JavaScript 简介

Javascript 的作用

实现网页的动态效果

例如：下拉菜单、用户验证、滚动文字与图片，
与 php 搭配可以实现一些高级的应用，
比如去哪儿网的地名拼音查询到地名全名

Javascript 与 java

是两个公司开发的不同的产品。Java 是 SUN 公司推出的新一代面向对象的程序设计语言，特别适合于 Internet 应用程序开发，服务器端语言，与数据库打交道；

而 JavaScript 是 Netscape 公司的产品，javaScript 是一种浏览器脚本语言，由客户端浏览器解析执行。

如何将 javascript 嵌入网页

方法一：嵌入 html 页面

<script type=“text/javascrpt”></script>
可以放在<head></head>之间
也可以放在<body></body>内部
尽管也可以放在<html>之前或</html>之后，但 web 标准并不提倡这样做
IE 与 FF 默认脚本语言就是 javascript，所以 type 经常省略

第一个 javascript 程序

范例 t1.html
动态改变网页背景

```
<script language="javascript">
        document.bgColor="red";
        alert(document.bgColor);
        document.bgColor="green";
        alert(document.bgColor);
        document.bgColor="yellow";
        alert(document.bgColor);
</script>
```

方法二：链接外部的 js 文件

<script src=“aa.js”></script>
使用外部文件优点：
1、最大限度的实现重用，方便维护；
2、浏览器会缓存，如果文件大，节省下载时间。
步骤：
1、在 HTML 文件中写入<script src="js.js"></script>
2、建立文件：js.js，其内容如下：

```
document.bgColor="red";
alert(document.bgColor);
document.bgColor="green";
alert(document.bgColor);
document.bgColor="yellow";
alert(document.bgColor);
```

如何将 javascript 嵌入网页

```
<a href="javascript:alert(new Date());">javaScript</a>
<input type="button" value="test" onclick="alert(new Date())">
```

Javascript 错误调试

错误调试：

FF 安装 firebug 插件；

IE 点击黄色小叹号标；

显然 firebug 能够更详细的阐述错误信息

JavaScript 的基本语法

JavaScript 中的标识符

标识符是指 JavaScript 中定义的符号，例如， 变量名，函数名，数组名等。标识符可以由任意顺序的大小写字母、数字、下划线（_）和美元符号（$）组成，但标识符不能以数字开头，不能是 JavaScript 中的保留关键字。

合法的标识符举例：indentifler、username、user_name、_userName、$username

非法的标识符举例：int、98.3、Hello World

JavaScript 严格区分大小写

computer 和 Computer 是两个完全不同的符号

JavaScript 程序代码的格式

每条功能执行语句的最后必须用分号（;）结束，每个词之间用空格、制表符、换行符或大括号、小括号这样的分隔符隔开。

JavaScript 程序的注释

/*…….*/中可以嵌套“//”注释，但不能嵌套“/*….*/”

JavaScript 中保留的关键字

abstract	boolean	break	byte	case	catch	char	class
continue	default	if	for	float	finally	final	
false	extends	else	double	do	implements	import	
instance	int	interface		long	native	new	null
backage	this	synchronized		switch	super	static	
short	return	public	protected		private	throw	
throws	transient	true	try		void	volatile	
while							

基本数据类型

数值数据（number）

a）整数

十六进制以 0x 或 0X 开头， 例如 0x8a。

八进制必须以 0 开头，例如：0123。

十进制的第一位不能是 0（数字 0 除外），例如：123。

整型数据范围：-253~253，所以，一般不用考虑其范围问题

b)小数

12.32、 192.98、 5E7、4e5 等。

2 文本数据（string）

“this is JavaScript ppt”、’ abc’、“a”、“”。

字符串中的特殊字符，需要以反斜杠（\）后跟一个普通字符来表示，例如：

\r（回车符）、\n（换行符） \t（制表符）、

\’、 \”、 \\ .

单引号或双引号需要配对使用，另字符串包含单引号或双引号，需要转义

布尔值（boolean）

true 和 false. 真与假

强制数据类型转换：

-字符串转换为数字：parseInt("12abc");

转换结果为：12;

变量

定义一个变量，系统就会为之分配一块内存，程序可以用变量名来表示这块内存中的数据

声明变量要使用 var 关键字

例如：var name=“abc”;

声明变量的同时为其赋值

例如：name = “zhangsan”;

对已赋值的变量赋予一个其他类型的数据

例如：name = 1243;

不事先声明变量而直接使用

例如：x = 1234;

变量类型

变量类型测试：

使用 typeof（）;

```
<form name="frm">
        name:<input type="text" name="username" /><br />
        age:<input type="text" name="age" /><br /><!--在此输入一个数字
-->
        <input type="button" onclick="show()" value="显示年龄" />
</form>
<script>
        function show(){
                var age=document.frm.age.value;
                alert(typeof(age));//输出: string
                var newage=parseInt(age);
                alert(typeof(newage));//输出: number
        }
</script>
```

```
<script>
```

```
        var age="abec";
        var newage=parseInt(age);
        alert(newage);//输出：NaN
        if(isNaN(age)){//输出：age is not a number
                alert("age is not a number");
        }else{
                alert("age is a number");
        }
</script>
```

undefined（变量申明却未赋值的数据类型）

```
<script>
        var a;
        alert(typeof(a));//输出：undefined
</script>
```

运算符

算术运算符

+ 加法运算符或正值运算符，例如： x+5， ＋6。

“+”还能实现多个字符串的相加，也能将字符串与其它的数据类型连成一个新的字符串，条件是该表达式中至少有一个字符串，例如：“x”+123 的结果是“x123”.（范例）

```
<script>
        var a=3;
        var b="hello";
        var c=a+b;
        alert(typeof(c));//输出：string
        alert(c);//输出：3hello
</script>
```

- 减法运算符或负值运算符，例如： 7 - 3, -8.

* 乘法运算符，例如： 3*6.

/ 除法运算符，例如，9/4.

% 求模运算符（也就算术中的求余）5/2.

++ 将变量值加 1 后再将结果赋给这个变量。

“＋＋”有两种用法：++x, x++. 前者是变量在参与其它运算之前先将自己加 1 后再用新的值参与其它的运算，而后者是先用原值参与其它运算后，再将自己加 1，例如：b=++a 是 a 先自增，即 a 的值加 1 后，才赋值给 b；而 b=a++是先将 a 赋值给 b 后，a 再自增。

-- 将变量值减 1 后再将结果赋给这个变量，与++的用法一样。

赋值运算符

赋值运算符的作用是将一个值赋给一个变量，最常用的赋值运算符是“=”。还可以由“=”赋值运算符和其它一些运算符组合产生一些新的赋值运算符，例如，“+=”，”*=”等。

= 将一个值或表达式的结果赋给变量

例如： x = 3;

+= 例如：x += 3 等价于 x = x + 3;

-=　　例如：x -= 3 等价于 x = x - 3;

*=　　例如：x *= 3 等价于 x = x * 3;

/=　　例如：x /= 3 等价于 x = x / 3;

%=　例如：x %= 3 等价于 x = x % 3;

比较运算符

常见的比较运算符号：

> < >= <= != ==

注意：不要将比较运算符“==”误写成“=”。

逻辑运算符

&&

逻辑与，当左右两边操作数都为 true 时，返回 true，否则返回 false.

||

逻辑或，当左右两边操作数都为 false 时，返回 false,否则返回 true.

!

逻辑非，当操作数为 true 时返回 false，　否则返回 true.

位运算符

任何信息在计算机中都是以二进制的形式保存的，位运算用于对操作数中的每一个二进制位进行运算，包括位逻辑运算符和位移运算符。

&　两边操作数转换为二进制按位与；

|　　两边操作数转换为二进制按位或；

^两边操作数转换为二进制按位异或；

程序的流程控制

顺序结构

If 条件选择语句

1、 if(条件语句)

```
{
        执行语句;
}
```

多加： if(x == null) 或 if(typeof(x) ==“undefined”) 可以简写成 if(!x).

2、if(条件语句)

```
{
    执行语句块 1;
}
else
{
    执行语句 2;
}
```

（范例 9：闰年）

```
<form name="frm">
```

```
        year:<input type="text" name="year" /><br />
        <input type="button" value="计算" onclick="js()">
</form>
<script>
        function js(){
                var year=document.frm.year.value;
                if(year%4==0 && year%100!=0 || year%400==0){
                        alert(year+"是闰年。");
                }else{
                        alert(year+"是平年。");
                }
        }
</script>
```

多加： 变量 = 布尔表达式? 语句1：语句2;

例如： y = x >0 ? x : -x;

3、if(条件语句1) {

执行语句1;

}

else if(条件语句2)

{

执行语句2;

}

.....

else if(条件语句n)

{

执行语句n;

}

else

{

执行语句n+1;

} （范例10：问好）

```
<script>
        var d=new Date();
        var hours=d.getHours();
        if(hours <5){
                alert("Good wee hours!");
        }else if(hours <8){
                alert("Good morning!");
        }else if(hours <11){
                alert("Good forenoon!");
        }else if(hours <14){
                alert("Good noon!");
        }else if(hours <20){
                alert("Good afternoon!");
```

```
        }else{
                alert("Good evening!");
        }
</script>
```

If 语句的嵌套

```
<form name="frm">
        name:<input type="text" name="uname" /><br />
        sex:<input type="text" name="sex"><br />
        age:<input type="text" name="age"><br />
        <input type="button" value="是否退休" onclick="show()">
</form>
<script>
        function show(){
                var sex=document.frm.sex.value;
                var age=document.frm.age.value;
                if(sex=="男"){
                        if(age >60){
                                alert("你已经退休了，退休"+(age-60)+"年了。");

                        }else{
                                alert("你没有退休，还有"+(60-age)+"年退休");

                        }
                }else{
                        if(age>55){
                                alert("你已经退休了，退休"+(age-55)+"年了");

                        }else{
                                alert("你没有退休，还有年"+(55-age)+"退休");

                        }
                }
        }
</script>
```

Switch 选择语句

```
switch(表达式)
{
      case 取值1:
              语句块1;
               break;
      case 取值2:
              语句块2;
               break;
```

……

case 取值n：

语句块n;

break;

defaule：

语句块n+1;

break;

}（范例：星期几）

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

While循环语句

while(条件表达式语句)

{

执行语句块;

}（范例：表格）

```
<script>
	var x=1;
	while(x <3){
		alert("x="+x);
```

```
                x++;
        }
</script>
```

表格隔行换色，鼠标经过换色。

```
<style>
table{
        border-collapse:collapse;
        width:80%;
}
td{
        border:solid 1px green;
        text-align:center;
}
</style>
<script>
        var i=0;
        document.write('<table align="center">');
        while(i <20){
                if(i%2 ==0)
                        bg ="#ffffff";
                else
                        bg ="#cccccc";
                document.write('<tr  bgcolor="'+bg+'"  onmouseover="show(this)" onmouseout="show1(this)" >');
                j=0;
                while(j<5){
                        document.write("<td>"+i+"*"+j+"="+i*j+"</td>");
                        j++;
                }
                document.write("</tr>");
                i++;
        }
        document.write("</table>");
```

```
var ys=null;
alert(typeof(ys));
function show(obj){
        ys=obj.bgColor;
        obj.bgColor="red";
}
function show1(obj){
        obj.bgColor=ys;
}
</script>
```

解析：将鼠标未经过时的背景颜色存到一个临时变量中，当鼠标位于行上时改变当前行

的背景色，当鼠标经过时将临时变量里的颜色赋值给当前行。

do while 语句

do
{
　　执行语句块;
}while(条件表达式语句）;
注意：条件不满足时，do while 至少执行一次

for 循环语句

for(初始化表达式; 循环条件表达式; 循环后的操作表达式)
{
　　执行语句块;
}

```
<script>
        var output="";
        for(var x=1;x<10;x++){
                output=output+x;
        }
        alert(output);//输出 123456789
</script>
```

（范例：for 双变量、99 乘法表）

```
<script>
        for(i=1;i<=9;i++){
                for(j=1;j<=i;j++){
                        document.write(i+'*'+j+'='+i*j+'  ');
                }
                document.write('<br />');
        }
</script>
```

Tab 面板

```
<style>
        #box{
                width:300px;
        }
        ul.menu{
                margin:0px;
                padding:0px;
        }
        ul.menu li{
                margin:0px;
                margin-right:5px;
                padding:0px;
                float:left;
```

```
                list-style:none;
                border:1px green solid;
                border-bottom:none;
        }
        .contents{
                border:1px blue solid;
                clear:both;
        }
        .content{
                display:none;
        }
</style>
<script>
        window.onload=function(){
                var
items=document.getElementById('box').getElementsByTagName('ul')[0].getElementsB
yTagName('li');
                var
contents=document.getElementById('box').getElementsByTagName('div')[0].getEleme
ntsByTagName('div');
                //var i;
                temp=contents[0];
                temp.style.display='block';
                for(var i=0;i<items.length;i++){
                        items[i].onmouseover=function(){
                                for(var j=0;j<items.length;j++){
                                        if(items[j]==this && temp!=contents[j]){

                                                temp.style.display="none";

contents[j].style.display="block";
                                                temp=contents[j];
                                        }

                                }
                        }
                }

        }
</script>
<div id="box">
        <ul class="menu">
                <li>item1</li>
                <li>item2</li>
```

```
                <li>item3</li>
                <li>item4</li>
        </ul>
        <div class="contents">
                <div class="content"> item1</div>
                <div class="content">item2</div>
                <div class="content">item3</div>
                <div class="content">item4</div>
        </div>
</div>
```

break 与 continue 语句

break 语句：

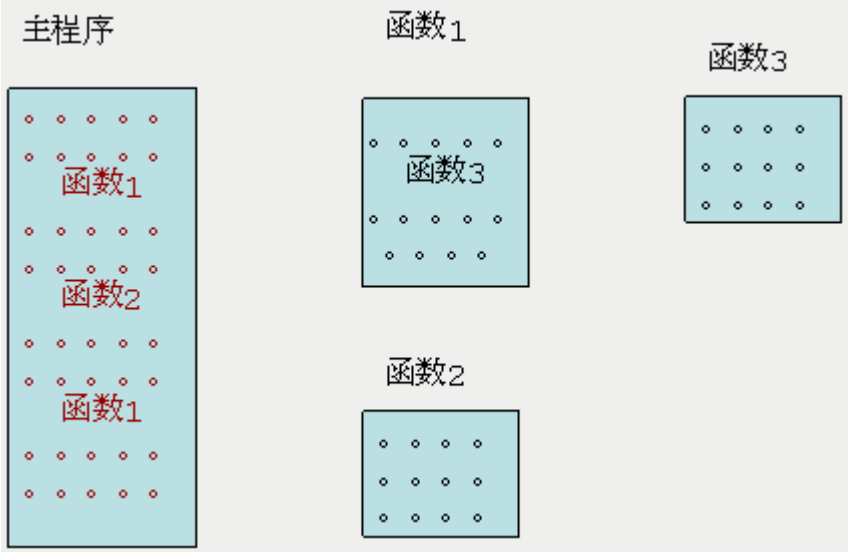
```
st: while(true)
{
    while(true)
    {
        break st;
    }
}
```

continue 语句：

```
<script>
        var output="";
        for(var x=1;x<10;x++){
                if(x%2==0)
                        continue;
                output=output+x;
        }
        alert(output);//输出 13579
</script>
```

