

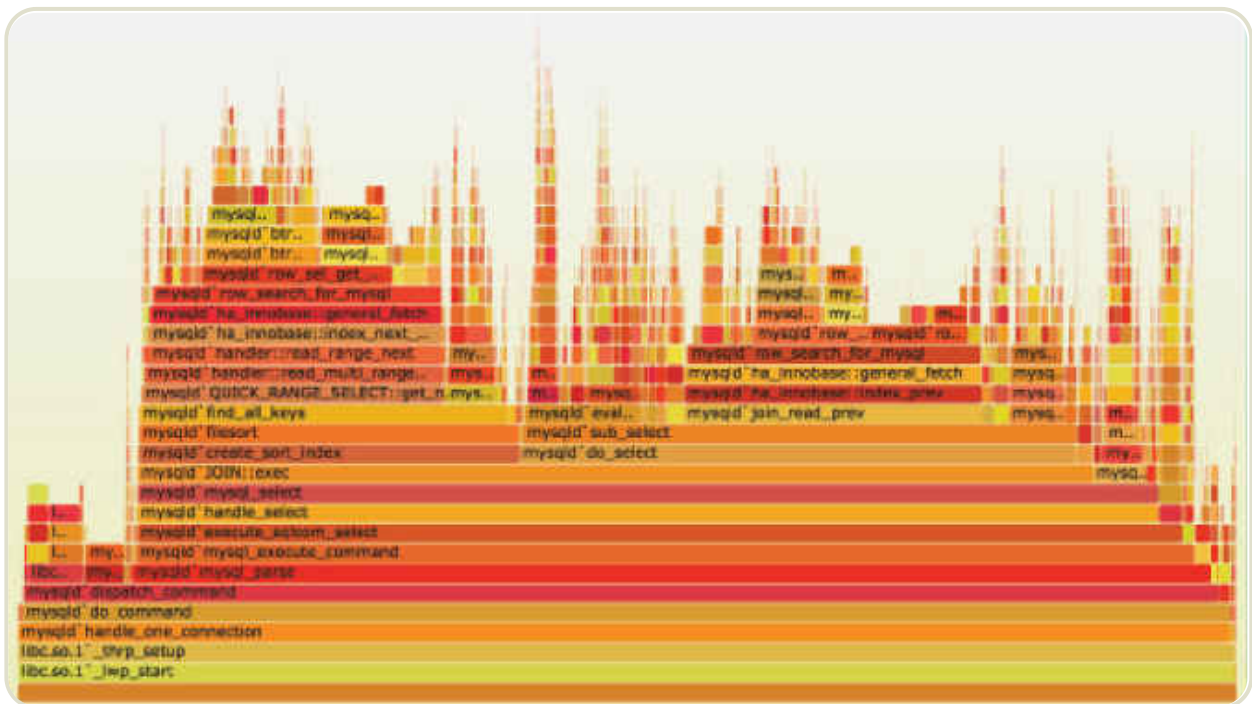
如何读懂火焰图？

作者： 阮一峰

日期： 2017年9月26日

软件的性能分析，往往需要查看 CPU 耗时，了解瓶颈在哪里。

[火焰图](#)（flame graph）是性能分析的利器。本文介绍它的基本用法。



一、perf 命令

让我们从 perf 命令（performance 的缩写）讲起，它是 Linux 系统原生提供的性能分析工具，会返回 CPU 正在执行的函数名以及调用栈（stack）。

通常，它的执行频率是 99Hz（每秒99次），如果99次都返回同一个函数名，那就说明 CPU 这一秒钟都在执行同一个函数，可能存在性能问题。

```
$ sudo perf record -F 99 -p 13204 -g -- sleep 30
```

上面的代码中，`perf record` 表示记录，`-F 99` 表示每秒99次，`-p 13204` 是进程号，即对哪个进程进行分析，`-g` 表示记录调用栈，`sleep 30` 则是持续30秒。

运行后会产生一个庞大的文本文件。如果一台服务器有16个 CPU，每秒抽样99次，持续30秒，就得到 47,520 个调用栈，长达几十万甚至上百万行。

为了便于阅读，`perf record` 命令可以统计每个调用栈出现的百分比，然后从高到低排列。

```
$ sudo perf report -n --stdio
```

