

Carousel

Contents

How it works	1
Example	2
Slides only	2
With controls	2
With indicators	3
With captions	3
Crossfade	4
Individual <code>.carousel-item</code> interval	4
Disable touch swiping	5
Dark variant	5
Custom transition	6
Sass	6
Variables	6
Usage	6
Via data attributes	6
Via JavaScript	7
Options	7
Methods	8
Events	9

How it works

The carousel is a slideshow for cycling through a series of content, built with CSS 3D transforms and a bit of JavaScript. It works with a series of images, text, or custom markup. It also includes support for previous/next controls and indicators.

In browsers where the Page Visibility API is supported, the carousel will avoid sliding when the webpage is not visible to the user (such as when the browser tab is inactive, the browser window is minimized, etc.).

```
{{< callout info >}} {{< partial "callout-info-prefersreducedmotion.md" >}}  
{{< /callout >}}
```

Please be aware that nested carousels are not supported, and carousels are generally not compliant with accessibility standards.

Example

Carousels don't automatically normalize slide dimensions. As such, you may need to use additional utilities or custom styles to appropriately size content. While carousels support previous/next controls and indicators, they're not explicitly required. Add and customize as you see fit.

The `.active` class needs to be added to one of the slides otherwise the carousel will not be visible. Also be sure to set a unique `id` on the `.carousel` for optional controls, especially if you're using multiple carousels on a single page. Control and indicator elements must have a `data-bs-target` attribute (or `href` for links) that matches the `id` of the `.carousel` element.

Slides only

Here's a carousel with slides only. Note the presence of the `.d-block` and `.w-100` on carousel images to prevent browser default image alignment.

```
{{< example >}}

<div class="carousel-item active">
  {{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="lightblue" >}}
</div>
<div class="carousel-item">
  {{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="lightblue" >}}
</div>
<div class="carousel-item">
  {{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="lightblue" >}}
</div>

{{< /example >}}
```

With controls

Adding in the previous and next controls. We recommend using `<button>` elements, but you can also use `<a>` elements with `role="button"`.

```
{{< example >}}

<div class="carousel-item active">
  {{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="lightblue" >}}
</div>
<div class="carousel-item">
  {{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="lightblue" >}}
</div>
<div class="carousel-item">
  {{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="lightblue" >}}
</div>
```

```

    {{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="lightgray" >}}
</div>

```

Previous Next

```

{{< /example >}}

```

With indicators

You can also add the indicators to the carousel, alongside the controls, too.

```

{{< example >}}

<button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="0" class="active" data-bs-toggle="carousel" ></button>
<button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="1" aria-label="Slide 2" data-bs-toggle="carousel" ></button>
<button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="2" aria-label="Slide 3" data-bs-toggle="carousel" ></button>

<div class="carousel-item active">
    {{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="lightgray" >}}
</div>
<div class="carousel-item">
    {{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="lightgray" >}}
</div>
<div class="carousel-item">
    {{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="lightgray" >}}
</div>

```

Previous Next

```

{{< /example >}}

```

With captions

Add captions to your slides easily with the `.carousel-caption` element within any `.carousel-item`. They can be easily hidden on smaller viewports, as shown below, with optional [display utilities]({{< docsref "/utilities/display" >}}). We hide them initially with `.d-none` and bring them back on medium-sized devices with `.d-md-block`.

```

{{< example >}}

<button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="0" class="active" data-bs-toggle="carousel" ></button>
<button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="1" aria-label="Slide 2" data-bs-toggle="carousel" ></button>
<button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="2" aria-label="Slide 3" data-bs-toggle="carousel" ></button>

<div class="carousel-item active">
    {{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="lightgray" >}}
    <div class="carousel-caption d-none d-md-block">
        <h5>First slide label</h5>
        <p>Some representative placeholder content for the first slide.</p>
    </div>