函数

函数的作用



函数的定义与调用

定义一个函数的格式如下：

```
function 函数名（参数列表）
{
        程序代码：
         return 表达式;
}
```

对函数进行调用的同种方式：

1. 函数名（传递给函数的参数 1，传递给函数的参数 2，…）

2. 变量 = 函数名（传递给函数的参数 1，传递给函数的参数 2，…）

3. 对于有返回值的函数调用，也可以在程序中直接使用返回的结果，例如：alert(“sum = ”+square(2, 3));

全局变量与局部变量的比较

```
<script>
        var msg="全局变量";
        function show(){
                //var msg = "局部变量";
                alert(msg);
        }
        show();
        alert(msg);
</script>
```

输出：全局变量、全局变量

如果不注释输出：局部变量、全局变量

结果是什么？

参数个数可变的函数

在函数内部使用 arguments 对象来访问调用程序传递的所有参数

```
<script>
        function show(){
                var str="您输入的参数为：";
                for(var i=0;i<arguments.length;i++){
                        str+=arguments[i]+','+' ';
                }
                alert(str);
        }
        show('aa','bb','cc');
        show('aa','bb','cc','11','22');
</script>
```

创建动态函数

创建动态函数的基本语法格式：

var varName = new function(argument1, …. , lastArgument);

说明：

所有的参数都必须是字符串型的，最后参数必须是这个动态函数的功能代码。

例子：

```
<script>
var square =new Function ("x","y","var sum; sum=x*x+y*y;return sum;");

alert(square(3,2));
</script>
```

？多想一想：

动态函数有什么用，在什么情况下用动态函数；

Javascript 中的系统函数

参照 API

prompt、parseInt()、parseFloat();

```
<script>
        var p =prompt("Please input a num:","5");
        var q =prompt("Please input another num:","10");
        var p1 = parseInt(p);
        var q1 = parseInt(q);
        var m=p1+q1;
        alert(m);
</script>
```

encodeURI decodeURI

```
<script>
```

```
        var str="http://localhost/index.php?m=中国&n=北京";
        var nstr=encodeURI(str);
        //                    输                    出                    :
http://localhost/index.php?m=%E4%B8%AD%E5%9B%BD&n=%E5%8C%97%E4%BA%AC
        document.write(nstr);
        var
code="http://localhost/index.php?m=%E4%B8%AD%E5%9B%BD&n=%E5%8C%97%E4%BA%AC";
        var ncode=decodeURI(code);
        //输出：http://localhost/index.php?m=中国&n=北京
        document.write(ncode);
</script>
```

isNaN();

escape();

```
<script>
        var str='escape 方法返回一个包含了 charstring 内容的字符串值（ Unicode
格式）。所有空格、标点、重音符号以及其他非 ASCII 字符都用 %xx 编码代替，其中 xx 等
于表示该字符的十六进制数。例如，空格返回的是 "%20" 。';
        var nstr =escape(str);
        document.write(nstr);
</script>
```

descape();

eval();

```
<script>
        var p =prompt("请输入一个表达式：","");//输入：2+3
        var q = eval(p);
        alert(q);//输出 5


        var a1 =3;
        var a2 =4;
        var a3 =5;
        for(var i=1; i <4; i++){
                document.write(eval("a"+i)+"<br />");
        }
</script>
```

鼠标经过切换图片：

```
<img src="a1.jpg" onmouseover="change()" onmouseout="change()" id="pic" />
<script>
        function change(){
                picname=pic.src.substr(pic.src.lastIndexOf('/')+1);
                if(picname=="a1.jpg")
                        pic.src="a2.jpg";
                else
                        pic.src="a1.jpg";
        }
```

```
</script>
```

定时轮换图片

```
<img src="a1.jpg" onmouseover="cle()" id="pic" />
<script>
        window.clearInterval(dt);
        var dt=setInterval('change()',500);
        var i=1;
        function change(){
                if(i==3)
                        i=1;
                pic.src='a'+i+'.jpg';
                i++;
        }
</script>
```

计数器

```
<html>
        <head>
                <style>
                        #main{
                                width:500px;
                                height:300px;
                                font-size:50px;
                                font-weight:bold;
                                color:red;
                        }
                </style>
        </head>
        <body>
                <div id="main"></div>
                <script>
                        //必须有<html><head><body>才能使下面的代码在 IE 中起作用
                        var dt=setInterval("change()",1000);
                        var m=document.getElementById("main");
                        var i=0;
                        function change(){
                                m.innerHTML=i;//Firefox 不支持 innerText
                                i++;
                        }
                </script>
        </body>
</html>
```

作业：

1、编写一个 js 脚本，要求使用 prompt 输入俩个数字，计算这两个数字的和，并进行

输出（document.write);

2、做一个表单，要求输入矩形的长与宽，然后使用编程计算出矩形的周长与面积，并使用 alert 输出；

3、使用 if elseif 编写一个基于成绩判断等级的程序；

score<60 不及格 60-70 及格 70-85 良好 85 以上 优秀；

4、使用 for 循环输出一个 20 行 5 列的隔行换色的表格；

5、使用 while 循环输出一个 99 乘法表；

对象

对象与对象实例

对象中所包含的变量就是对象的**属性**，对象中所包含的对属性进行操作的函数就是对象的**方法**，对象的属性和方法都叫**对象的成员**。

对象是对某一类事物的描述，是抽象上的概念；而对象实例是一类事物中的具体个例。

能够被用来创建对象实例的函数就叫对象的**构造函数**，只要定义了一个对象的构造函数就等于定义了一个对象，使用 new 关键字和对象的构造函数就可以创建对象实例，语法格式如下：

var objInstance = new ObjName(传递给该对象的实际参数列表);

```
<script>
        function Person(){//构造函数

        }
        var person1 =new Person();//对象实例
        person1.age=18;//属性
        person1.name="zxx";//可以为对象无限制的添加新成员
        alert(person1.name+' '+person1.age);
        function sayFunc(){
                //用"对象实例名。成员名"的格式访问
                //也可以使用（对象实例("成员变量名")）的格式。
                alert(person1.name+' '+person1.age);
        }
        person1.say=sayFunc;
        person1.say();
</script>
```

构造方法与 this 关键字

为一个对象实例新增加的属性和方法，不会增加到同一个对象所产生的其它对象实例上。

所有的实例对象在创建后都会自动调用构造函数，在构造函数中增加的属性和方法会被增加到每个对象实例上。

对象实例是用 new 关键字创建的，在构造方法中不要有返回结果的 return 语句。

调用对象的成员方法时，需要使用“对象实例。成员方法”的形式，很显然，用作成员方法的函数被调用时，一定伴随有某个对象实例。This 关键字代表某个成员方法执行时，引用该方法的当前对象实例，所以，this 关键字一般只在用作对象方法的函数中出现。

```
<script>
        function Person(name,age){
                this.age = age;
                this.name = name;
                this.say = sayFunc;
        }
        function sayFunc(){
                alert(this.name+':'+this.age);
        }
        var person1 =new Person("张三",18);
        person1.say();
        var person2 =new Person("李四",20);
        person2.say();
</script>
```

自定义对象

对象的使用：

1.自定义对象 function person(){ … };

2. 实例化对象 var p=new person();

3. 通过对象名访问其属性与方法：

   p.say()

   p.name;

实例化

Var p1=new person(“张三”,”30”,”男”);

Var p2=new person(“李四”,”28”,”女”);

执行相关的对象属性与方法：

调用对象的属性与方法：

p1.age=28;

p1.say();

对象属性的遍历

For（var 变量 in 对象名);

输出结果：

Name:张三;

Age：30;

Sex：男;

```
<script>
        function Person(name,age,sex){//自定义对象
                this.name = name;
                this.age = age;
                this.sex = sex;
                this.say =function(){
                        //输出：我的名字：张三，我的年龄：28 我的性别：男
```

```
                document.write("我的名字： "+this.name+"， 我的年龄：
"+this.age+"我的性别： "+this.sex+"<br />");
            }
            this.run=function(){
                document.write(this.name+"在走路");
            }
        }
        var p1 =new Person("张三","30","男");//实例化对象
        var p2 =new Person("李四","28","女");
        //调用对象的属性与方法
        p1.age ="28";
        p1.say();

        for(var shuxin p1){//对象属性的遍历
            /*
            输出：
            name 张三
            age 28
            sex 男
            say function(){ //输出： document.write("我的名字："+this.name+",
我的年龄："+this.age+"我的性别："+this.sex+"
            "); }
            run function(){ document.write(this.name+"在走路"); }
            */
            document.write(shux+" "+eval("p1."+shux)+"<br>");
        }
</script>
```

```
<script>
        function radio(w,h,p){
            this.width = w;
            this.height = h;
            this.pinpai = p;
            this.bf =function(){
                alert('正在播放音乐');
            }
            this.tt=xuantai;
            function xuantai(){
                alert('选择合适的频道！');
            }
        }
        var p =new radio('30','10','ds');
        with(p){
            bf();//正在播放音乐
```

```
                tt();//选择合适的频道!
                alert(width);//30
        }
        for(var ddin p){
                if(typeof(p[dd])!="function"){
                        document.write('属性名:'+dd+'的属性值:'+p[dd]+'<br />');

                }
        }
</script>
```

输出:

属性名:width 的属性值:30

属性名:height 的属性值:10

属性名:pinpai 的属性值:ds

在函数中修改参数值的问题

将基本数据类型的变量作为函数参数传递的情况:

```
<script>
        function changeValue(x){
                x =5;
        }
        var x =3;
        changeValue(x);
        alert(x);//输出 3
</script>
```

此外的 x 值是多少?

将对象类型变量作为函数参数传递的情况:

```
<script>
        function Person(name,age){
                this.age = age;
                this.name = name;
                this.say = sayFunc;
        }
        function sayFunc(){
                alert(this.name+':'+this.age);
        }
        function change(p1){
                p1.name ="李四";
        }
        var person1 =new Person("张三",18);
        change(person1);
        person1.say();
```

```
</script>
```

JavaScript 的内部对象

动态对象

使用“对象实例名.成员”的格式来访问其属性和方法

静态对象

直接使用“对象名.成员”的格式来访问其属性和方法。

Object 对象

Object 对象提供了一种创建自定义对象的简单方式，不需要程序员再定义构造函数。

```
<script>
        function getAttributeValue(attr){
                alert(person[attr]);
        }
        var person =new Object();
        person.name ='zs';
        person.age =18;
        getAttributeValue("name");
        getAttributeValue("age");
</script>
```

number 对象

toFixed();toFixed() 方法可把 Number 四舍五入为指定小数位数的数字。

```
<script>
        var t=-10.28351;
        var m=t.toFixed(3);//四舍五入为指定小数位数的数字。
        alert(m);
</script>
```

String 对象

length 属性

anchor、big、bold、fontcolor、link 等 方法

```
<script>
        var txt="Hellow World!";
        document.write('<p>Big:'+txt.big()+'</p>');
        document.write('<p>Small:'+txt.small()+'</p>');
        document.write('<p>Bold:'+txt.bold()+'</p>');
        document.write('<p>Italics:'+txt.italics()+'</p>');
        document.write('<p>Blink:'+txt.blink()+'(does not work in IE)</p>');
        document.write('<p>Fixed:'+txt.fixed()+'</p>');
        document.write('<p>Strike:'+txt.strike()+'</p>');
        document.write('<p>Fontcolor:'+txt.fontcolor("red")+'</p>');
```

```
        document.write('<p>Fontsize:'+txt.fontsize(16)+'</p>');
        document.write('<p>Tolowercase:'+txt.toLowerCase()+'</p>');
        document.write('<p>Uppercase:'+txt.toUpperCase()+'</p>');
        document.write('<p>Sub:'+txt.sub()+'</p>');
        document.write('<p>Sup:'+txt.sup()+'</p>');

document.write('<p>Link:'+txt.link('http://www.w3school.com.cn')+'</p>');
</script>
```

charAt 方法

注意：一个字符串的第一个字符的索引 位置 为 0，依次类推。

charCodeAt 方法

注意：返回的结果是字符的 unicode 编码 。

indexOf 方法

lastIndexOf 方法

replace 方法

```
<script>
        var str="hello world world";
        document.write(str.indexOf('r')+'<br />');
        document.write(str.lastIndexOf('r')+'<br />');
        document.write(str.replace('world','beiging')+'<br />');//只进行了一次替换，并没有对原字符串进行替换。
        document.write(str+'<br />');
</script>
输出：
8
14
hello beiging world
hello world world
```

match、search 方法

split 方法

toLowerCase、toUpperCase 方法

slice 方法

说明：str1.slice(0)和 str1.slice(0, -1)都 是返回 整个字符串。

substr、substring 方法

注意：substring 方法返回的内容不包含结束 位置 的字符。(兼容 ie4+)

Substring 可以兼容 ie3+的低版本浏览器；

replace 方法

Math 对象

Math 对象是一个静态对象， 不能使用 new 关键字创建对象实例，应直接使用“对象名。成员”的格式访问其属性或方法，例如：

```
var num = Math.random();
```

```
<script>
        var color =['red','green','blue','yellow','#ff00ff','#ffffff','#aaaaaa'];
```

```
        document.bgColor=color[parseInt(Math.random()*color.length)];
</script>
```

Math 对象的属性

E，代表数学常数 e,约等于 2.718.
LN10， 代表 10 的自然对数，约等于 2.302.
LN2, 代表 2 的自然对数，约等于 0.693.
PI， 代表数学常数 PI 的值，等于 3.14159.
SQRT1-2, 代表 2 的平方根分之一，约等于 0.707
SQRT2, 代表 2 的平方根，约等于 1.414.

