

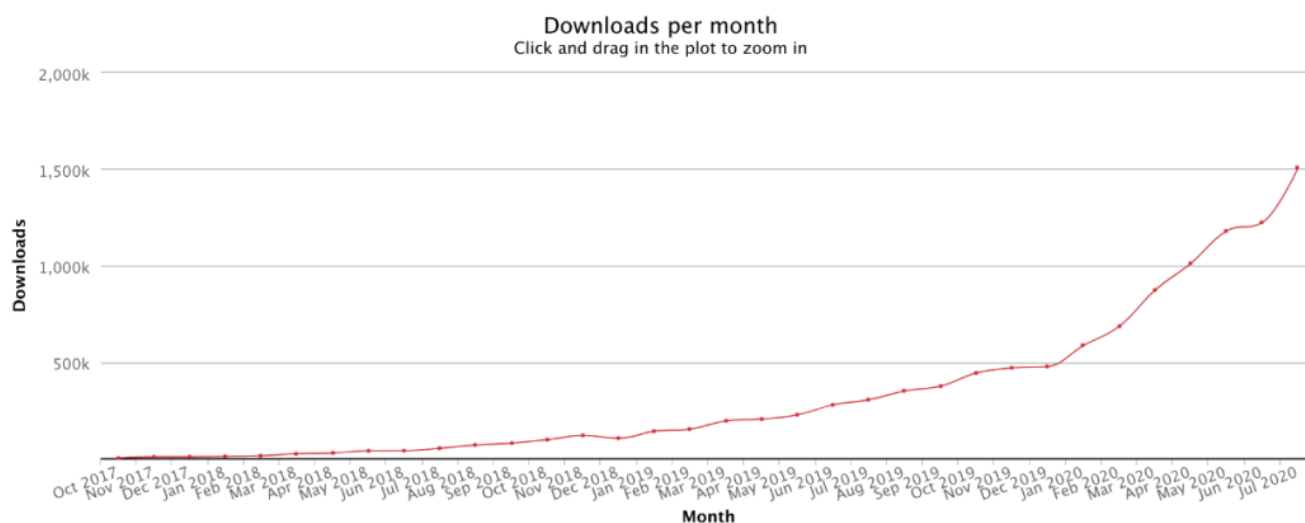
## Tailwind CSS: 从副业到数百万美元的业务

哥飞 2020-08-04 11:38:17

本文由谷歌翻译翻译自: <https://adamwathan.me/tailwindcss-from-side-project-byproduct-to-multi-million-dollar-business/>

它最初是作为主题发布在Twitter上的, 但我认为我会在这里重新发布它以提供一个合适的家。

因此, 大约一个月前, Tailwind的安装总数突破了1000万, 这开创了了不起的开端, 这真让我震惊。



我们还将从Tailwind UI中获得200万美元的收入, Tailwind UI是我们的第一个商业Tailwind CSS产品, 大约在5个月前发布, 距Tailwind CSS首次发布不到两年。

这是一开始的故事, 虽然它仍然很新鲜, 可以记住.....

## Reddit遇见Pinterest遇见Twitter


早在2015年, 我就告诉我的业务合作伙伴史蒂夫·舒格 (Steve Schoger) 一个网站项目的想法, 该网站可以让公司与他们的团队共享有趣的链接, 而外部人员可以订阅以查看他们崇拜的团队正在阅读什么。

我们称其为“摘要”。



# Welcome to Digest

A place for teams, organizations and individuals to share what they discover on the web.

 Sign in with Twitter

[Learn more](#)

By signing up, you agree to our [Terms](#) and that you have read our [Privacy Policy](#).

