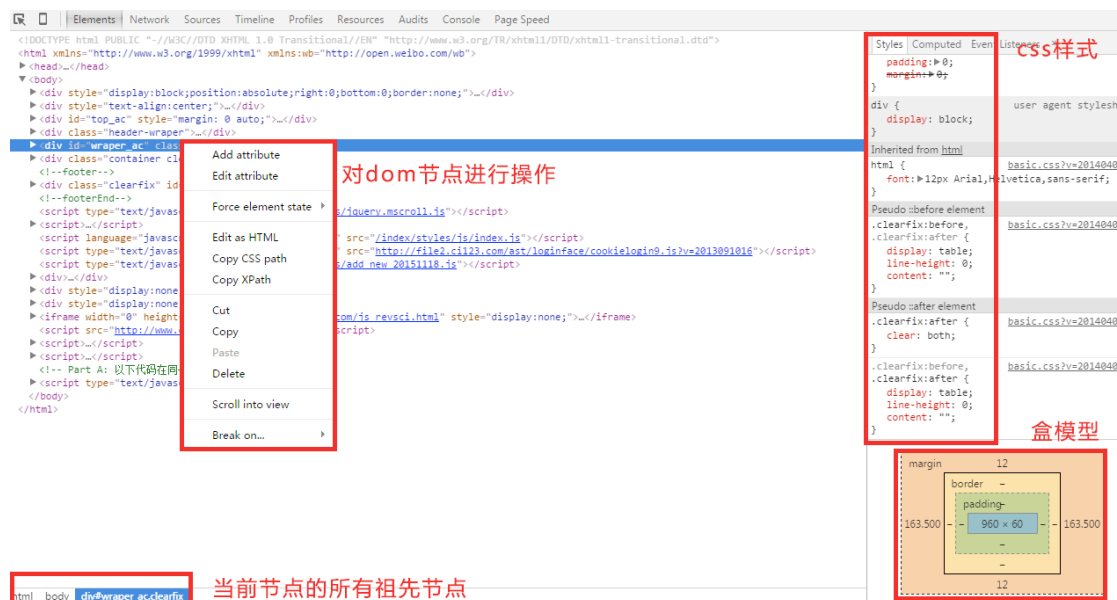


Chrome 调试功能

1.Elements 面板——DOM

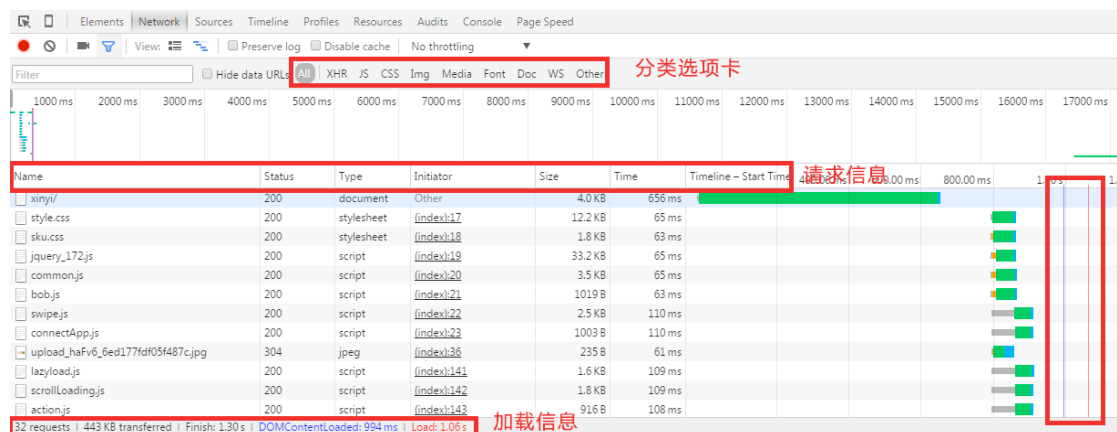


左侧展示了 dom 树，在某个 dom 节点上点击右键会出现操作菜单，常用功能：增加属性、编辑属性、以 html 形式编辑代码、复制 css 路径、复制 Xpath 路径、剪切/复制/粘贴/删除代码、设置断点。