Math 对象的方法

abs 方法， 返回数字的绝对值。
sin、cos 方法，分别返回数字的正弦、余弦值。
asin、acos 方法，分别返回数字的反正弦、反余弦值。
random()； 产生一个 0-1 之间的随机数；
pow(x,y)；计算 x 的 y 次方的结果；
ceil()； 返回等于或大于参数的最小整数；
floor()； 返回等于或小于参数的最小整数；
round()； 返回四舍五入后的结果；
toFixed(x)；返回保留 x 位小数的四舍五入的结果；

Date 对象

构造方法：Date()、Date(dateVal)、Date(year、month, date[, hours[, minutes[, seconds[, ms]]]])

parse 方法，分析一个表示日期时间的字符串，返回它所表示的时间值，该值以自 1970 年 1 月 1 日 0 点 0 分 0 秒算起的毫秒值表示。Parse 方法属于一个静态方法。

toGMTString 方法,返回 Date 对象实例 所表示 的日期的字符串形式,该字符串使用 用格林尼治标准时间（GMT）格式，例如， “05 Jan 1996 00:00:00 GMT”

getYear、getMonth、getDate、getDay 方法

getHours、getMinutes、getSeconds、getMilliseconds 方法

getTime 方法，返回自 1970 的 1 月 1 日 0 点 0 分 0 秒算起，至 Date 对象实例代表的时间为止的毫秒数。

获取时间实例

```
<script>
     var current_time =new Date();
     var strDate = current_time.getYear()+"年";
      strDate += current_time.getMonth()+'月';
      strDate += current_time.getDate()+'日';
      strDate += current_time.getHours()+':';
      strDate += current_time.getMinutes()+':';
      strDate += current_time.getSeconds();
```

```
    alert(strDate);
</script>
```

```
<script>
        var cdate =new Date();
        with(cdate){
                var year = getYear();
                var month = getMonth()+1;
                var date = getDate();
                var hour = getHours();
                var minute = getMinutes();
                var second = getSeconds();
        }
        var ndate = year+' 年 '+month+' 月 '+date+' 日 '+' '+hour+':'+minute+':'+second;
        document.write(ndate);
</script>
```

设置时间实例

```
<script>
        //Sun Jul 31 10:12:45 UTC+0800 2011
        var mydate1 =new Date();
        //Fri Apr 3 00:00:00 UTC+0800 2009
        var mydate2 =new Date('April 3,2009');
        //Fri Apr 3 08:20:05 UTC+0800 2009
        var mydate3 =new Date('April 3,2009 8:20:5');
        //Tue Mar 1 01:00:00 UTC+0800 1904
        var mydate4 =new Date(04,2,1,1);
        //Fri Jan 1 16:25:03 UTC+0800 1988
        var mydate5 =new Date(1988,0,1,16,25,3,4);
        //Sat Jan 3 15:33:20 UTC+0800 1970
        var mydate6 =new Date(200000000);

        for(var i=1;i <7; i++){
                document.write(eval('mydate'+i)+'<br />')
        }
</script>
```

```
<script>
        var mydate =new Date(2009,10,3);
        mydate.setMonth(3);
        //Fri Apr 3 00:00:00 UTC+0800 2009
        document.write(mydate+'<br />');

        var mydate =new Date();
```

```
        mydate.setTime(300000000);
        //Sun Jan 4 19:20:00 UTC+0800 1970
        document.write(mydate);
</script>
```

```
<script>
        var week =new Array('日','一','二','三','四','五','六');
        var mydate =new Date();
        with(mydate){
                var year = getFullYear();
                var month = getMonth()+1;
                var date = getDate();
                var w = week[mydate.getDay()];
        }
        document.write('格式 1：'+month+'/'+date+'/'+year+'<br />');
        document.write('格式 2：'+fmt(month)+'/'+fmt(date)+'/'+year+'<br />');
        document.write('格式 3：'+year+'年'+month+'月'+date+'日'+'  星期'+w);

        //设置前导零
        function fmt(s){
                if(s <10)
                        return '0'+s;
                else
                        return s;
        }
</script>
```

```
<script>
        var mydate =new Date();
        //1312081831046
        document.write(mydate.getTime()+'<br />');
        //Sun Jul 31 11:10:31 UTC+0800 2011
        document.write(mydate.toString()+'<br />');
        //11:10:31 UTC+0800
        document.write(mydate.toTimeString()+'<br />');
        //Sun, 31 Jul 2011 03:10:31 UTC
        document.write(mydate.toUTCString()+'<br />');
        //2011 年 7 月 31 日 11:10:31
        document.write(mydate.toLocaleString()+'<br />');
        //11:10:31
        document.write(mydate.toLocaleTimeString()+'<br />');
        //2011 年 7 月 31 日
        document.write(mydate.toLocaleDateString()+'<br />');
        //1312081831046
```

```
        document.write(mydate.valueOf()+'<br />');
</script>
```

toString方法

Tostring 方法是 JavaScript 中的所有内部对象的一个成员方法，它的主要作用就是将对象中的数据转换成某种格式的字符串来表示，具体的转换方式取决于对象的类型。

举例：

```
<script>
        var x=32812321;
        //NumberObject.toString(radix)
        //radix：可选。
        //规定表示数字的基数，使 2～36 之间的整数。若省略该参数，则使用基数 10。

        //但是要注意，如果该参数是 10 以外的其他值，则 ECMAScript 标准允许实现返回任意值。
        //输出：hes=1f4ad21 bin=1111101001010110100100001
        document.write("hes=" + x.toString(16)+" bin="+x.toString(2)+"<br />");



        //stringObject.valueOf()
        //valueOf() 方法可返回 String 对象的原始值。
        //原始值是由从 String 对象下来的所有对象继承的。
        //valueOf() 方法通常由 JavaScript 在后台自动进行调用，而不是显式地处于代码中
        var str="This is a string";
        document.write(str.valueOf()+'<br />');

        var arr =new Array(3);
        arr[0] ="George";
        arr[1] ="John";
        arr[2] ="Thomas"
        //输出：George,John,Thomas
        document.write(arr.toString()+'<br />');

        //arrayObject.toLocaleString()
        //首先调用每个数组元素的 toLocaleString() 方法，然后使用地区特定的分隔符把生成的字符串连接起来，形成一个字符串。
        //输出：George, John, Thomas
        document.write(arr.toLocaleString()+'<br />');

        var boo =new Boolean(true);
        //输出：true
        document.write(boo.toString()+'<br />');
```

```
        var d =new Date();
        //输出：Wed Jul 27 15:16:15 UTC+0800 2011
        document.write(d.toString()+'<br />');
        //dateObject.toLocaleString()
        //以本地时间区表示，并根据本地规则格式化。
        //输出：2011 年 7 月 27 日 15:24:52
        document.write(d.toLocaleString()+'<br />');
</script>
```

对象专用语句

1、with 语句：

```
with(对象名称)
{
        执行语句块
}
```

2、for…in 语句：

```
for(变量 in 对象)
{
        执行语句
}
```

With 举例

```
<script>
        var current_time =new Date();
        with(current_time){
                var strDate = getYear() +"年";//Firefox:111 年；IE:2011 年
                strDate += getMonth() +"月";
                strDate += getDate() +"日";
                strDate += getHours() +":";
                strDate += getMinutes() +":";
                strDate += getSeconds();
                document.write(strDate);//Firefox:111 年 6 月 27 日 16:3:8; IE:2011
年 6 月 27 日 16:2:58
        }
        alert(current_time.getFullYear());//2011
</script>
```

for…in 举例

```
 <script>
        function Person(){
                this.name ="sky";
                this.age =19;
                this.height =172;
        }
        var p1 =new Person();
        var prop, str =" ";
```

```
        for(propin p1){
                str += prop +":"+p1[prop]+" ";
        }
        document.write(str);//输出: name:sky age:19 height:172
</script>
```

数组

Javascript 数组也是一种对象，可以有属性和方法；

数组的声明：

```
<script>
        var arr=new Array();
        var arr=new Array(5);
        var arr=new Array("sun","mon","tue","web");
        var arr=["aa","bb","cc"];
        var arr={aa:33,bb:44,cc:88};
</script>;
```

数组的遍历

```
<script>
        var arr=['aa1','bb1','cc1'];
        for(var i=0; i < arr.length; i++){
                document.write('############'+arr[i]+'<br />');
        }
        for(xin arr){
                document.write("##########"+arr[x]+"<br />");
        }
</script>
############aa1
############bb1
############cc1
##########aa1
##########bb1
##########cc1
```

```
<script>
        var arr =new Array('aa','bb','cc','dd');
        var arr =['aa','bb','cc','dd','ee'];
        var narr ={'name':'zhangsan','age':'30','sex':'nan'};
        //数组的遍历
        for(var nnin narr){
                document.write(nn+'的值为: '+narr[nn]+'<br />');
        }
        var str='';
```

```
        for(var i=0;i < arr.length;i++){
                str+=arr[i]+',';
        }
        document.write(str);
</script>
```

```
<script>
        var arr=[["names","beansproul","pumpkin","max"],["age",6,5,4]];
        for(var i =0; i < arr.length; i++){
                for(var j=0; j < arr[i].length; j++){
                        alert(arr[i][j]);
                }
        }
</script>
```

```
<script>
        var arr =[['zhangsan','30','nan'], ['lisi','28','nv'], ['wangwu', '20',
'nv']];
        document.write('<table width="500" border="1">');
        for(var i=0; i < arr.length; i++){
                document.write('<tr>');
                        for(var j=0; j < arr[i].length; j++){
                                document.write('<td>'+arr[i][j]+'</td>');
                        }
                document.write('</tr>');
        }
        document.write('<table>');
</script>
```

数组列表

数组列表用于表示一组数据的集合，它由一对方括号（[ ]）包围，列表中的每个无素用逗号分隔，数组元素可以是任类型的数据（包括其他数组）。例如：

var arr=[“sky”, 123, ‘1i’, 3.5];

对于每个数组变量，都有一个 length 属性，表示该数组中元素的个数。

使用“数组变量名[索引号]”的格式来访问每个数组元素。数组列表中的第一个元素的索引号为 0,最后那个元素的索引号为数组的 length 属性值-1.

数组列表的每个元素既可以是有效的 JavaScript 表达式，也可以为空，空元素的值为 undefined.例如 ：

var arr = [1, , , , 5];

数组元素本身又是数组，就就叫**数组的数组**，例如：

[[“names”, “beansproul”, “pumpkin”, “max”], [“ages”, 6, 5, 4]]

使用“数组变量名[子数组索引号][子数组中的元素索引号]”的格式来访问数组的数组

中的元素。

</script>

用对象的方式实现数组

```
<script>
	function MyArray(){//将接受来的参数变为自己有成员属性
		this.length =arguments.length;
		for(var i =0; i <this.length; i++){
			this[i] =arguments[i];
		}
	}
	var str ="";
	var arr=new MyArray(4,3.4,"abc");
	for(var i =0; i < arr.length; i++){
		str += arr[i] +"\n";
	}
	alert(str);
</script>
```


```
<script>
	function MyArray(size){
		this.length = size;
		for(var j =0; j < size; j++){
			this[j] ="";
		}
	}
	var arr =new MyArray(2);
	arr[0] =3;
	arr[1] ="abc";
	arr[2] =4;
	arr[3] =5;
	var x,str="";
	for(xin arr){
		str +=x+":"+arr[x]+"<br />";
	}
	document.write(str);
</script>
```

输出：

```
length:2//输出的值赋予的值
0:3
1:abc
2:4
3:5
```

Array 对象

三种构造方法

Array()

Array(4)

Array(3.4, “abc”, 3)

数组的常用方法

sort() 排序

concat() 合并数组;

```
<script>
        var arr=new Array();
        var arr1=['aa','bb','cc'];
        var arr2=['11','22','33'];
        document.write(arr1.concat(arr2)+'<br />');
        document.write(arr.concat(arr1).concat(arr2)+'<br />');
</script>
```

join() 将数组转换为字符串;

```
<script>
        var arr=new Array('apple','banana','orange','pear','mango');
        document.write(arr.join('###')+'<br />');
        document.write(arr+'<br />');
        //arrayObject.pop()返回并删除数组中的最后的一个元素，对原数组进行了修改
        document.write(arr.pop()+'<br />');
        document.write(arr+'<br />');
        //arrayObject.slice(start,end)返回一个新的数组，包含从 start 到 end （不包括该元素）的 arrayObject 中的元素,并不会修改数组
        document.write(arr.slice(1,4)+'<br />');
        document.write(arr+'<br />');
</script>
```

数组排序

```
<script>
        var arr =new Array();
        arr[0] =3.4;
        arr[1] ="abc";
        arr[2] =3;
        arr.sort();//对原数组进行了修改
```

```
        var x, str="";
        for(xin arr){
                str += x+":"+arr[x]+"<br />";
        }
        document.write(str);
</script>
```

输出：

0:3

1:3.4

2:abc

课外：

获取焦点：

```
<form name="frm">
        user:<input type="text" name="user" value="cc" /><br />
        age:<input type="text" id="age" name="age" /><br />
        address:<input type="text" id="address" /><br />
</form>
<script>
        document.frm.age.focus();//IE 不适用
</script>
```

给 Select 添加 Opetion

```
<form name="frm">
        <select name="xueli">
        </select>
</form>
<script>
        var arr=['初中','高中','大专','本科','研究生'];
        var num = arr.length;
        for(var i=0;i < num; i++){
                document.frm.xueli[i]=new Option(arr[i]);
        }
</script>
```

改变背景色

```
<body>
<script>
        document.body.style.backgroundColor='yellow';
</script>
</body>
```

联动下拉菜单

```
<form name="frm">
        省份:
        <select name="optProvince" onchange="ChangeCity()">
                <option>--请选择--</option>
                <option>广东省</option>
```

```
                <option>湖南省</option>
                <option>浙江省</option>
        </select>
        城市：
        <select name="optCity">
                <option>--请选择--</option>
        </select>
</form>
<script>
        var aCity =new Array();
        aCity[0] =new Array();
        aCity[1] =new Array();
        aCity[2] =new Array();
        aCity[3] =new Array();

        aCity[0][0] ='--请选择--';

        aCity[1][0] ='--请选择--';
        aCity[1][1] ='广州市';
        aCity[1][2] ='深圳市';
        aCity[1][3] ='珠海市';
        aCity[1][4]="汕头市";
        aCity[1][5]="佛山市";

        aCity[2][0]="--请选择--";
        aCity[2][1]="长沙市";
        aCity[2][2]="株州市";
        aCity[2][3]="湘潭市";

        aCity[3][0]="--请选择--";
        aCity[3][1]="杭州市";
        aCity[3][2]="苏州市";
        aCity[3][3]="温州市";

        function ChangeCity(){
                var i,iProviceIndex;
                iProviceIndex=document.frm.optProvince.selectedIndex;//获取所选
择省份的序号
                iCityCount=0;
                //计算所选的省份内有多少个城市
                while(aCity[iProviceIndex][iCityCount]!=null)
                        iCityCount++;
                //将表单中城市数组的长度更改为新选省份所含的城市数量
                document.frm.optCity.length=iCityCount;
```

```
                //将城市名称添加到表单中城市的下拉列表中
                for(i=0;i<iCityCount;i++)
                        document.frm.optCity[i]=new
Option(aCity[iProviceIndex][i]);
                //让表单中的城市控件获取焦点
                document.frm.optCity.focus();
        }
</script>
```

DOM 编程

DHTML

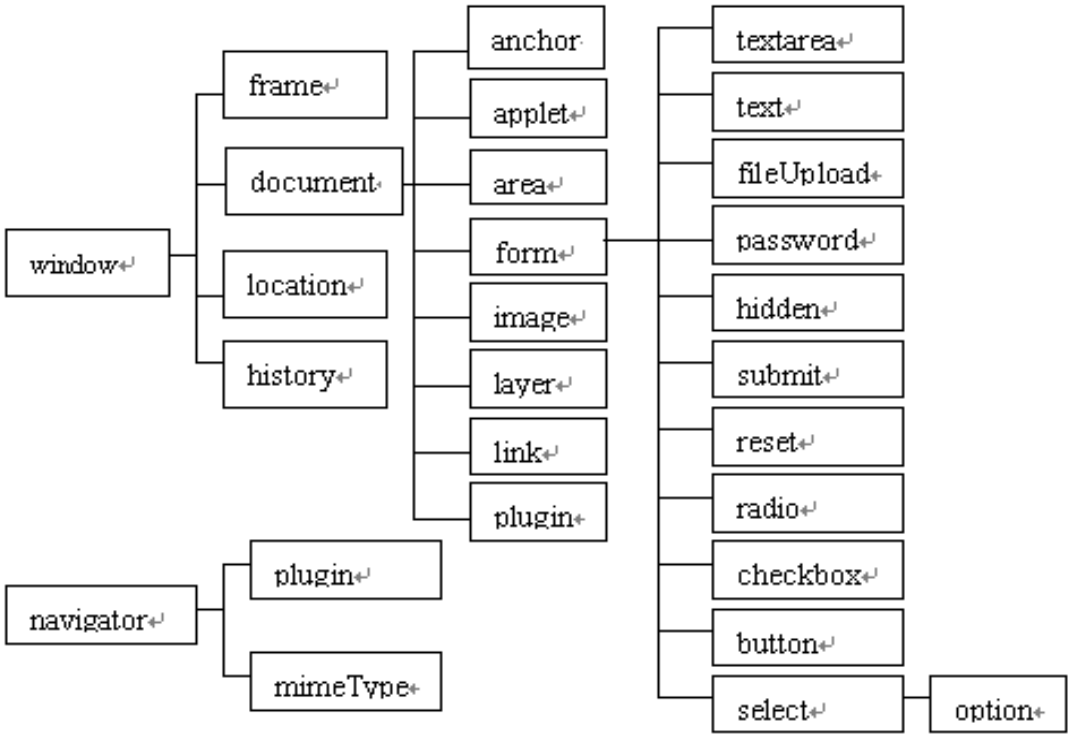
1. DHTML（Dynamic HTML，动态 HTML）.

