JS设置/获取日期和时间

<HTML>

<HEAD>

<TITLE> New Document </TITLE>

<script language="JavaScript">

//单个数字配零

function getDouble(number){

var numbers=["0","1","2","3","4","5","6","7","8","9"];

for(var i=0;i<numbers.length;i++){

if(numbers[i]==number){

return "0"+numbers[i];

}else if(i==9){

return number;

}

}

}

//得到当天时间

function getTodayTime(){

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

var today=new Date();

var str= (today.getYear()<1900?1900+today.getYear():today.getYear())+"年" + getDouble([today.getMonth()+1])+"月" +getDouble(today.getDate()) +"日 &nbsp;"+ days[today.getDay()]+"<br>"+getDouble(today.getHours())+":"+getDouble(today.getMinutes())+":"+getDouble(today.getSeconds());

document.getElementById('datetime').innerHTML=str;

}

//每隔一秒刷新一次

setInterval("getTodayTime()",1000);

</script>

</HEAD>

<BODY>

<div id="datetime"></div>

</BODY>

</HTML>

========================

日期函数列表:

dateObj.getTime()得到时间,dateObj.getYear()得到年份,dateObj.getFullYear()得到四位的年份

,dateObj.getMonth()得到月份,dateObj.getDate()得到日,dateObj.getDay()得到日期几,dateObj.getHours()得到小时,dateObj.getMinutes()得到分,dateObj.getSeconds()得到秒,dateObj.setTime(value)设置时间,dateObj.setYear(val)设置年,dateObj.setMonth(val)设置月,dateObj.setDate(val)设置日,dateObj.setDay(val)设置星期几,dateObj.setHours设置小时,dateObj.setMinutes(val)设置分,dateObj.setSeconds(val)设置秒 [注意:此日期时间从0开始计]43.FRAME的表示方式: [window.]frames[n].ObjFuncVarName,frames["frameName"].ObjFuncVarName,frameName.ObjFuncVarName

时间对象用来操作日期和时间。

Date(日期)型字符串，要想正确的转换为Date(日期)对象，必须用new Date(str)方式，直接用Date(str)强制转换将得到错误结果，另外转换时Date字符串的格式为"年/月/日"（也许还有其它写法，这里只测试了yyyy/mm/dd确实是可行的），而另一种很常见的"年-月-日"的表示方式，转换后将得到错误结果

eg:

var a = Date.parse("2009/04/22 16:30");

var now = new Date(a);

举例

Return today's date and time

使用Date()方法得到今天的日期

getTime()

用getTime()来计算1970年到现在之间的时间差距

setFullYear()

使用getFullYear()来设置指定的日期

toUTCString()

使用UTCString()来将今天的日期转换成字符串（依据UTC）

getDay()

使用getDay()和一数组来书写星期几，并不是简单的数字

Display a clock

怎样在你的页上显示一个时钟

Defining Dates

定义日期

时间对象用来操作日期跟时间

我们通过一个新的关键字来定义一个时间对象。下面的代码行就定义了一个时间对象（称作myDate）:

var myDate=new Date()

Note:

注意：时间对象将自动把当前的日期和时间作为初始值！

Manipulate Dates

操作（设置）时间

我们可以轻易地用有效的时间对象来操作（设置）日期或时间

在下面的例子中我们设置一个时间对象来指定日期（2010年一月14号）:

var myDate=new Date()

myDate.setFullYear(2010,0,14)

下面的例子我们将一时间对象设置为未来的5天：

var myDate=new Date()

myDate.setDate(myDate.getDate()+5)

Note:

注意：如果给一日期增加天改为给月或年增加的话，一些变化会由时间对象自动处理！

Comparing Dates

比较时间（日期）

时间对象同样使用在比较两个日期（时间）

下面的例子将今天的日期和2010年1月14号作比较：

var myDate=new Date()

myDate.setFullYear(2010,0,14)

var today = new Date()

if (myDate>today)

alert("Today is before 14th January 2010")

else

alert("Today is after 14th January 2010")

Complete Date Object Reference

完整的时间对象参考

For a complete reference of all the properties and methods that can be used with the Date object, Go to our complete Date object reference.

