目录

[document 2](#_Toc532490133)

[referrer 2](#_Toc532490134)

[cookie 2](#_Toc532490135)

[获取 2](#_Toc532490136)

[设置 2](#_Toc532490137)

[expires 3](#_Toc532490138)

[iframe 3](#_Toc532490139)

[window 3](#_Toc532490140)

[open 3](#_Toc532490141)

[devicePixelRatio 3](#_Toc532490142)

[decodeURIComponent/encode 3](#_Toc532490143)

[history 4](#_Toc532490144)

[location 4](#_Toc532490145)

[href 4](#_Toc532490146)

[Navigator 4](#_Toc532490147)

[userAgent 4](#_Toc532490148)

[内核 5](#_Toc532490149)

[ie 5](#_Toc532490150)

[wx 7](#_Toc532490151)

[mobile 7](#_Toc532490152)

# document

## referrer



## cookie

### 获取

document.cookie

"a=a; b=b" 注意只有键和值，两个cookie之间有个空格。

### 设置

document.cookie = "a=a; path=/" 不需要在前面加原来的cookie，浏览器会自己添加在后面。键值和属性间也要有空格(不加浏览器也能识别)。  
属性不同的情况下，键可以重复，相同的情况下重新设置值会覆盖。

### expires

默认情况下coolie是暂时存在的，他们存储的值只在浏览器会话期间存在，当用户推出浏览器后这些值也会丢失，如果想让cookie存在一段时间，就要为expires属性设置为未来的一个过期日期。现在已经被max-age属性所取代，max-age用秒来设置cookie的生存期。

默认是一个过期的时间。1969...

# iframe

详见html标签中的iframe

# window

## open

如果使用了 strWindowFeatures 参数，那么只需要列出新窗口中启用的特性，其它的特性（除了titlebar和close）将被禁用或移除。

## devicePixelRatio

## decodeURIComponent/encode

直接从js获取的url相关的字符串是encode的，因为url里有双引号的话直接转成字符串是出错的。所以获取query时一定要decode！

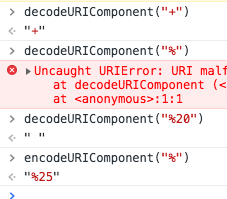
在小程序等native平台中，或者react-router等插件（应该不能在问题，因为是变量传递的）中，不需要手动从url读query的情况，一定要注意插件是否自动decode了。

### 规则

使用一到四个转义序列来表示字符串中的每个字符的UTF-8编码（只有由两个Unicode代理区字符组成的字符才用四个转义字符编码）。

转义除了字母、数字、(、)、.、!、~、\*、'、-和\_之外的所有字符。

注意：不会转义单引号，所以传字符串时最好用双引号





## encodeURI/decodeURI





# history

pushState不会使浏览器跳转，只是增加一条状态,当用户进入其他页面，再点返回才会到增加的页面。

# location

## href

可以通过这个加载新页面。

不能在http环境下加载https的页面

Mixed Content: The page at '<URL>' was loaded over HTTPS, but requested an insecure image '<URL>'. This content should also be served over HTTPS.

# Navigator

appCodeName: "Mozilla"

appName: "Netscape"

appVersion: "5.0 (Macintosh; Intel Mac OS X 10\_13\_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100 Safari/537.36"

userAgent: "Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_13\_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100 Safari/537.36"

appVersion是旧标准, 使用useAgent

## userAgent

浏览器的UA字串的标准格式：浏览器标识(操作系统标识;加密等级标识;浏览器语言)渲染引擎标识版本信息。

### 内核

trident: u.indexOf('Trident') > -1, //IE内核

presto: u.indexOf('Presto') > -1, //opera内核

webKit: u.indexOf('AppleWebKit') > -1, //苹果、谷歌内核

gecko: u.indexOf('Gecko') > -1 && u.indexOf('KHTML') == -1, //火狐内核

mobile: !!u.match(/AppleWebKit.\*Mobile.\*/), //是否为移动终端

ios: !!u.match(/\(i[^;]+;( U;)? CPU.+Mac OS X/), //ios终端

android: u.indexOf('Android') > -1 || u.indexOf('Linux') > -1, //android终端或uc浏览器

iPhone: u.indexOf('iPhone') > -1, //是否为iPhone或者QQHD浏览器

iPad: u.indexOf('iPad') > -1, //是否iPad

webApp: u.indexOf('Safari') == -1 //是否web应该程序，没有头部与底部

（上面这些是比较旧的判断方式）

### ie

User-Agent:Mozilla/5.0(compatible;MSIE9.0;WindowsNT6.1;Trident/5.0;)

#### 判断ie浏览器

window.navigator.appName == 'Microsoft Internet Explorer'

用Trident不行：很多多内核浏览器比如360,搜狗之类的包含Trident

#### ie版本

window.navigator.appName == 'Microsoft Internet Explorer'

parseInt(

window.navigator.appVersion.split(';')[1].replace(/[ ]/g, '').replace('MSIE','')

)

IE11的userAgent里是没有MSIE标志的，所以不支持ie11即以上。

#### 其他方案

function isIE() { //ie?

if (!!window.ActiveXObject || "ActiveXObject" in window)

return true;

else

return false;

}

#### 其他方案二

function IEVersion() {

var userAgent = navigator.userAgent; //取得浏览器的userAgent字符串

var isIE = userAgent.indexOf("compatible") > -1 && userAgent.indexOf("MSIE") > -1; //判断是否IE<11浏览器 ，这个不一定对，可能有国产浏览器

var isEdge = userAgent.indexOf("Edge") > -1 && !isIE; //判断是否IE的Edge浏览器

var isIE11 = userAgent.indexOf('Trident') > -1 && userAgent.indexOf("rv:11.0") > -1;

if(isIE) {

var reIE = new RegExp("MSIE (\\d+\\.\\d+);");

reIE.test(userAgent);

var fIEVersion = parseFloat(RegExp["$1"]);

if(fIEVersion == 7) {

return 7;

} else if(fIEVersion == 8) {

return 8;

} else if(fIEVersion == 9) {

return 9;

} else if(fIEVersion == 10) {

return 10;

} else {

return 6;//IE版本<=7

}

} else if(isEdge) {

return 'edge';//edge

} else if(isIE11) {

return 11; //IE11

}else{

return -1;//不是ie浏览器

}

}

### wx

/MicroMessenger/i

### mobile

ios

/(iPad|iPhone|iPod)\s+OS\s([\d\_.]+)/

android

/Android/

不是android和ios可以认为是pc端

# XMLHttpRequest