

Extracted from:

Modern CSS with Tailwind, Second Edition

Flexible Styling Without the Fuss

This PDF file contains pages extracted from *Modern CSS with Tailwind, Second Edition*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit <http://www.pragprog.com>.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printed versions; the content is otherwise identical.

Copyright © 2022 The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

The Pragmatic Bookshelf

Raleigh, North Carolina

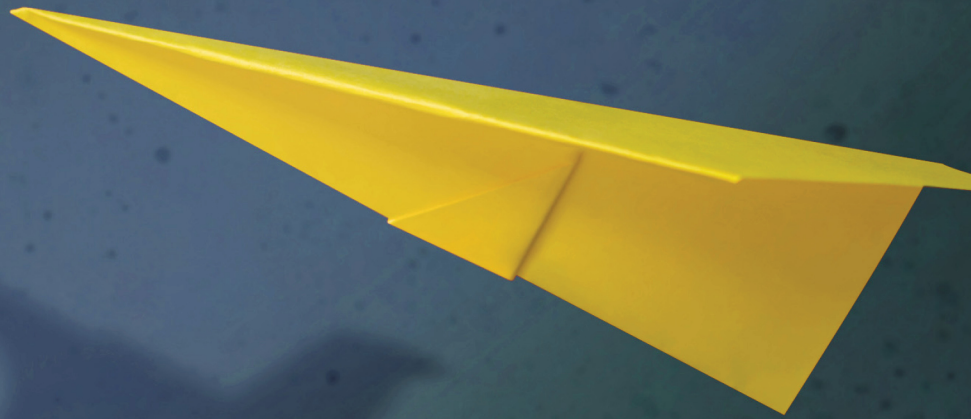
The
Pragmatic
Programmers

Pragmatic
expPress

Modern CSS with Tailwind

Second Edition

Flexible Styling Without the Fuss



Covers Tailwind 3.0

Noel Rappin
Edited by Katharine Dvorak

Modern CSS with Tailwind, Second Edition

Flexible Styling Without the Fuss

Noel Rappin

The Pragmatic Bookshelf

Raleigh, North Carolina



Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and The Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf, PragProg and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

For our complete catalog of hands-on, practical, and Pragmatic content for software developers, please visit <https://pragprog.com>.

The team that produced this book includes:

CEO: Dave Rankin

COO: Janet Furlow

Managing Editor: Tammy Coron

Development Editor: Katharine Dvorak

Copy Editor: L. Sakhi MacMillan

Layout: Gilson Graphics

Founders: Andy Hunt and Dave Thomas

For sales, volume licensing, and support, please contact support@pragprog.com.

For international rights, please contact rights@pragprog.com.

Copyright © 2022 The Pragmatic Programmers, LLC.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

ISBN-13: 978-1-68050-940-3

Encoded using the finest acid-free high-entropy binary digits.

Book version: P1.0—May 2022

Introduction

Many web developers underrate CSS.

Cascading Style Sheets (CSS) enable you to control the display of your information and enhance your page with visual effects. CSS is powerful, as a quick glance at a site like <http://www.csszengarden.com> shows. With CSS, you can do amazing things to the basic text and images on your site, and with a little bit of client-side code to add and remove CSS classes, you can do exponentially more.

CSS can also be hard to debug, complicated to write, and difficult to control.

But it doesn't have to be.

Enter Tailwind. Tailwind CSS—a “utility-first” CSS framework to “rapidly build modern websites without ever leaving your HTML”¹—can make the CSS for your site easier to control and debug. In this book, you'll dive into the Tailwind CSS framework, taking a look at its typography, page layout, responsive design, and more.

Why Tailwind?

Bootstrap or similar CSS frameworks provide CSS classes whose names describe the semantics how they are to be used, like “button” or “card” or “nav.” These classes tend to define a number of CSS styles together.

Tailwind is different.

Nearly all of the basic Tailwind utility classes are thin wrappers around a single CSS style setting, like using `m-4` to provide a margin: 1rem or `text-lg` to change the text size to `font-size: 1.125rem`.

For example, a button in the Bulma framework can be styled like this:

```
<button class="button is-primary">Click Me</button>
```

1. <https://tailwindcss.com>

But in Tailwind, you might use something more like this:

```
<button
  class="bg-green-500 text-white font-bold
        py-3 px-4 rounded-lg text-center">
  Click Me
</button>
```

In the Tailwind version, each individual style of the button—the green background, the white text, the bold font, the padding, the rounded corners, and the text centering—gets its own class in the class list.

