

目录

目录

问题

作业讲评

学习目标

数组

为什么有数组？

什么是数组？

为什么要使用数组？

数组的相关概念

什么是数组元素

数组元素的类型

数组下标(索引)

如何访问数组里面的某一个元素

数组的长度

定义数组 重点

使用[]来定义数组 常用

使用new Array()方法来定义数组 用的不多

数组的遍历 重点

什么是数组的遍历？

使用for循环来遍历数组 用的多

多维数组

一维数组

多维数组

如何访问二维数组的里面的数组元素呢？

遍历二维数组 重点难点

新增数组元素

今日总结

今日作业

问题

1. 问问题的时候代码截图
2. 把变量的值输出看看，测试下是不是自己想要的
3. 做作业前，建议整理下笔记，看下上课的源代码

作业讲评

2. 利用javascript打印下图九九乘法表，注意改字体颜色和字体大小

九九乘法表

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

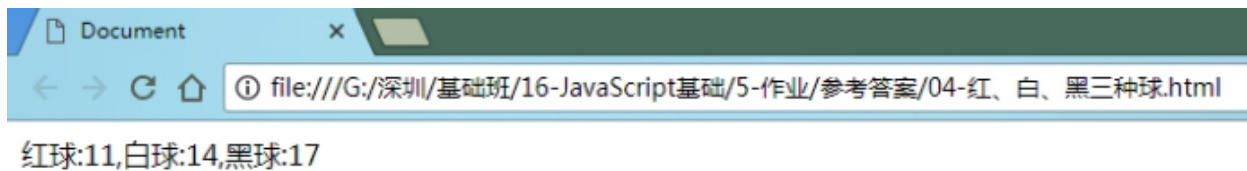
```

<script type="text/javascript">
// 九九乘法表
document.write("<h1 style='font-family:宋体;text-align:center'>九九乘法表</h1>");

var str = "<table border='1' width='600' rules='all' align='center'>";
for(var i=9; i>=1; i--){
    str += "<tr>";
    for(var j=1; j<=i; j++){
        str += "<td><font color='blue' size='5'>"+j+"</font>&times;<font color='red' size='5'>"+i+"</font>="+i*j+"</td>";
    }
    str += "</tr>";
}
str += "</table>";
document.write( str );
</script>

```

4. 有红、白、黑三种球若干个，其中红、白球共25个，白、黑球共31个，红、黑球共28个，求这三种球各多少个？



```

<script type="text/javascript">
// 4. 有红、白、黑三种球若干个，其中红、白球共25个，白、黑球共31个，红、黑球共28个，求这三种球各多少个？
// red + white = 25;
// white + black = 31;
// red + black = 28

// 假设红球
for(var red = 0; red <= 25; red++){
    var white = 25 - red; // 白球
    var black = 31 - white; // 黑球

    if( red + black == 28){
        document.write("红球"+red+"个,白球"+white+"个,黑球"+black+"个");
    }
}
</script>

```

5. 使用for循环输出以下图形

```
999999999
888888888
7777777
666666
55555
4444
333
22
1
22
333
4444
55555
666666
7777777
888888888
999999999
```

```
// 最后我们拼接上下部分
for(var i = 9;i>=1; i--){// 控制行数
    for(var j=1;j <= i;j++){// 控制数字的个数
        document.write(i);
    }
    document.write("<br/>");
}

for(var i = 2;i<= 9; i++){// 控制行数
    for(var j=1;j <= i;j++){// 控制数字的个数
        document.write(i);
    }
    document.write("<br/>");
}
```

6、接收用户输入的用户名和密码，若用户名为 “admin” ,且密码为 “123456” ,则提示用户登录成功! 否则，让用户一直输入提示(利用while循环或者do while 循环)

```

<script type="text/javascript">
    // 6、接收用户输入的用户名和密码，若用户名为“admin”，且密码为“123456”，
    则提示用户登录成功！ 否则，让用户一直输入 提示(利用while循环或者do
    while 循环)

    // 死循环,就是一直满足条件,循环体一直执行,不会结束
    while(true){
        var username = window.prompt("请输入用户名");
        var userpass = window.prompt("请输入密码");
        if(username == "admin" && userpass == "123456"){
            alert("登录成功");
            // 循环中断关键字break,跳出整个循环
            break;
        }
    }
}
</script>

