

课程回顾

1 数据类型，变量

2 运算符

3 流程控制

课后作业

计算器：添加了非空验证和非数值格式验证

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="UTF-8">
5          <title></title>
6      </head>
7      <script type="text/javascript">
8          function getResult(){
9              //1.获取两个数据  value获取值都是string
10             var num1=document.getElementById("txtNum1").value;//表单获取用户输入的值或选择的值都是value
11             // alert(num1.length);//没有任何显示。不是null
12             var num2=document.getElementById("txtNum2").value;//
13             // alert(num2)
14             //2.获取运算符
15             var op=document.getElementById("selOp").value;//
16             //验证
17             //非空验证
18             // if(num1==null || num1=='')
19             if(!num1||!num2){
20                 alert("必须填写数字1和数字2");
21                 return;
22             }
23             //验证num1和num2是不是数值 isNaN():boolean true-不是数字 false-是数字 is Not a Number?
24             // alert("num1不是数字吗? "+isNaN(num1))
25             if(isNaN(num1)||isNaN(num2)){
26                 alert("必须提供数值才能进行算术运算");
27                 return;
28             }
29             //3.求结果
30             switch (op){
31                 case '+':
32                     //加法
33                     alert(parseInt(num1)+parseInt(num2));
34                     break;
35                 default:
36                     break;
37             }
38         }
39     </script>
```

```

40 <body>
41     输入第一个数字: <input id="txtNum1" type="text" /><br />
42     选择运算符: <select id="selOp">
43         <option>+</option>
44     </select><br />
45     输入第二个数字: <input id="txtNum2" type="text" /><br />
46     <!--
47         onClick:称为事件监听器
48         在标签上, 添加监听器并设置对应功能, 称为事件监听机制
49
50     -->
51     <input type="button" onclick="getResult()" value="计算" />
52 </body>
53 </html>

```

打印空心菱形

```

1  <!DOCTYPE html>
2  <html>
3
4      <head>
5          <meta charset="UTF-8">
6          <title></title>
7      </head>
8      <script type="text/javascript">
9          function gettt() {
10              var a = document.getElementById('num').value;//9
11              if(a % 2 != 0) {
12                  for(var i = (parseInt(a/-2)); i <= parseInt(a / 2); i++) {
13                      for(var j = 1; j<=Math.abs(i); j++) {
14                          //document.write()向html网页输出数据
15                          document.write("&nbsp;");
16                      }
17                      for(var k = 1; k <= a - 2*Math.abs(i); k++) {
18                          //如果不是第一个* 且不是最后一个*, 不打印*
19                          if(k!=1 && k!=a - 2*Math.abs(i)){
20                              document.write("&nbsp;");
21                              continue;
22                          }
23                          document.write("*");
24                      }
25                      document.write('<br/>');
26                  }
27              } else {
28                  prompt('请输入奇数!', 9);
29              }
30          }
31      </script>
32
33      <body>
34          你要打一个多大的菱形?
35          <input type="text" name="num" id="num" />
36          <input type="button" id="right" onclick="gettt()" value="确认" />
37
38      </body>
39
40

```

课程目标

1 Math ==== 掌握

2 String ==== 掌握

3 Array ==== 掌握

4 Function函数 ==== 掌握

5 RegExp ==== 掌握

6 Date====掌握

7 表单验证思路 ==== 掌握

课程实施

1 Math对象 === java完全一样

Math 对象属性

FF: Firefox, IE: Internet Explorer

属性	描述	FF	IE
E	返回算术常量 e，即自然对数的底数（约等于2.718）。	1	3
LN2	返回 2 的自然对数（约等于0.693）。	1	3
LN10	返回 10 的自然对数（约等于2.302）。	1	3
LOG2E	返回以 2 为底的 e 的对数（约等于 1.414）。	1	3
LOG10E	返回以 10 为底的 e 的对数（约等于0.434）。	1	3
PI	返回圆周率（约等于3.14159）。	1	3
SQRT1_2	返回返回 2 的平方根的倒数（约等于 0.707）。	1	3
SQRT2	返回 2 的平方根（约等于 1.414）。	1	3

Math 对象方法

FF: Firefox, IE: Internet Explorer

方法	描述	FF	IE
abs(x)	返回数的绝对值。	1	3
acos(x)	返回数的反余弦值。	1	3
asin(x)	返回数的反正弦值。	1	3
atan(x)	以介于 -PI/2 与 PI/2 弧度之间的数值来返回 x 的反正切值。	1	3
atan2(y,x)	返回从 x 轴到点 (x,y) 的角度（介于 -PI/2 与 PI/2 弧度之间）。	1	3
ceil(x)	对数进行上舍入。	1	3
cos(x)	返回数的余弦。	1	3
exp(x)	返回 e 的指数。	1	3
floor(x)	对数进行下舍入。	1	3
log(x)	返回数的自然对数（底为e）。	1	3
max(x,y)	返回 x 和 y 中的最高值。	1	3
min(x,y)	返回 x 和 y 中的最低值。	1	3
pow(x,y)	返回 x 的 y 次幂。	1	3
random()	返回 0 ~ 1 之间的随机数。	1	3
round(x)	把数四舍五入为最接近的整数。	1	3
sin(x)	返回数的正弦。	1	3
sqrt(x)	返回数的平方根。	1	3
tan(x)	返回角的正切。	1	3
toSource()	返回该对象的源代码。	1	-
valueOf()	返回 Math 对象的原始值。	1	4

需求：水仙花

1 | 技能点：Math.pow(a,b)

参考代码

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title></title>
6   </head>
7   <script type="text/javascript">
8     //      Math.pow()+Math.pow()+Math.pow();
9     for(var i=100;i<1000;i++){
10       with(Math){
11         //个位
12         var gewei=floor(i%10);//
```

```

13      //十位
14      var shiwei=floor(i%100/10);
15      //百位
16      var baiwei=floor(i/100);
17      if(pow(gewei,3)+pow(shiwei,3)+pow(baiwei,3)===i){
18          document.write(i+"&nbsp;");
19      }
20  }
21
22  }
23      /*with(Math){
24          pow()
25      }*/
26  </script>
27  <body>
28  </body>
29  </html>

