



# Internet Explorer 10 兼容性白皮书



# 前言

你一定听说过 HTML5，因为大家都在谈论它，那么它究竟好在哪儿？

HTML5 增加了大量提升用户体验的功能标准：canvas 元素可以让开发者在一个特定区域内绘制各种复杂图形，同时可以用 javascript 控制你所绘制的图形的动效，使得非 flash 的丰富的视觉效果成为可能，用户在移动端和 PC 端可以享受到一致的视觉体验；试想一个你曾经使用过的网络应用程序，一个在线图片编辑程序，你会用它做简单的图片处理，但绝对不会拿它代替 PhotoShop，因为它的交互性、响应率和功能性都远不及本地应用程序。HTML5 能够很好地解决这些问题。对于交互而言，它支持的 menu 元素能够模拟菜单栏、工具栏、列表等本地应用控件，而且很容易实现，drag-and-drop 功能可以模仿“将文件拖拽进垃圾箱”的操作；对于响应率而言，Programmable HTTP Cache and Serving 功能能使 web app 内容动态写入浏览器缓存，使 app 响应率接近于本地应用；Web Workers 允许 app 执行多任务，提高多用户并发操作的稳定性能；Offline Web Application 允许 app 在离线状态下存储用户数据、正常浏览。你还能通过浏览器获取用户的位置信息以及更多等待你去探索的功能。HTML5 是你的舞台，你的同行，像你一样出色的前端工程师们，都在摩拳擦掌，要在这个平台上创造出意想不到的有意思的东西，Chrome，FireFox，Safari...已经为你的表演做好了准备。现在，Internet Explorer 10 也申请加入你的阵营。

Internet Explorer 10 比较于 Internet Explorer 9 支持更多更酷的 HTML5 特性，如 applicationcache, css 3D transform, websockets 等，硬件加速图形功能也很出色，因此可以很流畅地支持 HTML5 网站。如果你的网站可以兼容 Internet Explorer 10，那么它不仅能够在其他高级浏览器上运行顺畅，更棒的是，对于 Windows 8 消费者预览版的用户，你的网站将有机会成为他们桌面上的一个 App，就看你的本事了！

兼容 Internet Explorer 10 的工作并不复杂，这篇指南将提供你完整的步骤以及遇到的各种问题的解决方法，同时还会告诉你如何针对 Internet Explorer 10 进行优化（如何为你的网站带来触摸体验，如何将它固定到开始屏幕上...）。如果你担心低版本浏览器对 HTML5 的支持问题，这篇指南将告诉你如何让 Internet Explorer 8 和之前的浏览器也能支持 HTML5。开始动手让你的网站变得更酷吧！

## 目录

<b>让你的网站兼容 Internet Explorer 10</b> .....	3
一、简单几步让你的网站与 HTML5 标准兼容.....	3
1. 确认你的网站使用的是标准模式.....	3
2. 实现 Web 标准下的文档模式.....	4
3. 判断网站未处于标准模式的原因.....	4
4. 在更新文档模式时，解决常见 Internet Explorer 问题.....	5
5. 其他一些导致网页不能正常显示的原因.....	6
二、解决 CSS 兼容性问题.....	7
1. 3D 转换.....	7
2. 过渡.....	8
3. 动画.....	10
三、兼容性测试工具.....	13
1. 开始使用 Compat Inspector.....	13
2. 验证.....	14
3. 调试.....	15
4. 鉴别 Frames.....	17
5. JavaScript 库.....	17
6. 最后的步骤.....	17
<b>让你的网站在 Internet Explorer 10 上变得更酷</b> .....	18
一、为你的网站带来触摸体验.....	18
1. 在不影响鼠标的情况下，优先处理触摸输入.....	18
2. 内置的多点触摸支持.....	20
3. 高级手势输入.....	21
4. 功能检测、fallback 和其他模型的支持.....	22
二、创建一个无插件的浏览体验.....	23
1. 无插件浏览变得越来越普遍.....	23
2. 越来越好的浏览体验.....	24
3. 功能检测，网站开发人员如何做得更好.....	25
<b>更多商业机会</b> .....	27
一、在 Windows 8 中固定网站.....	27
1. 固定网站比收藏夹更好用.....	27
2. 以网站为中心的固定磁贴.....	28
3. 可以在后台更新锁屏提醒通知.....	29
二、链接预览.....	32
1. 标题标签.....	32
2. 描述标签.....	32
3. 缩略图标签.....	32
<b>让你的网站在低版本浏览器上同样显示出色</b> .....	33
一、功能检测的重要性.....	34
二、手动功能检测.....	35
三、使用 Modernizr 进行功能检测.....	37
四、什么是 Polyfill.....	38
五、使用 Polyfill 模拟 HTML5 功能.....	39
六、使用 Polyfill 协助进行优雅降级.....	41
<b>微软技术支持</b> .....	43

# 让你的网站兼容 Internet Explorer 10

## 一、简单几步让你的网站与 HTML5 标准兼容

文档兼容性定义了一个浏览器将如何呈现你的网站。把自己的要求对浏览器告诉得越详细，用户的体验就会越好。在使用 HTML5 这样的 Web 标准时，一开始就要明确地声明 HTML5 的文档类型：

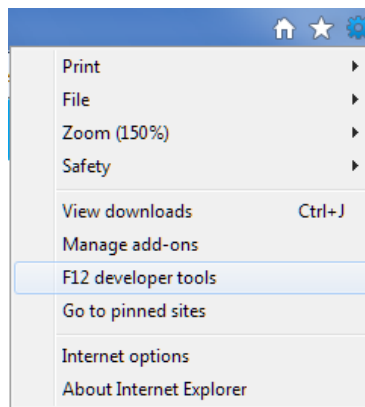
```
<!DOCTYPE html>
```

这个标记将触发在 Internet Explorer 9 和 Internet Explorer 10 中的标准模式，它在 Chrome 和 Firefox 中也工作得非常好。以下 4 个步骤可以使你的网站在各种浏览器和设备上呈现：

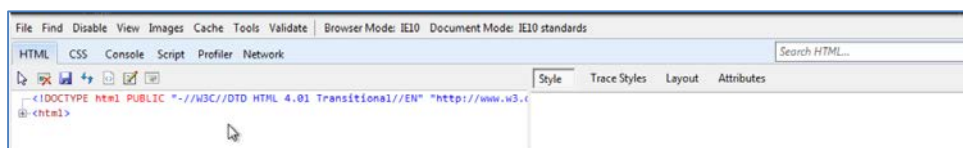
### 1. 确认你的网站使用的是标准模式

检查网站目前是否处于标准模式：

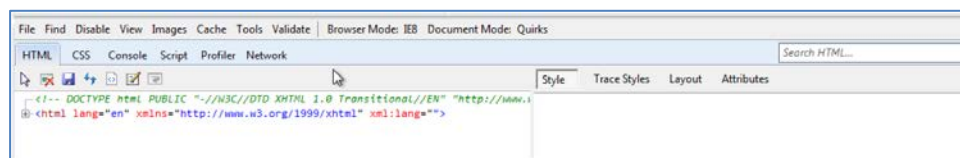
- 1) 在 Internet Explorer 10 平台预览版中打开网站。
  - **注意：**也可以按相同步骤更新 Internet Explorer 9 中的文档模式，这无需下载预览版。
- 2) 按 F12 键启动 IE 开发人员工具，或在 Tools (工具) 菜单中找到它，如下所示：



- **注意：**如果不习惯使用 IE F12 开发人员工具来调试网页，请阅读链接教程：  
<http://msdn.microsoft.com/zh-cn/library/gg589507%28v=VS.85%29.aspx>
- 3) 检查网站是否显示“Browser Mode: IE10”和“Document Mode: IE10 standards”，如下面的工具栏所示：



- 如果你的网站已经处于 Browser Mode: IE10 和 Document Mode: IE10 standards 状态，那么你已经完成了！**注意，如果你的网站的浏览器模式和文档模式不是这样，一个常见的情况是 Browser Mode = IE8 和 Document Mode = Quirks（浏览器模式=IE8，文档模式=Quirks），这表明网站是为旧版本 IE 设计的，可能不符合 Web 标准的要求。**



## 2. 实现 Web 标准下的文档模式

强制使用 IE10 标准模式来测试你的网站：

- 将以下标签 `<!DOCTYPE html>` 插入页面顶部
- 在浏览器中重新加载页面，使用 F12 开发人员工具再次检查浏览器模式和文档模式。如果没有显示 Browser Mode: IE10 和 Document Mode: IE10 standards，请继续执行以下步骤。

## 3. 判断网站未处于标准模式的原因

很多问题与支持旧版本 IE 有关。先要保证基于标准的代码在 IE9 和 IE10 中正常呈现。然后再保留用于支持旧版本 IE 的非标准代码。

- 我的页面不是“Browser Mode: IE10”（浏览器模式：IE10）
  - 可能的原因：**你的网站可能处于 Compatibility View（兼容性视图）中，被迫进入旧版浏览器模式以确保网站功能正常。
    - 解决办法：**检查网站是否在下面这个列表内 <http://ie9cvlist.ie.microsoft.com/ie9CompatViewList.xml>
- 我的网页不是“Document Mode =IE10”
  - 可能的原因：**你的网站的 doctype 无效或缺失
    - 解决办法：**检查是否使用了有效、格式正确的 doctype，如下所示：

```
<!DOCTYPE html>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

- **可能的原因**：文档模式受到 X-UA-Compatible 这个元标签的强行限制

- **解决办法**：在页面上检查是否有类似下面的代码

```
<metahttp-equiv="X-UA-Compatible" content="IE=Emulate IE7">
```

```
<metahttp-equiv="X-UA-Compatible" content="IE=8">
```

删除找到的代码，重新加载页面，继续测试。

## 4. 在更新文档模式时，解决常见 Internet Explorer 问题

很多问题与支持旧版本 IE 有关。先要保证基于标准的代码在 IE9 和 IE10 中正常呈现。然后再保留用于支持旧版本 IE 的非标准代码。

- **可能的原因**：条件注释 ( conditional comment ) 支持不同浏览器版本相关的功能

- **解决办法**：检查是否存在运行非标准代码的条件注释。这些通常用来支持旧版本 IE 的特定功能，以便让页面在旧版本下可以实现优雅降级。请检查是否存在类似下面的代码：

```
<!--[if IE 8]>
```

```
<p>Welcome to Internet Explorer 8.</p>
```

```
<![endif]-->
```

删除找到的代码，重新加载页面，继续测试。

- **可能的原因**：用户代理监听支持浏览器版本相关的功能

- **解决办法**：检查是否存在用户代理监听。用户代理监听通过浏览器模式中存在的用户代理字符串 ( user agent string ) 来定位浏览器。请检查是否存在类似下面的代码：

```
if(version = /MSIE (\d+\.\d+)/.exec(navigator.userAgent)) {
```

```
    isIE = true;
```

```
    browserVersion = parseFloat(version[1]);
```

```
}
```

用 Web 标准实现功能探测。

## 5. 其他一些导致网页不能正常显示的原因

- **可能的原因** :网站可能使用了浏览器已不再支持的特定功能。请尽可能使用 Web 标准。
  - **解决办法** :运行兼容性检查工具。
- **可能的原因** :网站可能使用了 Metro 下 Internet Explorer 10 不支持的第三方插件或 Flash、Quicktime 和 Silverlight。请尽可能使用 Web 标准。
  - **解决办法** :学习如何创建不使用插件的体验。完整的基础教程很快就将推出。
- **可能的原因** :网站可能加载了特定版本的浏览器 CSS 文件 :
  - **解决办法** :确保布局尽可能避免 CSS hack。

IE Compatibility Cookbook ( [http://msdn.microsoft.com/en-us/library/ie/hh801219\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/ie/hh801219(v=vs.85).aspx) ) 中提供了一个常见问题清单。如果你通过以上介绍的解决步骤还不能更新文档模式, 请参考 MSDN 的论坛。

## 二、解决 CSS 兼容性问题

如果你经常使用 CSS3 实现一些炫酷的效果，比如使用 transition 实现让一个 div 慢慢变宽的动效，你一定会为了兼容各个高级浏览器而这样写你的 CSS 代码：