左侧底部显示当前节点的所有祖先节点。

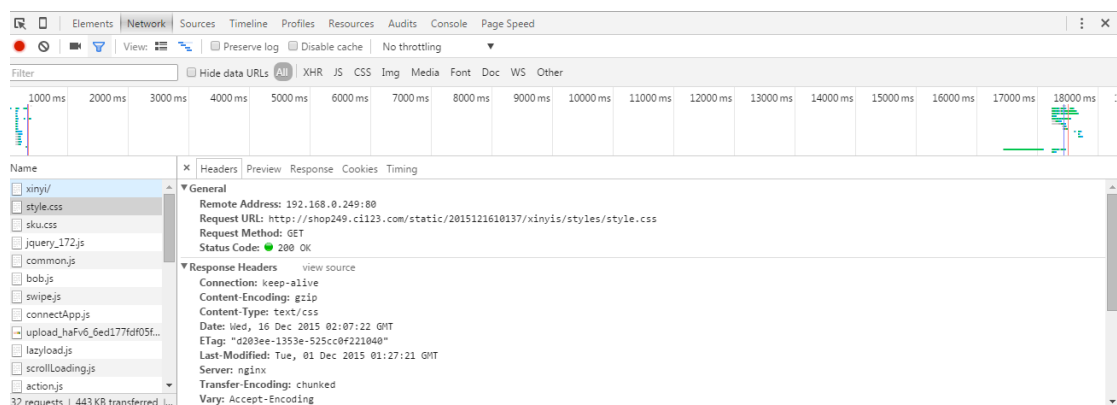
右侧面板主要包括 5 个功能块。①Styles：按权重从大到小顺序显示所选元素的 css 样式，可以进行修改，页面会实时响应变化；下方是盒模型图；②Computed：所选元素的最终样式，只可读不可修改；③Event Listeners：显示绑定到当前元素的事件监听函数；④Dom Breakpoints：显示 dom 中设置的断点，便于调试；⑤Properties：显示元素作为 dom 对象的各个属性。

2.Network 面板——请求加载



上图分为 7 列，从左到右分别是 Name、Status、Type、Initiator、Size、Time 和 Timeline。①Name 表示加载的文件名；②Status 表示状态码（200 为请求成功，304 表示从缓存读取）；③Type 表示文件的 MIME Type 的类型；④Initiator 表示发出这个文件请求的发出者；⑤Size 表示文件大小；⑥Time 表示每个请求的总时长；⑦Timeline 显示了请求和加载情况（Timeline 中鼠标移到每个时间条上的时候，会显示整个请求的详细信息）。通过它可以看到哪些请求是同时发出的，哪些请求被阻塞了。Timeline 中有一条蓝线和一条红线，蓝线表示 DOM Content Loaded 事件触发的时间，红线表示 Window onload 事件触发的时间。

