**JS 常用技能**

**数组操作：**增删改查在遍历

**正则操作：**检替分

**||，&&，！；**或与非（竖或地与感叹非）

**引入JS文件的三种方式：**

1. <script src="js/DataWarehouse.js" type="text/javascript" charset="utf-8"></script>
2. import '<script src="js/CommonHead.js" type="text/javascript" charset="utf-8"></script>';
3. document.write('<script src="js/CommonHead.js" type="text/javascript" charset="utf-8"></script>');

**浏览器当前窗口文档body的宽度：** document.body.clientWidth;(仅仅是body的width) 浏**览器当前窗口文档body的高度：** document.body.clientHeight;(仅仅是body的height)

**浏览器当前窗口文档body的宽度：** document.body.offsetWidth;(width+border); 浏览器当**前窗口文档body的高度：** document.body.offsetHeight;(height+border);

**浏览器整个文档的宽：** document.body.scrollWidth;

**浏览器整个文档的高：** document.body.scrollHeight;

**获取竖直滚动条到顶部的垂直高度** (即网页被卷上去的高度)(其他浏览器)：document.body.scrollTop;

**获取竖直滚动条到顶部的垂直高度** (即网页被卷上去的高度)(ie浏览器): document.documentElement.scrollTop;

**获取水平滚动条到左边的水平宽度** (即网页被卷左去的宽度)(其他浏览器)： document.body.scrollLeft;

**获取水平滚动条到左边的水平宽度** (即网页被卷左去的宽度)(ie浏览器)： document.documentElement.scrollLeft;

**滚动条中内容的高度：**boxx.scrollHeight;//利用这个可以使滚动条一直在底部

**网页正文部分上：** window.screenTop;

**网页正文部分左：** window.screenLeft;

**获取元素的宽度：**clientWidth;(width+padding)

**获取元素的高度：**clientHeight;(height+padding)

**获取元素的宽度：**offsetWidth;(width+padding+border)

**获取元素的高度：**offsetHeight;(height+padding+border)

**获取元素最左边距已定位的父级对象的长度**（若无父级对象或父级对象没有定位，就是距离文档顶部）:offsetLeft

**获取元素最上边距已定位的父级对象的长度**（若无父级对象或父级对象没有定位，就是距离文档左端）:offsetTop

**屏幕分辨率的高：** window.screen.height

**屏幕分辨率的宽：** window.screen.width

**屏幕可用工作区高度：** window.screen.availHeight

**屏幕可用工作区宽度：** window.screen.availWidth

**将JSON数据转化成JS对象语法：**JSON.parse()

**获取元素内的值：**.value;

**JS严格模式：**’use strict’

**数组操作：**.splice()

**字符串拼接：** html+=`<li>${arr[j]}</li>`;

**将文本数据转换成json数据：**var json = $.parseJSON(xhr.responseText);

**将json数据转换成JS对象：**var jsonobj = JSON.parse( );

**JS检测数据类型：**typeof(arr)等价于typeof arr

**定义Json数据类型：**var jsonstr = '[{"name":"小明","age":30},{"name":"小红","age":25},{"name":"小李","age":20}]';

**换行符：**\n

**返回字符串形式的响应：**responseText

**返回XML数据形式的响应：**responseXML

**储存函数（或函数名）当readystate属性改变是就会调用储存的函数：**onreadystatechange

**储存XMLHttpRequest的状态，从0—4发生变化：**readystate

**生成随机数据：**var agh = Math.random();地址后面加？后，其后面可以加任何字符串而不会改变页面内容，此事可以用随机数取消浏览器缓存问题

**以毫秒为单位获取当前时间：**new Data.getTime();

**创建cookie:** myCookie = document.cookie = 'name=张三';

**对cookie进行编码加密：**var myCookie = document.cookie = encodeURIComponent('name=李四');

**对cookie解密：**var myCookie = document.cookie = decodeURIComponent('name=李四');

**对cookie解密：**var myCookie = document.cookie = decodeURIComponent('name=李四'，es);

**JS 选择器：**

* **getElementById(ID)**: 返回对指定ID的第一个对象的引用
* **getElementsByName(name)**: 返回文档中name属性为name值的元素，因为name属性值不是唯一的，所以查询到的结果有可能返回的是一个数组，而不是一个元素。
* **getElementsByTagName(tagname)**: 返回文档中指定标签的元素
* **getElementsByClassName()**:返回文档中所有指定类名的元素
* **querySelector()**:返回文档中匹配指定css选择器的第一个元素
* **querySelectorAll()**:返回文档中匹配指定css选择器的第一个元素

