

TailwindCSS 3 Fundamentals

Introducing TailwindCSS



Shawn Wildermuth
Microsoft MVP, Speaker and Instructor

@shawnwildermuth wilder minds.com



Overview



TailwindCSS

Find where it fits into your web apps
Learn by doing
You'll see it all



What You'll Learn

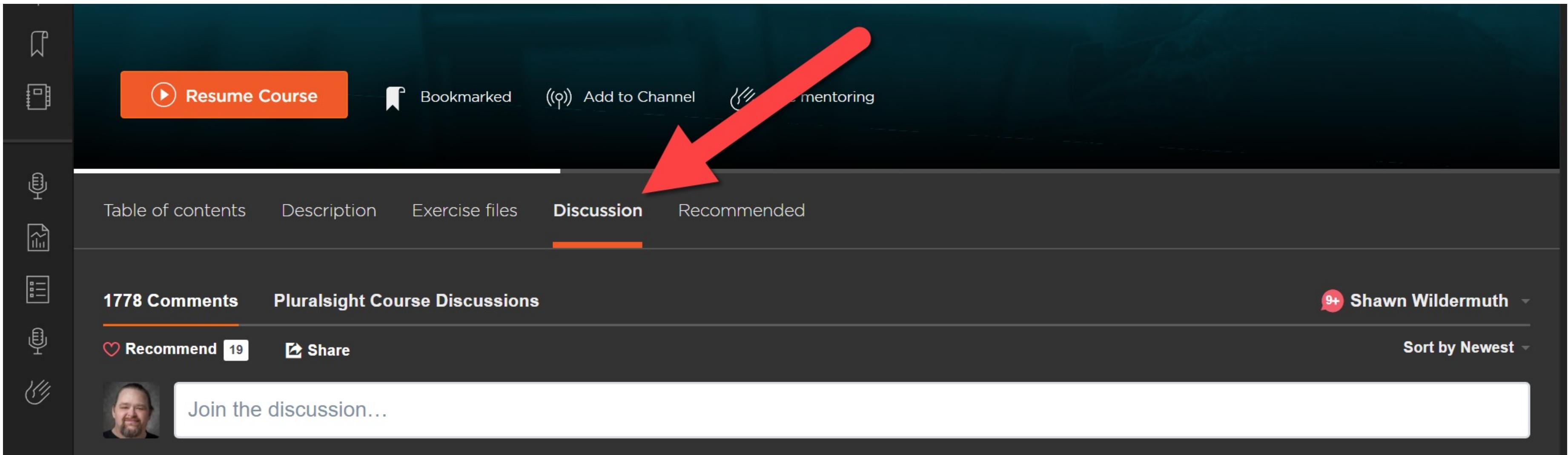


TailwindCSS

- Using Utility Classes
- Installation
- Styling Forms
- Integration with JS Frameworks
- Customizing TailwindCSS



Questions During the Course?



A screenshot of a Pluralsight course page for "Pluralsight Course Discussions". The page has a dark theme with orange and white accents. At the top, there's a navigation bar with icons for bookmarking, adding to a channel, and mentoring. Below that is a secondary navigation bar with links for "Table of contents", "Description", "Exercise files", "Discussion" (which is highlighted with an orange underline), and "Recommended". A large red arrow points from the question above to the "Discussion" link. Further down, there's a section for "1778 Comments" with a "Pluralsight Course Discussions" heading. On the left, there are icons for a microphone, a document, and a person. On the right, there's a profile picture for "Shawn Wildermuth" and a "Sort by Newest" dropdown. At the bottom, there's a text input field with the placeholder "Join the discussion...".



Version Check



Version Check



This version was created by using:
TailwindCSS 3



Version Check



This course is 100% applicable to:
TailwindCSS 3 or later



The Problem with CSS



Why is CSS Difficult?

