



## 免费阅读

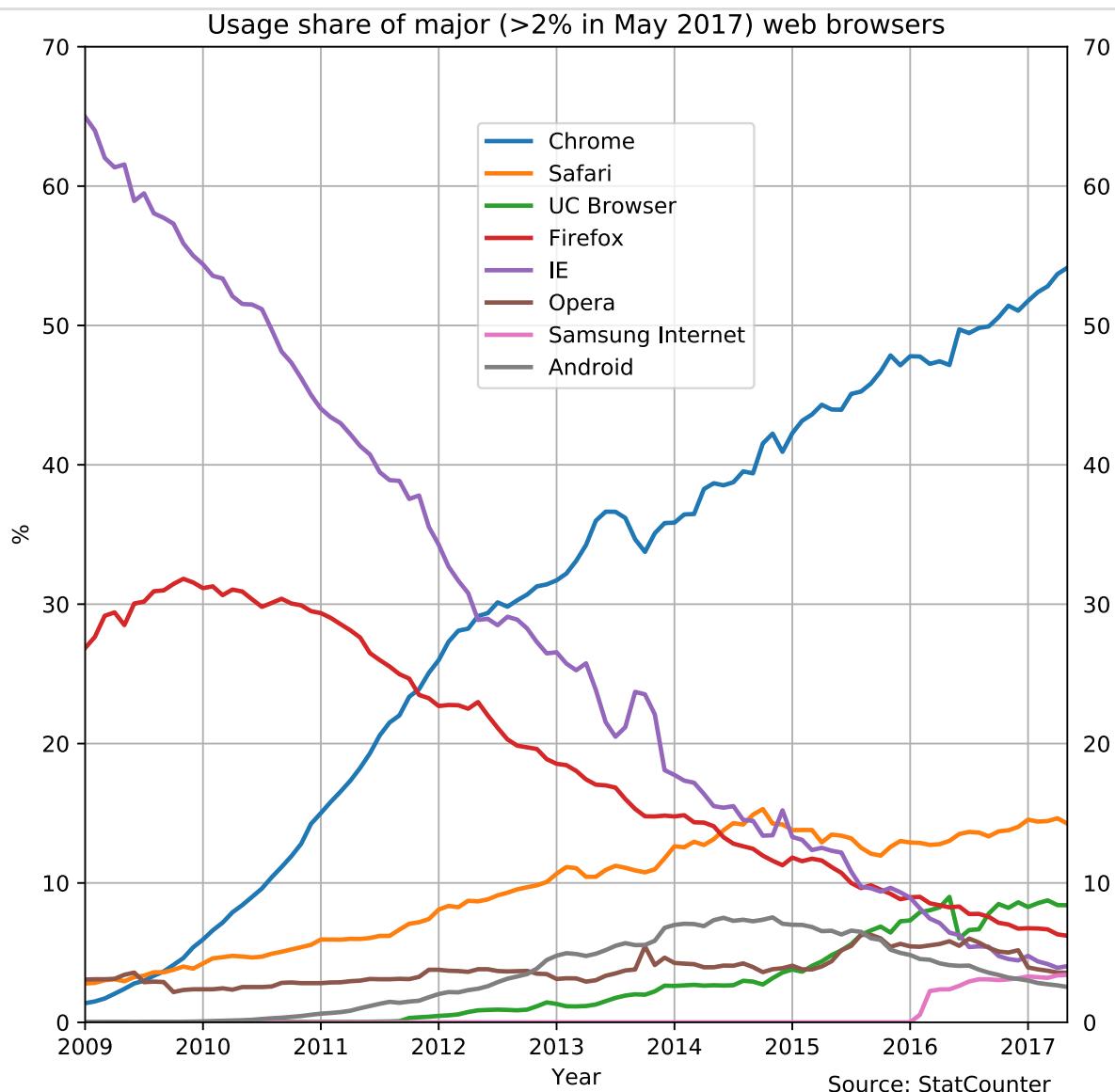
如果你不想购买，小册支持在线免费阅读，因为小册的本意是为了帮助更多的人了解 Chrome Devtools，免费阅读：[仓库地址](#)，欢迎 star 支持。

## 从 Chrome 说起

---

谷歌浏览器（通常简称为 `Chrome`）是由谷歌开发的网络浏览器。它于 2008 年首次针对 `Microsoft Windows` 发布，后来移植到 `Linux`，`macOS`，`iOS` 和 `Android`。浏览器也是 `Chrome OS` 的主要组件，它可以作为 `Web` 应用的平台。[Chrome-wikipedia](#)

浏览器的市场天下三分，`Chorme`，`Safari` 和 `FireFox`，从 2008 年 `Chrome` 横空出世以来，如今已经一家独大占据了半壁江山：



对于大部分人来说，`Chrome` 可能只是个浏览器，但是对于开发人员来说，它更是一个强大无比的工具，为了方便开发人员调试代码，主流的浏览器都内置了 `DevTools`，所以无论你是前端还是后端，掌握 `Chrome` 的调试技巧意味着效率直接的提高。而这本小册要介绍的，就是 `Chrome-DevTools` 的使用技巧。

## DevTools 简介

本段内容引用于 [Chrome DevTools](#) 说明 熟悉的同学可以跳过，也可以选择跳转到原链接访问，这里是为了给没有接触过 `chrome-devtools` 的同学一些基础概念。

## 打开 Chrome 开发者工具

- 在 `Chrome` 菜单中选择 `更多工具 > 开发者工具`



· 使用元素面板 **CTRL + SHIFT + I ( Mac: ⌘ + ⌥ + I )**

## 了解面板

我将从以下 8 个面板来讲述面板内容：

1. 元素面板
2. 控制台面板
3. 源代码面板
4. 网络面板
5. 性能面板
6. 内存面板
7. 应用面板
8. 安全面板

### 1. 元素面板

使用元素面板可以自由的操作 **DOM** 和 **CSS** 来迭代布局和设计页面。

- 检查和调整页面
- 编辑样式
- 编辑 **DOM**



## 2. 控制台面板

在开发期间，可以使用控制台面板记录诊断信息，或者使用它作为 `shell` 在页面上与 `JavaScript` 交互。

- 使用控制台面板
- 命令行交互



### 3. 源代码面板

在源代码面板中设置断点来调试 `JavaScript`，或者通过 `Workspaces`（工作区）连接本地文件来使用开发者工具的实时编辑器

- 断点调试
- 调试混淆的代码
- 使用开发者工具的 `Workspaces`（工作区）进行持久化保存



## 4. 网络面板

使用网络面板了解请求和下载的资源文件并优化网页加载性能。

- 网络面板基础
- 了解资源时间轴
- 网络带宽限制



## 5. 性能面板

使用时间轴面板可以通过记录和查看网站生命周期内发生的各种事件来提高页面的运行时性能。



## 6. 内存面板

如果需要比时间轴面板提供的更多信息，可以使用“配置”面板，例如跟踪内存泄漏。

- [JavaScript CPU 分析器](#)
- [内存堆区分析器](#)



## 7. 应用面板

使用资源面板检查加载的所有资源，包括 IndexedDB 与 Web SQL 数据库，本地和会话存储，cookie，应用程序缓存，图像，字体和样式表。

- 管理数据



## 8. 安全面板

使用安全面板调试混合内容问题，证书问题等等。

- 安全



输入评论...

lees5488

感谢作者

18天前



回复

30学前端

作者太强了， 笔芯

1月前



回复

lovelyJason Lv1 前端工程师 @ 某知名...

回一下我谢谢,想看看console篇以外的部分

4月前



回复

dendoink14454 Lv4 (作者) 摸鱼小能手

现在已经开放免费阅读的版本，并且对内容做了更新。

1月前

过气学长 软件测试工程师

作者好用心， 价格还那么美丽。

4月前



回复

dendoink14454 Lv4 (作者) 摸鱼小能手

现在已经开放免费阅读的版本，并且对内容做了更新。

1月前

lovelyJason Lv1 前端工程师 @ 某知名...

为什么这个6到12章全是console篇啊,你这里的8个面板好多没讲到,sources断点调试这么重要的没

4月前



回复

specialApe\_特种猿 JavaCoder @ ifinan

厉害

8月前



回复

negrochn Lv1

“或者通过 Workspaces（工作区）连接本地文件来使用开发者工具的实时编辑器”，这句话怎么理解呢？？

9月前



回复

chor Lv2 前端 @ 在读



SakuraMD Android @ 保密

都是干货，感觉又学到了很多。

9月前



回复

zhongchun·jī 巨小的公司

CMD + Opt + J 也可打开。

9月前



回复

音乐和茶 Lv2 前端 @ 面向本心开发

感谢分享！

10月前



回复

Harhao Lv2 高级切图仔 @ 摸鱼

调试定位问题所在在开发工作还是挺重要的

10月前



回复

六个周 Lv2 全栈工程师 @ 阿里巴...

留下足迹，这条评论可以存在一万年吗。

10月前



回复

小乔流水乔 Lv1 前端开发

有啥意义吗

10月前

xiamihaozi

戳中笑点

1月前

gnat35 Lv1 前端开发

十分感谢

1年前



回复

王州

很赞，一直想搞明白这个，无奈chrome英文文档看的难受

1年前



回复

Mr张先生 Lv1 前端开发

[🔗 developers.google.com/web/tools/chro...](https://developers.google.com/web/tools/chrome-devtools/), 感觉和这里的文档相同

10月前

东坡肉里没有苏东坡 Lv1 攻城狮



D冰城

赞

1年前

1

回复

---

StevenLikeWaterm... Lv2 web前端

绝对不止0.01啊，快给我赞助二维码！我要donate！！

1年前

17

回复

---

一知特立独行的猫 前端 @ 北京某大厂

学习了，6666

1年前

1

回复

---

CC酱45766

一直在找一个能完整的介绍chrome控制台的书籍，谢谢作者大大

1年前

1

回复

---

eukire Ctrl+C/V熟练工 @ 某...

我们用的不是同一个Chrome...

1年前

1

回复

---

南客同学 8888 @ 8888

还是系统的文章有价值，之前在网上到处看，没这么详细

1年前

---

Mandy\_1010

赞

13天前



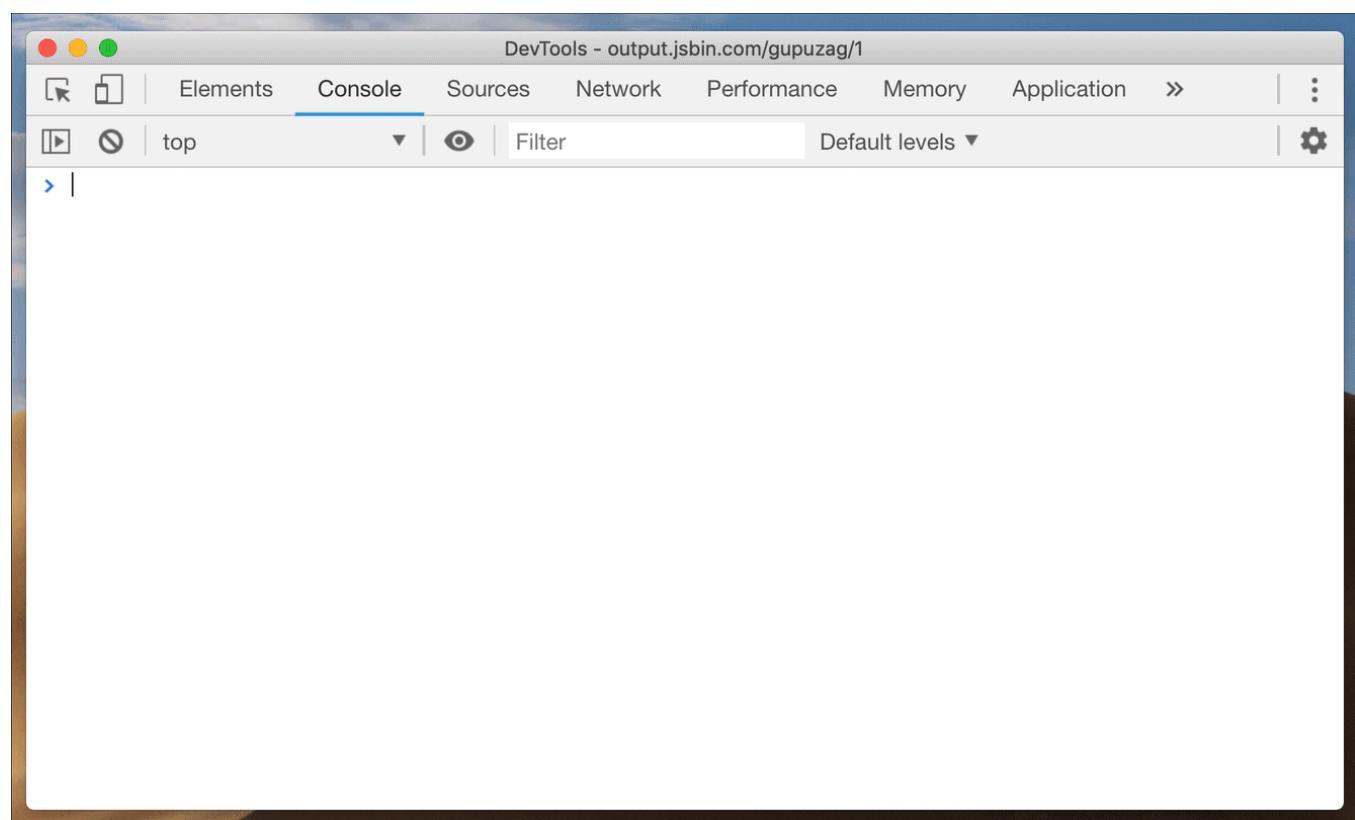
## 通用篇 - copying & saving

### 前言

在调试的过程中，我们总要对 `Dev Tools` 里面的数据进行 **复制** 或者 **保存** 的操作，所以我们来看看，关于这些，有什么小技巧呢？

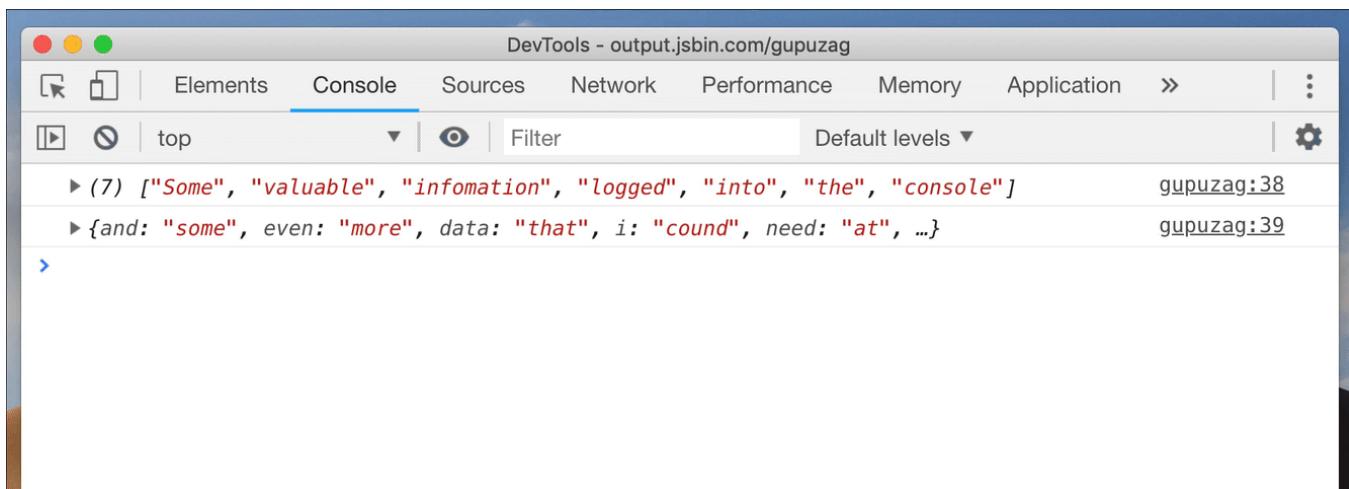
#### 1. `copy(...)`

你可以通过全局的方法 `copy()` 在 `console` 里 `copy` 任何你能拿到的资源，包括我们在后面[第六节]会提到的那些变量。例如 `copy($_)` 或 `copy($0)`