```

补充：with()代码块用法

在一段连续的程序代码中，如果多次使用到某个对象的属性或方法，那么只需要在 with 关键字后的小括号中写出这个对象的名称，然后就可以在其后大括号中的执行语句里直接引用该对象的属性名或方法名，不必再在每个属性和方法名前都加上对象实例名和点（.）。



```

with(对象名称){

    执行语句块

}

```

2 Date对象=====类似java中Calendar对象

2-1 创建日期对象

```

1  new Date();//获取当前时间

```

2-2 获取和设置日期的方法

Date()	返回当日的日期和时间。	1	3
getDate()	从 Date 对象返回一个月中的某一天 (1 ~ 31)。	1	3
getDay()	从 Date 对象返回一周中的某一天 (0 ~ 6)。	1	3
getMonth()	从 Date 对象返回月份 (0 ~ 11)。	1	3
getFullYear()	从 Date 对象以四位数字返回年份。	1	4
getYear()	请使用 getFullYear() 方法代替。	1	3
getHours()	返回 Date 对象的小时 (0 ~ 23)。	1	3
getMinutes()	返回 Date 对象的分钟 (0 ~ 59)。	1	3
getSeconds()	返回 Date 对象的秒数 (0 ~ 59)。	1	3
getMilliseconds()	返回 Date 对象的毫秒(0 ~ 999)。	1	4
getTime()	返回 1970 年 1 月 1 日至今的毫秒数。	1	3

parse()	返回1970年1月1日午夜到指定日期（字符串）的毫秒数。	1	3
setDate()	设置 Date 对象中月的某一天 (1 ~ 31)。	1	3
setMonth()	设置 Date 对象中月份 (0 ~ 11)。	1	3
setFullYear()	设置 Date 对象中的年份（四位数字）。	1	4
setYear()	请使用 setFullYear() 方法代替。	1	3
setHours()	设置 Date 对象中的小时 (0 ~ 23)。	1	3
setMinutes()	设置 Date 对象中的分钟 (0 ~ 59)。	1	3
setSeconds()	设置 Date 对象中的秒钟 (0 ~ 59)。	1	3
setMilliseconds()	设置 Date 对象中的毫秒 (0 ~ 999)。	1	4
setTime()	以毫秒设置 Date 对象。	1	3

