About

Placeholders can be used to enhance the experience of your application. They're built only with HTML and CSS, meaning you don't need any JavaScript to create them. You will, however, need some custom JavaScript to toggle their visibility. Their appearance, color, and sizing can be easily customized with our utility classes.

Example

In the example below, we take a typical card component and recreate it with placeholders applied to create a "loading card". Size and proportions are the same between the two.

{{< placeholder width="100%" height="180" class="card-img-top" text="false" background="#20c997" >}}
Card title

Some quick example text to build on the card title and make up the bulk of the card's content.

Go somewhere
{{< placeholder width="100%" height="180" class="card-img-top" text="false" >}}

```
<div class="card">
  <img src="..." class="card-img-top" alt="...">

  <div class="card-body">
    <h5 class="card-title">Card title</h5>
    <p class="card-text">Some quick example text to build on the card title and make
up the bulk of the card's content.</p>
    <a href="#" class="btn btn-primary">Go somewhere</a>
  </div>
</div>

<div class="card" aria-hidden="true">
  <img src="..." class="card-img-top" alt="...">
  <div class="card-body">
    <h5 class="card-title placeholder-glow">
      <span class="placeholder col-6"></span>
    </h5>
    <p class="card-text placeholder-glow">
      <span class="placeholder col-7"></span>
      <span class="placeholder col-4"></span>
      <span class="placeholder col-4"></span>
      <span class="placeholder col-6"></span>
      <span class="placeholder col-8"></span>
    </p>
    <a href="#" tabindex="-1" class="btn btn-primary disabled placeholder col-6">
</a>
  </div>
</div>
```

How it works

Create placeholders with the `.placeholder` class and a grid column class (e.g., `.col-6`) to set the `width`. They can replace the text inside an element or be added as a modifier class to an existing component.

We apply additional styling to `.btn`s via `::before` to ensure the `height` is respected. You may extend this pattern for other situations as needed, or add a `&nbsp;` within the element to reflect the height when actual text is rendered in its place.

{{< example >}}

{{< /example >}}

{{< callout info >}} The use of `aria-hidden="true"` only indicates that the element should be hidden to screen readers. The *loading* behavior of the placeholder depends on how authors will actually use the placeholder styles, how they plan to update things, etc. Some JavaScript code may be needed to *swap* the state of the placeholder and inform AT users of the update. {{< /callout >}}

Width

You can change the `width` through grid column classes, width utilities, or inline styles.

{{< example >}} {{< /example >}}

Color

By default, the `placeholder` uses `currentColor`. This can be overridden with a custom color or utility class.

{{< example >}} {{< placeholders.inline >}} {{- range (index $.Site.Data "theme-colors") }} {{- end -}} {{< /placeholders.inline >}} {{< /example >}}

Sizing

The size of `.placeholder`s are based on the typographic style of the parent element. Customize them with sizing modifiers: `.placeholder-lg`, `.placeholder-sm`, or `.placeholder-xs`.

{{< example >}} {{< /example >}}

Animation

Animate placeholders with `.placeholder-glow` or `.placeholder-wave` to better convey the perception of something being *actively* loaded.

{{< example >}}

{{< /example >}}

Sass

Variables

{{< scss-docs name="placeholders" file="scss/_variables.scss" >}}