```

7、简易ATM

题目描述：

- 里面现存有 100 块钱。
- 如果存钱，就用输入钱数加上先存的钱数, 之后弹出显示余额提示框
- 如果取钱，就减去取的钱数，之后弹出显示余额提示框
- 如果显示余额，就输出余额
- 如果退出，弹出退出信息提示框

操作界面如下： 提示window.prompt中的内容如果需要换行需要使用\n

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  <meta charset="UTF-8">
5  <title>Document</title>
6  <script type="text/javascript">
7  /*7、简易ATM
8  题目描述：
9  - 里面现存有 100 块钱。
10 - 1.如果存钱，就用输入钱数加上先存的钱数，之后弹出显示余额提示框
11 - 2.如果取钱，就减去取的钱数，之后弹出显示余额提示框
12 - 3.如果显示余额，就输出余额
13 - 4.如果退出，弹出退出信息提示框
14 操作界面如下：提示window.prompt中的内容如果需要换行需要使用\n*/
15
16 // 里面有100块钱

```

```
17  var money = 100;
18
19  // 用户选择的操作
20  var select = window.prompt("请输入您要操作\n1.存钱\n2.取钱\n3.显示余额\n4.退出");
21  if(select == null){
22      alert("取消了操作");
23  }else if( select >= 1 && select <= 4){// 如果是正确操作,又要分情况
24      if(select == 1){
25          // window.prompt不管你输入的是什么,都会变成字符串
26          var add = Number( window.prompt("请输入要存入的金额") );
27          if(add <= 0){
28              alert("请输入正数");
29          }else{
30              money = money + add;
31              // 大家出现错误以后,可以往前找,一般是前面的错误对后面造成了影响;我们可以输出变量的值,看看是不是我们想要的值
32              alert("存入成功,当前余额为:"+money);
33          }
34      }else if(select == 2){
35          var jian = Number( window.prompt("请输入取款金额") );
36          if(jian <= 0){
37              alert("请输入正数");
38          }else if(money < jian){
39              alert("余额不足,当前存款为:"+money);
40          }else{
41              money = money - jian;
42              alert("取款成功,当前余额为:"+money);
43          }
44      }else if(select == 3){
45          alert("当前余额为"+money);
46      }else if(select == 4){
47          alert("退出成功,欢迎下次使用~");
48      }
49      }else{
50          alert("操作有误,去输入1~4之间的数");
51      }
52  </script>
53 </head>
54 <body>
55
```

