Carousel

A slideshow component for cycling through elements—images or slides of text—like a carousel.

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 prevous/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.).

The animation effect of this component is dependent on the `prefers-reduced-motion` media query. See the reduced motion section of our accessibility documentation.

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

Lastly, if you're building our JavaScript from source, it requires `util.js`.

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 prevous/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-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.

```
<div id="carouselExampleSlidesOnly" class="carousel slide" data-ride="carousel">
  <div class="carousel-inner">
    <div class="carousel-item active">
      <img src="..." class="d-block w-100" alt="...">
    </div>
    <div class="carousel-item">
      <img src="..." class="d-block w-100" alt="...">
    </div>
    <div class="carousel-item">
      <img src="..." class="d-block w-100" alt="...">
    </div>
  </div>
</div>
```

With controls

Adding in the previous and next controls:

```
<div id="carouselExampleControls" class="carousel slide" data-ride="carousel">
  <div class="carousel-inner">
    <div class="carousel-item active">
      <img src="..." class="d-block w-100" alt="...">
    </div>
    <div class="carousel-item">
      <img src="..." class="d-block w-100" alt="...">
    </div>
    <div class="carousel-item">
      <img src="..." class="d-block w-100" alt="...">
    </div>
  </div>
  <a class="carousel-control-prev" href="#carouselExampleControls" role="button" data-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="sr-only">Previous</span>
  </a>
  <a class="carousel-control-next" href="#carouselExampleControls" role="button" data-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="sr-only">Next</span>
  </a>
</div>
```

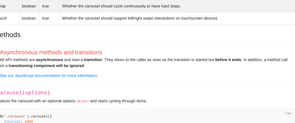
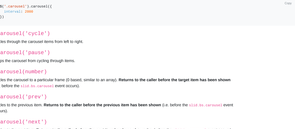
With indicators

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

```
<div id="carouselExampleIndicators" class="carousel slide" data-ride="carousel">
  <ol class="carousel-indicators">
    <li data-target="#carouselExampleIndicators" data-slide-to="0" class="active"></li>
    <li data-target="#carouselExampleIndicators" data-slide-to="1"></li>
    <li data-target="#carouselExampleIndicators" data-slide-to="2"></li>
  </ol>
  <div class="carousel-inner">
    <div class="carousel-item active">
      <img src="..." class="d-block w-100" alt="...">
    </div>
    <div class="carousel-item">
      <img src="..." class="d-block w-100" alt="...">
    </div>
    <div class="carousel-item">
      <img src="..." class="d-block w-100" alt="...">
    </div>
  </div>
  <a class="carousel-control-prev" href="#carouselExampleIndicators" role="button" data-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="sr-only">Previous</span>
  </a>
  <a class="carousel-control-next" href="#carouselExampleIndicators" role="button" data-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="sr-only">Next</span>
  </a>
</div>
```

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. We hide them initially with `.d-none` and bring them back on medium-sized devices with `.d-md-block`.

```
<div class="bd-example">
  <div id="carouselExampleCaptions" class="carousel slide" data-ride="carousel">
    <ol class="carousel-indicators">
      <li data-target="#carouselExampleCaptions" data-slide-to="0" class="active"></li>
      <li data-target="#carouselExampleCaptions" data-slide-to="1"></li>
      <li data-target="#carouselExampleCaptions" data-slide-to="2"></li>
    </ol>
    <div class="carousel-inner">
      <div class="carousel-item active">
        <img src="..." class="d-block w-100" alt="...">
        <div class="carousel-caption d-none d-md-block">
          <h5>First slide label</h5>
          <p>Nulla vitae elit libero, a pharetra augue mollis interdum.</p>
        </div>
      </div>
      <div class="carousel-item">
        
        <div class="carousel-caption d-none d-md-block">
          <h5>Second slide label</h5>
          <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
        </div>
      </div>
      <div class="carousel-item">
        
        <div class="carousel-caption d-none d-md-block">
          <h5>Third slide label</h5>
          <p>Praesent commodo cursus magna, vel scelerisque nisl consectetur.</p>
        </div>
      </div>
    </div>
    <a class="carousel-control-prev" href="#carouselExampleCaptions" role="button" data-slide="prev">
      <span class="carousel-control-prev-icon" aria-hidden="true"></span>
      <span class="sr-only">Previous</span>
    </a>
    <a class="carousel-control-next" href="#carouselExampleCaptions" role="button" data-slide="next">
      <span class="carousel-control-next-icon" aria-hidden="true"></span>
      <span class="sr-only">Next</span>
    </a>
  </div>
</div>
```