我们对此感到非常兴奋，因此我决定休假一周以构建初始原型。但是，以经典的开发人员方式，我花了整整一周的时间对技术堆栈进行决策，并且可能在最后一天花了整整一天才真正破解真正的功能。

这些决定之一是如何处理CSS。我一直是Bootstrap的忠实粉丝，但第一款Bootstrap 4 Alpha刚问世，他们放弃了Less for Sass。我讨厌萨斯！

Sass在2010年代中期的预处理程序大战中击败了Less，但在我看来，Less是更好的语言。它具有功能性和声明性，而不像Sass那样具有程序性和强制性，它具有Sass所没有的一项杀手级功能：类作为*mixins*。

您可以在Less中看到，任何现有的类都可以自动用作mixin。您不需要像在Sass中那样显式定义mixin。这使您可以轻松地从小小的实用程序类中创建较大的类抽象。

如果您 `@apply` 在Tailwind中使用过，这可能看起来很熟悉.....

```
// Padding utilities
.pad-t { padding-top: @spacer; }
.pad-r { padding-right: @spacer; }
.pad-b { padding-bottom: @spacer; }
.pad-l { padding-left: @spacer; }

.pad-x { .pad-l; .pad-r; }
.pad-y { .pad-t; .pad-b; }
.pad-a { .pad-x; .pad-y; }
.pad { .pad-x; .pad-y; }

// Component class
.btn-pad-lg {
  .pad-y-sm;
  .pad-x;
}
```

所以无论如何，回到摘要。通常我会用Bootstrap，但是Sass的东西对我来说就杀了它。我想继续使用Less。因此，当然，我唯一的选择是从头开始编写所有样式。

我花了大部分时间在这个最初的样式系统上。

我是建筑很引导，启发，有很多组件类一样的 `btn`，`card-list` 和 `radio-box`。但是它也有少量的实用程序类。

（顺便说一下，这是我们最终的结果，我认为即使在5年后它仍然看起来还不错！）

Have something to share? Create a Digest to post your own discoveries.

[Create New Digest](#)

Not right now

### The Dribbblisation of Design

 Ben Orenstein on  thoughtbot • 1 hour ago

### Testing Eloquent models with Faktory

 Taylor Otwell on  Laravel • 22 hours ago

### Faux OOP

 Ben Orenstein on  thoughtbot • 1 day ago

### Mocks aren't stubs

 Ben Orenstein on  thoughtbot • 2 days ago

### Laravel 5.2 released

 Taylor Otwell on  Laravel • 2 days ago


### Building RESTful APIs in Laravel

 Taylor Otwell on  Laravel • 2 days ago

#### Recommended Digests

 Basecamp  
David Heinemeier Hansson

[Subscribe](#)

 Boltmade  
Eric Roberts

[Subscribe](#)

无论如何，我们忙于其他项目，对这个想法失去了热情，最终所有想法都落在了旁项目的墓地中（就像大多数事情一样）。

除了样式表以外的所有内容。

在每个新项目中，我一直从Digest复制并粘贴所有Less文件，并以它们为起点，根据需要自定义它们以适应我正在构建的任何新设计。在我们放弃Digest之后，我一定将他们带到了至少4或5个其他项目中。

在复制样式时，我注意到了一些东西：这些实用程序（起初是简单的padding和margin实用程序）一直在增长和发展，涵盖了越来越多的CSS功能，而组件文件却越来越短。实用程序是唯一真正“可移植”的东西，而组件样式始终过于自以为是，无法在其他设计上重用。