Date[日期]对象中的所有属性和方法参数，我们将在完整Date对象参数 中罗列说明。

The reference contains a brief description and examples of use for each property and method!

我们将列举简要说明和典型案例来讲解每个参数的属性和方法的用法。

Date Object Methods

日期对象方法

FF: Firefox, N: Netscape, IE: Internet Explorer

FF:火狐，N:网景，IE

Method

方法 Description

描述 FF N IE

Date() Returns today's date and time

返回今天的日期和时间 1 2 3

getDate() Returns the day of the month from a Date object (from 1-31)

返回月中的第几天(1到31) 1 2 3

getDay() Returns the day of the week from a Date object (from 0-6)

返回一周中的第几天(0到6) 1 2 3

getMonth() Returns the month from a Date object (from 0-11)

返回月份数(0到11) 1 2 3

getFullYear() Returns the year, as a four-digit number, from a Date object

返回完整的年份数 1 4 4

getYear() Returns the year, as a two-digit or a four-digit number, from a Date object. Use getFullYear() instead !!

返回年份，可以是两位的或是四位的 1 2 3

getHours() Returns the hour of a Date object (from 0-23)

返回日期对象的小时数（0到23） 1 2 3

getMinutes() Returns the minutes of a Date object (from 0-59)

返回日期对象的分钟（0到59） 1 2 3

getSeconds() Returns the seconds of a Date object (from 0-59)

返回日期对象的秒（0到59） 1 2 3

getMilliseconds() Returns the milliseconds of a Date object (from 0-999)

返回毫秒（0到999） 1 4 4

getTime() Returns the number of milliseconds since midnight Jan 1, 1970

从1970年1月1号午夜到现在一共花去的毫秒数 1 2 3

getTimezoneOffset() Returns the difference in minutes between local time and Greenwich Mean Time (GMT)

本地时间和GMT相差多少分钟 1 2 3

getUTCDate() Returns the day of the month from a Date object according to universal time (from 1-31)

依据国际时间来得到月中的第几天（1到31） 1 4 4

getUTCDay() Returns the day of the week from a Date object according to universal time (from 0-6)

依据国际时间来得到现在是星期几（0到6） 1 4 4

getUTCMonth() Returns the month from a Date object according to universal time (from 0-11)

依据国际时间来得到月份（0到11） 1 4 4

getUTCFullYear() Returns the four-digit year from a Date object according to universal time

依据国际时间来得到完整的年份 1 4 4

getUTCHours() Returns the hour of a Date object according to universal time (from 0-23)

依据国际时间来得到小时（0-23） 1 4 4

getUTCMinutes() Returns the minutes of a Date object according to universal time (from 0-59)

依据国际时间来返回分钟（0到59） 1 4 4

getUTCSeconds() Returns the seconds of a Date object according to universal time (from 0-59)

依据国际时间来返回秒（0到59） 1 4 4

getUTCMilliseconds() Returns the milliseconds of a Date object according to universal time (from 0-999)

依据国际时间来返回毫秒（0到999） 1 4 4

parse() Takes a date string and returns the number of milliseconds since midnight of January 1, 1970

或得并返回自1970年1月1号凌晨到现在一共花掉了多少毫秒 1 2 3

setDate() Sets the day of the month in a Date object (from 1-31)

设置日 1 2 3

setMonth() Sets the month in a Date object (from 0-11)

设置月 1 2 3

setFullYear() Sets the year in a Date object (four digits)

设置年份 1 4 4

setYear() Sets the year in the Date object (two or four digits). Use setFullYear() instead !!