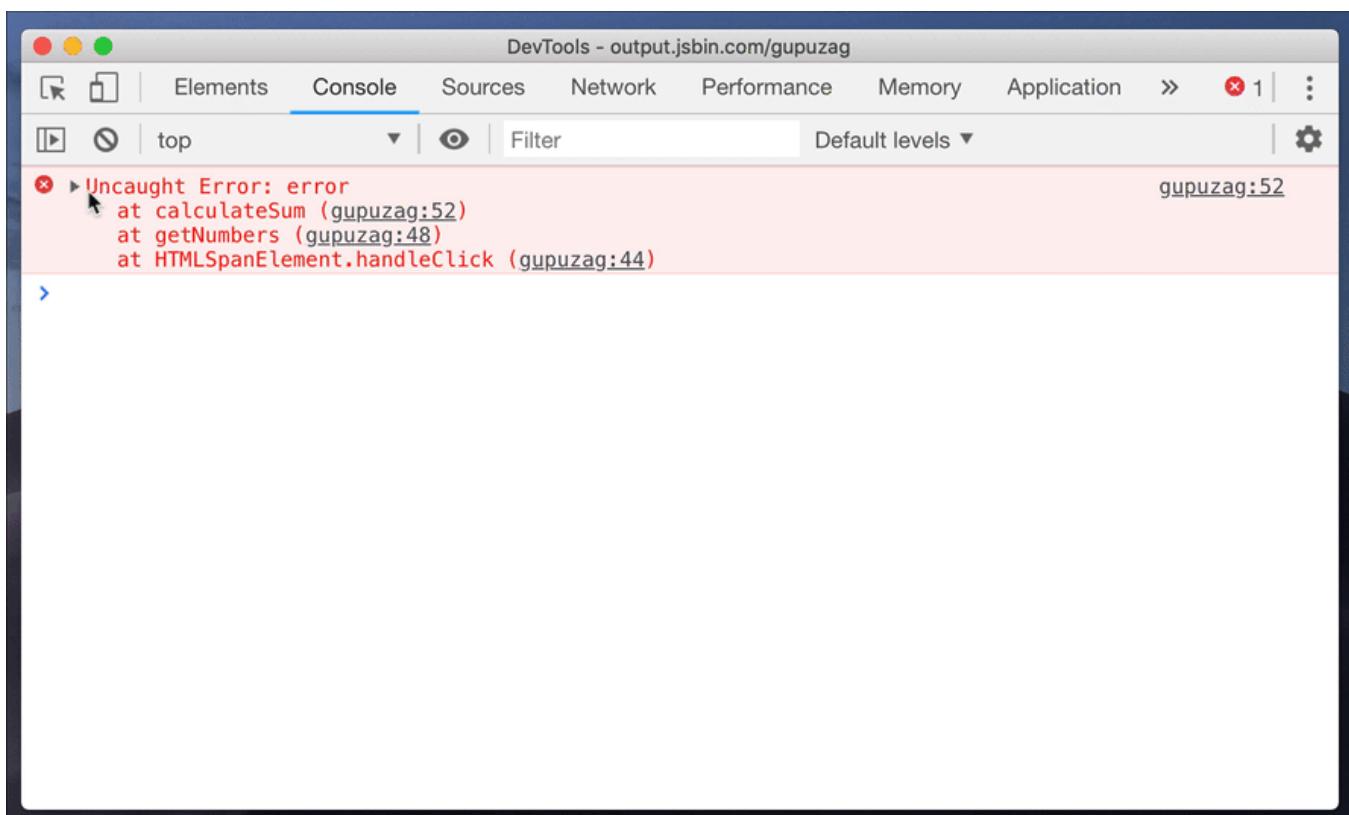
#### 2. `Store as global` (存储为一个全局变量)

如果你在 `console` 中打印了一堆数据（例如你在 `App` 中计算出来的一个数组），然后你想对这些数据做一些额外的操作比如我们刚刚说的 `copy`（在不影响它原来值的情况下）。那就可以将它转换成一个全局变量，只需要**右击**它，并选择“`Store as global variable`”（保存为全局变量）选项。



### 3. 保存堆栈信息( Stack trace )

大多数情况下都不是一个人开发一个项目，而是一个团队协作，那么 **如何准确的描述问题，就成为了沟通的关键**，这时候 `console` 打印出来的堆栈跟踪的信息对你和同事来说就起大作用了，可以省去很多沟通成本，所以你可以直接把堆栈跟踪的信息保存为一个文件，而不只是截图发给对方：



### 4. 直接Copy HTML

几乎所有人都知道，右击或者点击在 `HTML` 元素边上的省略号 (...) 就可以将它 `copy` 到剪贴板中，但是你不知道的是：古老的 `[ctrl] + [c]` 大法依旧可用！



## 留言

输入评论...

miku561 Front-end @ 小公司

copy 不能使用了。

系统: MAC OS

版本 83.0.4103.116 (正式版本) (64 位)

报错信息: copy is not a function

1月前

1

回复

qiuweihong Lv3 【全栈大佬的修炼之路...

保存堆栈信息get了

4月前

1

回复

ataola 前端开发 @ 公众号 @ ...

学习到了，好，很好，很好啊！

5月前

1

回复

风动

新技能get

5月前

1

回复

小米dds 前端开发

买不了吃亏，买不了上当，只能让你工作效率提升一档



提升一档

6月前

RaymondZhang 前端小白 @ 学生

请问这些是否适用于Microsoft Edge? (新版Beta)会不会有一些其他差别?

7月前

1

回复

fullmoonz233 Time Master @ O...

适用的

6月前

年轻向上的心 前端开发工程师

0.01 买不了吃亏，买不了上当，只能让你工作效率提升一档， 😊

7月前

1

回复

摘月 前端开发 @ Hello

0.01不亏

7月前

1

回复

zhoucuole

简单但很实用！

7月前

1

回复

zhongchun·ji 巨小的公司

delete variable\_name.

9月前

1

回复

Bufan4120 Lv1 前端

老哥，这个copy命令很好用啊，原来一直不知道

9月前

1

回复

潜潜潜 前端puppy

涨姿势了！

10月前

1

回复

negrochn Lv1

panels

(8) ["Elements", "Console", "Source", "Network", "Performance", "Memory", "Application",  
"Security"]

temp1

(8) ["Elements", "Console", "Source", "N...]

展开



想请问下大佬，这个动态gif图片是用什么软件做的啊？

1年前

1

回复

dendoink14454 **Lv4** (作者) 摸鱼小能手

你可以用 gifbrewery

1年前

---

夜舞暗澜

Chrome 版本 73.0.3683.86 (正式版本) (64 位) [Store as global]好像没有了。

1年前

1

回复

Senar **Lv1** 前端工程师 @ 德...

有的

1年前

---

wjfc

喜欢

1年前

1

回复

hock

copy 不出编辑器弹框，是需要设置什么吗

1年前

1

回复

未来重名了71674

同问？

1年前

---

beelinecode 数据

copy只是起到复制的作用，编辑器是作者用mac的快捷键启动的。。。

1年前

---

加载更多

---

dzxwind34380 **Lv1** 前端可视化攻城狮 @ ...

有时候用copy复制出来是[object Object]

copy是不是不能拷贝一些奇奇怪怪的东西，比如echarts实例化出来的对象。。

1年前

5

回复

---

zch2333 coder

store as global

这保存的还是引用，并不是....

1年前

2

回复



1年前

zhongchun·Ji 巨小的公司

作者的说法没错吧，vm 中心的变量，不影响原来的值？

9月前

---

Snailight

第一个copy的功能，是要先哪里设置关联到编辑器才会弹出copy后的结果？

1年前

回复

回复

Snailight

我错了，是可以copy，但是copy(对象)不生效，如copy(window) 会是[object Window]

1年前



# 通用篇 - 快捷键和通用技巧

## 前言

能直接快速提升开发效率的方式是什么？

- 掌握快捷键

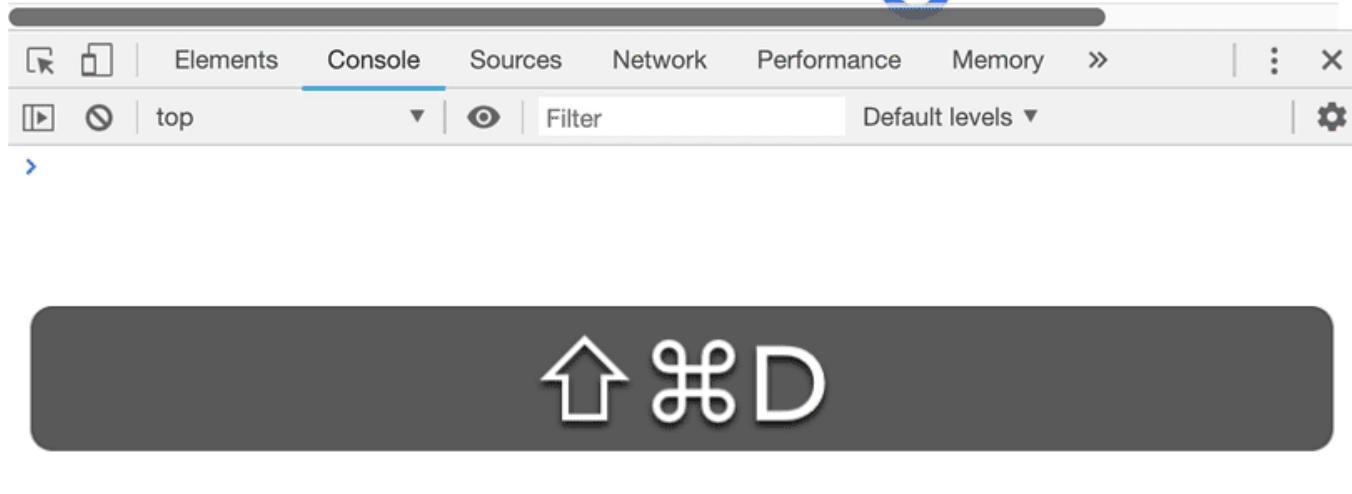
这里是一些我们在日常前端开发中，相当实用的快捷键：

### 1. 切换 DevTools 窗口的展示布局

一般我在使用 DevTools 时， dock 的展示窗口都在底部，但是有时候我想把 dock 的窗口切换到右边。

怎么做呢？

这时就可以通过 DevTools 的下拉菜单，或者命令菜单...或者使用一个快捷键 `ctrl + shift + D` (`⌘ + shift + D` Mac) 来实现位置的切换（通常是从 `开始的位置` 到 `右边位置`，但是如果一开始就是 `右边的位置` 那么会切换到 `左边的位置`）：

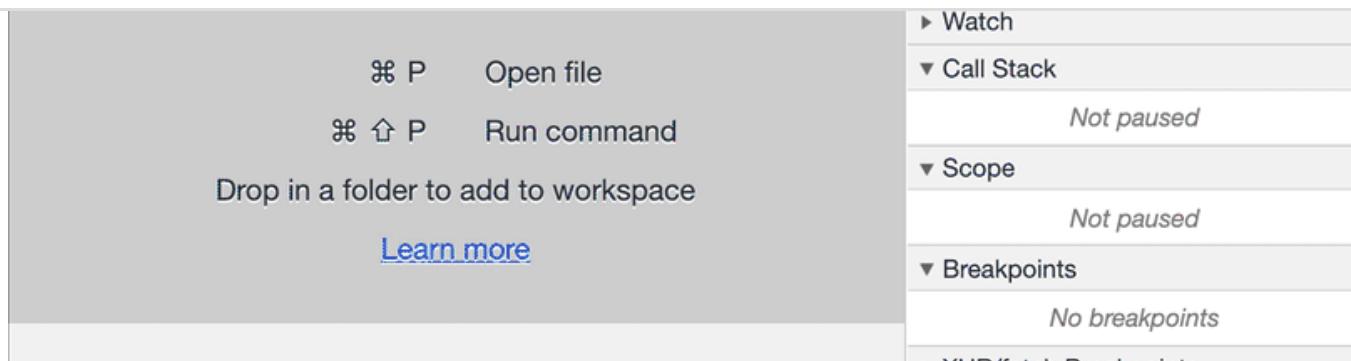
811px × 293px  
Gmail Image

## 2. 切换 DevTools 的面板

如果可以的话，我想成为一个不需要鼠标的开发者，日常开发中，我们常需要从 [元素面板](#) 跳转到 [资源面板](#) 并返回，这样往返的来调试我们的代码，怎么来节省鼠标点击的时间呢：

- 按下 `ctrl + [` 和 `ctrl + ]` 可以从当前面板的分别向左和向右切换面板。
- 按下 `ctrl + 1` 到 `ctrl + 9` 可以直接转到编号 `1 ... 9` 的面板( `ctrl + 1` 转到元素面板, `ctrl + 4` 转到 网络信息面板等等)

请注意! 我们在上面介绍的第二组快捷键默认被禁用了。你可以通过 [DevTools >> Settings >> Preferences >> \\*Appearance\\*](#) 打开这个选项：



### 3. 递增/递减

---

接下来这个技巧，对调整样式是最有用的：通过使用 带有 或者 不带有修饰键 的 上 / 下 箭头按键，你可以实现递增和递减 `0.1`，`1` 或者 `10` 这样数值类型的值。



DevTools 中的前4个主要的面板，每一个都支持 **[ctrl] + [f]** 快捷方式，你可以使用对应的查询方式来查找信息：

- 在 Elements 面板中 - 通过 `string`，选择器或者 `XPath` 来查找
- 而在 Console, Network 以及 Source 面板 - 通过区分大小写，或者可以被视为表达式的 `strings`，来查找

## 留言

输入评论...

