# HTML5 脚本编程

2016年6月25日 星期六 20:51

HTML5 规范定义了很多新 HTML 标记，并定义了很多 JavaScript API。目的是简化此前实现起
最终简化创建动态Web界面的工作。

跨文档消息传递（XDM）：在来自不同域的页面间 传递消息
XDM 的核心是 postMessage()方法，两个参数：一条消息和消息接收方来自哪个域的字符串
var iframeWindow = document.getElementById("myframe").contentWindow;
iframeWindow.postMessage("A secret", "http://www.wrox.com");

原生拖放
dragstart
drag
dragend
dragenter
dragover
dragleave/drop
dataTransfer：拖放操作时实现数据交换，url、text
addElement(element)、clearData(format)、setDragImage(element, x, y)、types
dropEffect：被拖动的元素能够执行哪种放置行为
effectAllowed：允许拖动元素的哪种dropEffect，none、move、copy、link、all
<img src="smile.gif" draggable="false" alt="让这个图像不可以拖动">

媒体元素 <audio>、<video>

```
<!-- 嵌入视频 -->
<video id="myVideo">
    <source src="conference.webm" type="video/webm; codecs='vp8, vorbis'">
    <source src="conference.ogv" type="video/ogg; codecs='theora, vorbis'">
    <source src="conference.mpg">
```

起来困难的任务，

```
    Video player not available.
</video>
<!-- 嵌入音频 -->
    <audio id="myAudio">
    <source src="song.ogg" type="audio/ogg">
    <source src="song.mp3" type="audio/mpeg">
    Audio player not available.
</audio>
```

## 历史状态管理

HTML5 通过更新 history 对象为管理历史状态提供了方便

通过 hashchange 事件，可以知道 URL 的参数什么时候发生了变化

通过状态管理 API ，能够在不加载新页面的情况下改变浏览器的 URL

pushState()、replaceState()

跨文档消息传递 API 能够让我们在不降低同源策略安全性的前提下，在来自不同域的文档间传

原生拖放功能让我们可以方便地指定某个元素可拖动，并在操作系统要放置时做出响应及自定义

新的媒体元素<audio>和<video>拥有自己的与音频和视频交互的 API

历史状态管理让我们不必卸载当前页面即可修改浏览器的历史状态栈，用户可以通过 后退、前
换，不必请求服务器

## XMLHttpRequest & AJAX

以异步方式从服务器取得更多信息，无须刷新页面即可从服务器取得数据

```
xhr.open("get", "example.php", false);
xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");//模仿表
xhr.send(null);
xhr.onreadystatechange = function(){
```

递消息

义

进 在页面状态间切

表单提交

响应：responseText、responseXML、status

}

### FormData

```
var data = new FormData(document.forms[0]);
data.append("name", "Nicholas");
```

### 同源策略

相同的域、相同的端口、相同的协议

CORS（跨源资源共享）

### 浏览器对 XML DOM 的支持

DOMParser：将 XML 解析为 DOM 文档

XMLSerializer：将 DOM 文档序列化为 XML 字符串

### XPath

XPath 是设计用来在 DOM 文档中查找节点的一种手段

### XSLT

利用 XPath 将文档从一种表现形式转换成另一种表现形式

### E4X

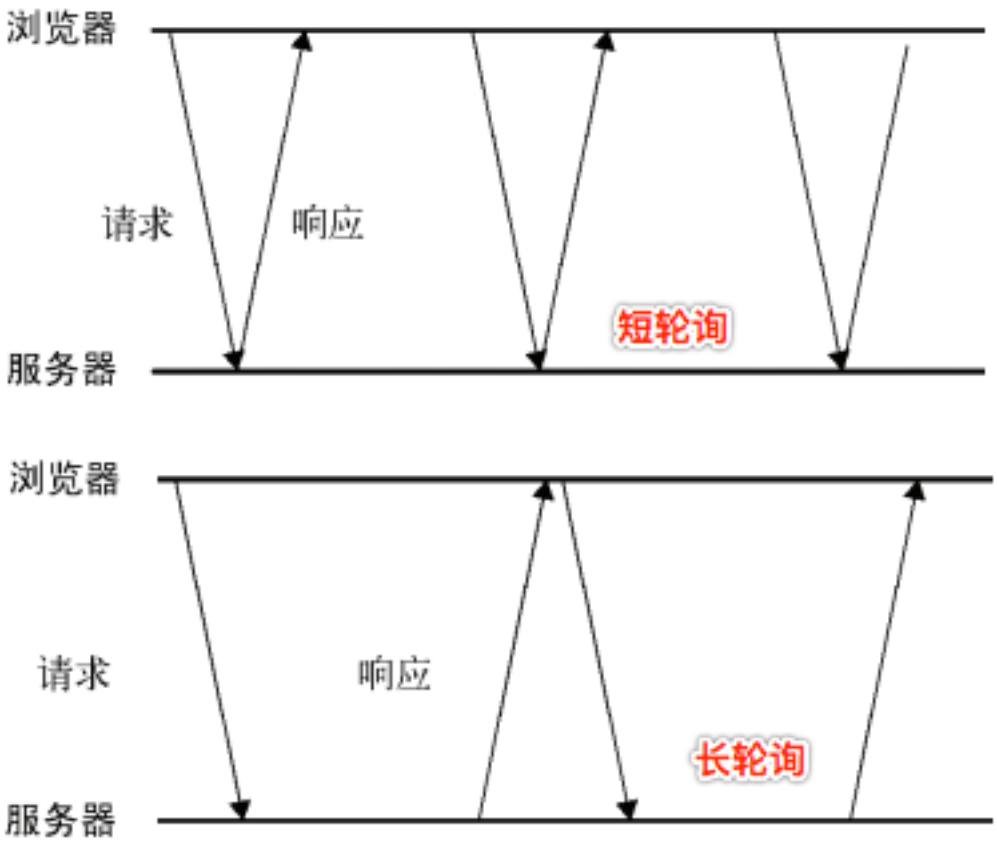
它只是 ECMAScript 语言的可选扩展。E4X 为处理 XML 定义了新的语法，也定义了特定于 X