用setFullYear()来取代 1 2 3

setHours() Sets the hour in a Date object (from 0-23)

设置小时 1 2 3

setMinutes() Set the minutes in a Date object (from 0-59)

设置分钟 1 2 3

setSeconds() Sets the seconds in a Date object (from 0-59)

设置秒 1 2 3

setMilliseconds() Sets the milliseconds in a Date object (from 0-999)

设置毫秒 1 4 4

setTime() Calculates a date and time by adding or subtracting a specified number of milliseconds to/from midnight January 1, 1970 1 2 3

setUTCDate() Sets the day of the month in a Date object according to universal time (from 1-31)

依据国际时间来设置日期 1 4 4

setUTCMonth() Sets the month in a Date object according to universal time (from 0-11)

依据国际时间来设置月 1 4 4

setUTCFullYear() Sets the year in a Date object according to universal time (four digits)

依据国际时间来设置年份 1 4 4

setUTCHours() Sets the hour in a Date object according to universal time (from 0-23)

依据国际时间来设置小时 1 4 4

setUTCMinutes() Set the minutes in a Date object according to universal time (from 0-59)

依据国际时间来设置分钟 1 4 4

setUTCSeconds() Set the seconds in a Date object according to universal time (from 0-59)

依据国际时间来设置秒 1 4 4

setUTCMilliseconds() Sets the milliseconds in a Date object according to universal time (from 0-999)

依据国际时间来设置毫秒 1 4 4

toSource() Represents the source code of an object

显示对象的源代码 1 4 -

toString() Converts a Date object to a string

将日期对象转换为字符串 1 2 4

toGMTString() Converts a Date object, according to Greenwich time, to a string. Use toUTCString() instead !!

根据格林威治时间将Date[日期]对象转换为一个字符串。可以使用toUTCString()替代这种方法 1 2 3

toUTCString() Converts a Date object, according to universal time, to a string

根据通用时间将一个Date[日期]对象转换为一个字符串 1 4 4

toLocaleString() Converts a Date object, according to local time, to a string

根据本地时间将一个Date[日期]对象转换为一个字符串 1 2 3

UTC() Takes a date and returns the number of milliseconds since midnight of January 1, 1970 according to universal time

根据通用时间将日期计算为从1970年1月1日午夜至今所经过的时间（单位：毫秒） 1 2 3

valueOf() Returns the primitive value of a Date object

返回日期对象的原始值 1 2 4

时间对象是一个我们经常要用到的对象，无论是做时间输出、时间判断等操作时都与这个对象离不开。除开JavaScript中的时间对象外，在VbScript中也有许多的时间对象，而且非常好用。下面还是按照我们的流程来进行讲解。

它是一个内置对象——而不是其它对象的属性,允许用户执行各种使用日期和时间的过程。

　方法：分为得到时间方法、设置时间方法和转换时间方法

得到时间方法：

　　getDate() 查看Date对象并返回日期

　　getDay() 返回星期几

　　getHours() 返回小时数

　　getMinutes() 返回分钟数

　　getMonth() 返回月份值

　　getSeconds() 返回秒数

　　getTime() 返回完整的时间

　　getYear() 返回年份

设置时间方法：

　　setDate() 改变Date对象的日期

　　setHours() 改变小时数

　　setMinutes() 改变分钟数

　　setMonth() 改变月份

　　setSeconds() 改变秒数

　　setTime() 改变完整的时间

　　setYear() 改变年份

转换时间方法：

　　toGMTString() 把Date对象的日期(一个数值)转变成一个GMT时间字符串,返回类似下面的值:Weds,15 June l997 14:02:02 GMT(精确的格式依赖于计算机上所运行的操作系统而变)

　　toLocaleString() 把Date对象的日期(一个数值)转变成一个字符串,使用所在计算机上配置使用的特定日期格式

　　UTC() 使用Date UTC(年、月、日、时、分、秒),以自从1970年1月1日00:00:00(其中时、分、秒是可选的)以来的毫秒数的形式返回日期

几个需要注意的地方：

1、得到日期和年和设置日期和年时间，其中很怪的问题就是不能对月份进行设置（比较的怪）：

<script language="javascript">

d = new Date();

alert(d.toLocaleString());

d.setDate(25);

alert(d.toLocaleString());

d.setYear(2000);

alert(d.toLocaleString());