**闭包**

闭包就是一个函数（级外函数）控制一个函数（即内函数）处理数据，而通过调用外函数调用内函数的一种手段，此事外函数就是一个闭包。

JS中**“类”**就是一个对象，不是函数。

**正则表达式**

**操作**

在不设置全局匹配，正则则是找到第一个符合条件的元素是就会终止查找

**str.indexOf(st)>-1 布尔值为真**表示查询的字符串str中含有字符串st

**str =str.replace(st,'');** 表示将str中的st字符串替换为‘’中的内

容

**search(str1)** 用于反回字符串中str1字符串所在的位子索引，找不到则返回-1，相当于indexOf();

**match(str1)** 用于将字符串中的str1输出，如果查找补刀则返回null

**replace(str1,str2)** 表示将字符串中的str1替换成str2

**split(str1, number)**  表示以字符串str1将整个字符串分割成数组，数组的长度为numberde 值

**正则表达式区分大小写**

**var reg1 = RegExp(‘a’, ‘g’);** 创建正则表达式，g为修饰符

**var reg2 = /a/g ;** 创建正则表达式，g为修饰符

**var reg2 = /[a-z]/g ;**表示全局查找小写a-z的26个字母

**语法**

**修饰符的作用**

i :执行对大小写不敏感的匹配

g ：执行全局查找，而非在找到的第一个匹配后停止

m ：执行多行匹配

**[ ]语法**

**var reg2 = /[a-z]/g ；中方括号的作用**

**[abc]** 查找给定集合之间的任何字符

**[^abc]** 查找给定集合之外的任何字符

**[0-9]** 查找任何0至9之间的数字

**[a-z]** 查找a到z的26个小写字符中的任何一个

**[A-Z]**  查找A到Z的26个大写字符中的任何一个

**[A-z]** 查找任何从大写A到小写z的52个字符中的任何字符

**‘|’语法**

**Var reg = / red|blue|green** **/g**

**red|blue|green** 查找任何指定的选项其中‘|’表示或的意思

**var reg = /[a-z0-9]/gi;**表示正则表达式的综合属性定义，即匹配A-Z和0到9 ,i控制了不区分大小写， g控制了全局查找

**var reg = /[1-2][0-9]/g;**表是选中范围为10-29

**var reg = /第[1-2][0-9]章/g;**表是选中范围为第10-29章

**元字符**

也就是含有特殊含义的字符

**‘\’** 为转义字符，其用途就是将所有特殊字符或者具有特殊意义的字符转义，否则正则不能匹配，如JS中的‘.’具有特殊用法

**如：**

**Str = ‘ty.png jpg + img&jpg + img+jpg’;**

**var reg = / ty.png jpg /g;返回值为**ty.png jpg ，img&jpg，img+jpg

**var reg = / ty\.png jpg /g;返回值为**ty.png jpg

**‘.’** 表示查找单个字符，除了换行和结束符

**‘\w’** 表示查找单词字符，包括英文字母，数字，下划线

**‘\W’** 表示查找英文字母，数字，下划线的字符

**‘\d’** 表示查找数字

**‘\D’** 表示查找非数字

**‘\s’** 表示查找空白字符

**‘\S’** 表示查找非空白字符

**‘\b’** 匹配单词边界

**‘\B’** 匹配非单词边界

**‘\0’** 查找nul字符

**‘\n’** 查找换行符

**‘\f’** 查找换页符

**‘\t’** 查找制表符

**元字符的作用**

**如：var reg = /\d/g;** 表示匹配0-9的所有数字

**量词**

**N+** 匹配任何至少包含一个N的字符串

**N\*** 匹配任何包含零个或多个N的字符串

**N?** 匹配任何包含零个或一个N的字符串

**N{X}** 匹配包含X个N的序列的字符串

**N{X,Y}** 匹配包含X或Y个N的序列的字符串

**N{X,}** 匹配包含至少X个N的序列的字符串

**N$** 匹配任何结尾为N的字符串

**^N** 匹配任何开头为N的字符串

**?=N** 匹配任何其后紧接着指定字符串为N的字符串

**语法结构**

**var reg = /go+d/g; +**至少有一次o

**var reg = /go\*d/g; \***

**构造函数创建对象**

function Person(name,age){

this.name = name;

this.age = age;

this.run = function(){

return this.name+'的年龄是'+this.age;

}

}

var XiaoMing = new Person('小明',20);

**原型模式创建对象**

function PersonO(){

PersonO.prototype.name = '';

PersonO.prototype.age = 28;

PersonO.prototype.sex = '女';

PersonO.prototype.height = 168;

PersonO.prototype.NVname = function(){

return (PersonO.prototype.name+'的年龄是'+PersonO.prototype.age);

}

}

var PersonOa = new PersonO();

PersonO.prototype.name = '小红';

**AJAX 原生方法请求数据**

function AjaxDataRequest(){

var xhr = new XMLHttpRequest();

xhr.open('POST','url',true);

xhr.send();

xhr.onreadystatechange = function(){

if(xhr.readyState == 4 && xhr.status == 200){

var json = $.parseJSON(xhr.responseText);

}

}

}

**CSS属性**

**自动换行：**

word-wrap:break-word;

word-break:normal;

**溢出省略号**

overflow:hidden;

text-overflow:ellipsis;

white-space:nowrap

**JS表达式大全**

**一、校验数字的js正则表达式**

1 数字：^[0-9]\*$

2 n位的数字：^d{n}$

3 至少n位的数字：^d{n,}$

4 m-n位的数字：^d{m,n}$

5 零和非零开头的数字：^(0|[1-9][0-9]\*)$

6 非零开头的最多带两位小数的数字：^([1-9][0-9]\*)+(.[0-9]{1,2})?$

7 带1-2位小数的正数或负数：^(-)?d+(.d{1,2})?$

8 正数、负数、和小数：^(-|+)?d+(.d+)?$

9 有两位小数的正实数：^[0-9]+(.[0-9]{2})?$

10 有1~3位小数的正实数：^[0-9]+(.[0-9]{1,3})?$

11 非零的正整数：^[1-9]d\*$ 或 ^([1-9][0-9]\*){1,3}$ 或 ^+?[1-9][0-9]\*$

12 非零的负整数：^-[1-9][]0-9"\*$ 或 ^-[1-9]d\*$

13 非负整数：^d+$ 或 ^[1-9]d\*|0$

14 非正整数：^-[1-9]d\*|0$ 或 ^((-d+)|(0+))$

15 非负浮点数：^d+(.d+)?$ 或 ^[1-9]d\*.d\*|0.d\*[1-9]d\*|0?.0+|0$

16 非正浮点数：^((-d+(.d+)?)|(0+(.0+)?))$ 或 ^(-([1-9]d\*.d\*|0.d\*[1-9]d\*))|0?.0+|0$

17 正浮点数：^[1-9]d\*.d\*|0.d\*[1-9]d\*$ 或 ^(([0-9]+.[0-9]\*[1-9][0-9]\*)|([0-9]\*[1-9][0-9]\*.[0-9]+)|([0-9]\*[1-9][0-9]\*))$

18 负浮点数：^-([1-9]d\*.d\*|0.d\*[1-9]d\*)$ 或 ^(-(([0-9]+.[0-9]\*[1-9][0-9]\*)|([0-9]\*[1-9][0-9]\*.[0-9]+)|([0-9]\*[1-9][0-9]\*)))$

19 浮点数：^(-?d+)(.d+)?$ 或 ^-?([1-9]d\*.d\*|0.d\*[1-9]d\*|0?.0+|0)$

**二、校验字符的js正则表达式**

1 汉字：^[u4e00-u9fa5]{0,}$

2 英文和数字：^[A-Za-z0-9]+$ 或 ^[A-Za-z0-9]{4,40}$

3 长度为3-20的所有字符：^.{3,20}$

4 由26个英文字母组成的字符串：^[A-Za-z]+$

5 由26个大写英文字母组成的字符串：^[A-Z]+$

6 由26个小写英文字母组成的字符串：^[a-z]+$

7 由数字和26个英文字母组成的字符串：^[A-Za-z0-9]+$

8 由数字、26个英文字母或者下划线组成的字符串：^w+$ 或 ^w{3,20}$

9 中文、英文、数字包括下划线：^[u4E00-u9FA5A-Za-z0-9\_]+$

10 中文、英文、数字但不包括下划线等符号：^[u4E00-u9FA5A-Za-z0-9]+$ 或 ^[u4E00-u9FA5A-Za-z0-9]{2,20}$

11 可以输入含有^%&',;=?$"等字符：[^%&',;=?$x22]+

12 禁止输入含有~的字符：[^~x22]+

**三、特殊需求js正则表达式**

1 Email邮箱地址：^w+([-+.]w+)\*@w+([-.]w+)\*.w+([-.]w+)\*$  另外一个：  /^[A-Za-z0-9.\_%-]+@([A-Za-z0-9-]+.)+[A-Za-z]{2,4}$/

2 url域名：[a-zA-Z0-9][-a-zA-Z0-9]{0,62}(/.[a-zA-Z0-9][-a-zA-Z0-9]{0,62})+/.?

