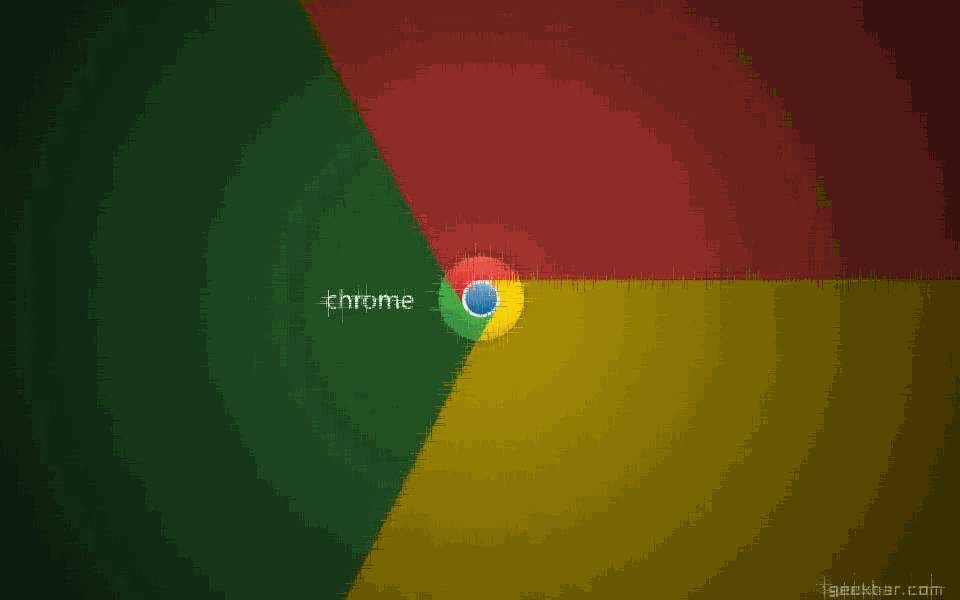
超完整Chrome浏览器客户端调试大全（一）

原创 2016-12-15 Cayley [极客学院官网](http://mp.weixin.qq.com/s?__biz=MjM5ODE0MTM1MA==&mid=2650227967&idx=1&sn=e3d9ee96c47007dc1af1bf72eba13ec8&chksm=becca83689bb21200539ecc409a81fece0378ddf86062751a2c2c488cf2978c93eba1f0b674f&mpshare=1&scene=5&srcid=1215KLJvdg7szyI7btymTguL##)





本文来自极客投稿，原作者授权发布。

欢迎极客投稿，加入小学妹朋友圈共同学习哟～

文章链接｜http://www.igeekbar.com/igeekbar/post/156.htm?from=singlemessage&isappinstalled=0

“工欲善其事，必先利其器”。没错，这句话个人觉得说的特别有道理，举个例子来说吧，厉害的化妆师都有一套非常专业的刷子，散粉刷负责定妆，眼影刷负责打眼影，各司其职，有了专业的工具才能干专业的事，这个灵感要来源于之前我想买化妆品时，店里的海报标语。

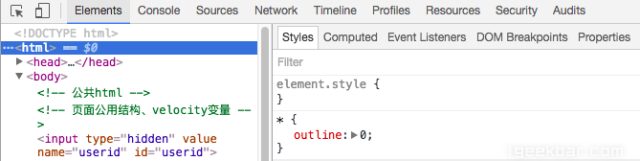
由此联想到，如果你想在某个事情上做好，并且做的专业，那么你一定要把你的工具用好，这样才能事半功倍，**我见过很多师兄师姐，写了很多代码，能够很快的完成工作，能够处理很多复杂的业务逻辑，但是对于浏览器的调试掌握的并不全面和深入。**

说说本姑娘吧，我的编程调试起源于自学滴前端课程，因为学习的时候看的都是基础的教学视频，对于调试也只是掌握了alert和console……请大家别笑话，认真看完再说话。试问谁当初入门时候不是走的这条路呢？

当你不再限于做静态页面，古老而经典的调试方式肯定不能帮你完成日常调试。日后我进入到了大公司去实习，才真正开始接触专业开发业务，开始跟着师兄和师傅一起上路，那时我才有了“js断点调试“的意识，开始一步步调试我的js代码~

**下面总结一下一些常用调试方法，这些方法能让开发的工作顺利并且高效，**这里小女子拿出来总结一下，与各位程序猿同仁分享一下 ~ （此处应有掌声…… ^\_^）

**先来认识一下这些按钮**

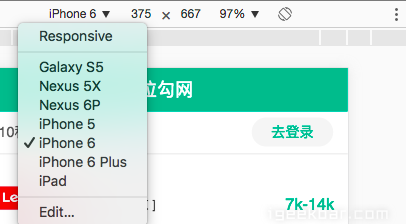


先来看这张图最上头的一行是一个功能菜单，每一个菜单都有它相应的功能和使用方法，依次从左往右来看

箭头按钮

* 用于在页面选择一个元素来审查和查看它的相关信息，当我们在Elements这个按钮页面下点击某个Dom元素时，箭头按钮会变成选择状态。

设备图标

* 点击它可以切换到不同的终端进行开发模式，移动端和pc端的一个切换，可以选择不同的移动终端设备，同时可以选择不同的尺寸比例，chrome浏览器的模拟移动设备和真实的设备相差不大，是非常好的选择。
* 
* 可选择的适配