2-3 日期转换为格式良好的字符串方法

toLocaleString()	根据本地时间格式，把 Date 对象转换为字符串。	1	3
toLocaleTimeString()	根据本地时间格式，把 Date 对象的时间部分转换为字符串。 Date==>String 格式的方法	1	3
toLocaleDateString()	根据本地时间格式，把 Date 对象的日期部分转换为字符串。	1	3

2-4 课堂案例

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="UTF-8">
5          <title></title>
6      </head>
7      <script type="text/javascript">
8          var now=new Date();
9          //now==>格式良好的日期格式
10         /*var strDate=now.toLocaleString();
11         document.write(strDate);*/

```

```

12 //获取日期的各个部分，自己拼接字符串
13
14 with(now){
15     //now设置昨天
16     //1.setXX
17     // now.setDate(19);
18     //2.获取当前时间距离1970...的毫秒值
19     // var milNum=now.getTime();
20     // var yesMilNu=milNum-24*60*60*1000*2;
21     // now.setTime(yesMilNu);
22     //3.指定时间转换为毫秒值
23     var aaa=Date.parse("2022-01-1");
24     now.setTime(aaa);
25     var year=getFullYear();//年份
26     var month=getMonth()+1;//月份（0-11）
27     var date=getDate();//日期
28     var weekday=getDay();//星期
29     var hour=getHours();//时
30     var minute=getMinutes();//分
31     var seconde=getSeconds();//秒
32     var milSecond=getMilliseconds();//毫秒
33
34     //switch-case
35     document.write(year+"年"+month+"月"+date+"日 星期"+weekday);
36 }
37 </script>
38 <body>
39 </body>
40 </html>

```

3 String对象 ===== java大部分方法一样

<u>length</u>	字符串的长度
<u>substr()</u>	从起始索引号提取字符串中指定数目的字符。 截取子字符串
<u>substring()</u>	提取字符串中两个指定的索引号之间的字符。

- 1 substr(startIndex,count): startIndex开始截取的字符串下标位置，count: 截取字符串的字符个数
- 2 substring(startIndex,endIndex): startIndex开始截取的字符串下标位置，endIndex: 结束截取下标位置

课堂案例

- 1 var str="helloworld";
- 2 分别使用substr()和substring()两种方式截取 llo

参考代码

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title></title>
6   </head>
7   <script type="text/javascript">
8     //substr()和substring()
9     //1.定义字符串对象
10    var objStr="helloworld";
11    //2.截取llowo
12    //2-1 substring(startIndex,endIndex)
13    var
14    result=objStr.substring(objStr.indexOf('l'),objStr.lastIndexOf('o')+1);
15    document.write(result);
16
17    document.write("<br/>");
18
19    //2-2 substr(startIndex,count)
20    result=objStr.substr(objStr.indexOf('l'),"llowo".length);
21    document.write(result);
22  </script>
23  <body>
24  </body>
25 </html>
```

4 Array对象=====类似java中的集合，难点

数组：一种用来保存多个数据的类型。

数组：长度可变、类型可以任意的

4-1 Array的定义

```
1 方式一：
2   var arr=[元素1,元素2,...,元素n];//由构造方法中提供的元素决定数组初始长度
3 方式二：
4   var arr2=new Array(长度);//数组初始长度
5 方式三：
6   var arr3=new Array(元素1,元素2,...,元素n);//由构造方法中提供的元素决定数组初始长度
```

4-2 获取和设置数据

```
1 数组名[下标];//获取下标位置的元素
2 数组名[下标]=值;//设置下标位置的元素
```

4-4 循环方式


```

1  普通for
2  for(var i=0;i<数组名.length;i++){
3
4  }
5
6  扩展:
7  for-in语句
8  foreach()

```

4-5 Array提供的方法

方法	描述	FF	IE
concat()	连接两个或更多的数组，并返回结果。	1	4
join()	把数组的所有元素放入一个字符串。元素通过指定的分隔符进行分隔。	1	4
pop()	删除并返回数组的最后一个元素	1	5.5
push()	向数组的末尾添加一个或更多元素，并返回新的长度。	1	5.5
reverse()	颠倒数组中元素的顺序。	1	4
shift()	删除并返回数组的第一个元素	1	5.5
slice()	从某个已有的数组返回选定的元素	1	4
sort()	对数组的元素进行排序	1	4
splice()	删除元素，并向数组添加新元素。	1	5.5
toSource()	返回该对象的源代码。	1	-
toString()	把数组转换为字符串，并返回结果。	1	4
toLocaleString()	把数组转换为本地数组，并返回结果。	1	4
unshift()	向数组的开头添加一个或更多元素，并返回新的长度。	1	6
valueOf()	返回数组对象的原始值	1	4