由 html+div/css 与浏览器脚本语言完成的动态页面

HTML+JAVASCRIPT+DIV/CSS

2 DOM （Document Object Model，文档对象模型）

JAVASCRIPT 将浏览器本身、网页文档、以及网页文档中的 HTML 元素等都用相应的内置对象来表示，这些对象及对象之间的层次关系称为 DOM



事 件

事件： 用户对浏览器所做的特定的动作。

onclick

onmouseover

onmouseout

onload

onkeyup

如何编写事件处理程序

一、 在事件源对象所对应的 HTML 标签上增加一个要处理事件属性，让事件属性值等于处理事件的函数名或程序代码。

格式：<tag onclick=“<语句组>|<函数名>” >

【例 1】

```
<body onload="alert('建议浏览器的分辨率：800X600')"></body>
```

【例 2】

```
<body onload="show()" ></body>
<script>
        function show(){
                var str="建议浏览器的分辨率：800X600";
                alert(str);
        }
</script>
```

屏蔽页面右击菜单（Firefox 不支持）

```
<html>
        <head>
                <script>
                        <!--
                        function hideContextmenu(){
                                window.event.returnValue=false;
                        }
                        //-->
                </script>
        </head>
        <body oncontextmenu="hideContextmenu()"></body>
</html>
```

```
<html>
        <head>
                <script>
                        <!--
                        function hideContextmenu(){
                                return false;
                        }
                        //-->
                </script>
        </head>
        <body oncontextmenu="return hideContextmenu()"></body>
</html>
```

```
<img id="tu" src="a1.jpg" alt="aaaaaaaaaaaaa" onmouseover="change()" />
```

```
<script>
        //setTimeout("change()",2000);
        var pic =document.getElementById("tu");
        function change(){
                pic.src="a2.jpg";
                pic.alt="########";
        }
</script>
```

二、直接在 JavaScript 代码中， 设置元素对象的事件属性， 让事件属性值等于处理该事件的函数名或程序代码。

格式： 对象名.on 事件=<语句>|<函数名>

例 1：

```
<script>
        document.onload=alert("建议浏览器的分辨率：800x600");
</script>
```

例 2：(IE 和 Firefox 都不支持)

```
<script>
function show(){
    var str="建议浏览器的分辨率：800x600";
    alert(str);
}
document.onload=show;
</script>
```

三、在一个专门的<script>标签对中编写某个元素对象的某种事件处理程序代码，并用 for 属性指定事件源和用 event 属性指定事件名， 这种<script>标签中的脚本程序只在指定事件源的指定事件发生时才被调用执行，这种方式常用于网页文档中的各种插件对象的事件处理程序：(Firefox 不支持)

格式：<script for="id" event="onclick">事件处理程序< /script>

```
<script language="javascript" for="document" event="oncontextmenu">
        window.event.returnValue=false;</script>
```

综合实例

```
<style>
        .one{
                width:200px;
                height:50px;
                border:solid 1px green;
                font-size:3cm;
        }
        .two{
                width:200px;
                height:50px;
                color:yellow;
                border:solid 5px red;
```

```
                font-size:3cm;
        }
</style>
<img id="tu" src="a1.jpg" alt="aaaaaaaaa" onmouseover="changepic()" />
<div id="ss">sssssssssssssss</div>
<h1 id="hh" onmouseout="cc()">ddddddddddddddd</h1>
<a href="http://www.baidu.com" target="_self" class="one" id="aa">baidu</a>
<script>
        var pic =document.getElementById("tu");
        var ss =document.getElementById("ss");
        var hh =document.getElementById("hh");
        var aa =document.getElementById("aa");

        aa.onmouseover = cha;
        aa.onmouseout =function(){
                aa.className ="one";
        }
        function cha(){
                aa.className ="two";
        }

        ss.onclick = change;
        function change(){
                ss.innerHTML ="ttttttt";
                ss.innerHTML ='<img src="a2.jpg">';
        }
        function changepic(){
                pic.src="a2.jpg";
        }

        function cc(){
                hh.style.backgroundColor ="green";
                hh.style.fontSize ="5cm";
        }
</script>

<script for="hh" event="onmouseover"><!--Firefox 不支持-->
        hh.style.backgroundColor="red";
</script>
```

window 对象

window 对象代表浏览器的整个窗口，编程人员可以利用 window 对象控制浏览器窗口的各个方面，如改变状态栏上的显示文字、弹出对话框、移动窗口的位置等。

对 window 对象的属性和方法的引用，可以省略“window.”这个前缀，例如，

window.alert("你好")可以直接写成 alert("你好")。

window 对象－方法

alert 方法
confirm 方法
prompt 方法

```
<body onload="welcome()" onunload="bye()">
        <ul oncontextmenu="return jy()"><!--Firefox 不支持-->
                <li>sssssssssssss</li>
                <li>sssssssssssss</li>
                <li>sssssssssssss</li>
                <li>sssssssssssss</li>
                <li>sssssssssssss</li>
                <li>sssssssssssss</li>
        </ul>
        <a href="http://localhost/del.php" onclick="return confirm('你确认要删除吗？')">删除</a>
        <a onclick="input()">输入</a>
</body>

<script>
        function jy(){
                alert("不容许复制，菜单被禁用");
                return false;
        }
        function welcome(){
                alert("欢迎光临本站点");
        }
        function bye(){
                alert("欢迎下次光临");
        }

        function input(){
                var dd =prompt("请输入一行字符串：","sss");
                alert(dd);
        }
</script>
```

navigate 方法
setInterval 方法
clearInterval 方法

```
<style>
        h1{
                font-size:15cm;
                color:red;
```

```
                text-align:center;
                font-weight:bold;
        }
</style>
<h1 id="hh" onmouseover="cle()" onmouseout="start()" >0</h1>
<script>
        var dsq;
        var i=0;
        window.onload=start;
        function start(){
                dsq=setInterval(function(){
                        document.getElementById("hh").innerHTML=i;
                        i++;
                },1000);
        }
        function cle(){
                clearInterval(dsq);
        }
</script>
```

setTimeout 方法

clearTimeout 方法

```
<input type="button" value="取消" onclick="qx()" />
<script>
        var d1=setTimeout("c1()",1000);
        var d2=setTimeout("c2()",2000);
        var d3=setTimeout("c3()",3000);
        var d4=setTimeout("c4()",4000);

        function qx(){
                clearTimeout(d2);
        }
        function c1(){
                document.bgColor="red";
        }
        function c2(){
                document.bgColor="green";
        }
        function c3(){
                document.bgColor="yellow";
        }
        function c4(){
                document.bgColor="blue";
        }
</script>
```

moveTo 方法

resizeTo 方法

```
<script>
        var nwin =window.open("test.html",'','');
        nwin.resizeTo(300,350);
</script>
```

open 方法

父窗口控制子窗口的背景颜色

```
<body onunload="cw()/*本文档关闭时，关闭子窗口*/">
        <form>
                <input type="button" value="red" onclick="c1()" />
                <input type="button" value="yellow" onclick="c2()" />
                <input type="button" value="green" onclick="c3()" />
        </form>
        <!--在打开的子窗口中打开新的文件-->
        <a href="http://www.baidu.com" target="nwin">baidu</a>
</body>
<script>
        var                                                                nwin
=window.open('test.html','nwin','left=100,top=100,width=500,height=300,location
=yes,resizable=1,scrollbars=1');
        function cw(){
                //如果子窗口没有关闭，就关闭子窗口
                if(!nwin.closed){
                        nwin.close();
                }
        }
        function c1(){
                nwin.document.body.style.background="red";
        }
        function c2(){
                nwin.document.body.style.background="yellow";
        }
        function c3(){
                nwin.document.body.style.background="green";
        }
</script>
```

showModalDialog 方法

showModelessDialog 方法

test.html 网页文件：

打开新窗口

```
<script>
        window.open('imformation.html','_blank',"top=0,    left=0,    width=200,
height=200,  toolbar=no")
```

```
</script>
```

information.html 网页文件：

```
<script>
        window.setTimeout('window.close()',5000);
</script>
<center><h3>通知</h3></center>
5 秒钟以后， 这个窗口会自动关闭!
```

test.html 网页文件：（全屏）

```
<script>
        window.open('information.html','_blank',"top=0,    left=0,    width=200,
height=200,  toolbar=no,fullscreen=yes")
</script>
```

nformation.html 网页文件：（Firefox 会增大不会关闭，IE8 会关闭不会增大，IE6 增大并关闭）

```
<script>
        window.setTimeout('window.close()',5000);
        window.setInterval('grow()',100);
        function grow(){
                window.resizeBy(1,1);
        }
</script>
<body>
        <center><h3>通知</h3></center>
        5 秒钟以后， 这个窗口会自动关闭!
</body>
```

window 对象－属性

--closed 属性

--opener 属性

子窗口控制父窗口

父窗口代码

```
<script>
        window.open('test.html');
</script>
```

子窗口代码

```
<input type="button" value="red" onclick="c1()" /><br />
<input type="button" value="yellow" onclick="c2()" />
<script>
        function c1(){
                window.opener.document.body.style.background="red";
        }
        function c2(){
                window.opener.document.body.style.background="yellow";
```

```
        }
</script>
```

--defaultstatus 属性

```
<script>
        window.defaultStatus="欢迎光临本网站";
</script>
```

--status 属性

--screenTop 属性

--screenLeft 属性

window.html 关闭子窗口

```
<script>
        var  child  =window.open('information.html','__blank','top=0,  left=0,
toolbar=no, fullscreen=yes, titlebar=no');
        function closeChild(){
                if(!child.close()){
                        child.close();
                }
        }
        window.onload=setTimeout('closeChild()',1000);
</script>
```

文本在状态栏上滚动显示

```
<script>
        setInterval('scroll()',100);
        var space_num =0;
        var dir =1;
        function scroll(){
                var str_space ='';
                space_num=space_num+dir;//计算空格数量
                //如果空格的数量大于 40 或小于等于 0，则增加或减少空格
                if(space_num >40 || space_num <=0){
                        dir =-1*dir;
                }
                //填充空格
                for(var i =0; i < space_num; i++){
                        str_space +=' ';
                }
                window.status = str_space+'www.phpchina.com';
        }
        function start(){
                setInterval('scroll()',100);
        }
</script>
```

information.html

<script>

```
        window.setTimeout("window.close()", 5000);
        window.setInterval("grow()", 100);
        function grow()
        {
            window.resizeBy(5, 5);
        }
            function closeit()
            {
                window.close();
                opener.start();
            }
    //or//  window.setTimeout("window.closeit()", 5000);
    <script>
    <body  onunload="window.opener.start()"> ////////
        <center><h3>通知</h3></center>
        5 秒钟以后， 这个窗口会自动关闭!
    </body>
```

window 对象－事件

专用事件：

－－onload 事件

－－onunload 事件

－－onbeforeunload 事件

通用事件

－－onclick 事件

－－onmousemove 事件

－－onmouseover 事件

－－onmouseout 事件

－－onmousedown 事件

－－onmouseup 事件

－－onkeydown 事件

－－onkeyup 事件

用户名自动大写，并检验输入年龄是否是数字

```
<style>
        #mes{
                font-size:14px;
                color:red;
        }
</style>
<form name="frm">
        用户名：<input type="text" name="uname" id="uname" onkeyup="dx(this)"
/><br />
        年龄：<input type="text" name="age" id="age" onkeyup="yz()" /><span
id="mes">请输入数字</span><br />
```

```
</form>
<script>
        var uname =document.getElementById("uname");
        var mes =document.getElementById("mes");

        function dx(){

if(document.frm.uname.value !=document.frm.uname.value.toUpperCase()){
                        document.frm.uname.value
=document.frm.uname.value.toUpperCase();
                }
        }

        function yz(){
                if(isNaN(parseInt(document.frm.age.value))){
                        mes.innerHTML="输入错误";
                }else{
                        mes.innerHTML="输入正确";
                }
        }
</script>
```

——onkeypress 事件

onkeyup、onkeydown 能获取键盘上的除 Windows 键之外的任意键。

onkeypress 只能获取可见字符。

window_event.html

```
<script>
        alert('ok1');
</script>
<body    onload="alert('    欢    迎    ')"    onunload="alert('    再    见    ')"
onbeforeunload="window.event.returnValue='请小心'">
hello
</body>
<script>
        alert('ok2');
</script>
<script>
        alert('ok3');
</script>
```

首先三次 alert，后是'欢迎'，关闭窗口前'请小心'，关闭窗口后'再见'。

window 对象－对象属性

——location 对象

——event 对象

——frames 数组对象

——screen 对象

```
<script>
        with(document){
                write("你的的屏幕显示设定值如下：<br />");
                write("屏幕的可用高度为",screen.availHeight,'<br />');
                write("屏幕的可用宽度为",screen.availWidth,'<br />');
                write("屏幕的色盘深度为",screen.colorDepth,'<br />');
                write("屏幕的实际高度为",screen.height,'<br />');
                write("屏幕的实际宽度为",screen.width,'<br />');
                write("文档的宽度为：",body.clientWidth,"<br />");
                write("文档的高度为：",body.clientHeight,"<br />");
        }
</script>
```

——clipboardDate 对象

——history 对象

——navigator 对象

```
<script>
        with(document){
                write("你的浏览器信息：<ol>");
                        write("<li>代码："+navigator.appCodeName+"</li>");
                        write("<li>名称："+navigator.appName+"</li>");
                        write("<li>版本："+navigator.appVersion+"</li>");
                        write("<li>语言："+navigator.language+"</li>");//IE8 不
支持
                        write("<li>编译平台："+navigator.platform+"</li>");
                        write("<li>用户表头："+navigator.userAgent+"</li>");
                write("</ol>");
        }
        if(document.all){
                document.write("你的浏览器是：MSIE");
        }else{
                document.write("你的浏览器是：Navigator");
        }
</script>
```

