[10分钟学会前端调试利器——FireBug](http://www.cnblogs.com/dunitian/p/4866342.html)

**概述**

**FireBug是一个用于网站前端开发的工具，它是FireFox浏览器的一个扩展插件。它可以用于调试JavaScript、查看DOM、分析CSS、监控网络流量以及进行Ajax交互等。它提供了几乎前端开发需要的全部功能。官方网站：[www.getfirebug.com](http://www.getfirebug.com/" \t "_blank)**

**如何获取Firebug？**

**因为它是Firefox浏览器的一个扩展插件，所以首先需要下载Firefox浏览器。读者可以访问[www.mozilla.com](http://www.mozilla.com/" \t "_blank)下载并安装Firefox浏览器。安装完成后用它访问[https://addons.mozilla.org/zh-CN/firefox/collections/mozilla/webdeveloper/](https://addons.mozilla.org/zh-CN/firefox/collections/mozilla/webdeveloper/" \t "_blank)**

**进入下图所示页面。点击"添加到Firefox"，然后点击"立即安装"，最后重新启动Firefox浏览器即可完成安装。**

**主面板**

**安装完成之后，在Firefox浏览器的地址后方就会有一个小虫子的图标。单击该图标后即可展开Firebug的控制台，也可以通过快捷键<F12>来打开控制台。使用Ctrl+F12快捷键可以使Firebug独立打开一个窗口而不占用Firefox页面底部的空间。**

**从上图中可以看出，Firebug包括7个面板：**

**控制台面板：用于记录日志、概览、错误提示和执行命令行，同时也用于Ajax的调试；**

**HTML面板：用于查看HTML元素，可以实时地编辑HTML和改变CSS样式，它包括3个子面板，分别是样式、布局和DOM面板；**

**CSS面板：用于查看所有页面上的CSS文件，可以动态地修改CSS样式，由于HTML面板中已经包含了一个CSS面板，因此该面板将很少用到；**

**脚本面板：用于显示Javascript文件及其所在的页面，也可以用来显示Javascript的Debug调试，包含3个子面板，分别是监控、堆栈和断点；**

**DOM面板：用于显示页面上的所有对象；**

**网络面板：用于监视网络活动，可以帮助查看一个页面的载入情况，包括文件下载所占用的时间和文件下载出错等信息，也可以用于监视Ajax行为；**

**Cookies面板：用于查看和调整cookie(需要安装下文资源中所提到的Firecookie)。**

**控制台面板**

**1.控制台面板概览**

**此面板可以用于记录日志，也可以用于输入脚本的命令行。**

**2.记录日志**

**Firebug提供如下几个常用的记录日志的函数：**

**console.log:简单的记录日志；**

**console.debug：记录调试信息，并且附上行号的超链接；**

**console.error：在消息前显示错误图标，并且附上行号的超链接；**

**console.info：在消息前显示消息图标，并且附上行号的超链接；**

**console.warn：在纤细钱显示警告图标，并且附行号的超链接。**

**在空白的html页面中，向<body>标签中加入<script>标签，代码如下：**

[复制代码](javascript:void(0);)

复制代码

1 <script type="text/javascript">

2 console.log('this is log message');

3 console.debug('this is debug message');

4 console.error('this is error message');

5 console.info('this is info message');

6 console.warn('this is warn message');

7 </script>

复制代码

[复制代码](javascript:void(0);)

**执行代码后可以在Firebug中看到下图所示的结果，以前习惯了用alert来调试程序，然而在Firebug下可以使用console。**

**3.格式化字符串输出和多变量输出**

**这个功能类似于C语言中的语法，可以在console记录日志的方法里使用。**

**%s:字符串　　%d,%i:数字　　%f:浮点数　　%o:链接对象**

**同时，这几个函数支持多个变量。代码如下：**

[复制代码](javascript:void(0);)

复制代码

1 <script type="text/javascript">

2 var kid="孩子",count="3",man="Allen";