点击某个请求会显示该请求的具体信息：

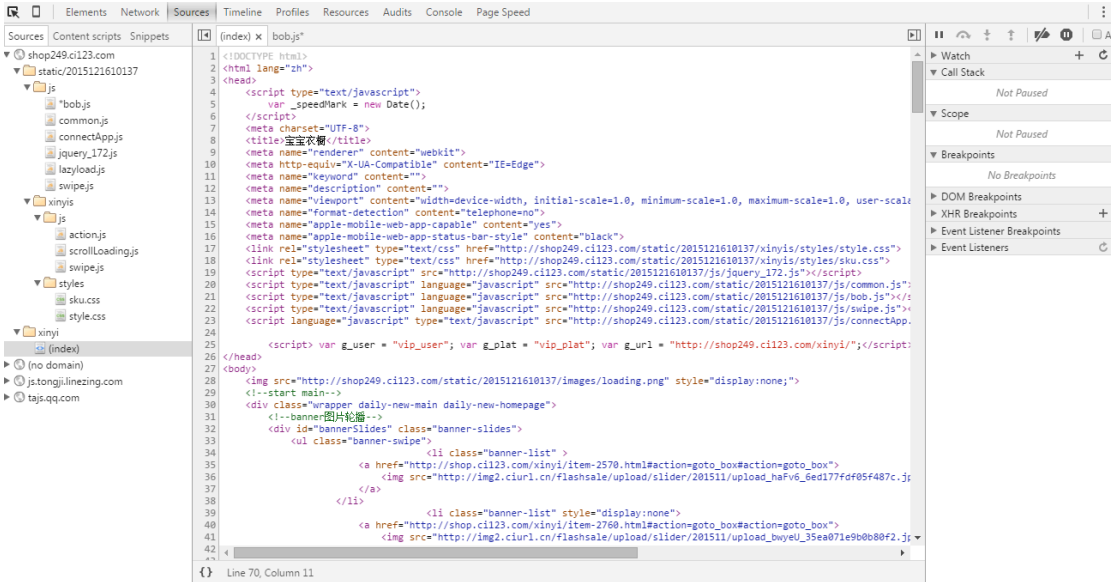


顶部的选项卡将请求的文件进行了分类，便于查看，如：XHR、JS、CSS、Img、Media、Font、Doc、Websockets 和 Other。

底部显示了网页请求的整体加载信息，如：请求个数、传输大小、dom 结构加载时长、

页面加载完成总时长。

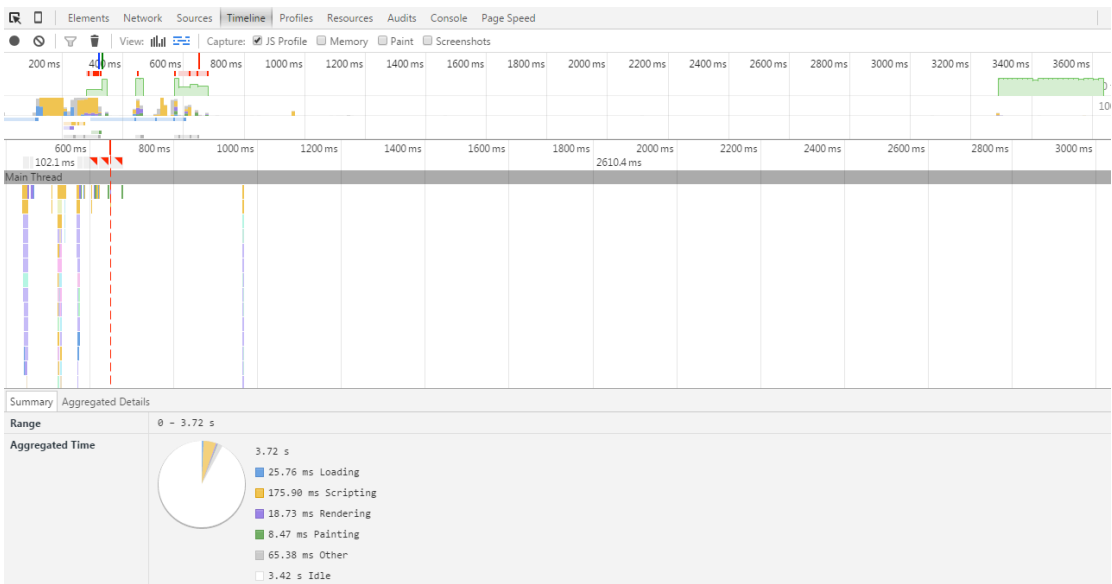
3.Sources 面板——样式脚本



显示所有加载的 html、css 和 js 资源，可以实时编辑 css 或 js 文件。右侧显示监听、事件绑定以及断点信息。常用于设置断点来调试 js。

设置断点调试 js 可参考：<http://segmentfault.com/a/1190000004123527>

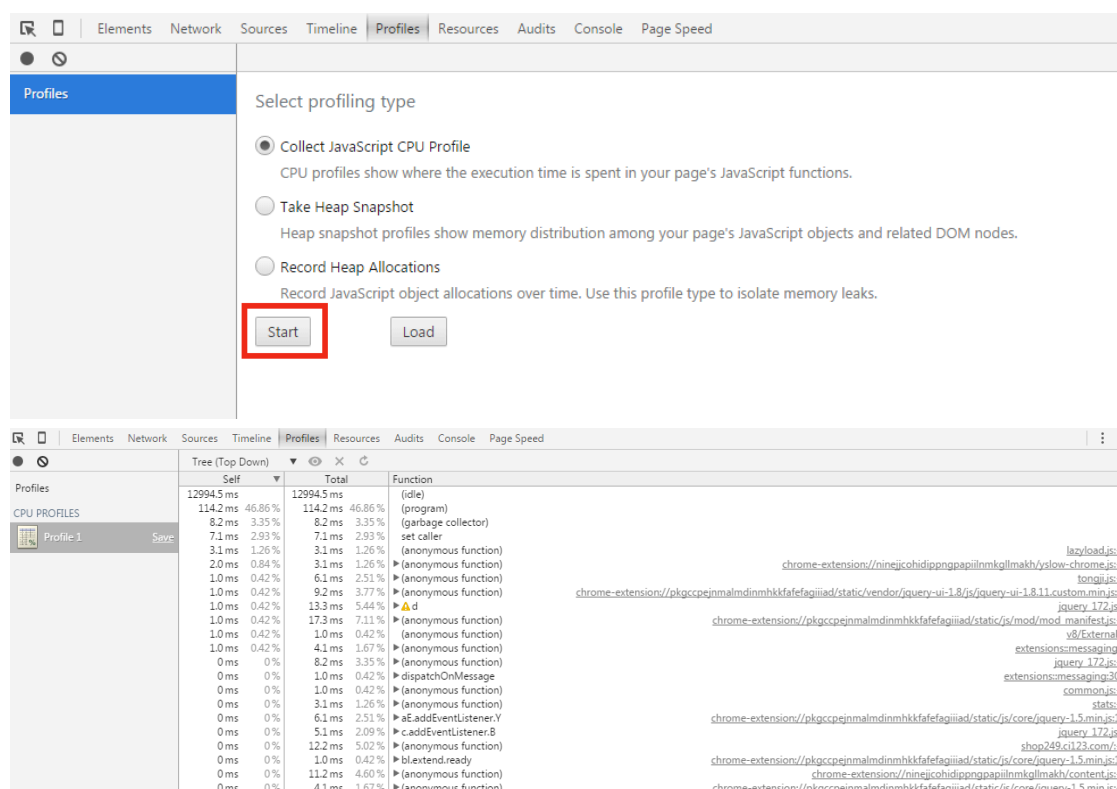
4.Timeline 面板——页面加载



Timeline 跟 Network 有类似的地方，他们都是按照页面的加载时间来统计数据的，不过 Timeline 统计的数据侧重点不一样。Timeline 主要统计了三个数据：Loading，Scripting，Rendering。

点击时间线上的任意位置，可以查看在这个点的详细信息。

5.Profiles 面板——CPU 信息



The screenshot shows the Chrome DevTools Profiles panel. The top section is titled 'Select profiling type' and has three radio buttons: 'Collect JavaScript CPU Profile' (selected), 'Take Heap Snapshot', and 'Record Heap Allocations'. Below these are 