ML 的对象

## Comet

一种服务器向页面推送数据的技术

## 长轮询




## HTTP流

在页面的整个生命周期内只使用一个 HTTP 连接，浏览器向服务器发送一个请求，而服务器保持
周期性地向浏览器发送数据

## SSE API （服务器发送事件）

用于创建到服务器的单向连接，服务器通过这个连接可以发送任意数量的数据，MIME（text/e
SSE 支持短轮询、长轮询和HTTP流，而且能在断开连接时自动确定何时重新连接

```
var source = new EventSource("myevents.php");
source.onmessage = function(event){
        var data = event.data;
        //处理数据
};
source.close();
```

持连接打开，然后

event-stream）

## Web Sockets

在一个单独的持久连接上提供全双工、双向通信，采用自定义协议，能够在客户端和服务器之间
数据，而不必担心 HTTP 那样字节级的开销，非常适合移动应用
先用HTTP建立连接，取得响应后HTTP-（升级协议）>Web Socket，故标准HTTP服务器无法实
http:// ---> ws://
https:// ---> wss://

## SSE与Web Sockets

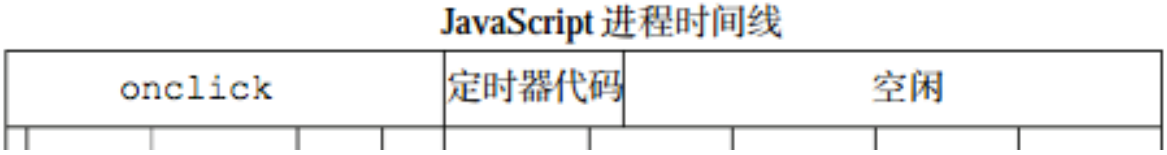
1、是否有自由度建立和维护 Web Sockets 服务器
2、到底需不需要双向通信（XHR+SSE=双向通信）

安全

1、要求以 SSL 连接来访问可以通过 XHR 请求的资源
2、要求每一次请求都要附带经过相应算法计算得到的验证码
3、要求发送 POST 而不是 GET 请求
4、检查来源 URL 以确定是否可信
5、基于 cookie 信息进行验证

## 定时器

定时器对队列工作方式：当特定时间过去后将代码插入，并不意味着对它立刻执行，但会尽快执
设定一个 150ms 后执行的定时器不代表到了 150ms 代码就立刻 执行，它表示代码会在 150m
中。如果在这个时间点上，队列中没有其他东西，那么这段代码就会被执行。

