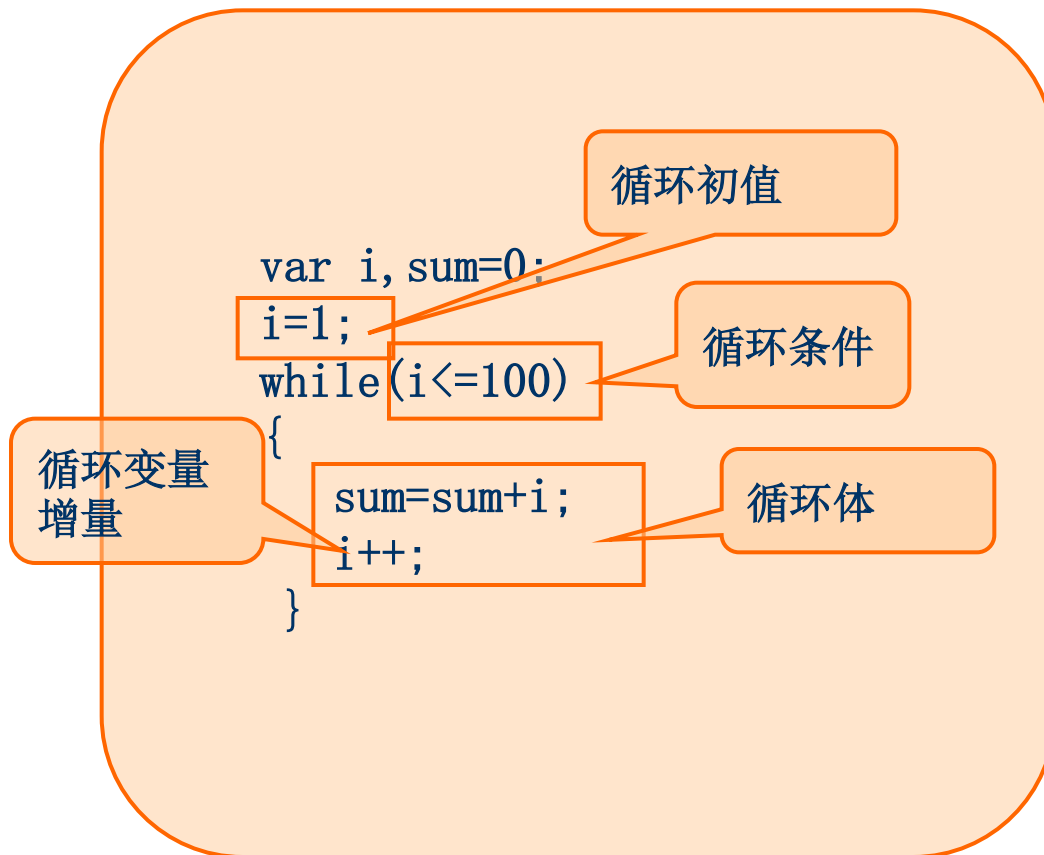
1. 先判断表达式, 表达式成立后执行语句。
2. 循环条件中使用的变量(循环变量)需要经过初始化。
3. 循环体中, 应有结束循环的条件, 否则会造成死循环。
4. 循环体中可以写若干句合法的JavaScript代码, 包括if, 也可以再套个循环语句。