</script>

2、获得年的时候最好用getFullYear()方法来做

3、由于针对月份，JS是从0开始的，因此需要对月份进行操作时要加1

下面是几个关于时间的经典而且经常会用到的例子，希望对大家会有提高的。谢谢继续关注该帖子。。。

1、将2005-8-5转换成2005-08-05格式

<script language="javascript">

var strDate = '2005-8-5';

window.alert(strDate.replace(/\b(\w)\b/g, '0$1'));

</script>

2、得到间隔天数

<script type="text/javascript">

<!--

alert("间隔天数为:"+(new Date('2005/8/15')-new Date('2003/9/18'))/1000/60/60/24+"天")

//-->

</script>

3、得到间隔时间

<script>

var d1=new Date("2004/09/16 20:08:00");

var d2=new Date("2004/09/16 10:18:03");

var d3=d1-d2;

var h=Math.floor(d3/3600000);

var m=Math.floor((d3-h\*3600000)/60000);

var s=(d3-h\*3600000-m\*60000)/1000;

alert("相差"+h+"小时"+m+"分"+s+"秒");

</script>

4、得到今天的日期

<script language="javascript">

d = new Date();

alert(d.getFullYear()+"年"+(d.getMonth()+1)+"月"+d.getDate()+"日");

</script>

6、数字日期转汉字

<html>

<head>

<title> New Document </title>

</head>

<body>

<script language=javascript>

Date.prototype.getRead = function()

{

var values = new Array("零", "一", "二", "三", "四", "五", "六", "七", "八", "九");

var returnValue, temp;

returnValue = this.getYear()+"年";

temp = (this.getMonth()+1)+"月"+this.getDate()+"日";

temp = temp.replace(/(\d)(\d)/g,"$1十$2").replace(/1十/g,"十").replace(/十0/g,"十");

returnValue += temp;

returnValue = returnValue.replace(/\d/g, function(sts){return values[parseInt(sts)]});

return returnValue;

}

var t=new Date();

document.write(t.getRead());

</script>

</body>

</html>

7、得到前N天或后N天的日期

方法一：

<script type="text/javascript">

function showdate(n)

{

var uom = new Date(new Date()-0+n\*86400000);

uom = uom.getFullYear() + "-" + (uom.getMonth()+1) + "-" + uom.getDate();

return uom;

}

window.alert("今天是："+showdate(0));

window.alert("昨天是："+showdate(-1));

window.alert("明天是："+showdate(1));

window.alert("10天前是："+showdate(-10));

window.alert("5天后是："+showdate(5));

</script>

方法二：

<script type="text/javascript">

function showdate(n)

{

var uom = new Date();

uom.setDate(uom.getDate()+n);

uom = uom.getFullYear() + "-" + (uom.getMonth()+1) + "-" + uom.getDate();

return uom;

}

window.alert("今天是："+showdate(0));

window.alert("昨天是："+showdate(-1));

window.alert("明天是："+showdate(1));

window.alert("10天前是："+showdate(-10));

window.alert("5天后是："+showdate(5));

</script>

方法三（不好意思，这个市用vsscript做的）：

<script language="vbscript">

function showdate(n)

showdate=dateadd("d",date(),n)

end function

msgbox "今天是:"&showdate(0)

msgbox "昨天是:"&showdate(-1)

msgbox "明天是:"&showdate(1)

msgbox "十天前是:"&showdate(-10)

msgbox "五天后是:"&showdate(5)

</script>

方法四：

<script language="Javascript">

Date.prototype.getDays=function(){

var \_newDate=new Date();

\_newDate.setMonth(\_newDate.getMonth()+1);

\_newDate.setDate(0);

$\_days=\_newDate.getDate();

delete \_newDate;

return $\_days;

}

function showdate(n)

{

var uom = new Date();

uom.setDate(uom.getDate()+n);

uom = uom.getFullYear() + "-" + (uom.getMonth()+1) + "-" + uom.getDate()+"\n星期"+('天一二三四五六'.charAt(uom.getDay()))+"\n本月有"+ uom.getDays()+"天";

return uom;

