

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:

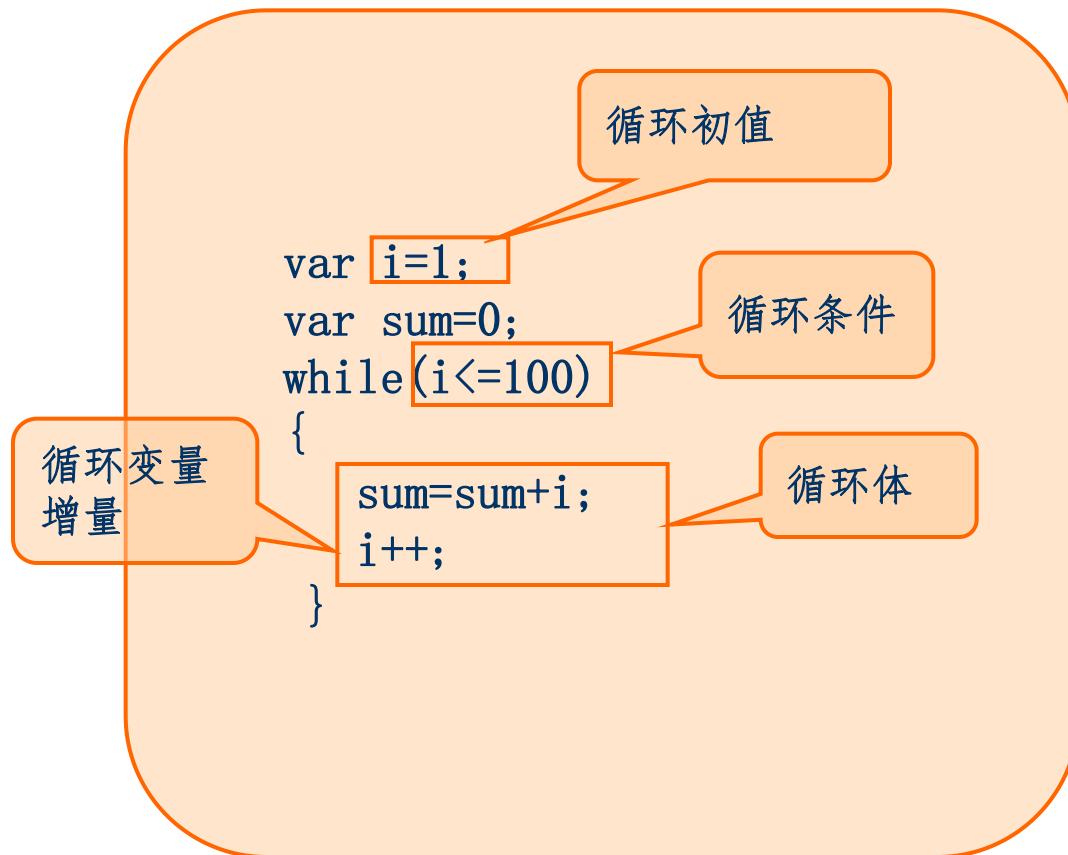
1. 先判断表达式, 表达式成立后执行语句。
2. 循环条件中使用的变量(循环变量)需要经过初始化。
3. 循环体中, 应有结束循环的条件, 否则会造成死循环。
4. 循环体中可以写若干句合法的JavaScript代码, 包括if, 也可以再套个循环语句。

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. 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

* * * * *

* * * * *

* * * * *

* * * * *



do while循环

□do...while循环

语法格式：

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

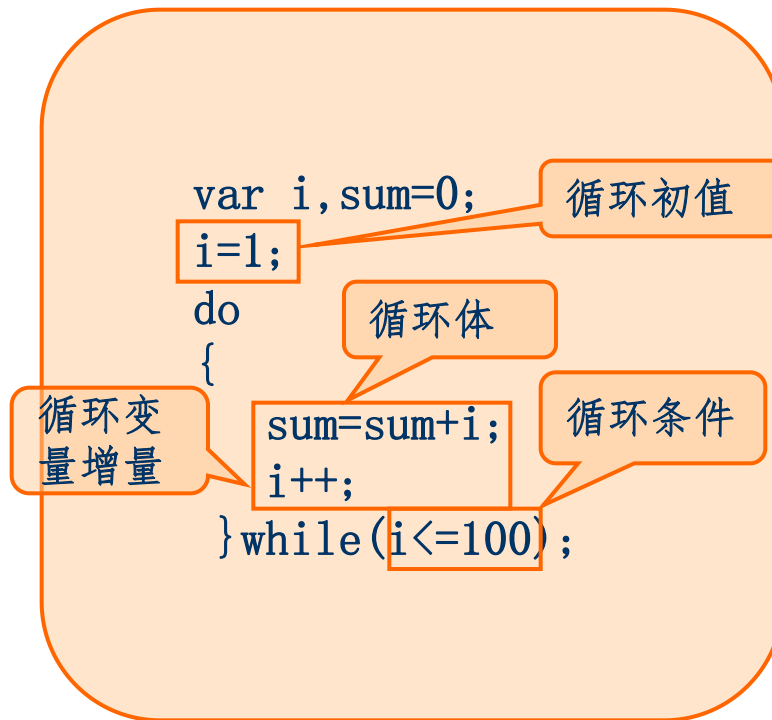
- 注意：
1. 最后的分号尽量写上。
 2. 循环条件中使用的变量需要经过初始化。
 3. 循环体中，应有结束循环的条件，否则会造成死循环。