3 InternetURL：[a-zA-z]+://[^s]\* 或 ^http://([w-]+.)+[w-]+(/[w-./?%&=]\*)?$

4 手机号码：^(13[0-9]|14[5|7]|15[0|1|2|3|5|6|7|8|9]|18[0|1|2|3|5|6|7|8|9])d{8}$  另外一个：/^1[3|4|5|7|8]d{9}$/

5 电话号码("XXX-XXXXXXX"、"XXXX-XXXXXXXX"、"XXX-XXXXXXX"、"XXX-XXXXXXXX"、"XXXXXXX"和"XXXXXXXX)：  
       ^((d{3,4}-)|d{3.4}-)?d{7,8}$

6 国内电话号码(0511-4405222、021-87888822)：d{3}-d{8}|d{4}-d{7}     //[懒人建站](http://www.51xuediannao.com/)http://www.51xuediannao.com/

7 身份证号：

15或18位身份证：^d{15}|d{18}$

15位身份证：^[1-9]d{7}((0d)|(1[0-2]))(([0|1|2]d)|3[0-1])d{3}$

18位身份证：^[1-9]d{5}[1-9]d{3}((0d)|(1[0-2]))(([0|1|2]d)|3[0-1])d{4}$

8 短身份证号码(数字、字母x结尾)：^([0-9]){7,18}(x|X)?$ 或 ^d{8,18}|[0-9x]{8,18}|[0-9X]{8,18}?$

9 帐号是否合法(字母开头，允许5-16字节，允许字母数字下划线)：^[a-zA-Z][a-zA-Z0-9\_]{4,15}$

10 密码(以字母开头，长度在6~18之间，只能包含字母、数字和下划线)：^[a-zA-Z]w{5,17}$

11 强密码(必须包含大小写字母和数字的组合，不能使用特殊字符，长度在8-10之间)：^(?=.\*d)(?=.\*[a-z])(?=.\*[A-Z]).{8,10}$

12 日期格式：^d{4}-d{1,2}-d{1,2}

13 一年的12个月(01～09和1～12)：^(0?[1-9]|1[0-2])$

14 一个月的31天(01～09和1～31)：^((0?[1-9])|((1|2)[0-9])|30|31)$

**15 钱的输入格式：**

1.有四种钱的表示形式我们可以接受:"10000.00" 和 "10,000.00", 和没有 "分" 的 "10000" 和 "10,000"：^[1-9][0-9]\*$

2.这表示任意一个不以0开头的数字,但是,这也意味着一个字符"0"不通过,所以我们采用下面的形式：^(0|[1-9][0-9]\*)$

3.一个0或者一个不以0开头的数字.我们还可以允许开头有一个负号：^(0|-?[1-9][0-9]\*)$

4.这表示一个0或者一个可能为负的开头不为0的数字.让用户以0开头好了.把负号的也去掉,因为钱总不能是负的吧.下面我们要加的是说明可能的小数部分：^[0-9]+(.[0-9]+)?$

5.必须说明的是,小数点后面至少应该有1位数,所以"10."是不通过的,但是 "10" 和 "10.2" 是通过的：^[0-9]+(.[0-9]{2})?$

6.这样我们规定小数点后面必须有两位,如果你认为太苛刻了,可以这样：^[0-9]+(.[0-9]{1,2})?$

7.这样就允许用户只写一位小数.下面我们该考虑数字中的逗号了,我们可以这样：^[0-9]{1,3}(,[0-9]{3})\*(.[0-9]{1,2})?$

8.1到3个数字,后面跟着任意个 逗号+3个数字,逗号成为可选,而不是必须：^([0-9]+|[0-9]{1,3}(,[0-9]{3})\*)(.[0-9]{1,2})?$

备注：这就是最终结果了,别忘了"+"可以用"\*"替代如果你觉得空字符串也可以接受的话(奇怪,为什么?)最后,别忘了在用函数时去掉去掉那个反斜杠,一般的错误都在这里

9 xml文件：^([a-zA-Z]+-?)+[a-zA-Z0-9]+\.[x|X][m|M][l|L]$

10 中文字符的正则表达式：[u4e00-u9fa5]

11 双字节字符：[^x00-xff]    (包括汉字在内，可以用来计算字符串的长度(一个双字节字符长度计2，ASCII字符计1))

12 空白行的正则表达式： s\*    (可以用来删除空白行)

29 HTML标记的正则表达式：<(S\*?)[^>]\*>.\*?</1>|<.\*? />

13 首尾空白字符的正则表达式：^s\*|s\*$或(^s\*)|(s\*$)    (可以用来删除行首行尾的空白字符(包括空格、制表符、换页符等等)，非常有用的表达式)

14 腾讯QQ号：[1-9][0-9]{4,}    (腾讯QQ号从10000开始)

15 中国邮政编码：[1-9]d{5}(?!d)    (中国邮政编码为6位数字)

16 IP地址：d+.d+.d+.d+    (提取IP地址时有用)  
17 不能为空：/S/  
19 匹配空行：^(s\*)

**JS方法大全**

[返回主页](https://www.cnblogs.com/ranran/)

一、基础知识

1 创建脚本块

1: <script language=”JavaScript”>

2: JavaScript code goes here

3: </script>

2 隐藏脚本代码

1: <script language=”JavaScript”>

2: <!--

3: document.write(“Hello”);

4: // -->

5: </script>

在不支持JavaScript的浏览器中将不执行相关代码

3 浏览器不支持的时候显示

1: <noscript>

2: Hello to the non-JavaScript browser.

3: </noscript>

4 链接外部脚本文件

1: <script language=”JavaScript” src="/”filename.js"”></script>

5 注释脚本

1: // This is a comment

2: document.write(“Hello”); // This is a comment

3: /\*

4: All of this

5: is a comment

6: \*/

6 输出到浏览器

1: document.write(“<strong>Hello</strong>”);

7 定义变量

1: var myVariable = “some value”;

8 字符串相加

1: var myString = “String1” + “String2”;

9 字符串搜索

1: <script language=”JavaScript”>

2: <!--

3: var myVariable = “Hello there”;

4: var therePlace = myVariable.search(“there”);

5: document.write(therePlace);

6: // -->

7: </script>

10 字符串替换

1: thisVar.replace(“Monday”,”Friday”);

11 格式化字串

1: <script language=”JavaScript”>

2: <!--

3: var myVariable = “Hello there”;

4: document.write(myVariable.big() + “<br>”);

5: document.write(myVariable.blink() + “<br>”);

6: document.write(myVariable.bold() + “<br>”);

7: document.write(myVariable.fixed() + “<br>”);

8: document.write(myVariable.fontcolor(“red”) + “<br>”);

9: document.write(myVariable.fontsize(“18pt”) + “<br>”);

10: document.write(myVariable.italics() + “<br>”);

11: document.write(myVariable.small() + “<br>”);

12: document.write(myVariable.strike() + “<br>”);

13: document.write(myVariable.sub() + “<br>”);

14: document.write(myVariable.sup() + “<br>”);

15: document.write(myVariable.toLowerCase() + “<br>”);

16: document.write(myVariable.toUpperCase() + “<br>”);

17:

18: var firstString = “My String”;

19: var finalString = firstString.bold().toLowerCase().fontcolor(“red”);

20: // -->

21: </script>

12 创建数组

1: <script language=”JavaScript”>

2: <!--

3: var myArray = new Array(5);

4: myArray[0] = “First Entry”;

5: myArray[1] = “Second Entry”;

6: myArray[2] = “Third Entry”;

7: myArray[3] = “Fourth Entry”;

8: myArray[4] = “Fifth Entry”;

9: var anotherArray = new Array(“First Entry”,”Second Entry”,”Third Entry”,”Fourth Entry”,”Fifth Entry”);

10: // -->

11: </script>

13 数组排序

1: <script language=”JavaScript”>

2: <!--

3: var myArray = new Array(5);

