# 一、数组

## 1、什么是数组？

数组是值的集合，可以根据需求向数组增加 更多的值。（有些像java中的集合）

## 2、创建数组

(1)方式1：创建空数组

var arrs = new Array();

(2)方式2：创建带指定长度的数组

var arrs = new Array(8);

(3)注意：

创建的新数组是空数组，没有值，如果输出，结果为undefined。

即使创建带指定长度的数组，但是数组的长度是可以根据需求变的。就是说可以存储指定长度之外数量的值。

## 3、给数组赋值

(1)给空数组赋值

var arrs = new Array();

arrs[0]="小白";

arrs[1]="小呆";

(2)创建数组的同时赋值

var arrs = new Array(2,3,4,5);

(3)直接输出一个数组

var arrs = [1,2,3,4];

(4)注意：

数组可以存储任意类型的数据（字符、数组、布尔等）。

即使是一个数组，也可以同时存储不同类型的数据。

## 4、数组的长度length

(1)同 Java一样，数组的下标是从0开始的。索引值范围是[0,length-1];

(2)增大数组长度的方式

1）方式1：为数组长度赋更大的值

var arr = new Array(8);//数组长度是8

arr.length= 10; //此时数组长度是10

2）方式2：直接向数组中添加数据，当值的数量超过原来数组的长度时，数组的长度自动改变

var arr = new Array(8);//数组长度是8

arr[11] = "33"; // 此时数组的长度变为12（11+1）

# 二、二维数组

## 1、什么是二维数组？

JavaScript的二维数组同java的二维数组一样

## 2、创建方式

(1)方式1：先创建一维数组，再创建二维数组。

var myarr=new Array(); //先声明一维

for(var i=0;i<2;i++){ //一维长度为2

myarr[i]=new Array(); //再声明二维

for(var j=0;j<3;j++){ //二维长度为3

myarr[i][j]=i+j; // 赋值，每个数组元素的值为i+j

}

}

(2)直接定义二维数组

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

## 3、对二维数组赋值

arrs[0][1]=5;//将5传入到数组中，覆盖原来位置的值

# 三、流程控制语句

## 1、if判断语句

(1)语法

if(条件){

条件成立时执行代码

}else{

条件不成立时执行代码

}

## 2、switch语句

与java 的switch相同。

## 3、for循环、while循环、do...while(条件)循环

## 4、break 跳出循环语句、continue 终止本次循环，进入下次循环

# 四、函数

## 1、函数的语法格式

function 函数名(参数){

函数体

}

## 2、参数说明

函数的参数可有可无，可以有1个，也可以有多个。

## 3、函数的调用

(1)方式一：直接调用

<script type="text/javascript">

function add2(){

sum = 1 + 1;

alert(sum);

}

add2();//调用函数，直接写函数名。

</SCRIPT>

(2)方式二：在HTML中触发式调用

<html>

<head>

<script type="text/javascript">

function add2(){

sum = 5 + 6;

alert(sum);

}

</script>

</head>

<body>

<form>

//按钮,onclick点击事件，直接写函数名

<input type="button" value="click it" onclick="add2()">

</form>

</body>

</html>

## 4、带返回值的函数

(1)语法：

function add(x,y){

var x,y,sum;

sum = x+y;

return sum;//sum就是返回值

}

(2)函数值处理

(1)在输出语句中直接返回

document.wrie(add(2,3));

(2)用一个变量接收

var result = add(2,3);

# 五、编程实战

1、需求：使用javascript代码写出一个函数：实现传入两个整数后弹出较大的整数

|  |
| --- |
| <!--使用javascript代码写出一个函数：实现传入两个整数后弹出较大的整数-->  <script>  function compare(x, y) {  if(x > y) {  alert("比较大的数值是：" + x);  } else if(x < y) {  alert("比较大的数值是：" + y);  } else {  alert("两个数相等");  }  }  var x = prompt("请输入第一个值：");  if(x != null) {  var y = prompt("请输入第二个值：");  if(y != null) {  compare(x, y);  }  }  </script> |

