JavaScript 程序设计基础与实战

JavaScript 流程控制及应用

Learning and practice

演讲人：

时间:

目录/
DIRECTORY

1.JavaScript 的条件语句

JavaScript的条件语句会基于不同的条件执行不同的语句。编写程序代码时，经常需要根据不同的
决定执行不同的动作，可以在代码中使用条件语句来完成该任务。
在JavaScript中，可以使用以下条件语句。
（1）if语句：只有当指定条件为true时，该语句才会执行指定的代码。
（2）if…else…语句：当条件为true时执行指定的代码，当条件为false时执行其他代码。
（3）if…else if…else…语句：使用该语句选择多个代码块中的一个来执行。
（4）switch语句：使用该语句可选择多个代码块中的一个来执行。

1．if 语句

其语法格式如下。

```
if (条件)
{
// 当条件为true时执行的代码
}
```

这里的关键字为小写的if，如果使用大写字母（IF），则会产生JavaScript错误。

1.JavaScript 的条件语句

示例 3-1】demo0301.html
以下代码实现的功能如下：当时间早于20:00时，问候语显示为“Good day”。

```
let time=12 ;
if (time<20)
 {
    x="Good day";
 }
document.write( x ); // 输出结果为"Good day”
```

该语句不包含else，只有在指定条件为true时才会执行指定的代码。

1.JavaScript 的条件语句

2. if…else…语句

其语法格式如下。

```
if (条件)
{
// 当条件为true时执行的代码
}
else
{
// 当条件为false时执行的代码
}
```

【示例 3-2】demo0302.html

以下代码实现的功能如下：当时间早于20:00时，显示问候语“Good day”，否则显示问候语“Good evening”。

```
let time=21 ;
if (time<20)
{
x="Good day";
}
else
{
x="Good evening";
}
document.write( x ); //输出结果为"Good evening"
```

1.JavaScript 的条件语句

3. if…else if…else…语句

其语法格式如下。

```
if (条件 1)
{
// 当条件1为true时执行的代码
}
else if (条件2)
{
// 当条件1为false且条件2为true时执行的代码
}
else
{
// 当条件1和条件2都不为true 时执行的代码
}
```

【示例 3-3】demo0303.html

以下代码实现的功能如下：当时间早于10:00时，显示问候语“Good morning”；当时间晚于或等于10:00且早于20:00时，显示问候语“Good day”；否则，显示问候语“Good evening”。

```
let time=8 ;
if (time<10)
{
x="Good morning";
}
else if (time<20)
{
x="Good day";
}
else
{
x="Good evening";
}
document.write( x ); //输出结果为"Good morning"
```

1.JavaScript 的条件语句

4. switch 语句

switch语句会基于不同条件执行不同动作。可以使用switch语句选择多个代码块中的一个来执行。其语法格式如下。

```
switch(表达式)
{
case m:
// 执行代码块1
break;
case n:
// 执行代码块2
break;
default:
// 表达式的值与m、n不同时执行的代码
}
```

实战演练

首先计算一次switch对应表达式（通常为一个变量）的值，随后表达式的值会与结构中每个case后面的值进行比较。如果存在匹配项，则与该case关联的代码块会被执行。使用break语句可阻止代码自动向下一个case运行，跳出switch语句。

switch语句中的表达式不一定是条件表达式，可以是普通的表达式，其值可以是数字、字符串或布尔值。执行switch语句时，首先将表达式的值与一个数据进行比较，当表达式的值与所列数据相等时，执行对应的代码块。如果表达式的值与所有列出的数据都不相等，则会执行 default 后的代码块；如果没有default关键字，则会跳出switch语句，执行switch语句后面的代码。

【示例 3-4】demo0304.html 以下代码实现的功能是显示今天是星期几。

```
var day=new Date().getDay();
switch(day)
{case 0:
x="星期日";
break;
case 1:
x="星期一";
break;
case 2:
x="星期二";
break;
case 3:
x="星期三";
break;
case 4:
x="星期四";
break;
case 5:
x="星期五";
break;
case 6:
x="星期六";
break;}  document.write("今天是"+x);
```

1.JavaScript 的条件语句

有时需要用不同的case语句来执行相同的代码块。
【示例 3-5】demo0305.html
代码如下：

```
switch (new Date().getDay()) {
case 4:
case 5:
info = "周末快到了。";
break;
case 0:
case 6:
info = "今天是周末。";
break;
default:
info = "期待周末！";
}
document.write(info);
```

在本例中，case 4和case 5分享相同的代码块，而case 0和case 6分享另一个代码块。
switch语句使用全等（===），即操作数的值和类型都必须相同，全等的结果才能为true。