4: myArray[0] = “z”;

5: myArray[1] = “c”;

6: myArray[2] = “d”;

7: myArray[3] = “a”;

8: myArray[4] = “q”;

9: document.write(myArray.sort());

10: // -->

11: </script>

14 分割字符串

1: <script language=”JavaScript”>

2: <!--

3: var myVariable = “a,b,c,d”;

4: var stringArray = myVariable.split(“,”);

5: document.write(stringArray[0]);

6: document.write(stringArray[1]);

7: document.write(stringArray[2]);

8: document.write(stringArray[3]);

9: // -->

10: </script>

15 弹出警告信息

1: <script language=”JavaScript”>

2: <!--

3: window.alert(“Hello”);

4: // -->

5: </script>

16 弹出确认框

1: <script language=”JavaScript”>

2: <!--

3: var result = window.confirm(“Click OK to continue”);

4: // -->

5: </script>

17 定义函数

1: <script language=”JavaScript”>

2: <!--

3: function multiple(number1,number2) {

4: var result = number1 \* number2;

5: return result;

6: }

7: // -->

8: </script>

18 调用JS函数

1: <a href=”#” onClick=”functionName()”>Link text</a>

2: <a href="/”javascript:functionName"()”>Link text</a>

19 在页面加载完成后执行函数

1: <body onLoad=”functionName();”>

2: Body of the page

3: </body>

20 条件判断

1: <script>

2: <!--

3: var userChoice = window.confirm(“Choose OK or Cancel”);

4: var result = (userChoice == true) ? “OK” : “Cancel”;

5: document.write(result);

6: // -->

7: </script>

21 指定次数循环

1: <script>

2: <!--

3: var myArray = new Array(3);

4: myArray[0] = “Item 0”;

5: myArray[1] = “Item 1”;

6: myArray[2] = “Item 2”;

7: for (i = 0; i < myArray.length; i++) {

8: document.write(myArray[i] + “<br>”);

9: }

10: // -->

11: </script>

22 设定将来执行

1: <script>

2: <!--

3: function hello() {

4: window.alert(“Hello”);

5: }

6: window.setTimeout(“hello()”,5000);

7: // -->

8: </script>

23 定时执行函数

1: <script>

2: <!--

3: function hello() {

4: window.alert(“Hello”);

5: window.setTimeout(“hello()”,5000);

6: }

7: window.setTimeout(“hello()”,5000);

8: // -->

9: </script>

24 取消定时执行

1: <script>

2: <!--

3: function hello() {

4: window.alert(“Hello”);

5: }

6: var myTimeout = window.setTimeout(“hello()”,5000);

7: window.clearTimeout(myTimeout);

8: // -->

9: </script>

25 在页面卸载时候执行函数

1: <body onUnload=”functionName();”>

2: Body of the page

3: </body>

JavaScript就这么回事2：浏览器输出

26 访问document对象

1: <script language=”JavaScript”>

2: var myURL = document.URL;

3: window.alert(myURL);

4: </script>

27 动态输出HTML

1: <script language=”JavaScript”>

2: document.write(“<p>Here’s some information about this document:</p>”);

3: document.write(“<ul>”);

4: document.write(“<li>Referring Document: “ + document.referrer + “</li>”);

5: document.write(“<li>Domain: “ + document.domain + “</li>”);

6: document.write(“<li>URL: “ + document.URL + “</li>”);

7: document.write(“</ul>”);

8: </script>

28 输出换行

1: document.writeln(“<strong>a</strong>”);

2: document.writeln(“b”);

29 输出日期

1: <script language=”JavaScript”>

2: var thisDate = new Date();

3: document.write(thisDate.toString());

4: </script>

30 指定日期的时区

1: <script language=”JavaScript”>

2: var myOffset = -2;

3: var currentDate = new Date();

4: var userOffset = currentDate.getTimezoneOffset()/60;

5: var timeZoneDifference = userOffset - myOffset;

6: currentDate.setHours(currentDate.getHours() + timeZoneDifference);

7: document.write(“The time and date in Central Europe is: “ + currentDate.toLocaleString());

8: </script>

31 设置日期输出格式

1: <script language=”JavaScript”>

2: var thisDate = new Date();

3: var thisTimeString = thisDate.getHours() + “:” + thisDate.getMinutes();

4: var thisDateString = thisDate.getFullYear() + “/” + thisDate.getMonth() + “/” + thisDate.getDate();

5: document.write(thisTimeString + “ on “ + thisDateString);

6: </script>

32 读取URL参数

1: <script language=”JavaScript”>

2: var urlParts = document.URL.split(“?”);

3: var parameterParts = urlParts[1].split(“&”);

4: for (i = 0; i < parameterParts.length; i++) {

5: var pairParts = parameterParts[i].split(“=”);

6: var pairName = pairParts[0];

7: var pairValue = pairParts[1];

8: document.write(pairName + “ ：“ +pairValue );

9: }

10: </script>

你还以为HTML是无状态的么？

33 打开一个新的document对象

1: <script language=”JavaScript”>

2: function newDocument() {

3: document.open();

4: document.write(“<p>This is a New Document.</p>”);

5: document.close();

6: }

7: </script>

34 页面跳转

1: <script language=”JavaScript”>

2: window.location = “http://www.liu21st.com/”;

3: </script>

35 添加网页加载进度窗口

1: <html>

2: <head>

3: <script language=’javaScript’>

4: var placeHolder = window.open(’holder.html’,’placeholder’,’width=200,height=200’);

5: </script>

6: <title>The Main Page</title>

7: </head>

8: <body onLoad=’placeHolder.close()’>

9: <p>This is the main page</p>

10: </body>

11: </html>

三、图像

36 读取图像属性

1: <img src="http://www.knowsky.com/”image1.jpg"” name=”myImage”>

2: <a href=”# ” onClick=”window.alert(document.myImage.width)”>Width</a>

3:

37 动态加载图像

1: <script language=”JavaScript”>

2: myImage = new Image;

3: myImage.src = “Tellers1.jpg”;

4: </script>

38 简单的图像替换

1: <script language=”JavaScript”>

2: rollImage = new Image;

3: rollImage.src = “rollImage1.jpg”;

4: defaultImage = new Image;

5: defaultImage.src = “image1.jpg”;

6: </script>

7: <a href="/”myUrl"” onMouseOver=”document.myImage.src = rollImage.src;”

8: onMouseOut=”document.myImage.src = defaultImage.src;”>

9: <img src="http://www.knowsky.com/”image1.jpg"” name=”myImage” width=100 height=100 border=0>

39 随机显示图像

1: <script language=”JavaScript”>

2: var imageList = new Array;

3: imageList[0] = “image1.jpg”;

4: imageList[1] = “image2.jpg”;

5: imageList[2] = “image3.jpg”;

6: imageList[3] = “image4.jpg”;

7: var imageChoice = Math.floor(Math.random() \* imageList.length);

8: document.write(‘<img src=”’ + imageList[imageChoice] + ‘“>’);

9: </script>

40 函数实现的图像替换

1: <script language=”JavaScript”>

2: var source = 0;

3: var replacement = 1;