2、给出学生的数据，从这些数据中找出大一年级女生的信息

|  |
| --- |
| <script type="text/javascript">  //第一步把之前的数据写成一个数组的形式,定义变量为  var infos = [  ["小A", "女", 21, "大一"],  ["小B", "男", 23, "大三"],  ["小C", "男", 24, "大四"],  ["小D", "女", 21, "大一"],  ["小E", "女", 22, "大四"],  ["小F", "男", 21, "大一"],  ["小G", "女", 22, "大二"],  ["小H", "女", 20, "大三"],  ["小I", "女", 20, "大一"],  ["小J", "男", 20, "大三"]  ];  //第一次筛选，找出都是大一的信息  for(var i = 0; i < infos.length; i++) {  if("大一" == infos[i][3]) {  //第二次筛选，找出都是女生的信息  if("女" == infos[i][1]) {  document.write(infos[i]);  document.write("<br/>");  }  }  }  document.write("<br/>");  document.write("<br/>");  document.write("<br/>");  document.write("<br/>");  </script> |

# 六、JavaScript事件

## 1、JavaScript主要事件

事件 说明

1.onclick 鼠标单击事件

2.onmouseover 鼠标经过事件

3.onchange 文本框内容改变事件

4.onselect 文本框被选中事件

5.onfocus 光标聚集事件

6.onblur 光标离开

7.onload 网页导入

8.onunload 关闭网页

## 2、事件实战练习

1.需求：

使用JS完成一个简单的计算器功能。实现2个输入框中输入整数后，点击第三个输入框能给出2个整数的加减乘除。

2.提示：

获取元素的值：var element = document.getElementById("id").value;

设置元素的值：document.getElementById("id").value = 1;

|  |
| --- |
| <head>  <meta charset="UTF-8">  <title>JavaScript事件响应，让网页交互</title>  <script>  function count() {  var txt1 = document.getElementById("txt1").value;  var select = document.getElementById("select").value;  var txt2 = document.getElementById("txt2").value;  var result;  switch(select) {  case "+":  result = parseInt(txt1) + parseInt(txt2);  document.getElementById("fruit").value = result; //赋值  break;  case "-":  result = txt1 - txt2;  document.getElementById("fruit").value = result; //赋值  break;  case "\*":  result = txt1 \* txt2;  document.getElementById("fruit").value = result; //赋值  break;  case "/":  result = txt1 / txt2;  document.getElementById("fruit").value = result; //赋值  break;  }  }  </script>  </head>  <body>  <input type='text' id='txt1' />  <select id='select'>  <option value='+'>+</option>  <option value="-">-</option>  <option value="\*">\*</option>  <option value="/">/</option>  </select>  <input type='text' id='txt2' />  <input type='button' value=' = ' onclick="count()" />  <!--通过 = 按钮来调用创建的函数，得到结果-->  <input type='text' id='fruit' />  </body> |

# 七、JavaScript内置对象。

1、JavaScript 中的所有事物都是对象，如:字符串、数值、数组、函数等，每个对象带有属性和方法

（1）对象的属性：反映该对象某些特定的性质的，如：字符串的长度、图像的长宽等

（2）对象的方法：能够在对象上执行的动作。例如，表单的“提交”(Submit)，时间的“获取”(getYear)等；

2、JavaScript 提供多个内建对象，比如 String、Date、Array 等等，使用对象前先定义。

## 1、Date日期对象

（1）Date对象日期可以存储任意一个日期，可以精确到毫秒数。

（2）创建一个Date对象

语法格式

var date= new Date();

初始值：

Date对象创建完成后，会赋予一个初始值（当前电脑系统时间）

（3）自定义初始值

var d = new Date(2012, 10, 1); //2012年10月1日

var d = new Date('Oct 1, 2012'); //2012年10月1日

（4）访问时间的方法

我们最好使用JavaScript推荐的方法来严格定义时间，语法格式：

<日期对象>.<方法>

Date对象常用的处理时间和日期的方法是 get/set方法。

（5）访问/设置年份

语法：

getFullYear(); //Date对象的获取年份

setFullYear(年,月,日); //设置Date对象的年月日，也可只设置年份

注意：

(1)getMonth():返回的月份是一个0-11之间的数字，因此中国实际月份需要加一

例如：中国的12月份，在JavaScript的Date对象中是11月份

(2)对象输出结果格式依次为：

星期、月、日、年、时、分、秒、时区。(火狐浏览器)

