**Window和js的内置对象和方法**

**window内置对象**

**一、navigator**

**代表着浏览器本身的信息：**

**代码名：navigator.appCodeName**

**浏览器语言：navigator.browserLanguage**

**操作系统的语言：navigator.systemLanguage**

**操作系统自然设置语言：navigator.userLanguage**

**cpu等级：navigator.cpuClass**

**客户机发送服务器的 user-agent 头部的值:navigator.userAgent**

**名称：navigator.appName**

版本号：navigator.appVersion

运行平台：navigator.platform

是否支持cookie:navigator.cookieEnabled

是否启用java:navigator.javaEnabled();

是否启用数据污点：navigator.taintEnabled

二、frames[]

描述浏览器框架组合，window.frames返回当前窗口的所有frame框架

三、location

描述浏览器地址

四、screen

窗口屏幕信息

高度：screen.height

宽度：screen.width

颜色深度：screen.colorDepth

五、history

描述浏览历史

history.back()

history.forward()

history.go(-2)返回两次

六、document

是最重要也是最复杂的。

**js内置对象与函数**

**1.Date**  
属性(1):  
constructor      所建立对象的函数参考  
prototype       能够为对象加入的属性和方法

方法(43):  
getDay()        返回一周中的第几天(0-6)  
getYear()       返回年份.2000年以前为2位,2000(包含)以后为4位  
getFullYear()     返回完整的4位年份数  
getMonth()      返回月份数(0-11)  
getDate()       返回日(1-31)  
getHours()       返回小时数(0-23)  
getMinutes()      返回分钟(0-59)  
getSeconds()      返回秒数(0-59)  
getMilliseconds() 返回毫秒(0-999)  
getUTCDay()      依据国际时间来得到现在是星期几(0-6)  
getUTCFullYear()    依据国际时间来得到完整的年份  
getUTCMonth()     依据国际时间来得到月份(0-11)  
getUTCDate()      依据国际时间来得到日(1-31)  
getUTCHours()     依据国际时间来得到小时(0-23)  
getUTCMinutes()    依据国际时间来返回分钟(0-59)  
getUTCSeconds()    依据国际时间来返回秒(0-59)  
getUTCMilliseconds()依据国际时间来返回毫秒(0-999)  
getTime()       返回从1970年1月1号0:0:0到现在一共花去的毫秒数  
getTimezoneoffset() 返回时区偏差值,即格林威治平均时间(GMT)与运行脚本的计算机所处时区设置之间相差的分钟数)  
parse(dateString)   返回在Date字符串中自从1970年1月1日00:00:00以来的毫秒数  
setYear(yearInt)    设置年份.2位数或4位数  
setFullYear(yearInt)设置年份.4位数  
setMonth(monthInt) 设置月份(0-11)  
setDate(dateInt)    设置日(1-31)  
setHours(hourInt) 设置小时数(0-23)  
setMinutes(minInt) 设置分钟数(0-59)  
setSeconds(secInt) 设置秒数(0-59)  
setMilliseconds(milliInt) 设置毫秒(0-999)  
setUTCFullYear(yearInt) 依据国际时间来设置年份  
setUTCMonth(monthInt) 依据国际时间来设置月(0-11)  
setUTCDate(dateInt)     依据国际时间来设置日(1-31)  
setUTCHours(hourInt)    依据国际时间来设置小时  
setUTCMinutes(minInt) 依据国际时间来设置分钟  
setUTCSeconds(secInt)    依据国际时间来设置秒  
setUTCMilliseconds(milliInt)依据国际时间来设置毫秒  
setTime(timeInt)    设置从1970年1月1日开始的时间.毫秒数  
toGMTString()     根据格林威治时间将Date对象的日期(一个数值)转变成一个GMT时间字符串,如:Weds,15 June l997 14:02:02 GMT  
toUTCString()     根据通用时间将一个Date对象的日期转换为一个字符串  
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  
toSource()       显示对象的源代码  
toString()       将日期对象转换为字符串  
UTC(yyyy, mm, dd, hh, mm, ss, msec)返回从格林威治标准时间到指定时间的差距,单位为毫秒  
valueOf()       返回日期对象的原始值

**2.Math**  
属性:  
constructor      所建立对象的函数参考  
prototype       能够为对象加入的属性和方法  
E           欧拉常量,自然对数的底(约等于2.718)  
LN2          2的自然对数(约等于0.693)  
LN10          10的自然对数(约等于2.302)  
LOG2E         以2为底的e的对数.(约等于1.442)  
LOG10E         以10为底的e的对数(约等于0.434)  
PI           ∏的值(约等于3.14159)  
SQRT1\_2        1/2(0.5)的平方根(即l除以2的平方根,约等于o.707)  
SQRT2         2的平方根(约等于1.414)

