

浏览器API - 学员预习

BOM 和 DOM

BOM

Navigator

Location

Screen

History

window

DOM - Events

事件级别

事件的捕获与冒泡

事件阻止函数

事件委托

Ajax & fetch

BOM 和 DOM

JavaScript 在浏览器环境下一般由三部分组成。

ECMAScript, DOM 和 BOM; (ie 加入了 ActiveXObject 类)

- ECMAScript 核心; 描述了JS 的语法和基本对象, 我们常说的ES6, 就是属于这个部分的内容;
 - Promise A+ 规范
- DOM 是文档对象模型, 处理网页的API, 来自于 W3C 的标准
 - <https://www.w3.org/TR/>
 - <https://zh.wikipedia.org/wiki/W3C%E6%8E%A8%E8%8D%90%E6%A0%87%E5%87%86>
 - **主要的区域是标签页中代码展示的部分。**
- BOM 是 Browser Object Model, 浏览器对象模型。BOM 顾名思义其实就是为了控制浏览器的行为而出现的接口。
 - **浏览器中, 除了代码展示的部分。**

题外话：W3C、ECMAScript 以及 MDN 这些关系是什么？

w3c是一个非标准化的组织，最重要的工作是发展 Web 规范

这些规范描述了 Web 的通信协议（比如 HTML 和 XHTML）和其他的构建模块。平时我们接触到的标准，比如：[超文本标记语言](#)、[HTML5规范](#)、[事件规范](#)，我们平时接触到DOM一级、二级规范都是该组织制定的。

标准制定后，就需要各方支持实现。得到了浏览器厂商的支持，所以会有一致的表现。

ECMAScript

ECMAScript也是一种标准，这个标准主要用于标准化JavaScript这种语言的，比如我们平时使用的es6、es7等都是该标准的产物。该标准由ECMA International进行，TC39 委员会进行监督。

在浏览器中，以V8、JSCore、SpiderMonkey等引擎进行解析。

所以，*node 为什么是 V8 引擎*。

- 语法（解析规则，关键词，流程控制，对象初始化，等等）
- 错误处理机制（[throw](#)、[try...catch](#)，以及创建用户定义[错误](#)类型的能力）
- 类型（布尔值，数字，字符串，函数，对象，等等）
- 全局对象。在浏览器环境中，这个全局对象就是 [window](#) 对象，但是 ECMAScript 只定义那些不特定于浏览器的 API（例如，[parseInt](#)、[parseFloat](#)、[decodeURI](#)、[encodeURIComponent](#)，等等）
- 基于原型的继承机制
- 内置对象和函数（[JSON](#)、[Math](#)、[Array.prototype](#) 方法、[对象内省](#)（自检、自我检查，[introspection](#)）方法，等等）
- [Strict mode](#)

MDN

MDN全称[Mozilla Developer Network](#)，它和前面的w3c和ECMAScript不太一样，这个组织不是为了标准化而诞生的。在MDN的官网上的左上角写着[MDN web docs](#)，很明显这是一个专为开发者服务的开发文档。

当然，w3c和ECMAScript也有对应的文档，但是平时开发的过程中，大家都比较习惯用MDN去查询资料，主要是MDN做的也太好了，有各种比较容易理解的使用说明和兼容性说明等。

MSDN

全称为microsoft developer network， 开发在IE浏览器上运行的程序需要参考的文档。鉴于平时IE支持到11后，很多属性都已经标准化了，所以平时用的比较少。

BOM

Navigator

属性	说明
appName	返回浏览器的名称 -- Netscape
appCodeName	返回浏览器的代码名
appVersion	返回浏览器的平台和版本信息
cookieEnabled	返回指明浏览器中是否启用 cookie 的布尔值
platform	返回运行浏览器的操作系统平台
userAgent	返回由客户机发送服务器的user-agent 头部的值
onLine	判断浏览器是否在线
connection	自动检测网络状况切换清晰度

Location

下边为Location的属性

属性	描述	栗子
hash	获取锚点，简单来说就是url的#后边	#detail?a=1
host	url的端口 + 端口	www.baidu.com:8080
hostname	主机路径	www.baidu.com
href	完整url	www.baidu.com?a=1
pathname	返回当前 URL 的路径部分	/index.html
port	端口	8080
protocol	协议	http或https，包括：号
search	协议	获取参数，简单来说就是url的?后边
origin	协议、主机名、端口	只有它是只读的，其余均可写。

下边为Location的方法

方法	描述
assign	加载新的文档
reload	重新刷新页面 相当于刷新按钮
replace	用新的文档替换当前文档 移动设备检测时的立刻跳转。

Screen

属性	说明
availHeight	返回屏幕的高度（不包括Windows任务栏）
availWidth	返回屏幕的宽度（不包括Windows任务栏）
colorDepth	返回目标设备或缓冲器上的调色板的比特深度
height	返回屏幕的总高度
pixelDepth	返回屏幕的颜色分辨率（每像素的位数）
width	返回屏幕的总宽度