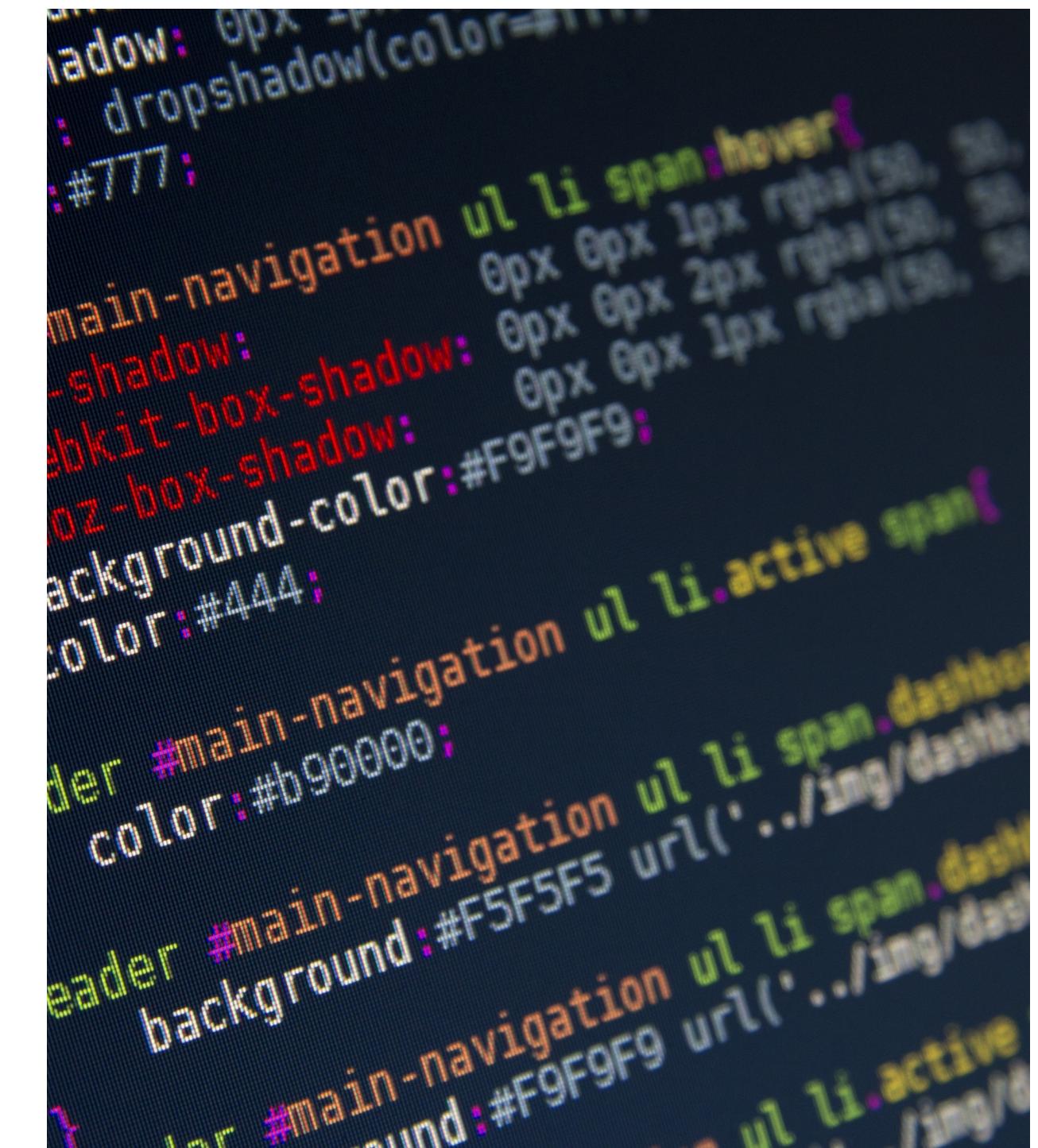
As CSS grows, it's hard to maintain

Authoring CSS rules is time consuming
Difficulty debugging CSS

Cascading rules are hard to understand

Copy/pasting CSS makes it worse

Selector Syntax is hard to grasp





There are no best practices that work

Semantic class names do not work

Designing for re-use leads to dogma

Block Element Modifiers are clumsy

CSS based on structure leads to bloat

Separation of concerns is a red herring

Component-based CSS is hard to manage

* Based on ideas from TailwindCSS's author, Adam Wathan
<https://shawnl.ink/awathan-sep-concerns>



Utility Classes

What are Utility Classes?