4-6 课堂案例

- 数组基本用法：定义、赋值、取值、普通for

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="UTF-8">
5          <title></title>
6      </head>
7      <script type="text/javascript">
8          //长度可变、类型可以不一样
9          //定义数组，长度初始值4
10         var arr=new Array(4); //数组每一个下标位置对应undefined
11         //给下标是99的位置存入数据HelloWorld
12         arr[0]=true;
13         arr[1]=12;
14         arr[2]=45.67;
15         //下标的范围[0,N-1]
16         arr[99]="helloWorld";
17
18         //输出数组长度
19         // alert("数组长度是: "+arr.length); //100

```

```

20 // alert("第98个数据是: "+arr[97]); //undefined
21
22 //循环for
23 for(var i=0;i<arr.length;i++){
24     document.write(arr[i]+"&nbsp;");
25 }
26 </script>
27 <body>
28 </body>
29 </html>

```

- 嵌套数组

```

1 <!DOCTYPE html>
2 <html>
3     <head>
4         <meta charset="UTF-8">
5         <title></title>
6     </head>
7     <script type="text/javascript">
8         //1. 定义嵌套数组
9         var arr=new Array([0,0,0],[1,1,1],[2,2,3,4,5]);
10        //2. 输出数组所有的数据
11        for(var i=0;i<arr.length;i++){
12            // var temp=arr[i]; //i 获取的数据又是数组
13            for(var j in arr[i]){
14                document.write(arr[i][j]);
15            }
16            //换行
17            document.write("<br/>");
18        }
19    </script>
20    <body>
21    </body>
22 </html>

```

- 数组循环方式的冒泡排序为例

```

1 <!DOCTYPE html>
2 <html>
3     <head>
4         <meta charset="UTF-8">
5         <title></title>
6     </head>
7     <script type="text/javascript">
8         //1. 定义数组
9         var arr=[7,2,-90,44,89];
10
11        //2. 排序前输出
12        //特点: for-in 获取所有的下标
13        for(var i in arr){
14            document.write(arr[i]+",");
15        }
16
17        //排序
18        for(var i=0;i<arr.length-1;i++){

```

```

19         for(var j=0;j<arr.length-1-i;j++){
20             if(arr[j]>arr[j+1]){
21                 //交换
22                 var temp=arr[j];
23                 arr[j]=arr[j+1];
24                 arr[j+1]=temp;
25             }
26         }
27     }
28     document.write("<br />");
29     //3.排序后输出
30     /*for(var index in arr){
31         document.write(arr[index]+",");
32     }*/
33     //4.了解
34     //element:元素
35     //index:下标
36     //arr:要循环的数组变量名
37     arr.forEach(function(element,index,arr){
38         document.write(index+": "+element+",");
39     });
40 </script>
41 <body>
42 </body>
43 </html>

```

补充:

4.13 for/in 语句

作用：用于遍历数组或者对象的属性（对数组或者对象的属性进行循环操作）

特点：

for... in 语句用于对数组或者对象的属性进行循环操作

for ... in 循环中的代码每执行一次，就会对数组的元素或者对象的属性进行一次操作



for (变量 in 对象或数组)

{

在此执行代码

}

forEach()

forEach() 方法无返回值，用于调用数组的每个元素，并将数组元素传递给回调函数。当数组值为空时则不执行回调函数。



```
数组对象.forEach(function(item, index, arr){  
  
})
```



item 为当前遍历的数组元素
index 为当前元素的索引值
arr 为当前所遍历的数组对象

4-7 嵌套数组

```
1 方式一：  
2   var arr=[[元素1,元素2],...,[元素n]]; //由构造方法中提供的元素决定数组初始长度  
3 方式二：  
4   var arr2=new Array(长度); //数组初始长度  
5   arr2[0]=new Array(3);  
6   arr2[1]=[];  
7   arr2[2]=new Array(1,3,3,4,4)  
8 方式三：  
9   // var arr3=new Array([元素1],[元素2],...,[元素n]); //由构造方法中提供的元素决定数组  
10  初始长度  
11 嵌套数组：模拟Map的特点：key--value
```

课堂案例：

1 | 使用数组保存一个国家对应的省份、每个省份对应的城市、每个城市对应的县/区

参考答案

