toLocaleString 方法返回一个 String 对象，这个对象中包含了用当前区域设置的默认格式表示的日期。  
  
对于公元 1601 和 1999 之间的时间，日期格式要按照用户的“控制面板”中“区域设置”来确定。  
F对于此区间外的其他时间，使用 toString 方法的默认格式。  
例如，同样是 3 月 21 日，在美国，toLocaleString 可能会返回 "03/21/08 01:02:03"，而在欧洲，返回值则可能是 "21/03/08 01:02:03"，因为欧洲的惯例是将日期放在月份前面。  
  
toLocaleString 只是用来显示结果给用户；最好不要在脚本中用来做基本计算，因为返回的结果是随机器不同而不同的。  
  
示例  
下面这个例子说明了 toLocaleString 方法的用法。  
  
var d, s; // 声明变量。   
d = new Date(); // 创建 Date 对象。   
s = "Current setting is ";   
s += d.toLocaleString(); // 转换为当前区域。   
return(s); // 返回转换的日期。   
}  
而toString()方法 在很多地方都被覆盖过 比如Integer有toString方法，BufferedString也有此方法 具体有什么作用是依据不同的类来看了

caller caller返回一个函数的引用，这个函数调用了当前的函数。 使用这个属性要注意: 1 这个属性只有当函数在执行时才有用 2 如果在javascript程序中，函数是由顶层调用的，则返回null functionName.caller: functionName是当前正在执行的函数。 复制代码 代码如下: var a = function() { alert(a.caller); } var b = function() { a(); } b(); 上面的代码中，b调用了a，那么a.caller返回的是b的引用，结果如下: 如果直接调用a(即a在任何函数中被调用，也就是顶层调用),返回null: 复制代码 代码如下: var a = function() { alert(a.caller); } var b = function() { a(); } //b(); a(); 输出结果:

callee callee放回正在执行的函数本身的引用，它是arguments的一个属性 使用callee时要注意: 1 这个属性只有在函数执行时才有效 2 它有一个length属性，可以用来获得形参的个数，因此可以用来比较形参和实参个数是否一致，即比较arguments.length是否等于arguments.callee.length 3 它可以用来递归匿名函数。 复制代码 代码如下: var a = function() { alert(arguments.callee); } var b = function() { a(); } b(); a在b中被调用，但是它返回了a本身的引用

强制转换

//3中强制转换，Number(value); String(value); Boolean(value)   
 var s=Number("123");   //强制转换为Number类型  
 alert(s+" :类型为"+(typeof s));

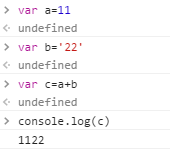
 var s1=String(12);  //强制转换为String类型  
 alert(s1+" :类型为"+(typeof s1));

 var s2=Boolean(123); //返回true。强制转换为Boolean类型  
 s2=Boolean(null);  //返回false  
 s2=Boolean(undefined);//返回false  
 s2=Boolean("null");//返回true  
 alert(s2+" :类型为"+(typeof s2));

隐式类型转换

如果把通过函数或方法调用，明确的将某种类型转换成另一种类型称为显示转换 ，相反则称为隐式类型转换

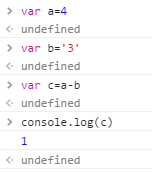
1，+运算

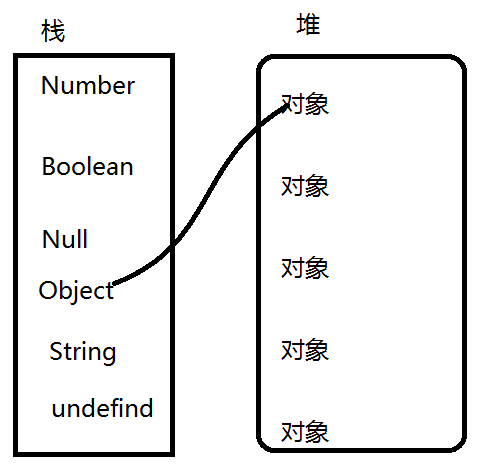


这里引擎将会先把a变成字符串"11"再与b进行连接，变成了"1122"。当运算符“+”两边一个是数字类型，一个是字符串类型时，js引擎规定进行字符串连接运算而非算术加运算。利用运算符“+”这个特性，可以很方便的将Number转换成String。

**2，“-”运算符**

“-”可以是一元运算符（取负），也可以是二元（减法运算）的。



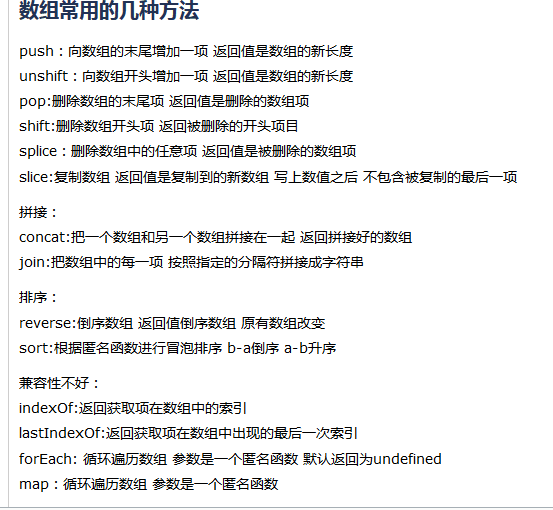


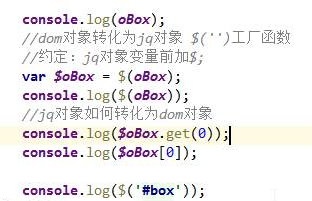
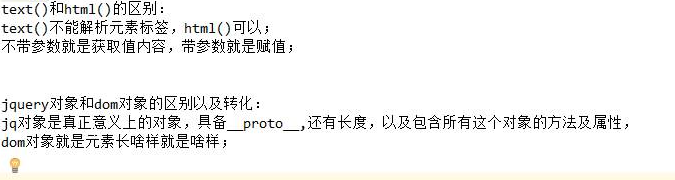
、

