

## 0. 浏览器内核（了解）

---

浏览器内核主要指的是浏览器的渲染引擎，2013 年以前，代表有 Trident (IE) , Gecko (firefox) , Webkit (Safari, chrome 等) 以及 Presto (opera)。2013 年，谷歌开始研发 blink 引擎，chrome 28 以后开始使用，而 opera 则放弃了自主研发的 Presto 引擎，投入谷歌怀抱，和谷歌一起研发 blink 引擎，国内各种 chrome 系的浏览器（360、UC、QQ、2345，百度，傲游 等等）也纷纷放弃 webkit，投入 blink 的怀抱。

渲染引擎:version版本  
v8引擎 chrome浏览器。

移动端的浏览器内核主要说的是系统内置浏览器的内核。

目前移动设备浏览器上常用的内核有 Webkit, Blink, Trident, Gecko 等，其中 iPhone 和 iPad 等苹果 iOS 平台主要是 WebKit, Android 4.4 之前的 Android 系统浏览器内核是 WebKit, Android 4.4 系统浏览器切换到了 Chromium，内核是 Webkit 的分支 Blink, Windows Phone 8 系统浏览器内核是 Trident。

## 1. 什么是BOM?

---

BOM: Browser Object Model浏览器对象模型，它提供一套操作浏览器的API，特别是5大对象（window,navigator,location,history,screen），分别控制浏览器的各个部分，但BOM尚无正式标准。  
BOM的应用场景：JavaScript BOM主要用来操作浏览器,比如创建新页面、弹出对话框、设置定时器、检测浏览器、获取URL信息等。

提示:

DOM是一套操作HTML标签的API，已被标准化。document对象是window对象下的子对象。如：  
window.document.body  
BOM是一套操作浏览器的API，未标准化。

## 2. 常见的BOM对象有哪些?

---

常见的BOM对象

window: 代表整个浏览器窗口（window是BOM中的一个对象，并且是顶级的对象）

Navigator: 代表浏览器当前的信息，通过Navigator我们可以获取用户当前使用的是哪种浏览器

Location: 代表浏览器当前的地址信息，通过Location我们可以获取或者设置当前的地址信息

History: 代表浏览器的历史信息，通过History我们可以实现上一步/刷新/下一步操作（出于对用户的隐私考虑，我们只能拿到当前的浏览记录，不能拿到所有的历史记录）

Screen: 代表用户的屏幕信息

## 3. 各种API（重点）

---

### 3.1. window对象:

方法

window.alert('提示信息')

window.confirm("确认信息")

window.prompt("弹出输入框")

window.open("url地址", '打开的方式（可以是\_self或\_blank）', '新窗口的宽,高,位置')

window.close() 关闭当前的网页。注：存在兼容性问题  
window.moveTo() 移动当前窗口(了解)注：存在兼容性问题  
window.resizeTo() 调整当前窗口的尺寸

**window.setTimeout(函数, 时间) 只执行一次**

**window.setInterval(函数, 时间) 无限执行**

**window.clearTimeout/window.clearInterval(定时器名称) 清除定时器**

## Window 尺寸

有三种方法能够确定浏览器窗口的尺寸。

对于Internet Explorer、Chrome、Firefox、Opera 以及 Safari:

window.innerHeight - 浏览器窗口的内部高度(包括滚动条), 浏览器可视区域的高

window.innerWidth - 浏览器窗口的内部宽度(包括滚动条), 浏览器可视区域的宽

对于 Internet Explorer 8、7、6、5:

document.documentElement.clientHeight

document.documentElement.clientWidth

或者

document.body.clientHeight

document.body.clientWidth

## 事件监听

### scroll 滚动

resize 尺寸改变

## 3.2. Location对象:

window.location对象: 用于获得当前页面的地址 (URL), 并把浏览器重定向到新的页面。在编写时可不使用 window 这个前缀。

浏览器的地址栏, 放一个地址, 专业叫URL。

URL: Uniform Resource Locator, 统一资源定位器, 访问网络的地址 (IP, 域名) DNS域名解析系统

URI: Uniform Resource Identifier, 统一资源标识符; URI>URL URL是URI的子集

域名:是IP地址通过DNS解析得到域名。IP地址: xxx.xxx.xxx.xxx

如何查看一个域名下的IP?

1. 打开命令行工具: win+R打开运行框, 在运行框输入cmd(command命令)。
2. 在cmd中输入ping 域名, 如: ping [www.baidu.com](http://www.baidu.com)
3. 输入ipconfig 可以知道自己的IP

URL格式:

[协议名]://用户名:密码@服务器地址:服务器端口号/路径?查询字符串#片段ID

<https://www.baidu.com:80/abc.html?name=wb&age=20#top>

协议名: http, https(比http安全), ftp, file

关于网络协议: <https://zhuanlan.zhihu.com/p/32059190>

IP	域名
127.0.0.1	localhost

**属性:**

**location.href = 'url地址' 有历史记录**

**location.hash 返回#号后面的字符串，不包含散列，则返回空字符串。**

location.host 返回服务器名称和端口号

location.pathname 返回目录和文件名。 /project/test.html

**location.search 返回?号后面的所有值。**

location.port 返回URL中的指定的端口号，如URL中不包含端口号返回空字符串

location.protocol 返回页面使用的协议。 http:或https:

**方法:**

**location.reload();刷新页面**

location.assign(); 有历史记录

location.replace();没有历史记录

扩展 purl.js

下载: <https://github.com/shakakira/purl>

使用文档中文: <https://www.jianshu.com/p/11e74982fae2>

### 3.3. Navigator对象:

window.navigator 对象包含有关访问者浏览器的信息。在编写时可不使用 window 这个前缀。

navigator.platform: 操作系统类型;

**navigator.userAgent: 浏览器设定的User-Agent字符串。**

navigator.appCodeName: 浏览器代号;

navigator.appName: 浏览器名称;

navigator.appVersion: 浏览器版本;

navigator.language: 浏览器设置的语言;

navigator.systemLanguage: 浏览器系统语言; //ie能获取

**navigator.cookieEnabled: 浏览器是否启用了cookie;**

其中userAgent是最常用的属性，用来完成浏览器判断。

### 3.4. History对象:

window.history 对象包含浏览器的历史。为了保护用户隐私，对 JavaScript 访问该对象的方法做出了限制。

**history.back() - 加载历史列表中的前一个 URL。返回上一页。**

**history.forward() - 加载历史列表中的下一个 URL。返回下一页。**

**history.go(-1) - 负数时返回上一页，正数时返回下一页，**

历史记录的最大条数，最多只能后退最大条数，前进也一样。

### 3.5. Screen对象:

window.screen 对象包含有关用户屏幕的信息。

screen.availWidth 属性返回访问者屏幕的宽度，以像素计，减去界面特性，比如窗口任务栏。

screen.availHeight 属性返回访问者屏幕的高度，以像素计，减去界面特性，比如窗口任务栏。

参考文档:

<https://www.runoob.com/js/js-window.html>

<https://blog.csdn.net/lhjuejiang/article/details/79453407>

