

Learning tailwind

**Rapidly build modern websites
without ever leaving your HTML.**

A utility-first CSS framework packed with classes like `flex`, `pt-4`, `text-center` and `rotate-90` that can be composed to build any design, directly in your markup.

[Get started](#)

Quick search...

⌘ K

Initialize project

<https://tailwindcss.com/docs/guides/vite>

The screenshot shows a web page with a light blue grid background. At the top left is the Tailwind CSS logo (a blue wavy icon followed by the text 'tailwindcss'). Below the logo is a 'Getting Started' link in blue. The main title 'Install Tailwind CSS with Vite' is displayed in a large, bold, dark blue font. A subtitle 'Setting up Tailwind CSS in a Vite project.' follows in a smaller, gray font. At the bottom of the page, there is a section titled 'Install Tailwind CSS with Vite - Tailwind CSS' in bold black text, with a subtitle 'Setting up Tailwind CSS in a Vite project.' below it. The Tailwind CSS logo is also present at the bottom of this section.

Getting Started

Install Tailwind CSS with Vite

Setting up Tailwind CSS in a Vite project.

Install Tailwind CSS with Vite - Tailwind CSS

Setting up Tailwind CSS in a Vite project.

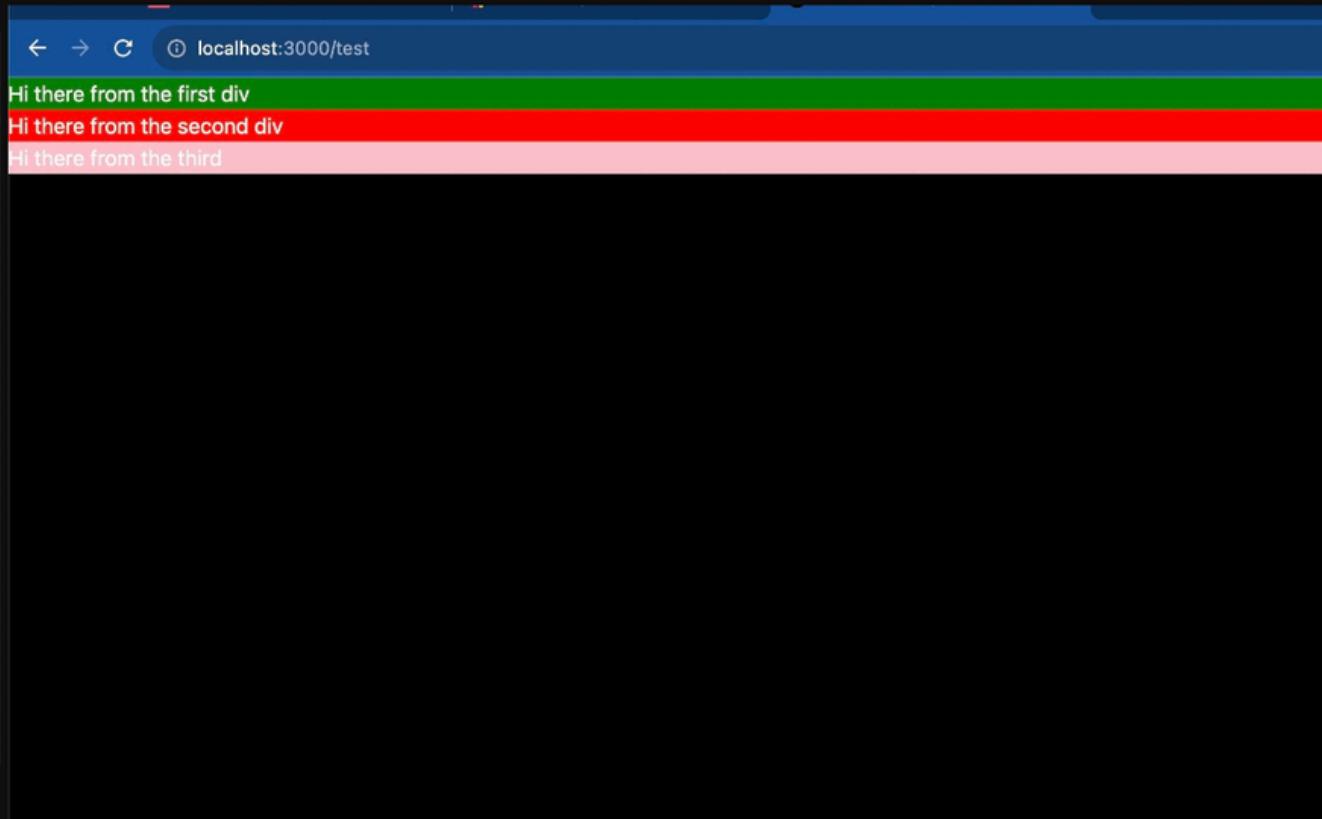
tailwindcss

Things to know in a frontend framework

1. Flex
2. Grids
3. Responsiveness
4. background color, text color, hover,

Flex

```
1
2  export default function Test() {
3    return <div>
4      <div style={{background: "green"}}>
5        |   Hi there from the first div
6      </div>
7      <div style={{background: "red"}}>
8        |   Hi there from the second div
9      </div>
10     <div style={{background: "pink"}}>
11       |   Hi there from the third
12     </div>
13   </div>
14 }
```



Flex

The image shows a split-screen environment. On the left is a code editor displaying a JavaScript file named 'Test.js'. The code defines a function 'Test' that returns a JSX component. This component consists of a main

element with a style object containing a 'display: flex' property. Inside this main

, there are three nested