Single-responsibility design classes

Design components with utility classes

Can still enable re-use

But only when you really need it

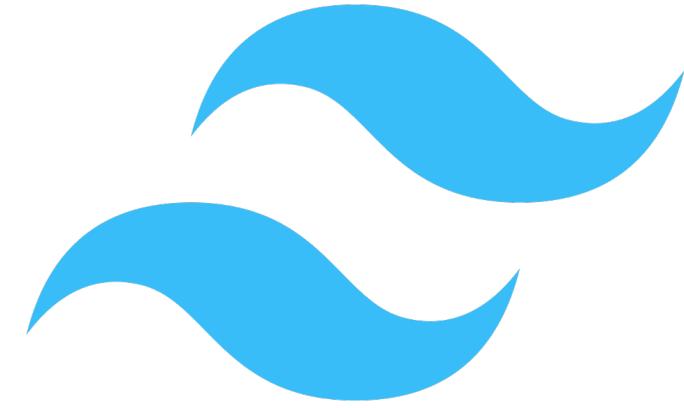
Better than inline styles

Promotes consistency



What is TailwindCSS?





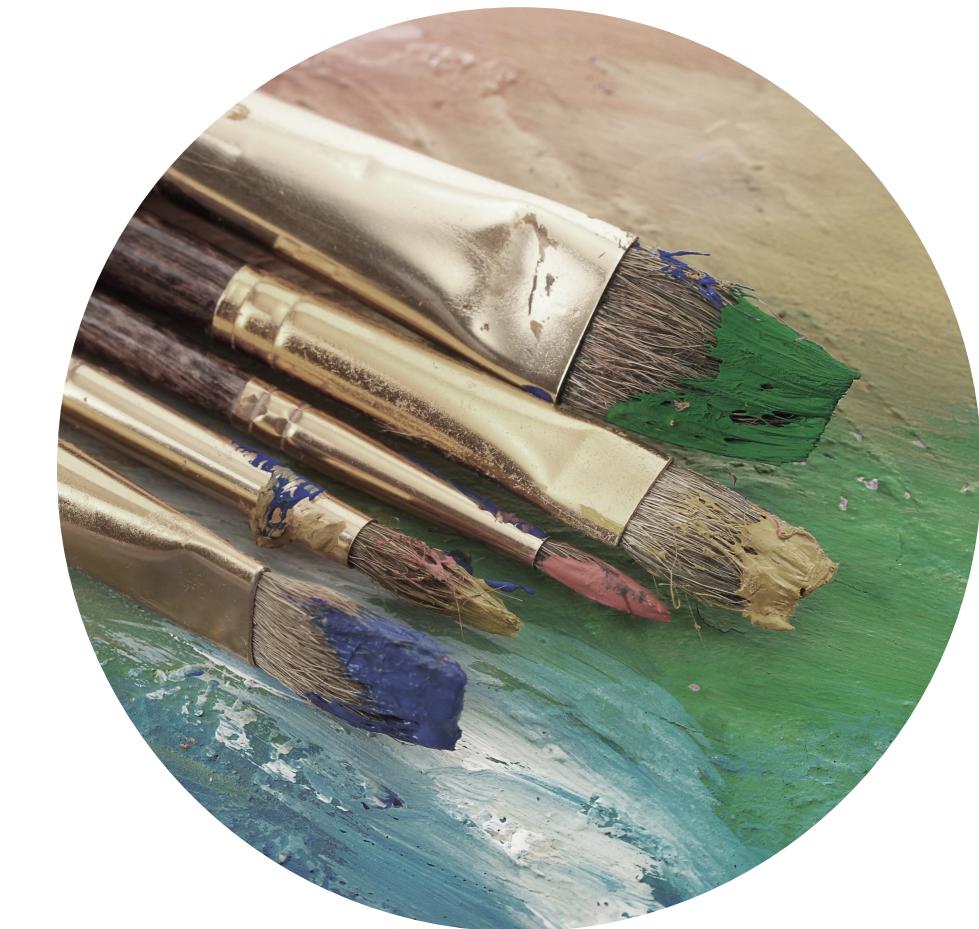
tailwindcss



Utility Classes



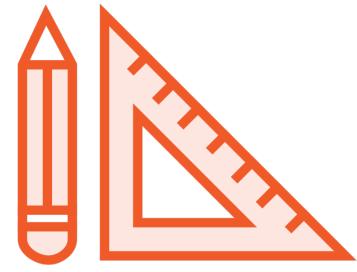
Efficient CSS Build



Extensible Theming



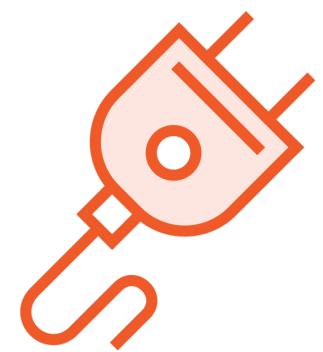
TailwindCSS Features



Consistent design system



Responsive thru and thru



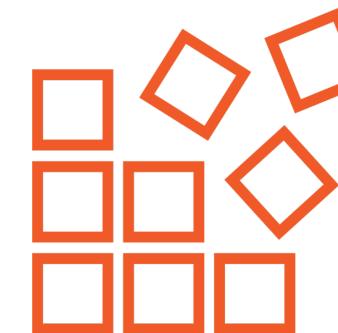
Powerful API