```

```

</div>
<div class="carousel-item">
  {{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color
  <div class="carousel-caption d-none d-md-block">
    <h5>Second slide label</h5>
    <p>Some representative placeholder content for the second slide.</p>
  </div>
</div>
<div class="carousel-item">
  {{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color
  <div class="carousel-caption d-none d-md-block">
    <h5>Third slide label</h5>
    <p>Some representative placeholder content for the third slide.</p>
  </div>
</div>

```

Previous Next

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

Crossfade

Add `.carousel-fade` to your carousel to animate slides with a fade transition instead of a slide.

```
{{< example >}}
```

```

<div class="carousel-item active">
  {{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color
</div>
<div class="carousel-item">
  {{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color
</div>
<div class="carousel-item">
  {{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color
</div>

```

Previous Next

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

Individual `.carousel-item` interval

Add `data-bs-interval=""` to a `.carousel-item` to change the amount of time to delay between automatically cycling to the next item.

```
{{< example >}}
```

```

<div class="carousel-item active" data-bs-interval="10000">
  {{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color

```

```

</div>
<div class="carousel-item" data-bs-interval="2000">
  {{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color=
</div>
<div class="carousel-item">
  {{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color=
</div>

```

Previous Next

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

Disable touch swiping

Carousels support swiping left/right on touchscreen devices to move between slides. This can be disabled using the `data-bs-touch` attribute. The example below also does not include the `data-bs-ride` attribute and has `data-bs-interval="false"` so it doesn't autoplay.

```

{{< example >}}
<div class="carousel-item active">
  {{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color=
</div>
<div class="carousel-item">
  {{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color=
</div>
<div class="carousel-item">
  {{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color=
</div>

```

Previous Next

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

Dark variant

Add `.carousel-dark` to the `.carousel` for darker controls, indicators, and captions. Controls have been inverted from their default white fill with the `filter` CSS property. Captions and controls have additional Sass variables that customize the color and `background-color`.

```

{{< example >}}
<button type="button" data-bs-target="#carouselExampleDark" data-bs-slide-to="0" class="active">
<button type="button" data-bs-target="#carouselExampleDark" data-bs-slide-to="1" aria-label="1">
<button type="button" data-bs-target="#carouselExampleDark" data-bs-slide-to="2" aria-label="2">
<div class="carousel-item active" data-bs-interval="10000">
  {{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color=

```

```

    <div class="carousel-caption d-none d-md-block">
      <h5>First slide label</h5>
      <p>Some representative placeholder content for the first slide.</p>
    </div>
  </div>
  <div class="carousel-item" data-bs-interval="2000">
    {{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="light" >}}
    <div class="carousel-caption d-none d-md-block">
      <h5>Second slide label</h5>
      <p>Some representative placeholder content for the second slide.</p>
    </div>
  </div>
  <div class="carousel-item">
    {{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="light" >}}
    <div class="carousel-caption d-none d-md-block">
      <h5>Third slide label</h5>
      <p>Some representative placeholder content for the third slide.</p>
    </div>
  </div>
  <div>
    <div class="d-flex justify-content-between">
      <div>
        <a href="#>Previous
      </div>
      <div>
        <a href="#>Next
      </div>
    </div>
  </div>
</div>
{{< /example >}}

```

Custom transition

The transition duration of `.carousel-item` can be changed with the `$carousel-transition-duration` Sass variable before compiling or custom styles if you're using the compiled CSS. If multiple transitions are applied, make sure the transform transition is defined first (eg. `transition: transform 2s ease, opacity .5s ease-out`).

Sass

Variables

```
{{< scss-docs name="carousel-variables" file="scss/_variables.scss" >}}
```

Usage

Via data attributes

Use data attributes to easily control the position of the carousel. `data-bs-slide` accepts the keywords `prev` or `next`, which alters the slide position relative to its current position. Alternatively, use `data-bs-slide-to` to pass a raw slide index to the carousel `data-bs-slide-to="2"`, which shifts the slide position to a particular index beginning with 0.