```
#box {
    width: 200 px;
    height: 200 px;
    -moz-transition: all 2s ease 0s;
    -webkit-transition: all 2s ease 0s;
    -o-transition: all 2s ease 0s;
}
#box:hover {
    width: 800 px;
}
```

在 Internet Explorer 10 上你也可以使用 CSS3 实现这个效果了，只需要为 #box 添加一行代码：`-ms-transition: all 2s ease 0s;`

下面列举出几个典型的需要兼容 Internet Explorer 10 的 CSS3 特性：

### 1. 3D 转换

Internet Explorer 10 和 Windows 8 中使用 JavaScript 的 Metro 风格的应用引入了对 CSS3 3D 转换的支持。Internet Explorer 9 增加了对 CSS3 2D 转换的支持，使用 JavaScript 的 Metro 风格的应用也同样支持它们。转换支持包括变换、旋转以及 2D 和 3D 空间中的元素拉伸，这些转换无需插件。CSS3 3D 转换由 W3C 在 CSS 3D Transforms Module Level 3 规范中定义的，目前这个规范处于工作草案阶段。

为了实现 CSS 转换，需要使用 `-ms-transform` 属性定义一个或多个转换函数。例如，以下代码段是一个选择器，它将一个 2D 偏移函数和一个 3D 旋转函数同时应用于一个 div 元素。

```
div {
    ...
    -ms-transform: skew(-45deg, 15deg) rotate3d(0.7, 0.8, 0.4, 55deg);
}
```

这将得到以下图像：





关于 CSS3 3D 转换、转换函数以及如何实现它们的更多信息，请参阅：“3-D 转换” (<http://go.microsoft.com/fwlink/p/?LinkId=238295>)。

关于 CSS3 3D 转换的实际操作，请访问 IE Test Drive (<http://go.microsoft.com/fwlink/p/?LinkId=196941>) 站点上的“动手实验：3D 转换” (<http://go.microsoft.com/fwlink/p/?LinkId=227893>)。

## 2. 过渡

Internet Explorer 10 和使用 JavaScript 的 Metro 风格的应用支持 CSS3 过渡特效。通过过渡特效，可以实现简单的动画效果，在一小段时间内平滑地改变 CSS 的属性值。例如，在 10 秒钟内改变一个对象的大小和颜色。以前需要通过 CSS 和 JavaScript 的复杂组合操作实现的效果，现在只需几行 CSS 代码即可完成。CSS3 过渡是由 W3C 在 CSS Transitions Module Level3 规范中定义的，目前这个规范处于工作草案阶段。

关于 CSS3 过渡的实际操作演示，请参见 IE Test Drive (<http://go.microsoft.com/fwlink/p/?LinkId=196941>) 站点上的“动手实验：过渡” (<http://go.microsoft.com/fwlink/p/?LinkId=227781>)。以下是一个完整的简单 CSS 过渡的示例，关于这个示例的更全面的解释，以及所有过渡属性，请参见“过渡” (<http://go.microsoft.com/fwlink/p/?LinkId=238302>)。

```
<!DOCTYPE html>
<html lang="en-us">
<head>
  <style type="text/css">
    body {
      padding:10px;
      font:bold 20pt "Segoe UI";
    }
  </style>
  <div {
```

```

        width:250px;
        background-color:lime;
        padding:10px;
        opacity:1;
        -ms-transition:opacity 5s linear 1s;
    }
    div:active {
        opacity:0;
    }
</style>
</head>
<body>
    <div>
        Duis ac leo sit ametlectustristiquepulvinarnecrutrum dolor.
        Etiamsedipsumenim, vitae euismododio.Suspendisseu.
    </div>
</body>
</html>

```

在这个示例中，属性 `-ms-transform` 被设置为 “opacity 5s linear 1s”。借助于 `: active` 伪类，当单击（或触摸）并按住柠檬绿色的 `div`，它将慢慢消失。特别地，它的 `opacity` 属性在 1 秒之后，会产生平滑的过渡效果（使用一个线性计时函数），在 5 秒之内从完全不透明变为完全透明。请参见这个页面（<http://go.microsoft.com/fwlink/p/?LinkId=227845>）。

`-ms-transform` 属性是一个简写属性，这个声明也可以使用以下属性来替代：

```

-ms-transition-property:opacity;
-ms-transition-duration:5s;
-ms-transition-timing-function:linear;
-ms-transition-delay:1s;

```

关于 CSS 过渡的更多功能，请参见“过渡”（<http://go.microsoft.com/fwlink/p/?LinkId=238302>）。

### 3. 动画

Internet Explorer 10 和使用 JavaScript 的 Metro 风格的应用支持 CSS3 动画。利用 CSS3 动画，可以创建具有炫酷视觉效果的应用程序，实现平滑流畅的动画效果。CSS3 动画是由 W3C 在 CSS Animations Module Level 3 规范中定义的，目前这个规范处于工作草案阶段。

只需 3 个基本步骤即可创建动画：指定动画属性，创建关键帧，将动画应用于一个或多个元素。

动画类似于过渡，都是通过改变元素的位置、大小、颜色和透明度，并通过旋转、位伸、变换等操作来实现动画。像使用过渡效果一样，可以指定计时函数来控制动画的前进速度。但是，利用 CSS3 动画，我们还可以使用关键帧，它指定了动画的各个时间点的属性值。这样，我们不仅可以定义动画起始和结束的行为，而且还可以定义在开始和结束之间的行为。动画还具有迭代和反转方向的功能，还可以暂停和恢复播放。

关于 CSS3 动画的实际操作演示，请参见 IE Test Drive (<http://go.microsoft.com/fwlink/p/?LinkId=196941>) 站点上的“动手实验：动画”。(<http://go.microsoft.com/fwlink/p/?LinkId=228082>) 以下是一个简单的 CSS 动画的完整示例。关于这个示例的更全面的解释，以及所有动画属性，请参见“动画”(<http://go.microsoft.com/fwlink/p/?LinkId=238298>)。

```
<!DOCTYPE html>
<html lang="en-us">

<head>
  <style type="text/css">
    body {
      padding: 10px;
      font-family: "Segoe UI";
    }
    div {                                     /* 定义动画的 div */
      width: 250px;
      background-color: lime;
      padding: 10px;
      font-weight: bold;
      font-size: 20pt;
    }
    div:active {                             /* 定义动画参数 */
      -ms-animation-delay: 0s;
      -ms-animation-duration: 5s;
      -ms-animation-iteration-count: 2;
```

```

        -ms-animation-name: demo;
    }
    @-ms-keyframes demo { /* 定义动画的关键帧*/
    from {
        -ms-animation-timing-function: ease;
    }
    50% { /* 使用 ease 计时函数*/
        background-color: purple; /* 来移动和更改 div 的颜色*/
        -ms-animation-timing-function: ease-in;
        -ms-transform: translate(20px,30px);
    }
    to {
        background-color: blue; /* 使用 ease 计时函数将 div 更改为蓝色*/
    }
    }
    </style>
</head>

<body>

    <h1>CSS3 Animations Example</h1>
    <p>Click and hold to start the animation. Internet Explorer 10 or later
    required.</p>
    <div>
        Duis ac leo sit ametlectustristiquepulvinarnecrutrum dolor.
        Etiam sed ipsum enim, vitae euismod odio. Suspendisse eu.
    </div>
</body>
</html>

```

这段标记代码的示例定义了两个动画属性：background-color 和 -ms-transform。在动画的一个循环周期中，div 元素的背景颜色由柠檬色（初始颜色）变为紫色，然后又变为蓝色。div 元素还向右和向下分别移动 20 像素和 30 像素，然后还原。这个动画使用了 ease 计时函数从开始过渡到中间点（关键帧为 50%），然后使用 ease-in 计时函数从中间点过渡到结束。动

画重复了一次。观看这个动画 ( <http://go.microsoft.com/fwlink/p/?LinkId=228195> ) ( 要求 Internet Explorer 10 ) 。

正如您所期望的，CSS3 动画提供了无限的可能性，我们可以利用它来创建丰富、炫酷的动画。关于如何在 Internet Explorer 10 和使用 JavaScript 的 Metro 风格应用中创建 CSS 动画的概述，以及 CSS3 动画的实际使用示例，请参见“动画” ( <http://go.microsoft.com/fwlink/p/?LinkId=238298> ) 。

关于 CSS3 动画的操作演示，请访问 IE Test Drive ( <http://go.microsoft.com/fwlink/p/?LinkId=196941> ) 站点上的“动手实验：动画 ( <http://go.microsoft.com/fwlink/p/?LinkId=228082> ) ” 。

更多兼容 Internet Explorer 10 的 CSS3 特性的内容请查看 Windows Internet Explorer 10 Consumer Preview Guide for Developers CSS ( [http://msdn.microsoft.com/en-us/library/ie/hh673536\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/ie/hh673536(v=vs.85).aspx) )

### 三、兼容性测试工具

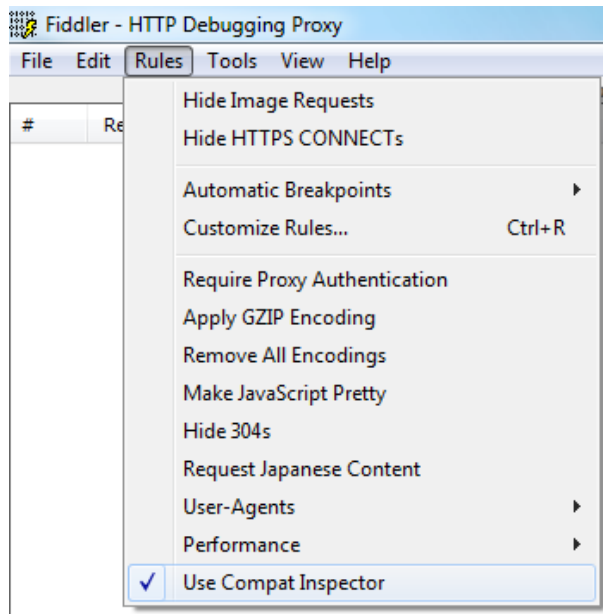
Compat Inspector 是一种基于 JavaScript 的测试工具，它在网站运行的时候分析网站。Compat Inspector 自动监视并报告已知会导致问题的交互模式。这让你可以迅速找出问题，而不需要先熟记一大堆文档。

#### 1. 开始使用 Compat Inspector

要运行 Compat Inspector，你需要将下面这段代码写到每一个要测试的页面中的所有 <script> 标签之前：