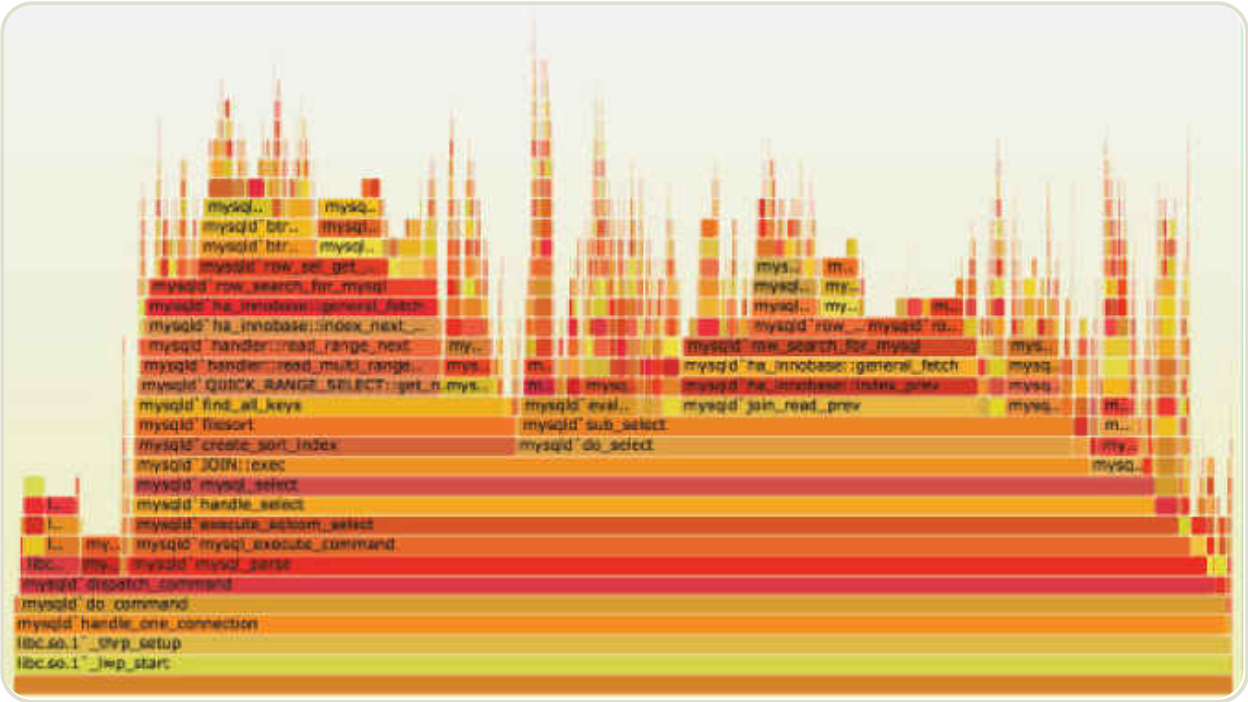
```
# perf report -n --stdio
[...]
```

Overhead	Samples	Command	Shared Object	Symbol
16.90%	490	dd	[kernel.kallsyms]	[k] xen_hypervisor_xen_version
--- xen_hypervisor_xen_version				
check_events				
--97.76%-- extract_buf				
extract_entropy_user				
urandom_read				
vfs_read				
sys_read				
system_call_fastpath				
__GI___libc_read				
--0.82%-- __GI___libc_write				
--0.82%-- __GI___libc_read				
--0.61%-- [...]				
5.83%	169	dd	[kernel.kallsyms]	[k] sha_transform
--- sha_transform				
extract_buf				
extract_entropy_user				
urandom_read				
vfs_read				
sys_read				
system_call_fastpath				
__GI___libc_read				
[...]				

