1. IE8对div添加click事件，点击div的空白区域无法触发事件，解决方法：给div的css添加background属性。
2. IE8对于相邻节点的空白字符会解析为空白节点，如果使用nextSibling会取到这个节点。
3. IE8/9浏览器会对ajax请求缓存，在IE8 9下，进行Ajax请求时，若与之前请求相同，则不会再从服务器获取数据，而是直接从本地获取。在使用$.ajax时可以将其cache属性设置为false。

//$.ajaxSetup() 方法设置全局 AJAX 默认选项。  
$.ajaxSetup({ cache: false });

## [IE8对css文件的限制](http://www.cnblogs.com/fengliang/p/5015256.html)

很多人在写css时，时常把很多css样式放到一个文件中。也有些框架在上线后，能对很多css文件进行合并。这样能减少对服务器的请求次数，从而加快服务器的响应速度。在IE8中，当css的规则个数大于4096时，它会忽略后面的所有样式。因此，在开发中需要注意.

IE10以下的浏览器对css有如下限制：

（1）单个css文件的选择器的个数不得大于4096

（2）css文件个数不得大于31个

（3）最多只能嵌入 @import 规则

Internet Explorer (IE for short) imposes some CSS file limitations that may directly affect page rendering. Below is a list of known CSS file limitations still found in IE version 9 and earlier.

**- Up to 31 CSS files or <style> tags per page. 单个页面最多31个css文件**   
**- Up to 288K per CSS file (uncompressed). 每个css文件大小最大288k**   
**- Up to 4095 selectors per CSS file. 每个css文件最多4095个选择器**

If a page exceeds the first limit, it fails loading all the required CSS files. Once a CSS file hits the second or third limit, any additional style in the file will be ignored.

Resource merging, which help you reduce the number of requests made to the server, may bring the following dilemma in IE. If you are not using resource merging, you can pretty easily hit the first limit. On the other hand, if resource merging is enabled you may eventually hit the other two limits. So, to avoid hitting the limits, we recommend that you enable or disable resource merging based on the CSS file structure you are using.

Currently, the solution to automatically manage our CSS files in order to keep them within the limits is not ready yet. We created a corresponding suggestion in our Support Center (Suggestion ID: S35991,  [In IE, automatically split CSS resources if their size exceeds 288K](http://www.devexpress.com/Support/Center/p/S35991.aspx)) so that you can track our progress on it.

The first limitation usually occurs when the  **<head>** tag contains several  **<link>** and  **<style>** tags. A number of linked styles to the page can be got via the  **document.styleSheets.length** property in the JavaScript console. Using the  **document.styleSheets** collection, you will be able to determine what styles are linked to a specific page (e.g. use the FireBug console).