

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 >}}
```

```
{{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="#555"
background="#777" text="First slide" >}}
{{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="#444"
background="#666" text="Second slide" >}}
{{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="#333"
background="#555" text="Third slide" >}}
{{< /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 >}}
```

```
{{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="#555"
background="#777" text="First slide" >}}
{{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="#444"
background="#666" text="Second slide" >}}
{{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="#333"
background="#555" text="Third slide" >}}
```

Previous

Next


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

With indicators

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

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

```


{{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="#555"
background="#777" text="First slide" >}}
{{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="#444"
background="#666" text="Second slide" >}}
{{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="#333"
background="#555" text="Third slide" >}}


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 >}}
```

```


{{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="#555"
background="#777" text="First slide" >}}
```

First slide label

Some representative placeholder content for the first slide.

```
{{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="#444"
background="#666" text="Second slide" >}}
```

Second slide label

Some representative placeholder content for the second slide.

```
{{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="#333"
background="#555" text="Third slide" >}}
```

Third slide label

Some representative placeholder content for the third slide.

```


Previous



Next


{{< /example >}}
```

Crossfade

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

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

```

{{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="#555"
background="#777" text="First slide" >}}
{{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="#444"
background="#666" text="Second slide" >}}
```

```
{{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="#333"
background="#555" text="Third slide" >}}
```

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 >}}
```

```
{{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="#555"
background="#777" text="First slide" >}}
```

```
{{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="#444"
background="#666" text="Second slide" >}}
```

```
{{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="#333"
background="#555" text="Third slide" >}}
```

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 >}}
```

```
{{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="#555"
background="#777" text="First slide" >}}
```

```
{{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="#444"
background="#666" text="Second slide" >}}
```

```
{{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="#333"
background="#555" text="Third slide" >}}
```

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 >}}
```

```
{{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="#aaa"
background="#f5f5f5" text="First slide" >}}
```

First slide label

Some representative placeholder content for the first slide.

```
{{< placeholder width="800" height="400" class="bd-placeholder-img-lg d-block w-100" color="#bbb"
background="#eee" text="Second slide" >}}
```

Second slide label

Some representative placeholder content for the second slide.

```
{{< placeholder width="800" height="400" class="bd-placeholder-img lg d-block w-100" color="#999"
background="#e5e5e5" text="Third slide" >}}
```

Third slide label

Some representative placeholder content for the third slide.

PreviousNext

```
{{< /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 <code>false</code> , carousel will not automatically cycle.

keyboard	boolean	true	Whether the carousel should react to keyboard events.
pause	string boolean	'hover'	<p>If set to <code>'hover'</code> , pauses the cycling of the carousel on <code>mouseenter</code> and resumes the cycling of the carousel on <code>mouseleave</code> . If set to <code>false</code> , hovering over the carousel won't pause it.</p> <p>On touch-enabled devices, when set to <code>'hover'</code> , cycling will pause on <code>touchend</code> (once the user finished interacting with the carousel) for two intervals, before automatically resuming. Note that this is in addition to the above mouse behavior.</p>
ride	string boolean	false	Autoplays the carousel after the user manually cycles the first item. If set to <code>'carousel'</code> , autoplays the carousel on load.
wrap	boolean	true	Whether the carousel should cycle continuously or have hard stops.
touch	boolean	true	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 <code>slid.bs.carousel</code> event occurs).
next	Cycles to the next item. Returns to the caller before the next item has been shown (e.g., before the <code>slid.bs.carousel</code> 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
to	Cycles the carousel to a particular frame (0 based, similar to an array). Returns to the caller before the target item has been shown (e.g., before the <code>slid.bs.carousel</code> event occurs).

<code>dispose</code>	Destroys an element's carousel. (Removes stored data on the DOM element)
<code>getInstance</code>	Static method which allows you to get the carousel instance associated to a DOM element, you can use it like this: <code>bootstrap.Carousel.getInstance(element)</code>
<code>getOrCreateInstance</code>	Static method which returns a carousel instance associated to a DOM element or create a new one in case it wasn't initialized. You can use it like this: <code>bootstrap.Carousel.getOrCreateInstance(element)</code>

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
<code>slide.bs.carousel</code>	Fires immediately when the <code>slide</code> instance method is invoked.
<code>slid.bs.carousel</code>	Fired when the carousel has completed its slide transition.

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

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