

循环

循环的目的：重复执行某些代码。

```
1      <script>
2          //循环的目的：可以重复执行某些代码。
3          console.log('哈哈');
4          console.log('哈哈');
5          console.log('哈哈');
6          console.log('哈哈');
7          console.log('哈哈');
8          console.log('-----');
9          for(var i = 1; i <= 100; i++){
10             console.log('哈哈');
11         }
12     </script>
```

JS中的循环

js中，主要有三种类型的循环语句。

- for循环
- while循环
- do...while循环

for循环

在程序中，一组被重复执行的语句被称为 **循环体**，能够继续重复执行，取决于循环的 **终止条件**。循环体及循环的终止条件组成的语句，被称为 **循环语句**。

语法结构

for循环主要用于把某些代码循环若干次，通常跟计数有关系。语法结构如下：

```
1  <script>
2      //for 语法结构
3      // for(初始化变量;条件表达式;操作表达式){
4      //      //循环体
5      // }
6      //1、初始化变量，使用var声明的一个普通变量，通常用于作为计数器使用
7      //2、条件表达式，用来决定每一次循环是否继续执行，简单来说是终止条件。
8      //3、操作表达式，每次循环最后执行的代码，通常用于计数器变量进行更新（递增或者递减）
9      //4、代码演示，循环100句，hello world
10     for(var i=1;i<=100;i++){
11         console.log('hello world');
12     }
13 </script>
```

for循环的执行过程

```
1      <script>
2          //for 循环的执行过程
3          for(var i=1;i<=100;i++){
4              console.log('hello world');
5          }
6          // 1、首先执行里面的计数器变量 var i = 1,在for里面只执行一次。
7          // 2、i<=100 用来判断是否满足条件。如果满足条件，就去执行循环体，不满足条件则退出循环。
8          // 3、最后去执行 i++ i++是单独写的代码，表示递增。
9          // 4、接着去执行 i <= 100 ,如果满足条件，则接着执行循环体，不满足条件退出循环（第二轮）。
10     </script>
```

断点调试

断点调试是指自己在程序的某一行设置一个断点，调试时，程序运行到这一行就会停住。然后就可以一步一步往下调试，调试过程中可以看到各个变量当前的值，出错的话，调试到出错的代码行会显示错误，停止。

断点调试可以放我们看到程序的运行过程。

使用方法:

浏览器中按 F12--> **sources** --> 找到需要调试的文件 --> 在js程序中某一行设置断点。

watch: 监视，通过watch可以监视变量的值的变化，很常用。

F11: 程序单步执行，让程序一行一行的执行，这个时候，可以观察watch中变量的值的变化。

目前只介绍这几个功能，后面涉及到会深入讲解。

for循环重复执行相同代码

```
1      <script>
2          // for循环可以执行相同的代码
3          for (var i = 1;i < 10;i++) {
4              console.log('123123');
5          }
6          // 通过prompt来控制输出循环的次数
7          var a = prompt('...');
8          for(var i = 1;i <= a;i++){
9              console.log('我是可以被控制得');
10         }
11     </script>
```

for循环重复不相同的代码

```
1      <script>
2          // 循环金额一重复执行不同的代码 ， 因为我们有计数器变量 i 的存在 i 每次循环值都在变化
3          // 我们要输出1个人的年龄，1~99岁
```

```

4      for(var i = 1; i < 100;i++){
5          console.log('我今年是'+ i +'岁');
6      }
7
8
9      // for循环中使用if else if多分支语句
10     for(var i = 1;i < 100 ;i++){
11         if(i == 1){
12             console.log('1岁刚出生');
13         }else if(i == 99){
14             console.log('真棒，活到了99岁了呢');
15         }else{
16             console.log('我今年是'+ i +'岁');
17         }
18     }
19     </script>

```

for循环重复某些相同的操作

for循环因为有了计数器的存在，我们还可以重复的执行某些操作，比如做一些算数运算。

案例1：求1-100之间所有整数的累加和（答案：5050）

思路：

- 1、1-100，需要循环100次。计数器限制为100
- 2、其次需要一个储存结果的变量 `sum`，初始值为 0
- 3、核心算法研究：1+2+3+4+5+6+7....., `sum = sum + i`;

```

1  1+2+3+4+5+6+7+8+9+10
2      3+3
3          6+4
4              10+5
5                  15+6
6                      sum + i
7  sum = sum + i;

```

```

1  <script>
2      var sum = 0;  //和的变量，值为0
3      for(var i = 1;i <= 100;i++){
4          // sum = sum + i;
5          // 赋值运算符 += 的写法
6          sum += i;
7      }
8      console.log(sum);
9  </script>

```

案例2：求1~100之间所有数总和的平均值（ $5050/100=50.5$ 答案50.5）

思路：

- 1、先求1~100的和，方法同上。
- 2、需要一个求 `average` 平均数变量。
- 3、用总和 `average = sum/100` 求平均值。

```
1 <script>
2     var sum = 0; //和的变量，值为0
3     var average = 0; //存储平均数的变量
4     for(var i = 1;i <= 100;i++){
5         // sum = sum + i;
6         // += 赋值运算符的写法
7         sum += i;
8     }
9     average = sum / 100;
10    console.log(average);
11 </script>
```

案例3：求1~100之间所有偶数和奇数的和（偶数的和答案为2550，奇数的和答案为2500）

思路：

- 1、能够被 2 整除的是偶数，否则就是奇数。
- 2、还是需要先设置储存变量的值，一个为偶数变量的值 `even`，一个为奇数变量的值 `odd`

```
1 <script>
2     var even = 0;
3     var odd = 0;
4     for(var i = 1;i <= 100; i++){
5         if(i % 2 == 0){
6             even += i;
7         }else{
8             odd += i;
9         }
10    }
11    console.log('偶数和为'+even);
12    console.log('奇数和为'+odd);
13 </script>
```

案例4：求1~100之间所有能被3整除的数字之和。（答案：1683）

思路：

- 1、只要能够被 3 整除的数字。 `i % 3 == 0`

```
1 <script>
2   var res = 0; //存储1~100能够被3整除的数字之和的变量
3   for(var i = 1;i <= 100;i++){
4       if(i % 3 ===0){
5           res += i;
6       }
7   }
8   console.log('1~100能够被3整除的数字之和为'+res);
9 </script>
```