Mandy\_1010

排版很好看~

13天前



回复

ataola 前端开发 @ 公众号 @ ...

涨知识了，好，很好，很好啊！

5月前



回复

linmu **Lv1** 前端工程师 @ TEST

递增、递减没看懂是干嘛的

7月前



回复

林不渡 **Lv1** 全栈工程师 @ 阿...

比如你调px就可以用箭头增加/减少

5月前



2月前

认真的来吧 **Lv1** 服务端写码的

sss

9月前



回复

Kana\_yang

现在惊喜的发现ctrl + [1...9]同样可以跳到对应的网页，以前都是ctrl+tab一个个跳或者直接鼠标点击的，麻烦多了！赞！

10月前



回复

andy凌云 **Lv1** web前端

(⊙o⊙)...，相对于跳转网页。我觉得跳转chrome内的面板可以pass了

21天前

StevenLikeWaterm... **Lv2** web前端

优秀

1年前



回复

X酱45920

官方快捷键 [https://developers.google.com/web/tools/chro...](https://developers.google.com/web/tools/chrome-devtools/)

1年前



回复

arronK 前端开发工程师 @ 罗...

cmd + shift + M 切换手机模拟模式和普通模式

1年前



回复

arronK 前端开发工程师 @ 罗...

在mac上，对像素大小的调整，

option + 箭头：步长为0.1

箭头：步长为1

shift + 箭头：步长为10

cmd + 箭头：步长为100

1年前



回复

angelayun **Lv1** 前端开发工程师

不错

10月前

不会做饭却想当厨子... 前端开发 @ 1

666

3月前



箭头：步长为1

cmd + 箭头：步长为100

cmd + 箭头：步长为100

1年前

1 ↗ 3

回复

---

风到这里就是粘 前端

记录一下：

1. Ctrl + Shift + D: 切换DevTools窗口的布局（下右）

2. 切换DevTools选项卡

- Ctrl + [ 或 ]: 向左和向右

- Ctrl + 1 到 9: DevTools>Settings>Preferences>\*Appearance\*中打开...

展开

1年前

1 ↗ 1

回复

风到这里就是粘 前端

CMD + - 或 =: 可缩放鼠标对应区域

1年前

---

zhongchun·jī 巨小的公司

优秀

9月前

---

宇智波鼬 Lv1 忍 @ 木叶

xpath要学习下是什么东东了

1年前

1 ↗ 回复

---

Sifan46825 程序员 (React+Golan...

您好，最后一张图片用什么工具画的？想学习下

1年前

1 ↗ 7 回复

---

心谭 Lv3 前端开发 @ Tencent C...

内容没了，快来补一下吧

1年前

1 ↗ 回复

---

许阳同学

快捷键篇好像内容没了？

1年前

1 ↗ 回复

---

ふ web前端工程师

console面板，搜索对象里面的字符串，比如打印\$0.\_\_vue\_\_，知道\$store对象里面有某个字符串，但是不展开是搜不出来的，怎么破。

1年前

1 ↗ 2 回复



- 下右切换
- 2. 切换devtools面板
- `ctrl + [` 和 `ctrl + ]` 可以从当前面板的分别向左和向右切换面板。 ...

[展开](#)

1年前

1

回复

bingoLin 前端开发

递增递减这个快捷键有点意思，

1年前

回复

daan

第三点，跨度为100和0.1的快捷键是啥啊，大佬没看懂，求知解

1年前

回复

daan

是出来了，`shift: 10, ctrl:100, alt: 0.1`

1年前

dendoink14454 (作者) 摸鱼小能手

我重新做下这个图 双横线加箭头指的是 `pageup` 和 `pagedown` 这组键。

1年前

[加载更多](#)



## 通用篇 - 代码块的使用

### 前言

我经常使用 `JavaScript` 作为我的自动化工具，来处理第三方网站和应用程序：

比方说，我想看看有多少人在我的所有媒体帖子上鼓掌。`Medium` 没有提供这样的总数，所以我把这个小脚本组合在一起：

```
?('td:last-child .sortableTable-number')
  .map(el => parseInt(el.innerText))
  .reduce((total, value) => total + value);
```

当我打开状态页并且输入这段脚本到 `Console` 面板的时候，它将会返回这个数值：

The screenshot shows the Medium website's 'DevTools tips' section. It lists two stories:

- DevTools tips—day 2: copying & saving**: 3 min read · View story · Details. Statistics: 1K, 684, 65%, 72.
- DevTools tips—day 1: the console dollars**: 2 min read · View story · Details. Statistics: 4.2K, 2.3K, 56%, 218.

Below the stories, there is a 'NOVEMBER 2018' section titled 'Theming Angular'. It includes a '5 min read · In The Startup · View story · Details' link. The DevTools console is open, showing the executed JavaScript code and its result:

```
> $$('td:last-child .sortableTable-number')
  .map(el => parseInt(el.innerText))
  .reduce((total, value) => total + value);
< 3679
> |
```

现在看来，即使这个脚本并没有花费我太多的精力来编写，但也只是偶尔运行一下，所以对于我来说，记住一段这样的脚本会很麻烦：

那怎么解决这个问题呢？

这就是 `Snippets` 的用武之地：它允许你存放 `JavaScript` 代码到 `DevTools` 中，方便你复用这些 `JavaScript` 代码块：

进入到 `Sources` 面板，在导航栏里选中 `Snippets` 这栏，点击 `New snippet(新建一个代码块)`，然后输入你的代码之后保存，大功告成！现在你可以通过右击菜单或者快捷键：`[ctrl] + [enter]` 来



The screenshot shows the Chrome DevTools interface with the 'Sources' tab selected. On the left, the 'Snippets' panel is open, displaying a list of snippets: 'angular console sandbox', 'consoles', and 'fans'. The 'fans' snippet is currently selected. A context menu is open over the snippet list, with options '⌘ P Open file' and '⌘ ⌘ P Run command' visible. The main content area below the tabs is empty.

## 运行其他来源的代码块

当我在 `DevTools` 中预设了一组很棒的代码块以后，甚至都不必再通过 `Sources` 来运行它们。使用 `Command Menu` 才是最快的方式。只需在它的输入框中输入 `!`，就可以根据名字来筛选预设代码块：

The screenshot shows the Chrome DevTools interface with the 'Sources' tab selected. The 'Command Menu' input field at the top contains the character '!' (exclamation mark). Below the input field, the snippet list from the previous screenshot is visible, with 'fans' still selected. The main content area is empty.

## 留言

输入评论...

binwin20 Android码畜 @ 厦门

MAC 打开这个网页， Google Chrome help 占用30+%的 CPU，风扇一直转。

1月前



回复



linlinlinlinlin

请问最后的示例在哪里可以打开输入框？

4月前

1

回复

玲曦 前端工程师

ctrl shift p

3月前

dendoink14454 Lv4 (作者) 摸鱼小能手

用 cmd + p 就可以了

小册现在有了开源版本，更新了大量新内容，欢迎大佬继续支持哇

1月前

hooy8352 前端盘子 @ 这...

这个snippets缓存在哪的，关闭浏览器，清缓存会有影响吗

6月前

1

回复

hooy8352 前端盘子 @ 这...

就目前测试结果看，清cookie，清缓存，关闭浏览器都不会导致snippets丢失，而且是跨域名全局可用，应该是强存储于浏览器上的。类似于vscode snippets

6月前

梦短情长

用EDGE试了下，本地创建了个文件才可以用。反推谷歌应该也是这么干的

3月前

dendoink14454 Lv4 (作者) 摸鱼小能手

不会清除的，目前存储的机制还不是特别清楚。有兴趣可以看看开源的浏览器 [chromium.org](#)

小册现在有了开源版本，更新了大量新内容，欢迎大佬继续支持哇

1月前

9°新一 底层码农

ctrl+p，打开menu。输入?, !, :, >, ... 各有不同作用

9月前

5

回复

dendoink14454 Lv4 (作者) 摸鱼小能手

小册现在有了开源版本，更新了大量新内容，欢迎大佬继续支持哇

1月前



dendoink14454 Lv4 (作者) 摸鱼小能手

小册现在有了开源版本，更新了大量新内容，欢迎大佬继续支持哇

1月前

---

yoohohoho 前端开发

也可以把这种代码放到书签页点击执行

10月前



回复

dendoink14454 Lv4 (作者) 摸鱼小能手

这个我还没试过，到时候试试，谢谢你的提醒

小册现在有了开源版本，更新了大量新内容，欢迎大佬继续支持哇

1月前

---

username\_job? @ corp?

\$\$('.comment-list')[0].childElementCount

10月前



回复

jsgoshu Lv1 前端开发工程师 @...

为什么俩\$? 😂

10月前

---

一只羊\_ Lv1 前端

这一分钱花的真香

10月前



回复

SakuraMD Android @ 保密

是挺香的

9月前

---

dendoink14454 Lv4 (作者) 摸鱼小能手

小册现在有了开源版本，更新了大量新内容，欢迎大佬继续支持哇

1月前

---

dendoink14454 Lv4 (作者) 摸鱼小能手

回复 SakuraMD: 小册现在有了开源版本，更新了大量新内容，欢迎大佬继续支持哇

1月前

---

江湖再见\_ Lv3 一个前端 @ 公众号和...

学习了，已收录在 [ruizhengyun.cn/tutorial/chrome](http://ruizhengyun.cn/tutorial/chrome) 中（已注明来源）

1年前



回复



1年前

江湖再见\_ Lv3 一个前端 @ 公众...

回复 LeonVincent: 哈哈，你不赞赏不就好了

1年前

chor Lv2 前端 @ 在读

回复 江湖再见\_: 的确是不要脸

10月前

RogueSword

转载后自己还开赞赏的吗，这么不要脸的吗

10月前

郭靖同学 高级卫生管理总监 ...

回复 江湖再见\_: 怎么会有这么厚脸皮的说法

9月前

\_大白熊 前端

确实有点不要脸

9月前

江湖再见\_ Lv3 一个前端 @ 公众...

你们都是有脸面的，你们都在高地，面对舆论的压力，赞赏功能已去掉，各位开心就好

9月前

Sawyawoo 前端 🎨 @ 某某某

毕竟文章就不是你写的，你没有获得文章带来收益的权利。收益你可以当做宣传，与作者商量搞分成给原作者，但你如果转载后未经作者同意就开启赞赏，那已经是侵权行为，无论价值多与少。是法律更是尊重

3月前

晟功在望 前端攻城狮

23333.学到了（这么做确实是招喷的）

1月前

森林同学96302 前端弱狗

呦呦，不要脸还委屈上了。。。

1月前

风到这里就是粘 前端

1. Sources面板New snippet(新建一个代码块): 复用这些JavaScript代码块

1年前

12

回复



14-10

回复

曹小心\_Sunny 前端开发

可以直接Ctrl+P

1年前

---

浮...云 Lv1 web前端开发

回复 曹小心\_Sunny: 会掉起打印

1年前

小\_柒 CURD工程师

谢谢提醒！

1年前

---

薇尔莉特 Java程序员 @ 程...