```
<script src="http://ie.microsoft.com/testdrive/HTML5/CompatInspector/inspector.js"></script>
```

你也可以使用 Fiddler 自动将 Compat Inspector 注入到要测试的页面中，只要将这个代码片段 (<http://ie.microsoft.com/testdrive/HTML5/CompatInspector/help/snippet.txt>) 加入到你的 Fiddler 脚本 (<http://www.fiddlertool.com/fiddler/dev/scriptsamples.asp>) 中就好了。之后你就可以在菜单栏的“Rules”菜单中启动和关闭 Compat Inspector 了，如下图所示：

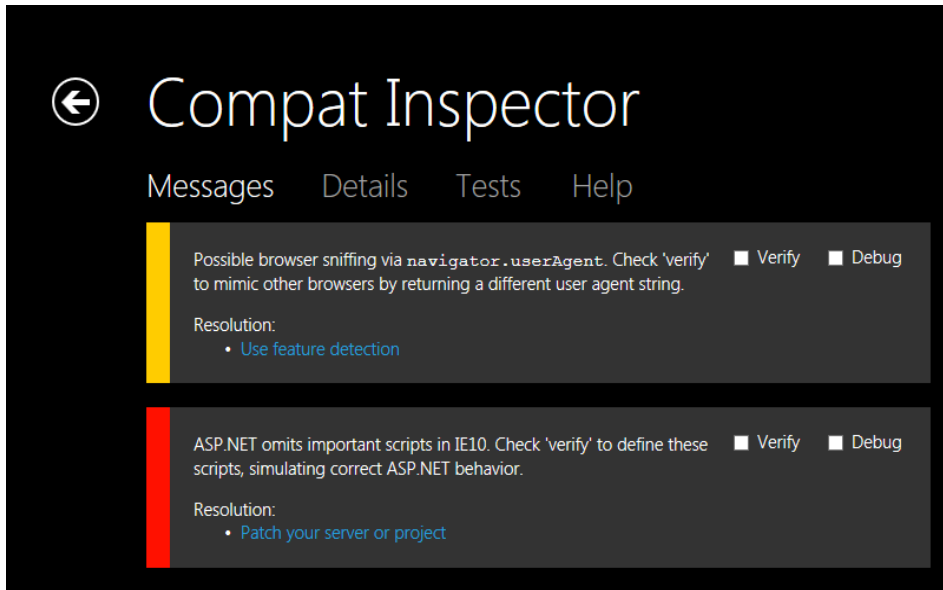


将上述的脚本添加好之后在 Internet Explorer 10 中打开你的页面，Compat Inspector 就会立即开始分析这个页面。需要注意的是，一些运行时问题只有在代码运行过程中才能够被捕捉到，因此需要你手动触发这些不正常的交互过程来进行测试。

Compat Inspector 会按照错误(红色)、警告(黄色)和提示信息(蓝色)分别统计检测到的问题的总和，统计结果显示在页面的右上角：



点击这个统计结果视图，你可以从展开的视图获得每个问题的详细信息和相关文档：



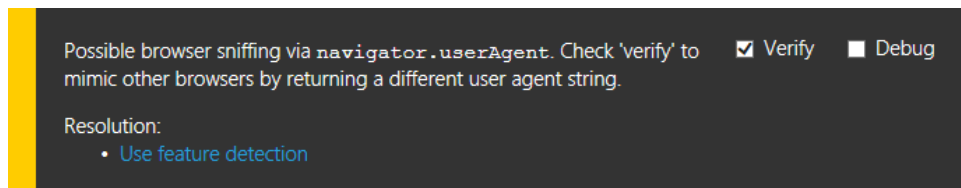
Compat Inspector 还为大多数的错误和警告提供“验证”和“调试”功能，以便对问题进行更精准的定位。

## 2. 验证

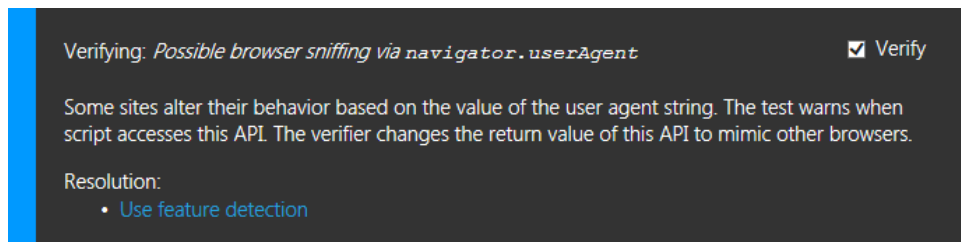
部分测试用例具有内置的“验证器”，可模拟遗留的行为(legacy behavior)。这让你在不修改任何代码的情况下就能够找出问题的根本原因。

**前提**：有错误或警告消息出现在 Compat Inspector 的 Details 视图中，并且这些消息包含“Verify”复选框。

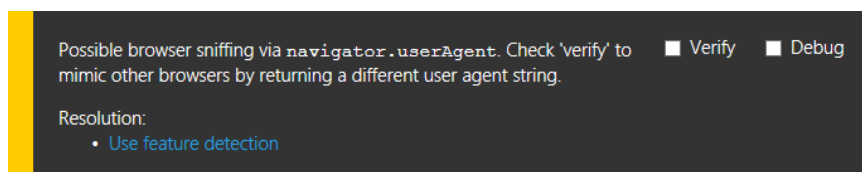
- 1) 勾选一个你想要验证的错误或警告的“Verify”复选框



- 2) 刷新页面（你会看到一个通知，提示你验证器已经被启用）



- 3) 再次触发这个错误或警告
- 4) 如果同样的问题依然存在
  - 如果仍然显示错误或警告消息，则使用其他消息重复步骤 1-4；否则 Compat Inspector 无法确认这个问题，请访问 Internet Explorer 10 Guide for Developers ( <http://msdn.microsoft.com/library/hh673549.aspx> ) 获取更多帮助信息。
- 5) 否则，你就已经找到了问题的根源所在
  - 取消所有“Verify”复选框的勾选



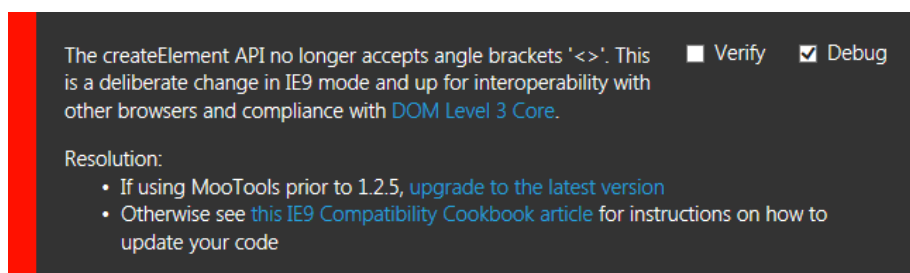
- 刷新页面
- 依照下面“调试”一节中叙述的步骤定位你代码中各条消息的来源
- 按照所显示的错误/警告消息中的说明解决该问题

### 3. 调试

Compat Inspector 还可以通过调试功能帮助你发现代码中的错误。

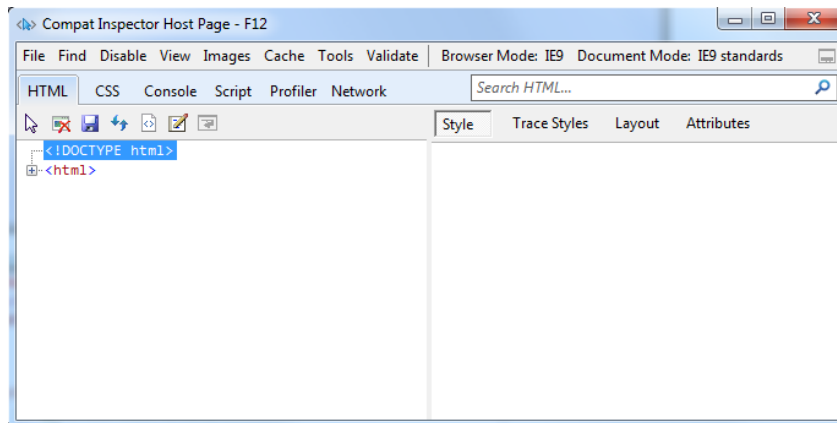
**前提**：有错误或警告消息出现在 Compat Inspector 的 Details 视图中，并且这些消息包含“Debug”复选框。

- 1) 勾选一个你想要调试的错误或警告的“Debug”复选框

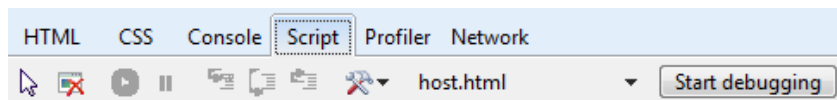




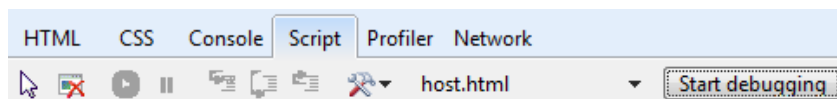
2) 按 F12 键打开开发者工具



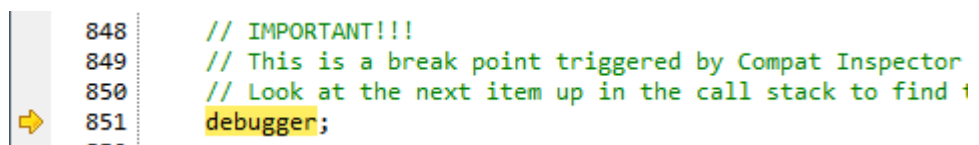
3) 切换到"Script" 选项卡



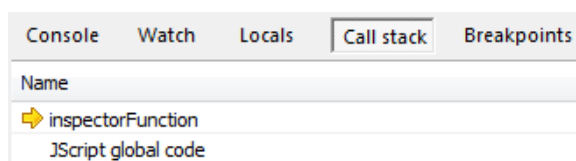
4) 点击"Start debugging"



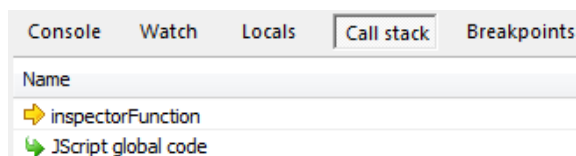
5) 重复最初触发该错误或警告的步骤(这将调用脚本调试器)



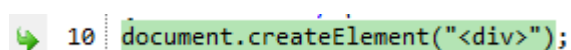
6) 切换到 F12 开发者工具右侧的" Call stack" 子选项卡



7) 在调用堆栈中，双击黄色箭头指向条目的下方条目

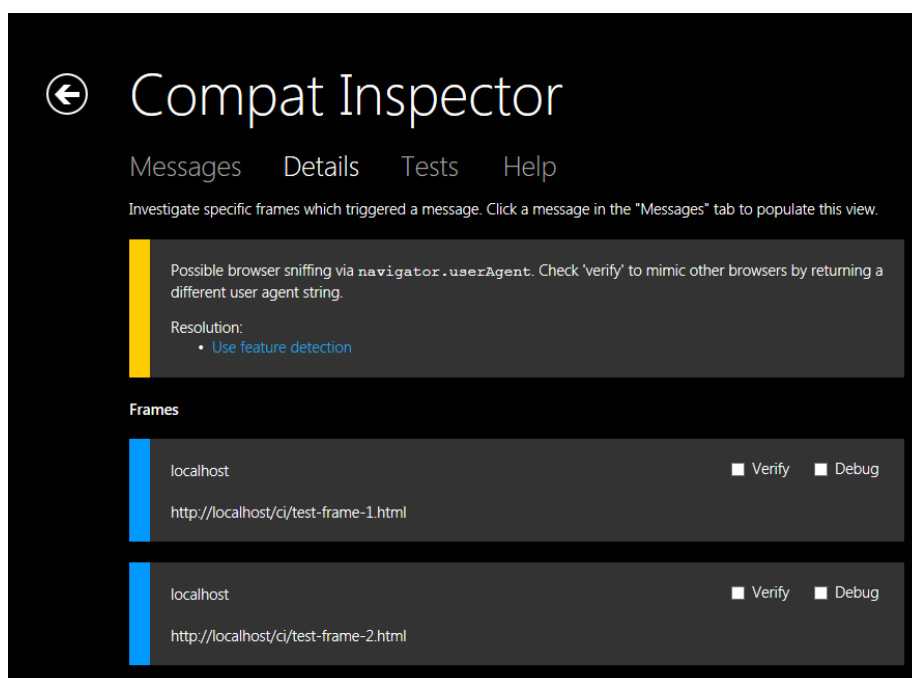


8) 查看 F12 开发者工具左侧部分，检查触发错误或警告的代码段



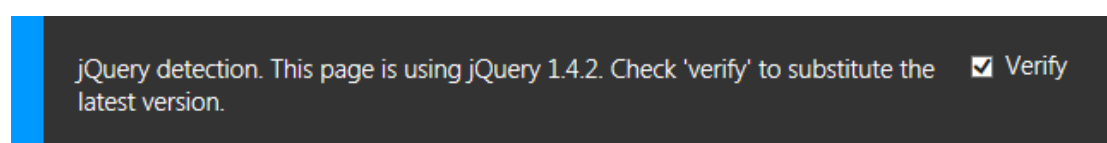
## 4. 鉴别 Frames

许多网页都嵌入了 frame，有些来自于其他域名的网页。如果你使用 Fiddler 向 frame 中注入 Compat Inspector，它也能够发现 frame 中的问题，并产生相应的消息。Compat Inspector 会帮助你找出哪一个或哪些 frame 触发了一个消息。只需要点击这个消息并且在“Details”视图中打开它，视图中就会显示出每一个触发该消息的 frame。每个 frame 条目包含了该 frame 的 URL 以及“验证”和“调试”选项。使用这些选项你能够更准确地定位某个 frame 中的问题。



## 5. JavaScript 库

一个常见的问题是使用“过时”的 JavaScript 库。Compat Inspector 使用两个步骤帮助发现由使用“过时”的 JS 库所引发的问题。首先，它会检测并报告你所使用的库的版本，然后勾选“Verify”复选框将当前版本的库替换为最新版本。使用这个方法你就能够快速鉴别问题是否由使用“过时”的 JS 库引起的。



## 6. 最后的步骤

由于 Compat Inspector 是一个测试工具，不能用在发布的产品当中，因此在发布你的网站之前，要确保将 Compat Inspector 脚本从所有页面中移除。

# 让你的网站在 Internet Explorer 10 上

## 变得更酷

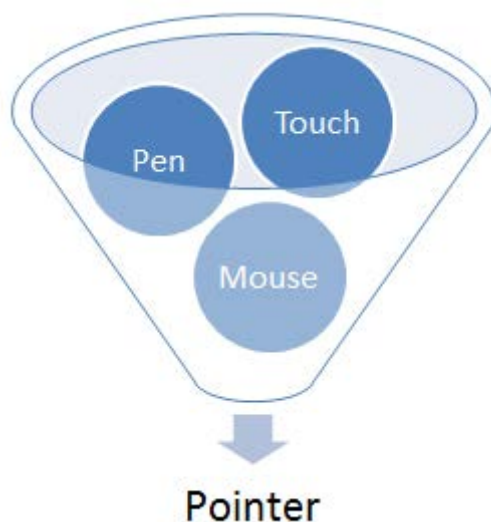
### 一、为你的网站带来触摸体验

随着新的输入设备和触摸屏的到来，网络变得更有互动性、更有趣、更令人身临其境。Windows 开发人员预览版在网站和应用程序中引入了对触摸输入和手写笔输入处理的支持。开发人员现在可以确保他们的网站可以很好地支持触摸操作，构建出震撼的新体验，最大限度地利用多点触摸设备、手势等高级输入方式。在 Internet Explorer 测试驱动网站上新增和更新的演示触摸效果、捕鸟游戏、飞行图片中可以看到实际的效果。

#### 1. 在不影响鼠标的情况下，优先处理触摸输入

“Windows 8” Metro 风格的 IE 及应用程序给 Windows 带来了一流的触摸体验，同时并没有牺牲其他形式的输入，开发人员现在可以构建出带有同样触摸优先体验的网站和应用程序。

首先从基本的输入处理开始。在 Internet Explorer 10 和 Metro 风格的应用程序中，开发人员能够写入更抽象的输入形式，称为“Pointer”。Pointer 可以是鼠标光标、笔、手指、或多个手指在屏幕上的任何接触点。利用这个模型编写的网站和应用程序，不论用户使用什么样的输入硬件，都可以很好地工作。与硬件加速的方法类似，硬件越好，这个体验越好，但开发人员编程时使用的 API 却是与硬件无关的。



Pointer 事件封装了来自触摸、笔、鼠标的输入，有助于构建与独立于硬件的体验

捕捉通用 Pointer 输入的事件，看起来与鼠标事件很像：MSPointerDown, MSPointer Move, MSPointerUp, MSPointerOver, MSPointerOut 等。

Pointer 事件提供了鼠标事件的全部常规属性( 客户的 X/Y 坐标、目标元素、按钮状态等 )，还提供了针对其他输入形式的新属性：压力、接触几何、倾斜等。所以开发人员很容易针对 Pointer 事件编程，不论使用什么样的输入硬件，他们的程序都可以工作。

有些时候，开发人员想为触摸输入提供不同的体验。这时，Pointer 事件也可以通过 event.pointerType 属性指定输入的类型（触摸、笔、鼠标）。

下面是对 IE10 开发人员指南 (<http://msdn.microsoft.com/library/ie/hh673549.aspx>) 中的绘图程序稍加修改的一个基本的绘图程序：