（6）访问/设置星期

语法：

getDay(); // 获取Date对象表示的时间是星期几。

注意：

getDay() 返回的是一个0-6之间的数字，0表示星期日。如果要返回对应的星期，我们一般用数组来完成。

例如：

var day = date.getDay();

document.write("getDay()的返回值：" + day + "<br>");

var week = ["星期日", "星期一", "星期二", "星期三", "星期四", "星期五", "星期六"];

document.write("今天是星期几？" + week[day] + "<br>");

（7）访问/设置时间

语法：

setTime()： //设置时间，单位毫秒。

getTime() //获取时间，单位毫秒。

注意：

(1)该方法计算的是从1970年1月1日零时开始，到Date对象所表示的日期的毫秒数。

（8）案例

|  |
| --- |
| <!-- Date时间对象案例1:访问/设置年份 -->  <script type="text/javascript">  var date = new Date(); //创建时间对象  document.write("时间对象：" + date + "<br>"); //输出时间对象  date.setFullYear(2019, 10, 5); //设置年月日  //date.setMonth(11); //设置月份  document.write("时间对象：" + date.getFullYear() + date.getMonth() + "<br>");  </script>  <!-- Date时间对象案例2：访问/设置星期 -->  <script type="text/javascript">  var date = new Date();  var day = date.getDay();  document.write("getDay()的返回值：" + day + "<br>");  var week = ["星期日", "星期一", "星期二", "星期三", "星期四", "星期五", "星期六"];  document.write("今天是星期几？" + week[day] + "<br>");  </script> |

（9）时钟

|  |
| --- |
| <!-- Date时间对象案例3：使用Date对象表示时钟 -->  <script type="text/javascript">  function startTime() {  var date = new Date(); //获取时间对象  var year = date.getFullYear(); //年  var month = checkTime(date.getMonth() + 1); //月  var day = checkTime(date.getDate()); //日  var hourse = date.getHours(); //时  var minutes = checkTime(date.getMinutes()); //分  var seconds = checkTime(date.getSeconds()); //秒  var week = ["星期日", "星期一", "星期二", "星期三", "星期四", "星期五", "星期六"];  var weekDay = week[date.getDay()]; //星期  // 向HTML页面输出时间  document.getElementById("txt3").innerHTML = year + "-" + month + "-" + day + "- " + hourse + ":" + minutes + ":" + seconds + " " + weekDay;  t = setTimeout('startTime()', 500);  }  // 如果传入的参数小于10，则在参数前多输出一个0  function checkTime(i) {  if(i < 10) {  i = "0" + 1;  }  return i;  }  </script>  <body onload="startTime()">  <div id="txt3"></div>  </body> |

## 2、String 字符串对象

### （1）直接创建字符串对象并赋值

var str = " I Love JavaScript and Java！！！";

### （2）使用字符串对象调用字符串的方法和属性

var lenth = str.length;//调用字符串的属性

var length = str.toUpperCase();// 调用字符串的方法，将字符串转换为大写

var length = str.toLowerCase();// 调用字符串的方法，将字符串转换为小写

### （3）返回字符串指定位置的字符 charAt

(1)语法：

str.charAt(Number index);

(2)注意事项

1. index取值 0- (str.length-1)之间，返回对应下标所表示的字符

2. 如果index的值不在索引值范围内，那么返回一个空字符串

### （4）返回指定字符首次出现在字符串的位置 indexOf

语法：

str.indexOf(sbuString, index);

参数说明

subString:要查询的字符串

index：从指定下标位置开始查找，如果没有次参数则从下标0开始查询

如果找到字符串出现的位置，则返回子串第一个下标的位置。如果没有找到，返回-1.