回复 浮...云: 在 DevTools 里按 Ctrl+P不会调起打印

10月前

---

嘉成重名了 Lv1

这个说的对

8月前

---

RandomYang Lv2 D2

看到Medium我就意识到这可能是一篇翻译。

1年前



回复

肖炎 Lv1 前端开发 @ battmd

开头就说整个翻译国外一个人的系列了、

1年前

---

jionchen Lv1 前端 @ 18

这不重要

1年前

---

beval

这都不重要

3月前

---

dendoink14454 Lv4 (作者) 摸鱼小能手

新版本已经开源了，并且新增了很多原版没有的技巧。

其实相对于原来作者的内容来说，无论是排版还是归类还是翻译都是需要精力的。如果你觉得这些都无所谓，你可以自己尝试付出一次，然后听听别人对你的付出不屑一顾的声音。



上五楼快活 前端打杂

在command menu 用! +代码块名 查找预设代码块 有时候会出现失效的情况

1年前

1

回复

dendoink14454 Lv4 (作者) 摸鱼小能手

这个情况在最新的Chrome83 里面没有发现了。

新版本的小册已经开源了，并且新增了很多原版没有的技巧。欢迎阅读

1月前

---

石小阳 Lv2 web前端 @ 深演智能

1. Snippets允许你存放 JavaScript 代码到 DevTools 中，方便你复用这些 JavaScript 代码块：
2. sources面板snippets中new snippet。可以直接右键run运行也可以用[ctrl] + [enter]
3. 在command menu 用! +代码块名 查找预设代码块

1年前

1

回复



## console 篇 - console 中的 '\$'

### 前言

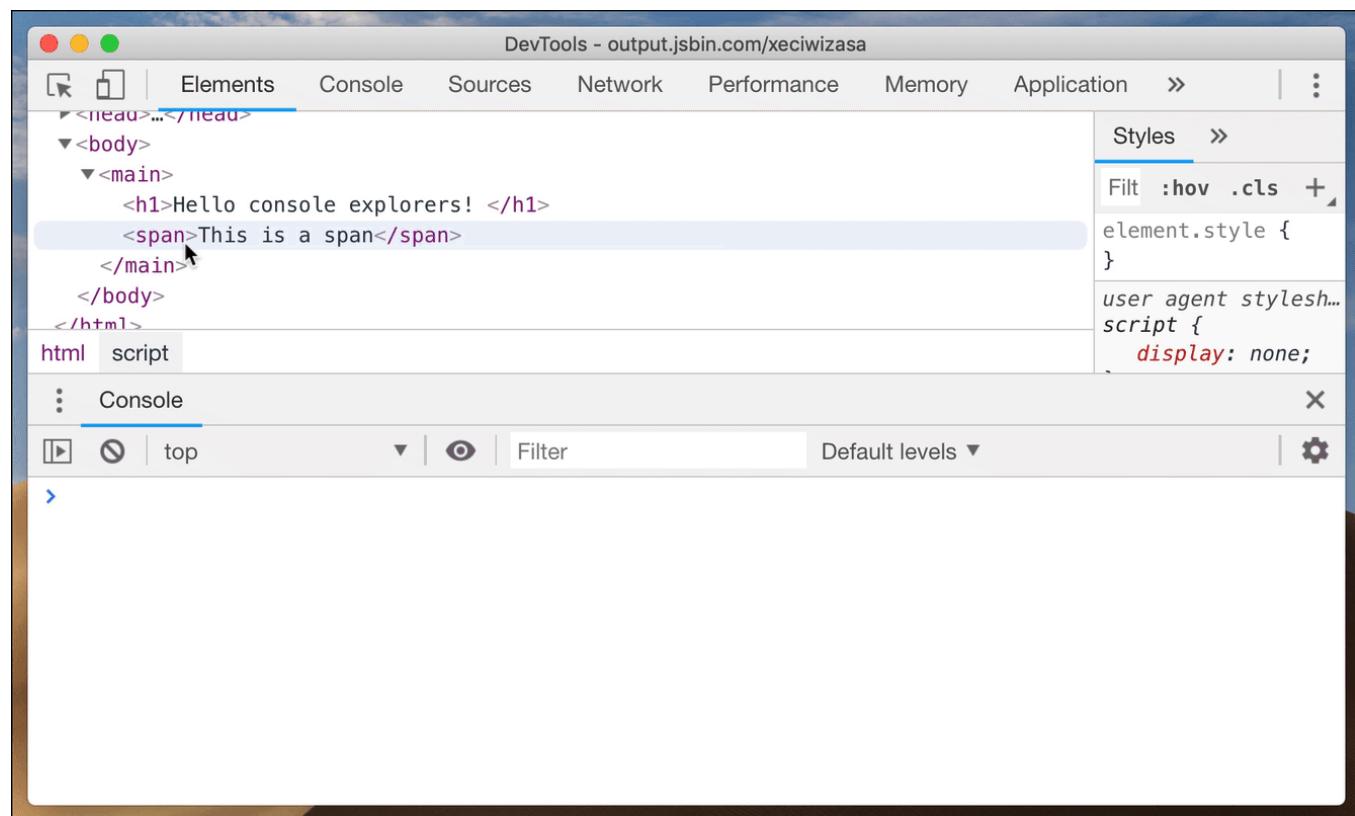
\$ 作为 jQuery 的选择器，承载了一代前端的太多记忆，但是你可能没有想到的是，在我们使用 Dev Tools 进行调试的时候，\$ 也有大放异彩的一天呢？

#### 1. \$0

在 Chrome 的 Elements 面板中，\$0 是对我们当前选中的 html 节点的引用。

理所当然，\$1 是对上一次我们选择的节点的引用，\$2 是对在那之前选择的节点的引用，等等。一直到 \$4

你可以尝试一些相关操作(例如: \$1.appendChild(\$0) )



#### 1. \$ 和 ?



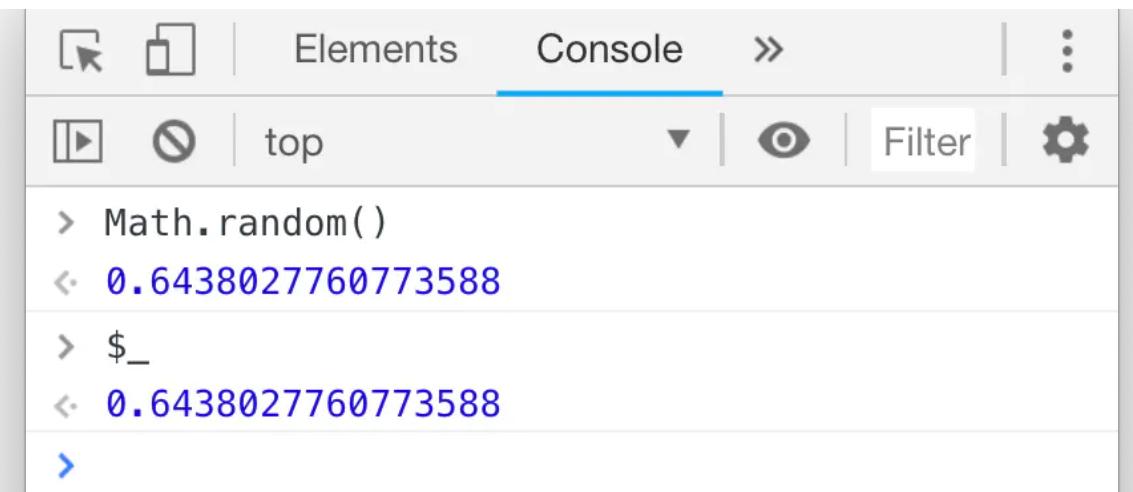
如果是 `?`  就更加厉害了，还能节省更多的时间，因为它不仅执行 `document.querySelectorAll` 并且它返回的是：一个节点的 `数组`，而不是一个 `Node list`

本质上来说 `Array.from(document.querySelectorAll('div')) === ?('div')`，但是 `document.querySelectorAll('div')` 和 `?('div')` 哪一种方式更加优雅呢？

## 2. `$_`

调试的过程中，你经常会通过打印查看一些变量的值，但如果你想看一下上次执行的结果呢？再输一遍表达式吗？

这时候 `$_` 就派上了用场，`$_` 是对上次执行的结果的引用：



## 3. `$i`

现在的前端开发过程，离不开各种 `npm` 插件，但你可能没有想过，有一天我们竟然可以在 `Dev Tools` 里面来使用 `npm` 插件！

有时你只是想玩玩新出的 `npm` 包，现在不用再大费周章去建一个项目测试了，只需要在 [Chrome插件:Console Importer](#) 的帮助之下，快速的在 `console` 中引入和测试一些 `npm` 库。

运行 `$i('lodash')` 或者 `$i('moment')` 几秒钟后，你就可以获取到 `lodash / momentjs` 了：



## 留言

输入评论...

beval

\$ i 好强

3月前



回复

dendoink14454 Lv4 (作者) 摸鱼小能手

新版本的小册已经开源了，并且新增了很多原版没有的技巧。欢迎阅读

1月前



回复

三只萌新 Lv2 前端

厉害厉害

6月前

dendoink14454 Lv4 (作者) 摸鱼小能手

新版本的小册已经开源了，并且新增了很多原版没有的技巧。欢迎阅读

1月前

qwei Lv1 前端开发 @ 高德地图

学到了！厉害厉害！

7月前



回复



10万+回复

回复

tmoe 前端工程师

Array.from(document.querySelectorAll('div')) === \$\$('div'), 顺手就粘到控制台试了试，我是杠精哈哈

10月前

1

回复

不动地丁

false~

8月前

make\_a\_sentence 前端

哈哈哈 俩数组引用肯定不一样 但是 Array.from(document.querySelectorAll('div'))[0] === \$\$('div')[0] ----->true

4月前

岁月\_ 前端工程师

为什么呢？求解

3月前

btc001

值对比和引用对比的区别。

2月前

1

回复

bb老猫 Lv1 三年级 @ studen

\$i有安全策略限制

10月前

1

回复

Hiwayne Lv2 菜鸟

\$i 下载的资源存储在哪里？是不是页面关闭了 就会清楚下载的内容 还是永久保存着

1年前

23

回复

hualin

\$i('immutable')安装成功后怎么调用呢？import { Map } from 'immutable'报错，let a = Map({select: 'users'}) 报错，该如何使用该包里的方法呢

1年前

3

回复

Hiwayne Lv2 菜鸟

直接用全局变量就可以了

1年前

hualin

可以具体点吗？直接使用全局变量immutable, Map 都不行

1年前



9月前

风到这里就是粘 前端

1. `$_`: 是对上次表达式的引用
2. `$i('lodash')`: 可将npm包引入console进行测试（需要安装Console Importer插件）

1年前

1 3

回复

风到这里就是粘 前端

- 1.
- `$i('lodash')`: 可将npm包引入console进行测试（需要安装Console Importer插件）

1年前

1

回复

蟹丸 Lv2 前端开发`$i` 掘金屏蔽了估计，百度可以

1年前

1

回复

vWEI 前端

现在也不行了

4月前

Neo绝心

`QuerySelectorAll`第一个字母应该小写？

1年前

1

回复

Crazy\_Joe Lv1 前端工程师`$i('moment')` 这么安装违反了一大堆安全协议，emmm 还是demo吧，，，

1年前

1

回复

Hzu\_zuoxiong 前端工程师

`$i` 报 `Uncaught ReferenceError: $i is not defined`  
at :1:1

1年前

1

回复

dendoink14454 Lv4 (作者) 摸鱼小能手

需要配合Chrome插件一起用。

1年前

慷慨 Lv2 前端 @ Alibaba

装了插件需要刷新一下页面

1年前



importer.js:1 Refused to load the script...

展开

1年前



回复

上五楼快活 前端打杂

\$i 运行报错 [\$i]: Fail to load lodash.js, is this URL([object Object]) correct?

1年前



回复

建磊 前端工程师 @ 紫...

网站不支持，换百度网站控制台就可以了

1年前

石小阳 Lv2 web前端 @ 深演智能

1. \$0

- Elements 面板中\$0 是对我们当前选中的 html 节点的引用。

- \$1 是对上一次我们选择的节点的引用，\$2 是对在那之前选择的节点的引用,一直到 \$4

2. \$ 和 \$\$

- console面板中 \$ 变量如果未曾被定义过就是document.quer...

展开

1年前



回复

Sungy Web前端攻城狮 @...

好认真的切图仔

1年前

石小阳 Lv2 web前端 @ 深演智...

回复 Sungy: 谢谢

1年前

lovelyJason Lv1 前端工程师 @ 某...

这一分钱花的真香

4月前

光辉GuangHui GuangHui @ BryanAd...

get \$\_、\$i

1年前



回复

MH在掘金

我安装后也不能使用,

1年前



回复

API搬运师 Lv1 API搬运师

很强！！





## console 篇 - console 中的 '\$'

### 前言

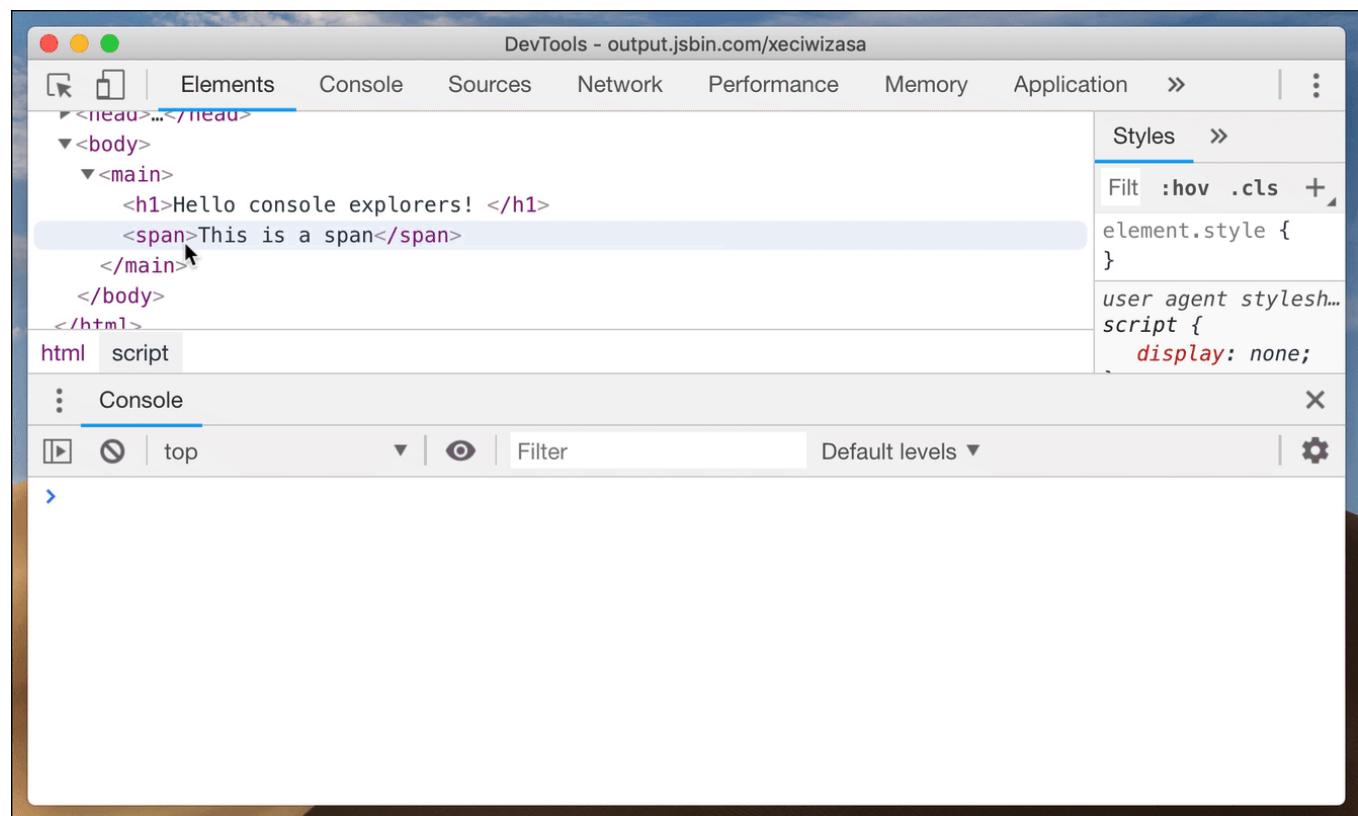
\$ 作为 jQuery 的选择器，承载了一代前端的太多记忆，但是你可能没有想到的是，在我们使用 Dev Tools 进行调试的时候，\$ 也有大放异彩的一天呢？