'Start' and 'Load' buttons. The 'Start' button is highlighted with a red rectangle. The bottom section shows a table of recorded functions.

Self	Total	Function
12994.5 ms	12994.5 ms	(idle)
114.2 ms	114.2 ms	(program)
8.2 ms	8.2 ms	(garbage collector)
7.1 ms	7.1 ms	set caller
3.1 ms	3.1 ms	(anonymous function)
2.0 ms	3.1 ms	(anonymous function)
1.0 ms	6.1 ms	(anonymous function)
1.0 ms	9.2 ms	(anonymous function)
1.0 ms	13.3 ms	(anonymous function)
1.0 ms	17.3 ms	(anonymous function)
1.0 ms	1.0 ms	(anonymous function)
1.0 ms	4.1 ms	(anonymous function)
0 ms	8.2 ms	(anonymous function)
0 ms	1.0 ms	dispatchOnMessage
0 ms	1.0 ms	(anonymous function)
0 ms	3.1 ms	(anonymous function)
0 ms	6.1 ms	a.addEventListener.Y
0 ms	5.1 ms	c.addEventListener.B
0 ms	12.2 ms	(anonymous function)
0 ms	1.0 ms	bl.extend.ready
0 ms	11.2 ms	(anonymous function)
0 ms	4.1 ms	(anonymous function)