IE8 输出：

```
你的浏览器信息：
    1    代码：Mozilla
    2    名称：Microsoft Internet Explorer
    3    版 本 ： 4.0 (compatible; MSIE 8.0; Windows NT 5.1;
Trident/4.0; .NET4.0C; .NET4.0E; .NET CLR 2.0.50727; .NET CLR
3.0.4506.2152; .NET CLR 3.5.30729)
    4    语言：undefined
    5    编译平台：Win32
    6    用 户 表 头 ： Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1;
```

Trident/4.0; .NET4.0C; .NET4.0E; .NET CLR 2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)
你的浏览器是：MSIE

Google Chrome输出：

你的浏览器信息：

1 代码：Mozilla
2 名称：Netscape
3 版本：5.0 (Windows NT 5.1) AppleWebKit/534.30 (KHTML, like Gecko) Chrome/12.0.742.122 Safari/534.30
4 语言：zh-CN
5 编译平台：Win32
6 用户表头：Mozilla/5.0 (Windows NT 5.1) AppleWebKit/534.30 (KHTML, like Gecko) Chrome/12.0.742.122 Safari/534.30

你的浏览器是：Navigator

Firefox 5.0输出：

你的浏览器信息：

1 代码：Mozilla
2 名称：Netscape
3 版本：5.0 (Windows)
4 语言：zh-CN
5 编译平台：Win32
6 用户表头：Mozilla/5.0 (Windows NT 5.1; rv:5.0) Gecko/20100101 Firefox/5.0

你的浏览器是：Navigator

——document 对象

window 对象—location 对象

——window.location.href="http://www.phpchina.com";
等效于
window.navigate("http://www.thizlinux.com.cn/news.html");
--location 对象的 replace 方法也可用于载入一个新的网页
Location 对象的 reload() 方法可以重新装载当前文档，replace() 可以装载一个新文档而无须为它创建一个新的历史记录，也就是说，在浏览器的历史列表中，新文档将替换当前文档。
——location 对象的 reload 方法用于重新载入（刷新）窗口中的当前网页。

语法

location.reload(force)

说明

如果该方法没有规定参数，或者参数是 false，它就会用 HTTP 头 If-Modified-Since 来检测服务器上的文档是否已改变。如果文档已改变，reload() 会再次下载该文档。如果

文档未改变，则该方法将从缓存中装载文档。这与用户单击浏览器的刷新按钮的效果是完全一样的。

如果把该方法的参数设置为 true，那么无论文档的最后修改日期是什么，它都会绕过缓存，从服务器上重新下载该文档。这与用户在单击浏览器的刷新按钮时按住 Shift 健的效果是完全一样。

```
window.location.reload();
```

location.html

```
<script>
        setTimeout('window.location.href="http://www.baidu.com"',2000);
        setTimeout('window.location.replace("http://www.baidu.com")',2000);
        window.navigate('http://www.baidu.com');
        window.location.href="http://www.baidu.com";
</script>
```

```
This is a web page!
<script>
        dt = setTimeout(function(){
                        //window.location.href="1.html";
                        //window.location="1.html";
                        window.location.replace('1.html');
                },1000);
</script>
```

显示隐藏菜单

```
<style>
        #c1{
                display:none;
        }
</style>
<body>
<ul class="menu">
        <li onmouseover="show()" onmouseout="hide()">用户管理
                <ul class="list" id="c1">
                        <li>增加用户</li>
                        <li>修改用户</li>
                </ul>
        </li>
        <li>新闻管理</li>
        <li>内容管理</li>
        <li>留言管理</li>
        <li>邮箱管理</li>

</ul>
</body>
```

```
<script>
        var c1=document.getElementById("c1");
        function show(){
                c1.style.display="block";
        }
        function hide(){
                c1.style.display="none";
        }
</script>
```

作业

1.幻灯播放。

复习

javaScript 语言

1. 语法
   大小写 区分
   变量 aaa bbb ccc aaaBbbCcc
   getElementsByTagName();
   var a=10
   var b=20
2. 变量和类型
3. 运算符和表达式
4. 流程控制
5. 函数

有回调函数的函数

版本一

```
<script>
        function demo(a,b,fun){
                return a+b+fun(a,b);
        }
        alert(demo(2,3,test));
        function test(x,y){
                return x*y;
        }
</script>
```

版本二

```
<script>
        function demo(a,b,fun){
                return a+b+fun(a,b);
        }
        alert(demo(2,3,function(x,y){
                                return x*y+5;
                        }));
</script>
```

计数器

```
<div id="one" style="font-size:5cm;text-align:center;color:red">0</div>
<script>
        var obj =document.getElementById('one');
        var i =0;
        setInterval(function(){
                                i++;
                                obj.innerHTML = i
                        },1000);
</script>
```

创建参数有默认值的函数

```
<script>
        function demo(a,b,c){
                if(typeof(c)=="undefined")
                        c=10;
                alert(a+"-----"+b+"-------"+c);
        }
        demo(5,6);
        demo(4,8,12);
</script>
```

全局变量与局部变量

```
<script>
        var i=0;
        function demo(a,b,c){
                //var i=5;//下面函数外的alert(i)输出 0
                i=5//函数外的alert(i)下面输出 6
                i++;
        }
        demo();
        demo();
        demo();
        demo();
        demo();
        demo();
        demo();
        alert(i);
</script>
```

6. 对象和数组

对象

方法一：

```
<script>
        function hello(){
```

```
        }
        var h =new hello();//也可 var h = hello;
        h.aa=10;
        h.bb=20;
        h.func=function(){
                alert(this.aa);
        }
        h.func();
</script>
```

方法二、

```
<script>
        function hello(){
                //成员属性
                this.aa=10;
                this.bb=20;
                //成员方法
                this.func=function(){
                        alert(5);
                }
                this.fun5(){

                }
        }
        var h=new hello();
</script>
```

方法三、

```
<script>
        function hello(){
                var h =new Object();
                h.aa =10;
                h.bb =20;
                h.func =function(){
                        alert(h.aa);
                }
                return h;
        }
        var h = hello();
</script>
```

幻灯播放的原理

```
<script>
        function play(width, height, num, time){
                this.width = width;
                this.height = height;
                this.num = num;
```

```
                this.time = time*1000;
                this.auto=function(){
                        alert(this.width+'      '+this.height+'      '+this.num+'
'+this.time)
                }
                this.zhi =function(){

                }
        }
        var p =new play(400,300,10,3);
        p.auto();
</script>
```

方法四、

```
<script>
        var obj ={name:"zhangsan", age:10, height:34.5};
        alert(obj.name);
        alert(obj.age);
</script>
```

随机提取数组中的元素

```
<body></body>
<script>
        var colors =["red","green","yellow","blue","#FF00FF"];
        //document.bgColor                                                    =
colors[parseInt(Math.random()*colors.length)];//Firefox 不支持，IE 支持

document.body.bgColor=colors[parseInt(Math.random()*colors.length)];//IE      、
Firefox 都支持，但文档中必须有<body></body>标签
</script>
```

利用自定义对象的方式声明数组

```
<body></body><!--加上此为了在 Firefox 中也能改变背景颜色，IE 中加与不加都可以-->
<script>
        function MyArray(){
                this.length =arguments.length;//必须给长度赋值
                for(var i=0; i<arguments.length; i++){
                        this[i] =arguments[i];
                }
                this.sort =function(){
                        return;
                }
        }
        var colors =new MyArray('red','green','yellow','blue','#FF00FF');
        colors.sort();
        document.bgColor = colors[parseInt(Math.random()*colors.length)];
</script>
```

棋盘

```
<script>
        var w =20;
        var h =20;
        var rows =20;
        var cols =20;
        var map =new Array();
        for(var i=0; i < rows; i++){
                map[i] =new Array();
                for(var j=0; j < cols; j++){
                        map[i][j] =0;
                }
        }
        map[Math.floor(Math.random()*rows)][Math.floor(Math.random()*cols)] =1;
        map[Math.floor(Math.random()*rows)][Math.floor(Math.random()*cols)] =2;
        var start =100;
        for(var i =0; i < rows; i++){
                for(var j =0; j < cols; j++){
                        if(map[i][j] ==1){
                                var bg ="yellow";
                        }else if(map[i][j] ==2){
                                var bg ="blue";
                        }else{
                                var bg ="red";
                        }
                        document.write('<div
style="position:absolute;top:'+(start+(h+1)*i)+'px;left:'+(start+(w+1)*j)+';wid
th:'+w+'px;height:'+h+'px;background:'+bg+'"></div>');
                }
        }
</script>
```

b

dom (document object model)

d --- document 文档 html 或 xml 文件才可以称为 document --- ml 标记语言

o --- object 对象， 将每一个元素可以做为一个 javaScript 对象 （操作对象中的属性和方法....）

m --- model

```
<style>
        .hello{
                color:blue;
                font-size:3cm;
        }
```

```
        .world{
                border:5px solid green;
        }
</style>
<a id="one" href="http://www.baidu.com" target="_self">baidu</a>
<script>
        var aobj =document.getElementById('one');
        //输出对象属性
        alert(aobj.href);
        alert(aobj.id);
        alert(aobj.target);
        alert(aobj.tagName);
        alert(aobj.innerHTML);
        alert(aobj.innerText);//IE 输出 baidu，Firefox 输出 undefined
        alert(aobj.childNodes[0].nodeValue);//输出 baidu
        //更改对象属性
        aobj.innerHTML ="google";
        aobj.href ="http://www.google.com";
        aobj.target ="_blank";
        aobj.title ="this is a demo";
        aobj.className ="hello";
        aobj.className +=" world";
        //aobj.className = "";//去掉所有样式
        aobj.style.color ="red";
        aobj.style.fontSize ="3cm";
        //aobj.style.display = "none";
        alert(aobj.parentNode.tagName);//输出 body
        aobj.parentNode.style.display="none";
</script>
```

点名机

```
<style >
        #main{
                position:absolute;
                left:50%;
                top:50%;
                width:200px;
                height:200px;
                margin-top:-100px;
                margin-left:-100px;
                text-align:center;
        }
        #names{
                float:left;
```

```
                width:200px;
                height:150px;
                border:2px solid red;
                line-height:150px;
                text-align:center;
                margin-bottom:10px;
                font-size:1cm;
                font-weight:bold;
        }
</style>
<div id="main">
        <div id="names">
                --请选择--
        </div>
        <input          id="two"          type="button"          value="start"
style="float:left;width:200px;" />
</div>
<script>
        var name =document.getElementById("names");
        var but =document.getElementById("two");
        var colors =["red","blue","yellow","green","#ff00ff","#CCCCCC"];
        var names =["张三","五 五","六 六","七 七","李小四","在要在","工有地"];
        var dt =null;
        but.onclick=function(){
                if(but.value=="start"){
                        dt = setInterval(function(){
                                name.innerHTML                                  =
names[Math.floor(Math.random()*names.length)];
                                name.style.color                                =
colors[Math.floor(Math.random()*colors.length)];
                                },100);
                        but.value ="stop";
                }else{
                        setTimeout(function(){
                                        clearInterval(dt);
                                },300);
                        but.value ="start";
                }
        }
</script>
```

form元素值的获取

```
<form name="frm">
        username:<input type="text" name="username" value="zhangsan" />
</form>
```

```
<form name="one">

</form>

<script>
        alert("1"+document.frm.username.value);
        alert("2"+document.forms[0].username.value);
        alert("3"+document.forms["frm"].username.value);
        alert("4"+document.forms.frm.username.value);
        alert("5"+document["frm"].username.value);
        alert("6"+document.forms.item(0).username.value);
        alert("7"+document.forms.item("frm").username.value);
</script>
```

bom

b--browser

window

open() close() setInterval() clearInterval() setTimeout() clearTimeout() alert() confirm() resize() ....

opener

...

document body images links scripts styles forms ....

history

location

screen

.....

ajax

jQuery

window 对象－event 对象

——altKey 属性

——ctrlKey 属性

——shiftKey 属性

```
<body onclick="show(event)">
        <script>
                function show(event){
                        if(event.altKey){
                                alert("You press ALT-Key");
                        }else if(event.ctrlKey){
                                alert("You press CTRL-Key");
                        }else if(event.shiftKey){
                                alert("You press SHIFT-KEY");
                        }else{
```

```
                        alert("You don't press any key");
                    }
                }
        </script>
</body>
```

——clientX、clientY 属性

——screenX、screenY 属性

```
<html>
        <head>
                <script type="text/javascript">
                    function show_coords(event){
                        x=event.clientX;
                        y=event.clientY;
                        screenX=event.screenX;
                        screenY=event.screenY;
                        alert("你点击文档的 X 坐标： "+x+", "+"Y 坐标："+y+"\n"+"你点击屏幕的 X 坐标： "+screenX+", "+"Y 坐标： "+screenY);
                    }
                </script>
        </head>
        <body onmousedown="show_coords(event)">
                点击空白将会提示点击时的坐标。
        </body>
</html>
```

跟随鼠标效果及状态栏显示鼠标坐标

```
<body onmousemove="show(event)">
        <div                                                    id="one"
style="position:absolute;left:-100px;top:-100px;">hello</div>
        <script>
                var one =document.getElementById("one");
                function show(event){
                        var x =event.clientX;
                        var y =event.clientY;

                        one.style.top=y+10;
                        one.style.left=x+10;
                        window.status="x="+x+","+"y="+y;
                }
        </script>
</body>
```

——offsetX、offsetY 属性

——x，y 属性

——**returnValue** 属性：如果设置了该属性，它的值比事件句柄的返回值优先级高。把

这个属性设置为 fasle，可以取消发生事件的源元素的默认动作。

```
<a href="http://localhost/del.php" onclick="dd()">删除</a>
<script>
        function dd(){
                if(!confirm('你确认删除吗？')){
                        event.returnValue=false;
                }
        }
</script>
```

——cancelBubble 属性:**取消级联**

——srcElement 属性**（只有 IE 支持）**

```
<a href="http://www.a.com" onmouseover="ch()">Hello Hello Hello</a>
<script>
        function ch(){
                window.event.srcElement.style.background="green";
        }
</script>
```

——keyCode 属性

方向键控制方块的移动

```
<body onkeydown="show(event)">
        <div id="cont"></div>
        <div                                                    id="box"
style="width:50px;height:50px;position:absolute;top:100px;left:100px;background
-color:red;"></div>
<script>
        var box=document.getElementById("box");
        left=parseInt(box.style.left);
        topy=parseInt(box.style.top);
        function show(event){
                document.getElementById("cont").innerText=event.keyCode;
                if(event.keyCode==37){

                        left-=10;
                        box.style.left=left;
                }
                if(event.keyCode==38){
                        topy-=10;
                        box.style.top=topy;
                }
                if(event.keyCode==39){
                        left+=10;
                        box.style.left=left;
                }
```

```
                if(event.keyCode==40){
                        topy+=10;
                        box.style.top=topy;
                }
        }
</script>
</body>
```