这个结果还是不易读，所以才有了火焰图。

二、火焰图的含义

火焰图是基于 `perf` 结果产生的 [SVG 图片](#)，用来展示 CPU 的调用栈。



y 轴表示调用栈，每一层都是一个函数。调用栈越深，火焰就越高，顶部就是正在执行的函数，下方都是它的父函数。

x 轴表示抽样数，如果一个函数在 x 轴占据的宽度越宽，就表示它被抽到的次数多，即执行的时间长。注意，x 轴不代表时间，而是所有的调用栈合并后，按字母顺序排列的。

火焰图就是看顶层的哪个函数占据的宽度最大。只要有"平顶"（plateaus），就表示该函数可能存在性能问题。

颜色没有特殊含义，因为火焰图表示的是 CPU 的繁忙程度，所以一般选择暖色调。

三、互动性

火焰图是 [SVG 图片](#)，可以与用户互动。

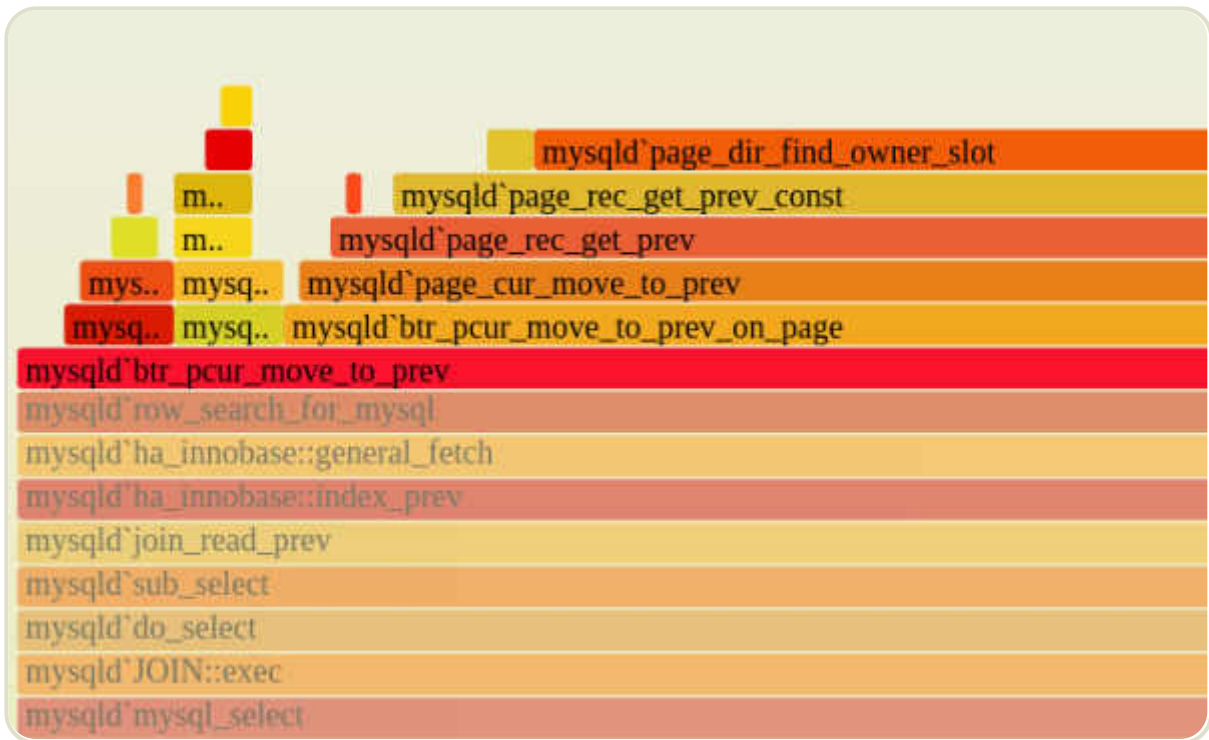
（1）鼠标悬浮

火焰的每一层都会标注函数名，鼠标悬浮时会显示完整的函数名、抽样抽中的次数、占据总抽样次数的百分比。下面是一个例子。

mysql'd JOIN::exec (272,959 samples, 78.34 percent)