### （5）分割字符串 split

语法：

str.split(startPos,limit);

参数说明

index：字符串或正则表达式，字符串分割的标识，匹配不到则返回这个字符串对象，不对字符串进行分割

limit：分割的次数，如果指定则分割次数不会超过limit次；如果不指定，会无限分割，直到将字符串全部分割完

注意：

如果将("")空字符用作index，那么字符串的每个字符之间都会被分割。

### （6）截取字符串 substring

* 语法：

str.substring(startPos,endPos);

* 参数说明

startPos：开始截取字符串的下标

endPos：截取字符串结束的下标

* 注意：

截取字符从的长度包含start,不包含end（即 截取start到end-1之间的的字符串），其长度是end-start

如果start=end，那么返回的是一个空字符串

如果start>end，那么截取前会先交换这两个参数的位置。

### （7）截取指定数量的字符串 substr

* 语法：

str.substr(start,length);

(2)参数说明

start：开始截取字符串的位置

length：截取字符串的长度

* 注意事项

如果start是负数，则从字符串的尾部开始截取，-1表示最后一个字符，-2表示倒数第二个...

如果start为负数且绝对值大于字符串长度，则start=0；

### （8）parseInt() 字符串转数字

* 将字符串转换成数字

var aa = “33”;

var bb = parseInt(aa);

document.write(bb);//33

# 八、Math数学对象

Math对象中提供了对数据的数学计算的方法

1、ceil() 向上取整

2、floor() 向下取整

3、round() 四舍五入

4、random() 随机数

# 九、数组对象

## 1、数组定义的方法

(1)定义一个空数组

var arrs = new Array();

(2)定义时指定数组长度

var arrs = new Array(6);

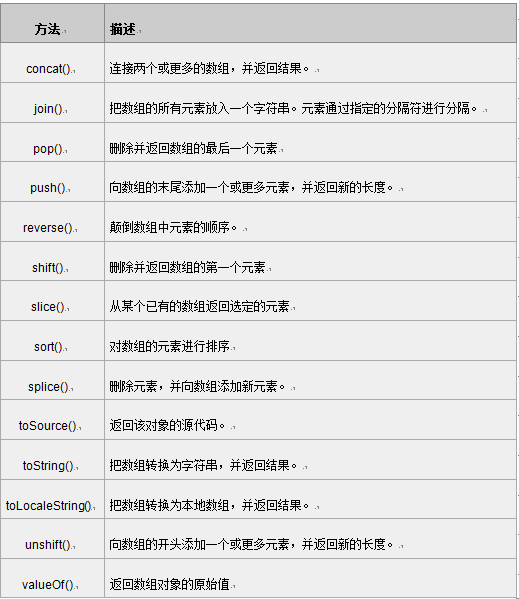
(3)定义时直接初始化数据

var arrs = [值1,值2,值3,……];

## 2、数组的使用

数组名[下标] = 值;

## 3、数组的方法



### （1）concat数组的连接

* 语法：

数组对象.concat(arrs1,arrs2,…,arrsn)

* 注意：

该方法不会改变现有的数组，而仅仅会返回被连接数组的一个副本

### （2）join将数字转换成字符串

* 语法：

数组对象.join(separator);

* 参数说明

separator：分割符

* 返回一个字符串，该字符串将数组中所有元素串联起来，各元素之间用分割符separator隔开。
* 如果不使用分割符，那么系统默认使用 英文逗号(,) 对元素进行分隔。
* 这并不会影响原来的数组。

### （3）reverse反转数组元素

* 语法：

数组对象.reverse();

* 注意：

该方法会改变原来数组中元素的顺序，而不是新建一个数组。

### （4）选定元素slice()

* 语法：

数组对象.slice(start,end);

* 参数说明

start：必须。规定从何处选取数组元素

end：可选。规定从何处结束，如果不选，则切分到数组结束。

* 注意：

该方法返回一个新的子数组，不会改变原数组。数组元素为start到end(不包含)。

start可以是负数，表示从尾部开始选取。

String.slice()与Array.slice相似。

### （5）排序 sort()

* 语法：

对象.sort(比较函数);

* 注意：
  + 如果不指定<方法函数>，则按Unicode码顺序排序。
  + 函数至少有a,b两个参数，并且比较两个函数值，然后返回一个用于说明这两个值相对顺序的数字。
* sort默认比较规则：
  + 如果比较数字，会从数字的第一位开始比较，小的在前，大的在后。然后比较第二位，知道比较结束。注意，比较的不是两个数字的值的大小。
  + 如果比较字母，会从第一个字母开始比较，按a-z排序，第一个字母相同的比较第二个字母，直到排序完成。