```
56 </body>
```

```
57 </html>
```

8、菱形表格

									9×9=81	1×1=1									
								8×8=64	2×2=4										
							7×7=49	3×3=9	3×2=6	3×3=9									
						6×6=36	4×4=16	4×1=4	4×2=8	4×3=12	4×4=16								
					5×5=25	5×4=20	5×3=15	5×2=10	5×3=15	5×4=20	5×5=25								
				4×4=16	4×3=12	4×2=8	4×1=4	4×2=8	4×3=12	4×4=16	4×5=20	4×6=24	4×7=28	4×8=32	4×9=36				
			3×3=9	3×2=6	3×1=3	3×2=6	3×3=9	3×4=12	3×5=15	3×6=18	3×7=21	3×8=24	3×9=27	3×10=30	3×11=33	3×12=36			
		2×2=4	2×1=2	2×1=2	2×2=4	2×3=6	2×4=8	2×5=10	2×6=12	2×7=14	2×8=16	2×9=18	2×10=20	2×11=22	2×12=24	2×13=26	2×14=28	2×15=30	2×16=32
	1×1=1	1×2=2	1×3=3	1×4=4	1×5=5	1×6=6	1×7=7	1×8=8	1×9=9	1×10=10	1×11=11	1×12=12	1×13=13	1×14=14	1×15=15	1×16=16	1×17=17	1×18=18	1×19=19
1×9=9	1×8=8	1×7=7	1×6=6	1×5=5	1×4=4	1×3=3	1×2=2	1×1=1	9×1=9	9×2=18	9×3=27	9×4=36	9×5=45	9×6=54	9×7=63	9×8=72	9×9=81		
	2×9=18	2×8=16	2×7=14	2×6=12	2×5=10	2×4=8	2×3=6	2×2=4	8×1=8	8×2=16	8×3=24	8×4=32	8×5=40	8×6=48	8×7=56	8×8=64			
		3×9=27	3×8=24	3×7=21	3×6=18	3×5=15	3×4=12	3×3=9	7×1=7	7×2=14	7×3=21	7×4=28	7×5=35	7×6=42	7×7=49				
			4×9=36	4×8=32	4×7=28	4×6=24	4×5=20	4×4=16	6×1=6	6×2=12	6×3=18	6×4=24	6×5=30	6×6=36					
				5×9=45	5×8=40	5×7=35	5×6=30	5×5=25	5×1=5	5×2=10	5×3=15	5×4=20	5×5=25						
					6×9=54	6×8=48	6×7=42	6×6=36	4×1=4	4×2=8	4×3=12	4×4=16							
						7×9=63	7×8=56	7×7=49	3×1=3	3×2=6	3×3=9								
							8×9=72	8×8=64	2×1=2	2×2=4									
								9×9=81	1×1=1										

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <title>Document</title>
6 <style>
7 table{
8 float:left;
9 }
10 </style>
11 <script type="text/javascript">
12 var str = "<table border='1' width='600'>";
13 for(var i=9; i>=1; i--){
14 str += "<tr>";
15
16 /*i的变量 i-1转换 空白单元格的数目
17 9 8
18 8 7
19 7 6
20 6 5
21 5 4
22 ...
23 1 0*/
24
25 for(var k=i-1; k>=1 ; k--){
26 str += "<td> </td>";
```

```
27  }
28
29  for(var j=9;j>=i;j--){
30    str += "<td bgcolor='green'>" + i + "x" + j + "=" + i * j + "</td>";
31  }
32
33  str += "</tr>";
34
35
36  }
37  str += "</table>";
38  document.write( str );
39
40
41  var str = "<table border='1' width='600'>";
42  for(var i=1; i<=9; i++){
43    str += "<tr>";
44
45    for(var j=1;j<=i;j++){
46      str += "<td bgcolor='skyblue'>" + i + "x" + j + "=" + i * j + "</td>";
47    }
48
49    // 额外再加了td
50    for(var k = 9-i ; k>=1 ; k--){
51      str += "<td> </td>";
52    }
53
54    str += "</tr>";
55  }
56  str += "</table>";
57  document.write( str );
58
59
60
61  var str = "<table border='1' width='600'>";
62  for(var i=1; i<=9; i++){
63    str += "<tr>";
64
65    for(var k=i-1; k>=1 ; k--){
66      str += "<td> </td>";
67    }
```



```

68
69  for(var j=9;j>=i;j--){
70  str += "<td bgcolor='orange'>" + i + "x" + j + "=" + i * j + "</td>";
71  }
72
73  str += "</tr>";
74
75
76  }
77  str += "</table>";
78  document.write( str );
79
80
81  var str = "<table border='1' width='600'>";
82  for(var i=9; i>=1; i--){
83  str += "<tr>";
84
85  for(var j=1;j<=i;j++){
86  str += "<td bgcolor='pink'>" + i + "x" + j + "=" + i * j + "</td>";
87  }
88
89  // 额外再加了td
90  for(var k = 9-i ; k>=1 ; k--){
91  str += "<td> </td>";
92  }
93  str += "</tr>";
94  }
95  str += "</table>";
96  document.write( str );
97
98  </script>
99 </head>
100 <body>
101
102 </body>
103 </html>

```

学习目标

- 能够了解什么是数组
- 能够定义数组,定义数组的方法有哪些

- 能够获取数组的长度
- 能够访问数组中的某个元素,通过数组的下标访问元素
- 能够遍历数组,就是访问数组中每一个元素
- 能够新增数组元素
- 能够定义二维数组
- 能够遍历二维数组

数组

为什么有数组？

之前学习的数据类型，只能存储一个值。我们想存储班级中所有学生的姓名，此时该如何存储？

- 数组的作用： 可以把一组相关的数据一起存放，并提供方便的访问(获取) 方式。

```
1 //基本变量
2 //var name1 = 1;
3 //var name2 = "lisi";
4 //var name3 = "wangwu"
5
6 //数组的写法
7 var studentNames = ["zhangsan" ,"lisi" ,"wangwu"];
```