```
1  <!DOCTYPE html>  
2  <html>  
3    <head>  
4      <meta charset="UTF-8">  
5      <title></title>  
6    </head>  
7    <script type="text/javascript">  
8      //定义数组保存所有的省  
9      var province=["湖北省","河南省","湖南省","河北省"];  
10     //使用省份名称为key，存入该省对应的市  
11     province["湖北省"]=["武汉市","黄冈市","襄阳市","荆州市"];  
12     province["河南省"]=["郑州市","信阳市","洛阳市","驻马店"];  
13     province["湖南省"]=["长沙市","郴州市","岳阳市","衡阳市"];  
14     province["河北省"]=["石家庄市","邯郸市","秦皇岛市"];
```

```

15 //一个格式，实现市和区的对应
16 province["武汉市"]=["洪山区","武昌区","汉口区"];
17 province["襄阳市"]=["襄州区","宜城市","谷城市"];
18 province["长沙市"]=["羊区","人区"];
19
20 //输出：
21 /*
22  * 湖北省：
23  *     武汉市：洪山区
24  *     荆州市：
25  * 河南省：
26  *     郑州市：。。。。
27  */
28 for(var i=0;i<province.length;i++){//province.length: 所有的省
29     //变量存储
30     var provinceStr=province[i];
31     document.write(provinceStr+":<br/>");
32     //获取该省对应的市
33     var cities=province[provinceStr]
34     for(var j=0;j<cities.length;j++){//cities.length是一个省所有的市的
        个数
35         var cityStr=cities[j];
36         document.write(cityStr+":");
37         //获取该市对应的所有的县区
38         var regions=province[cityStr];
39         for(){
40             //
41         }
42     }
43
44 }
45 </script>
46 <body>
47 </body>
48 </html>

```

5-1 定义语法

```

1 function 函数名(形参列表){
2     //方法体
3     return 值;
4 }

```

5-2 方法调用

```

1 var 变量名=函数名(实参列表)//有返回值
2
3 函数名(实参列表);//无返回值
4 举例：
5     alert()
6     prompt():String
7     isNaN():boolean

```

5-3 课堂案例

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="UTF-8">
5          <title></title>
6      </head>
7
8      <body>
9          <!--
10             设置文本框不能操作: disabled="disabled"
11
12             -->
13          <input type="text" id="txtSum" disabled="disabled" />
14      </body>
15      <script type="text/javascript">
16          //JavaScript不支持重载的,所以重载模拟的!!
17          //如果同一个html网页,出现同名方法,后面的方法会自动覆盖上面的方法
18          //求两个整数和
19          //形参不需要类型,只需要定义变量名和个数
20          function add(a,b){
21              //arguments:每个函数中有一个arguments属性,保存实参列表,arguments就是
22              Array! !
23              //      alert("两个参数方法")
24              //      var sum=a+b;
25              //      var sum=0;
26              //      for(var i=0;i<arguments.length;i++){
27              //          sum+=arguments[i];
28              //      }
29              //      return sum;
30              //      document.write(sum);
31              }
32              /*function add(a,b,c){
33                  alert("三个参数方法")
34                  var sum=a+b;
35                  document.write(sum);
36              }*/
37              //调用
38              //      add(12.45,"abc");
39              //      var result=add(12.45,12,3,4,5);
40              //      //获取text
41              //      var objTxtSum=document.getElementById("txtSum");
42              //      //text.value=计算结果
43              //      objTxtSum.value=result;
44              //      alert(result);
45              //      add(12,23,45,89);
46      </script>
47  </html>
```

6 RegExp对象

7 表单验证思路

课程总结

Math Date String

Array --数组，类似java集合

Function-- 定义和调用

预习安排

DOM: 获取html上标签、操作标签（插入、删除、修改、克隆）

全选 全不选 反选

三级联动