}

window.alert("今天是："+showdate(0));

window.alert("昨天是："+showdate(-1));

window.alert("明天是："+showdate(1));

window.alert("10天前是："+showdate(-10));

window.alert("5天后是："+showdate(5));

</script>

From: http://www.blogjava.net/parable-myth/archive/2008/01/12/174761.html

=====================

javascript 日期函数

From: http://www.3800hk.com/Article/web/JavaScript/yyjqjs/2005-08-06/Article\_47290.html

=====================

javascript对日期处理的常用方法类

<scriptlanguage="JavaScript">

<!--

//程序:常用公历日期处理程序

/\*\*//\*用相对不规则的字符串来创建日期对象,不规则的含义为:顺序包含年月日三个数值串,有间隔\*/

String.prototype.parseDate=function(){

　　varar=(this+",0,0,0").match(/d+/g);

　　returnar[5]?(newDate(ar[0],ar[1]-1,ar[2],ar[3],ar[4],ar[5])):(newDate());

}

/\*\*//\*

\*功能:根据输入表达式返回日期字符串

\*参数:dateFmt:字符串,由以下结构组成

\*　　　yy:长写年,YY:短写年mm:数字月,MM:英文月,dd:日,hh:时,

\*　　　mi:分,ss秒,ms:毫秒,we:汉字星期,WE:英文星期.

\*　　　isFmtWithZero:是否用0进行格式化,trueorfalse

\*返回:对应日期的中文字符串

\*/

Date.prototype.toString=function(dateFmt,isFmtWithZero){

　　dateFmt=(dateFmt==null?"yy-mm-ddhh:mi:ss":dateFmt);

　　if(typeof(dateFmt)!="string")

　　　　throw(newError(-1,'toString()方法的第一个参数为字符串类型!'));

　　isFmtWithZero=!!isFmtWithZero;

　　varweekArr=[["日","一","二","三","四","五","六"],["SUN","MON","TUR","WED","THU","FRI","SAT"]];

　　varmonthArr=["JAN","FEB","MAR","APR","MAY","JUN","JUL","AUG","SEP","OCT","NOV","DEC"];

　　varstr=dateFmt;

　　varo={

　　　　"yy":this.getFullYear(),

　　　　"YY":this.getYear(),

　　　　"mm":this.getMonth()+1,

　　　　"MM":monthArr[this.getMonth()],

　　　　"dd":this.getDate(),

　　　　"hh":this.getHours(),

　　　　"mi":this.getMinutes(),

　　　　"ss":this.getSeconds(),

　　　　"we":"星期"+weekArr[0][this.getDay()],

　　　　"WE":weekArr[1][this.getDay()]

　　}

　　for(variino){

　　　　str=str.replace(newRegExp(i,"g"),o[i].toString().fmtWithZero(isFmtWithZero));

　　}

　　str=str.replace(/ms/g,this.getMilliseconds().toString().fmtWithZeroD(isFmtWithZero));

　　returnstr;

}

/\*\*//\*将一位数字格式化成两位,如:9to09\*/

String.prototype.fmtWithZero=function(isFmtWithZero){

　　return(isFmtWithZero&&/^d$/.test(this))?"0"+this:this;

}

String.prototype.fmtWithZeroD=function(isFmtWithZero){

　　return(isFmtWithZero&&/^d{2}$/.test(this))?"00"+this:((isFmtWithZero&&/^d$/.test(this))?"0"+this:this);

}

/\*\*//\*功能:返回与某日期相距N天(N个24小时)的日期

\*参数:numnumber类型可以为正负整数或者浮点数,默认为1;

\*　　　　type0(秒)or1(天),默认为天

\*返回:新的Date对象

\*/

Date.prototype.dateAfter=function(num,type){

　　num=(num==null?1:num);

　　if(typeof(num)!="number")thrownewError(-1,"dateAfterDays(num,type)的num参数为数值类型.");

　　type=(type==null?1:type);

　　vararr=[1000,86400000];

　　returnnewDate(this.valueOf()+num\*arr[type]);