什么是数组？

数组是一组数据**有序的**集合。数组它是属于复合数据类型。至少可以存储一个值。

为什么要使用数组？

因为在我们工作中 有很多数据是有关联的 我们要表示的时候想把这些数据用一个“东西”来存储，这个时候就可以用到数组！

所谓数组，就是将多个元素（通常是同一类型）按一定顺序排列放到一个集合中，那么这个集合我们就称之为数组。



数组就类似一个储物柜，储物柜有按顺序排好的小储物箱，每个小储物箱都有一个序号，可以放置很多东西，同时我们也可以按照序号取东西出来；

数组的相关概念

什么是数组元素

指数组中的**每一个数据**！我们就把它们称之为数组元素

数组元素的类型

数组中可以存放**任意类型的数据,可以是字符串,可以是数值,可以是布尔,undefined,null,array...**

数组下标(索引)

每一个数组元素都对应着一个数组下标 数组的**下标是从0开始** **第一个数组元素的下标是0**，第二个数组元素的下标为1 第三个数组元素的下标为2 依此类推！**下标的主要作用就是为了方便找到数组中的元素！**

如何访问数组里面的某一个元素

格式：

```
1 数组变量名[下标]
```

举例：

```
1 // 通过[]定义数组,里面放的内容就是数组元素
2 var studentNames = ["zhangesan","张飞","张良","张学友"];
3 // 通过数组下标访问数组元素 下标从0开始 0就代表第一个数组元素
4 console.log(studentNames[0]); //zhangesan
5 console.log(studentNames[3]); //张学友
```

```
6 console.log(studentNames[4]); //如果访问下标对应数组元素不存在,就会得到一个undefined
```

数组的长度

数组的长度：数组中元素的**总个数**

其实数组的变量名是一个数组对象

如何去计算数组的长度：

- 1 数组变量名.length
- 2 说明:在JS中 数组对象下面有一个length的属性 这个属性可以得到数组的长度 得到是一个整数结果

总结:只算**逗号前面的元素**才会算进长度里面

```
1 // 通过[]定义数组,里面放的内容就是数组元素
2 var studentNames = ["zhangesan","张飞","张良","张学友"]; //长度4
3 // var studentNames = ["zhangesan","张飞"]; //长度2
4 // var studentNames = ["zhangesan","张飞",]; //长度2
5 // var studentNames = [,"zhangesan","张飞"]; //长度3
6 // var studentNames = [ ,,"zhangesan","张飞"]; //长度4
7 // var studentNames = [ ,,"zhangesan","张飞",,]; //长度5
8 // 获取数组的长度 语法:数组变量名.length
9 var arr_length = studentNames.length;
10 document.write( arr_length );
11
12 // 总结一下: 只算逗号前面的元素才会算进长度里面
```

数组的长度与最大下标之间有什么关系？

最大下标=数组的长度-1

```
1 // 通过[]定义数组,里面放的内容就是数组元素
2 var studentNames = ["zhangesan","张飞","张良","张学友"];
3 var arr_length = studentNames.length;
4 document.write( "数组长度为" +arr_length+"<hr/>" ); //数组长度是4
5 document.write(studentNames[0]+"<br/>"); //第一个元素 "zhangesan"
6 document.write(studentNames[1]+"<br/>"); //第二个元素 "张飞"
7 document.write(studentNames[2]+"<br/>"); //第三个元素 "张良"
8 document.write(studentNames[3]+"<br/>"); //第四个元素 "张学友"
```

```
9 document.write(studentNames[4]+"<br/>");//第五个元素 undefined
10
11 // 数组的最大下标就是数组的长度-1
```

举例:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <script type="text/javascript">
7     // 定义数组
8     var arr = [10,20,30];
9     // 数组的元素指数组中每个值,比如arr数组中的10,20,30都可以称之为数组元素
10    // 数组元素的类型可以是任何数据类型,字符串,数值,布尔,null,undefined,数组,对象
11
12    // 如果访问数组中某个数组元素,通过数组下标 语法:数组变量名[下标]
13    // 数组下标是从0开始的递增数组
14    console.log( arr );
15
16    // 如果我们想输出arr中的30
17    console.log( arr[2] );// 30
18
19    // 如果我们下标对应的数组元素在数组中不存在,会得到undefined
20    console.log( arr[3] );// undefined
21
22    // 数组的长度,指的就是数组中数组元素的"总个数" 语法:数组变量名.length
23    console.log( arr.length );// 数组长度返回的是一个数字
24
25    // 数组最大的下标是什么? 语法:最大的下标 = 数组的长度-1
26  </script>
27 </head>
28 <body>
29
30 </body>
31 </html>
```

定义数组 重点

使用[]来定义数组 常用

第一种方式：先定义一个空数组 再往这个空数组中添加数组元素