elements, each with a different background color (green, red, and pink) and some descriptive text. A red rectangular highlight is placed over the 'display: flex' declaration in the main style object. On the right is a browser window titled 'localhost:3000/test'. The page displays the rendered output of the code, which is three separate

elements stacked vertically, each with its own background color and text content.

```
2 export default function Test() {
3   return <div style={{display: "flex"}}>
4     <div>
5       <div style={{background: "green">
6         |   Hi there from the first div
7       </div>
8       <div style={{background: "red">
9         |   Hi there from the second div
10      </div>
11      <div style={{background: "pink">
12        |   Hi there from the third
13      </div>
14    </div>
15  </div>
16 </div>
17 }
```

← → ⌂ ⌂ localhost:3000/test

Hi there from the first div Hi there from the second div Hi there from the third

Flex

The image shows a split-screen environment. On the left is a code editor with dark mode styling, displaying a file named 'test.tsx'. The code defines a component 'Test' that returns a flex container with three items. The first item has a green background, the second has a red background, and the third has a pink background. Each item contains the text 'Hi there from the [color] div'. A red box highlights the CSS rule 'justifyContent: "space-between"' in the style object of the first item's div. On the right is a web browser window titled 'localhost:3000/test' showing the rendered output of the code. Three separate boxes are displayed horizontally, each with its own background color and text.

```
pp / src / pages / < test.tsx > / test / < justifyContent >
export default function Test() {
  return <div style={{display: "flex", justifyContent: "space-between"}}>
    <div style={{background: "green"}}>
      | Hi there from the first div
    </div>
    <div style={{background: "red"}}>
      | Hi there from the second div
    </div>
    <div style={{background: "pink"}}>
      | Hi there from the third
    </div>
  </div>
}
```

