

DAY03 循环结构

2023年4月20日 9:41

循环的概念和意义

1. 循环结构，就是根据某些给出的条件，重复的执行同一段代码

重复,规律,限制条件

2. 循环必须要有某些固定的内容组成

- 初始化(给变量第一次赋值)
- 条件判断(限制条件)
- 要执行的代码(重复执行的代码)
- 自身改变

3. 死循环：无法靠自身的控制终止的循环即死循环。

```
// for (; ;) { }  
// console.log('hehe');
```

那些循环以及的意义

遍历和循环有很大的关系，遍历就是将某些值一一拿到

while循环

do...while循环

for循环

for...in...循环

for...of...循环

循环的意义就是将重复的事情，利用循环完成，提升开发效率。

注意循环的特点：程序的循环速度非常快，大部分循环需要执行完成，才显示最终的结果。

```
// for (var i = 1; i <= 1000; i++) {  
//   document.write('<li>11111111111</li>');  
// }
```

while循环的语法规则

while (限制条件) {

// 满足限制条件不断执行的循环体

// 必须设定变化的值,防止出现死循环,无限循环。

}

while循环的嵌套 - 重点

外层循环一次，内层的循环完成一轮(全部)。

循环不适合嵌套太深。

// 例子

```

// 第一层
// var i = 1;
// while (i <= 10) {
//     document.write('我是第一层循环' + i + '<br>');
//     i++;
//     // 第二层
//     var j = 1;
//     while (j <= 10) {
//         document.write('&emsp;&emsp;我是内层循环' + j + '<br>');
//         j++;
//     }
// }

```

do...while...循环

是一个和while循环类似的循环

while会先进行条件判断，满足就执行，不满足直接就不执行了

但是do while循环是，先不管条件，**先执行一回，然后在开始进行条件判断**

语法：**do { 要执行的代码 } while (限制条件)**

注意：如果限制条件一次都不满足，dowhile会执行一次，否则dowhile和while是一样的。

```

// var i = 11;
// do {
//     console.log(i);
//     i++;
// } while (i <= 10);

```

for循环

for循环和while循环在JavaScript里面可以相互使用

但是for循环的结构要紧凑一些

for循环的语法

for(初始值;限制条件;循环值的改变){ 循环体 }

while循环和for循环的初始值都可以是多条语句，中间用逗号分隔。

while循环和for循环的限制条件都可以是多条语句，中间用逗号分隔，注意起作用的是后面的限制条件

```

// for (var i = 1, j = 1; i <= 5, j <= 10; i++, j++) {
//     console.log(i); //输出的依然是1-10，多条限制条件语句，起作用的是后面的限制条件
// }

```

while循环和for循环的循环值的改变都可以是多条语句，逗号隔开

```

// 3.经典面试题
// for (var i = 1, j = 1; i <= 5, j <= 10; i++, j++) {

```

```
// var sum = i + j;
// }
// console.log(i);//11
// console.log(j);//11
// console.log(sum);//20 sum最后接收的值
```

for循环和while循环的对比

```
// 1.对比
// while循环
var i = 1, sum = 0; 第一步, 仅执行一次
while (i <= 100) { 第二步
    sum += i; 第三步
    i++; 第四步-->第二步-->开启下一轮
}
console.log(sum);//5050

// for循环
for (var i = 1, sum = 0; i <= 100; i++) {
    sum += i; 第三步
}
console.log(sum);//5050
```

The diagram illustrates the execution flow of while and for loops. For the while loop, the steps are: 1. Initialization (var i = 1, sum = 0), 2. Condition check (i <= 100), 3. Body execution (sum += i), and 4. Increment (i++) which loops back to the condition check. For the for loop, the steps are: 1. Initialization (var i = 1, sum = 0), 2. Condition check (i <= 100), 3. Body execution (sum += i), and 4. Increment (i++) which loops back to the condition check. Red arrows and boxes highlight these steps in the code snippets.

for循环的嵌套

外层循环一次，内层循环全部完成一轮

```
// document.write('<table border=1>');
// for (var i = 1; i <= 10; i++) {
//     document.write('<tr>');
//     for (var j = 1; j <= 10; j++) {
//         document.write('<td>11111</td>')
//     }
//     document.write('</tr>');
// }
// document.write('</table>');
```

break和continue关键字介绍

1.break终止循环

结束整个循环，执行循环以外的代码(起到优化作用，提升性能)

```
// for (var i = 1; i <= 10; i++) {
//     if (i === 5) {
//         break;
//     }
//     console.log(i);
// }
// console.log('循环之外的代码');
```

2.CONTINUE

结束本次循环,继续执行下一次循环,直到不满足条件

```
// for (var i = 1; i <= 10; i++) {  
//   if (i === 5) {  
//     continue;  
//   }  
//   console.log(i);  
// }  
// console.log('循环之外的代码');
```

总结:

如果for循环里面的break有效的话,循环的值一定满足限制条件的,

如果循环里面break无效,不会满足循环的限制条件。

通过循环的值是否满足限制条件来确定break是否有效

```
// for (var i = 1; i <= 10; i++) {  
//   if (i === 5) {  
//     break;  
//   }  
// }  
// if (i <= 10) { //满足,证明break一定有效  
//   console.log('有效');  
// } else { //break无效  
//   console.log('无效');  
// }
```

Debug(delete bug漏洞, 程序错误)

1.断点调试可以进行debug

在代码内设置断点(debugger),代码执行到断点处就会暂停,等待指示。

通过控制面板查看代码执行的过程

查看每一步,也可以直接查看每一轮循环



2.了解常见的错误类型

2.1.引用错误referenceError: 引用的某个变量或者标识符(自定义的名称)根本不存在。

```
// console.log(a);
```

2.2.语法错误`syntaxError` : vscode会提示错误, 一般是使用的关键字违背作者约定的规则。

```
// var 1abc = 456;
```

2.3.类型错误`TypeError` : 该语法不是这样使用, 或者说某些标识符不能这样使用

```
// 123();
```

2.4.范围错误`rangeError` : 使用的时候设置的某些值超过语法约定的范围

```
// var num = 3.141111;  
// console.log(num.toFixed(-2));
```