```
<style>
html {
overflow: hidden;
-ms-content-zooming: none; /* Disable pan/zoom */
}
</style>

<canvas id="drawSurface" width="500" height="500" style="border: 1px solid
black;"></canvas>

<script type='text/javascript'>
var canvas = document.getElementById("drawSurface");
var context = canvas.getContext("2d");
context.fillStyle = "rgba(255, 0, 0, 0.5)";
canvas.addEventListener("MSPointerMove", paint, false);

function paint(event) {
context.fillRect(event.offsetX, event.offsetY, 5, 5);
}
</script>
```

Internet Explorer 10 默认启用触摸屏幕的触控 ( panning ) 和缩放。开发人员有时可能想自行管理网站的触控和缩放。在这个示例中，我们将介绍如何在网站上处理触摸输入，以及如何使用样式规则 overflow: hidden 和 -ms-content-zooming: none 准确地禁止触控和缩放。

## 2. 内置的多点触摸支持

每次触摸接触都会触发按下、移动、抬起一系列事件。所以上面绘图示例这样的应用程序无需任何特殊代码就能支持多点触摸。某些情况下可能需要了解屏幕上其他 Pointer 的信息。在任何 Pointer 事件中，都可以方便地得到屏幕上所有 Pointer 的完整列表：

```
<style>
html {
  overflow: hidden;
  -ms-content-zooming: none; /* 禁用触控和缩放*/
}
#foo {
  width: 500px;
  height: 500px;
  background-color: red;
}
</style>

<div id="foo"></div>

<script>
function handleEvent(event) {
  var currentPointers = event.getPointerList();

  if (currentPointers.length == 1) {
    event.target.style.backgroundColor = "red";
  } else {
    event.target.style.backgroundColor = "green"; //使用了多点触摸点
  }
}

document.getElementById("foo").addEventListener("MSPointerMove", handleEvent, false);
</script>
```

### 3. 高级手势输入

Windows 开发人员预览版还支持高级 Pointer 手势的识别，例如缩放、触控和旋转。开发人员可以通过 MSGestureStart、MSGestureChange 和 MSGestureEnd 事件方便地利用这些支持。对于每个手势，都提供了手势转换的信息（旋转、缩放、转变，等等），可以用多种方式在应用程序中使用，例如 CSS 转换：

```
<style>
html {
    overflow: hidden;
    -ms-content-zooming: none; /* 禁用触控和缩放*/
}
#foo {
    background-color: red;
    width: 500px;
    height: 500px;
    -ms-transform-origin: 50%;
    -ms-transform-origin: 50%;
}
</style>

<div id="foo"></div>

<script>
function handleEvent(event) {
    event.target.style.msTransform = "scale(" + event.scale + ")";
}

document.getElementById("foo").addEventListener("MSGestureChange",
handleEvent, false);
</script>
```

## 4. 功能检测、fallback 和其他模型的支持

对于跨其他平台使用的代码，IE10 为 Pointer 事件提供了简单的功能检测：

```
if (window.navigator.msPointerEnabled) {  
    // 系统将触发 Pointer 事件  
}
```

**注意：**在当前的 Windows 开发人员预览版中，这个属性代表系统支持触摸输入或笔输入的 Pointer 事件。但未来会更新，代表对鼠标、笔、触摸设备的 Pointer 事件支持。

使用功能检测，可以让网站支持不同的输入模式。“捕鸟”游戏示例在 Windows 8 开发人员预览版、Apple iOS、Google Android、以及纯鼠标系统上都运行得很好。在 Windows 8 上，它使用 Pointer 事件在一条代码路径中处理所有输入。在其他平台上，它使用鼠标事件和专门的触摸事件结合，提供类似的体验。

