

Offcanvas

Contents

How it works	1
Examples	2
Offcanvas components	2
Live demo	2
Placement	3
Backdrop	3
Accessibility	4
Sass	4
Variables	4
Usage	4
Via data attributes	4
Via JavaScript	5
Options	5
Methods	5
Events	6

How it works

Offcanvas is a sidebar component that can be toggled via JavaScript to appear from the left, right, or bottom edge of the viewport. Buttons or anchors are used as triggers that are attached to specific elements you toggle, and **data** attributes are used to invoke our JavaScript.

- Offcanvas shares some of the same JavaScript code as modals. Conceptually, they are quite similar, but they are separate plugins.
- Similarly, some source Sass variables for offcanvas's styles and dimensions are inherited from the modal's variables.
- When shown, offcanvas includes a default backdrop that can be clicked to hide the offcanvas.
- Similar to modals, only one offcanvas can be shown at a time.

Heads up! Given how CSS handles animations, you cannot use **margin** or **translate** on an **.offcanvas** element. Instead, use the class as an independent wrapping element.

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

Examples

Offcanvas components

Below is an offcanvas example that is shown by default (via `.show` on `.offcanvas`). Offcanvas includes support for a header with a close button and an optional body class for some initial `padding`. We suggest that you include offcanvas headers with dismiss actions whenever possible, or provide an explicit dismiss action.

```
{{< example class="bd-example-offcanvas p-0 bg-light overflow-hidden" >}}
<h5 class="offcanvas-title" id="offcanvasLabel">Offcanvas</h5>
<button type="button" class="btn-close text-reset" data-bs-dismiss="offcanvas" aria-label="Close"></button>

Content for the offcanvas goes here. You can place just about any Bootstrap component or custom HTML here.

{{< /example >}}
```

Live demo

Use the buttons below to show and hide an offcanvas element via JavaScript that toggles the `.show` class on an element with the `.offcanvas` class.

- `.offcanvas` hides content (default)
- `.offcanvas.show` shows content

You can use a link with the `href` attribute, or a button with the `data-bs-target` attribute. In both cases, the `data-bs-toggle="offcanvas"` is required.

```
{{< example >}} Link with href Button with data-bs-target
<h5 class="offcanvas-title" id="offcanvasExampleLabel">Offcanvas</h5>
<button type="button" class="btn-close text-reset" data-bs-dismiss="offcanvas" aria-label="Close"></button>

<div class="">
  Some text as placeholder. In real life you can have the elements you have chosen. Like, text and just about anything.
</div>

<div class="dropdown mt-3">
  <button class="btn btn-secondary dropdown-toggle" type="button" id="dropdownMenuButton" data-bs-toggle="dropdown">
    Dropdown button
  </button>
  <ul class="dropdown-menu" aria-labelledby="dropdownMenuButton">
    <li><a class="dropdown-item" href="#">Action</a></li>
    <li><a class="dropdown-item" href="#">Another action</a></li>
    <li><a class="dropdown-item" href="#">Something else here</a></li>
  </ul>
</div>
```

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

Placement

There's no default placement for offcanvas components, so you must add one of the modifier classes below;

- `.offcanvas-start` places offcanvas on the left of the viewport (shown above)
- `.offcanvas-end` places offcanvas on the right of the viewport
- `.offcanvas-top` places offcanvas on the top of the viewport
- `.offcanvas-bottom` places offcanvas on the bottom of the viewport

Try the top, right, and bottom examples out below.

```
{{< example >}} Toggle top offcanvas
```

```
<h5 id="offcanvasTopLabel">Offcanvas top</h5>
<button type="button" class="btn-close text-reset" data-bs-dismiss="offcanvas" aria-label="Close" ></button>
...

```

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

```
{{< example >}} Toggle right offcanvas
```

```
<h5 id="offcanvasRightLabel">Offcanvas right</h5>
<button type="button" class="btn-close text-reset" data-bs-dismiss="offcanvas" aria-label="Close" ></button>
...