3 var sport1="篮球",sport2="羽毛球",sport3="网球";

4 console.log("%d个%s在玩游戏",count,kid);

5 console.log(count,"个",kid,"在玩游戏");

6 console.log("%s擅长的运动有：",man,sport1,sport2,sport3);

7 </script>

复制代码

[复制代码](javascript:void(0);)

**运行代码后效果如下图所示：**

**Firebug控制台还提供了其他功能，例如检测函数执行时间、消息分组、测试驱动、跟踪、计数以及查看Javascript概况等。更多资料可以访问[http://getfirebug.com/logging](http://getfirebug.com/logging" \t "_blank).**

**4.面板内的子菜单**

**控制台面板内有一排子菜单，分别是清除、保持、概况、全部等。**

**“清除”用于清除控制台中的内容。“保持”则是把控制台中的内容保存，即使刷新了依然还存在。“全部”则是显示全部的信息。后面的“错误”、“警告”、“消息”、“调试信息”、“Cookies”菜单则是对所有进行了一个分类。**

**“概况”菜单用于查看函数的性能。下面通过一个例子来演示，代码如下：**

[复制代码](javascript:void(0);)

复制代码

1 <button type="button" id="btn1">执行循环1</button>

2 <button type="button" id="btn2">执行循环2</button>

3 <button type="button" id="btn3">执行循环3</button>

4 <script type="text/javascript">

5 var f1=function(){

6 for(var i =0;i<1000;i++)

7 for(var j=0;j<1000;j++);

8 }

9 function f2(){

10 for(var i =0;i<1000;i++)

11 for(var j=0;j<1000;j++);

12 }

13 document.getElementById("btn1").onclick=f1;

14 document.getElementById("btn2").onclick=f2;

15 document.getElementById("btn3").onclick=function(){

16 for(var i =0;i<1000;i++)

17 for(var j=0;j<1000;j++);

18 }

19 </script>

复制代码

[复制代码](javascript:void(0);)

**打开页面，显示三个按钮：**

**打开页面后，先启用Firebug控制台面板，然后单击“概况”菜单，如下图所示：**

**从上图中可以看到，出现了一行字，“概况收集中。再次点击“概况”查看结果。”，接着，依次单击“执行循环1”、“执行循环2”、“执行循环3”三个按钮各一次，并再次单击“概况菜单”，即可看到如下图所示结果：**

**可以看到Firebug显示出了非常详细的报告。包括每个函数的函数名、调用次数、占用时间的百分比、占用时间、时间、平均时间、最小时间、最大时间以及所在的文件的行数等信息。**

**5.Ajax调试**

**控制台面板也可用于Ajax调试，在一定程度上可以取代网络面板。例如我打开一个页面，可以在Firebug控制台看到本次Ajax的Http请求头信息和服务器响应头信息。如下图，它会显示出本次使用的Ajax的GET方法、地址、耗时以及调用Ajax请求的代码行数。最重要的是有5个标签，即参数、头信息、响应、HTML、Cookies.第一个标签用于查看传递给服务器的参数；第二个标签用于查看响应头信息和请求头信息；第三个标签用于查看服务器返回的内容；第四个标签则是查看服务器返回的HTML结构；第五个标签用于查看相应的Cookies。**

**如果看不到任何信息的出现，可能是将此功能关闭了，可以单击“控制台”旁边的下拉箭头，将“显示XMLHttpRequests”前面的勾勾选上即可。**

**HTML面板**

**1.查看和修改HTML代码**

**Html面板的强大之处就是能查看和修改HTML代码，而且这些代码都是经过格式化的。下面以我的一个浏览器主页来做讲解。**

**在HTML控制台的左侧可以看到整个页面当前的文档结构，可以通过单击“+”来展开。当单击相应的元素时，右侧面板中就会显示出当前元素的样式、布局以及DOM信息。而当光标移动到HTML树中相应元素上时，上面页面中相应的元素将会被高亮显示。**

**例如，将光标移动到一个<P>标签上，显示效果如下图所示：**

**在页面中蓝色部分表示元素本身，紫色表示padding部分，黄色表示margin部分。同时可以实时地添加、修改和删除HTML节点以及属性，如下图所示。另外，单击script节点还可以直接查看脚本，此处的脚本无论是内嵌在HTML中还是外部导入的，都可以查看到。同样这也适用于<style>标签内嵌或者导入的CSS样式和动态创建HTML代码。**