```
1 // 先定义一个空数组 再往这个空数组中添加数组元素
2 var arr = [];//定义了一个空数组
3 console.log(arr);//输出数组
4
5 // 添加元素,我们是使用 数组变量名[下标]=值
6
7 // 添加第一个元素
8 arr[0] = 10;
9 // 添加第二个元素
10 arr[1] = 20;
11
12 console.log(arr);//输出数组
13
14 // 如果这个元素已经存在,就会覆盖前面的值
15 arr[1] = 22;
16
17 console.log(arr);//输出数组
```

第二种方式：在定义数组时候直接添加数组元素 用的比较多

```
1 // 定义数组的时候直接添加数组元素;
2 var arr = [10,20,30];
3 console.log(arr);
```

举例:

```
1
2 // 数组(array)
3 // 定义数组的方式有多种
4 // 第一种:使用[]定义数组
5 // 先定义空数组,后赋值,语法如下
6 /*var 数组变量名 = [];
7 数组变量名[下标0] = 值1;
8 数组变量名[下标1] = 值2;
9 数组变量名[下标2] = 值3;
10 ...*/
11
```

```

12 // 数组的下标是从0开始的递增数字
13 var arr1 = [];
14 arr1[0] = 10;
15 arr1[1] = 20;
16 arr1[2] = 30;
17 arr1[3] = null;
18 arr1[4] = true;
19 arr1[5] = undefined;
20 arr1[6] = false;
21 console.log( arr1 );// 使用console.log可以更加直观的查看数组
22
23 // 注意:如果使用document.write或者alert输出数组,js会帮我们自动转换string类型的字符串,再输出;其实就是把每个数组元素以"逗号"连接,拼接成一个大字符串
24 document.write( arr1 );
25 alert( arr1 );
26
27 // 其他类型转成字符串类型方法有哪些?
28 // 变量名.toString(),
29 console.log( arr1.toString() );
30 // 加引号
31
32 // String(变量名)
33 console.log( String(arr1) );
34
35 // 和字符串连接
36 console.log( arr1 + "" );
37 console.log("");
38
39
40 // 定义数组,并给数组元素赋值,语法如下,推荐使用这种
41 // var 数组变量名 = [值1,值2,值3...]
42 var arr2 = [25,23,10,true,false,null,undefined];
43 console.log( arr2 );

```

使用new Array()方法来定义数组 用的不多

Array 对象

Array 对象用于在单个的变量中存储多个值。

创建 Array 对象的语法:

```
new Array();  
new Array(size);  
new Array(element0, element0, ..., elementn);
```

定义一个空数组

定义一个指定长度的数组

定义一个有元素元素的数组

第一种方式: 创建一个空数组,然后再给数组中的元素赋值

```
1 // 通过new Array()定义了一个空数组  
2 var arr = new Array();  
3 console.log( arr );  
4 arr[0] = "张三";  
5 arr[1] = "李四";  
6 console.log( arr );
```

第二种方式: 创建一个数组 要指定数组的长度 但不影响其添加多于这个长度的元素个数

```
1 比如new Array(3)表示数组长度为3,但是可以添加超过长度3个的元素个数  
2 var arr = new Array();//定义长度为0的空数组  
3 console.log(arr.length);//长度为0  
4 console.log(arr);
```

```
1 // new Array(数组长度)  
2 var arr2 = new Array(5);//定义一个长度为5个空数组  
3 console.log(arr2.length);//长度为5  
4 console.log(arr2);  
5 document.write(arr2);  
6 document.write("<hr/>");  
7 arr2[2] = "关羽";
```

```
1 //创建一个数组 要指定数组的长度 但不影响其添加多于这个长度的元素个数  
2 arr2[10] = "夏侯惇";  
3 document.write(arr2);  
4 第三种方式: 创建一个数组 并指定数组元素 用的比较多  
5 // new Array创建数组并指定元素  
6 var arr = new Array("张三","男",23,"java从入门到放弃");  
7 console.log(arr);  
8 console.log(arr[3]);
```


举例:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <script type="text/javascript">
7     // 使用new Array()方法来定义数组
8     // 第一种 使用new Array()定义一个空数组
9     var arr1 = new Array();
10    arr1[0] = 10;
11    arr1[1] = 20;
12    console.log( arr1 );
13    console.log("");
14
15    // 第二种 注意: 使用new Array(长度)定义一个指定长度的"空"数组
16    var arr2 = new Array(10); // 这个是代表长度为10, 不是第一个数组元素的值为10
17    console.log( arr2 );
18    console.log("");
19
20    // 第三种 使用new Array(值1,值2,值3...)定义一个有数组元素的数组
21    var arr3 = new Array(10,20,30);
22    console.log( arr3 );
23
24    // 老师是这样理解的 new Array( 等同于 [
25    // ) 等同于 ]
26    // 比如:
27    var arr4 = [50,60,70];
28    var arr5 = new Array(50,60,70);
29    console.log(arr4);
30    console.log(arr5);
31
32  </script>
33 </head>
34 <body>
35
36 </body>
37 </html>
```