（2）点击放大

在某一层点击，火焰图会水平放大，该层会占据所有宽度，显示详细信息。



左上角会同时显示"Reset Zoom", 点击该链接, 图片就会恢复原样。

(3) 搜索

按下 **Ctrl + F** 会显示一个搜索框, 用户可以输入关键词或正则表达式, 所有符合条件的函数名会高亮显示。

四、火焰图示例

下面是一个简化的火焰图例子。

首先, CPU 抽样得到了三个调用栈。

```
func_c
func_b
func_a
start_thread

func_d
func_a
start_thread

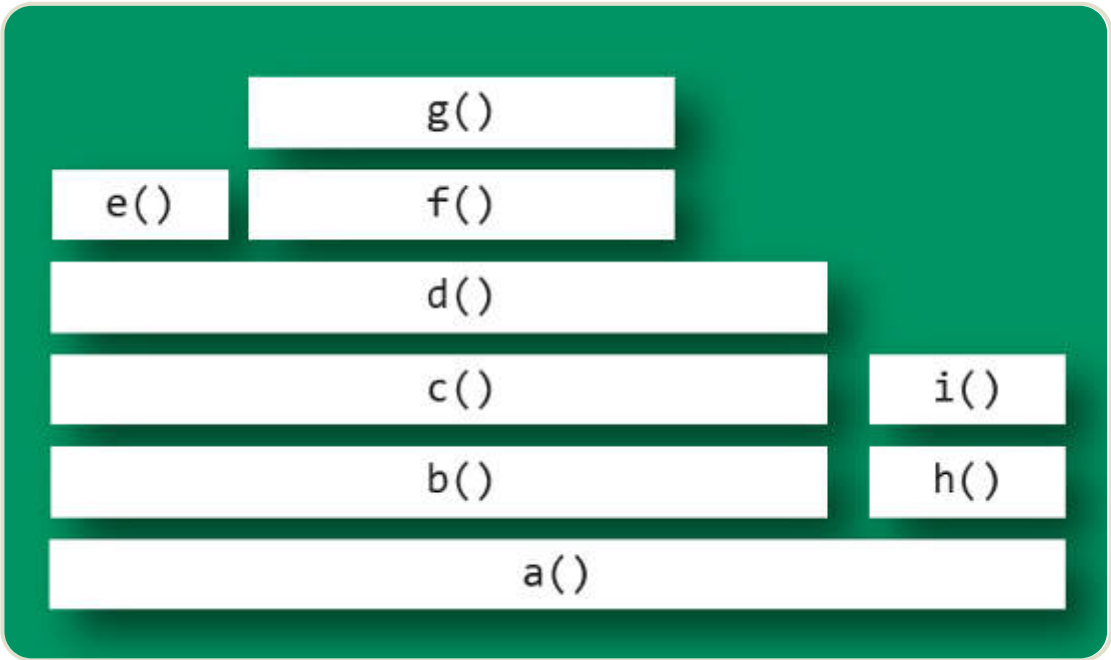
func_d
func_a
start_thread
```

上面代码中， `start_thread` 是启动线程，调用了 `func_a` 。后者又调用了 `func_b` 和 `func_d` ，而 `func_b` 又调用了 `func_c` 。

经过合并处理后，得到了下面的结果，即存在两个调用栈，第一个调用栈抽中1次，第二个抽中2次。

```
start_thread;func_a;func_b;func_c 1
start_thread;func_a;func_d 2
```

有了这个调用栈统计，[火焰图工具](#)就能生成 `SVG` 图片。



上面图片中，最顶层的函数 `g()` 占用 **CPU** 时间最多。`d()` 的宽度最大，但是它直接耗用 **CPU** 的部分很少。`b()` 和 `c()` 没有直接消耗 **CPU**。因此，如果要调查性能问题，首先应该调查 `g()` ，其次是 `i()` 。

另外，从图中可知 `a()` 有两个分支 `b()` 和 `h()` ，这表明 `a()` 里面可能有一个条件语句，而 `b()` 分支消耗的 **CPU** 大大高于 `h()` 。

五、局限

两种情况下，无法画出火焰图，需要修正系统行为。

（1）调用栈不完整

当调用栈过深时，某些系统只返回前面的一部分（比如前10层）。

（2）函数名缺失

有些函数没有名字，编译器只用内存地址来表示（比如匿名函数）。

六、Node 应用的火焰图

Node 应用的火焰图就是对 Node 进程进行性能抽样，与其他应用的操作是一样的。

```
$ perf record -F 99 -p `pgrep -n node` -g -- sleep 30
```