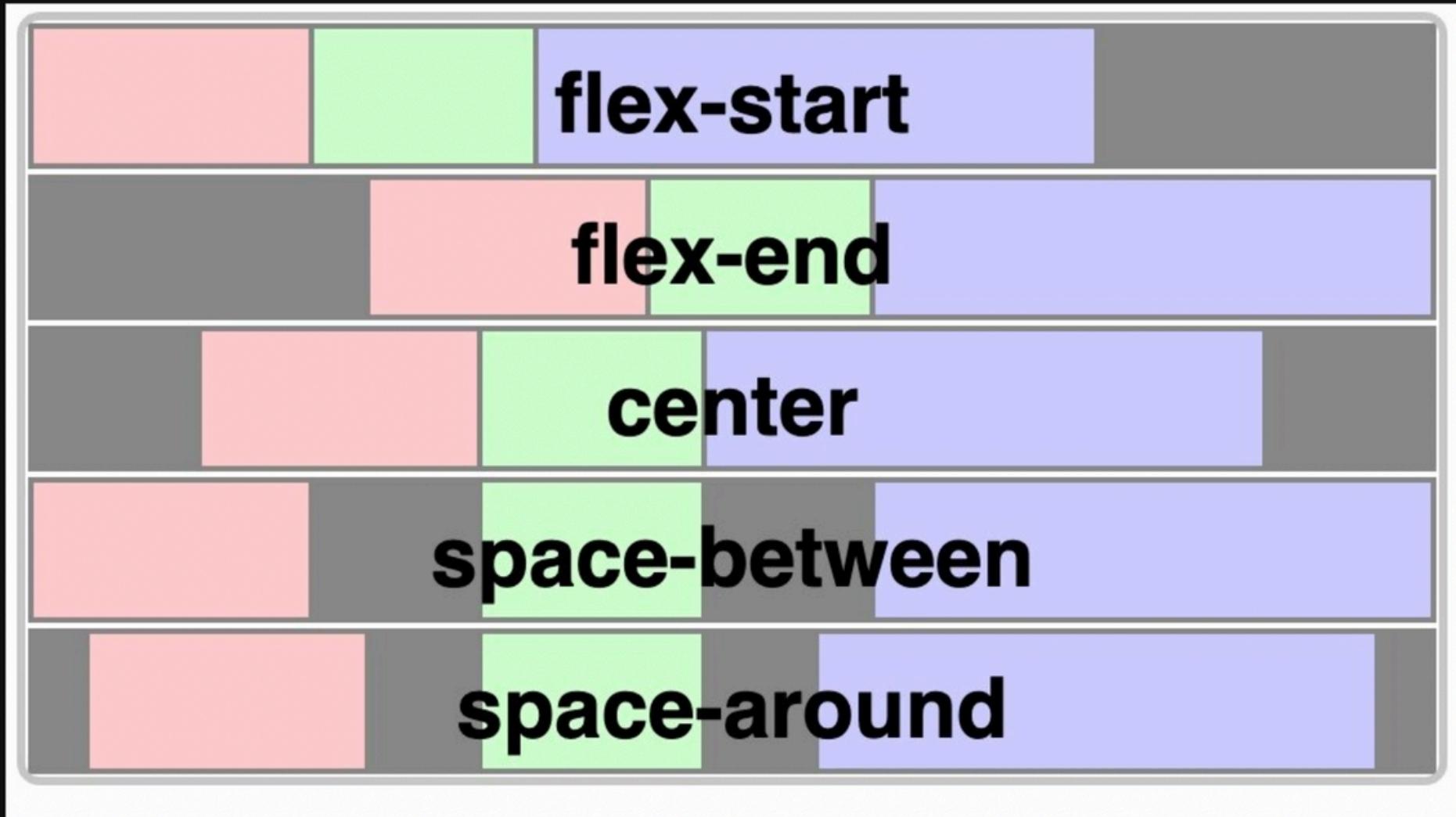
localhost:3000/test

Hi there from the first div

Hi there from the second div

Hi there from the third

Flex



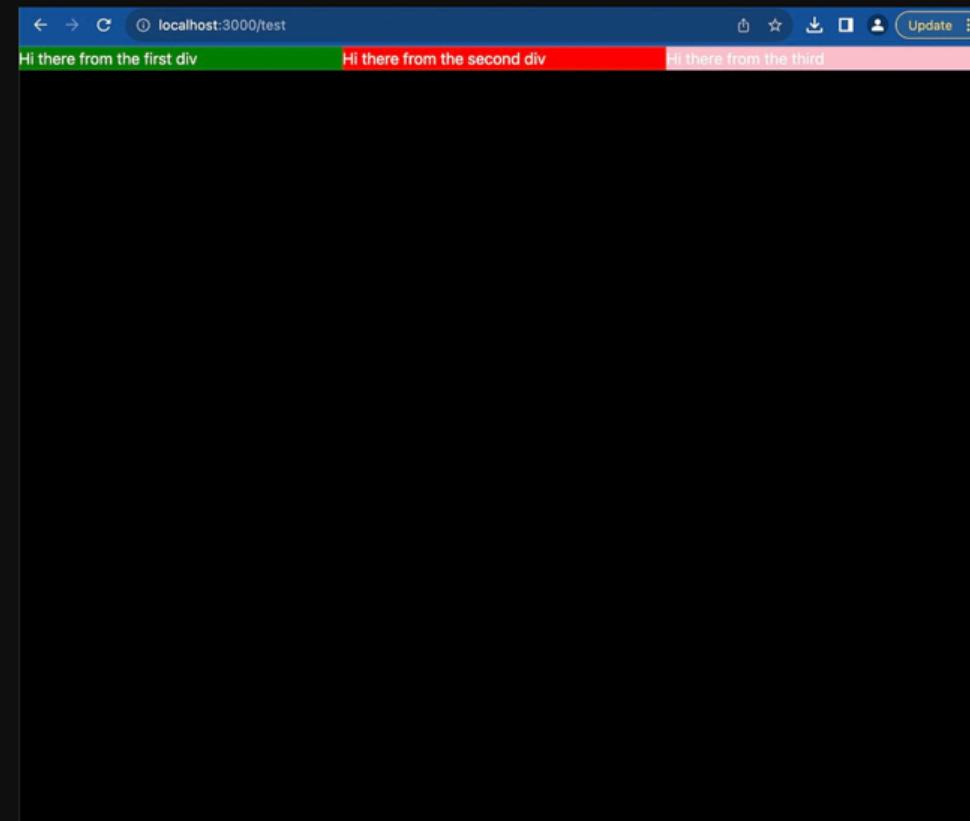
Grids

Grids in tw

Equal widths

my-app > src > pages > `TS` test.tsx > ...

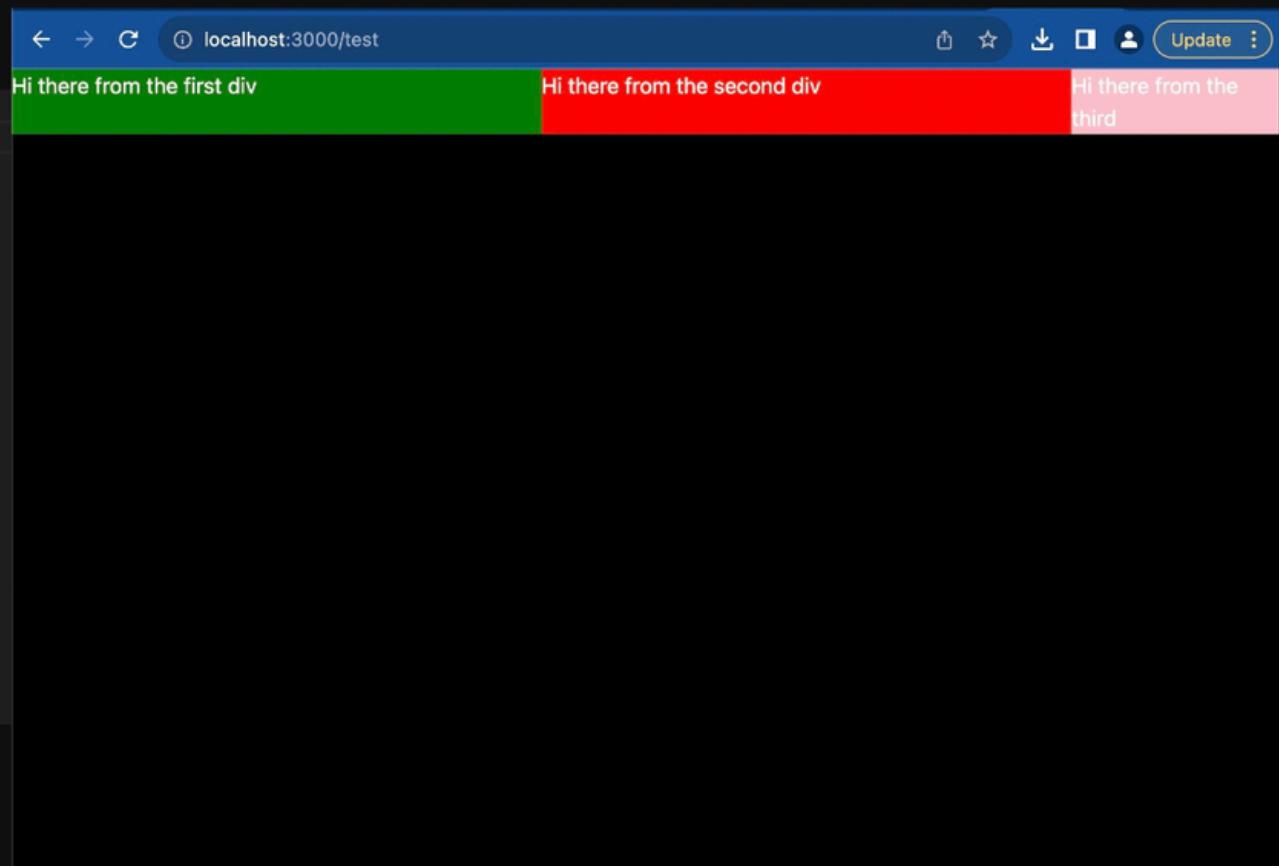
```
1
2  export default function Test() {
3      return <div className="grid grid-cols-3">
4          <div style={{background: "green"}}>
5              Hi there from the first div
6          </div>
7          <div style={{background: "red"}}>
8              Hi there from the second div
9          </div>
10         <div style={{background: "pink"}}>
11             Hi there from the third
12         </div>
13     </div>
14 }
```