4: function createRollOver(originalImage,replacementImage) {

5: var imageArray = new Array;

6: imageArray[source] = new Image;

7: imageArray[source].src = originalImage;

8: imageArray[replacement] = new Image;

9: imageArray[replacement].src = replacementImage;

10: return imageArray;

11: }

12: var rollImage1 = createRollOver(“image1.jpg”,”rollImage1.jpg”);

13: </script>

14: <a href=”#” onMouseOver=”document.myImage1.src = rollImage1[replacement].src;”

15: onMouseOut=”document.myImage1.src = rollImage1[source].src;”>

16: <img src="http://www.knowsky.com/”image1.jpg"” width=100 name=”myImage1” border=0>

17: </a>

41 创建幻灯片

1: <script language=”JavaScript”>

2: var imageList = new Array;

3: imageList[0] = new Image;

4: imageList[0].src = “image1.jpg”;

5: imageList[1] = new Image;

6: imageList[1].src = “image2.jpg”;

7: imageList[2] = new Image;

8: imageList[2].src = “image3.jpg”;

9: imageList[3] = new Image;

10: imageList[3].src = “image4.jpg”;

11: function slideShow(imageNumber) {

12: document.slideShow.src = imageList[imageNumber].src;

13: imageNumber += 1;

14: if (imageNumber < imageList.length) {

15: window.setTimeout(“slideShow(“ + imageNumber + “)”,3000);

16: }

17: }

18: </script>

19: </head>

20: <body onLoad=”slideShow(0)”>

21: <img src="http://www.knowsky.com/”image1.jpg"” width=100 name=”slideShow”>

42 随机广告图片

1: <script language=”JavaScript”>

2: var imageList = new Array;

3: imageList[0] = “image1.jpg”;

4: imageList[1] = “image2.jpg”;

5: imageList[2] = “image3.jpg”;

6: imageList[3] = “image4.jpg”;

7: var urlList = new Array;

8: urlList[0] = “http://some.host/”;

9: urlList[1] = “http://another.host/”;

10: urlList[2] = “http://somewhere.else/”;

11: urlList[3] = “http://right.here/”;

12: var imageChoice = Math.floor(Math.random() \* imageList.length);

13: document.write(‘<a href=”’ + urlList[imageChoice] + ‘“><img src=”’ + imageList[imageChoice] + ‘“></a>’);

14: </script>

四、表单

还是先继续写完JS就这么回事系列吧～

43 表单构成

1: <form method=”post” action=”target.html” name=”thisForm”>

2: <input type=”text” name=”myText”>

3: <select name=”mySelect”>

4: <option value=”1”>First Choice</option>

5: <option value=”2”>Second Choice</option>

6: </select>

7: <br>

8: <input type=”submit” value=”Submit Me”>

9: </form>

44 访问表单中的文本框内容

1: <form name=”myForm”>

2: <input type=”text” name=”myText”>

3: </form>

4: <a href=’#’ onClick=’window.alert(document.myForm.myText.value);’>Check Text Field</a>

45 动态复制文本框内容

1: <form name=”myForm”>

2: Enter some Text: <input type=”text” name=”myText”><br>

3: Copy Text: <input type=”text” name=”copyText”>

4: </form>

5: <a href=”#” onClick=”document.myForm.copyText.value =

6: document.myForm.myText.value;”>Copy Text Field</a>

46 侦测文本框的变化

1: <form name=”myForm”>

2: Enter some Text: <input type=”text” name=”myText” onChange=”alert(this.value);”>

3: </form>

47 访问选中的Select

1: <form name=”myForm”>

2: <select name=”mySelect”>

3: <option value=”First Choice”>1</option>

4: <option value=”Second Choice”>2</option>

5: <option value=”Third Choice”>3</option>

6: </select>

7: </form>

8: <a href=’#’ onClick=’alert(document.myForm.mySelect.value);’>Check Selection List</a>

48 动态增加Select项

1: <form name=”myForm”>

2: <select name=”mySelect”>

3: <option value=”First Choice”>1</option>

4: <option value=”Second Choice”>2</option>

5: </select>

6: </form>

7: <script language=”JavaScript”>

8: document.myForm.mySelect.length++;

9: document.myForm.mySelect.options[document.myForm.mySelect.length - 1].text = “3”;

10: document.myForm.mySelect.options[document.myForm.mySelect.length - 1].value = “Third Choice”;

11: </script>

49 验证表单字段

1: <script language=”JavaScript”>

2: function checkField(field) {

3: if (field.value == “”) {

4: window.alert(“You must enter a value in the field”);

5: field.focus();

6: }

7: }

8: </script>

9: <form name=”myForm” action=”target.html”>

10: Text Field: <input type=”text” name=”myField”onBlur=”checkField(this)”>

11: <br><input type=”submit”>

12: </form>

50 验证Select项

1: function checkList(selection) {

2: if (selection.length == 0) {

3: window.alert(“You must make a selection from the list.”);

4: return false;

5: }

6: return true;

7: }

51 动态改变表单的action

1: <form name=”myForm” action=”login.html”>

2: Username: <input type=”text” name=”username”><br>

3: PassWord: <input type=”password” name=”password”><br>

4: <input type=”button” value=”Login” onClick=”this.form.submit();”>

5: <input type=”button” value=”Register” onClick=”this.form.action = ‘register.html’; this.form.submit();”>

6: <input type=”button” value=”Retrieve Password” onClick=”this.form.action = ‘password.html’; this.form.submit();”>

7: </form>

52 使用图像按钮

1: <form name=”myForm” action=”login.html”>

2: Username: <input type=”text” name=”username”><br>

3: Password: <input type=”password”name=”password”><br>

4: <input type=”image” src="http://www.knowsky.com/”login.gif"” value=”Login”>

5: </form>

6:

53 表单数据的加密

1: <SCRIPT LANGUAGE=’JavaScript’>

2: <!--

3: function encrypt(item) {

4: var newItem = ’’;

5: for (i=0; i < item.length; i++) {

6: newItem += item.charCodeAt(i) + ’.’;

7: }

8: return newItem;

9: }

10: function encryptForm(myForm) {

11: for (i=0; i < myForm.elements.length; i++) {

12: myForm.elements[i].value = encrypt(myForm.elements[i].value);

13: }

14: }

15:

16: //-->

17: </SCRIPT>

18: <form name=’myForm’ onSubmit=’encryptForm(this); window.alert(this.myField.value);’>

19: Enter Some Text: <input type=text name=myField><input type=submit>

20: </form>

五、窗口和框架

54 改变浏览器状态栏文字提示

1: <script language=”JavaScript”>

2: window.status = “A new status message”;

3: </script>

55 弹出确认提示框

1: <script language=”JavaScript”>

2: var userChoice = window.confirm(“Click OK or Cancel”);

3: if (userChoice) {

4: document.write(“You chose OK”);

5: } else {

6: document.write(“You chose Cancel”);

7: }

8: </script>

56 提示输入

1: <script language=”JavaScript”>

2: var userName = window.prompt(“Please Enter Your Name”,”Enter Your Name Here”);

3: document.write(“Your Name is “ + userName);