#### 1. \$0

在 Chrome 的 Elements 面板中，\$0 是对我们当前选中的 html 节点的引用。

理所当然，\$1 是对上一次我们选择的节点的引用，\$2 是对在那之前选择的节点的引用，等等。一直到 \$4

你可以尝试一些相关操作(例如: \$1.appendChild(\$0) )



#### 1. \$ 和 ?



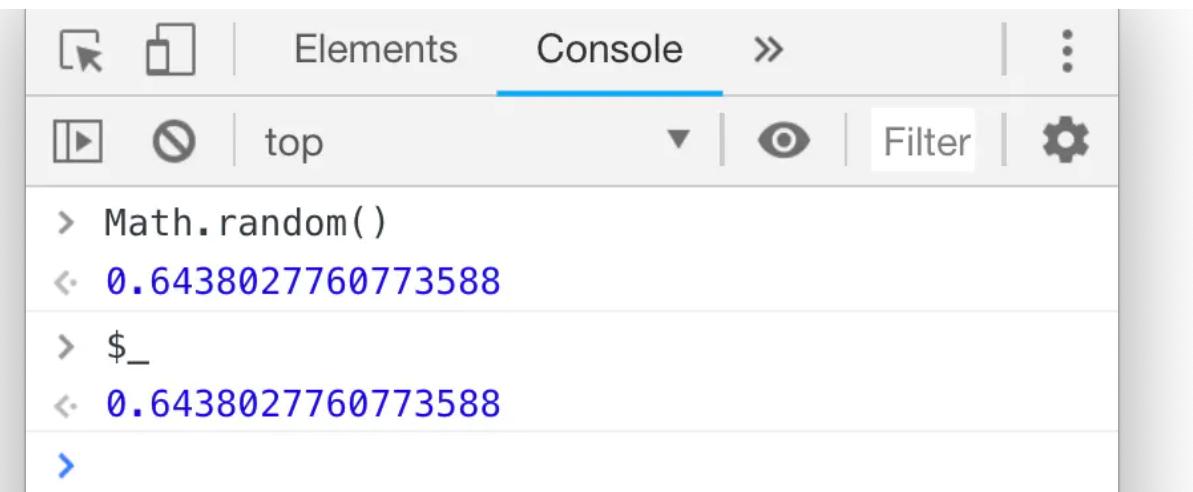
如果是 `?`  就更加厉害了，还能节省更多的时间，因为它不仅执行 `document.querySelectorAll` 并且它返回的是：一个节点的 `数组`，而不是一个 `Node list`

本质上来说 `Array.from(document.querySelectorAll('div')) === ?('div')`，但是 `document.querySelectorAll('div')` 和 `?('div')` 哪一种方式更加优雅呢？

## 2. `$_`

调试的过程中，你经常会通过打印查看一些变量的值，但如果你想看一下上次执行的结果呢？再输一遍表达式吗？

这时候 `$_` 就派上了用场，`$_` 是对上次执行的结果的引用：



## 3. `$i`

现在的前端开发过程，离不开各种 `npm` 插件，但你可能没有想过，有一天我们竟然可以在 `Dev Tools` 里面来使用 `npm` 插件！

有时你只是想玩玩新出的 `npm` 包，现在不用再大费周章去建一个项目测试了，只需要在 [Chrome插件:Console Importer](#) 的帮助之下，快速的在 `console` 中引入和测试一些 `npm` 库。

运行 `$i('lodash')` 或者 `$i('moment')` 几秒钟后，你就可以获取到 `lodash / momentjs` 了：



## 留言

输入评论...

beval

\$ i 好强

3月前



回复

dendoink14454 Lv4 (作者) 摸鱼小能手

新版本的小册已经开源了，并且新增了很多原版没有的技巧。欢迎阅读

1月前



回复

三只萌新 Lv2 前端

厉害厉害

6月前



回复

dendoink14454 Lv4 (作者) 摸鱼小能手

新版本的小册已经开源了，并且新增了很多原版没有的技巧。欢迎阅读

1月前



回复

qwei Lv1 前端开发 @ 高德地图

学到了！厉害厉害！

7月前



回复



10万+ 分享

收藏 10000+

tmoe 前端工程师

Array.from(document.querySelectorAll('div')) === \$\$('div'), 顺手就粘到控制台试了试，我是杠精哈哈

10月前

1

回复

不动地丁

false~

8月前

make\_a\_sentence 前端

哈哈哈 俩数组引用肯定不一样 但是 Array.from(document.querySelectorAll('div'))[0] === \$\$('div')[0] ----->true

4月前

岁月\_ 前端工程师

为什么呢？求解

3月前

btc001

值对比和引用对比的区别。

2月前

1

回复

bb老猫 Lv1 三年级 @ studen

\$i有安全策略限制

10月前

1

回复

Hiwayne Lv2 菜鸟

\$i 下载的资源存储在哪里？是不是页面关闭了 就会清楚下载的内容 还是永久保存着

1年前

23

回复

hualin

\$i('immutable')安装成功后怎么调用呢？import { Map } from 'immutable'报错，let a = Map({select: 'users'}) 报错，该如何使用该包里的方法呢

1年前

3

回复

Hiwayne Lv2 菜鸟

直接用全局变量就可以了

1年前

hualin

可以具体点吗？直接使用全局变量immutable, Map 都不行

1年前



9月前

风到这里就是粘 前端

1. `$_`: 是对上次表达式的引用
2. `$i('lodash')`: 可将npm包引入console进行测试（需要安装Console Importer插件）

1年前

1 3

回复

风到这里就是粘 前端

- 1.
- `$i('lodash')`: 可将npm包引入console进行测试（需要安装Console Importer插件）

1年前

1

回复

蟹丸 Lv2 前端开发`$i` 掘金屏蔽了估计，百度可以

1年前

1

回复

vWEI 前端

现在也不行了

4月前

Neo绝心

`QuerySelectorAll`第一个字母应该小写？

1年前

1

回复

Crazy\_Joe Lv1 前端工程师`$i('moment')` 这么安装违反了一大堆安全协议，emmm 还是demo吧，，，

1年前

1

回复

Hzu\_zuoxiong 前端工程师

`$i` 报 `Uncaught ReferenceError: $i is not defined`  
at :1:1

1年前

1

回复

dendoink14454 Lv4 (作者) 摸鱼小能手

需要配合Chrome插件一起用。

1年前

慷慨 Lv2 前端 @ Alibaba

装了插件需要刷新一下页面

1年前



importer.js:1 Refused to load the script...

展开

1年前



回复

上五楼快活 前端打杂

\$i 运行报错 [\$i]: Fail to load lodash.js, is this URL([object Object]) correct?

1年前



回复

建磊 前端工程师 @ 紫...

网站不支持，换百度网站控制台就可以了

1年前

石小阳 Lv2 web前端 @ 深演智能

1. \$0

- Elements 面板中\$0 是对我们当前选中的 html 节点的引用。

- \$1 是对上一次我们选择的节点的引用，\$2 是对在那之前选择的节点的引用,一直到 \$4

2. \$ 和 \$\$

- console面板中 \$ 变量如果未曾被定义过就是document.quer...

展开

1年前



回复

Sungy Web前端攻城狮 @...

好认真的切图仔

1年前

石小阳 Lv2 web前端 @ 深演智...

回复 Sungy: 谢谢

1年前

lovelyJason Lv1 前端工程师 @ 某...

这一分钱花的真香

4月前

光辉GuangHui GuangHui @ BryanAd...

get \$\_、\$i

1年前



回复

MH在掘金

我安装后也不能使用,

1年前



回复

API搬运师 Lv1 API搬运师

很强！！





## console篇 - Ninja console.log (忍者打印)

有时你设置的断点是不是被执行了太多次？假设有一个包含 200 个元素的循环，但是你只对第 110 次循环的结果感兴趣，又或者你只对一些满足某些条件的结果感兴趣，怎么办呢？这就是我们要说的条件断点：

### 1. Conditional breakpoints 条件断点

这样的情况下，你可以设置一个条件断点：

- 右击行号，选择 `Add conditional breakpoint...`(添加条件断点)
- 或者右击一个已经设置的断点并且选择 `Edit breakpoint`(编辑断点)
- 然后输入一个执行结果为 `true` 或者 `false` 的表达式（它的值其实不需要完全为 `true` 或者 `false` 尽管那个弹出框的描述是这样说的）。

在这个表达式中你可以使用任何这段代码可以获取到的值（当前行的作用域）。

如果条件成立，这个断点就会暂停代码的执行：

The screenshot shows the Chrome DevTools interface with the 'Sources' tab selected. A file named '12-conditional-breakpoint.js' is open, containing the following code:

```
1 let sth = 0;
2
3 Array.from(Array(15))
4   .map(Math.random)
5   .forEach((v, i) => {
6     sth = sth + v * i;
7   })
```

Line 6 has a handwritten note in blue: "I suspect the issue is in line #6 so I'm putting a breakpoint here". The developer tools sidebar on the right shows the following state:

- Watch: Not paused
- Call Stack: Not paused
- Scope: Not paused
- Breakpoints: No breakpoints
- XHR/fetch Breakpoints
- DOM Breakpoints
- Object Breakpoints

The status bar at the bottom indicates "Line 6, Column 5".

### 2. The ninja (忍者) `console.log`

得益于条件断点，`console.log` 也有了新玩法：



的值，例如 `undefined`），它并不会暂停..

与其在你的源码的不同地方去添加 `console.log` / `console.table` / `console.time` 等等，不如你直接使用条件判断来将它们“连接”到 `Source` 面板中。这样的话，它们会一直执行，并且当你不再需要它们的时候，在 `Breakpoints section` 会清晰的列出它们。点两下鼠标你就可以把所有的都移除，就像一堆忍者一样突然消失！

The screenshot shows the Chrome DevTools interface with the 'Sources' tab selected. A file named '12-conditional-breakpoint.js' is open, containing the following code:

```
1 let sth = 0;
2
3 const objects = Array.from(Array(15), (v, i) => ({ value:
4
5 objects
6   .forEach(({ value }, i) => {
7     sth = sth + value * i;
8   })
9 }
```

The cursor is positioned at the end of line 9. In the bottom right corner of the code editor, there is a small icon with a red dot, which is the conditional breakpoint indicator. To the right of the code editor, the 'Breakpoints' panel is open, showing the following options:

- ▶ Watch
- ▶ Call Stack
- ▶ Scope
- ▼ Breakpoints
  - No breakpoints
  - ▶ XHR/fetch Breakpoints
  - ▶ DOM Breakpoints

这个技术在调试生产环境的问题时同样很有用，因为你通过这样的方式轻松将 `console logs` 插入到 `source` 里。

## 留言

输入评论...

绿叶萌飞 前端

怎么在Sources单独写一段js调试？大家都是先写一个html引入js调试的吗？

2月前



回复



4月前

回复

有蚊子 **Lv1** 捡勒色的 @ 大自然环...

我来，他的意思是，把某种判断或是console.log绑定到断点上面，由于断点是可以忽略的，因此其他东西跟着断点一起被忽略，就像一只大橘一样消失

5月前

回复

StevenLikeWaterm... **Lv2** web前端

这里少了一位读者的总结。。

1年前

回复

Leedom **Lv1**

一句卧槽走天下

1年前

回复

风到这里就是粘 前端

再也不用在源码里添加console.log了

1. Add conditional breakpoint...(添加条件断点)
2. 输入console.log()或值为true/false的条件表达式
3. 不用时在Breakpoints右键remove all

计时的：

console.time()

console.timeEnd()

1年前

回复

冯冯墨墨 **Lv2** web前端 @ 某线上教...

gif图的箭头引导是什么软件吗？感觉很棒

1年前

回复

tsintergy

我也觉得我也想知道

1年前

Ihoey **Lv3** 前端 @ 🇺🇸

标注用的字体很喜欢

1年前

光辉GuangHui GuangHui @ BryanAd...

条件断点这个比较实用

1年前

回复





## console篇 - 自定义格式转换器

### 前言

大多数的情况下，我们习惯使用 `DevTools` 的 `console` 默认对 `object` 的转换，但有时候我们想用与众不同的方式来处理。那我们就可以自定义输出对象的函数，它通常被称为 `Custom Formatter`。

请注意：在我们写一个之前，需要在 `DevTools` 进行设置（在 `DevTools` 的 `:` 下拉框找到设置，或者按下 `F1`）中把对应的设置打开：

### Settings

#### Preferences

#### Workspace

#### Blackboxing

#### Devices

#### Throttling

#### Shortcuts

### Preferences

#### Console

Hide network messages

Selected context only

Log XMLHttpRequests

Show timestamps

Autocomplete from history

Group similar

Eager evaluation

Preserve log upon navigation

Enable custom formatters

<-- turn this on  
(it's turned off by default)

`formatter` 长什么样呢？它是一个对象，最多包含三个方法：

- `header`：处理如何展示在 `console` 的日志中的主要部分。
- `hasbody`：如果你想显示一个用来展开对象的 ► 箭头，返回 `true`
- `body`：定义将被显示在展开部分的内容中。



```
window.devtoolsFormatters = [ {  
    header: function(obj){  
        return ['div', {}, ` ${JSON.stringify(obj, null, 7)} `]  
    },  
    hasBody: function(){  
        return false;  
    }  
}]
```

例子里移除了循环的结构的错误处理，让它看起来更加简洁

`header` 方法返回了一个 [JsonML](#) (注： `JsonML` : `JSON Markup Language` - `JSON` 标记语言) 数组，由这些组成：

1. 标签名
2. 属性对象
3. 内容 (文本值或者其他元素)

(如果看起来很眼熟的话，那可能是因为你之前写过一些 [React 代码](#) :D)

在输出的时候，这个简单的 `formatter` 对于每一层嵌套，直接以 7 个空格的缩进打印这个对象。所以现在我们的输出看起来是这样：



## 1. 自定义格式化转换器的应用实践

现有好几种 `custom formatter` 可供选择，例如：你可以在这个 [immutable-devtools](#) 仓库中找到对于 [Immutable.js](#) 结构的完美展示。但你同样可以自己造一个。

一般来说，每当你遇到结构不寻常的对象时，或大量的日志(最好避免这样的情况，但是有时候很有用)而你想从中做区分时，你可以采用 `custom formatter` 来处理。

一个很实用的窍门：直接将你不关心，不需要区别对待的对象过滤出来，直接在 `header` 方法里面 `return null`。让 `DevTools` 使用默认的格式化方式来处理这些值。

撇开实用性，我们还可以找点乐子：

这是一个关于 `console` 的蠢萌例子：它叫做 `console.clown()` :将打印对象进行转换，而且在对象前面加上一个 `emoji` 表情 ...