注意:

在工作中一般都是使用**[]来创建数组**基本上不会使用new Array()方法来创建

练习: 使用**数组**保存下列信息,并打印如下效果图

周更生的基本信息

姓名：周更生
性别：男
年龄：30
学历：大专
基本工资：2000
奖金：500
实发工资：2500
婚否：未婚



```
4 <meta charset="UTF-8">
5 <title>Document</title>
6 <script type="text/javascript">
7     var arr = ["周更生","男",30,"大专",2000,500,2500,"未婚"];
8     var str = "<h1>" + arr[0] + "的基本信息</h1>";
9         str += "姓名:" + arr[0] + "<br>";
10        str += "性别:" + arr[1] + "<br>";
11        str += "年龄:" + arr[2] + "<br>";
12        str += "学历:" + arr[3] + "<br>";
13        str += "基本工资:" + arr[4] + "<br>";
14        str += "奖金:" + arr[5] + "<br>";
15        str += "实发工资:" + arr[6] + "<br>";
16        str += "婚否:" + arr[7];
17
18        document.write( str );
19    </script>
20 </head>
21 <body>
22
23 </body>
24 </html>
```

Document

文件 | F:/GZ20H5直播6班/Javascript基础/0-

周更生的基本信息

姓名:周更生
性别:男
年龄:30
学历:大专
基本工资:2000
奖金:500
实发工资:2500
婚否:未婚

数组的遍历 重点

什么是数组的遍历？

- 数组的遍历就是**操作数组中的每一个数组元素**
- 也可以理解数组遍历就是把**每个元素从头到尾都访问一次**（类似我们**每天早上中午晚上学生的点名**）

使用for循环来遍历数组 用的多

因为数组的**下标是连续的**，数组的**下标是从0开始**。

我们也可以得到**数组的长度**。

格式：

```
1 for(var i=0;i<数组的长度;i++){
2   数组变量名[i]
3 }
```

注意：其中条件表达式的写法有两种

```
1 第一种：i<数组的长度
2 第二种：i<=数组的长度-1
3
4 其中第一种较为常用
```

举例：使用for遍历一维数组中所有的元素

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <script type="text/javascript">
7     // 一维数组指的是数组元素的数据类型不是数组;简单理解数组里面没有数组
8     // 比如 var arr = [10,true,false,null,undefind];
9
10    // 二维数组,数组元素又是一个数组;简单理解数组里面又有数组
11    /*比如
12    var arr = [
13      ["张三",23,"男"],
14      ["李四",24,"女"],
15    ]
16    */
```

```
17
18 // 那么什么叫做数组的遍历呢? 遍历就是把数组中每个元素都访问一次,类似我们每天早
    上让同学签到一样
19
20
21 // 定义数组
22 var arr = [10,20,30,40,50,60,70,80];
23 // 访问数组中每个元素,这就叫数组的遍历
24 // console.log( arr[0] );
25 // console.log( arr[1] );
26 // console.log( arr[2] );
27 // console.log( arr[3] );
28 // console.log( arr[4] );
29 // console.log( arr[5] );
30 // console.log( arr[6] );
31 // console.log( arr[7] );
32
33 // 我们说过,只要是重复有规律的代码,咱们就可以使用循环来做
34 /*for(var i=0; i<= 7; i++){
35     console.log( arr[i] );
36 }*/
37
38 // 以上我们会发现7刚好就是最大的数组下标
39 // 最大的数组下标 = 数组的长度 - 1
40 // 数组的长度 数组变量名.length
41 // console.log( arr.length );// 8
42
43 /*for(var i=0; i<= arr.length - 1; i++){
44     console.log( arr[i] );
45 }*/
46
47
48 // 接着,我们又发现了 循环条件i<=7 跟i<8效果是一样的;刚刚好,数组的长度就是8
49 for(var i=0; i< arr.length ; i++){
50     console.log( arr[i] );
51 }
52
53
54 /*最后,我们总结一下,一维数组遍历的语法 要求非常熟练,因为我们要从数组元素筛选东
    西,就必须对数组元素每个都访问访问一遍,比如找数组中最大的元素
55
```