The `data-bs-ride="carousel"` attribute is used to mark a carousel as animating starting at page load. If you don't use `data-bs-ride="carousel"` to initialize your carousel, you have to initialize it yourself. **It cannot be used in combination with (redundant and unnecessary) explicit JavaScript initialization of the same carousel.**

Via JavaScript

Call carousel manually with:

```
var myCarousel = document.querySelector('#myCarousel')
var carousel = new bootstrap.Carousel(myCarousel)
```

Options

Options can be passed via data attributes or JavaScript. For data attributes, append the option name to `data-bs-`, as in `data-bs-interval=""`.

Name

Type

Default

Description

interval

number

5000

The amount of time to delay between automatically cycling an item. If false, carousel will not automatically cycle.

keyboard

boolean

true

Whether the carousel should react to keyboard events.

pause

string | boolean

'hover'

If set to 'hover', pauses the cycling of the carousel on mouseenter and resumes the cycling of the carousel on mouseleave. If set to false, hovering over the carousel won't pause it.

`<p>On touch-enabled devices, when set to <code>'hover'</code>, cycling will pause on <code>`
`</tr>`

	<code>ride</code>	string boolean	<code>false</code>	Autoplays the carousel after the user manually cycles the first item. If set to <code>true</code> , the carousel will automatically cycle through items.
	<code>wrap</code>	boolean	<code>true</code>	Whether the carousel should cycle continuously or have hard stops.
	<code>touch</code>	boolean	<code>true</code>	Whether the carousel should support left/right swipe interactions on touchscreen devices.

Methods

```
{{< callout danger >}} {{< partial "callout-danger-async-methods.md" >}}
{{< /callout >}}
```

You can create a carousel instance with the carousel constructor, for example, to initialize with additional options and start cycling through items:

```
var myCarousel = document.querySelector('#myCarousel')
var carousel = new bootstrap.Carousel(myCarousel, {
  interval: 2000,
  wrap: false
})
```

Method

Description

cycle

Cycles through the carousel items from left to right.

pause

Stops the carousel from cycling through items.

prev

Cycles to the previous item. Returns to the caller before the previous item has been shown (e.g., before the slid.bs.carousel event occurs).

next

Cycles to the next item. Returns to the caller before the next item has been shown (e.g., before the `slid.bs.carousel` event occurs).

`nextWhenVisible`

Don't cycle carousel to next when the page isn't visible or the carousel or its parent isn't visible. Returns to the caller before the target item has been shown

```
<tr>
  <td><code>to</code></td>
  <td>Cycles the carousel to a particular frame (0 based, similar to an array). <strong>Retu
</tr>
<tr>
  <td><code>dispose</code></td>
  <td>Destroys an element's carousel. (Removes stored data on the DOM element)</td>
</tr>
<tr>
  <td>
    <code>getInstance</code>
  </td>
  <td>
    Static method which allows you to get the carousel instance associated to a DOM element.
  </td>
</tr>
<tr>
  <td>
    <code>getOrCreateInstance</code>
  </td>
  <td>
    Static method which returns a carousel instance associated to a DOM element or create a
    You can use it like this: <code>bootstrap.Carousel.getOrCreateInstance(element)</code>
  </td>
</tr>
```

Events

Bootstrap's carousel class exposes two events for hooking into carousel functionality. Both events have the following additional properties:

- **direction**: The direction in which the carousel is sliding (either `"left"` or `"right"`).
- **relatedTarget**: The DOM element that is being slid into place as the active item.
- **from**: The index of the current item
- **to**: The index of the next item

All carousel events are fired at the carousel itself (i.e. at the `<div class="carousel">`).

Event type

Description

slide.bs.carousel

Fires immediately when the slide instance method is invoked.

slid.bs.carousel

Fired when the carousel has completed its slide transition.

```
var myCarousel = document.getElementById('myCarousel')

myCarousel.addEventListener('slide.bs.carousel', function () {
  // do something...
})
```