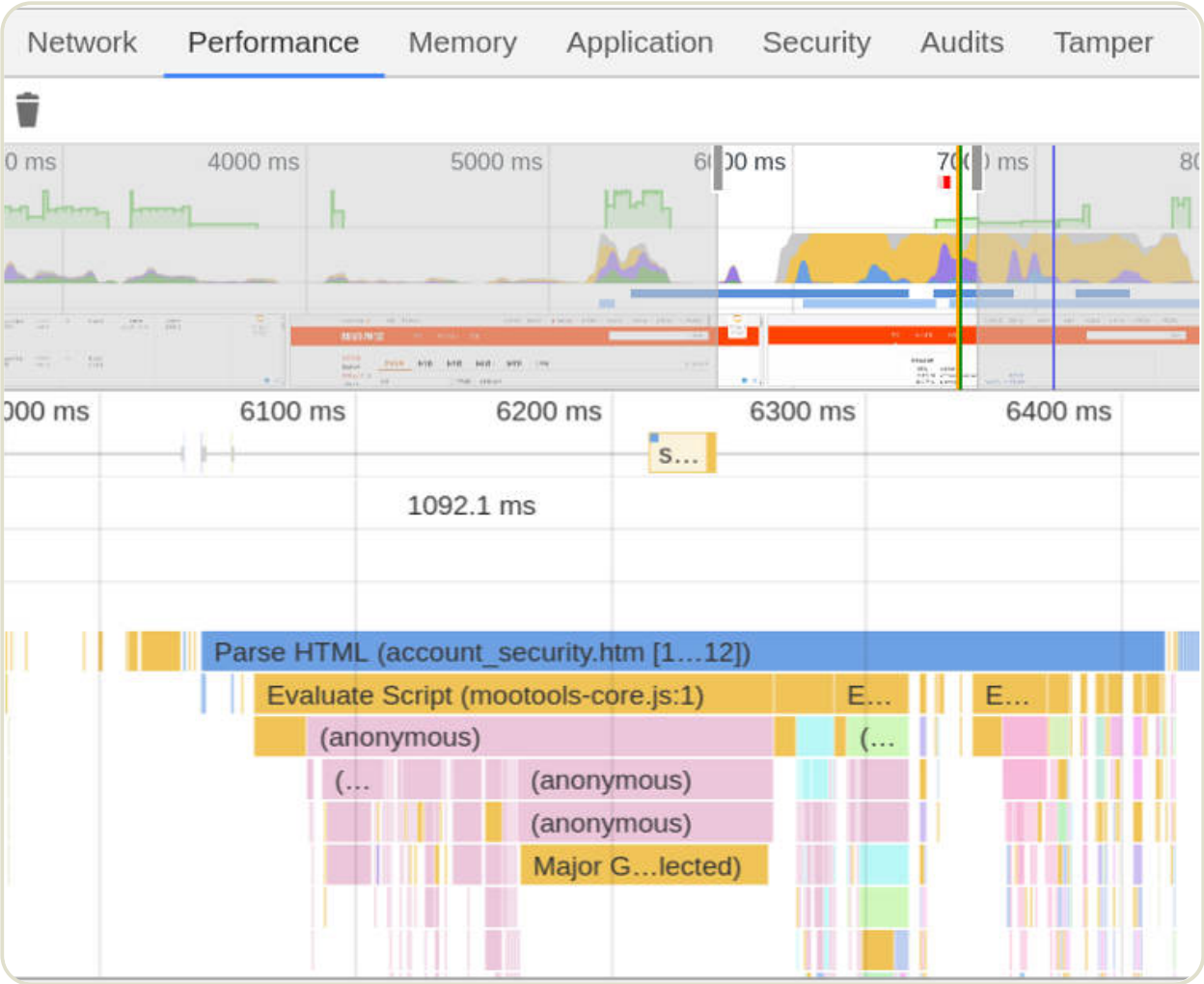
详细的操作可以看[这篇文章](#)。

七、浏览器的火焰图

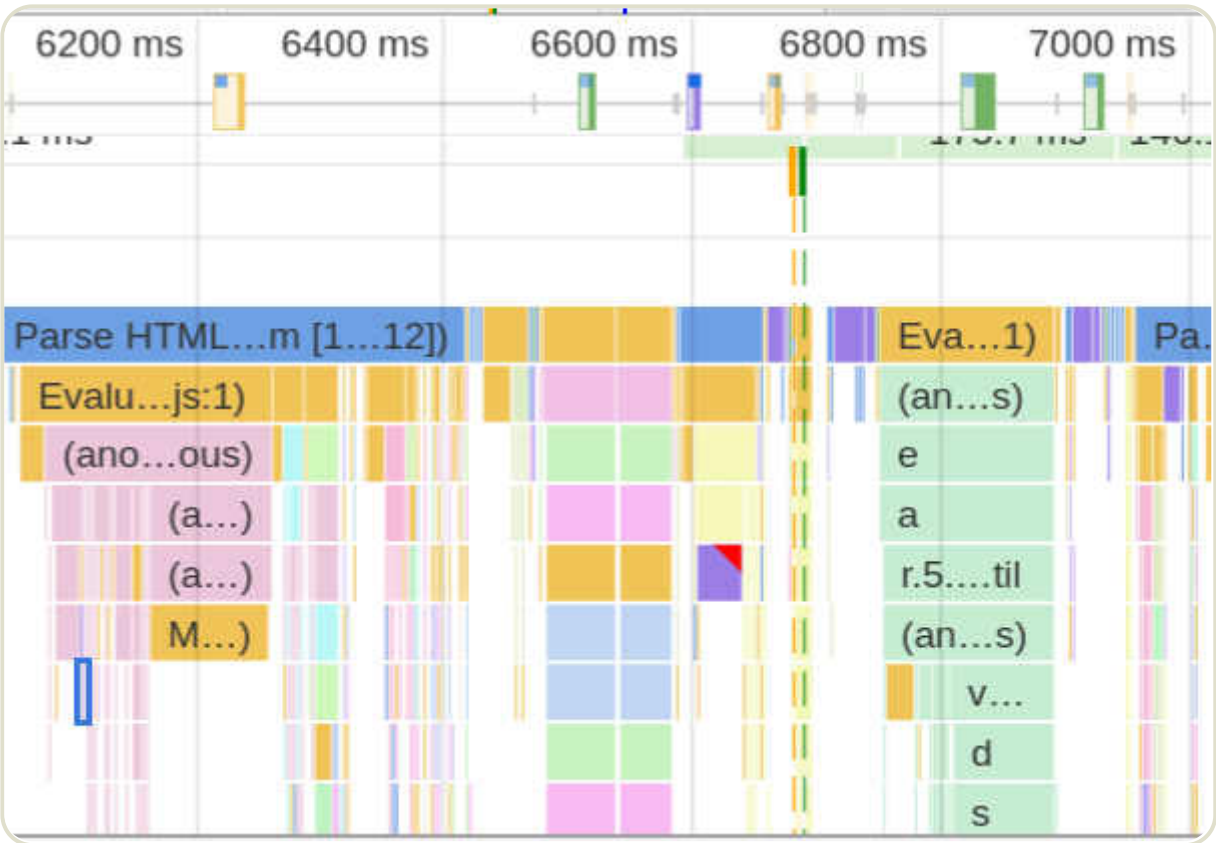
Chrome 浏览器可以生成页面脚本的火焰图，用来进行 CPU 分析。

打开开发者工具，切换到 **Performance** 面板。然后，点击"录制"按钮，开始记录数据。这时，可以在页面进行各种操作，然后停止"录制"。

这时，开发者工具会显示一个时间轴。它的下方就是火焰图。



浏览器的火焰图与标准火焰图有两点差异：它是倒置的（即调用栈最顶端的函数在最下方）；x 轴是时间轴，而不是抽样次数。



八、参考链接

- [火焰图的介绍论文](#)
- [火焰图官方主页](#)
- [火焰图生成工具](#)

(完)

文档信息

- 版权声明： 自由转载-非商用-非衍生-保持署名（[创意共享3.0许可证](#)）
- 发表日期： 2017年9月26日

相关文章

- **2020.12.13:** [《SSH 入门教程》发布了](#)

SSH 是登录 Linux 服务器的必备工具，只要你在做互联网开发，多多少少都会用到它。

- **2020.11.02:** [微信小程序入门教程之四：API 使用](#)

今天是这个系列教程的最后一篇。

- **2020.10.29:** [微信小程序入门教程之三：脚本编程](#)

这个系列教程的前两篇，介绍了小程序的项目结构和页面样式。

- **2020.10.27:** [微信小程序入门教程之二：页面样式](#)

这个系列的上一篇教程，教大家写了一个最简单的 Hello world 微信小程序。



Weibo | Twitter | GitHub

Email: yifeng.ruan@gmail.com