方法:  
abs(x)     返回数字的绝对值  
acos(x)    返回数字的反余弦值  
asin(x)    返回数字的反正弦值  
atan(x)    返回位于-PI/2 和 PI/2 的反正切值  
atan2(y,x) 返回（x,y）位于 -PI 到 PI 之间的角度  
ceil(x)    返回 x 四舍五入后的最大整数  
cos(x)     返回一个数字的余弦值  
exp(x)     返回 E^x 值  
floor(x)    返回 x 四舍五入后的最小整数  
log(x)     返回底数为E的自然对数  
max(x,y)    返回 x 和 y 之间较大的数  
min(x,y)    返回 x 和 y 之间较小的数  
pow(x,y)    返回 y^x 的值  
random()    返回位于 0 到 1 之间的随机函数  
round(x)    四舍五入后取整  
sin(x)     返回数字的正弦值  
sqrt(x)    返回数字的平方根  
tan(x)     返回一个角度的正切值  
toSource() 显示对象的源代码  
valueOf() 返回数学对象的原始值

**3.Number**  
属性:  
MAX\_VALUE       The largest possible value a number in JavaScript can have 1.7976931348623157E+308  
MIN\_VALUE       The smallest possible value a number in JavaScript can have 5E-324  
NaN          Equal to a value that is not a number.  
NEGATIVE\_INFINITY A value that is less than MIN\_VALUE.  
POSOTIVE\_INFINITY A value that is greater than MAX\_VALUE.  
prototype       A static property of the Number object

方法:  
toString       Returns a string representing the specified object  
valueOf()      返回数学对象的原始值

**4.Boolean**  
属性:  
constructor 所建立对象的函数参考  
prototype    能够为对象加入的属性和方法

方法:  
toSource()    显示对象的源代码  
toString()    将布尔值转换为字符串,并且返回结果  
valueOf()    返回布尔对象的原始值

**5.String**  
属性:  
constructor 所建立对象的函数参考  
prototype    能够为对象加入的属性和方法  
length      返回字符串的字符长度

方法(20):  
anchor("name")用来把字符串转换为HTML锚点标记内(<A NAME=>)  
big()      把字符串中的文本变成大字体(<BIG>)  
blink()     把字符串中的文本变成闪烁字体(<BLINK>)  
bold()      把字符串中的文本变成黑字体(<B>)  
fixed()     把字符串中的文本变成固定间距字体,即电报形式(<TT>)  
fontcolor(color)设置字符串中文本的颜色(<FONT COLOR=>)  
Fontsize(size) 把字符串中的文本变成指定大小(<FONTSIZE=>)  
italics()    把字符串中的文本变成斜字体(<I>)  
Link(url)用来把字符串转换-HTML链接标记中(<A HREF=>)  
small()     把字符串中的文本变成小字体(<SMALL>)  
strike()     把字符串中的文本变成划掉字体(<STRIKE>)  
sub()      把字符串中的文本变成下标(subscript)字体((SUB>)  
sup()      把字符串中的文本变成上标(superscript)字体(<SUP>)

charAt(index) 返回指定索引处的字符  
charCodeAt(index)返回一个整数,该整数表示String对象中指定位置处的字符的Unicode编码  
concat(string2)连接两条或多条字符串  
fromCharCode(num1, num2, …, numN)获取指定的Unicode值并返回字符串  
indexOf(searchString, startIndex) 返回字符串中第一个出现指定字符串的位置  
lastlndexOf(searchString, startIndex) 返回字符串中最后一个出现指定字符串的位置  
match(regex) 在字符串中查找指定值  
replace(regex, newString)将字符串中的某些字符替换成其它字符  
search(regex) 针对某执行值对字符串进行查找  
slice(startIndex, endIndex)将部分字符抽出并在新的字符串中返回剩余部分  
split(delimiter)将字符串分配为数组  
substr(startIndex, length) 从startIndex取,取length个字符  
substring(startIndex, endIndex) 从startIndex和endIndex之间的字符,不包括endIndex

toLowerCase() 把字符串中的文本变成小写  
toUpperCase() 把字符串中的文本变成大写  
toSource()    显示对象的源代码  
valueOf()    返回字符串对象的原始值

**6.Array**  
属性:  
constructor 所建立对象的函数参考  
prototype    能够为对象加入的属性和方法  
index      For an array created by a regular expression match, the zero-based index of the match in the string.  
input      For an array created by a regular expression match, reflects the original string against which the regular expression was matched.  
length      获取数组元素的个数,即最大下标加1