```
if (window.navigator.msPointerEnabled) {  
    elem.addEventListener("MSPointerDown", handleInput, false); // 对触摸、笔、鼠标  
    触发  
} else {  
    elem.addEventListener("mousedown", handleInput, false); // 仅对鼠标触发  
}
```

Pointer 事件和手势事件只是触摸开发人员模型的一部分。在未来的文章中，我们将深入介绍其他触摸 API 以及“捕鸟”游戏。

关于 Pointer 事件、手势事件、以及其他触摸 API 的更多细节，请参阅 IE10 开发人员指南（<http://msdn.microsoft.com/library/ie/hh673549.aspx>）。我们期待着您所构建的触摸体验，也欢迎您通过 <http://connect.microsoft.com/ie> 提供反馈。

## 二、创建一个无插件的浏览体验

因为越来越多的浏览活动是在日益增多的设备以及这些设备自带的浏览器上发生的,因此越来越多的消费者正在没有插件的情况下浏览。向这些不使用插件浏览的消费者提供优秀的网站体验,是一项非常重要的工作,因为网站必须要面向最广泛的目标群体。利用 HTML5,现代的浏览器和网站即使不用插件,也能提供优秀的消费者体验。

“插件”泛指使用浏览器的底层接口运行本地客户端代码的浏览器扩展。例如,这里是 Webkit 方法的简单描述 ;在 IE 里对应的就是 ActiveX 控件和浏览器辅助对象( Browser Helper Object )。网站使用各种各样的插件,ActiveX 就是最常见的插件之一( 为了让用户享受到更多更丰富的在线内容,微软将会和 Adobe 一起开发一款适用于 Metro IE 的 Flash 插件,但是其他插件见不被支持 )。

### 1. 无插件浏览变得越来越普遍

今天的许多 Web 浏览活动,都是在不支持“插件”的设备上发生的。支持插件的浏览器提供了许多不用插件的运行方式。例如 Internet Explorer 9 就提供了 ActiveX 筛选,其他浏览器则提供了控制插件的附件。虽然可以将插件编译为在 64 位浏览器中运行,但许多开发人员只发布了在 32 位浏览器中运行的版本,运行 64 位浏览器是无插件浏览变得越来越普遍的另一原因:

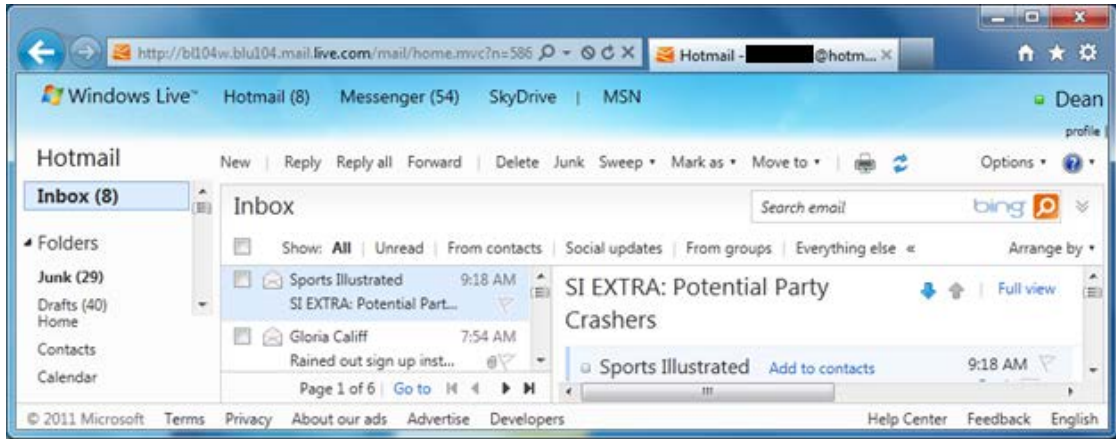


64 位 Windows 同时包含 32 位和 64 位的 IE



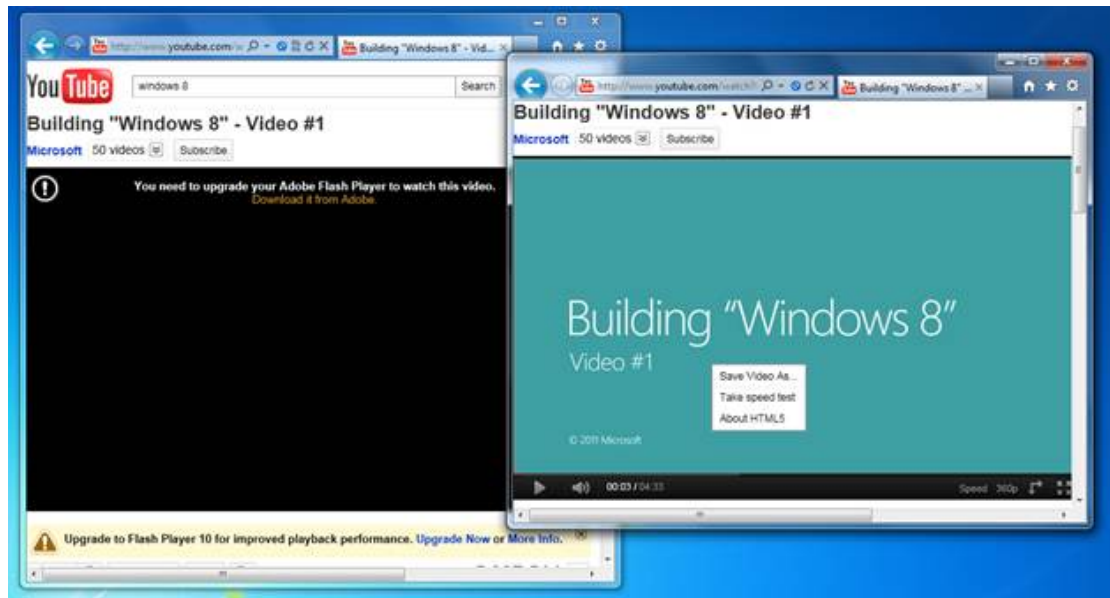
## 2. 越来越好的浏览体验

许多网站现在已经提供了插件不可用时的良好体验。例如，没有插件时（没有安装插件，如在 64 位浏览器中，或者通过 ActiveX 筛选禁用）用 Internet Explorer 9 访问 Hotmail 时工作得很好：



IE9 没有插件时访问 Hotmail 收件箱

有些网站在使用无插件方式工作之前，需要使用者采取一些操作。例如，如果想在没有插件的情况下使用 YouTube，需要先访问 <http://www.youtube.com/html5>，并单击“Join the HTML5 Trial”（加入 HTML5 试验）：



在 IE9 中无插件时显示的 YouTube ；

左侧是加入 HTML5 试验之前，右侧是加入 HTML5 试验之后；

右侧的上下文菜单表明，视频使用 HTML5 播放

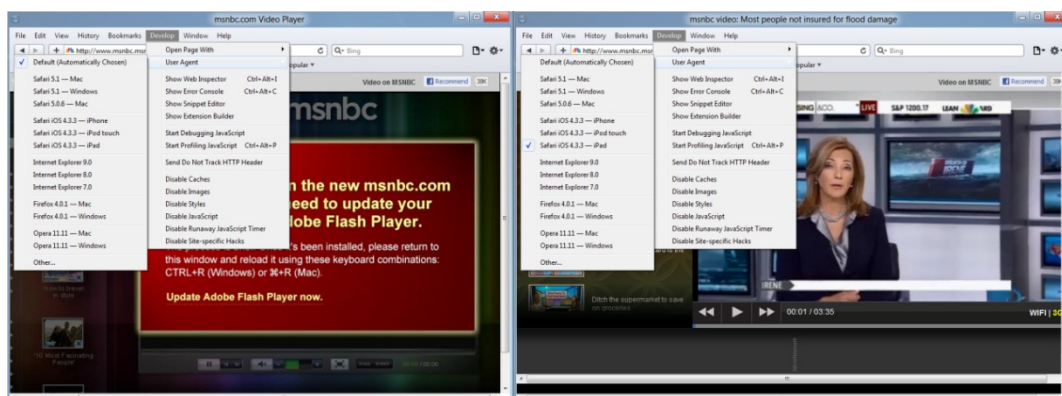
其他网站在禁用插件时，则屏蔽部分或全部功能。例如，MSNBC.com 和 CNN.com 取消视频；Gmail 目前屏蔽禁用插件的 Internet Explorer 9，因为它检查 XHR ActiveX 对象，而不是使用从 Internet Explorer 7 以来就一直可用的 Web 标准的 XHR。从 Internet Explorer 7 之后，Web 已经有了很大发展，所以网站应该回头审视那些特定于旧版本浏览器或旧版本标准的代码。



有些网站在没有插件时禁用部分或全部功能

### 3. 功能检测，网站开发人员如何做得更好

许多网站已经提供了在浏览器插件不可用时的优秀体验。但当网站根据用户使用的设备或浏览器提供这种体验的时候，用户有一个问题。例如，如果一台 PC 上的浏览器发送某台设备的 User-Agent 字符串，则 MSNBC.com 视频可以在无插件的情况下播放，但在同一台机器同一浏览器上，如果发送不同的 User-Agent 字符串，视频就无法播放：



左图是 MSNBC.com 在没有安装 Flash 的 Apple Safari 中的显示效果；

右图是同一浏览器把自己标记为 iPad 时的显示效果

如果开发人员使用功能检测和降级（fallback），而不是将网站硬编码到特定的浏览器和配置上，对用户会更好。例如，在没有插件的时候，检测并使用 HTML5 视频，可以给用户提供更好的体验。许多网站在没有插件却要显示广告的时候，已经在做这种降级处理，证明这是一个实用而且具备伸缩性的解决方案。

如果网站按照最佳做法，先测试基于标准的功能是否可用，然后只在必要的时候用插件进行降级，用户体验会更好。例如，下面分别使用 XMLHttpRequest 进行功能检测的良好模式和不良模式：

```
// 不良模式：不要这样做！
var xhr = window.ActiveXObject
? new ActiveXObject("Microsoft.XMLHTTP")
: new XMLHttpRequest();

// 最佳做法：如果可用，使用本地 XHR
if (window.XMLHttpRequest) {
// 如果是 IE7+, Gecko, WebKit：使用本地对象
var xmlHttp = new XMLHttpRequest();
}
else if (window.ActiveXObject) {
// ...如果不是，尝试 ActiveX 控件
var xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");
}
else {
// 没有 XMLHttpRequest 机制可用。
}
```

优雅地降级到其他内容，对于避免为消费者造成不佳体验非常重要。例如，在没有 Adobe Flash 的时候，Gmail 视频聊天会优雅降级。有些网站只根据 User Agent 字符串或其他特定于浏览器的属性提供降级。这种短期解决方案无法适应浏览器和用户浏览方式的变化。构建一个应用程序是有些网站（例如 IMDB.com）对浏览器不支持扩展项的设备所采用的另一种解决方案。User Agent 字符串已经不再是决定应该提供哪个 HTML 和脚本的可靠或健壮的方式了。

HTML5 的能力使得消费者可以在更多设备、更多配置上体验当今的 Web。对于 Web 开发人员来说，这意味着有更多机会可以让用户访问他们的网站，使得他们更有动力让网站在没有插件的情况下也有出色的表现。

# 更多商业机会

Windows 8 消费者预览版允许开发者将自己的网站与系统紧密整合在一起，比如，如果用户喜欢你的网站，他可以将网站固定到自己的“开始”屏幕中，你的网站就成为一个磁贴随时向他推送信息，用户点击这个磁贴就会直接通过 Internet Explorer 10 打开你的网站，大大提升了用户对你网站的关注度，为你创造更多的商业机会。

## 一、在 Windows 8 中固定网站

Windows 8 “开始”屏幕是查找和保持连接所有喜欢的应用和内容的最佳位置。应用的磁贴是随着活动实时更新的，显示了最新的和定制的内容，让您可以了解与你相关的最新动态。

我们撰写了一篇介绍 Windows 8 的 Internet Explorer 10 Metro 风格的浏览体验的博客文章 (<http://blogs.msdn.com/b/b8/archive/2012/03/13/web-browsing-in-windows-8-consumer-preview-with-ie10.aspx>)。这篇文章详细描述了 Internet Explorer 10 的网站固定及其在 Windows 8 “开始”屏幕上的使用，以及以网站为中心的视图并通过锁屏提醒通知来提醒您有新内容可用。我们还从 Web 开发人员的角度详细介绍了如何支持网站固定。

### 1. 固定网站比收藏夹更好用

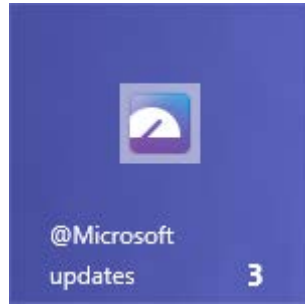
Windows 用户花费大量时间上网冲浪 我们通过 Windows 用户访问跟踪测试 参见官网：<http://www.microsoft.com/products/ceip/EN-US/default.aspx>) 知道，用户经常会反复访问相同的一些网站。Windows 8 的固定网站功能使您可以更快、更轻松地从返回您喜欢的网站。利用锁屏提醒通知，网站磁贴可以实时更新最新的信息，让您知道有新内容可用。

开发人员可以利用固定网站来加强网站与用户之间的联系，直接在 Windows “开始”屏幕上固定网站还有助于提升网站的知名度。我们发现在 Windows 7 的 Internet Explorer 9 中，使用此功能的网站的访问量提高 15% - 50%。在 Windows 8 中，网站磁贴更新为用户提供了更好的体验，用户甚至不必在浏览器中打开网站。

以下两幅屏幕截图显示了网站固定磁贴和锁屏提醒通知：



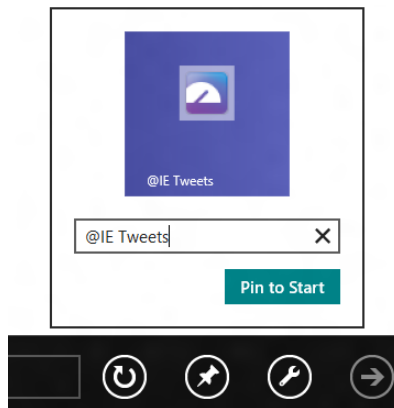
“开始”屏幕上的以网站为中心的锁定画面和锁屏提醒通知



网站固定磁贴上的锁屏提醒通知示例

## 2. 以网站为中心的网站固定磁贴

作为网站开发人员，您可以提供一个网站图标（favicon），Internet Explorer 10 将在浏览器中使用该图标表示您的站点，在地址栏、新的标签页和“开始”屏幕上均使用该图标。当将网站固定到“开始”屏幕上时，Internet Explorer 10 使用网站的大图标（32 x 32 像素）来标识网站，这与 Internet Explorer 9 用来固定任务栏的方式是相同的。



将一个网站固定到 Metro 风格的 IE10 中，显示了网站磁贴的预览

Internet Explorer 10 从图标中提取主色并自动使用该颜色作为“开始”屏幕上磁贴的背景颜色。一种用于创建图标的工具是 x-icon editor。使用该工具可以为您的网站创建 32x32 像素的图标。它还可以将图像转换为图标文件格式（.ico）。然后可以使用传统的 favicon 标记来关联 .ico 文件。