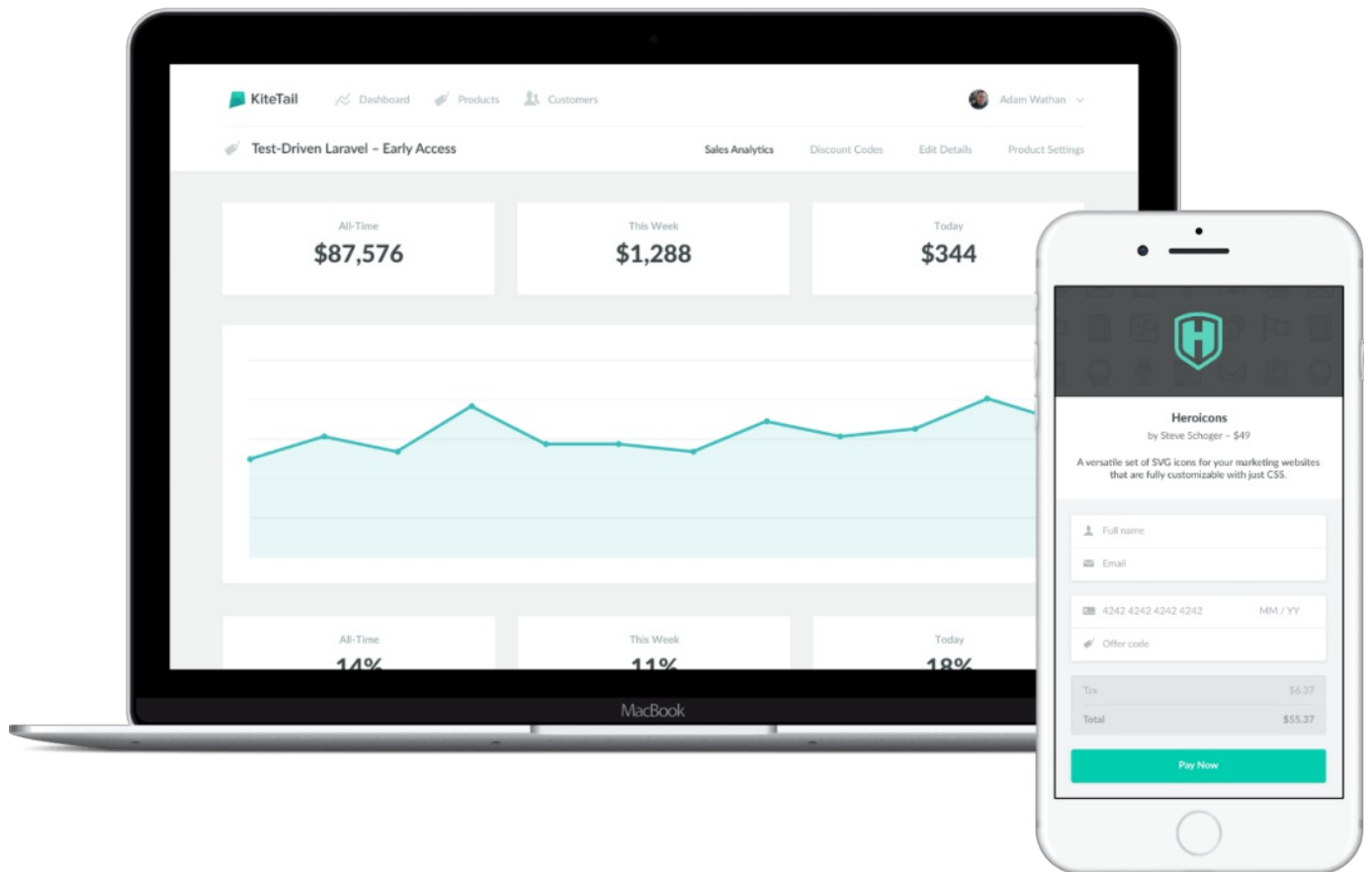
这是我真正开始将整个“实用程序至上”的事物识别为一种体系结构哲学的时候，而不是将实用程序视为只是有用的技巧，可以根据需要在我的HTML中随处使用。

## 被遗弃的第二项目

几年后快速发展，我和史蒂夫（Steve）开始研究KiteTail，它将成为以开发人员为中心，由Webhook驱动的就账平台：



# KiteTail



当时我们非常认真地对待这一风格，并且以旧的Digest样式为起点，我开始构建它，并尽力使这些样式尽可能“与项目无关”。

您可以观看我在YouTube上构建大量应用程序的过程，并且可以一直在其中看到所有这些实用程序样式：

在YouTube上观看“Building KiteTail”系列

现在，在这一点上，我不得不零意图保持任何形式的开源CSS框架的。我甚至没有想到，我一直在建造的东西对任何人来说都会很有趣。但是，源源不断的人们总是在问CSS：



**nerdalertdk** b 3 years ago

Please make an package of your Less, it looks bawler :D



3



REPLY

这是在公共场所工作的好处 - 如果我还没有现场直播我的工作，史蒂夫和我将永远不会建立Tailwind Labs业务（该业务在不到2年的时间内就实现了超过400万美元的收入）-被遗弃的项目。