方法(13):  
concat(array1,arrayn)将两个或两个以上的数组值连接起来,合并后返回结果  
join(string) 将数组中元素合并为字符串,string为分隔符.如省略参数则直接合并,不再分隔  
pop()      移除数组中的最后一个元素并返回该元素  
push(value) 在数组的末尾加上一个或多个元素,并且返回新的数组长度值  
reverse()    颠倒数组中元素的顺序,反向排列  
shift()     移除数组中的第一个元素并返回该元素  
slice(start, deleteCount, [item1[, item2[,...[,itemN]]]]) 返从一个数组中移除一个或多个元素,如果必要,在所移除元素的位置上插入新元素,返回所移除的元素  
sort(compare Function) 在未指定排序号的情况下,按照元素的字母顺序排列,如果不是字符串类型则转换成字符串再排序,返回排序后的数组  
splice()     为数组删除并添加新的元素  
toSource()    显示对象的源代码  
toString()    将数组所有元素返回一个字符串,其间用逗号分隔  
unshift(value)为数组的开始部分加上一个或多个元素,并且返回该数组的新长度  
valueOf()    返回数组对象的原始值

**7.RegExp**  
属性:  
$1, ..., $9 Parenthesized substring matches, if any.  
$\_        See input.  
$\*        See multiline.  
$&        See lastMatch.  
$+        See lastParen.  
$`        See leftContext.  
$'        See rightContext.  
global      Whether or not to test the regular expression against all possible matches in a string, or only against the first.  
ignoreCase    Whether or not to ignore case while attempting a match in a string.  
input      The string against which a regular expression is matched.  
lastIndex    The index at which to start the next match.  
lastMatch    The last matched characters.  
lastParen    The last parenthesized substring match, if any.  
leftContext The substring preceding the most recent match.  
multiline    Whether or not to search in strings across multiple lines.  
rightContext The substring following the most recent match.  
source      The text of the pattern.

方法:  
compile     Compiles a regular expression object.  
exec       Executes a search for a match in its string parameter.  
test       Tests for a match in its string parameter

**8.Function**  
属性:  
arguments    An array corresponding to the arguments passed to a function.  
arity      Indicates the number of arguments expected by the function.  
caller      Specifies which function called the current function.  
prototype    Allows the addition of properties to a Function object.

方法:  
toString     Returns a string representing the specified object.

**9.Object**  
属性:  
constructor Specifies the function that creates an object's prototype.  
prototype    Allows the addition of properties to all objects.

方法:  
eval       Evaluates a string of JavaScript code in the context of the specified object.  
toString     Returns a string representing the specified object.  
unwatch     Removes a watchpoint from a 属性源 the object.  
valueOf     Returns the primitive value of the specified object.  
watch      Adds a watchpoint to a 属性源 the object.

**10.全局**  
属性:  
Infinity     指定一个正负无穷大的数值  
NaN       指定一个 “非数字” 值  
undefined    指定一个未被赋值的变量

方法:  
decodeURI()       为加密的URI进行解码  
decodeURIComponent() 为加密的URI组件解码  
encodeURI()       将字符串加密为URI  
encodeURIComponent() 将字符串加密为URI组件  
escape(string)      加密一个字符串  
()        使用escape()对一个字符串进行解码  
eval\_r(string)       判断一个字符串并将其以脚本代码的形式执行  
isFinite(number)     检测一个值是否为一个有限数字,返回True或False  
isNaN(string)      检测一个值是否不是一个有限数字  
Number()         将一个对象的值转换为一个数字  
parseFloat(string)    将一个字符串解析为一个浮点数字  
parseInt(string)     将一个字符串解析为一个整数,不是四舍五入操作,而是切尾  
String(object)      将一个对象值转换为一个字符串  
number(object)

**11.事件**  
属性:  
a.窗口事件,只在body和frameset元素中才有效  
onload      页面或图片加载完成时  
onunload     用户离开页面时

b.表单元素事件,在表单元素中才有效  
onchange     框内容改变时  
onsubmit     点击提交按钮时  
onreset     重新点击鼠标按键时  
onselect     文本被选择时  
onblur      元素失去焦点时  
onfocus     当元素获取焦点时

c.键盘事件,在base,bdo,br,frame,frameset,head,html,iframe,meta,param,script,style,title元素里都无效  
onkeydown    按下键盘按键时  
onkeypress    按下或按住键盘按键时  
onkeyup     放开键盘按键时

d.在base,bdo,br,frame,frameset,head,html,iframe,meta,param,script,style,title元素里都无效  
onclick     鼠标点击一个对象时  
ondblclick    鼠标双击一个对象时  
onmousedown 鼠标被按下时  
onmousemove 鼠标被移动时  
onmouseout    鼠标离开元素时  
onmouseover 鼠标经过元素时  
onmouseup    释放鼠标按键时

e.其他  
onresize     当窗口或框架被重新定义尺寸时  
onabort     图片下载被打断时  
onerror     当加载文档或图片时发生错误时

自定义对象:有初始化对象和定义构造函数的对象两种方法  
a：初始化对象  
例如： 对象={属性1:值1;属性2:值2;......属性n:值n} 注意:每个属性/值对之间用分号隔开；

b： 定义构造函数的对象  
例如:  
function 函数名(属性1, 属性2,......属性N){  
this.属性1=属性值1;  
this.属性2=属性值2;  
this.属性n=属性值n;

this.方法名1=函数名1;  
this.方法名2=函数名2;  
}