4: </script>

57 打开一个新窗口

1: //打开一个名称为myNewWindow的浏览器新窗口

2: <script language=”JavaScript”>

3: window.open(“http://www.liu21st.com/”,”myNewWindow”);

4: </script>

58 设置新窗口的大小

1: <script language=”JavaScript”>

2: window.open(“http://www.liu21st.com/”,”myNewWindow”,’height=300,width=300’);

3: </script>

59 设置新窗口的位置

1: <script language=”JavaScript”>

2: window.open(“http://www.liu21st.com/”,”myNewWindow”,’height=300,width=300,left=200,screenX=200,top=100,screenY=100’);

3: </script>

60 是否显示工具栏和滚动栏

1: <script language=”JavaScript”>

2: window.open(“http:

61 是否可以缩放新窗口的大小

1: <script language=”JavaScript”>

2: window.open(’http://www.liu21st.com/’ , ’myNewWindow’, ’resizable=yes’ );</script>

62 加载一个新的文档到当前窗口

1: <a href=’#’ onClick=’document.location = ’125a.html’;’ >Open New Document</a>

63 设置页面的滚动位置

1: <script language=”JavaScript”>

2: if (document.all) { //如果是IE浏览器则使用scrollTop属性

3: document.body.scrollTop = 200;

4: } else { //如果是NetScape浏览器则使用pageYOffset属性

5: window.pageYOffset = 200;

6: }</script>

64 在IE中打开全屏窗口

1: <a href=’#’ onClick=”window.open(’http://www.juxta.com/’,’newWindow’,’fullScreen=yes’);”>Open a full-screen window</a>

65 新窗口和父窗口的操作

1: <script language=”JavaScript”>

2: //定义新窗口

3: var newWindow = window.open(“128a.html”,”newWindow”);

4: newWindow.close(); //在父窗口中关闭打开的新窗口

5: </script>

6: 在新窗口中关闭父窗口

7: window.opener.close()

66 往新窗口中写内容

1: <script language=”JavaScript”>

2: var newWindow = window.open(“”,”newWindow”);

3: newWindow.document.open();

4: newWindow.document.write(“This is a new window”);

5: newWIndow.document.close();

6: </script>

67 加载页面到框架页面

1: <frameset cols=”50%,\*”>

2: <frame name=”frame1” src="/”135a.html"”>

3: <frame name=”frame2” src="/”about:blank"”>

4: </frameset>

5: 在frame1中加载frame2中的页面

6: parent.frame2.document.location = “135b.html”;

68 在框架页面之间共享脚本

如果在frame1中html文件中有个脚本

1: function doAlert() {

2: window.alert(“Frame 1 is loaded”);

3: }

那么在frame2中可以如此调用该方法

1: <body onLoad=”parent.frame1.doAlert();”>

2: This is frame 2.

3: </body>

69 数据公用

可以在框架页面定义数据项，使得该数据可以被多个框架中的页面公用

1: <script language=”JavaScript”>

2: var persistentVariable = “This is a persistent value”;

3: </script>

4: <frameset cols=”50%,\*”>

5: <frame name=”frame1” src="/”138a.html"”>

6: <frame name=”frame2” src="/”138b.html"”>

7: </frameset>

这样在frame1和frame2中都可以使用变量persistentVariable

70 框架代码库

根据以上的一些思路，我们可以使用一个隐藏的框架页面来作为整个框架集的代码库

1: <frameset cols=”0,50%,\*”>

2: <frame name=”codeFrame” src="/”140code.html"”>

3: <frame name=”frame1” src="/”140a.html"”>

4: <frame name=”frame2” src="/”140b.html"”>

5: </frameset>

shift：删除原数组第一项，并返回删除元素的值；如果数组为空则返回undefined   
var a = [1,2,3,4,5];   
var b = a.shift(); //a：[2,3,4,5] b：1

unshift：将参数添加到原数组开头，并返回数组的长度   
var a = [1,2,3,4,5];   
var b = a.unshift(-2,-1); //a：[-2,-1,1,2,3,4,5] b：7   
注：在IE6.0下测试返回值总为undefined，FF2.0下测试返回值为7，所以这个方法的返回值不可靠，需要用返回值时可用splice代替本方法来使用。

pop：删除原数组最后一项，并返回删除元素的值；如果数组为空则返回undefined   
var a = [1,2,3,4,5];   
var b = a.pop(); //a：[1,2,3,4] b：5

push：将参数添加到原数组末尾，并返回数组的长度   
var a = [1,2,3,4,5];   
var b = a.push(6,7); //a：[1,2,3,4,5,6,7] b：7

concat：返回一个新数组，是将参数添加到原数组中构成的   
var a = [1,2,3,4,5];   
var b = a.concat(6,7); //a：[1,2,3,4,5] b：[1,2,3,4,5,6,7]

splice(start,deleteCount,val1,val2,...)：从start位置开始删除deleteCount项，并从该位置起插入val1,val2,...   
var a = [1,2,3,4,5];   
var b = a.splice(2,2,7,8,9); //a：[1,2,7,8,9,5] b：[3,4]   
var b = a.splice(0,1); //同shift   
a.splice(0,0,-2,-1); var b = a.length; //同unshift   
var b = a.splice(a.length-1,1); //同pop   
a.splice(a.length,0,6,7); var b = a.length; //同push

reverse：将数组反序   
var a = [1,2,3,4,5];   
var b = a.reverse(); //a：[5,4,3,2,1] b：[5,4,3,2,1]

sort(orderfunction)：按指定的参数对数组进行排序   
var a = [1,2,3,4,5];   
var b = a.sort(); //a：[1,2,3,4,5] b：[1,2,3,4,5]

slice(start,end)：返回从原数组中指定开始下标到结束下标之间的项组成的新数组   
var a = [1,2,3,4,5];   
var b = a.slice(2,5); //a：[1,2,3,4,5] b：[3,4,5]

join(separator)：将数组的元素组起一个字符串，以separator为分隔符，省略的话则用默认用逗号为分隔符   
var a = [1,2,3,4,5];   
var b = a.join("|"); //a：[1,2,3,4,5] b："1|2|3|4|5"

窗体顶端

**Js字符串操作函数大全**

/\*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
                        字符串函数扩充                                   
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*/

/\*  
===========================================  
//去除左边的空格  
===========================================

\*/  
String.prototype.LTrim = function()  
{  
        return this.replace(/(^\s\*)/g, "");  
}

/\*  
===========================================  
//去除右边的空格  
===========================================  
\*/  
String.prototype.Rtrim = function()  
{  
        return this.replace(/(\s\*$)/g, "");  
}

/\*  
===========================================  
//去除前后空格  
===========================================  
\*/  
String.prototype.Trim = function()  
{  
        return this.replace(/(^\s\*)|(\s\*$)/g, "");  
}

/\*  
===========================================  
//得到左边的字符串  
===========================================  
\*/  
String.prototype.Left = function(len)  
{

        if(isNaN(len)||len==null)  
        {  
                len = this.length;  
        }  
        else  
        {  
                if(parseInt(len)<0||parseInt(len)>this.length)  
                {  
                        len = this.length;  
                }  
        }  
          
        return this.substr(0,len);  
}

/\*  
===========================================  
//得到右边的字符串  
===========================================  
\*/  
String.prototype.Right = function(len)  
{

        if(isNaN(len)||len==null)  
        {  
                len = this.length;  
        }  
        else  
        {  
                if(parseInt(len)<0||parseInt(len)>this.length)  
                {  
                        len = this.length;  
                }  
        }  
          
        return this.substring(this.length-len,this.length);  
}

/\*  
===========================================  
//得到中间的字符串,注意从0开始  
===========================================  
\*/  
String.prototype.Mid = function(start,len)  
{  
        return this.substr(start,len);  
}