示例: 计算1+2+3+.....+100之和。



while循环

□ 示例：用while循环求1~100之和。





while循环

□ 示例：

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

5. 用循环输出一个年份的下拉框（结合HTML代码的）



do while循环

□do...while循环

语法格式：

```
do  
{  
    语句;  
}while(表达式);
```

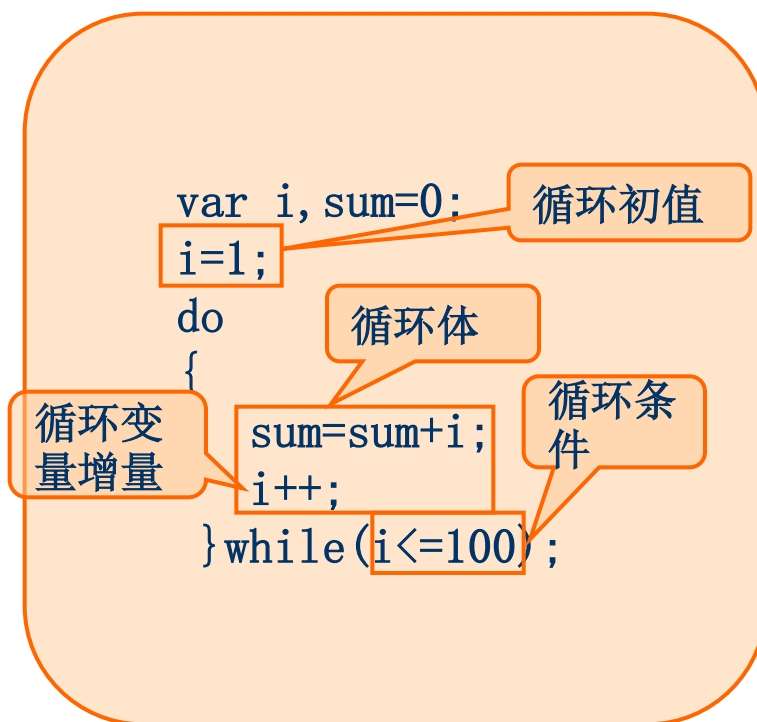
- 注意：
1. 最后的分号不要忘记。
 2. 循环条件中使用的变量需要经过初始化。
 3. 循环体中，应有结束循环的条件，否则会造成死循环。

示例：计算 $1+2+3+\dots+100$ 之和。



do while循环

□例题：用do...while循环求1~100之和。





while和do while的区别

□ 执行顺序的问题，while是先判断，do while，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。



for循环

□for循环

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

执行过程：

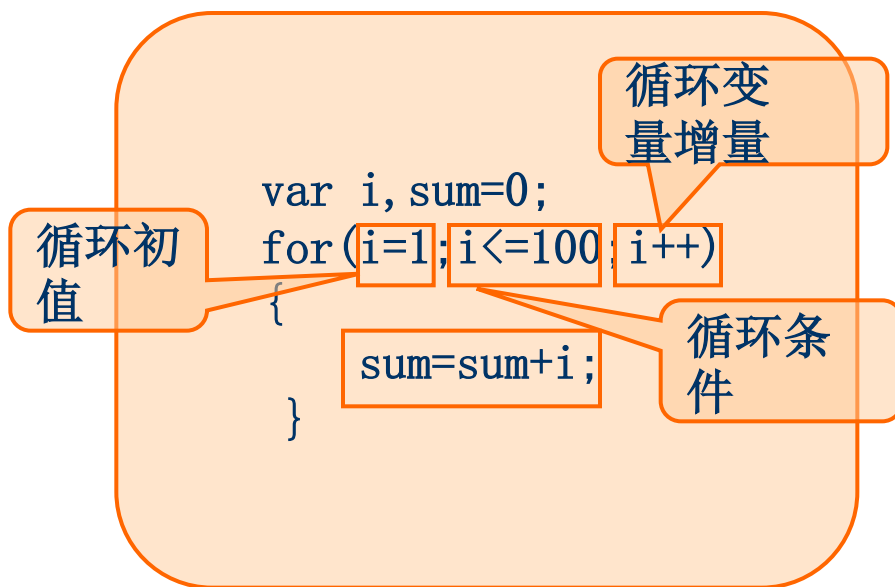
1. 先求解表达式1（只求一次）；
2. 求解表达式2，若其值为真（非0），
则执行for语句中指定的内嵌语句，
然后求解表达式3；再求解表达式2，若为假，
则结束循环，执行for循环外的语句。

示例：计算**1+2+3+.....+100**之和。



for循环

□用for循环求1~100之和。





for循环表达式

□for循环三个表达式的说明：

三个表达式都可以省略，但分号不能省略，省略后都可改写。

如：省略表达式1，需在循环外初始化循环变量。

```
var i;  
for(i=1;i<=100;i++)  
{  
    sum=sum+i;  
}
```



```
var i=1;  
for( ;i<=100;i++)  
{  
    sum=sum+i;  
}
```

3个表达式都省略：例： `for(;;);` 这是个死循环



循环嵌套

□ 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。
- 3中循环可以互相嵌套，下面几种都是合法的形式。

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

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

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

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

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

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



循环嵌套示例

□示例

1、打三角



2、打三角



3、打钻石





break

□break语句功能:

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

□强调:

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

□示例:

判断一个数是不是合数。

判断一个数是不是素数。



continue

□ continue语句的功能：

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。

□ 强调：

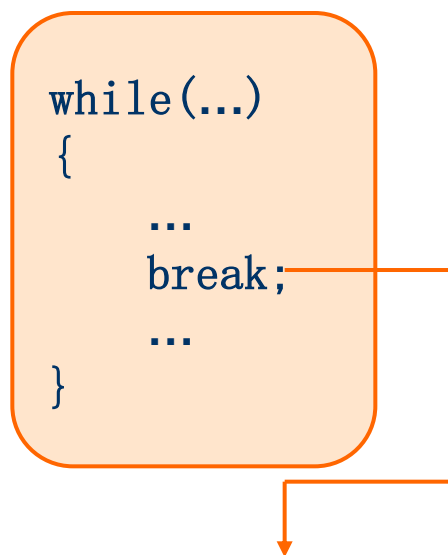
- continue语句只能用在循环里。
- 对于 while 和 do-while 循环，continue 语句执行之后的动作是条件判断；对于 for 循环，随后的动作是变量更新。

□ 示例：

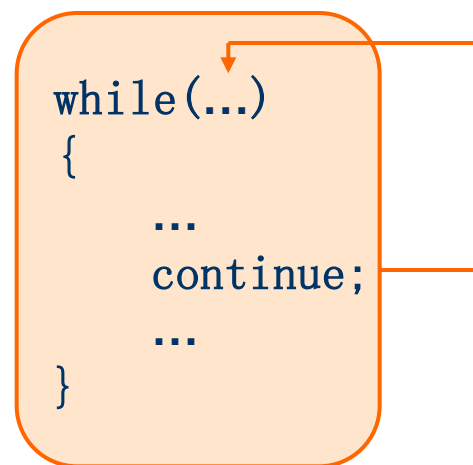
- 1、求整数1~100的累加值，但要求跳过所有个位为3的数。
- 2、输出100以内的数，逢7的倍数，含7的数，输出“过”。



break和continue的对比



跳出整个循环



继续下一次循环



注意点

□选择for还是选择while

一般情况下，已知循环次数时，用for，否则用while。

□如何造死循环： **while (true) { }**

□循环嵌套：

长循环在内层，短循环在外层，效率高,影响效率的主要体现：两个循环之间的切换，切换次数越少，效率越高。

```
for (col=0; col<5; col++)  
{  
    for (row=0; row<100; row++)  
    {  
        .....  
    }  
}
```

```
for (row=0; row<100; row++)  
{  
    for ( col=0; col<5; col++)  
    {  
        .....  
    }  
}
```



注意点

❑不要在 **for** 循环体内修改循环变量，防止循环失控。

❑循环要尽可能的短，要使代码清晰，一目了然
如果你写的一个循环的代码超过一显示屏，那会让读代码的人发狂的。解决的办法有两个：第一，重新设计这个循环，确认是否这些操作都必须放在这个循环里；第二，将这些代码改写成一个子函数，循环中只调用这个子函数即可。

❑把循环嵌套控制在**3**层以内。

解决方法：建议你重新设计循环或是将循环内的代码改写成一个子函数。



示例

1. 一个新入职，月工资为2000元的员工，每年涨工资5%，到退休时的月工资是多少？
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. 输出100–200之间所有能被3或者7整除的数
4. 计算10的阶乘
5. 计算 $1+3+5+\dots+99$ 的和
6. 99乘法表
7. 求100-999之间的水仙花数。（水仙花数：三位数中，每位数字立方和等于自身，如： $153 == 1*1*1+5*5*5+3*3*3$ ）



作业

必做:

1. 把课堂案例独立敲一遍
2. 输出出1-100里所有的偶数
3. 输出出1-100里所有的奇数
4. 求出 $1 - 1/2 + 1/3 - 1/4 + \dots + 1/100$ 的结果
5. 输出20-80之间能被3整除的整数，每5个一行
6. 输出出1000-2000年中所有的闰年，并以每行四个数的形式输出
7. 输出九九乘法表
8. 输入一个数，判断其是否为素数
9. 求100-200之间的素数

可选:

1. 任给正整数 n , 计算 $1! + 2! + 3! + \dots + n!$
2. 求 $S_n = a + aa + aaa + \dots + \underbrace{aa\dots aa}_n$ 之值, 其中 a 是一个数字, n 是文本框输入的。
3. 求 Fibonacci 数列: 1, 1, 2, 3, 5, 8, 前40个数。
4. 一球从100米高度自由落下，每次落地后反跳回原高度的一半；再落下，求它在第10次落地时，共经过多少米？第10次反弹多高？
5. 找数字：求所有满足条件的四位数 $abcd$:
 - (1) 这四位数是11的倍数；
 - (2) a, b, c, d 均是小于10的互不相同的自然数；
 - (3) $b + c = a$ ；
 - (4) bc 是完全平方数。