Grids in tw

unequal widths

```
my-app > src > pages > TS test.tsx > ...  
1 |  
2 export default function Test() {  
3     return <div className="grid grid-cols-12">  
4         <div className="col-span-5" style={{background: "green"}}>  
5             Hi there from the first div  
6         </div>  
7         <div className="col-span-5" style={{background: "red"}}>  
8             Hi there from the second div  
9         </div>  
10        <div className="col-span-2" style={{background: "pink"}}>  
11            Hi there from the third  
12        </div>  
13    </div>  
14 }
```



Responsiveness

<https://tailwindcss.com/docs/responsive-design>

Breakpoint prefix	Minimum width	CSS
`sm`	640px	`@media (min-width: 640px) { ... }`
`md`	768px	`@media (min-width: 768px) { ... }`
`lg`	1024px	`@media (min-width: 1024px) { ... }`
`xl`	1280px	`@media (min-width: 1280px) { ... }`
`2xl`	1536px	`@media (min-width: 1536px) { ... }`

Mobile first

Working mobile-first

By default, Tailwind uses a mobile-first breakpoint system, similar to what you might be used to in other frameworks like Bootstrap.

What this means is that unprefixed utilities (like `uppercase`) take effect on all screen sizes, while prefixed utilities (like `md:uppercase`) only take effect at the specified breakpoint *and above*.

Targeting mobile screens

Where this approach surprises people most often is that to style something for mobile, you need to use the unprefixed version of a utility, not the `sm:` prefixed version. Don't think of `sm:` as meaning "on small screens", think of it as "at the small *breakpoint*".