Elements

* 功能标签页：用来查看，修改页面上的元素，包括DOM标签，以及css样式的查看，修改，还有相关盒模型的图形信息。
* 下图我们可以看到当我鼠标选择id 为lg\_tar的div元素时，右侧的css样式对应的会展示出此id 的样式信息。
* 此时可以在右侧进行一个修改，修改即可在页面上生效， 灰色的element.style样式同样可以进行添加和书写。
* 唯一的区别是，在这里添加的样式是添加到了该元素内部，实现方式即：该div元素的style属性，这个页面的功能很强大，在我们做了相关的页面后，修改样式是一块很重要的工作，细微的差距都需要调整，但是不可能说做到每修改一点即编译一遍代码，再刷新浏览器查看效果，这样很低效，一次性在浏览器中修改之后，再到代码中进行修改。
* 
* 对应的样式
* 盒模型信息
* 同时，当我们浏览网站看到某些特别炫酷的效果和难做的样式时候，打开这个功能，我们即可看到别人是如何实现的，学会它这知识就是你的了，仔细钻研也会有意想不到的收获

Console控制台

* 用于打印和输出相关的命令信息，其实console控制台除了我们熟知的报错，打印console.log信息外，还有很多相关的功能，下面简单介绍几个：
* a：一些对页面数据的指令操作，比如打断点正好执行到获取的数据上，由于数据都是层层嵌套的对象，这个时候查看里面的key/value不是很方便，即可用这个指令开查看，obj的json string 格式的key/value，我们对于数据里面有哪些字段和属性即可一目了然
* 其他功能
* b： 除了console.log还有其他相关的指令可用
* console也有相关的API

Sources

 js资源页面：这个页面内我们可以找到当然浏览器页面中的js 源文件，方便我们查看和调试。

在我还没有走出校园时候，我经常看一些大站的js代码，那时候其实基本都看不懂，但是最起码可以**看看人家的代码风格，人家的命名方式**，所有的代码都是压缩之后的代码，我们可以点击下面的{}大括号按钮将代码转成可读格式

Sources Panel 的左侧分别是 Sources 和 Content scripts和Snippets

对应的源代码

格式化的代码

关于打断点调试的内容下面介绍，先来说一些，其他平时基本没人用但是很有用的小点。

比如当我们想不起某个方法的具体使用时候，会打开控制台随意写一些测试代码，或者想测试一下刚刚写的方法是否会出现期待的样子，但是控制台一打回车本想换行但是却执行刚写的半截代码。

所以推荐使用**Sources**下面的左侧的Sinppets代码片段按钮，这时候点击创建一个新的片段文件，写完测试代码后把鼠标放在新建文件上run，再结合控制台查看相关信息（**新建了一个名叫：app.js的片段代码，在你的项目环境页面内，该片段可执行项目内的方法**）

自己书写的片段

Content scripts 是 Chrome 的一种扩展程序，它是按照扩展的ID来组织的，这些文件也是嵌入在页面中的资源，这类文件可以读写和操作我们的资源，需要调试这些扩展文件，则可以在这个目录下打开相关文件调试，但是几乎我们的项目还没有相关的扩展文件，所以啥也看不到，平时也不需要关心这块。

无结果

Network

* 网络请求标签页：可以看到所有的资源请求，包括网络请求，图片资源，html,css，js文件等请求，可以根据需求筛选请求项，一般多用于网络请求的查看和分析，分析后端接口是否正确传输，获取的数据是否准确，请求头，请求参数的查看
* 所有的资源
* 以上我选择了All，就会把该页面所有资源文件请求下来，如果只选择XHR 异步请求资源，则我们可以分析相关的请求信息
* 请求的相关信息

打开一个Ajax异步请求，可以看到它的请求头信息，是一个POST请求，参数有哪些，还可以预览它的返回的结果数据，这些数据的使用和查看有利于我们很好的和后端工程师们联调数据，也方便我们前端更直观的分析数据。

预览请求的数据

Timeline

* 可以显示JS执行时间、页面元素渲染时间，不做过多介绍

Profiles

     可以查看CPU执行时间与内存占用，不做过多介绍

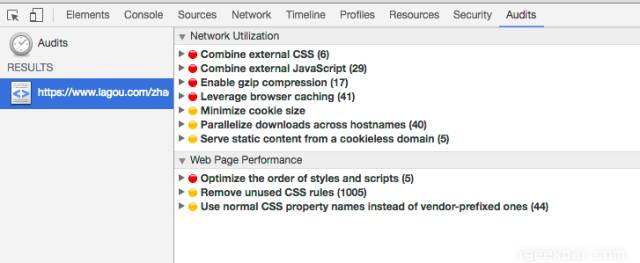
Resources

   会列出所有的资源，以及HTML5的Database和LocalStore等，你可以对存储的内容编辑和删除，不做过多介绍

Security

* 可以告诉你这个网站的安全性，查看有效的证书等

Audits

* 可以帮你分析页面性能，有助于优化前端页面，分析后得到的报告
* 
* **—— END ——**