[源码在这里](#)为了方便大家尝试，源码贴在下面：

```
window.devtoolsFormatters = [{  
  header: function(obj){  
    if (!obj.__clown) {  
      return null;  
    }  
    // ...  
  }  
}];  
  
// 使用  
// console.clown()  
// console.log({__clown: true});  
// 输出:  
// 🎭 {__clown: true}
```



```
color: red;
border: dotted 2px gray;
border-radius: 4px;
padding: 5px;
`  

const content = `⌚ ${JSON.stringify(obj, null, 2)}`;  

try {
  return ['div', {style}, content]
} catch (err) { // for circular structures
  return null; // use the default formatter
}
},
hasBody: function(){
  return false;
}
}  

}  

console.clown = function (obj) {
  console.log({...obj, __clown: true});
}  

console.log({message: 'hello!'}); // normal log
console.clown({message: 'hello!'}); // a silly log
```

如你所见，我使用 `console.clown` 方法打印出来的对象被添加了一个特殊的属性，便于将它区分出来，并且在 `formatter` 中对它区别处理，但在大部分现实的案例中，这样更好：比如检查这个对象是不是某一个特殊类的实例等等。

对了，`clown` 打印出来了什么东西呢？在下面：



输入评论...

linmu Lv1 前端工程师 @ TEST

z这一串代码写在哪里呀？？？？

7月前



回复

下雨天DY Lv2 前端研发工程师 @ 瓜子

immutable-devtools 安装完插件就可以了么？但是没有效果啊

9月前



回复

奈司兔咪丘... 前端攻城狮 @ 屯儿

可以把代码放在snippes，每次先执行snipptes代码，虽说有点麻烦.....

7月前

StevenLikeWaterm... Lv2 web前端

这个太骚气了。。。

1年前



回复

dendoink14454 Lv4 (作者) 摸鱼小能手

所以为什么不关注一波呢

1年前

风到这里就是粘 前端

这个骚操作的确没有学会 😊

1年前



回复

Sungy Web前端攻城狮 @ 随...

这个骚操作没有学会呢，之前都是这么玩的JSON.stringify(obj, null, 4)

1年前



回复

自律的路上

请求这个formatter已经enable了 那自定义的格式代码写在哪里呢？

1年前



回复

ゝ .bindy 网页编辑 @ nd

自己的代码里面

1年前

小马哥8 前端开发工程师 @...



jinX21213 Lv1 前端工程师 @ 普...

回复 > .bindy: 所以如果需要这个功能的话，每个项目都要这么设置一遍？

10月前



## console篇 - 对象&方法

### 前言

---

在我们调试 `Javascript` 的代码时，`对象` 和 `方法` 作为经常被我们调试的对象，所以这里介绍关于 `对象` 和 `方法` 的调试技巧。

#### 1. `queryObjects`（对象查询）方法

---



```
class Person {  
    constructor(name, role) {  
        this.name = name;  
        this.role = role;  
    }  
}  
  
const john = new Person('John', 'dad');  
  
let kids = [  
    new Person('Mary', 'kid'),  
    new Person('Luke', 'kid'),  
];  
  
new Person('Lucius', 'uncle');  
  
console.log('How many people are do we have?');
```

假如我们有这样一段代码，我们在里面定义了一些对象。

问一个问题，我们怎么知道，在 **特定的时刻 + 特定的执行上下文** 有哪些对象呢？

DevTools 里的 `queryObjects` 函数可以展示这些信息。

请注意，列表中创建的最后一个对象是不可用的：在代码执行后，对于它的引用并没有留存下来，也就是说，我们只有 3 个 `person` 对象：



&gt;

## 2. monitor (镜像) 方法

`monitor` 是 `DevTools` 的一个方法，它能够让你“潜入”到任何 `_function calls`(方法的调用)中：每当一个 被潜入 的方法运行的时候，`console` 控制台 会把它的实例打印出来，包含 函数名 以及 调用它的参数。

我们把前面例子里面的 `Person` 类拿过来，并且给它扩展两个方法：

```
javascript
class Person {
  constructor(name, role) {
    this.name = name;
    this.role = role;
  }

  greet() {
    return this.getMessage('greeting');
  }

  getMessage(type) {
    if (type === 'greeting') {
      return `Hello, I'm ${this.name}!`;
    }
  }
}
```

如你所见，`greet` 方法通过一个特殊的参数来执行 `getMessage` 方法，让我们看看对 `getMessage` 方法进行追踪会产生什么结果：



这样做，会让我们少写很多 `console.logs` !

### 3. `monitorEvents` (镜像事件) 方法

在上文中，我们讨论了用 `monitor` 方法来监听函数，其实还可以使用名为 `monitorEvents` 的方法，对 `events` 做一样的事情：



输入评论...

蟹小酱 Lv1 前端开发

monitor怎么翻译成镜像了

7天前



回复

zliuCode Lv1 Senior iOS developer ...

干货满满，简洁实用。

1年前



回复

linmu Lv1 前端工程师 @ TEST

没理解其中的应用场景，难受

7月前



回复

qiufelihong Lv3 【全栈大佬的修炼...

哈哈，下面说浏览器已兼容

4月前

江湖再见\_ Lv1 一个前端 @ 公众号和...

monitor（镜像）方法，经过测试浏览器已经兼容此方法了。

1年前



回复

阿飞呀呀呀

学习了，monitor及monitorEvents可以让我们少写很多console.log，不过建议修改下翻译，monitor本意是监听/监视，若译为监听器，会更贴切一点。

1年前



回复

光辉GuangHui GuangHui @ BryanAd...

queryObjects(Constructor)应该是查询当前上下文下某构造函数创建的实例

疑问1：为什么和文章中返回结果不一致啊？

...

```
class Person {  
  constructor(name, role) {...
```

展开

1年前



回复

dendoink14454 Lv4 (作者) 摸鱼小能手

hi 不好意思这么晚回复你，估计你是在console面板直接运行了前面的代码，所以会出现数组长度为4的情况，这是因为chrome的console是默认保留了你new出来的第三个对象Luke但是如果是在js文件中，不会保留这个引用。

1年前



9月前



## console篇 - 对象&方法

### 前言

---

在我们调试 `Javascript` 的代码时，`对象` 和 `方法` 作为经常被我们调试的对象，所以这里介绍关于 `对象` 和 `方法` 的调试技巧。

#### 1. `queryObjects`（对象查询）方法

---



```
class Person {  
    constructor(name, role) {  
        this.name = name;  
        this.role = role;  
    }  
}  
  
const john = new Person('John', 'dad');  
  
let kids = [  
    new Person('Mary', 'kid'),  
    new Person('Luke', 'kid'),  
];  
  
new Person('Lucius', 'uncle');  
  
console.log('How many people are do we have?');
```

假如我们有这样一段代码，我们在里面定义了一些对象。

问一个问题，我们怎么知道，在 **特定的时刻 + 特定的执行上下文** 有哪些对象呢？

DevTools 里的 `queryObjects` 函数可以展示这些信息。

请注意，列表中创建的最后一个对象是不可用的：在代码执行后，对于它的引用并没有留存下来，也就是说，我们只有 3 个 `person` 对象：



&gt;

## 2. monitor (镜像) 方法

`monitor` 是 `DevTools` 的一个方法，它能够让你“潜入”到任何 `_function calls`(方法的调用)中：每当一个 被潜入 的方法运行的时候，`console` 控制台 会把它的实例打印出来，包含 函数名 以及 调用它的参数。

我们把前面例子里面的 `Person` 类拿过来，并且给它扩展两个方法：

```
javascript
class Person {
  constructor(name, role) {
    this.name = name;
    this.role = role;
  }

  greet() {
    return this.getMessage('greeting');
  }

  getMessage(type) {
    if (type === 'greeting') {
      return `Hello, I'm ${this.name}!`;
    }
  }
}
```

如你所见，`greet` 方法通过一个特殊的参数来执行 `getMessage` 方法，让我们看看对 `getMessage` 方法进行追踪会产生什么结果：



这样做，会让我们少写很多 `console.logs` !

### 3. `monitorEvents` (镜像事件) 方法

在上文中，我们讨论了用 `monitor` 方法来监听函数，其实还可以使用名为 `monitorEvents` 的方法，对 `events` 做一样的事情：



输入评论...

蟹小酱 Lv1 前端开发

monitor怎么翻译成镜像了

7天前



回复

zliuCode Lv1 Senior iOS developer ...

干货满满，简洁实用。

1年前



回复

linmu Lv1 前端工程师 @ TEST

没理解其中的应用场景，难受

7月前



回复

qiufelihong Lv3 【全栈大佬的修炼...

哈哈，下面说浏览器已兼容

4月前

江湖再见\_ Lv1 一个前端 @ 公众号和...

monitor（镜像）方法，经过测试浏览器已经兼容此方法了。

1年前



回复

阿飞呀呀呀

学习了，monitor及monitorEvents可以让我们少写很多console.log，不过建议修改下翻译，monitor本意是监听/监视，若译为监听器，会更贴切一点。

1年前



回复

光辉GuangHui GuangHui @ BryanAd...

queryObjects(Constructor)应该是查询当前上下文下某构造函数创建的实例

疑问1：为什么和文章中返回结果不一致啊？

...

```
class Person {  
  constructor(name, role) {...
```

展开

1年前



回复

dendoink14454 Lv4 (作者) 摸鱼小能手

hi 不好意思这么晚回复你，估计你是在console面板直接运行了前面的代码，所以会出现数组长度为4的情况，这是因为chrome的console是默认保留了你new出来的第三个对象Luke但是如果是在js文件中，不会保留这个引用。

1年前



9月前



## console篇 - console中骚操作

### 前言

我最开始接触前端的时候，学会用的就是 `console.log`，甚至现在，大部分情况也还在用它调试，但是，在不同的场景下，除了 `log`，其实有更好的选择。

#### 1. `console.assert`

在 [MDN](#) 中是这样定义的

```
javascript
console.assert(assertion, obj1 [, obj2, ..., objN]);
console.assert(assertion, msg [, subst1, ..., substN]); // c-like message formatting
```

当我们传入的第一个参数为 `假` 时，`console.assert` 打印跟在这个参数后面的值。

这个方法适用于什么情况呢？举个栗子：

>

通过它，你可以摆脱令人讨厌的 `if` 表达式，还可以获得堆栈信息。

请注意，如果你使用的 `NodeJS` 版本 `≤ 10.0`，`console.assert` 可能会中断后面代码的执行，但是在 `.10` 的版本中被修复了(当然，在浏览器中不存在这个问题)

#### 2. 增强 `log` 的阅读体验



假如有这么一堆你想要输出但看起来并不易读的数据

*Given the set of variables we want to log*  
假设我们现在有这么一组数据要打印

```
const myName = 'Tomek';
const pencilsCount = 7;
const timestampNow = +new Date();
const id = 5;
```

```
console.log(myName, pencilsCount, timestampNow, id);
```

*this won't look very readable:*

他们看起来不是那么易读



“哪一个值对应哪一个变量来着？”

为了让它变得更加易读，你可以打印一个对象 - 只需将所有 `console.log` 的参数包装在大括号中。感谢 ECMAScript 2015 中引入了 `enhanced object literal`(增强对象文字面量)，所以加上 `{}` 已经是你需要做的全部事情了：

### 3. `console.table`



打印出来。它不仅会根据数组中包含的对象的所有属性，去计算出表中的列名，而且这些列都是可以缩放甚至还可以排序!!!

如果你觉得展示的列太多了，使用第二个参数，传入你想要展示的列的名字：

对于后台而言，只有 `node` 版本大于 `10` 以上，`console.table` 才能起作用

## 4. `table` 和 `{}` 的配合

我们刚刚看到了 `console.table` 这个技巧，也了解了在他上面的 `{}`，那么我们为什么不将他们结合起来打造一个终极 `log` 呢？



## 5. `console.dir`

---

有时候你想要打印一个 `DOM` 节点。`console.log` 会将这个交互式的元素渲染成像是从 `Elements` 中剪切出来的一样。如果说你想要查看 这个节点所关联到的真实的js对象 呢？并且想要查看他的 属性 等等？

在那样的情况下，就可以使用 `console.dir`：



我们总是需要打印各式各样的信息，之前我们讨论了如何让输出的信息更加直观，但是如果我们需要打印相关的时间信息呢？这就用到了计时的相关操作。

如果你想要给你的应用中发生的事件加上一个确切的时间记录，开启 *timestamps*。你可以在设置(在调试工具中的 `:` 下拉中找到它，或者按下 `F1`)中来开启或者使用 [Commands Menu](#)：

## 7. 监测执行时间

---

与其在所有事上展示一个时间戳，或许你对脚本中的特殊的节点之间执行的时间跨度更加感兴趣，对于这样的情况，我们可以采用一对有效的 `console` 方法

- `console.time()` — 开启一个计时器
- `console.timeEnd()` — 结束计时并且将结果在 `console` 中打印出来

如果你想一次记录多件事，可以往这些函数中传入不同的标签值。(例如: `console.time('loading')`, `console.timeEnd('loading')`)



## 8. 给你的 `console.log` 加上 CSS 样式

如果你给打印文本加上 `%c` 那么 `console.log` 的第二个参数就变成了 `CSS` 规则！这个特性可以让你的日志脱颖而出(例如 [Facebook](#) 在你打开 `console` 的时候所做的一样)

## 9. 让 `console.log` 基于调用堆栈自动缩进

配合 `Error` 对象的 `stack` 属性，让你的 `log` 可以根据堆栈的调用自动缩进：

```
function log(message) {
  console.log(
    // 这句话是重点当我们 new 出来的 Error 对象时，会匹配它的stack 信息中的换行符，换行符出现的次数
    //   .repeat(new Error().stack.match(/\n/g).length - 2) + message
  );
}
```



```
return bar() + bar();
}

function bar() {
  log('bar');
  return baz() + baz();
}

function baz() {
  log('baz');
  return 17;
}

foo();
```

运行结果如下：

## 10. 直接在回调中使用 `console.log`

是不是经常有这样的情况，就是我确定要将什么传递给回调函数。在这种情况下，我会在里面添加一个 `console.log` 来检查。

有两种方式来实现：

- 在回调方法的内部使用 `console.log`
- 直接使用 `consolelog` 来作为回调方法。

我推荐使用第二种，因为这不仅减少了输入，还可能在回调中接收多个参数。(这在第一个解决方案中是没有的)



## 11. 使用实时表达式

---

在本文形成的不久前，`DevTools` 在 `Console` 面板中引入了一个非常漂亮的附加功能，这是一个名为 `Live expression` 的工具

只需按下 "眼睛" 符号，你就可以在那里定义任何 `JavaScript` 表达式。它会不断更新，所以表达的结果将永远，存在 :-)

同时支持定义好几个：



zliuCode Lv1 Senior iOS developer ...

