window 对象的属性

[来源 阮一峰](https://github.com/wangdoc/javascript-tutorial) 浏览 374[扫码 分享](https://www.bookstack.cn/read/javascript-tutorial/spilt.2.docs-bom-window.md)2020-06-28 21:58:55

* [window 对象的属性](https://www.bookstack.cn/read/javascript-tutorial/spilt.2.docs-bom-window.md#9ecn1m)
  + [window.name](https://www.bookstack.cn/read/javascript-tutorial/spilt.2.docs-bom-window.md#bcwudz)
  + [window.closed，window.opener](https://www.bookstack.cn/read/javascript-tutorial/spilt.2.docs-bom-window.md#42ip32)
  + [window.self，window.window](https://www.bookstack.cn/read/javascript-tutorial/spilt.2.docs-bom-window.md#e5f896)
  + [window.frames，window.length](https://www.bookstack.cn/read/javascript-tutorial/spilt.2.docs-bom-window.md#925wti)
  + [window.frameElement](https://www.bookstack.cn/read/javascript-tutorial/spilt.2.docs-bom-window.md#99risx)
  + [window.top，window.parent](https://www.bookstack.cn/read/javascript-tutorial/spilt.2.docs-bom-window.md#77iqu7)
  + [window.status](https://www.bookstack.cn/read/javascript-tutorial/spilt.2.docs-bom-window.md#7n2gwj)
  + [window.devicePixelRatio](https://www.bookstack.cn/read/javascript-tutorial/spilt.2.docs-bom-window.md#8o2dps)
  + [位置大小属性](https://www.bookstack.cn/read/javascript-tutorial/spilt.2.docs-bom-window.md#67ldex)
  + [组件属性](https://www.bookstack.cn/read/javascript-tutorial/spilt.2.docs-bom-window.md#geljwn)
  + [全局对象属性](https://www.bookstack.cn/read/javascript-tutorial/spilt.2.docs-bom-window.md#7q3nfq)
  + [window.isSecureContext](https://www.bookstack.cn/read/javascript-tutorial/spilt.2.docs-bom-window.md#5bukfr)

**window 对象的属性**

**window.name**

window.name属性是一个字符串，表示当前浏览器窗口的名字。窗口不一定需要名字，这个属性主要配合超链接和表单的target属性使用。

1. window.name ='Hello World!';
2. console.log(window.name)
3. // "Hello World!"

该属性只能保存字符串，如果写入的值不是字符串，会自动转成字符串。各个浏览器对这个值的储存容量有所不同，但是一般来说，可以高达几MB。

只要浏览器窗口不关闭，这个属性是不会消失的。举例来说，访问a.com时，该页面的脚本设置了window.name，接下来在同一个窗口里面载入了b.com，新页面的脚本可以读到上一个网页设置的window.name。页面刷新也是这种情况。一旦浏览器窗口关闭后，该属性保存的值就会消失，因为这时窗口已经不存在了。

**window.closed，window.opener**

window.closed属性返回一个布尔值，表示窗口是否关闭。

1. window.closed // false

上面代码检查当前窗口是否关闭。这种检查意义不大，因为只要能运行代码，当前窗口肯定没有关闭。这个属性一般用来检查，使用脚本打开的新窗口是否关闭。

1. var popup = window.open();
2. if((popup !==null)&&!popup.closed){
3. // 窗口仍然打开着
4. }

window.opener属性表示打开当前窗口的父窗口。如果当前窗口没有父窗口（即直接在地址栏输入打开），则返回null。

1. window.open().opener === window // true

上面表达式会打开一个新窗口，然后返回true。

如果两个窗口之间不需要通信，建议将子窗口的opener属性显式设为null，这样可以减少一些安全隐患。

1. var newWin = window.open('example.html','newWindow','height=400,width=400');
2. newWin.opener =null;

上面代码中，子窗口的opener属性设为null，两个窗口之间就没办法再联系了。

通过opener属性，可以获得父窗口的全局属性和方法，但只限于两个窗口同源的情况（参见《同源限制》一章），且其中一个窗口由另一个打开。<a>元素添加rel="noopener"属性，可以防止新打开的窗口获取父窗口，减轻被恶意网站修改父窗口 URL 的风险。

1. <ahref="https://an.evil.site"target="\_blank"rel="noopener">
2. 恶意网站
3. </a>

**window.self，window.window**

window.self和window.window属性都指向窗口本身。这两个属性只读。

1. window.self === window // true
2. window.window === window // true

**window.frames，window.length**

window.frames属性返回一个类似数组的对象，成员为页面内所有框架窗口，包括frame元素和iframe元素。window.frames[0]表示页面中第一个框架窗口。

如果iframe元素设置了id或name属性，那么就可以用属性值，引用这个iframe窗口。比如<iframe name="myIFrame">可以用frames['myIFrame']或者frames.myIFrame来引用。

frames属性实际上是window对象的别名。

1. frames === window // true

因此，frames[0]也可以用window[0]表示。但是，从语义上看，frames更清晰，而且考虑到window还是全局对象，因此推荐表示多窗口时，总是使用frames[0]的写法。更多介绍请看下文的《多窗口操作》部分。

window.length属性返回当前网页包含的框架总数。如果当前网页不包含frame和iframe元素，那么window.length就返回0。

1. window.frames.length === window.length // true

上面代码表示，window.frames.length与window.length应该是相等的。

**window.frameElement**

window.frameElement属性主要用于当前窗口嵌在另一个网页的情况（嵌入<object>、<iframe>或<embed>元素），返回当前窗口所在的那个元素节点。如果当前窗口是顶层窗口，或者所嵌入的那个网页不是同源的，该属性返回null。

