# Canvas API

如果需要在ie中使用canvas，可以选择explorercanvas开源项目，需要先判断浏览器是否是ie，如果是则在页面中嵌入script标签来加载explorercanvas。 p25

判断浏览器是否支持canvas

try{

document.creatElement("canvas").getContext("2d");

//supported

}catch (e){

//not supported

}

p26

context.save();//这样即使进行了绘制和变换操作，也可以恢复到初始状态 p30

context.translate(70, 140); 通过这个操作，当平移发生时，我们提供的变换坐标会被加到结果坐标上，结果就是将要绘制的对角线移动到新的位置上 p30

context.restore();//**后续的**canvas操作就不会被刚才的平移操作影响 p30

# Geolocation API

位置信息是通过支持html5地理定位功能的底层设备（电脑、手机等）提供给浏览器。 P72

HTML5 Geolocation API不指定设备使用哪种底层技术来定位应用程序得用户。相反，它只是用于检测位置信息的API，而且通过该API检索到的数据只具有某种程度的精确度。并不能保证设备返回的实际位置是精确的。 P73

设备使用的数据来源：ip地址、gps、手机、用户自定义。 p74-75

Ip：任何地方都可用；不精确

Gps：精确；定位时间长耗电大，室内效果不好，需要额外硬件设备

Wifi：精确，室内可用，简单快捷；无线接入点少的地区效果不好

手机定位：优点通wifi；需要能够访问手机或modem的设备，基站少效果不好

自定义：可能不准确，特别是位置变更后

不调用html5Geolocation代码，就不会触发隐私保护机制 p77

定位你的地址原理都是一样的：先寻找你的WiFi接入点和你周围的WiFi接入点来确定位置，如果你周围没有WiFi而且你的电脑也是通过有线线路连接互联网的，那么就是通过IP确定位置了，这样可能就不如WiFi定位来的准。

<http://blog.csdn.net/huanghr_1/article/details/6114400>

参考：<http://www.ivershuo.com/2010/03/geolocation/>

# Communication API

跨文档消息通信可以确保**iframe**、**标签页**、**窗口间**安全地进行跨源通信。它把postMessag API定义为发送消息的标准方式。 p94

当某个消息到达时，通过**检查消息的来源**来决定是否对这条消息进行处理。 p94

父页面和部件通过把彼此的源加到可信源的白名单中，就能收到来自对方的信息。 P96

同源文档间通信时也推荐使用postMessage。 P96

源（origin）由规则（scheme）、主机（host）、端口（port）组成。 P96

跨源通信通过源来确定发送者，这就使得接收方可以忽略或者拒绝来自不可信任源（白名单之外）的消息。 P96

通过调用**目标页面**window对象中的postMessage()函数发送消息。 P98

# WebSockets API

基于同一底层TCP/IP连接在客户端和服务器之间的初始握手阶段，讲HTTP协议升级至WebSocket协议，WebSocke连接就建立完成了。 P116

# WebWorkers API

对多线程支持非常好，可以充分利用多核CPU带来的优势。 p158

Web Workers不能直接访问Web页面和DOM API（不能访问显示列表？）。 P158

Web workers初始化时会接受一个js文件url地址，其中包含了供Worker执行的代码。可以是相对或者绝对路径，这要同源（协议、主机、端口）即可（不能跨域？）。 p160

使用terminate函数或者启用它的页面终止webworkers，终止后不能被重新启用，但可以使用同样的url创建一个新的worker。P162