States and Variants



Eliminates unused CSS



Easy SPA integration





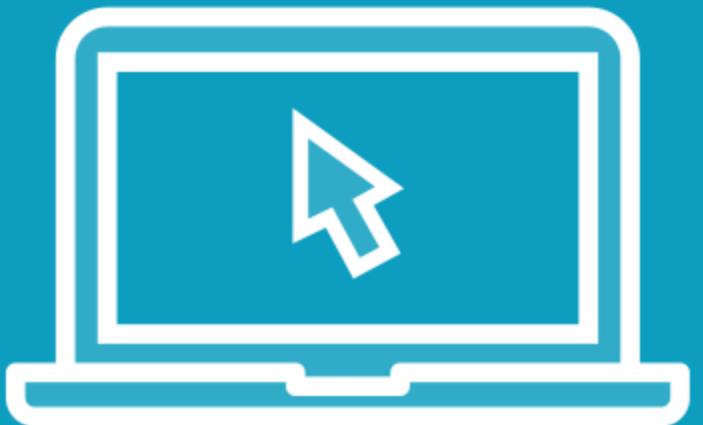
What about component frameworks?
(e.g. Bootstrap, Foundation, Bulma)
No need to give them up
There is no magic bullet
They can be used in conjunction



Getting Started



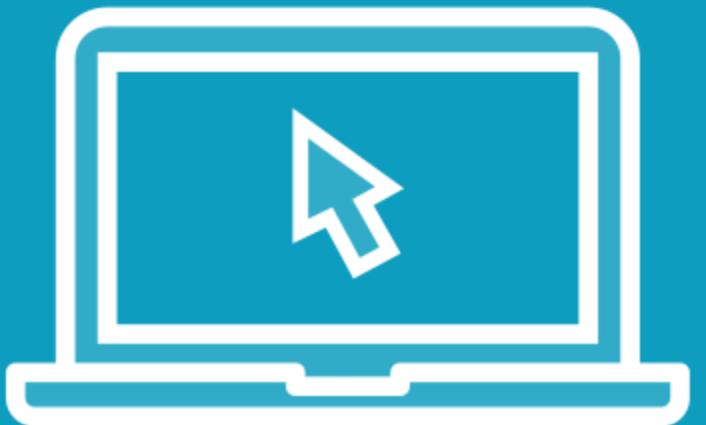
Demo



Using editors and IDEs



Demo



Starting out



What We've Learned



TailwindCSS wants to change the way you think about CSS



Utility classes do not mean you will never create components



Consistency is the central idea behind TailwindCSS



Up Next:

Using TailwindCSS