1.JavaScript 的条件语句

【示例 3-6】demo0306.html

代码如下：

```
var x = "0";
switch (x) {
      case 0:
            text = "Off";
            break;
      case 1:
            text = "On";
            break;
      default:
            text = "No value found";
}
document.write(text); // 输出的值为"No value found"
```

以上代码中，x与每个case后面的值都不匹配。

1.JavaScript 的条件语句

（1）break关键字。在执行switch语句时，如果遇到break关键字，则程序会跳出switch代码块，此举将中断代码块中其他代码的执行以及后续case比较。break能够节省大量代码执行时间，因为它会“忽略”switch语句中其他代码的执行。
switch语句中的最后一个case语句不必中断，执行至该语句处时，switch语句会自然结束。
（2）default关键字。使用default关键字可指定switch语句中不存在case匹配项时所执行的代码。default在switch语句中是可选的，并非必不可少，但通常会把它放在switch语句的最后来执行兜底操作。
【示例 3-7】demo0307.html 以下代码实现的功能如下：如果今天不是星期六或星期日，则会输出默认信息。

```
var day=new Date().getDay();
switch (day)
{
case 6:
x="今天是星期六";
break;
case 0:
x="今天是星期日";
break;
default:
x="期待周末！";
}
document.write(x);
```

2.JavaScript 的 循环语句

如果需要多次执行相同的代码，并且每次需要的值都不同，那么使用循环是很方便的，循环可以将
代码块反复执行指定的次数。
数组num的定义如下。

```
var num=[ 0 , 1 , 2 , 3 , 4 , 5 ];
```

以下代码可以输出数组中元素的值。

```
document.write(num[0] + "<br>");
document.write(num[1] + "<br>");
document.write(num[2] + "<br>");
document.write(num[3] + "<br>");
document.write(num[4] + "<br>");
document.write(num[5] + "<br>");
```

循环语句的代码通常写成如下形式。

```
for ( var i=0 ; i<num.length ; i++ )
{
document.write(num[i] + "<br>");
}
```

2.JavaScript 的 循环语句

JavaScript支持不同类型的循环，包括以下几种循环。
（1）while循环：当指定的条件为true时执行指定的代码块。
（2）do…while循环：当指定的条件为true时执行指定的代码块。
（3）for循环：多次遍历代码块，并且循环的次数固定。
（4）for…in循环：循环遍历对象的属性。
（5）for…of循环：循环遍历可迭代对象的值。

1. while 循环

while循环会在指定条件为true时循环执行代码块，只要指定条件始终为true，while循环就可以一直执行代码块。
其语法格式如下。

```
while (条件)
{
// 需要执行的代码块
}
```

2.JavaScript 的 循环语句

【示例 3-8】demo0308.html
只要变量i小于5，本例中的循环就继续执行。
代码如下：

```
var x="" ;
var i=0;
while (i<5)
{
x=x + "The number is " + i + "<br>";
i++;
}
```

如果忘记增加条件中所用变量的值，即没有i++语句，则该循环永远不会结束。这可能会导致浏览器崩溃。

2.JavaScript 的 循环语句

2. do…while 循环

do...while循环是while循环的变体，该循环在检查条件是否为true之前会执行一次代码块，如果条件为true，则重复执行代码块。其语法格式如下。

```
do
{
// 需要执行的代码块
}
while (条件) ;
```

【示例 3-9】demo0309.html

代码如下：

```
var x="" ;
var i=0 ;
do
{
x=x + "The number is " + i + "<br>";
i++;
}
while (i<5);
```

以上示例代码使用do…while循环，即使条件为false，该循环也至少会执行一次。

2.JavaScript 的 循环语句

3．for 循环

其语法格式如下。

```
for(表达式1；表达式2；表达式3)
{
// 需要执行的代码块
}
```

表达式1：在循环开始前执行。

表达式2：执行循环的条件。

表达式3：在循环被执行之后执行。

其执行过程如下：执行表达式1，完成初始化；判断表达式2的值是否为true，如果为true，则执行循环代码块，否则退出循环；执行循环代码块之后，执行表达式3；重新判断表达式2的值，若其值
为true，再次重复执行循环代码块，如此循环。

2.JavaScript 的 循环语句

【示例 3-10】demo0310.html

代码如下：

```
var x="";
for (var i=0 ; i<5 ; i++)
{
x=x + "The number is " + i + "<br>";
}
```

从上述代码可以看出以下信息。

表达式1在循环开始之前定义了变量：var i=0。

表达式2定义了循环执行的条件：i必须小于5。

表达式3在代码块已被执行一次后使i增加1：i++。

（1）表达式1。通常使用表达式1来初始化for循环中所用的变量（var i=0），也可以在表达式1中初始化由逗号分隔的多个变量。