/\*  
===========================================  
//在字符串里查找另一字符串:位置从0开始  
===========================================  
\*/  
String.prototype.InStr = function(str)  
{

        if(str==null)  
        {  
                str = "";  
        }  
          
        return this.indexOf(str);  
}

/\*  
===========================================  
//在字符串里反向查找另一字符串:位置0开始  
===========================================  
\*/  
String.prototype.InStrRev = function(str)  
{

        if(str==null)  
        {  
                str = "";  
        }  
          
        return this.lastIndexOf(str);  
}

/\*  
===========================================  
//计算字符串打印长度  
===========================================  
\*/  
String.prototype.LengthW = function()  
{  
        return this.replace(/[^\x00-\xff]/g,"\*\*").length;  
}

/\*  
===========================================  
//是否是正确的IP地址  
===========================================  
\*/  
String.prototype.isIP = function()  
{

        var reSpaceCheck = /^(\d+)\.(\d+)\.(\d+)\.(\d+)$/;

        if (reSpaceCheck.test(this))  
        {  
                this.match(reSpaceCheck);  
                if (RegExp.$1 <= 255 && RegExp.$1 >= 0   
                 && RegExp.$2 <= 255 && RegExp.$2 >= 0   
                 && RegExp.$3 <= 255 && RegExp.$3 >= 0   
                 && RegExp.$4 <= 255 && RegExp.$4 >= 0)   
                {  
                        return true;       
                }  
                else  
                {  
                        return false;  
                }  
        }  
        else  
        {  
                return false;  
        }  
     
}

/\*  
===========================================  
//是否是正确的长日期  
===========================================  
\*/  
String.prototype.isLongDate = function()  
{  
        var r = this.replace(/(^\s\*)|(\s\*$)/g, "").match(/^(\d{1,4})(-|\/)(\d{1,2})\2(\d{1,2}) (\d{1,2}):(\d{1,2}):(\d{1,2})$/);   
        if(r==null)  
        {  
                return false;   
        }  
        var d = new Date(r[1], r[3]-1,r[4],r[5],r[6],r[7]);   
        return (d.getFullYear()==r[1]&&(d.getMonth()+1)==r[3]&&d.getDate()==r[4]&&d.getHours()==r[5]&&d.getMinutes()==r[6]&&d.getSeconds()==r[7]);

}

/\*  
===========================================  
//是否是正确的短日期  
===========================================  
\*/  
String.prototype.isShortDate = function()  
{  
        var r = this.replace(/(^\s\*)|(\s\*$)/g, "").match(/^(\d{1,4})(-|\/)(\d{1,2})\2(\d{1,2})$/);   
        if(r==null)  
        {  
                return false;   
        }  
        var d = new Date(r[1], r[3]-1, r[4]);   
        return (d.getFullYear()==r[1]&&(d.getMonth()+1)==r[3]&&d.getDate()==r[4]);  
}

/\*  
===========================================  
//是否是正确的日期  
===========================================  
\*/  
String.prototype.isDate = function()  
{  
        return this.isLongDate()||this.isShortDate();  
}

/\*  
===========================================  
//是否是手机  
===========================================  
\*/  
String.prototype.isMobile = function()  
{  
        return /^0{0,1}13[0-9]{9}$/.test(this);  
}

/\*  
===========================================  
//是否是邮件  
===========================================  
\*/  
String.prototype.isEmail = function()  
{  
        return /^\w+((-\w+)|(\.\w+))\*\@[A-Za-z0-9]+((\.|-)[A-Za-z0-9]+)\*\.[A-Za-z0-9]+$/.test(this);  
}

/\*  
===========================================  
//是否是邮编(中国)  
===========================================  
\*/

String.prototype.isZipCode = function()  
{  
        return /^[\\d]{6}$/.test(this);  
}

/\*  
===========================================  
//是否是有汉字  
===========================================  
\*/  
String.prototype.existChinese = function()  
{  
        //[\u4E00-\u9FA5]為漢字﹐[\uFE30-\uFFA0]為全角符號  
        return /^[\x00-\xff]\*$/.test(this);  
}

/\*  
===========================================  
//是否是合法的文件名/目录名  
===========================================  
\*/  
String.prototype.isFileName = function()  
{  
        return !/[\\\/\\*\?\|:"<>]/g.test(this);  
}

/\*  
===========================================  
//是否是有效链接  
===========================================  
\*/  
String.prototype.isUrl = function()  
{  
        return /^http[s]?:\/\/([\w-]+\.)+[\w-]+([\w-./?%&=]\*)?$/i.test(this);  
}

/\*  
===========================================  
//是否是有效的身份证(中国)  
===========================================  
\*/  
String.prototype.isIDCard = function()  
{  
        var iSum=0;  
        var info="";  
        var sId = this;

        var aCity={11:"北京",12:"天津",13:"河北",14:"山西",15:"内蒙古",21:"辽宁",22:"吉林",23:"黑龙江",31:"上海",32:"江苏",33:"浙江",34:"安徽",35:"福建",36:"江西",37:"山东",41:"河南",42:"湖北",43:"湖南",44:"广东",45:"广西",46:"海南",50:"重庆",51:"四川",52:"贵州",53:"云南",54:"西藏",61:"陕西",62:"甘肃",63:"青海",64:"宁夏",65:"新疆",71:"台湾",81:"香港",82:"澳门",91:"国外"};

        if(!/^\d{17}(\d|x)$/i.test(sId))  
        {  
                return false;  
        }  
        sId=sId.replace(/x$/i,"a");  
        //非法地区  
        if(aCity[parseInt(sId.substr(0,2))]==null)  
        {  
                return false;  
        }

        var sBirthday=sId.substr(6,4)+"-"+Number(sId.substr(10,2))+"-"+Number(sId.substr(12,2));

        var d=new Date(sBirthday.replace(/-/g,"/"))  
          
        //非法生日  
        if(sBirthday!=(d.getFullYear()+"-"+ (d.getMonth()+1) + "-" + d.getDate()))  
        {  
                return false;  
        }  
        for(var i = 17;i>=0;i--)   
        {  
                iSum += (Math.pow(2,i) % 11) \* parseInt(sId.charAt(17 - i),11);  
        }

        if(iSum%11!=1)  
        {  
                return false;  
        }  
        return true;

}

/\*  
===========================================  
//是否是有效的电话号码(中国)  
===========================================  
\*/  
String.prototype.isPhoneCall = function()  
{  
        return /(^[0-9]{3,4}\-[0-9]{3,8}$)|(^[0-9]{3,8}$)|(^\([0-9]{3,4}\)[0-9]{3,8}$)|(^0{0,1}13[0-9]{9}$)/.test(this);  
}