**2.查看(Inspect)**

**利用查看(Inspect)功能，可以快速地寻找到某个元素的HTML结构，如图：**

**当单击"Inspect"按钮后，用鼠标在网页上选中一个元素时，元素会被一个蓝色的框框住，同时下面的HTML面板中相应的HTML树也会展开并且高亮显示。再次单击后即可退出该模式，并且底部的HTML树也保持在这个状态。通过这个功能，可以快速寻找页面内的元素，调试和查找相应代码非常方便。刷新网页后，页面显示的仍然是用Inspect选中的区域。**

**HTML面板下方的“编辑”按钮可以用于直接编辑选中的HTML代码，而后面显示的是当前元素在整个DOM中的结构路径。**

**3.查看DOM中被脚本更改的部分**

**通过JavaScript来改变样式属性的值可以完成一些动画效果。打开页面后，利用查看(Inspect)功能来选择相应的HTML代码，例如，选中“要闻”，如下图所示：**

**单击“国内”标签后，出现下图所示效果：**

**通过上图可以看出，HTML查看器会将页面上改变的内容页记录下来，并以黄色高亮表示。有了这个功能，网页的暗箱操作将彻底成为历史。我们可以使用该功能查看其它网站的动画效果是如何实现的。**

**4.查看和修改元素的样式**

**在右侧的样式面板中，展示了此元素当前所有的样式。所有的样式都可以实时地禁用和修改，如下图所示，通过在"text-align:center"前单击会出现禁用的标记，这样就可以禁用此规则。通过直接在样式value值上单击就可以修改。**

**单击“布局”面板即可看到此元素具体的布局属性，这是一个标准的盒模型。通过“布局”面板，可以很容易地看到元素的偏移量、外边距、边框、内边距和元素的高度、宽度等信息，如下图所示：**

**5.查看DOM的信息**

**单击“DOM”面板后可以看到此元素的详细的DOM信息以及函数和事件，如下图所示：**

**CSS、DOM和网络面板**

**这3个面板相对于前面2个面板比较次要，CSS和DOM面板与HTML面板中右侧的面板功能相似，但不如HTML面板灵活，因此一般使用得很少。**

**CSS面板特有的一个功能就是可以分别详细查看页面中内嵌以及动态导入的样式。如下图所示：**

**单击CSS面板后就可以分别查看相应的样式。此处展开的样式都是经过格式化的，适合于学习CSS的代码格式和规范。**

**而在网络面板中，相对有一些强大的功能，例如打开某个网站首页，Firebug显示效果如下图所示：**

**该页面可以监视每一项元素的加载情况，包括脚本，图片等的大小以及加载用时等，对于页面优化有着极其重要的意义。**

**此外，顶部还可以分类查看元素的HTML、CSS、JS等的加载情况，使分析更加灵活。**

**脚本面板**

**脚本面板不仅可以查看页面内的脚本，而且还有强大的调试功能。**

**在脚本面板的右侧有“监控”、“堆栈”和“断点”三个面板，利用Firebug提供的设置断掉的功能，可以很方便地调试程序，如下图所示：**

**1.静态断点**

**例如test.html文件，代码如下：**

[复制代码](javascript:void(0);)

复制代码

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

2 <html xmlns="http://www.w3.org/1999/xhtml">

3 <head>

4 <script type="text/javascript">

5 function doSomething(){

6 var lab = document.getElementById('messageLabel');

7 arrs=[1,2,3,4,5,6,7,8,9];