Profile 记录的主要是 CPU 和内存占用的信息。点击 start 按钮，开始记录 CPU 使用信息，刷新页面，等页面加载完成之后点击 stop 按钮，完成录制。

上图的左边两列是时间，第一列是 self，第二列是 total，self 表示函数自己的运行时间，不包含调用其他函数的时间，而 total 表示这个函数运行的总时间。第三列是运行的函数。通过比较时间的百分比可以找出性能差的函数进行优化。

6.Resources 面板——所有资源信息

Name	Value	Domain	Path	Expires / ...	Size	HTTP	Secure	First-Party
PHPSESSID	1hd5fsgdovoo01qj8rouaoa5	shop249.c...	/	Session	35			
lzstat_ss	778501129_5_1450260424_3593054j27656428_3_145026646...	ci123.com	/	Session	70			
lzstat_uv	3379332666915857250j3604478@3593054@3413434	ci123.com	/	2025-12-...	52			
pgv_pi	8576913408	ci123.com	/	2038-01-...	17			
pgv_si	s2646127616	ci123.com	/	Session	17			
vip_plat_dev	1%2C1%2C0.0%2C	shop249.c...	/	2016-11-...	26			
vip_user_dev	745788%2C1%2C37856688%2Cmu9848259%2C%E6%89%8B...	shop249.c...	/	2016-12-...	192			

Resources 显示所有加载的资源，除了 Sources 面板中的 html/css/js 文件外，还包括图片文件；并且还可以查看本地存储、会话存储、cookies 信息、缓存等。

7.Audits 面板——性能检测优化

Select audits to run

☒ Select All

☒ Network Utilization

☒ Web Page Performance

☐ Audit Present State

☐ Reload Page and Audit on Load

Run **Clear**

RESULTS

Network Utilization

- Combine external JavaScript (9)**
There are multiple resources served from same domain. Consider combining them into as few files as possible.
9 JavaScript resources served from shop249.ci123.com.
- Enable gzip compression (1)**
- Leverage browser caching (31)**
- Minimize cookie size**
- Serve static content from a cookieless domain (6)**
- Specify image dimensions (9)**

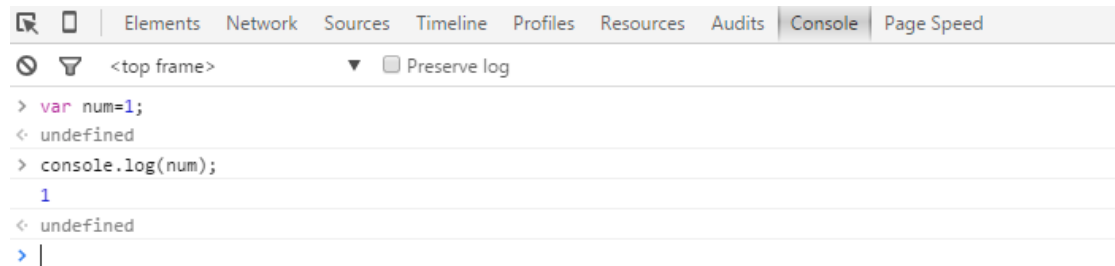
Web Page Performance

- Optimize the order of styles and scripts (1)**
- Remove unused CSS rules (628)**
- Use normal CSS property names instead of vendor-prefixed ones (49)**

Audits 面板的作用于 pagespeed、Yslow 类似，用于检测页面的性能，提出优化建议。

点击 Run 按钮，即可得到页面性能的检测结果和优化建议。

8.Console 面板——控制台输出



Console 可以用于输出一些信息，常用于调试。支持的一些方法函数如下：

- `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。
- `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.error(object[, object, ...])`在控制台中输出一条错误信息。 🗯
- `console.warn(object[, object, ...])`在控制台中输出一条警告信息。
- `console.assert([, 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.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。