无论如何，最终我想到了“也许我会开源这个小小的Less框架？”

在工作@kitetailapp少框架有点一边看真实的谎言今晚☐想我会以这足以擦亮到开源☐☐ [pic.twitter.com/3Y7xHgMtaC](https://pic.twitter.com/3Y7xHgMtaC)  
— Adam Wathan (@adamwathan) 2017年6月18日

## 开源

大约这段时间，一些人与我联系，以防他们可以以任何方式与我合作。其中一个人就是Stefan Bauer，如果我没记错的话，他就是建议像这样的前缀的人， `sm:font-bold` 而不是 `sm-font-bold` Tailwind的响应式实用程序的人。

我的好朋友Jonathan Reinink在这段时间也向我介绍了该框架，并说他将重新设计SaaS项目，并想尝试这个疯狂的实用程序，我一直在胡扯。

事实证明，这对于使框架真正良好至关重要，因为我们的项目具有完全不同的设计，并且需要Tailwind支持这两个项目。这是一个强大的强制功能，使其与项目无关。

顺便说一下，名称为Tailwind吗？因为我和Steve仍在努力推动这个想法，并且梦见它是一家我们将经营一天的出色公司，因此我希望将名字以某种方式与KiteTail绑定。我只是“tail\*”进入onelook.com，看看是否可以找到任何很酷的相关词。

回到故事-那是在2017年6月/7月左右，在接下来的2-3个月中，我和乔纳森（Jonathan）热情地致力于制作出足以开源的东西。

我在此过程中面临的挑战之一是，为了使Tailwind如我所愿地可配置，我必须认真地突破使用Less可能实现的界限，并编写一些真正神秘而可怕的东西：

```
30 lines (25 sloc) | 1003 Bytes
Raw Blame

1 .apply-match(@map; @search; @ruleset) {
2   @current: extract(extract(@map, 1), 1);
3   .apply-match(@map; @search; @current; @ruleset; 1);
4 }
5
6 .apply-match(@map; @search; @current; @ruleset; @i) when not (@search = @current) and (@i <= length(@map)) {
7   @current: extract(extract(@map, @i + 1), 1);
8   .apply-match(@map; @search; @current; @ruleset; @i + 1);
9 }
10
11 .apply-match(@map; @search; @current; @ruleset; @i) when (@search = @current) and (@i <= length(@map)) {
12   & {
13     @_match: extract(extract(@map, @i), 2);
14     @ruleset();
15   }
16 }
17
18 .extract-match(@map; @search) {
19   @current: extract(extract(@map, 1), 1);
20   .extract-match(@map; @search; @current; 1);
21 }
22
23 .extract-match(@map; @search; @current; @i) when not (@search = @current) and (@i <= length(@map)) {
24   @current: extract(extract(@map, @i + 1), 1);
25   .extract-match(@map; @search; @current; @i + 1);
26 }
27
28 .extract-match(@map; @search; @current; @i) when (@search = @current) and (@i <= length(@map)) {
29   @_match: extract(extract(@map, @i), 2);
30 }
```

就我所知，为这种事情编写测试套件并不实际，到了我甚至不再了解系统的地步，只好希望并祈祷解决一个问题没有实现。不介绍另一个。

大概是8月中旬，我的朋友David Hemphill建议我弄乱PostCSS，看看是否可以用JS编写框架。

我不知道使用PostCSS来构建Tailwind之类的东西是什么意思（我认为这仅限于autoprefixer用于其工作的那种东西），但是David引导我了解了一些高级技巧，例如使用自定义规则和自定义属性为“挂钩”，以插入生成的代码。

我开始弄乱它，立即对代码中的自信程度以及使用适当的编程语言可以做的令人惊奇的事情感到惊讶。

几周后，我在Full Stack Radio上谈论了这一切：

在Full Stack广播上收听“使用PostCSS构建CSS框架”