/\*  
===========================================  
//是否是数字  
===========================================  
\*/  
String.prototype.isNumeric = function(flag)  
{  
        //验证是否是数字  
        if(isNaN(this))  
        {

                return false;  
        }

        switch(flag)  
        {

                case null:        //数字  
                case "":  
                        return true;  
                case "+":        //正数  
                        return                /(^\+?|^\d?)\d\*\.?\d+$/.test(this);  
                case "-":        //负数  
                        return                /^-\d\*\.?\d+$/.test(this);  
                case "i":        //整数  
                        return                /(^-?|^\+?|\d)\d+$/.test(this);  
                case "+i":        //正整数  
                        return                /(^\d+$)|(^\+?\d+$)/.test(this);                          
                case "-i":        //负整数  
                        return                /^[-]\d+$/.test(this);  
                case "f":        //浮点数  
                        return                /(^-?|^\+?|^\d?)\d\*\.\d+$/.test(this);  
                case "+f":        //正浮点数  
                        return                /(^\+?|^\d?)\d\*\.\d+$/.test(this);                          
                case "-f":        //负浮点数  
                        return                /^[-]\d\*\.\d$/.test(this);                  
                default:        //缺省  
                        return true;                          
        }  
}

/\*  
===========================================  
//是否是颜色(#FFFFFF形式)  
===========================================  
\*/  
String.prototype.IsColor = function()  
{  
        var temp        = this;  
        if (temp=="") return true;  
        if (temp.length!=7) return false;  
        return (temp.search(/\#[a-fA-F0-9]{6}/) != -1);  
}

/\*  
===========================================  
//转换成全角  
===========================================  
\*/  
String.prototype.toCase = function()  
{  
        var tmp = "";  
        for(var i=0;i<this.length;i++)  
        {  
                if(this.charCodeAt(i)>0&&this.charCodeAt(i)<255)  
                {  
                        tmp += String.fromCharCode(this.charCodeAt(i)+65248);  
                }  
                else  
                {  
                        tmp += String.fromCharCode(this.charCodeAt(i));  
                }  
        }  
        return tmp  
}

/\*  
===========================================  
//对字符串进行Html编码  
===========================================  
\*/  
String.prototype.toHtmlEncode = function()  
{  
        var str = this;

        str=str.replace(/&/g,"&amp;");  
        str=str.replace(/</g,"&lt;");  
        str=str.replace(/>/g,"&gt;");  
        str=str.replace(/\'/g,"&apos;");  
        str=str.replace(/\"/g,"&quot;");  
        str=str.replace(/\n/g,"<br>");  
        str=str.replace(/\ /g,"&nbsp;");  
        str=str.replace(/\t/g,"&nbsp;&nbsp;&nbsp;&nbsp;");

        return str;  
}

/\*  
===========================================  
//转换成日期  
===========================================  
\*/  
String.prototype.toDate = function()  
{  
        try  
        {  
                return new Date(this.replace(/-/g, "\/"));  
        }  
        catch(e)  
        {  
                return null;  
        }  
}

1.Asc(x),Chr(x):转换字符，字符码

2。Filter:搜寻字符串数组中的特定字符串

格式：v=filter(x,s[,include[,compare]])

实例：

Dim x()={"kjwang","wangkj","peter"}

Dim v

v=filter(x,"kj") '结果v(0)="kjwang",v(1)="wangkj"

v=filter(x,"kj",false) '结果v(0)="peter"

v=filter(x,"kj",true,vbTextCompare) '不分大小写搜寻

3.InStr:寻找字符串位置(InstrRev:倒过来寻找字符串）

格式：

v=instr(x,y) '从x字符串第1个字符起找出y字符串出现的位置

v=instr(n,x,y) '从x字符串第n个字符起找出y字符串出现的位置

格式：

v=InstrRev(x,s[,n[,Compare]])

4。Join:将字符串连接

格式：v=join(x[,d]）’d为分隔字符

5。Len(x):计算字符串x的长度

格式：v=len(x)

6.Left(x,n):返回字符串x左边n个字符(对应Right(x,n))

7。Mid:读取字符串x中间的字符

格式：v=mid(x,n,m)

8。LTrim(x),RTim(x),Trim(x)去空白字符

9.Replace:字符串取代

格式：v=Replace(x,s,r)

实例：x="i saw a saw a saw"

v=replace(x,"saw","so") 'v="i so a so a so"

10.Split:字符串分割

格式：v=split(s[,d])

实例：v=split("vb.net,iis6.0,asp.net",",")

'结果v(0)="vb.net",v(1)="iis6.0",v(2)="asp.net"

11.StrReverse:反转字符串

实例：v=strreverse("kjwang") 'v="gnawjk"

12.UCase(x),LCase(x):变换英文字母的大小写

实例：x="hello,VB中文!"

v=UCase(x) 'v="HELLO,VB中文！"   
1.取出日期时间

－1）DateValue(x),TimeValue(x)

格式：v=DateValue(x) :取出的“日期”部分

　　　v=timevalue(x) '类上

－2）Year(x),Month(x),Day(x)

格式：v=Year(x)

　　　v=Month(x)

　　　v=Day(x)

　　　Hour(x),Minute(x),Second(x):取出时，分，秒

－3）DateSerial(Y,M,D):合并年、月、日成为日期

实例：Dim v

v=DateSerial(1996,10,5) 'v=1996/10/5

　　　TimeSerial(H,M,S):合并时、分、秒成为时间

2.Now:读取系统日期时间

格式：v=now

3.Timer:从凌晨12：00：00AM起所经过的秒数

格式：v=timer

4.DatePart(p,x):取出年、月、日等各部分的数值

实例：Dim X=#2/10/1996 16:45:30#

v=DatePart("yyyy",X) 'v=1996年

v=DatePart("m",X) 'v=2月

v=DatePart("d",X) 'v=10日

v=DatePart("h",X) 'v=16时

v=DatePart("n",X) 'v=45分

v=DatePart("s",X) 'v=30秒

v=DatePart("q",X) 'v=1季（第一季，春季）

v=DatePart("y",X) 'v=41天（1996年的第41天）

v=DatePart("ww",X) 'v=6周（1996年的第6周）

v=DatePart("w",X) 'v=7（第6周的第7天，即星期六）

5。DateAdd(p,n,x):加减若干时间后的新日期时间

格式：v=DateAdd(p,n,x) 'p值同上如："yyyy"、"m"等

实例：Dim x=#1/31/1996 16:45:30#

v=dateadd("yyyy",-3,x) '减3年，v=1993/1/31 16:45:30

6.DateDiff(p,x1,x2):计算两日期时间的差值 'p值同上

实例：dim x1=#12/31/1996 16:45:30#

x2=#2/10/1997 9:30:50#

v=datediff("yyyy",x1,x2) 'v=1年

7。FormatDateTime:日期时间的格式化

格式：v=formatdateyime(x[,日期格式])

日期格式值：DateFormat.GeneralDate　、　DateFormat.LongDate、

DateFotmat.ShortDate、DateFormat.LongTime、DateFormat.ShortTime

8.MonthName:返回月份名称

实例：v=monthname(5) 'v="五月"

9.WeekDayName:返回星期名称　’用法同8.

窗体底端