游戏设计

```
<body onkeydown="show(event)">
        <img        style="position:absolute;left:0px;top:0px;"        id="ren"
src="images/ren_q_1.gif">
        <script>
                //思路
                //1.当页面加载时图片会动，实现靠的是计时器
                        //1.1.获取图片对象
                        //1.2.获取图片的名称
                        //1.3.更换图片，实现 dong()
                //2. 当按方向键时图片移动，实现靠的是 <body
onkeydown="show(event)">
                        //2.1.根据方向健设置移动的方向和相应的图片,实现 show()
                        //2.2.改变图片的位置，实现计时器中的 run()
                //要点：1.计时器；2.更换图片；3 改变图片的位置

                //获取图片对象
                var ren=document.getElementById("ren");
                //获取图片的名称
                function basename(url){
                        var loc=url.lastIndexOf("/")+1;
                        return url.substr(loc);
                }

                //更换图片，实现 dong()
                var d="q";
                var i=0;//控制速度
                function dong(){
                        if(i%3==0){//速度为原来的 1/3
                                //需要 8 张图片，每个方向两张图片
                                if(basename(ren.src)=="ren_"+d+"_1.gif"){
                                        ren.src="images/ren_"+d+"_2.gif";
                                }else{
                                        ren.src="images/ren_"+d+"_1.gif";
                                }
                                i=0;//整 3 后归 0
```

```
        }
        i++;//不断自增
}

//改变图片的位置，实现计时器中的 run()
var a=0;
var b=0;
var x=0;
var y=0;
function run(){
        a+=x;
        b+=y;
        ren.style.left=a;
        ren.style.top=b;
}

//设置计时器，使得小人不断的移动
setInterval(function(){
        dong();
        run();
},100);

//根据方向健设置移动的方向和相应的图片
function show(event){
        switch(event.keyCode){
                case 37:
                        x=-5;
                        y=0;
                        d="l";
                        break;
                case 38:
                        x=0;
                        y=-5;
                        d="h";
                        break;
                case 39:
                        x=5;
                        y=0;
                        d="r";
                        break;
                case 40:
                        x=0;
                        y=5;
                        d="q";
```

```
                                        break;
                                }
                        }
                </script>
</body>
```

——button 属性

eventobject.html　按下键盘事件(只有 IE 支持)

```
<body onkeypress="window_onkeypress()"></body>
<script>
        function window_onkeypress(){
                alert(window.event.keyCode);
        }
</script>
```

按 esc 关闭窗口(只有 IE 支持)

```
<body onkeypress="window_onkeypress()"></body>
<script>
        function window_onkeypress(){
                if(window.event.keyCode ==27){
                        window.close();
                }
        }
</script>
```

阻止冒泡事件（只适 IE)（由里向外的事件的顺序发生）

```
<body onclick="two()">
        <img src="a1.jpg" onclick="one()" id="pic" />
</body>
<script>
        var pic=document.getElementById("pic");
        function one(){
                event.cancelBubble=true;
                alert("Event of body is already canceled!");
        }
        function two(){
                alert("Event of body");
        }
</script>
```

```
<body onclick="showSrc()">
        <img onclick="checkCancel()" src="a1.jpg" />
</body>
<script>
        function checkCancel(){
                if(window.event.shiftKey){
```

```
                window.event.cancelBubble=true;//当按下shift键单击图片时将不会触发点击图片的事件
            }
        }
        function showSrc(){
            if(window.event.srcElement.tagName.toLowerCase() =="img"){
                alert(window.event.srcElement.src);
            }
        }
</script>
```

Window对象——frames数组对象（1）

Window对象的frames属性是一个数组，它与window对象的parent、top等对象属性，都是用于对HTML的帧标签(<frameset>或<iframe>)进行编程的javaScript对象。

通过顶级窗体访问子窗体的方式：

```
window.top.framse[i]
window.top.frames['name']
window.top.frames.item(2)
window.top.frames.item('name');
window.top.frames.name;
window.top.name;
window.top['name']
```

framedomeo.html

```
<html>
        <head></head>
        <frameset rows="20%,80%">
            <frame name="top" src="top.html">
            <frame name="bottom" src="bottom.html">
        </frameset>
</html>
```

Top.html

```
<input type="button" value="red" onclick="window.parent.frames[1].document.bgColor=this.value">
<input type="button" value="green" onclick="window.parent.frames.bottom.document.bgColor=this.value">
<input type="button" value="yellow" onclick="window.parent.frames['bottom'].document.bgColor=this.value">
<input type="button" value="blue" onclick="window.parent.frames.item(1).document.bgColor=this.value">
<input type="button" value="gray" onclick="window.parent.frames.item('bottom').document.bgColor=this.value">
<input type="button" value="black" onclick="window.parent['bottom'].document.bgColor=this.value">
```

```
<form name="aa">
        <input type="text" name="user" value="zhangsan" />
</form>
<form name="bb">
        <input type="text" name="age" value="30" />
</form>

<a href="http://www.a.com" name="aaa">aaaaa</a><br />
<a href="http://www.b.com" name="bbb">bbbbb</a><br />
<a href="http://www.c.com" name="ccc">bbbbb</a><br />

<img src="a.jpg" /><img src="b.jpg" /><img src="c.jpg" /><img src="d.jpg" />
<br />
<script>
        document.write("forms:"+document.forms.length+"<br />");
        document.write("links:"+document.links.length+"<br />");
        document.write("images:"+document.images.length+"<br />");

        document.write("age:"+document.forms[1].age.value+"<br />");
        document.write("user:"+document.aa.user.value+"<br />");

        document.write("linka:"+document.links[0].href+"<br />");
        //不能使用[]
        document.write("linkb:"+document.links("bbb").href+"<br />");

        document.write("linka:"+document.links.item(0).href+"<br />");
        document.write("linkb:"+document.links.item("bbb").href+"<br />");

</script>
```

框架页面

```
<frameset rows="100,*">
        <frame src="top.html" name="tt" />
        <frameset cols="20%,*">
                <frame src="menu.html" name="menu" />
                <frame src="main.html" name="main" />
        </frameset>
</frameset>
```

top.html

```
This is top page.
```

main.html

```
This is main page.
```

menu.html

```
This is menu page.<br />
<input type="button" value="blue" onclick="change(this.value)" />
<input type="button" value="green" onclick="change(this.value)" />
<script>
        function change(color){
                window.top.main.document.bgColor=color;
        }
</script>
```

Window对象——frames数组对象（2）

Top.html(顶级页面):

```
<frameset rows="20%,*">
        <frame name="a" />
        <frame name="x" src="bottom.html" />
</frameset>
```

Bottom.html:

```
<frameset cols="30%,*" >
        <frame name="b" />
        <frame name="c" src="bottom_right.html">
</frameset>
```

Bottom_right.html:

```
<script language="javascript">
        top.a.document.write('www.phpchina.com');
        parent.parent.a.document.bgColor="red";
</script>
```

Top.html

```
<frameset rows="20%,*">
        <frame name="a" />
        <frameset cols="30%,*" >
                <frame name="b" />
                <frame name="c" src="bottom_right.html" />
        </frameset>
</frameset>
```

Bottom_right.html:

```
<script language="javascript">
        top.a.document.write('www.phpchina.com');
        parent.parent.a.document.bgColor="red";
</script>
```

Document 对象 ——方法

Write 方法
writeln 方法
open 方法
close 方法

```
<script language="javascript">
        document.write("这是 write 方法动态写入的内容");
        function updatedoc(){
                //清除本窗口中原来的内容，写下以下新的内容
                document.writeln('abc<br />');//在源代码后加入换行符
                document.writeln('def<br />');//在源代码后加入换行符
                document.close();//关不掉
                //打开一个空白窗口，在空白窗口中写入内容，然后清除空白窗口中的内
容，再写入新的内容。
                var owin =window.open('','_blank');
                owin.document.writeln('xyz<br />');
                owin.document.close();//关不掉
                以下代码使用 write 方法写入，源代码在一行中
                owin.document.write('abc<br />');
                owin.document.write('defj<br />');
                owin.document.close();//关不掉
        }
</script>
<input type="button" name="update" value="更新" onclick="updatedoc()" />
```

向打开的新文档流中写入 JavaScript 代码

```
<script language="javascript">
        document.write("这是 write 方法动态写入的内容");
        //向打开的新文档流中写入 JavaScript 代码
        function updatedoc(){
                document.writeln('abc<br />');
                document.writeln('def<br />');
                document.writeln('<script language="javascript">');
                        document.writeln('  function updatedoc(){');
                        document.writeln('    document.writeln("abc<br />")');
                        document.writeln('    document.writeln("def<br />")');
                        document.writeln('}');
                document.writeln('</scr'+'ipt>');//此处需要分开写
                document.writeln('<input type="button" name="update" value="更新
" onclick="updatedoc()" />');
        }
</script>
<input type="button" name="update" value="更新" onclick="updatedoc()" />
```

clear 方法

getElementById 方法

getElementsByName()方法

getElementsByTagName 方法

createElement 方法

创建和删除结点

```
<body>
        ####################<br />
        <div id="one"></div>
        RRRRRRRRRRRRRRRRRRR<br />
        <script>
                var obj=document.getElementById("one");
                function demo(){
                        var a=document.createElement("a");
                        a.href="http://www.baidu.com";
                        a.innerText="baidu";
                        a.style.backgroundColor="red";

                        var br=document.createElement("br");

                        obj.appendChild(a);
                        obj.appendChild(br);

                }
                function dell(){
                        obj.removeChild(obj.lastChild);
                        obj.removeChild(obj.lastChild);
                }
        </script>
        <input type="button" value="add" onclick="demo()" />
        <input type="button" value="del" onclick="dell()" />
</body>
```

```
<input type="button" value="创建" onclick="create()" />
<input type="button" value="删除" onclick="del()" />
<div id="ab"></div>
<script>
        var ab=document.getElementById("ab");
        var img=null;

        function create(){
                img=document.createElement("img");
                img.src="a.jpg";
                img.alt="This is a image!";
                ab.appendChild(img);
```

```
        }

        function del(){
                //首先获取图片的数量，如果图片的数量大于0，则除最后的张图片。
                if(ab.getElementsByTagName('img').length >0){
                        imgs=ab.getElementsByTagName('img');
                        ab.removeChild(imgs[imgs.length-1]);
                }else{
                        alert('There is not image!');
                }

        }
</script>
```

复选框的全选、反选、全不选、删除操作

```
<a href="javascript:qx()">全选</a>
<a href="javascript:fan()">反选</a>
<a href="javascript:qbx()">全不选</a>
<a href="javascript:sc()">删除</a>
<br />
<script>
        for(var i=0;i<20;i++){
                document.write('<span><input     type="checkbox"     name="del[]" />'+i+'#####'+'<br></span>');
        }

        var dels=document.getElementsByName('del[]');
        function qx(){
                for(var i=0;i<dels.length;i++){
                        dels[i].checked=true;
                }
        }
        function qbx(){
                for(var i=0;i<dels.length;i++){
                        dels[i].checked=false;
                }
        }
        function fan(){
                for(var i=0;i<dels.length;i++){
                        if(dels[i].checked){
                                dels[i].checked=false;
                        }else{
                                dels[i].checked=true;
                        }
                }
```

```
        }
        function sc(){
                //必须以倒着的顺序删除节点，因为在删除节点时，节点所在数组的长度在循环删除节点过程中是不断变化的
                for(var i=dels.length-1;i>=0;i--){
                        if(dels[i].checked){

dels[i].parentNode.parentNode.removeChild(dels[i].parentNode);
                        }
                }
                alert(dels.length);
        }
</script>
```

createStyleSheet 方法

Document 对象属性

属性	描述
body	提供对 <body> 元素的直接访问。 对于定义了框架集的文档，该属性引用最外层的 <frameset>。
cookie	设置或返回与当前文档有关的所有 cookie。
domain	返回当前文档的域名。
lastModified	返回文档被最后修改的日期和时间。
referrer	返回载入当前文档的文档的 URL。
title	返回当前文档的标题。
URL	返回当前文档的 URL。

```
<html>
        <head>
                <title>This is a page title!</title>
        </head>
<script type="text/javascript">
        document.write(document.cookie+"<br />");
        document.write(document.referrer+"<br />");//IE 不支持
        document.write(document.domain+"<br />");
        document.write(document.lastModified+"<br />");
        document.write(document.title+"<br />");
        document.write(document.URL+"<br />");
</script>
</html>
```

Firefox 输出：

```
ECS[visit_times]=3
http://localhost/Javascript/9.html
localhost
08/01/2011 09:54:12
```

```
This is a page title!
http://localhost/Javascript/1.html
```

Form 对象

Form 对象代表一个 HTML 表单。

在 HTML 文档中 <form> 每出现一次，Form 对象就会被创建。

Form 对象集合

集合	描述
elements[]	包含表单中所有元素的数组。

Form 对象属性

属性	描述
acceptCharset	服务器可接受的字符集。
action	设置或返回表单的 action 属性。
enctype	设置或返回表单用来编码内容的 MIME 类型。
id	设置或返回表单的 id。
length	返回表单中的元素数目。
method	设置或返回将数据发送到服务器的 HTTP 方法。
name	设置或返回表单的名称。
target	设置或返回表单提交结果的 Frame 或 Window 名。

标准属性

属性	描述
className	设置或返回元素的 class 属性。
dir	设置或返回文本的方向。
lang	设置或返回元素的语言代码。
title	设置或返回元素的 title 属性。

Form 对象方法

方法	描述
reset()	把表单的所有输入元素重置为它们的默认值。
submit()	提交表单。

Form 对象事件句柄

事件句柄	描述
onreset	在重置表单元素之前调用。
onsubmit	在提交表单之前调用。

```
<body onload="document.frm.username.focus();/*文档在加载时让 username 获得焦点*/">
        <form name="frm" action="one.php" method="post">
```

```
                username:<input  type="text"  name="username"  value="zhangsan" />
<br />
                password:<input  type="password"  name="password"  value="1234" />
<br />
                <!--提交按钮的 name 属性不能为 submit，否则会与 JavaScript 中的表单的提交方法 submit()产生冲突-->
                <input type="submit" name="submit3" value="login" />
        </form>
        <div onclick="show()">hello</div>
        <script>
                function show(){
                        var frmobj=document.frm;
                        frmobj.action="bbb.php";
                        frmobj.method="get";
                        frmobj.target="_blank";
                        frmobj.username.value="lisit";
                        frmobj.password.value="456";
                        frmobj.submit();
                }
        </script>
</body>
```

表单验证

```
<body onload="document.frm.username.focus()">
        <form name="frm" action="one.php" method="post" onsubmit="return show()">
                username:<input type="text" name="username" onblur="one()" /><br />
                password:<input  type="password"  name="password"  onblur="two()"  /><br />
                <input type="submit" name="submit3" value="login" />
        </form>
        <script>
                //1.进行焦点离开时的验证
                //为了防止死锁，one()中的 document.frm.username.focus();不能与 two()中的 document.frm.username.focus();同时存在。
                function one(){
                        if(!document.frm.username.value.match(/^\S+$/)){
                                alert("用户名不能为空");
                                //document.frm.username.focus();
                        }
                }
                function two(){
                        if(!document.frm.password.value.match(/^\S+$/)){
                                alert("用户密码不能为空");
```

```
                        //document.frm.password.focus();
                    }
                }
                //2.进行数据提交时的验证
                function show(){
                    var flag=true;
                    var mess="";
                    var focu;
                    //if(!focu)是在 focu 中没有对象时获取对象，这样做是为了使靠前的空元素获取焦点
                    //obj.match((/^\S+$/)匹配非空白字串
                    if(!document.frm.username.value.match(/^\S+$/)){//如果为空或有空白字符就执行下面的代码
                        mess="用户名不能为空\n";
                        flag=false;
                        if(!focu){
                            focu=document.frm.username;
                        }
                    }
                    if(!document.frm.password.value.match(/^\S+$/)){
                        mess+="用户密码不能为空\n";
                        flag=false;
                        if(!focu){
                            focu=document.frm.password;
                        }
                    }
                    if(flag){
                        return true;
                    }else{
                        alert(mess);
                        focu.focus();
                        return false;
                    }
                }
        </script>
</body>
```