Crossfade

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

```
<div id="carouselExampleFade" class="carousel slide carousel.fade" data-ride="carousel">
  <div class="carousel-inner">
    <div class="carousel-item active">
      
    </div>
    <div class="carousel-item">
      
    </div>
    <div class="carousel-item">
      
    </div>
  </div>
  <a class="carousel-control-prev" href="#carouselExampleFade" role="button" data-slide="prev">
    
    <span class="sr-only">Previous</span>
  </a>
  <a class="carousel-control-next" href="#carouselExampleFade" role="button" data-slide="next">
    
    <span class="sr-only">Next</span>
  </a>
</div>
```

Individual `.carousel-item` interval

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


```
<div id="carouselExampleInterval" class="carousel slide" data-ride="carousel">
  <div class="carousel-inner">
    <div class="carousel-item active" data-interval="10000">
      
    </div>
    <div class="carousel-item" data-interval="2000">
      
    </div>
    <div class="carousel-item">
      
    </div>
  </div>
  <a class="carousel-control-prev" href="#carouselExampleInterval" role="button" data-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="sr-only">Previous</span>
  </a>
  <a class="carousel-control-next" href="#carouselExampleInterval" role="button" data-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="sr-only">Next</span>
  </a>
</div>
```

Usage

Via data attributes

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

The `data-ride="carousel"` attribute is used to mark a carousel as animating starting at page load. If you don't use `data-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:

```
$('.carousel').carousel()
```

Options

Options can be passed via data attributes or JavaScript. For data attributes, append the option name to `data-`, as in `data-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. On touch-enabled devices, when set to "hover", cycling will pause on `touchend` (once the user finished interacting with the carousel) for two intervals, before automatically resuming. Note that this is in addition to the above mouse behaviors.
ride	string	false	Autoplays the carousel after the user manually cycles the first item. If "carousel", 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

Asynchronous methods and transitions

All API methods are **asynchronous** and start a **transition**. They return to the caller as soon as the transition is started but **before it ends**. In addition, a method call on a **transitioning component will be ignored**.

See our JavaScript documentation for more information.

`.carousel(options)`

Initializes the carousel with an optional options `object` and starts cycling through items.

```
$('.carousel').carousel({
  interval: 2000
})
```

`.carousel('cycle')`

Cycles through the carousel items from left to right.

`.carousel('pause')`

Stops the carousel from cycling through items.

`.carousel(number)`

Cycles the carousel to a particular frame (0-based, similar to an array). **Returns to the caller before the target item has been shown** (i.e. before the `slid.bs.carousel` event occurs).

`.carousel('prev')`

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

`.carousel('next')`

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

`.carousel('dispose')`

Destroys an element's carousel.

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	This event fires immediately when the `slide` instance method is invoked.
slid.bs.carousel	This event is fired when the carousel has completed its slide transition.

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

Change transition duration

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

How it works
Example
Slides only
With controls
With indicators
With captions
Crossfade
Individual .carousel-item interval
Usage
Via data attributes
Via JavaScript
Options
Methods
.carousel(options)
.carousel('cycle')
.carousel('pause')
.carousel(number)
.carousel('prev')
.carousel('next')
.carousel('dispose')
Events
Change transition duration