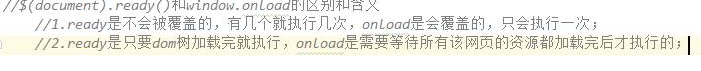
Em和Rem的区别

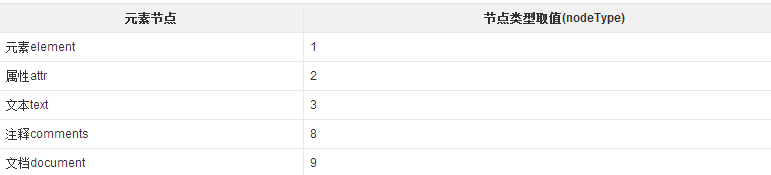
px是你屏幕设备物理上能显示出的最小的一个点，这个点不是固定宽度的，不同设备上点的[长宽](https://www.baidu.com/s?wd=%E9%95%BF%E5%AE%BD&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3nhNbmHR4nWmsrH6YuHuW0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPHR1rHb3PWcvPW6dn10YrjDz)、比例有可能会不同。假设：你现在用的显示器上1px宽=1毫米，但我用的显示器1px宽=两毫米，那么你定义一个div宽度为100px，你显示器上看这个div是10厘米，我显示器上看是20厘米。另外一个px点的[长宽](https://www.baidu.com/s?wd=%E9%95%BF%E5%AE%BD&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3nhNbmHR4nWmsrH6YuHuW0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPHR1rHb3PWcvPW6dn10YrjDz)不一定是1:1的正方形，有的设备上[长宽](https://www.baidu.com/s?wd=%E9%95%BF%E5%AE%BD&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3nhNbmHR4nWmsrH6YuHuW0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPHR1rHb3PWcvPW6dn10YrjDz)比是不一样的。  
  
em尺寸：所有现代浏览器下默认字体尺寸是16px，这时1em=16px。然后你人为的把body里面定义[font-size](https://www.baidu.com/s?wd=font-size&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3nhNbmHR4nWmsrH6YuHuW0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPHR1rHb3PWcvPW6dn10YrjDz):12px;（把浏览器默认16px改小了），此刻1em=12px，如果0.8em实际等于12px\*0.8；em的用处是你要整个网页字体统一变大变小你只要改body里面[font-size](https://www.baidu.com/s?wd=font-size&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3nhNbmHR4nWmsrH6YuHuW0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPHR1rHb3PWcvPW6dn10YrjDz)的值就行了。  
  
另外：em会继承父元素的字体大小，比如：  
body{[font-size](https://www.baidu.com/s?wd=font-size&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3nhNbmHR4nWmsrH6YuHuW0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPHR1rHb3PWcvPW6dn10YrjDz): 16px;}  
p{font-size:0.75em;}  
span{font-size:2em;}  
  
<html>  
我大小为16px;  
<p>  
 段落文字大小为12px(16\*0.75);  
 <span>  
 我大小是2em，即24px，这里是相对父级字号\*2的，而不是相对body里面的16px  
 </span>  
</p>  
</html>  
  
Rem，上面你看到了，em相对父级，嵌套一多了算字体到底多大就很操蛋，所以有了Rem(浏览器支持还不是很理想)，他只相对html或body的字体尺寸(默认还是16px，除非你自己用font-size定义为其他)，没有了继承父级尺寸这个关系。

C:\Users\Administrator\AppData\Roaming\Tencent\Users\2241368332\QQ\WinTemp\RichOle\LQ$TF9%XD(2[}65]]%C%J~J.png









json字符串转化为json对象的方法(三种):

**可以使用toJSONString()或者全局方法JSON.stringify()将JSON对象转化为JSON字符串**

**例如：**

var last=obj.toJSONString(); //将JSON对象转化为JSON字符

**或者**

var last=JSON.stringify(obj); //将JSON对象转化为JSON字符

JSON字符串:  
var str1 = '{ "name": "cxh", "sex": "man" }';   
JSON对象:  
var str2 = { "name": "cxh", "sex": "man" };

**JSON字符串转换为JSON对象**

要使用上面的str1，必须使用下面的方法先转化为JSON对象：

//由JSON字符串转换为JSON对象

var obj = eval('(' + str + ')');

**或者**

var obj = str.parseJSON(); //由JSON字符串转换为JSON对象

**或者**

var obj = JSON.parse(str); //由JSON字符串转换为JSON对象

然后，就可以这样读取：

Alert(obj.name);

Alert(obj.sex);

Jsonp的跨域方式  jsonp xml html text

page 当前浏览器的包括滚动条    client当前可视宽度     screen

阻止事件冒泡

[**jQuery**](http://lib.csdn.net/base/jquery) 提供了两种方式来阻止事件冒泡。

方式一：event.stopPropagation();

        $("#div1").mousedown(function(event){  
            event.stopPropagation();  
        });

方式二：return false;

        $("#div1").mousedown(function(event){  
            return false;  
        });

return false 不仅阻止了事件往上冒泡，而且阻止了事件本身。event.stopPropagation() 则只阻止事件往上冒泡，不阻止事件本身。