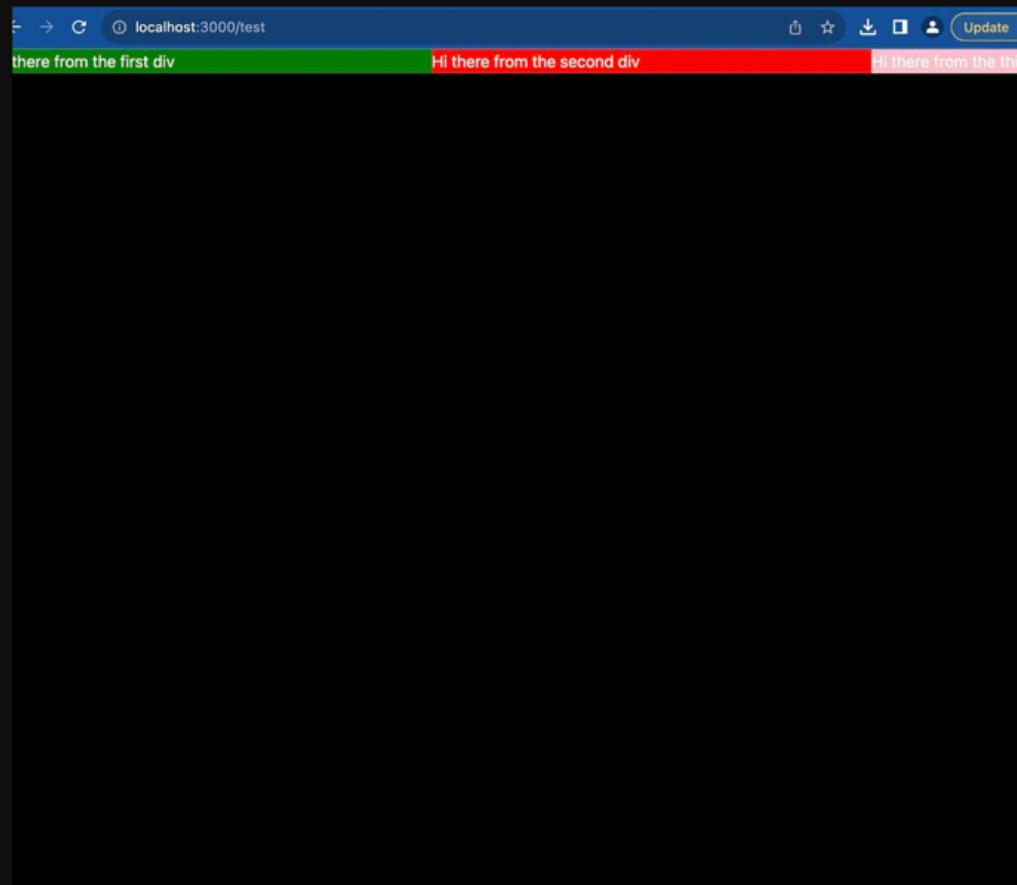
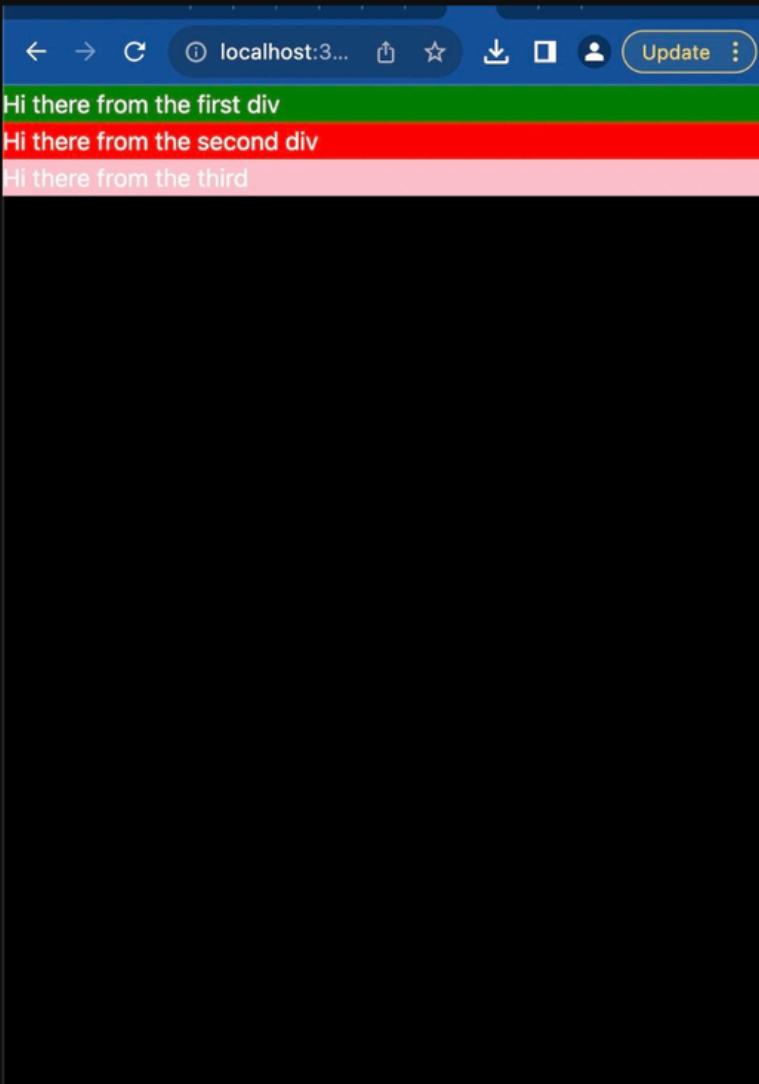
✗ Don't use `sm:` to target mobile devices

```
<!-- This will only center text on screens 640px and wider, not on small screens -->
<div class="sm:text-center"></div>
```