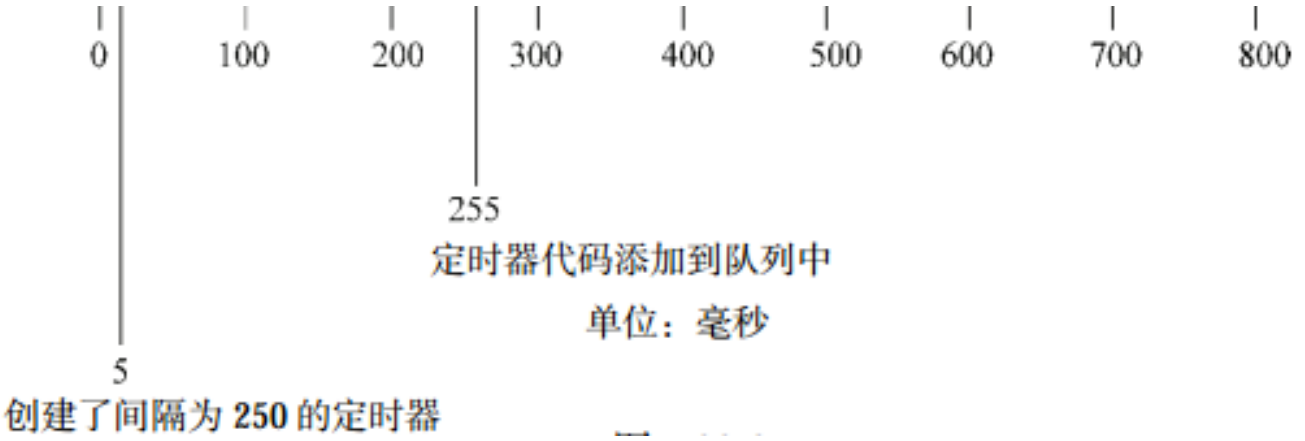


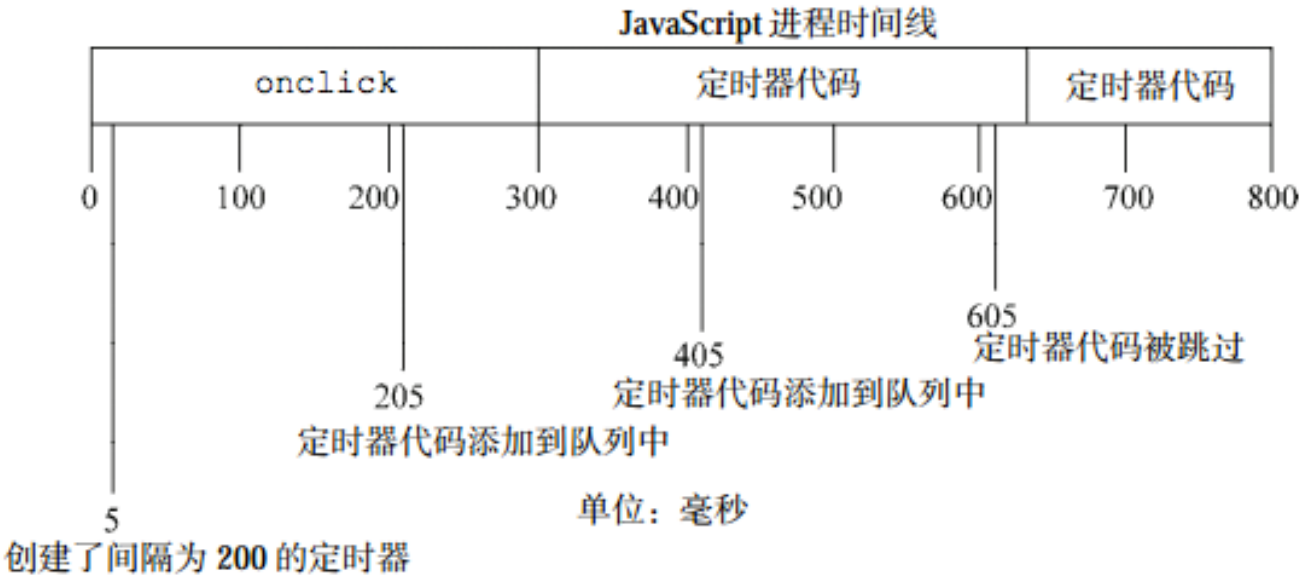
间发送非常少量的

实现

执行

ms 后被加入到队列




## HTML5 应用缓存（applicationCache）

专门为开发离线 Web 应用而设计的，从浏览器的缓存中分出来的一块缓存区

## Cookie

Cookie实际上是一小段的文本信息。客户端请求服务器，如果服务器需要记录该用户状态，就
户端浏览器颁发一个Cookie。客户端浏览器会把Cookie保存起来。当浏览器再请求该网站时，
址连同该Cookie一同提交给服务器。服务器检查该Cookie，以此来辨认用户状态。绑定在特定
状态信息，储存在 cookie 中的信息只能让批准的接受者访问。

domain、path、expires、secure标志都是服务器给浏览器的指示，都匹配时才发送cookie的

使用response向客
浏览器把请求的网
定域名下存储客户端

的键值

## Web存储机制

提供一种在 cookie 之外存储会话数据的途径

提供一种存储大量可以跨会话存在的数据的机制

### Storage

sessionStorage：会话中存储数据，浏览器关闭后立即删除

localStorage：用于跨会话持久化数据并遵循跨域安全策略

## IndexedDB

在浏览器中使用对象异步保存结构化数据的一种数据库（位于相同命名空间下的对象的集合）

同源策略，上限5MB

## 新兴API

1、动画 requestAnimationFrame()

2、Page Visibility API

3、Geolocation API

4、File API

5、Web Timing API（性能分析）

6、Web Workers （多线程）

Web Workers它所执行的JavaScript代码完全在另一个作用域中，不能访问DOM，多个标记页

## 性能

1、避免全局查找和不适用with语句

2、避免不必要的属性查找

3、优化循环

，替代WebSQL

页共享Worker

3、优化循环

4、原生方法较快、Switch 语句较快、位运算符较快

5、最小化语句数（多个变量声明用一个var）

6、使用数组和对象字面量

7、优化DOM交互（尽少访问操作遍历DOM）

三方框架

jQuery 函数式编程接口、是构建于 CSS 选择器之上操作DOM、链式调用

Backbone.js 是构建于 Underscore.js 基础之上的一个迷你 MVC 开源库，它针对单页应用进行
着应用状态变化方便地更新页面的任意部分。

动画和特效

script.aculo.us

moo.fx

Lightbox 任意页面上创建图像浮动层

行优化，让你能够随