

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

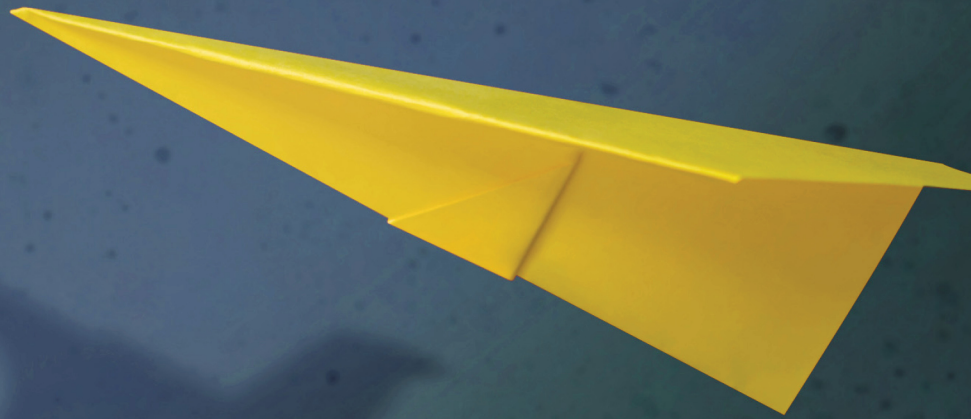
The  
Pragmatic  
Programmers

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](mailto:support@pragprog.com).

For international rights, please contact [rights@pragprog.com](mailto: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

Odds are your web application is displaying text to a reader. Much of the design of a website is about the placement, size, weight, and layout of text. In this chapter, we'll look at how Tailwind lets you control the display of text.

## Size and Shape

Perhaps the first thing you notice about text on a web page is its size and styling. Tailwind provides a series of utilities for each.

The effective default for text size is the `text-base` class, which defines the CSS properties `font-size: 1rem` and `line-height: 1.5rem`, meaning the font size for `text-base` is the same as the font size for the root element of your page, and the line height is 1.5 times that size. Tailwind provides a family of utilities, `text-{size}`, which includes two smaller steps, ten larger ones, and the base class, giving us thirteen sizes overall. Each step defines a font size and a line height, as listed in the [table on page 6](#).

This is our first encounter with one of Tailwind's explicit design goals, which is to provide a consistent set of steps for a potentially infinite set of values. With the `text-{size}` family of utilities, Tailwind makes it easy to keep sizing and spacing consistent throughout the site. As mentioned in [Utilities, on page ?](#), you can use square brackets to define an arbitrary value as the size, as in `text-[20px]`; you do need to include the unit as well as the number.

Font styling—your basic bold, italic, underline—is covered by a few different CSS properties, but from Tailwind's perspective, these styles are just utility classes: italic and not-italic, and underline and no-underline. You'd only use not-italic and no-underline if you want the text to have different characteristics under different states. This is usually tied to responsive behavior at different screen sizes, which we'll look at later in [Chapter 7, Responsive Design, on page ?](#). You can also use `overline` and `line-through`.

If you have underline, overline, or line-through specified, you can add an additional class to style the line. Your options are `decoration-solid`, `decoration-double`, `decoration-dotted`, `decoration-dashed`, and `decoration-wavy`, all of which basically do what they say they do. A pattern specifies the width of the decoration, `decoration-{width}`, where the default values are 0, 1, 2, 4, and 8 corresponding to width in pixels. There's also `decoration-auto` and `decoration-from-font`, or you can use an arbitrary measurement. A similar pattern, `underline-offset-{width}`, specifies the offset from the line and uses the same set of numerical values, plus `auto`. You can specify the color of the underline as well with the pattern `decoration-{color}`. (See the next section for what goes into the color placeholder.)

For bold fonts, CSS provides nine grades of boldness from 100 to 900; normal text is 400. Tailwind also provides nine utility classes, one for each grade:

Class	Font Size	Line Height
text-xs	0.75rem	1rem
text-sm	0.875rem	1.25rem
text-base	1rem	1.5rem
text-lg	1.125rem	1.75rem
text-xl	1.25rem	1.75rem
text-2xl	1.5rem	2rem
text-3xl	1.875rem	2.25rem
text-4xl	2.25rem	2.5rem
text-5xl	3rem	1
text-6xl	3.75rem	1
text-7xl	4.5rem	1
text-8xl	6rem	1
text-9xl	8rem	1

font-hairline

font-thin

font-light

font-normal

font-medium

font-semibold

font-bold

font-extrabold

font-black

Not all fonts will have distinct lettering at all weights, but commonly used web fonts should. I don't know why Tailwind doesn't use `font-weight-100`, which would seem to be more consistent with other naming. You can get it to do so by changing the configuration, though (see [Chapter 8, Customizing Tailwind, on page ?](#)). You can also put in an arbitrary value, as in `font-[1200]`.

You might also want to ensure the case of the text. For example, you might want a header to be all uppercase. Tailwind provides four utility classes for case, all of which wrap the CSS `text-transform` property to provide the behavior the utility name describes:

uppercase

lowercase

capitalize  
normal-case

With these in hand, we can start to build up styles for our actual headers. The following is, more or less, the default styling for a popular CSS framework's title and subtitle defaults:

```
<h1 class="text-3xl font-semibold">Title</h1>
<h2 class="text-xl">Subtitle</h2>
```

This gives us a title that's 1.875rem (30-point type) with a line height of 2.5rem (36 points) and semibold, and a subtitle that's 1.25rem (20 points) at normal weight and a line height of 1.75rem. I often like my headers to be a little more attention-grabbing, so I might do something like this:

```
<h1 class="text-4xl font-bold">Title</h1>
<h2 class="text-2xl font-semibold">Subtitle</h2>
<h3 class="text-lg font-medium italic">Header</h3>
```

This gives us a slightly bolder and bigger title and subtitle, plus a third-level header that's a little bit bigger than regular text, a little bit bolder, and also italic. I might also add some spacing around the headers; we'll look at how to do this in [Chapter 4, The Box, on page ?](#).

Remember that Tailwind's reset styles make it so that h1, h2, and so on have no default styling, so using those tags with Tailwind is a semantic note that the text is a header of some kind. It has no stylistic effect.

Right now, there's a good chance you're asking whether I'm seriously telling you it's a good idea to have to type text-lg font-medium italic every time you want a header. That's a lot of typing, it's on the cryptic side, and it's a lot of typing. See [Duplication, on page ?](#), for ideas on how to manage duplication in Tailwind.