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：

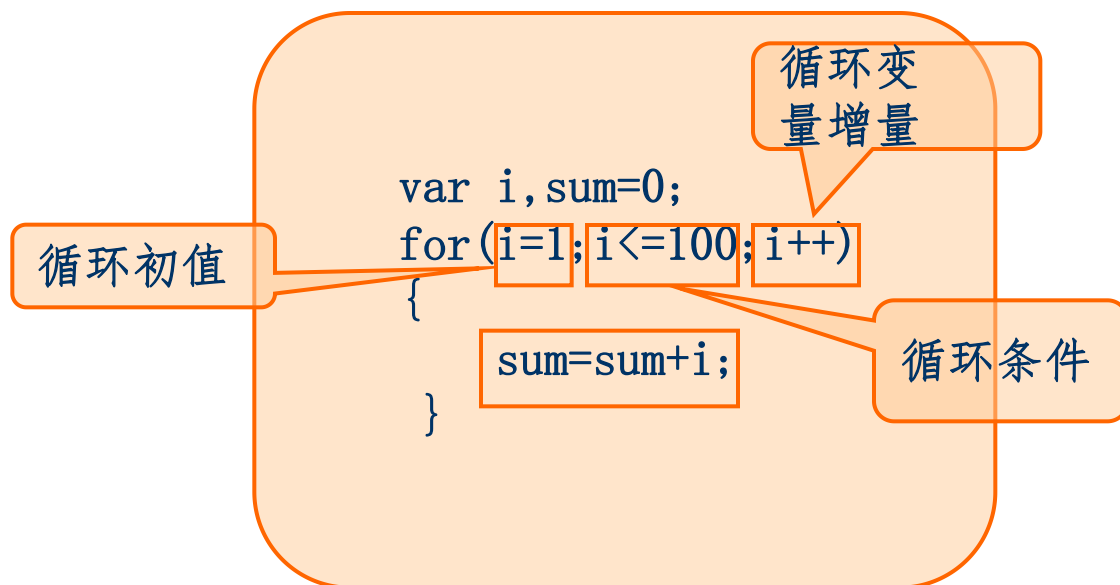
1. 先求解表达式1（只求一次）；
2. 求解表达式2，若其值为真（非0），
则执行for语句中指定的内嵌语句，
然后求解表达式3；再求解表达式2，若为假，
则结束循环，执行for循环外的语句。

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种循环可以互相嵌套，下面几种都是合法的形式。

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

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

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

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

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

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



循环嵌套示例

□ 示例

1、打三角



2、打三角





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。(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。)

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。)



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:

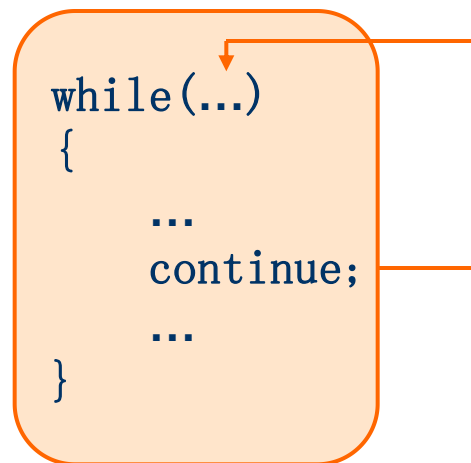
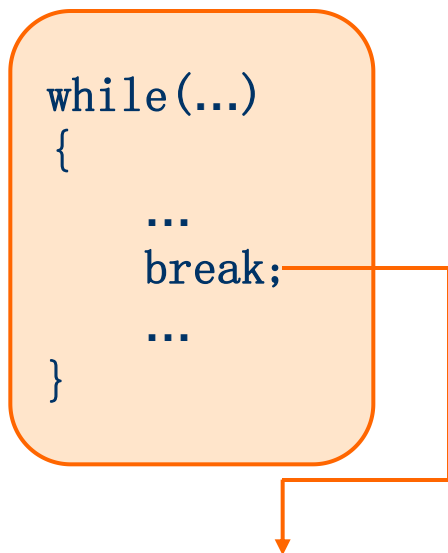
- continue语句只能用在循环里。
- 对于 while 和 do-while 循环，continue 语句执行之后的动作是条件判断；对于 for 循环，随后的动作是变量更新。

示例:

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的对比





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++ )  
    {  
        .....  
    }  
}
```



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. 判断一个数是不是质数。(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。)
8. 判断一个数是不是素数。(除了1和它本身以外不再有任何其他的除数整除。)



作业

必做：

1. 输出20-80之间能被3整除的整数，每5个一行
2. 打印出1000-2000年中所有的闰年，并以每行四个数的形式输出
3. 输出九九乘法表
4. 求出 $1-1/2+1/3-1/4+\dots+1/100$ 的和 `var i=1; (-1)(i+1) (1/i)`

可选：

输出100-1000以内的所有水仙花数； $abc = a^3 + b^3 + c^3$

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？