1. // HTML 代码如下
2. // <iframe src="about.html"></iframe>
3. // 下面的脚本在 about.html 里面
4. var frameEl = window.frameElement;
5. if(frameEl){
6. frameEl.src ='other.html';
7. }

上面代码中，frameEl变量就是<iframe>元素。

**window.top，window.parent**

window.top属性指向最顶层窗口，主要用于在框架窗口（frame）里面获取顶层窗口。

window.parent属性指向父窗口。如果当前窗口没有父窗口，window.parent指向自身。

1. if(window.parent !== window.top){
2. // 表明当前窗口嵌入不止一层
3. }

对于不包含框架的网页，这两个属性等同于window对象。

**window.status**

window.status属性用于读写浏览器状态栏的文本。但是，现在很多浏览器都不允许改写状态栏文本，所以使用这个方法不一定有效。

**window.devicePixelRatio**

window.devicePixelRatio属性返回一个数值，表示一个 CSS 像素的大小与一个物理像素的大小之间的比率。也就是说，它表示一个 CSS 像素由多少个物理像素组成。它可以用于判断用户的显示环境，如果这个比率较大，就表示用户正在使用高清屏幕，因此可以显示较大像素的图片。

**位置大小属性**

以下属性返回window对象的位置信息和大小信息。

**（1）window.screenX，window.screenY**

window.screenX和window.screenY属性，返回浏览器窗口左上角相对于当前屏幕左上角的水平距离和垂直距离（单位像素）。这两个属性只读。

**（2） window.innerHeight，window.innerWidth**

window.innerHeight和window.innerWidth属性，返回网页在当前窗口中可见部分的高度和宽度，即“视口”（viewport）的大小（单位像素）。这两个属性只读。

用户放大网页的时候（比如将网页从100%的大小放大为200%），这两个属性会变小。因为这时网页的像素大小不变（比如宽度还是960像素），只是每个像素占据的屏幕空间变大了，因为可见部分（视口）就变小了。

注意，这两个属性值包括滚动条的高度和宽度。

**（3）window.outerHeight，window.outerWidth**

window.outerHeight和window.outerWidth属性返回浏览器窗口的高度和宽度，包括浏览器菜单和边框（单位像素）。这两个属性只读。

**（4）window.scrollX，window.scrollY**

window.scrollX属性返回页面的水平滚动距离，window.scrollY属性返回页面的垂直滚动距离，单位都为像素。这两个属性只读。

注意，这两个属性的返回值不是整数，而是双精度浮点数。如果页面没有滚动，它们的值就是0。

举例来说，如果用户向下拉动了垂直滚动条75像素，那么window.scrollY就是75左右。用户水平向右拉动水平滚动条200像素，window.scrollX就是200左右。

1. if(window.scrollY <75){
2. window.scroll(0,75);
3. }

上面代码中，如果页面向下滚动的距离小于75像素，那么页面向下滚动75像素。

**（5）window.pageXOffset，window.pageYOffset**

window.pageXOffset属性和window.pageYOffset属性，是window.scrollX和window.scrollY别名。

**组件属性**

组件属性返回浏览器的组件对象。这样的属性有下面几个。

* window.locationbar：地址栏对象
* window.menubar：菜单栏对象
* window.scrollbars：窗口的滚动条对象
* window.toolbar：工具栏对象
* window.statusbar：状态栏对象
* window.personalbar：用户安装的个人工具栏对象

这些对象的visible属性是一个布尔值，表示这些组件是否可见。这些属性只读。

1. window.locationbar.visible
2. window.menubar.visible
3. window.scrollbars.visible
4. window.toolbar.visible
5. window.statusbar.visible
6. window.personalbar.visible

**全局对象属性**

全局对象属性指向一些浏览器原生的全局对象。

* window.document：指向document对象，详见《document 对象》一章。注意，这个属性有同源限制。只有来自同源的脚本才能读取这个属性。
* window.location：指向Location对象，用于获取当前窗口的 URL 信息。它等同于document.location属性，详见《Location 对象》一章。
* window.navigator：指向Navigator对象，用于获取环境信息，详见《Navigator 对象》一章。
* window.history：指向History对象，表示浏览器的浏览历史，详见《History 对象》一章。
* window.localStorage：指向本地储存的 localStorage 数据，详见《Storage 接口》一章。
* window.sessionStorage：指向本地储存的 sessionStorage 数据，详见《Storage 接口》一章。
* window.console：指向console对象，用于操作控制台，详见《console 对象》一章。
* window.screen：指向Screen对象，表示屏幕信息，详见《Screen 对象》一章。

**window.isSecureContext**

window.isSecureContext属性返回一个布尔值，表示当前窗口是否处在加密环境。如果是 HTTPS 协议，就是true，否则就是false。