

第八课 语句

学习目录

- 语句分类
- 声明语句（变量声明、函数声明、标签声明）
- 分支控制语句（if 条件分支语句、switch 多重条件分支语句、三元运算语句）
- 循环控制语句（for 循环、for in 循环、while 循环、do while 循环）
- 流程异常控制语句（异常处理、return、break、continue、throw、debugger）

①语句分类

JavaScript 语句按照用途划分可以分为：声明、分支控制、循环控制、流程控制异常处理和其他各类语句。JavaScript 程序就是一系列的编程语句。JavaScript 语句由值、运算符、表达式、关键词和注释构成。

关键词	描述
break	终止 switch 或循环。
continue	跳出循环并在顶端开始。
debugger	停止执行 JavaScript，并调用调试函数（如果可用）。
do ... while	执行语句块，并在条件为真时重复代码块。
for	标记需被执行的语句块，只要条件为真。
function	声明函数。
if ... else	标记需被执行的语句块，根据某个条件。
return	退出函数。
switch	标记需被执行的语句块，根据不同的情况。

try ... catch	对语句块实现错误处理。
var	声明变量。

②声明语句

1.变量声明

var a = 0,b = 1,c,d; //声明四个变量，并部分赋值

2.函数声明

function fn() {console.log("声明函数");}

3.标签声明

{abc:123}

③分支控制语句

1.if 条件分支语句

条件语句用于基于不同条件执行不同的程序操作。

if (条件) {

如果条件为 true 时执行的代码

}

if (条件) {

条件为 true 时执行的代码

} else {

条件为 false 时执行的代码

}

```
if (条件 1) {  
    条件 1 为 true 时执行的代码  
} else if (条件 2) {  
    条件 1 为 false 而条件 2 为 true 时执行的代码  
} else {  
    条件 1 和条件 2 同时为 false 时执行的代码  
}
```

2.switch 多重条件分支语句

switch 语句来选择判断多个条件并根据条件结果来执行相关的代码，只要表达式的值等于 n 就会执行相关条件下的语句代码。

```
switch(表达式) {  
    case n:代码  
        break;  
    case n:代码  
        break;  
    default:  
        默认代码  
}
```

break 语句表示立即退出循环，进而继续执行 switch 语句之后的代码。

default 语句规定当不存在 case 匹配时，switch 语句默认要运行的代码。

```
switch (true) {  
  
    case a > 0 && a < 60:console.log('不及格');  
  
        break;  
  
    case a >= 60 && a < 90:console.log('良好');  
  
        break;  
  
    case a >= 90 && a <= 100:console.log('优秀');  
  
        break;  
  
    default:console.log('成绩统计出错');  
  
        break;  
  
}
```

3.三元运算语句

```
var a = 6;  
  
var test = a > 0 ? a : -a;  
  
alert(test);
```

③循环控制语句

1.for 循环

循环可多次执行代码块，如果需要运行代码多次，且每次使用不同的值，那么就要用到 for 循环。

for (语句 1; 语句 2; 语句 3) {要执行的代码块}

语句 1 在循环（代码块）开始之前执行。

语句 2 定义运行循环（代码块）的条件。

语句 3 会在循环（代码块）每次被执行后执行。

```
for (i = 0; i < 5; i++) {  
  
    console.log("数字是 " + i);  
  
}
```

2.for in 循环

for/in 语句遍历对象的属性，是 for 语句的一种特殊形式，对于数组来说，值是数组元素的下标；对于对象来说，值是对象的属性名或方法名。

```
var person = {name:"Bill", job:"Gates", age:62};  
  
for (var x in person) {  
  
    console.log(x + ' : ' + person[x]);  
  
}
```

3.while 循环

只要条件为 true，循环能够一直执行代码块。在使用 while 语句的时候忘了对条件中使用的变量进行改变，让循环能够根据改变之后的条件结束，那么循环永不会结束，这会导致浏览器崩溃，所以在使用 while 循环的时候要非常注意。

```
while (条件) {  
  
    要执行的代码块  
  
}
```



学习前端，最快的进步是持续！

```
var i = 0;

while (i < 10) {

    console.log("数字是 " + i);

    i++;

}
```

3.do while 循环

do/while 循环是 while 循环的另一种使用形式，在进行 while 条件判断之前就先要执行一次代码，之后只要条件为真就会重复循环。

```
do {

    要执行的代码块

}
```

```
while (条件);
```

```
var i = 0;

do {

    console.log("数字是" + i);

    i++;

}

while (i < 10);
```

④流程异常控制语句

1.try/catch/finally 与 throw 语句

try/catch/finally 语句用于处理代码中可能出现的错误信息。try 语句允许我们定义在执行时可能会产生错误的代码块。catch 语句允许我们定义当 try 代码块发生错误时，所执行的代码块。finally 语句在 try 和 catch 之后无论有无异常都会执行。JavaScript 语句 try 和 catch 是成对出现的。

```
var x = 100;  
  
try {  
  
    alert(x);//这里如果出现错误就会触发 catch 捕获错误并 throw 错误信息。  
  
}catch(error) {  
  
    throw("错误: " + error);  
  
}finally{  
  
    alert('执行完毕');  
  
}
```

2.return 语句

return 语句会终止函数的执行并返回函数的值，return 之后的函数代码不会执行。另外返回值是可选的，如果不指定返回值，默认会返回 undefined。

```
function getSum(x,y) {  
  
    return x + y;  
  
}
```

3.continue 语句

break 语句用于跳出循环，break 语句跳出循环后，会继续执行该循环之后的代码。

continue 用于跳过循环中的一个满足指定条件的迭代，然后继续循环中的下一个迭代。

```
for (i=0;i<=10;i++){  
  
    if (i==6) continue;  
  
    console.log( "数字是" + i);  
  
}
```

4.debugger 语句

debugger 语句用于停止执行 JavaScript，并调用调试函数，如果没有调试可用，

debugger 语句将无法工作。

```
var x = 10 * 6;  
  
var y = 100;  
  
debugger;  
  
var z = x + y;  
  
alert(z);
```

谢谢观看！

如果你自己难以解决的问题，请发送邮件到 xingxingclassroom@aliyun.com，并备注

星星课堂问题咨询这几个字+问题介绍为邮件标题，谢谢。

我是星星课堂老师：周小周