



Concepts de base

Conception réactive

Utilisation de variantes d'utilitaires réactifs pour créer des interfaces utilisateur adaptatives.

Chaque classe utilitaire dans Tailwind peut être appliquée de manière conditionnelle à différents points d'arrêt, ce qui facilite la création d'interfaces réactives complexes sans jamais quitter votre HTML.

Tout d'abord, assurez-vous d'avoir ajouté la [balise meta viewport](#) à votre document :`<head>`

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Ensuite, pour ajouter un utilitaire mais ne le faire prendre effet qu'à un certain point d'arrêt, tout ce que vous avez à faire est de préfixer l'utilitaire avec le nom du point d'arrêt, suivi du caractère :`:`

```
<!-- Width of 16 by default, 32 on medium screens, and 48 on large screens -->

```

Par défaut, il existe cinq points d'arrêt, inspirés des résolutions courantes des appareils :

| Préfixe de point d'arrêt | Largeur minimale | CSS |
|--------------------------|------------------|--------------------------------------|
| `sm` | 640px | `@media (min-width: 640px) { ... }` |
| `md` | 768 pages | `@media (min-width: 768px) { ... }` |
| `lg` | 1024px | `@media (min-width: 1024px) { ... }` |

| Préfixe de point d'arrêt | Largeur minimale | CSS |
|--------------------------|------------------|--------------------------------------|
| `xl` | 1280px | `@media (min-width: 1280px) { ... }` |
| `2xl` | 1536px | `@media (min-width: 1536px) { ... }` |

Cela fonctionne pour **toutes les classes utilitaires du framework**, ce qui signifie que vous pouvez modifier littéralement n'importe quoi à un point d'arrêt donné, même des choses comme l'espacement des lettres ou les styles de curseur.

Voici un exemple simple d'un composant de page marketing qui utilise une mise en page empilée sur les petits écrans et une mise en page côte à côte sur les grands écrans :



Company retreats

Incredible accommodation for your team

Looking to take your team away on a retreat to enjoy awesome food and take in some sunshine? We have a list of places to do just that.

```
<div class="max-w-md mx-auto bg-white rounded-xl shadow-md overflow-hidden n
  <div class="md:flex">
    <div class="md:shrink-0">
      
      <div class="uppercase tracking-wide text-sm text-indigo-500 font-semik
      <a href="#" class="block mt-1 text-lg leading-tight font-medium text-k
```

```
<p class="mt-2 text-slate-500">Looking to take your team away on a ret
</div>
</div>
</div>
```

Here's how the example above works:

- By default, the outer is `display: block`, but by adding the utility, it becomes `display: flex` on medium screens and larger. `md:flex`
- When the parent is a flex container, we want to make sure the image never shrinks, so we've added `md:shrink-0` to prevent shrinking on medium screens and larger. Technically we could have just used `md:shrink` since it would do nothing on smaller screens, but since it only matters on screens, it's a good idea to make that clear in the class name. `md:shrink-0`
- On small screens the image is automatically full width by default. On medium screens and up, we've constrained the width to a fixed size and ensured the image is full height using `md:h-full md:w-48`

We've only used one breakpoint in this example, but you could easily customize this component at other sizes using the `sm`, `lg`, `xl`, or responsive prefixes as well. `sm:lg:xl:2xl`

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. `md:uppercase`

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 prefixed version. Don't think of it as meaning "on small screens", think of it as "at the small *breakpoint*". `sm:` `sm:`

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

```
<!-- 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>
```

For this reason, it's often a good idea to implement the mobile layout for a design first, then layer on any changes that make sense for screens, followed by screens, etc. `sm` `md`

Targeting a breakpoint range

By default, styles applied by rules like `md:flex` will apply at that breakpoint and stay applied at larger breakpoints.

If you'd like to apply a utility *only* when a specific breakpoint range is active, stack a responsive modifier like `md` with a modifier to limit that style to a specific range: `md:max-xl`

```
<div class="md:max-xl:flex">
  <!-- ... -->
</div>
```

Tailwind generates a corresponding modifier for each breakpoint, so out of the box the following modifiers are available: `max-sm`

| Modifier | Media query |
|------------------------|---|
| <code>`max-sm`</code> | <code>`@media not all and (min-width: 640px) { ... }`</code> |
| <code>`max-md`</code> | <code>`@media not all and (min-width: 768px) { ... }`</code> |
| <code>`max-lg`</code> | <code>`@media not all and (min-width: 1024px) { ... }`</code> |
| <code>`max-xl`</code> | <code>`@media not all and (min-width: 1280px) { ... }`</code> |
| <code>`max-2xl`</code> | <code>`@media not all and (min-width: 1536px) { ... }`</code> |

Targeting a single breakpoint

To target a single breakpoint, target the range for that breakpoint by stacking a responsive modifier like with the modifier for the next breakpoint: ``md`` ``max-*``

```
<div class="md:max-lg:flex">
  <!-- ... -->
</div>
```

Read about [targeting breakpoint ranges](#) to learn more.

Using custom breakpoints

Customizing your theme

You can completely customize your breakpoints in your file: ``tailwind.config.js``

`tailwind.config.js`

```
/** @type {import('tailwindcss').Config} */
module.exports = {
  theme: {
    screens: {
      'tablet': '640px',
      // => @media (min-width: 640px) { ... }

      'laptop': '1024px',
```

```
// => @media (min-width: 1024px) { ... }

'desktop': '1280px',
// => @media (min-width: 1280px) { ... }
},
}
}
```

Learn more in the [customizing breakpoints documentation](#).

Arbitrary values

If you need to use a one-off breakpoint that doesn't make sense to include in your theme, use the `min` or `max` modifiers to generate a custom breakpoint on the fly using any arbitrary value. `min-[320px]:text-center` `max-[600px]:bg-sky-300`

```
<div class="min-[320px]:text-center max-[600px]:bg-sky-300">
  <!-- ... -->
</div>
```

Learn more about arbitrary value support in the [arbitrary values](#) documentation.

[◀ Hover, Focus, and Other States](#)

[Dark Mode ▶](#)