```
<link href="testdrive.ico"rel="shortcut icon"/>
```



### 3. 可以在后台更新锁屏提醒通知

Windows 8 可以在“开始”屏幕的磁贴上动态显示内容。这非常适合于显示有更新内容的通知。例如来自其他用户的新消息（电子邮件和社交网络）、购物网站的最新打折信息、新闻网站的新文章，等等。

利用 Internet Explorer 10 和 Windows 8，您可以直接在网站固定磁贴上提供锁屏提醒通知。这意味着用户可以无需在浏览器中打开网站即可获得网站的更新。例如使用 Windows 8 消费者预览版的 Internet Explorer 10 来固定 Fresh Tweets demo（<http://ietestdrive2.com/pinnedsites/>）。网站固定磁贴将定期更新，并通知用户何时有了新的微博发布。

后台通知需要网站提供的组件包括：(1) 锁屏提醒通知 XML，这是一个 XML 响应，它描述了 Windows 的锁屏提醒通知；(2) 网站固定元标签，这是网页标记，它指向 Windows 动态显示通知的位置和动态显示的频率。

#### 1) 锁屏提醒通知 XML

Windows 将处理动态显示和锁屏提醒通知的绘制。Windows 动态显示锁屏提醒通知 XML，这个 XML 描述了网站固定磁贴的视图。badge XML schema（<http://msdn.microsoft.com/en-us/library/windows/apps/br212849.aspx>）定义了这个简单的 XML 响应。例如，为了用数字“3”来更新磁贴，我们发送以下 XML：

```
<?xml version="1.0" encoding="utf-8" ?>
<badge value="3"/>
```

锁屏提醒可以是数字或图像字符，例如“警告”和“新消息”提示。有关锁屏提醒可以显示的内容的完整列表，请参见 MSDN 上的 Choosing a badge image（<http://msdn.microsoft.com/en-us/library/windows/apps/hh761458.aspx>）。

#### 2) 网站固定元标签

下一步是将锁屏提醒通知 XML 关联到网页。Internet Explorer 10 使用元标签“application-name”来确定网页是否支持网站固定功能（例如通知和跳转列表）。为了支持锁屏提醒通知，需要在标记中包含一个特定的(msapplication-badge)的元标签，并指定锁屏提醒通知 XML 的 URL 和 Windows 对其更新的频率。IE 在固定时将检查页面上是否存在“msApplication-badge”元标签，之后从网站固定磁贴启动网站时也会进行这项检查。

```
<meta name="msapplication-badge" value="frequency=360;polling-
uri=http://ietestdrive2.com/PinnedSites/TweetCounter/microsoft/360"/>
```

值参数有两个部分：polling-uri（必需）和 frequency（可选）

polling-uri 是绝对 URL，Windows 将从它请求上面定义的简单 XML 文档。

Frequency 是可选的，它表示两次更新之间的分钟数，必须使用以下值：

- 30 ( Windows 将每隔 30 分钟变换一次 URI )
- 60 ( 1 小时 )
- 360 ( 6 小时 )
- 720 ( 12 小时 )
- 1440 ( 1 天, 这是默认值 )

如果省略 frequency, 或者未使用上面列出的值, 则将默认每天更新一次 ( 1440 分钟 )。

### 3) 用于更新锁屏提醒通知的开发人员 API

您还可以直接从网页清除和更新网站的锁屏提醒磁贴, 以确保网站磁贴是最新的。

当用户从“开始”屏幕启动网站固定磁贴时, 网站将运行其自己的一个名为“SiteMode”的会话, 并且可以使用以下 JavaScript 函数来更新锁屏提醒:

`window.external.msSiteModeClearBadge()`将清除磁贴上的锁屏提醒通知。在 Fresh Tweets demo ( <http://ietestdrive2.com/pinnedsites/> ) 中, 当用户收到一个通知并点击磁贴来启动浏览器时, 网页使用 `msSiteModeClearBadge()`来清除磁贴上的通知。下次磁贴获取锁屏提醒更新而变亮时, 用户就会知道有新内容可用。

`window.external.msSiteModeRefreshBadge()`调用 Windows 来使用变换 URI 更新网站的锁屏提醒。例如, 用户看到网站固定磁贴显示有 3 条未读消息, 单击磁贴返回网站。如果用户只读了第一条更新, 您可以触发一次更新, 以便锁屏提醒反映出正确的已读/未读的消息数 ( 2 条未读 )。

在 Windows 8 消费者预览版中, 这两个 API 函数仅能用于运行在本地局域网或受信任的网站区域中的网站。这一问题将在下一个预览版中解决。要在您自己的网站上测试消费者预览版中的这些 API, 请使用“Internet 属性”对话框中的“安全”选项卡将网站的域名添加到受信任的网站列表中。

### 4) 桌面任务栏通知

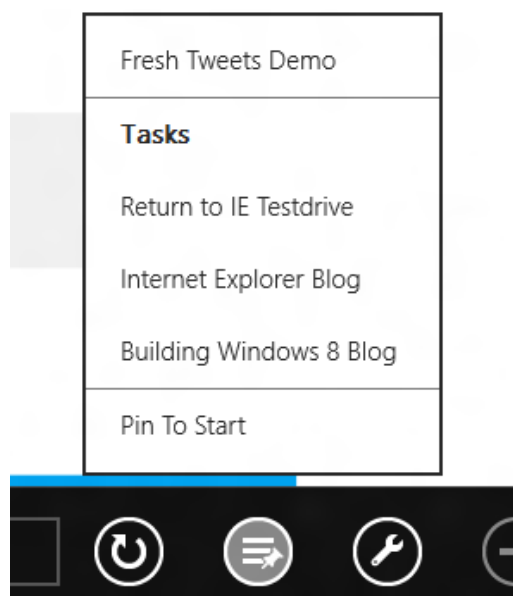
Windows 8 和 Internet Explorer 10 仍然可以在桌面任务栏上使用以图标覆盖形式出现的通知来进行站点的固定。这种形式的通知不适用于 Windows 8 “开始”屏幕上的网站固定。Windows 8 将以一种省电模式处理“开始”屏幕上的所有磁贴通知。

### 5) 使用跳转列表在网站中快速导航

很多顶级网站, 例如 NYTimes.com、CNN.com 和 Amazon.com, 都支持 Internet Explorer 9 的网站固定功能, 例如通过跳转列表 ( <http://windows.microsoft.com/en-US/internet-explorer/products/ie-9/features/pinned-sites> )

来获取特定任务或网站的部分内容。在 Internet Explorer 10 中，跳转列表作为导航栏的一部分提供，以一种“友好触摸”的方式提供了网站的导航。

当用户从“开始”屏幕打开网站时，固定按钮将显示此站点有跳转列表可用。



Fresh Tweets Demo 的跳转列表

跳转列表的显示方式与 Internet Explorer 9 在 Windows 7 任务栏上的显示方式是相同的。

您可以通过网页标记将静态任务添加到网站的跳转列表中，也可以根据用户交互动态添加。Windows 8 锁屏提醒和跳转列表都是以网站为中心的功能。每个完全限定域名有且仅有一组变换数据和跳转列表数据。



## 二、链接预览

链接预览功能扩展了 Windows8 的分享体验。当用户从超级按钮栏 ( Charm Bar ) 中选择“分享”时，Internet Explorer 会生成一个链接预览——这是一个分享的数据对象，里面包含当前网页的强化信息。除了页面的 URL，链接预览还包含网页的标题、描述、缩略图。为了生成这个链接预览，Internet Explorer10 会寻找特定的 HTML 标签。要想利用这个功能，可以在网站的 HTML 中包含以下支持链接预览的标签。

### 1. 标题标签

标题内容的长度限定在 160 个字符以内，更长的标题在链接预览中会被截断。

```
<meta name="title" content="My Title Text" />
<title>My Title Text</title>
```

### 2. 描述标签

描述内容的长度限定在 160 个字符以内，更长的描述在链接预览中会被截断。

```
<meta name="description" content="My description text." />
```

### 3. 缩略图标签

图片 URL 的长度限定在 2048 个字符以内。

```
<link rel="image_src" href="my_image_link.jpg" />
<meta name="image" content="my_image_link.jpg" />
<meta name="thumbnail" content="my_image_link.jpg" />
<meta property="og:image" content="my_image_link.jpg" />
```

以上这些标签可以任意组合，但我们推荐每类至少使用一个，以创建尽可能好的链接预览。如果使用了同一类的多个标签，则链接预览使用网站 HTML 中列出的第一个标签。

# 让你的网站在低版本浏览器上同样显示出色

HTML5 的采用策略：一个浏览器也不放弃。

下载示例代码：<http://code.msdn.microsoft.com/mag201109HTML5>

HTML5 有许多激动人心的特性，有了新的标签、新的 CSS 能力和新的 JavaScript API，Web 的能力范围有了大的飞跃。除了浏览器厂商的士气高涨之外，令人激动的新功能列表几乎每天都在增加。从“nightly builds”（每夜都构建一版）到开发渠道发行版和正常的平台预览版，浏览器在飞速变化，世界各地 Web 开发人员们正在加入这个狂欢。

但是，尽管开发和浏览器社区正在把 HTML5 的喧嚣推到一个极度兴奋的高潮，网上的大多数人却不像我们一样使用最新的浏览器和最新版本。如果你是一个大型开发机构的 Web 开发人员或者是拥有庞大用户群的大企业，那你对此可能很清楚。即使你为通过 Web 提供服务的小型机构或新创立的企业工作，你可能也要花上大量时间来确保自己的网站能够支持尽可能多的浏览器和浏览器版本。

基于这一现实，很容易看出，HTML5 还谈不到它是否已为当今的使用做好准备，而是你是否为它做好了准备。例如，假设你用一些新的语法标签（例如<header>和<article>）新建了一个页面，添加了一些新的 CSS 功能，例如圆角（border-radius）和阴影（box-shadow），甚至添加了一个<canvas>元素在页面上绘制出一个 HTML5 标识。

在较新的浏览器上，例如 Internet Explorer 9、Firefox 4 及以上版本、或者 Google Chrome 上，这个页面的显示如图 1 所示。但如果尝试在 Internet Explorer 8 或更早的浏览器上加载页面，很有可能看到的是图 2 所示的效果：一个残缺不全的页面。



图 1 使用 HTML5 语法、样式和<canvas>元素的简单页面在 IE9 中显示的效果



图 2 同一个页面，在 IE8 中显示时，没有样式，也没有<canvas>

如果你在研究 HTML5 的所有强大功能却得到上述体验之后告诉自己说：最好还是等等，那么我不会对你有任何责怪。如果我问你准备好了吗？你很容易得出这样的结论：HTML5 还没有为你或你的用户做好准备。

在你决定等到 2022 年再考虑 HTML5 的之前，我建议你继续阅读本文的后面部分，我将向你提供一些实用的策略，让你现在就能采用 HTML5 技术，同时避免出现图 2 所示的糟糕的降级情况。我将从下面三个主题进行详细地介绍：

- 功能检测与用户代理 ( UA ) 嗅探比较
- 用 JavaScript 实现填补 ( Polyfill )
- 优雅降级

这些应该可以教会你很多构建支持各种浏览器的网站所需要了解的知识。在本文结束时，你会拥有一个可靠的策略，可以充满自信、毫不犹豫地采用 HTML5 技术。你还会拥有一些工具在手，可以逐步地为新浏览器增强网站，同时更好地适应旧的浏览器。

## 一、功能检测的重要性

为了提供跨浏览器的稳定且一致的体验，开发人员经常需要获得一些关于用户浏览器的信息。以前的普遍做法是像下面这样用 JavaScript 检测这些信息：

