



工业和信息化“十三五”人才培养规划教材

1+X 证书制度 Web 前端开发系列丛书



# JavaScript +jQuery

## 交互式 Web 前端开发

黑马程序员 ● 编著

本书涵盖了 1+X 证书制度 Web 前端开发的相关专业知识，培养复合型技能人才。本套丛书将延续黑马程序员系列图书的品质，从知识点的选取、体例的设置、配套资源的制作等 3 个方面，进行升级和优化。



 中国工信出版集团

 人民邮电出版社  
POSTS & TELECOM PRESS

在编写代码时,外层的 for 控制行数  $i$ ,内层的 for 控制每行的星星个数  $j$ 。由于内层的星星个数  $j$  在每行的个数都不同,因此  $j$  的初始值会随  $i$  而改变。具体代码如下。

```
1 var str = '';
2 for (var i = 1; i <= 5; i++) {
3   for (var j = i; j <= 5; j++) { // j 的初始值为 i
4     str = str + '☆';
5   }
6   str += '\n';
7 }
8 console.log(str);
```



图 3-5 三角形图案

### 3. 生成九九乘法表

九九乘法表是双重 for 循环的一个非常经典的案例,效果如图 3-6 所示。

```
1×1=1
1×2=2 2×2=4
1×3=3 2×3=6 3×3=9
1×4=4 2×4=8 3×4=12 4×4=16
1×5=5 2×5=10 3×5=15 4×5=20 5×5=25
1×6=6 2×6=12 3×6=18 4×6=24 5×6=30 6×6=36
1×7=7 2×7=14 3×7=21 4×7=28 5×7=35 6×7=42 7×7=49
1×8=8 2×8=16 3×8=24 4×8=32 5×8=40 6×8=48 7×8=56 8×8=64
1×9=9 2×9=18 3×9=27 4×9=36 5×9=45 6×9=54 7×9=63 8×9=72 9×9=81
```

图 3-6 九九乘法表

在编写代码时,使用外层 for 控制行数  $i$ ,一共循环 9 次,使用内层 for 循环控制每行的公式  $j$ ,每一行的个数和行数一致,且  $j \leq i$ 。具体代码如下。

```
1 var str = '';
2 for (var i = 1; i <= 9; i++) {
3   for (var j = 1; j <= i; j++) {
4     str += j + 'x' + i + '=' + i * j + '\t';
5   }
6   str += '\n';
7 }
8 console.log(str);
```

#### 3.1.4 while 语句

while 语句可以在条件表达式为 true 的前提下,循环执行指定的一段代码,直到条件表达式为 false 时结束循环。具体语法结构如下。

```
while (条件表达式) {
  // 循环体
}
```

使用 while 语句输出 1 ~ 100 范围内的数字,具体代码如下。

```
1 var num = 1;
2 while (num <= 100) {
3   console.log(num);
```



```

4   num++;
5 }

```

从上述代码可以看出，while 语句的使用方法和 for 语句类似，同样可以利用计数器来控制循环的次数。需要注意的是，在循环体中需要对计数器的值进行更新，以防止出现死循环。为了直观地理解 while 的执行流程，下面我们通过图 3-7 进行演示。

使用 while 语句也可以完成各种各样的循环计算。例如，计算 1 ~ 100 之间的所有整数的和，具体代码如下。

```

1  var sum = 0;
2  var i = 1;
3  while(i <= 100) {
4    sum += i;
5    i++;
6  }
7  console.log(sum); // 输出结果：5050

```

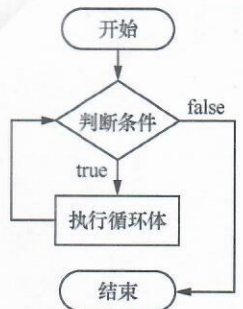


图 3-7 while 循环流程图

### 3.1.5 do...while 语句

do...while 语句的功能和 while 语句类似，其区别在于，do...while 会无条件地执行一次循环体中的代码，然后再判断条件，根据条件决定是否循环执行；而 while 是先判断条件，再根据条件决定是否执行循环体。具体语法结构如下。

```

do {
    // 循环体
} while (条件表达式);

```

使用 do...while 语句输出 1 ~ 100 范围内的数字，具体代码如下。

```

1  var num = 1;
2  do {
3    console.log(num);
4    num++;
5  } while (num <= 100);

```

在上述代码中，首先执行 do 后面“{}”中的循环体，然后再判断 while 后面的循环条件，当循环条件为 true 时，继续执行循环体，否则结束本次循环。do...while 循环语句的执行流程如图 3-8 所示。

使用 do...while 计算 1 ~ 100 之间的所有整数的和，具体代码如下。

```

1  var sum = 0;
2  var i = 1;
3  do {
4    sum += i;
5    i++;
6  } while(i <= 100)
7  console.log(sum); // 输出结果：5050

```

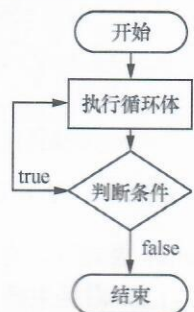


图 3-8 do...while 循环流程图