通过下拉列表的选择更换图片

```
<img id="tu" src="images/1.jpg" height="200" />
<br />
<form>
        <select onchange="show(this)">
            <option value="0">--请选择--</option>
            <option value="1">第一个图</option>
```

```
                <option value="2">第二个图</option>
                <option value="3">第三个图</option>
                <option value="4">第四个图</option>
        </select>
</form>
<script>
        var tu=document.getElementById("tu");
        function show(obj){
                var val=obj.options[obj.selectedIndex].value;
                tu.src="images/"+val+".jpg";
        }
</script>
```

Ajax

ajax 是必会的

ajax 异步传输a javascript and xml

ajax 引擎 就是 javaScript 中的一个对象（六个属性 和 6 个方法）

用浏览器请求 PHP 时， PHP 中的所有输出都给浏览器

如果使用 Ajax 请求 PHP 时， 只不过 PHP 中的所有输出都给 Ajax

做什么用的

都在那用

怎么用

局部刷新

接需取数据

一、创建 Ajax 对象

（6 个属性 和 6 个方法）

浏览器分为两种 （IE（不同版本） 和 非 IE）

ajax 类 ajax3.0.js

```
//recvType 有两个值 HTML 和 XML , 默认为 HTML
function Ajax(recvType) {
        var aj = new Object();
        aj.targetUrl = '';    //请求的地址 可以是 PHP 也可以 XML 文件
        aj.sendString = '';   //请求服务器传递的字符串  ? & 格式 url

        aj.recvType=recvType ? recvType.toUpperCase() : 'HTML';//HTML XML
        aj.resultHandle = null;

        aj.createXMLHttpRequest = function() {
                var request = false;
                //非 IE
                if(window.XMLHttpRequest) {
                        request = new XMLHttpRequest();
                        if(request.overrideMimeType) {
```

```
                                request.overrideMimeType('text/xml');//有些浏览
器默认没有加载处理'text/xml'的功能，在此加上这个功能
                        }
                //IE
                } else if(window.ActiveXObject) {
                        //XMLHttpRequest 的版本
                        var versions = ['Microsoft.XMLHTTP', 'MSXML.XMLHTTP',
'Microsoft.XMLHTTP',          'Msxml2.XMLHTTP.7.0',          'Msxml2.XMLHTTP.6.0',
'Msxml2.XMLHTTP.5.0',          'Msxml2.XMLHTTP.4.0',          'MSXML2.XMLHTTP.3.0',
'MSXML2.XMLHTTP'];
                        for(var i=0; i<versions.length; i++) {
                                try {
                                        request = new ActiveXObject(versions[i]);

                                        if(request) {
                                                return request;
                                        }
                                } catch(e) {
                                        request=false;
                                }
                        }
                }
                return request;
        }

        aj.XMLHttpRequest = aj.createXMLHttpRequest();

        aj.processHandle = function() {
                if(aj.XMLHttpRequest.readyState == 4) {
                        if(aj.XMLHttpRequest.status == 200) {
                                if(aj.recvType == 'HTML') {

aj.resultHandle(aj.XMLHttpRequest.responseText);
                                } else if(aj.recvType == 'XML') {

aj.resultHandle(aj.XMLHttpRequest.responseXML);
                                }
                        }
                }
        }

        aj.get = function(targetUrl, resultHandle) {
                aj.targetUrl = targetUrl;
                if(resultHandle!=null){
```

```
                aj.XMLHttpRequest.onreadystatechange = aj.processHandle;

                aj.resultHandle = resultHandle;
            }
            if(window.XMLHttpRequest) {
                aj.XMLHttpRequest.open('GET', aj.targetUrl);
                aj.XMLHttpRequest.send(null);
            } else {
                aj.XMLHttpRequest.open("GET", targetUrl, true);
                aj.XMLHttpRequest.send();
            }
        }

        aj.post = function(targetUrl, sendString, resultHandle) {
            aj.targetUrl = targetUrl;

            if(typeof(sendString)=="object"){
                var str="";
                for(var pro in sendString){
                    str+=pro+"="+sendString[pro]+"&";
                }

                aj.sendString=str.substr(0, str.length-1);//去掉最后的
"&"
            }else{
                aj.sendString = sendString;
            }



            if(resultHandle!=null){
                aj.XMLHttpRequest.onreadystatechange = aj.processHandle;

                aj.resultHandle = resultHandle;
            }
            aj.XMLHttpRequest.open('POST', targetUrl);
            aj.XMLHttpRequest.setRequestHeader('Content-Type',
'application/x-www-form-urlencoded');
            aj.XMLHttpRequest.send(aj.sendString);
         }
        return aj;
}
```

二、用 Ajax 请求服务器

get 和 post 两种请求方式　用的是方法（ajax 中的 6 个）

abort()　　停止当前请求

getAllResponseHeaders()　　作为字符串返回完整的 headers

getResponseHeader("headerLabel")　　作为字符串返回单个的 header 标签

open("method","URL"[,asyncFlag[,"userName"[, "password"]]]) 设置未决的请求的目标 URL，方法，和其他参数

send(content)　　发送请求

setRequestHeader("label", "value")　设置 header 并和请求一起发送

三、从 Ajax 中获取服务器输出的的数据

用的是属性（ajax 中的 6）

onreadystatechange　状态改变的事件触发器

readyState　对象状态(integer):

0 = 未初始化 1 = 读取中　2 = 已读取　3 = 交互中　4 = 完成

responseText　　服务器进程返回数据的文本版本

responseXML 服务器进程返回数据的兼容 DOM 的 XML 文档对象

status　服务器返回的状态码，如：404 = "文件未找到"、200 ="成功"

statusText　服务器返回的状态文本信息

实例一

前台页

```
<script>
        function getAjax(){
                var request =false;
                if(window.XMLHttpRequest){
                        request =new XMLHttpRequest()
                        if(request.overrideMimeType){
                                request.overrideMimeType('text/xml');
                        }
                }else if(window.ActiveXObject){
                        var                                                 versions
=['Microsoft.XMLHTTP','MSXML.XMLHTTP','Microsoft.XMLHTTP','Msxml2.XMLHTTP.7.0',
'Msxml2.XMLHTTP.6.0','Msxml2.XMLHTTP.5.0','Msxml2.XMLHTTP.4.0'];
                        for(var i =0;i < versions.length; i++){
                                try{
                                        request =new ActiveXObject(versions[i]);

                                        if(request){
                                                return request;
                                        }
                                }catch(e){
                                        request =false;
```

```
                                }
                        }
                }
                return request;
        }

        function show(){
                var ajax = getAjax();

                ajax.onreadystatechange =function(){
                        alert(ajax.readyState);//在状态改变时显示 4 种不同的状态码
                        if(ajax.readyState ==4){
                                if(ajax.status ==200){
                                        alert(ajax.responseText);
                                }
                        }
                }
        /*
                //POST 方法提交数据
                ajax.open('post','test.php',true);

ajax.setRequestHeader('Content-Type','application/x-www-form-urlencoded');
                ajax.send("name=zhangsan&email=ab@aa.com&rand="+Math.random());
        */

                //GET 方法提交数据

ajax.open('get','test.php?name=zhangsan&email=ab@aa.com&rand='+Math.random(),true);
                ajax.send(null);

        }

        document.write(new Date()+'<br />');
</script>
<div></div>
<input type="button" onclick="show()" value="add" />
```

后台页面（test.php）

```
<?php
//$data = $_POST;//这样做的好处是，如果改变提交数据方式时只需更改一个变量名即可。

$data = $_GET;
$name = $data['name'];
```

```php
$email = $data['email'];
$rand = $data['rand'];

$str= $name.'---'.$email.'---'.$rand.'---'."\n";
//可以将得到的数据写入文件
$fp=fopen('data.txt','a');
fwrite($fp,$str);
fclose($fp);

echo $str;
?>
<h1>wwwwwwwwwwwwwwwwwww</h1>
```

实例二、用户名是否存在的验证

前台页面

```php
<script src="ajax3.0.js"></script>
<script>
        function show(obj){
                /*
                //get 方法提交数据
                Ajax().get("yan.php?username="+obj.value,function(data){
                        alert(data);
                });
                */
                //post 方法提交数据
                Ajax().post('yan.php',{username:obj.value},function(data){
                        alert(data);
                });
        }
</script>
<form>
        username:<input type="text" onblur="show(this)" name="username" />
</form>

<script>
        document.write(new Date()+'<br />');
</script>
```

验证页面 yan.php

```php
<?php
header('Content-Type:text/html;charset=utf-8');
//$name=$_GET['username'];
$name=$_POST['username'];
try{
        $pdo = new PDO('mysql:host=localhost;dbname=db31','root','123456');
}catch(PDOException $e){
```

```
        echo $e->getMessage();
}
$sql = 'select * from users where name=?';
$stmt=$pdo->prepare($sql);
$stmt->execute(array($name));
if($stmt->rowCount()>0){
        echo '用户名'.$name.'已经存在!';
}else{
        echo '用户名'.$name.'可以使用!';
}
```

实时改变状态：

数据表

```
Create Table: CREATE TABLE `test` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `stat` enum('1','0') NOT NULL,
  PRIMARY KEY  (`id`)
) ENGINE=MyISAM AUTO_INCREMENT=7 DEFAULT CHARSET=utf8
INSERT INTO `test` VALUES (1, '0');
INSERT INTO `test` VALUES (2, '1');
INSERT INTO `test` VALUES (3, '0');
INSERT INTO `test` VALUES (4, '0');
INSERT INTO `test` VALUES (5, '0');
INSERT INTO `test` VALUES (6, '0');
```

index.php(前端页面)

```
<html>
        <script src="../ajax3.0.js"></script>
<?php
        $pdo = new PDO('mysql:host=localhost;dbname=db31','root','123456');
        $stmt = $pdo->query("select id,stat from test");
        echo '<ul>';
        foreach($stmt as $row){
                echo '<li>';
                echo $row['id'].'---';
                echo '<span onclick="show(this,'.$row['id'].')">';
                if($row['stat']=='1')
                        echo '√';
                else
                        echo '×';
                echo '</span>';
                echo '</li>';
        }
        echo '</ul>';
?>
        <script>
```

```
                function show(obj,id){
                        var ajax = Ajax();
                        if(obj.innerText=='√'){

ajax.get('test.php?id='+id+'&stat=0',function(data){
                                        if(data=='1'){
                                                obj.innerText='×';
                                        }
                                })
                        }else{

ajax.get('test.php?id='+id+'&stat=1',function(data){
                                        if(data=='1'){
                                                obj.innerText='√';
                                        }
                                })
                        }
                }
        </script>
</html>
```

test.php（后端页面）

```
<?php
try{
        $pdo=new PDO('mysql:host=localhost;dbname=db31','root','123456');
}catch(PDOException $e){
        echo $e->getMessage();
}
$sql = "update test set stat=? where id=?";
$stmt=$pdo->prepare($sql);
$stmt->execute(array($_GET['stat'],$_GET['id']));
if($stmt->rowCount() > 0)
        echo '1';
else
        echo '0';
?>
```

实例——ajax 分页

ajax3.0.js（ajax 操作类）

index.html 前台页面

```
<html>
        <head>
                <title>Ajax 分页</title>
                <meta                                   http-equiv="content-type"
content="text/html;charset=utf-8" />
```

```
        <script src="./ajax3.0.js"></script>
    </head>
    <body>
        <div id="page">
            页面加载中....
        </div>
        <script>
            //设置一个数组作为一个缓存变量
            var cache=new Array();
            //存放页面对象
            var content=document.getElementById('page');

            //每个页面的请求方法
            function setPage(url){
                //如果没有定义 cache[url]，就说明这个页面的数据还没有取过
                if(typeof(cache[url])=='undefined'){
                    //使用 Ajax 的 get 方法取数据
                    Ajax().get(url,function(data){
                        //取过来的内容放入 content 对象中

                        content.innerHTML=data;
                        //再把取过来的内容放入缓存中
                        cache[url]=data;
                    })
                    //调试
                    alert(url);
                }else{
                    content.innerHTML=cache[url];
                }
            }
            setPage('test.php?page=1');
            document.write(new Date()+'<br />');
        </script>
    </body>
</html>
```

test.php（获取操作数据页）

```
<?php
include 'page.class.php';
$pdo=new PDO('mysql:host=localhost;dbname=db31','root','123456');
$stmt=$pdo->prepare('select count(*) as count from users');
$stmt->execute();
$row=$stmt->fetch();
$total=$row['count'];
```

```php
$page=new Page($total,5);
$stmt=$pdo->prepare("select * from users order by id".$page->limit);
$stmt->execute();
echo '<table border="1" width="900" align="center">';
while($row=$stmt->fetch(PDO::FETCH_NUM)){
        echo '<tr>';
        foreach($row as $col){
                echo '<td>'.$col.' </td>';
        }
        echo '</tr>';
}
echo '<tr><td colspan="12" align="right">'.$page->fpage().'</td></tr>';
echo '<table>';
```