Now, if you're like me, your first reaction to seeing that list of classes may be something along the lines of, and I quote, "ugh." It certainly takes some time to get used to having all those small class names in the HTML markup. If that's your reaction, I get it. All I ask is that you give it a chance and see how you feel after you've tried it out.

The Tailwind code is extremely explicit and makes it possible to understand the display simply by looking at the HTML markup. It works well with front-end frameworks that have an aesthetic of putting a bunch of CSS or JavaScript data in the HTML markup. If you want to package a collection of classes for reuse, Tailwind provides an `@apply` directive that you can use to build new CSS classes out of Tailwind's utilities, but it's recommended that you use features of your web programming language and framework to manage the duplication.

One advantage of the Tailwind setup is that it's extremely easy to prototype, iterate, and customize the display. If you want to change the horizontal padding on a button, you can do so by changing `px-4` to, say, `px-6`. You don't need to guess about the scope of the change or what other parts of your page might be affected. You can keep making small changes until you get the display just right. And you don't need to continually come up with names for CSS property combinations that might not be reused.

Another advantage is that Tailwind offers a set of modifiers that allow you to specify behavior in particular cases. For example, you could add a class such as `hover:bg-blue-500`, where the `hover` modifier causes the background color to be changed only when the user hovers over the button. Tailwind also provides a set of modifiers that allow you to specify different behaviors at different screen sizes. These modifiers are one reason why using a Tailwind class like `bg-blue-500` is better than using the document object model (DOM) style attribute directly, as in `style="background-color: #cddcdd"`.

And last but not least, a Tailwind app requires less CSS to be written, with most of the design coming from the composition of Tailwind utilities. This means you spend less time naming CSS and managing global CSS, allowing you to spend more effort on the actual display of your site. Tailwind makes it easy to make incremental changes, see the results, and understand the scope of your changes, which makes it especially useful when prototyping a new site.

About This Book

In this book, you're going to look at how to design web pages using Tailwind CSS version 3.0 and up. You'll start with the typography of individual elements, and then you'll get to "the box"—the rectangle of space each element takes up—and how to manipulate it. Once you have your elements in boxes, you'll take a look at page layout with flexbox or grids.

After that, you'll look at turning individual pages into full sites. You'll also look at common site-wide page layouts, managing a design on different screen sizes, and handling a site-wide amount of styles and CSS.

Tailwind has been evolving pretty quickly, so there's a good chance new features have been added since I wrote this. The Tailwind documentation includes pages for release notes and upgrade guides. (Sorry, the release notes' URLs change with each release, but they're linked from the main Tailwind documentation at <https://tailwindcss.com/docs>.) Check those out for the latest changes.

Who This Book Is For

To keep this book short and to the point, I've made some assumptions:

- I'm assuming you already know the basics of CSS syntax and concepts. This book focuses on Tailwind, not raw CSS. If you want to get better grounded in CSS and its quirks, you might want to try the zine *Hell Yes! CSS!* by Julia Evans.²
- I'm assuming you're able to access the Tailwind reference documentation.³ The Tailwind documentation is comprehensive and easy to navigate. This book isn't going to be a complete reference on all of Tailwind's features; instead, it'll focus on the most common ones and how to use them successfully.

2. <https://wizardzines.com/zines/css>

3. <https://tailwindcss.com/docs>

Running the Sample App

The sample code for this book is extremely simple. It consists of a few pages of static HTML linked to a static CSS page that was generated using the Tailwind CLI without using any framework or dynamic content.

To run the sample code, you need to download it from the book's page on the Pragmatic Bookshelf website.⁴ To view the code samples, just open any of the HTML pages in the `html` directory in a browser as a file. They will link to the `css/output.css` file, which has been pre-generated using the Tailwind CLI. (For more information about using the Tailwind CLI with that code, jump ahead to [Using the Sample Code, on page ?](#).)

Please note that the world has possibly changed since this book was released, and the distribution setup and commands for the Tailwind CLI has changed. If these instructions don't work for you, please check out the book forum⁵ for more discussion and support. Thanks!

4. <https://www.pragprog.com/titles/tailwind2>

5. <https://devtalk.com/books/modern-css-with-tailwind-second-edition/errata>