

Layout

Tailwind CSS Container

A component for fixing an element's width to the current breakpoint.

Class	Breakpoint	Properties
container	None	width: 100%;
	sm (40rem ~ 640px)	max-width: 40rem;
	md (48rem ~ 768px)	max-width: 48rem;
	lg (64rem ~ 1024px)	max-width: 64rem;
	xl (80rem ~ 1280px)	max-width: 80rem;
	2xl (96rem ~ 1536px)	max-width: 96rem;

Basic usage

Using the container

The `container` class sets the `max-width` of an element to match the `min-width` of the current breakpoint. This is useful if you'd prefer to design for a fixed set of screen sizes instead of trying to accommodate a fully fluid viewport.

Note that unlike containers you might have used in other frameworks, **Tailwind's container does not center itself automatically and does not have any built-in horizontal padding.**

To center a container, use the `mx-auto` utility:

```
<div class="container mx-auto">
```

```
  .      -->
```

On this page

```
</div>
```

To add horizontal padding, use the ``px-{size}`` utilities:

```
<div class="container mx-auto px-4">
  <!-- ... -->
</div>
```

If you'd like to center your containers by default or include default horizontal padding, see the [customization options](#) below.

Applying conditionally

Responsive variants

The ``container`` class also includes responsive variants like ``md:container`` by default that allow you to make something behave like a container at only a certain breakpoint and up:

```
<!-- Full-width fluid until the `md` breakpoint, then lock to container -->
<div class="md:container md:mx-auto">
  <!-- ... -->
</div>
```

Customizing

Centering by default

To center containers by default, set the ``margin-inline`` CSS rule to ``auto`` in the ``@utility container`` directive of your config file:

```
@utility container {
  margin-inline: auto;
}
```

Adding horizontal padding

To add horizontal padding by default, specify the amount of padding you'd like using the ``ne`` CSS rule in the ``@utility container`` directive of your config file:

```
@utility container {  
  padding-inline: 2rem;  
}
```

If you want to specify a different padding amount for each breakpoint, use an object to provide a `default` value and any breakpoint-specific overrides:

```
:root {  
  --container-padding: 1rem;  
}  
  
@media (min-width: var(--breakpoint-sm)) {  
  :root {  
    --container-padding: 2rem;  
  }  
}  
  
@media (min-width: var(--breakpoint-lg)) {  
  :root {  
    --container-padding: 4rem;  
  }  
}  
  
@media (min-width: var(--breakpoint-xl)) {  
  :root {  
    --container-padding: 5rem;  
  }  
}  
  
@media (min-width: var(--breakpoint-2xl)) {  
  :root {  
    --container-padding: 6rem;  
  }  
}  
  
@utility container {  
  padding-inline: var(--container-padding);  
}
```