```

56  for( var i= 0; i < 数组长度;i++){
57  console.log( 数组变量名[下标] );
58  }*/
59
60
61
62  </script>
63  </head>
64  <body>
65
66  </body>
67  </html>

```

练习:

要求: 求一组数组 [2,6,1,77,52,25,7] 的最大值

先思考如何 比较2个数最大值?

```

<script type="text/javascript">
    // 求一组数组 [2,6,1,77,52,25,7] 的最大值

    // 定义数组
    var arr = [2,6,1,77,52,25,7];

    // 求最大值或者最小值,我们都会先假设一个变量保存最大值或者最小值,
    我们建议假设数组中某个元素的值为最大值或者最小值
    var maxValue = arr[0];

    // 遍历数组
    for(var i=1;i<arr.length;i++){
        // 如何比较两个值之间大的值
        if(maxValue < arr[i] ){//如果 我们假设的最大值小于数组元素的值

            // 我们就数组元素的值赋给我们假设的最大值变量,让它成为新的最大值
            maxValue = arr[i];
        }
    }
    console.log( maxValue );
</script>

```

练习:

求数组 [2,6,1,7,4]的所有元素的和 以及 所有元素平均值

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <script type="text/javascript">
7     // 求数组 [2,6,1,7,4]的所有元素的和 以及 所有元素平均值
8     var arr = [2,6,1,7,4];
9     // 定义一个变量保存和
10    var sum = 0;
11    for(var i=0;i<arr.length;i++){
12      sum += arr[i];
13    }
14
15    // 循环外面,代表循环已经结束了
16    console.log( "和为:" + sum );
17
18    // 平均值 = 和/个数
19    // 数组元素的总个数就是数组长度
20
21    console.log( "平均值为:" + sum/arr.length );
22
23  </script>

```

练习:

将数组里面的数据转换为字符串redgreenbluepink

```
var arr = ['red', 'green', 'blue', 'pink'];
```

```

11-练习-将数组里面的数据转换为字符串.html x
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <script type="text/javascript">
7     // 将数组里面的数据转换为字符串redgreenbluepink
8     var arr = ['red', 'green', 'blue', 'pink'];
9     // 定义一个空字符串
10    var str = "";
11    for(var i=0; i < arr.length; i++){
12      str = str + arr[i];
13    }
14    console.log( str );
15  </script>
16 </head>
17 <body>
18

```

思路：就是把里面的值相加 就好了，但是注意保证 是 字符相加。

1. 需要一个新变量用于存放 转换完的字符串。
2. 遍历原来的数组，分别把里面数据取出来， 加到 字符串里面

多维数组

JavaScript它本身是没有多维数组的概念，**因为在JavaScript中 数组元素的数据类型可以是任意数据类型。**假设在一个数组中有一些数组元素的类型还是数组这个时候我们就将它称之为多维数组！

一维数组

简单来说，**一个数组内所有的元素的数据类型的都不是数组类型，就是一维数组。**

多维数组

超过一维都是可以叫多维，**多维数组指的是一个数组中的元素又是一个数组。**

如何访问二维数组的里面的数组元素呢？

格式：

- 1 访问 一维数组的里面的数组元素：数组变量名[一维数组的下标]；
- 2 访问 二维数组的里面的数组元素：数组变量名[一维数组的下标][二维数组的下标]；

举例：

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <script type="text/javascript">
7     // JavaScript它本身是没有多维数组的概念，因为在JavaScript中 "数组元素的数据类型可以是任意数据类型"。假设在"一个数组中有一些数组元素的类型还是数组" 这个时候我们就将它称之为多维数组！
8
9     // 一句话:数组元素又是数组,那么我们可以叫多维数组
10
11    // 二维数组就是数组中的元素还是一个数组
```