```
var userAgent = navigator.userAgent;

if (userAgent.indexOf('MSIE') >= 0) {
    console.log("Hello, IE user");
} else if (userAgent.indexOf('Firefox') >= 0) {
    console.log("Hello, Firefox user");
} else if (userAgent.indexOf('Chrome') >= 0) {
    console.log("Hello, Chrome user");
}
```

这个技术称为用户代理 ( UA ) 嗅探，广泛地用于判断正在请求页面的是哪个浏览器。这里的思路是：知道了用户的浏览器 ( 例如 IE7 )，就能在运行的时候决定启用或禁用网站的哪项功能。UA 嗅探就相当于对浏览器说：“你是谁？” ( 对 UA 嗅探以及其他检测技术的深入分析，请参阅 [bit.ly/mlgHHY](http://bit.ly/mlgHHY)。 )

这种做法的问题在于，浏览器会撒谎。UA 字符串是一个用户可以配置的信息，并不会提供 100% 正确的浏览器信息。而且，随着这一技术的广泛采用，许多浏览器厂商在自己的 UA 字符串中增加了额外内容，用来欺骗脚本，让脚本对于实际使用的浏览器做出错误判断，从而避免检测。现在有些浏览器甚至提供小工具，允许用户只要轻轻点击几下鼠标，就能修改 UA 字符串。

UA 嗅探的目的从来就不是确定用户的浏览器和版本。而且它肯定也不是为了在你不喜欢用户使用的浏览器时，让你可以告诉用户说“请下载另一个浏览器”——即使有些人就是这样使用 UA 嗅探技术的。用户有权选择自己使用什么浏览器，开发人员的职责则是提供最可靠且一致的体验，不要把浏览器的偏好强加给用户。UA 嗅探的目标是让你能够准确地了解在用户当前的浏览器中，有哪些能力或功能可以利用。对浏览器本身的了解，只是获得这些信息的一个途径。

目前有一些 UA 嗅探的替代技术，其中一项正在日益流行的技术称为对象检测或功能检测。这两个术语多数时候可以互换使用，但本文统一使用“功能检测”（feature detection）。

功能检测的目标是判断某项功能或能力在用户当前的浏览器中是否受支持。如果 UA 嗅探是问浏览器“你是谁”，“功能检测”就是问浏览器“你能干什么”，这个问题更直接，对于根据条件向用户提供功能来说，这种方法也更可靠。如果功能检测脚本实现正确，用户或浏览器将很难造假或错报功能支持。

## 二、手动功能检测

那么，与 UA 嗅探的示例相比，功能检测到底是什么样呢？为了回答这个问题，我们先来看看如果在 Internet Explorer 8 中查看前面的 HTML5 页面（如图 1 所示），如何解决出现的问题。这个页面的标签内容如代码段 1 所示。

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>My Awesome Site</title>
  <style>
    body { font-size: 16px; font-family: arial, helvetica, clean, sans-serif; }
    header h1 { font-size: 36px; margin-bottom: 25px; }
    article
    {
      background: lightblue;
      margin-bottom: 10px;
      padding: 5px;
      border-radius: 10px;
      box-shadow: 4px 4px 10px rgba(0, 0, 0, 0.5);
    }
    article h1 { font-size: 12px; }
  </style>
</head>
```

```
<body>
  <header> <h1>My Awesome Site</h1> </header>
  <article>
    <header> <h1>An Article</h1> </header>
    <p>Isn't this awesome?</p>
  </article>
  <canvas width="250" height="500"> </canvas>
</body>
<script src="../../js/html5CanvasLogo.js" type="text/javascript"> </script>
</html>
```

代码段 1 带有 HTML5 新语义标签的页面

如图 1 和图 2 所示，Internet Explorer 9 和 Internet Explorer 8 的显示效果有很大的差别。对于初学者来说，我的页面完全没有样式，因为这个页面的 CSS 并不存在。而且，页面底部丢失了好玩的 HTML5 盾牌。每个问题都可以轻松解决，而功能检测则是明确问题的第一步。

两个问题的原因都很简单：对于 Internet Explorer 8 来说，<header>、<article>和<canvas>都不是有效的 HTML 元素，所以无法使用。要解决<canvas>问题，我们不用 UA 嗅探来判断所使用的浏览器/版本，而要通过 JavaScript 询问浏览器是否支持<canvas>元素和它的 JavaScript API，对 Canvas 的功能检测如下所示：

```
!!document.createElement('canvas').getContext
```

这条语句做了好几件事：首先，它使用两个否定 (!!) 操作符强行将未定义的值显式地设为 false；然后，它手动新建一个 canvas 元素，并将它加到 DOM 中；最后，它调用 getContext 函数，这是<canvas>元素的一个新函数，是通过 JavaScript 操纵 Canvas API 的途径。如果使用 IE9，则这条语句会返回 true。如果使用 IE8，则 getContext 会返回“undefined”（未定义），会被前面的两个否定操作符强行变为 false。

这是最基本的功能检测。利用这条语句以及其他类似语句，就有了查询浏览器所支持功能的更可靠方法。关于手动功能检测的更多信息，请参阅 [diveintohtml5.info/everything.html](http://diveintohtml5.info/everything.html)。

### 三、使用 Modernizr 进行功能检测

手动功能检测肯定是对 UA 嗅探的提高,但这种做法仍然需要你做大量工作来检测功能是否可用,以及在功能不存在的时候决定做什么。虽然 Canvas 示例很简单,只需要一行代码,但不是每个要检测的功能都这么简单——不同浏览器的检测代码也各不相同。例如,要检测是否支持前面使用的 CSS3 模块 ( border-radius 和 box-shadow ) 就有些麻烦。

值得庆幸的是, Modernizr ([modernizr.com](http://modernizr.com)) 提供了更好的方法。Modernizr 是一个 JavaScript 库 “.....检测下一代 Web 技术 ( 即源于 HTML 5 和 CSS3 规范的功能 ) 的本地实现是否可用”。在页面上添加对 Modernizr 的引用可以提供四大功能 :

- 1) 全面列出支持的功能,智能地加入标签,从而实现 CSS 的条件定义。
- 2) 一个 JavaScript 对象,方便进行基于脚本的功能检测。
- 3) 在运行的时候将全部 HTML5 新标签加入 DOM,方便 IE8 和之前的 IE 浏览器 ( 稍后就会知道不仅如此 )。
- 4) 一个脚本加载器,可以根据条件将 polyfill 加载到页面中。

本文对第 1 项不做进一步介绍,但鼓励你访问 [modernizr.com](http://modernizr.com) 网站,学习这一功能及其余功能的文档。

上面的第 2 项功能,可以将下面的代码 :

```
!!document.createElement('canvas').getContext
```

改为这行代码 :

```
Modernizr.canvas
```

这行代码会返回一个布尔值,表明页面是否支持 Canvas 元素。使用 Modernizr 比自行执行功能检测的好处是, Modernizr 是一个经过良好测试、健壮的、广为采用的库,它已经完成了许多繁重的工作。Twitter、Google、Microsoft 以及无数其他机构和开发人员都在使用 Modernizr,你当然也可以使用。在 ASP.NET MVC 3 工具更新 ( 2011 年 4 月发布 ) 中, Microsoft 甚至随新的 ASP.NET MVC 应用程序一起配备了 Modernizr。当然,我们迄今为止所做的,不过是检测是否支持 <canvas> 元素。通过功能检测知道了浏览器是否支持某一功能之后,接下来通常是创建一些条件逻辑,在功能不存在的时候阻止特定代码的执行或者换个路径来执行,例如 :

```
if (Modernizr.canvas) {  
    // 这里执行 canvas 代码。  
}
```

根据附加的浏览器功能是否存在来给网站增加功能,这种做法称为“渐进式增强”,因为体验增强针对的能力更强的浏览器。另一方面是“优雅降级”,即某项功能的缺失不会造成浏览器出错或发生故障,而是应该向用户提供一些削弱的功能或替代能力。对于旧版浏览器来说,不必将优雅降级作为默认选择。在许多情况下,甚至可能不是最佳选择。相反,在 Modernizr 的帮助下,你通常可以使用许多可用的浏览器 polyfill,将类似于 HTML5 的功能添加到不支持 HTML5 的浏览器中。

## 四、什么是 Polyfill

根据 Modernizr 网站的说法，polyfill 是“在旧版浏览器上复制标准 API 的 JavaScript 补充”。“标准 API”指的是 HTML5 技术或功能，例如 Canvas。“JavaScript 补充”指的是可以动态地加载 JavaScript 代码或库，在不支持这些标准 API 的浏览器中模拟它们。例如，geolocation（地理位置）polyfill 可以在 navigator 对象上添加全局的 geolocation 对象，还能添加 getCurrentPosition 函数以及“坐标”回调对象，所有这些都是 W3C 地理位置 API 定义的对象和函数。因为 polyfill 模拟标准 API，所以能够以一种面向所有浏览器未来的方式针对这些 API 进行开发，最终目标是：一旦对这些 API 的支持变成绝对大多数，则可以方便地去掉 polyfill，无需做任何额外工作。

通过在页面上添加对 Modernizr 的引用，我就得到了与**代码段 1** 示例相关的 polyfill 的直接好处。页面显示没有样式，是因为 IE8 不认识 <article> 和 <header> 标签。因为它不认识这些标签，所以没将它们加入 DOM，而 CSS 选择元素要发挥样式作用，需要在 DOM 中有这些元素。

当我在页面上添加 <script> 标签和对 Modernizr 的引用时，结果就有了样式，如图 3 所示。我之所以得到这个好处，是因为 Modernizr 用 JavaScript (document.createElement('nav')) 手动地将所有 HTML5 的新标签添加到 DOM，这样 CSS 就能选择标签并给标签加上样式。

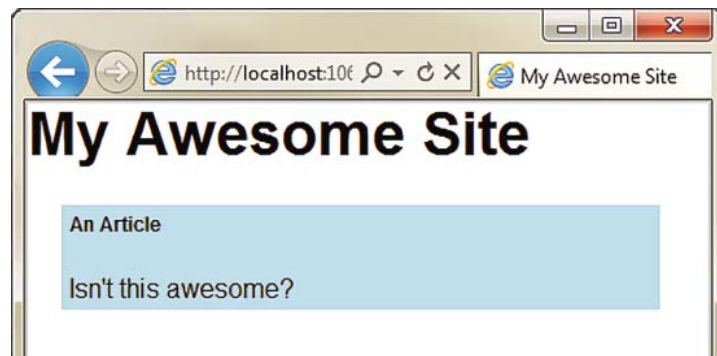


图 3 HTML5 页面在 Modernizr 协助下在 IE8 中显示的效果

除了在 Internet Explorer 中添加对新的 HTML5 元素的支持外，Modernizr 库默认不提供任何额外的 polyfill。额外的 polyfill 需要自行提供，或者使用自己的脚本，或者从 Modernizr 网站上日益增加的选项列表中选择。在 2.0 版中，Modernizr 提供了一个条件脚本加载器（基于 yepnope.js—yepnopejs.com），可以帮助你只在需要的时候异步下载 polyfill 库。使用 Modernizr 配合一个或多个 polyfill 库来提供需要的功能，是一个强大的组合。