✓ Use unprefixed utilities to target mobile, and override them at larger breakpoints

```
<!-- This will center text on mobile, and left align it on screens 640px and wider -->
<div class="text-center sm:text-left"></div>
```

Use case



Code

```
my-app > src > pages > TS test.tsx > ⚡ Test

1
2  export default function Test() {
3      return <div className="grid grid-cols-12">
4          <div className="col-span-12 md:col-span-5" style={{background: "green"}}>
5              |   Hi there from the first div
6          </div>
7          <div className="col-span-12 md:col-span-5" style={{background: "red"}}>
8              |   Hi there from the second div
9          </div>
10         <div className="col-span-12 md:col-span-2" style={{background: "pink"}}>
11             |   Hi there from the third
12         </div>
13     </div>
14 }
```

Color system

Tailwind includes an expertly-crafted default color palette out-of-the-box that is a great starting point if you don't have your own specific branding in mind.

Slate	
	50 100 200 300 400 500 600 700 800 900 950 #f8fafc #f1f5f9 #e2e8f0 #cbd5e1 #94a3b8 #64748b #475569 #334155 #1e293b #0f172a #020617

Gray	
	50 100 200 300 400 500 600 700 800 900 950 #f9fafb #f3f4f6 #e5e7eb #d1d5db #9ca3af #6b7280 #4b5563 #374151 #1f2937 #111827 #030712

Zinc	
	50 100 200 300 400 500 600 700 800 900 950 #fafafa #f4f4f5 #e4e4e7 #d4d4d8 #a1a1aa #71717a #52525b #3f3f46 #27272a #18181b #09090b

Neutral	
	50 100 200 300 400 500 600 700 800 900 950 #fafafa #f5f5f5 #e5e5e5 #d4d4d4 #a3a3a3 #737373 #525252 #404040 #262626 #171717 #0a0a0a

Stone	
	50 100 200 300 400 500 600 700 800 900 950 #fafaf9 #f5f5f4 #e7e5e4 #d6d3d1 #a8a29e #78716c #57534e #44403c #292524 #1c1917 #0c0a09

Red	
	50 100 200 300 400 500 600 700 800 900 950 #fef7f2 #fee2e2 #fecaca #fcfa5a #f87171 #ef4444 #dc2626 #b91c1c #991b1b #7f1d1d #450a0a

Orange	
	50 100 200 300 400 500 600 700 800 900 950 #fff7ed #ffedad #fed7aa #fdb462 #fb923c #f97316 #ea580c #c2410c #9a3412 #7c2d12 #431407

Extending the config

Extending the colors

```
/** @type {import('tailwindcss').Config} */
export default {
  content: [
    './index.html',
    './src/**/*.{js,ts,jsx,tsx}',
  ],
  theme: {
    extend: {
      colors: {
        red: {
          500: "red"
        }
      },
      screens: {
        md: '768px', // Adjust this to your desired `md` breakpoint
        lg: '1024px', // Adjust this to your desired `lg` breakpoint
      },
    },
    plugins: []
  }
}
```

Changing the breakpoints

Backgrounds/font colors

The image shows a code editor with two side-by-side snippets of JSX code. Both snippets define a function named `Test` which returns a single `<div>` element containing the text "Hi there from the first div".

The left snippet uses inline styles:

```
1 export default function Test() {  
2   return <div style={{background: "green", color: 'red'}}>  
3     | Hi there from the first div  
4   </div>  
5 }
```

The right snippet uses class names:

```
1 app / src / pages / ts test.tsx / ↗ test  
2  
3 export default function Test() {  
4   return <div className="bg-green-500 text-red-500" >  
5     | Hi there from the first div  
6   </div>  
7 }
```

Font size

```
y-app > src > pages > ts test.tsx > ...
1
2 export default function Test() {
3   return <div className="bg-green-500 text-red-500 text-xs" >
4     Hi there from the first div
5   </div>
6 }
```

Typography

Font Size

Utilities for controlling the font size of an element.