```

12 // 比如
13 // var arr = [
14 // ["张三",23,"男"],
15 // ["李四",24,"女"],
16 // ["王五",25,"男"],
17 // ];
18
19
20 // 定义二维数组,数组元素又是一个数组
21 var arr = [
22 ["张三","男",23,"野鸡科技大学"],
23 ["李四","女",24,"家里蹲大学"]
24 ];
25
26 // 如何访问二维数组中的数组元素
27 console.log( arr );
28 console.log( arr[0] );
29 console.log( arr[0][0] );// 张三
30 console.log( arr[0][3] );// 野鸡科技大学
31 console.log( arr[1][2] );// 24
32 </script>
33 </head>
34 <body>
35
36 </body>
37 </html>

```

遍历二维数组 重点难点

遍历二维数组一般会使用**两层for嵌套**!因为一层for只能遍历一维数组!

语法:

```

1 var 数组变量名 = [
2   [11,8,35],
3   [7,51,99]
4 ];
5
6 for(var i=0;i<数组变量名.length;i++){
7   for(var j=0;j<数组变量名[i].length;j++){

```



```
8 console.log(数组变量名[i][j])
9 }
10 }
```

举例:请把下列二维数组所有元素遍历出来

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <title>Document</title>
6 <script type="text/javascript">
7 var arr = [
8   [11,8,35],
9   [7,51,99],
10 ];
11
12 // console.log( arr[0][1] );// 8
13 // console.log( arr[1][2] );// 99
14
15 // 我们想把二维数组中所有元素都打印出来
16 // console.log( arr[0][0] );
17 // console.log( arr[0][1] );
18 // console.log( arr[0][2] );
19
20 // console.log( arr[1][0] );
21 // console.log( arr[1][1] );
22 // console.log( arr[1][2] );
23
24 // 修改成for循环
25 // console.log( arr.length );//2
26 // console.log("");
27
28 /*for(var i=0;i < arr.length;i++){
29   console.log( arr[i][0] );
30   console.log( arr[i][1] );
31   console.log( arr[i][2] );
32 }*/
33
34 // 修改成两个for循环嵌套
35 for(var i=0;i < arr.length;i++){
36   for(var j=0;j < arr[i].length; j++){
```

```

37 console.log( arr[i][j] );
38 }
39 }
40
41 // 总结一下:遍历二维数组的语法
42 /*
43 for( var i = 0;i< 数组变量名.length ;i++){
44 for( var j = 0;j< 数组变量名[i].length ;j++){
45 console.log( 数组变量名[i][j] );
46 }
47 }
48 */
49 </script>
50 </head>
51 <body>
52
53 </body>
54 </html>

```

新增数组元素

新增数组元素,就是在已经创建好的数组基础上,往数组中添加元素

格式:

```
1 javascript里的数组可以通过直接访问    数组变量名[下标]=值    实现新增数组元素
```

比如:

```

1 var arr = ["小白","小黑","大黄","瑞奇"];
2 arr[4] = "大白";
3 console.log(arr);

```

注意:一定 要用 数组加下标的方式,并且这个下标不存在,追加数组元素, 不能直接给数组名赋值, 否则会覆盖掉以前的数据。

举例:

```

1 <!DOCTYPE html>
2 <html lang="en">

```

```
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <script type="text/javascript">
7     // 新增数组元素
8     // 格式:
9     // javascript里的数组可以通过直接访问 数组变量名[下标]=值 实现新增数组元素
10
11    // 注意:一定 要用 数组加下标的方式 , 并且这个下标不存在, 追加数组元素;如果下
12    // 标存在,会覆盖原下标对应的的值;
13
14    // 不能直接给数组名赋值, 否则会覆盖掉以前的数据。
15
16    var arr = [10,20,30,40];
17    arr[0] = 50;
18    arr[2] = 60;
19    arr[6] = 70;
20    console.log( arr );// [50, 20, 60, 40, empty × 2, 70]
21    console.log( arr[4] );// undefined
22    console.log( arr[5] );// undefined
23    console.log( arr.length );// 7
24    console.log("");
25
26    var arr2 = [50,78,90];
27    // 如果没有设置下标,整个数组都会被覆盖
28    arr2 = 5;
29    arr2 = 6;
30    console.log(arr2);
31    console.log("");
32
33    // 巧妙利用数组的长度,给数组添加元素
34    var arr3 = [10,20,30];
35    arr3[arr3.length] = 40;
36    arr3[arr3.length] = 50;
37    arr3[arr3.length] = 60;
38    console.log(arr3);
39    console.log(arr3.length);
40  </script>
41 </head>
42 <body>
43
44 </body>
```

今日总结

xmind要做

今日作业

4个题,全都要做,在作业文件夹里面