**Console API**

当打开 firebug (也包括 Chrome 等浏览器的自带调试工具)，window 下面会注册一个叫做 console 的对象，它提供多种方法向控制台输出信息，供开发人员调试使用。下面是这些方法的一个简单介绍，适时地运用它们，对于提高开发效率很有帮助。

**console.log(object[, object, ...])**  
使用频率最高的一条语句：向控制台输出一条消息。支持 C 语言 printf 式的格式化输出。当然，也可以不使用格式化输出来达到同样的目的：

**var** animal='frog', count=10; console.log("The %s jumped over %d tall buildings", animal, count); console.log("The", animal, "jumped over", count, "tall buildings");

**console.debug(object[, object, ...])**  
向控制台输出一条信息，它包括一个指向该行代码位置的超链接。

**console.info(object[, object, ...])**  
向控制台输出一条信息，该信息包含一个表示“信息”的图标，和指向该行代码位置的超链接。

**console.warn(object[, object, ...])**  
同 info。区别是图标与样式不同。

**console.error(object[, object, ...])**  
同 info。区别是图标与样式不同。error 实际上和 throw new Error() 产生的效果相同，使用该语句时会向浏览器抛出一个 js 异常。

**console.assert(expression[, object, ...])**  
断言，测试一条表达式是否为真，不为真时将抛出异常（断言失败）。

**console.dir(object)**  
输出一个对象的全部属性（输出结果类似于 DOM 面板中的样式）。

**console.dirxml(node)**  
输出一个 HTML 或者 XML 元素的结构树，点击结构树上面的节点进入到 HTML 面板。

**console.trace()**  
输出 Javascript 执行时的堆栈追踪。

**console.group(object[, object, ...])**  
输出消息的同时打开一个嵌套块，用以缩进输出的内容。调用 console.groupEnd() 用以结束这个块的输出。

**console.groupCollapsed()**  
同 console.group(); 区别在于嵌套块默认是收起的。

**console.time(name)**  
计时器，当调用 console.timeEnd(name);并传递相同的 name 为参数时，计时停止，并输出执行两条语句之间代码所消耗的时间（毫秒）。

**console.profile([title])**  
与 profileEnd() 结合使用，用来做性能测试，与 console 面板上 profile 按钮的功能完全相同。

**console.count([title])**  
输出该行代码被执行的次数，参数 title 将在输出时作为输出结果的前缀使用。

**console.clear()**  
清空控制台

**命令行**

控制台的输出面板右边，是控制台的输入面板（Chrome 调试工具对应为下方），在这里除了可以运行常规的 javascript 代码，还内置了相当数量的命令行可以辅助我们的调试工作，下面是一些常用命令行的简单介绍。

**$(id)**  
返回一个给定 id 的元素。

**$$(selector)**  
返回给定的 css 选择器匹配到的一组元素。

**$x(xpath)**  
返回给定的 XPath 表达式匹配到的一组元素。

**$0**  
在 HTML 面板中选中的元素。

**$1**  
上一次在 HTML 面板中选中的元素。

**$n(index)**  
访问最近 5 个被选中过的元素，index 的范围： 0 – 4。

**dir(object)**  
同 console.dir(object)。

**dirxml(node)**  
同 console.dirxml(node)。

**clear()**  
同 console.clear()。

**inspect(object[, tabName])()**  
在合适的（或一个指定的） tab 中检视一个对象。

**keys(object)**  
返回一个对象的所有属性的键。

**values(object)**  
返回一个对象的所有属性的值。

**debug(fn)**  
在函数第一行添加一个断点，使用 undebug(fn) 移除断点。

**monitor(fn)**  
开启一个函数的调用日志，使用 unmonitor(fn) 关闭该功能。非常有用的一个命令，但是它似乎并没有很好地工作。

**monitorEvents(object[, types])**  
开启一个元素的某个事件（或所有事件）被触发时的日志记录。用例如下：

monitorEvents($0,['click'])

上面的命令行被执行后，将开启当前在 HTML 面板中被选中元素的 click 事件监控，一旦这个元素的 click 事件被触发，事件对象将会在控制台输出。如果不指定第二个参数，将对所有事件进行记录。

**profile([title])**  
同 console.profile([title])