page.class.php（分页类）

```php
<?php
        class Page {
                private $total;
                private $num;
                private $limit;
                private $cpage;
                private $pnum;
                private $uri;
                private $info=array("head"=>"条记录", "first"=>"首页", "next"=>"
下一页", "prev"=>"上一页", "last"=>"末页");
                function __construct($total, $num){
                        $this->total=$total;
                        $this->num=$num;

                        $this->cpage=!empty($_GET["page"]) ? $_GET["page"] : 1;

                        $this->pnum=ceil($total/$num);
                        if($this->cpage < 1)
                                $this->cpage=1;

                        if($this->cpage > $this->pnum)
                                $this->cpage=$this->pnum;

                        $this->limit=$this->getLimit();

                        $this->uri=$this->geturi();

                }

                function set($key, $value=""){
```

```
                if(array_key_exists($key, $this->info)) {
                        $this->info[$key]=$value;
                }
                return $this;
        }

        private function getLimit(){
                $str="Limit";
                $offset=($this->cpage-1)*$this->num;

                $str.="{$offset},{$this->num}";
                return $str;
        }

        private function geturi(){
                //获取当前的url，如果没有参数时，就没有?，在没有?的时候就在 URL 上加上 ？
                $url=strstr($_SERVER["REQUEST_URI"],"?") ? $_SERVER["REQUEST_URI"] : $_SERVER["REQUEST_URI"]."?";

                //把一个完整的 URL 分成每一部分，每个部分都在放在$arr 数组， 其中数组下标 $arr[path]是路径和文件部分
                //$arr["query"] 是参数或是查询字符串 page=5&cid=6&aaa=bbb
                $arr=parse_url($url);

                if(isset($arr["query"])){
                        // 将参数转成数组，  $shu=array("cid"=>6,"aaa"=>bbb)
                        parse_str($arr["query"], $shu);
                        unset($shu["page"]);
                        // http_buid_query   cid=6&aaa=bbb
                        $url=$arr["path"].'?'.http_build_query($shu);
                }

                return $url;
        }

        function __get($proName){
                if($proName=="limit"){
                        return $this->limit;
                }
        }
```

```
        private function first(){
                $str="";
                if($this->cpage!=1){
                        $prev=($this->cpage > 1) ? $this->cpage-1 : 1;
                        $str.='  <a href="javascript:setPage(\''.$this->uri.'&page=1\')">'.$this->info["first"].'</a>  ';
                        $str.='  <a href="javascript:setPage(\''.$this->uri.'&page='.$prev.'\')">'.$this->info["prev"].'</a>  ';
                }
                return $str;
        }
        private function last(){
                $str="";
                if($this->cpage!=$this->pnum && $this->pnum!=0){

                        $next=($this->cpage < $this->pnum) ? $this->cpage+1 : $this->pnum;
                        $str.='  <a href="javascript:setPage(\''.$this->uri.'&page='.$next.'\')">'.$this->info["next"].'</a>  ';
                        $str.='  <a href="javascript:setPage(\''.$this->uri.'&page='.$this->pnum.'\')">'.$this->info["last"].'</a>  ';
                }
                return $str;
        }

        private function list1(){
                $str="";

                for($i=4; $i > 0; $i--){
                        $page=$this->cpage-$i;
                        if($page > 0)
                                $str.=' <a href="javascript:setPage(\''.$this->uri.'&page='.$page.'\')">'.$page.'</a> ';
                }

                if($this->pnum > 1)
                        $str.=' '.$this->cpage.' ';
```

```
                for($i=1; $i < 5; $i++){
                        $page=$this->cpage+$i;

                        if($page <= $this->pnum)
                                $str.=' <a href="javascript:setPage(\''.$this->uri.'&page='.$page.'\')">'.$page.'</a> ';
                }

                return $str;

        }


        private function go(){
                return  '<input  type="text"  style="width:20px"><input style="width:25px" type="button" value="GO">';
        }

        private function start(){
                if($this->total == 0){
                        return 0;
                }
                return ($this->cpage-1)*$this->num+1;
        }

        private function stop(){
                if($this->cpage==$this->pnum){
                        return $this->total;
                }else if($this->total==0){
                        return 0;
                }else{
                        return $this->cpage*$this->num;
                }
        }

        private function  cpage() {
                if($this->total == 0){
                        return 0;
                }else{
                        return $this->cpage;
                }
```

```
                }

                private function num(){
                        if($this->total==0)
                                return 0;
                        else
                                return($this->stop()-$this->start()+1);
                }

                function fpage(){

                        $args=func_get_args();

                        if(count($args) > 0)
                                $arr=$args;
                        else
                                $arr=array(0,1,2,3,4,5,6,7);

                        $str[0]="                                          共
<b>{$this->total}</b>{$this->info["head"]} ";
                        $str[1]="        本       页       显       示
<b>".$this->start()."-".$this->stop()."</b> 条 ";
                        $str[2]=" 本页 <b>".$this->num()."</b> 条 ";

$str[3]=" <b>".$this->cpage()."/{$this->pnum}</b> ";


                        $str[4]=$this->first();

                        $str[5]=$this->list1();
                        $str[6]=$this->last();
                        $str[7]=$this->go();

                        $p="";
                        foreach($arr as $i){
                                $p.=$str[$i];
                         }
                        return $p;
                }
        }
```

JSON 数据

PHP 生成 JSON 数据

```
<?php
```

```php
$arr=array('name'=>'zhangsan','age'=>30,'sex'=>'female','achievement'=>array('math'=>'30','java'=>'56','english'=>'78'));
echo json_encode($arr);
```

输出：

```
{"name":"zhangsan","age":30,"sex":"female","achievement":{"math":"30","java":"56","english":"78"}}
```

用 JavaScript 将 JSON 数据生成对象

```
<script>
        var json='{"name":"zhangsan","age":30,"sex":"female","achievement":{"math":"30","java":"56","english":"78"}}';
        var obj=eval('('+json+')');
        //eval('var obj='+json);
        for(key in obj){
                if(typeof(obj[key])=='object'){
                        for(key1 in obj[key]){

document.write(key+':'+key1+':'+obj[key][key1]+'<br />');
                        }
                }else
                        document.write(key+':'+obj[key]+'<br />');
        }
</script>
```

输出：

```
name:zhangsan
age:30
sex:female
achievement:math:30
achievement:java:56
achievement:english:78
```

可以使 JSON 数据用 JavaScript 生成对象，然后作为 Ajax 中使用 POST 方法提交的数据。

jQuery

jquery j--javascript query--查询

用尽量少的代码完成尽可能多的功能

100 行 写的程序 -不好理解

10 行 ---- 好理解

使用 jquery 开发的程序 可以直接有兼容性 各种浏览器

53k

```
$(function(){//    等    同    于    $(document).ready(function(){})    以    及    window.onload=function(){}，在页面加载完成后运行
        //找对象
```

```
        //$('#one').text('wwwwwwwwww');
        //操作复合容易（数组不用遍历）
        //$('div').text('mmmmmmmmmmmmmmmmmmmmmm');
/*      //操作设置内容容易

        $('div').text('aaaaaaa');
        $('div').html('<b>bbbbbbbbbbbb</b>');
        $('input').val('zhangsan');
        //获取内容
        $('div').text(); $('div').html(); $('input').val();
        //属性操作
        $a('a').attr('href','www.baidu.com');
        $a('a').attr('href');
*/
        //操作样式容易
/*      $('#one').css('background-color','red');
        $('#one').css('font-size','3cm');
        $('#one').css('color','yellow');
        alert($('#one').css('font-size'));
*/
        //连贯操作

//$('div').html('wwwwwwwww').css('font-size','3cm').css('background-color','yel
low').attr('align','center');
        //$('<input
type="text">').css('background-color','red').appendTo($('div'));
        //$('<h2></h2>').text('hello').appendTo($('#one'));

        //事件
/*
        $('div').click(function(){
                $(this).text('wwwwwww');
        });
*/
        //$('#two').siblings('div').css('background-color','red');
/*      $.get('one.php',null,function(data){
                alert(data);
        });
*/
});
```

one.php

```
<?php
echo 'this is an event!';
```

实例：可以编辑的 USER 表

数据表

```
CREATE TABLE `users` (
  `id` int(3) unsigned NOT NULL auto_increment COMMENT '用户编号',
  `name` varchar(16) NOT NULL COMMENT '真实姓名',
  `sex` enum('male','female') NOT NULL COMMENT '性别',
  `age` tinyint(3) unsigned default NULL,
  `email` varchar(100) NOT NULL COMMENT '电子邮箱',
  PRIMARY KEY  (`id`),
) ENGINE=InnoDB AUTO_INCREMENT=155 DEFAULT CHARSET=utf8
```

conn.inc.php（数据库连接文件）

```
<?php
try{
        $pdo=new PDO('mysql:host=localhost;dbname=db31','root','123456');
}catch(PDOException $e){
        echo $e->getMessage();
}
```

index.php(前台用户信息页)

```
<html>
        <head>
                <title>可以编辑的表格</title>
                <meta                                   http-equiv="content-type"
content="text/html;charset=utf-8" />
                <link rel="stylesheet" type="text/css" href="css/table.css" />
                <script src="js/jquery.js" ></script>
                <script src="js/table.js"></script>
        </head>
        <body>
                <table>
                        <caption><h1>可以编辑的 USER 表</h1></caption>
                        <tr>
                                <th>ID</th>
                                <th>NAME</th>
                                <th>AGE</th>
                                <th>SEX</th>
                                <th>EMAIL</th>
                        </tr>
                        <?php
                                include 'conn.inc.php';
                                $sql="select  id,name,age,sex,email  from  users
order by id limit 20";
                                $stmt=$pdo->prepare($sql);
                                $stmt->execute();
```

```
while(list($id,$name,$age,$sex,$email)=$stmt->fetch(PDO::FETCH_NUM)){
                                        echo '<tr>';
                                        echo '<td class="first">'.$id.'</td>';//
添加 class="first"是为了便于 JavaScript 对第一列进行操作
                                        echo '<td>'.$name.'</td>';
                                        echo '<td>'.$age.'</td>';
                                        echo '<td>'.$sex.'</td>';
                                        echo '<td>'.$email.'</td>';
                                        echo '</tr>';
                                    }
                            ?>
                    </table>
            </body>
</html>
```

css/table.css(表格样式文件)

```
body{
        font-size:12px;
        background:#FFFFFF;
        text-align:center;
}
table{
        text-align:left;
        border:1px solid #050;
        width:800px;
        margin-left:auto;
        margin-top:auto;
        border-collapse:collapse;
}
th{
        text-align:center;
        background:#efe;
}
td,th{
        border:1px solid #050;
        width:160px;
        padding:0px;
        height:20px;
}
```

jquery.js(jQuery 文件)

table.js（js 操作文件）

```
//加载
$(function(){
        $('tr:even').css('background-color','#AAAAFA');
```

```
        //找到需要操作的表格，并加事件
        $('td:not(.first)').click(function(){
                //0.先要判断一下，如果已经使用了 input 就不再使用
                //1.先将这个表格的内容拿出来
                //2.将表格内容清空处理
                //3.在表格内容中加一个 input 标记
                //4.将值放入 input 标记中
                //5.将 input 的样式调成和表格原来一样
                if($(this).children('input').length > 0)
                        return;

                //先将 td 中的数据存放到临时变量 data 中，便于在加入 input 标签后赋于该值，以及在取消输出时，还原原来的数据
                var data=$(this).text();
                //将表格中的数据清空
                $(this).html('');
                //将当前的 td 对象赋于变量 td，便于以后的引用 td 对象
                var td=$(this);
                $('<input type="text">').val(data).css('border','0').css('background-color','transparent').css('width','100%').css('height','100%').css('font-family',td.css('font-family')).css('font-size',td.css('font-size')).appendTo($(this)).trigger('focus').trigger('select').keyup(function(event){
                        switch(event.keyCode){
                                case 13:
                                        save(td,$(this));
                                        break;
                                case 27:
                                        td.html(data);
                                        break;
                        }
                }).blur(function(){
                        save(td,$(this));
                });
        });
});
//保存修改的数据
function save(td,input){
        //在此用 html 方法将 input 中的值作为 td 中的内容，因此 td 中没有 input 元素
        td.html(input.val());

        //得到该行的所有单元格
        var tds=td.parent('tr').children('td');
```

```
        var id=tds.eq(0).text();
        var name=tds.eq(1).text();
        var age=tds.eq(2).text();
        var sex=tds.eq(3).text();
        var email=tds.eq(4).text();


$.post('save.php',{id:id,name:name,age:age,sex:sex,email:email},function(data){
                alert(data);
        })
}
```

save.php（修改数据文件）

```
<?php
include 'conn.inc.php';
$sql="update users set name=:name,age=:age,sex=:sex,email=:email where id=:id";
$stmt=$pdo->prepare($sql);
$stmt->execute($_POST);
if($stmt->rowCount() > 0)
        echo '修改成功!';
else
        echo '修改失败!';
```

jQuery 动画

```
<script src="jquery.js"></script>
<button id="left">left</button> <button id="right">right</button>
<div class="block" style="position:absolute;left:100px;top:300px;">wwwwww</div>

<script>
        $('#right').click(function(){
                $('.block').animate({left:'+50px'},'slow');
        })
        $('#left').click(function(){
                $('.block').animate({left:'-50px'},'slow');
        });
</script>
```

利用 jQuery 做的 Ajax-Tab 面板

card.html（前台页面）

```
<html>
        <head>
                <title></title>
                <meta                                   http-equiv="content-type"
content="text/html;charset=utf-8" />
                <link rel="stylesheet" type="text/css" href="css/card.css" />
                <script src="js/jquery.js"></script>
```

```
                <script src="js/card.js"></script>
        </head>
        <body>
                <div id="card">
                        <div id="tit">
                                <h3 class="titin"><a href="data.php?cid=1">第一
项</a></h3>
                                <h3><a href="data.php?cid=2">第二项</a></h3>
                                <h3><a href="data.php?cid=3">第三项</a></h3>
                        </div>
                        <div id="content">
                        </div>
                </div>
        </body>
</html>
```

css/card.css

```
#card{
        width:300px;
        height:200px;
}
#tit{
        width:100%;
        height:25px;
}
#tit h3{
        float:left;
        background:#ccc;
        border:2px solid #fff;
        font-size:14px;
        width:80px;
        margin:0px;
        padding:0px;
        text-align:center;
        line-height:25px;
}
#content{
        width:100%;
        height:175px;
        background:#888;
}
#tit .titin{
        background:#888;
        border:2px solid #888;
}
```

js/jquery.js

js/card.js

```
/*
//这种方法对于提取所操作元素的索引不太方便
$(function(){
        var dt=null;
        $('#tit h3').mouseover(function(){
                var h3=$(this);
                dt=setTimeout(function(){

h3.addClass('titin').siblings('h3').removeClass('titin');
                },300);
        }).mouseout(function(){
                clearTimeout(dt);
        });
});
*/
/*
$(function(){
        //这种方法可以获取所查看栏目在所有栏目中的序号
        $('#tit h3').each(function(i){
                $(this).mouseover(function(){
                        var h3=$(this);
                        dt=setTimeout(function(){
                                alert(i);

h3.addClass('titin').siblings('h3').removeClass('titin');
                        },300);
                }).mouseout(function(){
                        clearTimeout(dt);
                });
        });
});
*/
$(function(){
        //dt 用作定时器
        var dt=null;
        //默认的所要请求的数据
        var url='data.php?cid=1';
        show(url);
        $('#tit h3').hover(function(){
                //将当前对象存入 h3 变量中便于在定时器中引用
                var h3=$(this);
                //让鼠标在标题上停留 300 毫秒后将该栏目样式改为选中状态的样式,如
```

```
果停留时间小于 300 毫秒，那么表示鼠标滑过，不是要看该栏目中的内容，这样可以减少对服务器不必要的请求
                dt=setTimeout(function(){
                        //给当前所要查看的栏目加上选中状态样式,对其其他栏目去掉选中状态样式

h3.addClass('titin').siblings('h3').removeClass('titin');
                        var url=h3.children('a').eq(0).attr('href');
                        show(url);
                },300);
        },function(){
                clearTimeout(dt);
        });
});
var cache=new Array();
function show(url){
        //如果所要请求的数据没有缓存，则请求服务器，并将请求的数据放入缓存中,如果缓存了，使用缓存的数据
        if(typeof(cache[url])=='undefined'){
                $.get(url,function(data){
                        $('#content').html(data);
                        cache[url]=data;
                });
                alert(url);
        }else{
                $('#content').html(cache[url]);
        }
}
```

data.php

```
<?php
$pdo=new PDO('mysql:host=localhost;dbname=db31','root','123456');
$sql="select id,title from card where cid=?";
$stmt=$pdo->prepare($sql);
$stmt->execute(array($_GET['cid']));

echo '<ul>';
while($row=$stmt->fetch(PDO::FETCH_ASSOC)){
        echo '<li><a href="artical.php?id='.$row['id'].'">';
        echo $row['title'];
        echo '</a></li>';
}
echo '<ul>';
```