（顺便说一句，直到今天，我仍然感觉到Tailwind完全以一种从未想过的方式滥用了PostCSS，而且我暗中相信，每当安德烈·西特尼克（Andrey Sitnik）考虑到我们对他美丽的图书馆所做的工作时，他都会感到有些畏缩）

无论如何，在2017年万圣节之夜，我们在第一版中进行了最后润色，并在最初的文档中进行了总结：

当前@tailwindcss球队地位： [pic.twitter.com/6SnRytiLCY](https://pic.twitter.com/6SnRytiLCY)

— Adam Wathan (@adamwathan)，2017年10月31日

我们发布了它，并获得了许多积极的关注，即使对于v0.1.0：

☐ 神圣的烟雾@tailwindcss 0.1.0已经到来！☐ <https://t.co/zCaaNls0zp>

-Adam Wathan (@adamwathan) 2017年11月1日

几天后，Andrew Del Prete写了该框架历史上最重要的博客文章之一，向我介绍了PurgeCSS的奇妙世界。

阅读“使用 净化CSS PurgeCSS删除未使用的Tailwind CSS类”

经过大约一年的v0.x新发行，其中包含许多很酷的新功能以及不断发展的社区，我宣布我将全职使用Tailwind CSS。

## 在Tailwind上全职

我原本应该和一个朋友一起开始一个新的SaaS项目，但是在Refactoring UI成功（我和史蒂夫于2018年12月发行了一本书）以及Tailwind的发展之后，我知道我很后悔没有进一步推动它。

迄今为止，Tailwind CSS是我从事过的最具影响力的项目-感觉就像这几乎是我的“宇宙缩影”，并且不投入工作来克服它的想法使我感到震惊生病。

我很幸运能够在Refactoring UI上获得大笔资金，而且我知道有很多方法可以围绕框架本身构建商业产品（主题，UI工具包，课程等！），所以我决定去做。

我捣蛋清理了所有东西，然后应用我们学到的知识整理了一个正确的v1.0版本，该版本于2019年5月13日发布：

☐ 超级激动地宣布@tailwindcss v1.0终于发布了！

头向网站，并给它一个旋转☐☐ <https://t.co/zCaaNls0zp> [pic.twitter.com/NkpCN8Bq3q](https://pic.twitter.com/NkpCN8Bq3q)

-Adam Wathan (@adamwathan) 2019年5月13日

之后，我和史蒂夫（Steve）在一年余下的时间里努力研究到底是什么。我们原型化并丢弃了许多不同的想法，但最终决定采用现在的Tailwind UI。

这是2019年3月的第一个想法：

正在为我本周末计划与@steveschoger一起计划的@tailwindcss组件库/工作室项目制作原型☐

想想成百上千的完全响应式专业设计的组件，它们是预先构建的，因此您可以复制HTML并进行调整以品尝□□

有用吗？[pic.twitter.com/WobOoMGwxH](https://pic.twitter.com/WobOoMGwxH)

-Adam Wathan (@adamwathan) 2019年3月30日

我们花了几个月的时间在Tailwind UI上孜孜不倦地工作，直到在我们自己设定的截止日期之前连续工作了36个小时之后，我们终于在2020年2月发布了早期访问权限。

□废话Tailwind UI直播了！

您需要了解的有关抢先使用的所有信息都在新网站□□

（也有很多免费内容可供预览！）<https://t.co/CX0wtUgwGC>

-Adam Wathan (@adamwathan) 2020年2月26日

它取得了成功，超出了我们最野心的梦想（下周将突破200万美元的收入），因此，我们已经能够建立一支了不起的团队（布拉德·科恩斯，西蒙·弗拉奇里奥蒂斯和神秘的开发商#3）来继续前进Tailwind的未来向前发展。

事情只会从这里变得更加不可思议，我迫不及待地想将围绕我们的想法转变为新功能，新产品和新工具，以使Tailwind的体验在未来几年变得更好。

感谢您支持我们♥□

[阅读原文](#)