## 五、使用 Polyfill 模拟 HTML5 功能

对于 Canvas 来说，在 Modernizr 和 JavaScript 库 excanvas 的帮助下，用 polyfill 可以在 IE8 及之前的版本中实现 Canvas 支持，可以在 IE6、IE7 和 IE8 上添加 API 级别的 Canvas 支持。可以从 [bit.ly/bSgyNR](http://bit.ly/bSgyNR) 下载 excanvas，将它添加到自己的脚本文件夹，然后在页面的脚本块中添加一些代码，如**代码段 2** 所示。

```
Modernizr.load({
  test: Modernizr.canvas,
  nope: '../js/excanvas.js',
  complete: function () {
    Modernizr.load('../js/html5CanvasLogo.js');
  }
});
```

**代码段 2** 用 Modernizr 和 Polyfill 实现 Canvas 支持

在这里，我使用 Modernizr 脚本加载器指定了三件事：

- 1) 用来测试的 Boolean 表达式。
- 2) 表达式测试为 false 时，加载脚本的路径。
- 3) 检查或脚本加载完成时运行的回调。

对于 Canvas 来说，在应用程序中要添加的智能处理和 polyfill 就是这些了。Modernizr 会异步加载 excanvas.js，而且只会为那些不支持 Canvas 的浏览器加载，然后加载脚本库，在页面上绘制出 HTML5 标识。

下面来看另外一个示例来理解 Modernizr 的价值。重视细节的你可能已经注意到，**图 3** 中网站的样式与**图 1** 中在 Internet Explorer 9 中显示的原始页面相同。这个页面在 Internet Explorer 8 中显示时没有阴影和圆角，而我不可能让网站在没有这两个效果的情况下丑陋地交付出去，所以我们再次求助于 Modernizr。

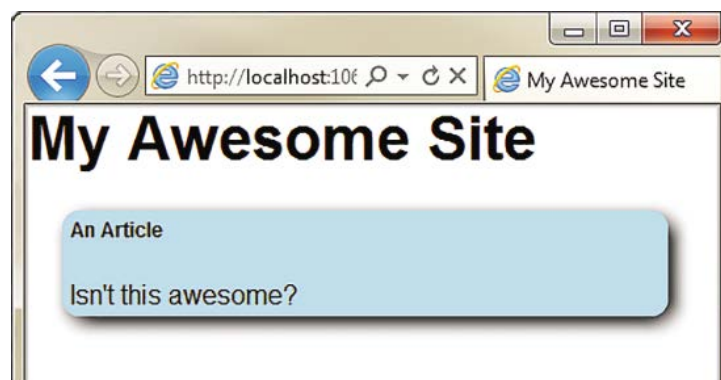
同处理 Canvas 一样，Modernizr 可以告诉我 CSS 模块不受支持，但提供一个填补 (polyfill) 这些 CSS 模块的库，要我来做。幸运的是，有个名为 PIE(css3pie.com)的库在一个库中提供了这两个模块。



为了增加对圆角和阴影的支持,在下载了 PIE 之后,我可以将**代码段 3** 中的代码加入脚本。这次,我要测试是否支持圆角或阴影模块(而不是假定两者都支持或都不支持),如果有哪个模块不支持,则动态地加载 PIE.js。PIE 加载完成后,我再执行一段 jQuery 代码选择全部<article>标签,并调用 PIE.attach 函数,添加 CSS 中已经定义的圆角和阴影样式。最终结果如**图 4** 所示。

```
Modernizr.load({
  test: Modernizr.borderradius || Modernizr.boxshadow,
  nope: '../js/PIE.js',
  callback: function () {
    $('article').each(function () {
      PIE.attach(this);
    });
  }
});
```

**代码段 3** 用 Modernizr 和 PIE 添加 CSS3 支持



**图 4** Modernizr 和 PIE 实现的 CSS3 支持

## 六、使用 Polyfill 协助进行优雅降级

除了使用这里讨论的 polyfill 技术，在希望应用程序优雅降级的地方也可以借助于 Modernizr，而不是用另一个库进行填补（polyfill）。

假设网页上有一个 Bing Maps 控件，而且我希望使用 Geolocation 来确定用户当前位置，然后将这个位置作为大头钉放在地图控件上。

虽然新版本的浏览器都支持 Geolocation，但在旧版浏览器中并不支持。纯粹使用 JavaScript 提供完整的 Geolocation 支持还确实有些麻烦，即使有针对 Geolocation 的填补（polyfill）实现起来也不轻松，所以我决定要对自己的应用程序进行优雅降级。当用户的浏览器不支持 Geolocation 时，我会提供一个表单，用户可以在表单中手动输入位置，我将用用户输入的位置定位和固定地图。

通过 Modernizr，只要用一个简单的加载脚本，调用我创建的两个脚本中的一个即可，如代码段 4 所示。在这个示例中，我测试的是 Modernizr.geolocation 属性。如果为 true（“yep”分支），就加载 fullGeolocation.js 脚本，这个脚本使用 Geolocation API 定位（要得到用户许可），并将位置放在地图上，如图 5 所示。如果测试为 false（“nope”分支），则加载备用脚本，在页面上显示一个地址表单。用户提交表单时，我会使用用户提供的地址将地图居中并固定，如图 6 所示。这样，我的页面为最新的浏览器提供了优秀的体验，同时对旧版浏览器提供了降级到合理替代品的优雅方式。

```
Modernizr.load({
  test: Modernizr.geolocation,
  yep: '../js/fullGeolocation.js',
  nope: '../js/geolocationFallback.js'
});
```

代码段 4 用 Modernizr 提供优雅降级

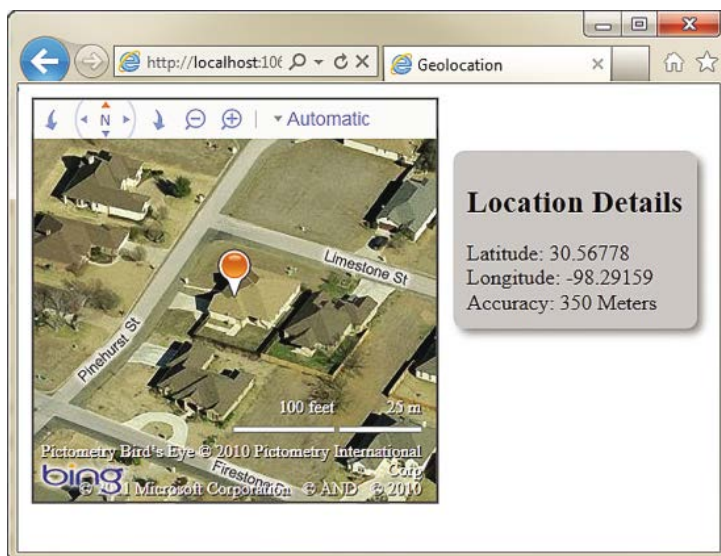


图 5 使用 Geolocation 的地图

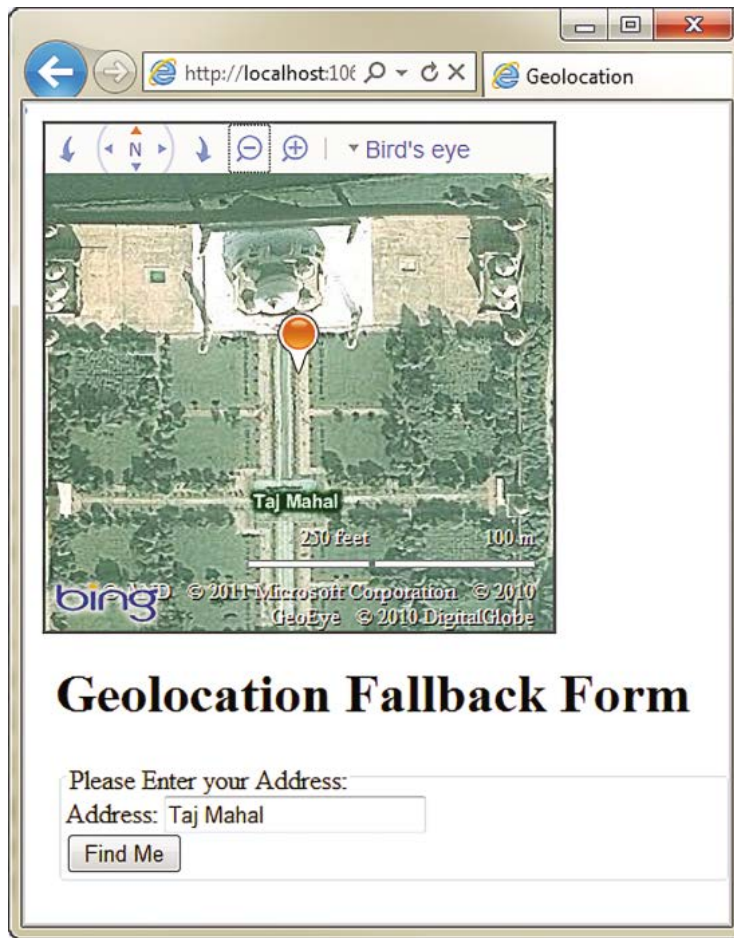


图 6 提供 Geolocation 的后备支持

在面对庞大的用户群仍然使用旧版浏览器的时候,很容易认为 HTML5 的一些高级功能对你的网站不适用。但是现在已经有了很好的解决方案,不仅能够帮助你优雅地降级,还能提升旧版浏览器的能力,让你的用户立即就能体验到 HTML5 的能力。在本文中,你看到了功能检测、Modernizr 和 polyfill,所以你可以毫不迟疑地采用 HTML5,既满足日益增长的使用最新浏览器的用户,又不会丢失使用旧版浏览器的庞大用户群。

# 微软技术支持

---

- 更多 Internet Explorer 10 对 HTML5 新特性的支持请查看：  
[http://msdn.microsoft.com/en-us/library/ie/hh673546\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/ie/hh673546(v=vs.85).aspx)
- 更多 Internet Explorer 10 对 CSS 新特性的支持请查看：  
[http://msdn.microsoft.com/en-us/library/ie/hh673536\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/ie/hh673536(v=vs.85).aspx)
- Internet Explorer 10 新增的 DOM 支持请查看：  
[http://msdn.microsoft.com/en-us/library/ie/hh673538\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/ie/hh673538(v=vs.85).aspx)
- Internet Explorer 10 Test Drive 请查看：  
<http://ie.microsoft.com/testdrive/>
- 如何构建 HTML5 游戏请查看：  
<http://buildnewgames.com/>
- 如何在 Windows 8 中固定网站的参考资料请查看：  
[http://msdn.microsoft.com/en-us/library/ie/hh781490\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/ie/hh781490(v=vs.85).aspx)
- 如何使用 Internet Explorer 10 F12 开发者工具请查看：  
<http://msdn.microsoft.com/library/dd565622.aspx>

如有任何关于 IE10/Metro IE 开发过程的问题，请发送邮件到 [IEReady\\_PRC@microsoft.com](mailto:IEReady_PRC@microsoft.com)。还可以登录 Internet Explorer 10 虚拟实验室，体验最新开发环境。（我们会将虚拟实验室地址发送到您的注册邮箱）