学到了。

1月前



回复

贝贝不想说话

直接在回调中使用 console.log , 有测试成功的小伙伴吗

2月前



回复

DanielLeefu

console.log('%c666','color:#f40;font-size:50px')

2月前



回复

qiufei hong Lv3 【全栈大佬的修炼之路...

console.log('%chello world!','color:#fff;background:black;font-size:20px')

4月前



回复

狗子\_ Lv1 前端开发工程师 @ YYI...

1 好像现在直接打印console也能点击展开里面的内容了

2 console.log 的阅读体验增强，其实就是我们直接把需要打印的变量组合成一个对象传给console了~

6月前



回复

代码弱鸡 Lv1 打杂 @ 网易/腾讯

那眼睛和之前的source里那个snippets有啥区别

10月前



回复

陈陈陈RL Lv1 前端

牛逼了

1年前



回复

洋就是我25672

厉害啊

1年前



回复

烦英俊 Lv1 前端工程师

Got it

1年前



回复

问题先森 高级清洁工-找bug

一句卧槽走天下

1年前



回复

我是谁我在哪aaa

console.log('%c牛逼','font-size: 50px; color: red;')



1. console.log({ var1, var2 }): 用{}包起来，标记哪个是哪个
2. console.table(): 适用于数组、类数组、对象
3. console.dir(): 查看DOM关联到的真实对象
4. li = \$('li'): 可以创建一个DOM元素
4. console.time() — 开启一个计时器...

[展开](#)

1年前

[15](#)[回复](#)

本末 web @ 散人

console.log('%ca','font-size:30px;color: red;')

1年前

[1](#)[回复](#)

本末 web @ 散人

console.log('%ca','font-size:30px;color: red;')

1年前

[1](#)[回复](#)

} .bindy 网页编辑 @ nd

console.log加样式，是要在输出的字符串之前加"%c"，像这样 console.log('%ca','color:red')。文中没有明确说，写给不知道的人看下~

1年前

[4](#)[回复](#)

小汤哥 CTO @ 长沙荣沐...

谢谢

1年前



# Drawer 篇 - Drawer 常识

## 前言

说到 **Drawer** 大部分的朋友可能都很陌生，那 **Drawer** 是个什么东西？**Chrome DevTools** 有很多部分，被分为9个 **tab** (俗称选项卡) (**Elements** , **Console** , **Sources** , **Network** , 等等...)

但是，那仅仅是它的一部分而已！有一组平行的选项卡，被隐藏在主窗口之下。这个组合被称为 **Drawer**

## 1. 如何打开 **Drawer** ?

当你在 **DevTools** (任何选项卡) 中时，按 **[esc]** 来显示它，再次按 **[esc]** 隐藏它：

The screenshot shows the Chrome DevTools interface with the **Elements** tab active. The left pane displays the DOM tree with the **body** element selected. The right pane shows the **Styles** panel with the following CSS rules for the **body** element:

```
element.style {  
}  
body {  
    display: block;  
    margin: 8px;  
}
```

A detailed box model diagram is overlaid on the bottom right, illustrating the element's bounding box with its margin, border, padding, and content dimensions.

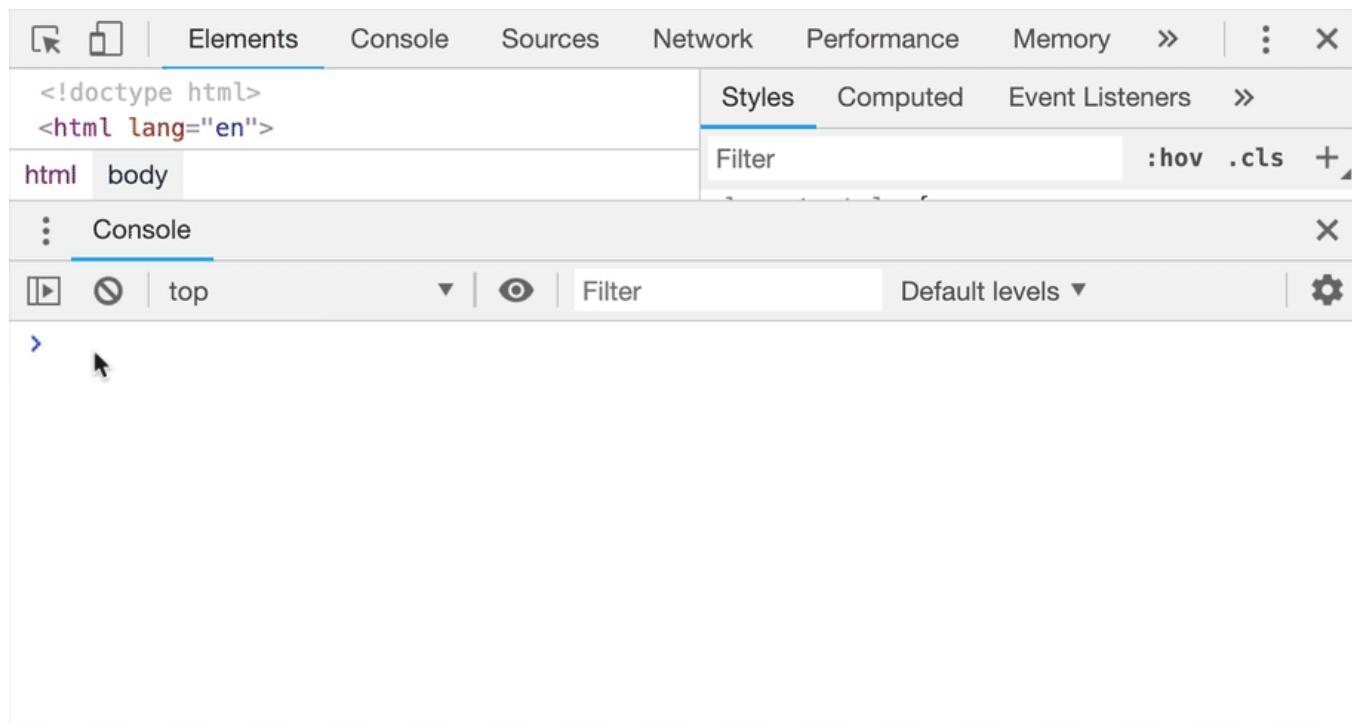
默认情况下，您会看到一个 **console** 选项卡。与主面板的 **console** 完全相同。这就是为什么主面板会显示除了 **console** 之外的每个主要标签 (**Elements** , **Sources** , **Network** ...) - 因为在主面板中显示 **console** 没有意义。

这样的 **console** 很方便，例如在 **Elements** 选项卡打开时，我们同时可以看到 **console** 面板。但是在 **Drawer** 中其实还隐藏了更多细节。



Drawer 里隐藏着许多其他功能，大多数时候你可能不需要用到它们，这也是它们为什么被隐藏起来的原因，然鹅，你可以直接选择你想展示在这里的功能。

点击主页面在 Drawer 的 console 面板前面的 : 图标来打开完整选项列表。另外，你也可以打开之前我们提到的 command Menu ，然后输入 Drawer 来打开



Okay，让我们再看一遍所有的选项：

- Animations
- Changes
- Console
- Coverage
- Network conditions
- Performance monitor
- Quick source
- Remote devices
- Rendering
- Request blocking
- Search
- Sensors
- What's new

### 3. 控制传感器



Drawer 里的 Sensors(传感器) 面板可以让你模拟特定的位置: 支持从预定义的位置中进行选择, 添加自己的位置, 或者手动键入纬度/经度。选定的值将被 `navigator.geolocation.watchPosition` (或 `.getCurrentPosition`) 报告。

如果你的 App 使用加速计, 传感器面板也可以模拟你设备在3D空间中的位置!

## 4 .模拟网络状态

---

就像伪造你的位置一样, 你可以使用 Drawer 里的 Network conditions 面板模拟特定的网络行为: 模拟互联网为典型的3G网络甚至离线! 这通常用来了解页面资源的大小。或者测试应用的离线功能。除此之外, Network conditions 面板还可以模拟特定的用户代理。



## 5.拿到 source

---

与 `drawer console` 一样，当我要专注于 `Elements` 面板时，有时我也想看源代码，这时就可以在 `drawer` 中选择显示 `Source`。



`sources` 中，而是显示在主 `Source` 中。

但是对于 快速查看代码，或者 设置断点 来说，它却是非常有用的技巧。

## 6. 检查代码 coverage

---

实际开发中，你的页面可能并不需要位于尾部的许多代码。也许那是来自外部库的一大块 `JavaScript`，也许是被遗忘的选择器上的一些 `CSS` 规则，它们不再匹配任何东西。

使用 `coverage` 面板来获得关于冗余代码的摘要-细节信息。使用 `Drawer` 菜单或者 `Command` 菜单来打开它。

`DevTools` 的 `coverage` 工具可以跟踪当前加载的 `JS` 和 `CSS` 文件的 哪些行正在被执行，并显示 未使用字节的百分比。

它用 绿色 的线条标记 `运行` 和用 红色 的线条标记 `未运行`。举个例子：



## 7. 检查你修改的内容

---

通过浏览器进行设计和调整 `css`：能够在代码执行的地方进行调试是方便又有趣的开发方式。但在某些时候，你可能希望将 [已更改的内容](#) 与 [最初加载的样式表](#) 进行比较。

为此，可以使用 [DevTools' Drawer](#) 中的更改工具。

它不仅会使用差异形式的变化（像 [Git](#) 这样的源控制工具一样）向你展示，同时还可以撤销它们。

举个例子：



## 留言

输入评论...

zliuCode Lv1 Senior iOS developer ...

耐何鄙人没文化，一句卧槽走天下。厉害了，醍醐灌顶！

1月前



回复

绿叶萌飞 前端

耐何鄙人没文化，一句卧槽走天下

2月前



回复

DanielLeefu

学到了

2月前



回复

cnjs 前端开发

6和7很实用

3月前



回复

豆豆大魔王 Lv2 前端工程师 @ 007加班...

厉害了。学到了！

3月前



回复



5月前

回复

linmu **Lv1** 前端工程师 @ TEST

在vue项目里面改了css 没有展示呀 是因为文件打过包的原因嘛?

7月前

回复

zsnmwyt

最后一个解决了我的痛点。

10月前

回复

丰木木 **Lv1**

最后一个学到了

1年前

回复

前端太难了 **Lv1** 前端小菜鸟

学到了

1年前

回复

Sungy Web前端攻城狮 @ 随...

怎么没人看了

1年前

回复

若川 **Lv3** 公号：若川视野 源...

可能是看完没评论罢了～

1年前

烦英俊 **Lv1** 前端工程师

Got it

1年前



# 结束语

## 前言

如果我用一句话来总结所有的 **Tips** 的话，读起来像是这样：

*“DevTools 是一款功能强大的软件，很容易被忽视，它隐藏着许多我们甚至常常怀疑是否需要的那些功能”*

好吧，那是两句话，无论如何，学无止境，最后也推荐给大家几个优质的内容

## 1. Google Chrome Developers

高度推荐你订阅 [Google Chrome Developers](#) 的 YouTube 频道，这是在 [Chrome](#) 中展示已有的新内容的地方。不仅如此，你还可以去那里寻找有关最佳实践，[JavaScript](#) 演变等的讨论。对于任何一位前端开发人员来说，这里都有许多有价值的内容。

[Google Chrome Developers - YouTube](#)

## 2. [Developers.google.com](#)

当我在撰写这个系列时，除了分享最喜欢的功能和技术积累之外，也在查阅 [Developers.google.com](#)，你可以（“按年”）或 技术（“按标签”）去查询一些有关特定更新的信息，例如关于最近发布的 [Chrome 71](#)：

[New in Chrome 71](#)

## 留言

输入评论...



感谢分享，只是没能看到最想看的用浏览器网络面板来分析性能和优化的信息的。

3月前



回复

Awaity 成长💪 快乐🎮 赚钱💰

感谢😊 发现很多没用过的骚操作，点赞

3月前



回复

律悠 前端工程师

学到了很多东西。在实际开发中，copy(...); 条件断点; console.table(); console.log({}); 切换主题等都能起到很大的作用，感谢作者分享，大赞。

4月前



回复

ataola 前端开发 @ 公众号 @ ...

谢谢分享，太赞了

5月前



回复

yuKe

给作者一个大大的赞 ❤️

6月前



回复

你在想什么

感谢分享

7月前



回复

qhdxdwdm

谢谢分享 受教了

7月前



回复

pistachio

谢谢分享 干货满满

7月前



回复

McChen99094 Lv1 高级前端开发工程师 @...

