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 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.).

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 previous/next controls and indicators, they're not explicitly required. Add and customize as you see fit.

Be sure to set a unique id on the `.carousel` for optional controls, especially if you're using multiple carousels on a single page.

Slides only

Here's a carousel with slides only. Note the presence of the `.d-block` and `.img-fluid` on carousel images to prevent browser default image alignment.
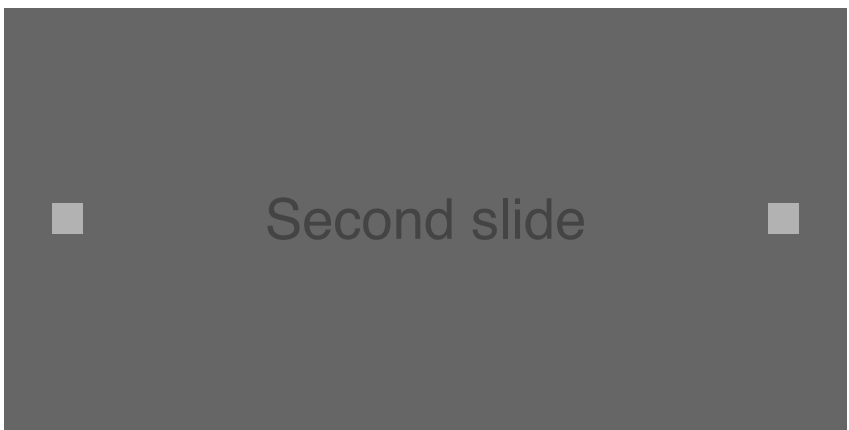
```
<div id="carouselExampleSlidesOnly" class="carousel slide" data-
ride="carousel">
  <div class="carousel-inner">
    <div class="carousel-item active">
      <img class="d-block w-100" src="..." alt="First slide">
    </div>
    <div class="carousel-item">
      <img class="d-block w-100" src="..." alt="Second slide">
    </div>
    <div class="carousel-item">
      <img class="d-block w-100" src="..." alt="Third slide">
    </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 class="d-block w-100" src="..." alt="First slide">
    </div>
    <div class="carousel-item">
      <img class="d-block w-100" src="..." alt="Second slide">
    </div>
    <div class="carousel-item">
      <img class="d-block w-100" src="..." alt="Third slide">
    </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>
```
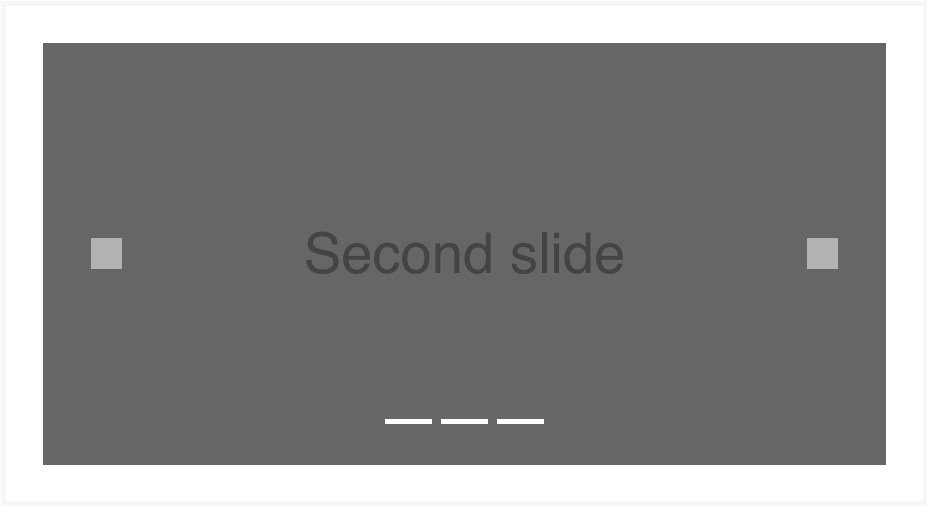
With indicators

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

Second slide

— — —

Copy

```html
<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 class="d-block w-100" src="..." alt="First slide">
    </div>
    <div class="carousel-item">
      <img class="d-block w-100" src="..." alt="Second slide">
    </div>
    <div class="carousel-item">
      <img class="d-block w-100" src="..." alt="Third slide">
    </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>
```
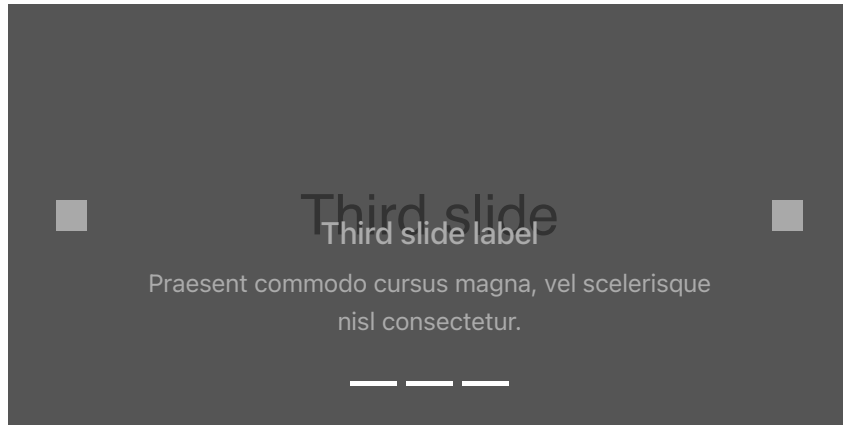
Initial active element required

The `.active` class needs to be added to one of the slides. Otherwise, the carousel will not be visible.

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`.

Third slide

Third slide label

Praesent commodo cursus magna, vel scelerisque
nisl consectetur.

```
<div class="carousel-item">
  <img src="..." alt="...">
  <div class="carousel-caption d-none d-md-block">
    <h5>...</h5>
    <p>...</p>
  </div>
</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. **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 behavior.
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.

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
})
```

Copy

`.carousel('cycle')`