Class	Properties
text-xs	font-size: 0.75rem; /* 12px */ line-height: 1rem; /* 16px */
text-sm	font-size: 0.875rem; /* 14px */ line-height: 1.25rem; /* 20px */
text-base	font-size: 1rem; /* 16px */ line-height: 1.5rem; /* 24px */
text-lg	font-size: 1.125rem; /* 18px */ line-height: 1.75rem; /* 28px */
text-xl	font-size: 1.25rem; /* 20px */ line-height: 1.75rem; /* 28px */
text-2xl	font-size: 1.5rem; /* 24px */ line-height: 1.75rem; /* 32px */
text-3xl	font-size: 1.75rem; /* 28px */ line-height: 1.75rem; /* 32px */
text-4xl	font-size: 2rem; /* 32px */ line-height: 1.75rem; /* 32px */
text-5xl	font-size: 2.25rem; /* 36px */ line-height: 1.75rem; /* 36px */
text-6xl	font-size: 2.5rem; /* 40px */ line-height: 1.75rem; /* 40px */

Basic usage

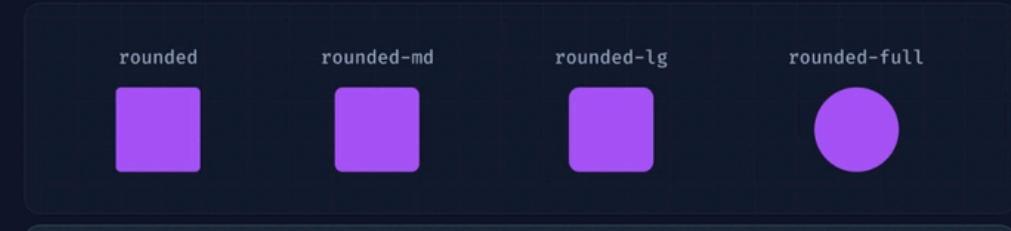
Border radius

Class	Properties
	border-end-start-radius: 0.25rem; /* 4px */
rounded-s-md	border-start-start-radius: 0.375rem; /* 6px */ border-end-start-radius: 0.375rem; /* 6px */
rounded-s-lg	border-start-start-radius: 0.5rem; /* 8px */ border-end-start-radius: 0.5rem; /* 8px */
rounded-s-xl	border-start-start-radius: 0.75rem; /* 12px */ border-end-start-radius: 0.75rem; /* 12px */
rounded-s-2xl	border-start-start-radius: 1rem; /* 16px */ border-end-start-radius: 1rem; /* 16px */
rounded-s-3xl	border-start-start-radius: 1.5rem; /* 24px */ border-end-start-radius: 1.5rem; /* 24px */

Basic usage

Rounded corners

Use utilities like `rounded-sm`, `rounded`, or `rounded-lg` to apply different border radius sizes to an element.



Vanilla to tailwind

```
/-app > src > pages > TS test.tsx > TestNative
1
2 // With tailwind
3 export default function TestTw() {
4     return <div className="bg-green-500 text-red-500 text-xs rounded" >
5         |   Hi there from the first div
6     </div>
7 }
8
9 // Without tailwind
10 export function TestNative() {
11     return <div style={{background: "green", color: "red", fontSize: "12px", borderRadius: "4px"}}>
12         |   Hi there from the first div
13     </div>
14 }
```

Project

Screen #1



Verify Your Age

Please confirm your birth year. This data will not be stored.

Your Birth Year

Continue

Screen #1



Verify Your Age

Please confirm your birth year. This data will not be stored.

27 July 2002

Continue

Screen #2



Let's Get Started

Email Id

Continue

Screen #2



Let's Get Started

prabhleen@gmail.com

Continue

Screen #3



Check Your Email For A Code

Please enter the verification code sent to your email id prabhleen@gmail.com



Verify

Can't find the email? Click [here](#) to resend.

Screen #3



Check Your Email For A Code

Please enter the verification code sent to your email id prabhleen@gmail.com

1 1 2 2 3 3

⌚ 09:32

Verify

Can't find the email? Click [here](#) to resend.

Take home assignment



Home



Webinars



Billing



User Management



Settings

**Prabhleen Kaur**

prabhleen@gmail.com

9899999882

Delhi, India

Monday, 14 October

Good morning, Prabhleen! 🙌

Monday, 14 October 2024



11:30 AM Live UX Webinar

11:30 AM Upcoming My first Webinar

11:30 AM Upcoming Important Webinar

11:30 AM Upcoming Webinar 1



Schedule a Webinar



Join a Webinar



Open Recordings