浏览器的各种宽度、高度总结

screen.width/height

screen.availWidth/availHeight

浏览器的视口宽高 – window.innerWidth/Height

`window.innerWidth` 和 `window.innerHeight` 是只读属性，无默认值。

浏览器的窗口宽高 – window.outerWidth/Height

较于前者，一般情况下多了滚动条

Element.clientWidth、Element.clientHeight

`document.documentElement.clientWidth`

width + padding

Element.offsetWidth、Element.offsetHeight

width + padding + border

Element.scrollWidth、Element.scrollHeight

width + padding + 视口之外的宽度

History

属性	说明
back	返回上一页
forward	返回下一页
go	加载 history 列表中的某个具体页面
push	页面不刷新，不触发 onPopState 事件
replace	页面不刷新，不触发 onPopState 事件

window

方法	说明
prompt	
alert	
confirm	
close	
open	
setTimeout	
setInterval	
encodeURIComponent	
encodeURIComponent	
decodeURI	
decodeURIComponent	
devicePixelRatio	物理、逻辑像素比

DOM – Events

事件级别

DOM 的事件级别?

DOM 级别 0

DOM 级别 0 不是 W3C 规范。而仅仅是对在 Netscape Navigator 3.0 和 Microsoft Internet Explorer 3.0 中的等价功能性的一种定义。

```
document.querySelector('button').onclick = clickFn1
```

```
<div onclick="clickFn()">点击一下</div>
```

DOM 级别 1

DOM 级别 1 专注于 HTML 和 XML 文档模型。它含有文档导航和处理功能。

DOM 级别 2

DOM 级别 2 对 DOM 级别 1 添加了样式表对象模型，并定义了操作附于文档之上的样式信息的功能性。DOM 级别 2 同时还定义了一个事件模型（Events，规定了访问文档事件的 API），并提供了对 XML 命名空间的支持。

```
document.querySelector('button').addEventListener('click',clickBtn);
```

DOM 级别 3

DOM Level 3 规定了内容模型 (DTD 和 Schemas) 和文档验证。同时规定了文档加载和保存、文档查看、文档格式化和关键事件。

常用的事件有哪些？

鼠标事件：

- click：当用户点击某个对象时
- contextmenu：当用户点击鼠标右键打开上下文菜单时
- dblclick：当用户双击某个对象时
- mousedown：当鼠标按钮被按下
- mouseenter：当鼠标指针移动到元素上时
- mouseleave：当鼠标指针移出元素时
- mousemove：当鼠标被移动时
- mouseover：当鼠标移动到某元素上时
- mouseout：当鼠标从元素移开
- mouseup：当鼠标按键被松开

键盘事件：

- keydown：某个键盘按键被按下
- keypress：某个键盘按键被按下并松开
- keyup：某个键盘按键被松开

框架/对象事件：

- abort：图像的加载被中断
- beforeunload：在即将离开页面（刷新或关闭）时触发
- error：在加载文档或图像时发生错误
- hashchange：该事件在当前URL的锚部分发生修改时触发
- load：一张页面或一副图像加载完成
- pageshow：在用户访问页面时触发
- pagehide：在用户离开当前网页跳转到另外一个页面时触发
- resize：窗口或框架被重新调整大小
- scroll：当文档被滚动时
- unload：用户退出页面

表单事件：

- blur：元素失去焦点
- change：表单元素的内容发生改变时
- focus：元素获取焦点时
- focusin：元素即将获取焦点时
- focusout：元素即将失去焦点时
- input：获取用户输入时
- reset：表单重置时
- search：用户向搜索域输入文本时
- select：用户选取文本时
- submit：表单提交时

剪贴板事件：

- copy：用户拷贝元素内容时
- cut：在用户剪切元素内容时
- paste：在用户粘贴元素内容时

打印事件：

- afterprint：在页面已经开始打印，或者打印窗口已经关闭时触发
- beforeprint：在页面即将开始打印时触发

拖动事件：

- drag：元素正在拖动时触发
- dragend：用户完成元素的拖动时触发
- dragenter：拖动的元素进入放置目标时触发
- dragleave：在拖动元素离开放置目标时触发
- dragover：在拖动元素在放置目标上时触发
- dragstart：在用户开始拖动元素时触发
- drop：在拖动元素放置在目标区域时触发

事件的捕获与冒泡

addEventListener的第三个参数, 如果为true, 就是代表在捕获阶段执行。如果为false, 就是在冒泡阶段进行:

事件阻止函数

事件委托

Ajax & fetch