8 for(var arr in arrs){

9 lab.innerHTML+=arr+"<br />"

10 }

11 }

12 </script>

13 </head>

14 <body>

15 <div>

16 <div id="messageLabel"></div>

17 <input type="button" value="Click Here" onClick="doSomething();"/>

18 </div>

19 </body>

20 </html>

复制代码

[复制代码](javascript:void(0);)

**运行代码后可以看到下图所示效果。图中加粗并有颜色的行号表示此处为JavaScript代码，可以在此处设置断点。比如在第6行这句代码前面单击一下，那它前面就会出现一个红褐色的圆点，表示此处已经被设置了断点。此时，在右侧断点面板的断点列表中就出现了刚才设置的断点。如果想暂时禁用某个断点，可以在断点列表中去掉某个断点的前面的复选框中的勾，那么此时左侧面板中相应的断点就从红褐色变成了红灰褐色了。**

**设置完断点之后，我们就可以调试程序了。单击页面中的“Click Here”按钮，可以看到脚本停止在用谈黄色底色标出的那一行上。此时用鼠标移动到某个变量上即可显示此时这个变量的value。显示效果如下：**

**此时JavaScript内容上方的四个按钮已经变得可用了。它们分别代表“继续执行”、“单步进入”、“单步跳过”和“单步退出”。**

**继续执行<F8>：当通过断点来停止执行脚本时，单击<F8>就会恢复执行脚本。**

**单步进入<F11>：允许跳到页面中的其他函数内部。**

**单步跳过<F10>：单击<F10>来直接跳过函数的调用即跳到return之后。**

**单步退出<shift+F11>：允许恢复脚本的执行，直到下一个断点为止。**

**单击“单步进入”按钮，代码会跳到下一行，显示效果如下：**

**从上图可以看出，当鼠标移动到“lab”变量上时，就可以显示出它的内容是一个DOM元素，即“div#messageLabel”。**

**此时将右侧面板切换到“监控”面板，这里列出了几个变量，包括“this”指针的指向以及“lab”变量。单击“+”可以看到详细的信息。显示如下：**

**2.条件断点**

**在“lab.innerHTML+=arr+"<br />"”这行代码前面的序号上单击鼠标右键，就可以出现设置条件断点的输入框。在该框内输入“arr==5”，然后回车确认，显示效果如下：**

**最后单击页面的“Click Here”按钮。可以发现，脚本在“arr==5”这个表达式为真时停下了，因此“5”以及之后的数字没有显示到页面中。下图分别是正常效果和设置了条件断点之后的显示效果对比：**

**资源**

**快捷键：按<F12>可以快速开启Firebug，如果想获取完整的快捷键列表，可以访问[http://getfirebug.com/wiki/index.php/Keyboard\_and\_Mouse\_Shortcuts](http://getfirebug.com/wiki/index.php/Keyboard_and_Mouse_Shortcuts" \t "_blank).**

**问题：如果安装过程中遇到了困难，可以查看Firebug的Q&A，网址为[http://getfirebug.com/wiki/index.php/FAQ](http://getfirebug.com/wiki/index.php/FAQ" \t "_blank).**

**Firebug插件：Firebug除了本身强大的功能之外，还有基于Firebug的插件，它们用于扩充Firebug的功能。比如Google公司开发Page Speed插件，开发人员可以使用它来评估他们网页的性能，并获得有关如何改进性能的建议。Yahoo公司开发的用于检测页面整体性能的YSlow和用于调试PHP的FirePHP。还有用于调试Cookie的Firecookie等。**

**总结**

**通过本文的学习，读者可以掌握Firebug的基本功能。Firebug已经逐渐成为一个调试平台，而不仅仅是一个简单的Firefox扩展插件。学好Firebug能给以后的学习和工作提供一定的帮助。**

**参考文献：《锋利的JQuery(第2版)》**

**原文：http://www.cnblogs.com/allenben/p/4757274.html**