抱着一读的心态，确实有不少是之前没有用过的，受益匪浅。推荐闲时阅读~

8月前



回复

Gopal Lv3 前端开发工程师 @ 神...

thanks! 受益良多。

9月前



回复

像风一样的男人

这0.01值了

9月前



回复



9月前



回复

力娃陈咬金

都看完了，很赞

9月前



回复

力娃陈咬金

感谢分享

9月前



回复

如此也好

感谢分享

9月前



回复

Binggo43412 Lv1 前端开发 @ 转转

感谢这0.01带给我满满的干货

9月前



回复

余温撑到天亮 前端工程师

感谢分享~

9月前



回复

奋逗的码农哥

感谢分享！！！

9月前



回复

道廷途说 Lv2 前端工程师

感谢分享！

10月前



回复



# 结束语

## 前言

如果我用一句话来总结所有的 **Tips** 的话，读起来像是这样：

*“DevTools 是一款功能强大的软件，很容易被忽视，它隐藏着许多我们甚至常常怀疑是否需要的那些功能”*

好吧，那是两句话，无论如何，学无止境，最后也推荐给大家几个优质的内容

## 1. Google Chrome Developers

高度推荐你订阅 [Google Chrome Developers](#) 的 YouTube 频道，这是在 [Chrome](#) 中展示已有的新内容的地方。不仅如此，你还可以去那里寻找有关最佳实践，[JavaScript](#) 演变等的讨论。对于任何一位前端开发人员来说，这里都有许多有价值的内容。

[Google Chrome Developers - YouTube](#)

## 2. [Developers.google.com](#)

当我在撰写这个系列时，除了分享最喜欢的功能和技术积累之外，也在查阅 [Developers.google.com](#)，你可以（“按年”）或 技术（“按标签”）去查询一些有关特定更新的信息，例如关于最近发布的 [Chrome 71](#)：

[New in Chrome 71](#)

## 留言

输入评论...



感谢分享，只是没能看到最想看的用浏览器网络面板来分析性能和优化的信息的。

3月前



回复

Awaity 成长💪 快乐🎮 赚钱💰

感谢😊 发现很多没用过的骚操作，点赞

3月前



回复

律悠 前端工程师

学到了很多东西。在实际开发中，`copy(...)`; 条件断点; `console.table()`; `console.log({})`; 切换主题等都能起到很大的作用，感谢作者分享，大赞。

4月前



回复

ataola 前端开发 @ 公众号 @ ...

谢谢分享，太赞了

5月前



回复

yuKe

给作者一个大大的赞 ❤️

6月前



回复

你在想什么

感谢分享

7月前



回复

qhdxdwdm

谢谢分享 受教了

7月前



回复

pistachio

谢谢分享 干货满满

7月前



回复

McChen99094 高级前端开发工程师 @...

抱着一读的心态，确实有不少是之前没有用过的，受益匪浅。推荐闲时阅读~

8月前



回复

Gopal 前端开发工程师 @ 神...

thanks! 受益良多。

9月前



回复

像风一样的男人

这0.01值了

9月前



回复



9月前



回复

力娃陈咬金

都看完了，很赞

9月前



回复

力娃陈咬金

感谢分享

9月前



回复

如此也好

感谢分享

9月前



回复

Binggo43412 Lv1 前端开发 @ 转转

感谢这0.01带给我满满的干货

9月前



回复

余温撑到天亮 前端工程师

感谢分享~

9月前



回复

奋逗的码农哥

感谢分享！！！

9月前



回复

道廷途说 Lv2 前端工程师

感谢分享！

10月前



回复



# 结束语

## 前言

如果我用一句话来总结所有的 **Tips** 的话，读起来像是这样：

*“DevTools 是一款功能强大的软件，很容易被忽视，它隐藏着许多我们甚至常常怀疑是否需要的那些功能”*

好吧，那是两句话，无论如何，学无止境，最后也推荐给大家几个优质的内容

## 1. Google Chrome Developers

高度推荐你订阅 [Google Chrome Developers](#) 的 YouTube 频道，这是在 [Chrome](#) 中展示已有的新内容的地方。不仅如此，你还可以去那里寻找有关最佳实践，[JavaScript](#) 演变等的讨论。对于任何一位前端开发人员来说，这里都有许多有价值的内容。

[Google Chrome Developers - YouTube](#)

## 2. [Developers.google.com](#)

当我在撰写这个系列时，除了分享最喜欢的功能和技术积累之外，也在查阅 [Developers.google.com](#)，你可以（“按年”）或 技术（“按标签”）去查询一些有关特定更新的信息，例如关于最近发布的 [Chrome 71](#)：

[New in Chrome 71](#)

## 留言

输入评论...



感谢分享，只是没能看到最想看的用浏览器网络面板来分析性能和优化的信息的。

3月前



回复

Awaity 成长💪 快乐🎮 赚钱💰

感谢😊 发现很多没用过的骚操作，点赞

3月前



回复

律悠 前端工程师

学到了很多东西。在实际开发中，`copy(...)`; 条件断点; `console.table()`; `console.log({})`; 切换主题等都能起到很大的作用，感谢作者分享，大赞。

4月前



回复

ataola 前端开发 @ 公众号 @ ...

谢谢分享，太赞了

5月前



回复

yuKe

给作者一个大大的赞 ❤️

6月前



回复

你在想什么

感谢分享

7月前



回复

qhdxdwdm

谢谢分享 受教了

7月前



回复

pistachio

谢谢分享 干货满满

7月前



回复

McChen99094 高级前端开发工程师 @...

抱着一读的心态，确实有不少是之前没有用过的，受益匪浅。推荐闲时阅读~

8月前



回复 2

Gopal 前端开发工程师 @ 神...

thanks! 受益良多。

9月前



回复 1

像风一样的男人

这0.01值了

9月前



回复



9月前



回复

力娃陈咬金

都看完了，很赞

9月前



回复

力娃陈咬金

感谢分享

9月前



回复

如此也好

感谢分享

9月前



回复

Binggo43412 Lv1 前端开发 @ 转转

感谢这0.01带给我满满的干货

9月前



回复

余温撑到天亮 前端工程师

感谢分享~

9月前



回复

奋逗的码农哥

感谢分享！！！

9月前



回复

道廷途说 Lv2 前端工程师

感谢分享！

10月前



回复



# 结束语

## 前言

如果我用一句话来总结所有的 **Tips** 的话，读起来像是这样：

*“DevTools 是一款功能强大的软件，很容易被忽视，它隐藏着许多我们甚至常常怀疑是否需要的那些功能”*

好吧，那是两句话，无论如何，学无止境，最后也推荐给大家几个优质的内容

## 1. Google Chrome Developers

高度推荐你订阅 [Google Chrome Developers](#) 的 YouTube 频道，这是在 [Chrome](#) 中展示已有的新内容的地方。不仅如此，你还可以去那里寻找有关最佳实践，[JavaScript](#) 演变等的讨论。对于任何一位前端开发人员来说，这里都有许多有价值的内容。

[Google Chrome Developers - YouTube](#)

## 2. [Developers.google.com](#)

当我在撰写这个系列时，除了分享最喜欢的功能和技术积累之外，也在查阅 [Developers.google.com](#)，你可以（“按年”）或 技术（“按标签”）去查询一些有关特定更新的信息，例如关于最近发布的 [Chrome 71](#)：

[New in Chrome 71](#)

## 留言

输入评论...



感谢分享，只是没能看到最想看的用浏览器网络面板来分析性能和优化的信息的。

3月前



回复

Awaity 成长💪 快乐🎮 赚钱💰

感谢😊 发现很多没用过的骚操作，点赞

3月前



回复

律悠 前端工程师

学到了很多东西。在实际开发中，`copy(...)`; 条件断点; `console.table()`; `console.log({})`; 切换主题等都能起到很大的作用，感谢作者分享，大赞。

4月前



回复

ataola 前端开发 @ 公众号 @ ...

谢谢分享，太赞了

5月前



回复

yuKe

给作者一个大大的赞 ❤️

6月前



回复

你在想什么

感谢分享

7月前



回复

qhdxdwdm

谢谢分享 受教了

7月前



回复

pistachio

谢谢分享 干货满满

7月前



回复

McChen99094 高级前端开发工程师 @...

抱着一读的心态，确实有不少是之前没有用过的，受益匪浅。推荐闲时阅读~

8月前



回复

Gopal 前端开发工程师 @ 神...

thanks! 受益良多。

9月前



回复

像风一样的男人

这0.01值了

9月前



回复



9月前



回复

力娃陈咬金

都看完了，很赞

9月前



回复

力娃陈咬金

感谢分享

9月前



回复

如此也好

感谢分享

9月前



回复

Binggo43412 Lv1 前端开发 @ 转转

感谢这0.01带给我满满的干货

9月前



回复

余温撑到天亮 前端工程师

感谢分享~

9月前



回复

奋逗的码农哥

感谢分享！！！

9月前



回复

道廷途说 Lv2 前端工程师

感谢分享！

10月前



回复



# 结束语

## 前言

如果我用一句话来总结所有的 **Tips** 的话，读起来像是这样：

*“DevTools 是一款功能强大的软件，很容易被忽视，它隐藏着许多我们甚至常常怀疑是否需要的那些功能”*

好吧，那是两句话，无论如何，学无止境，最后也推荐给大家几个优质的内容

## 1. Google Chrome Developers

高度推荐你订阅 [Google Chrome Developers](#) 的 YouTube 频道，这是在 [Chrome](#) 中展示已有的新内容的地方。不仅如此，你还可以去那里寻找有关最佳实践，[JavaScript](#) 演变等的讨论。对于任何一位前端开发人员来说，这里都有许多有价值的内容。

[Google Chrome Developers - YouTube](#)

## 2. [Developers.google.com](#)

当我在撰写这个系列时，除了分享最喜欢的功能和技术积累之外，也在查阅 [Developers.google.com](#)，你可以（“按年”）或 技术（“按标签”）去查询一些有关特定更新的信息，例如关于最近发布的 [Chrome 71](#)：

[New in Chrome 71](#)

## 留言

输入评论...



感谢分享，只是没能看到最想看的用浏览器网络面板来分析性能和优化的信息的。

3月前



回复

Awaity 成长💪 快乐🎮 赚钱💰

感谢😊 发现很多没用过的骚操作，点赞

3月前



回复

律悠 前端工程师

学到了很多东西。在实际开发中，copy(...); 条件断点; console.table(); console.log({}); 切换主题等都能起到很大的作用，感谢作者分享，大赞。

4月前



回复

ataola 前端开发 @ 公众号 @ ...

谢谢分享，太赞了

5月前



回复

yuKe

给作者一个大大的赞 ❤️

6月前



回复

你在想什么

感谢分享

7月前



回复

qhdxdwdm

谢谢分享 受教了

7月前



回复

pistachio

谢谢分享 干货满满

7月前



回复

McChen99094 Lv1 高级前端开发工程师 @...

抱着一读的心态，确实有不少是之前没有用过的，受益匪浅。推荐闲时阅读~

8月前



回复

Gopal Lv3 前端开发工程师 @ 神...

thanks! 受益良多。

9月前



回复

像风一样的男人

这0.01值了

9月前



回复



9月前



回复

力娃陈咬金

都看完了，很赞

9月前



回复

力娃陈咬金

感谢分享

9月前



回复

如此也好

感谢分享

9月前



回复

Binggo43412 Lv1 前端开发 @ 转转

感谢这0.01带给我满满的干货

9月前



回复

余温撑到天亮 前端工程师

感谢分享~

9月前



回复

奋逗的码农哥

感谢分享！！！

9月前



回复

道廷途说 Lv2 前端工程师

感谢分享！

10月前



回复



# 结束语

## 前言

如果我用一句话来总结所有的 **Tips** 的话，读起来像是这样：

*“DevTools 是一款功能强大的软件，很容易被忽视，它隐藏着许多我们甚至常常怀疑是否需要的那些功能”*

好吧，那是两句话，无论如何，学无止境，最后也推荐给大家几个优质的内容

## 1. Google Chrome Developers

高度推荐你订阅 [Google Chrome Developers](#) 的 YouTube 频道，这是在 [Chrome](#) 中展示已有的新内容的地方。不仅如此，你还可以去那里寻找有关最佳实践，[JavaScript](#) 演变等的讨论。对于任何一位前端开发人员来说，这里都有许多有价值的内容。

[Google Chrome Developers - YouTube](#)

## 2. [Developers.google.com](#)

当我在撰写这个系列时，除了分享最喜欢的功能和技术积累之外，也在查阅 [Developers.google.com](#)，你可以（“按年”）或 技术（“按标签”）去查询一些有关特定更新的信息，例如关于最近发布的 [Chrome 71](#)：

[New in Chrome 71](#)

## 留言

输入评论...



感谢分享，只是没能看到最想看的用浏览器网络面板来分析性能和优化的信息的。

3月前



回复

Awaity 成长💪 快乐🎮 赚钱💰

感谢😊 发现很多没用过的骚操作，点赞

3月前



回复

律悠 前端工程师

学到了很多东西。在实际开发中，copy(...); 条件断点; console.table(); console.log({}); 切换主题等都能起到很大的作用，感谢作者分享，大赞。

4月前



回复

ataola 前端开发 @ 公众号 @ ...

谢谢分享，太赞了

5月前



回复

yuKe

给作者一个大大的赞 ❤️

6月前



回复

你在想什么

感谢分享

7月前



回复

qhdxdwdm

谢谢分享 受教了

7月前



回复

pistachio

谢谢分享 干货满满

7月前



回复

McChen99094 Lv1 高级前端开发工程师 @...

抱着一读的心态，确实有不少是之前没有用过的，受益匪浅。推荐闲时阅读~

8月前



回复

Gopal Lv3 前端开发工程师 @ 神...

thanks! 受益良多。

9月前



回复

像风一样的男人

这0.01值了

9月前



回复



9月前



回复

力娃陈咬金

都看完了，很赞

9月前



回复

力娃陈咬金

感谢分享

9月前



回复

如此也好

感谢分享

9月前



回复

Binggo43412 Lv1 前端开发 @ 转转

感谢这0.01带给我满满的干货

9月前



回复

余温撑到天亮 前端工程师

感谢分享~

9月前



回复

奋逗的码农哥

感谢分享！！！

9月前



回复

道廷途说 Lv2 前端工程师

感谢分享！

10月前



回复