}

//判断是否是闰年,返回true或者false

Date.prototype.isLeapYear=function(){

　　varyear=this.getFullYear();

　　return(0==year%4&&((year%100!=0)||(year%400==0)));

}

//返回该月天数

Date.prototype.getDaysOfMonth=function(){

　　return(newDate(this.getFullYear(),this.getMonth()+1,0)).getDate();

}

//日期比较函数,参数date:为Date类型,如this日期晚于参数:1,相等:0早于:-1

Date.prototype.dateCompare=function(date){

　　if(typeof(date)!="object"||!(/Date/.test(date.constructor)))

　　　　thrownewError(-1,"dateCompare(date)的date参数为Date类型.");

　　vard=this.getTime()-date.getTime();

　　returnd>0?1:(d==0?0:-1);

}

/\*\*//\*功能:返回两日期之差

\*参数:pd　PowerDate对象

\*　　type:返回类别标识.yy:年,mm:月,ww:周,dd:日,hh:小时,mi:分,ss:秒,ms:毫秒

\*　　intOrFloat:返回整型还是浮点型值0:整型,不等于0:浮点型

\*　　output:输出提示,如:时间差为#周!

\*/

Date.prototype.calDateDistance=function(date,type,intOrFloat,output){

　　if(typeof(date)!="object"||!(/Date/.test(date.constructor)))

　　　　thrownewError(-1,"calDateDistance(date,type,intOrFloat)的date参数为Date类型.");

　　type=(type==null?'dd':type);

　　if(!((newRegExp(type+",","g")).test("yy,mm,ww,dd,hh,mi,ss,ms,")))

　　　　thrownewError(-1,"calDateDistance(pd,type,intOrFloat,output)的type参数为非法.");

　　variof=(intOrFloat==null?0:intOrFloat);

　　varnum=0;

　　varo={

　　　　"ww":7\*86400000,

　　　　"dd":86400000,

　　　　"hh":3600000,

　　　　"mi":60000,

　　　　"ss":1000,

　　　　"ms":1

　　}

　　switch(type){

　　　　case"yy":num=this.getFullYear()-date.getFullYear();break;

　　　　case"mm":num=(this.getFullYear()-date.getFullYear())\*12+this.getMonth()-date.getMonth();break;

　　　　default:

　　　　　　varsub=this.valueOf()-date.valueOf();

　　　　　　if(o[type])

　　　　　　　　num=(sub/o[type]).fmtRtnVal(iof);

　　　　　　break;

　　}

　　return(output?output.replace(/#/g,""+num+""):num);

}

//返回整数或者两位小数的浮点数

Number.prototype.fmtRtnVal=function(intOrFloat){

　　return(intOrFloat==0?Math.floor(this):parseInt(this\*100)/100);

}

//根据当前日期所在年和周数返回周日的日期

Date.prototype.RtnByWeekNum=function(weekNum){

　　if(typeof(weekNum)!="number")

　　　　thrownewError(-1,"RtnByWeekNum(weekNum)的参数是数字类型.");

　　vardate=newDate(this.getFullYear(),0,1);

　　varweek=date.getDay();

　　week=(week==0?7:week);

　　returndate.dateAfter(weekNum\*7-week,1);

}

//根据日期返回该日期所在年的周数

Date.prototype.getWeekNum=function(){

　　vardat=newDate(this.getFullYear(),0,1);

　　varweek=dat.getDay();

　　week=(week==0?7:week);

　　vardays=this.calDateDistance(dat,"dd")+1;

　　return((days+6-this.getDay()-7+week)/7);

}

//-->

</script>

<scriptlanguage="JavaScript">

<!--

vardw=document.write;

vardate2="2005-12-30".parseDate();

dw(date2,"<br>");

vardate3="2006-1-10111111".parseDate();

dw(date2.calDateDistance(date3,null,null,"时间差为#天!"));

//alert(newDate(2000,2,-29));

//-->

</script>