```

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

```
{{< example >}} Toggle bottom offcanvas
```

```
<h5 class="offcanvas-title" id="offcanvasBottomLabel">Offcanvas bottom</h5>
<button type="button" class="btn-close text-reset" data-bs-dismiss="offcanvas" aria-label="Close" ></button>
...

```

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

Backdrop

Scrolling the `<body>` element is disabled when an offcanvas and its backdrop are visible. Use the `data-bs-scroll` attribute to toggle `<body>` scrolling and `data-bs-backdrop` to toggle the backdrop.

```
{{< example >}} Enable body scrolling Enable backdrop (default) Enable both scrolling & backdrop
```

```
<h5 class="offcanvas-title" id="offcanvasScrollingLabel">Colored with scrolling</h5>
<button type="button" class="btn-close text-reset" data-bs-dismiss="offcanvas" aria-label="Close" ></button>

```

```

<p>Try scrolling the rest of the page to see this option in action.</p>

<h5 class="offcanvas-title" id="offcanvasWithBackdropLabel">Offcanvas with backdrop</h5>
<button type="button" class="btn-close text-reset" data-bs-dismiss="offcanvas" aria-label="Close"></button>

<p>.....</p>

<h5 class="offcanvas-title" id="offcanvasWithBothOptionsLabel">Backdrop with scrolling</h5>
<button type="button" class="btn-close text-reset" data-bs-dismiss="offcanvas" aria-label="Close"></button>

<p>Try scrolling the rest of the page to see this option in action.</p>
{{< /example >}}

```

Accessibility

Since the offcanvas panel is conceptually a modal dialog, be sure to add `aria-labelledby="..."`—referencing the offcanvas title—to `.offcanvas`. Note that you don’t need to add `role="dialog"` since we already add it via JavaScript.

Sass

Variables

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

Usage

The offcanvas plugin utilizes a few classes and attributes to handle the heavy lifting:

- `.offcanvas` hides the content
- `.offcanvas.show` shows the content
- `.offcanvas-start` hides the offcanvas on the left
- `.offcanvas-end` hides the offcanvas on the right
- `.offcanvas-bottom` hides the offcanvas on the bottom

Add a dismiss button with the `data-bs-dismiss="offcanvas"` attribute, which triggers the JavaScript functionality. Be sure to use the `<button>` element with it for proper behavior across all devices.

Via data attributes

Toggle Add `data-bs-toggle="offcanvas"` and a `data-bs-target` or `href` to the element to automatically assign control of one offcanvas element. The `data-bs-target` attribute accepts a CSS selector to apply the offcanvas to. Be sure to add the class `offcanvas` to the offcanvas element. If you’d like it to default open, add the additional class `show`.

Dismiss `{{% js-dismiss “offcanvas” %}}`

`{{< callout warning >}}` While both ways to dismiss an offcanvas are supported, keep in mind that dismissing from outside an offcanvas does not match the WAI-ARIA modal dialog design pattern. Do this at your own risk. `{{< /callout >}}`

Via JavaScript

Enable manually with:

```
var offcanvasElementList = [].slice.call(document.querySelectorAll('.offcanvas'))
var offcanvasList = offcanvasElementList.map(function (offcanvasEl) {
  return new bootstrap.Offcanvas(offcanvasEl)
})
```

Options

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

<code>{{< bs-table “table” >}}</code>	Name	Type	Default	Description
<code> — — — — </code>	<code>backdrop</code>	boolean	<code>true</code>	Apply a backdrop on body while offcanvas is open
<code> — — — — </code>	<code>keyboard</code>	boolean	<code>true</code>	Closes the offcanvas when escape key is pressed
<code> — — — — </code>	<code>scroll</code>	boolean	<code>false</code>	Allow body scrolling while offcanvas is open

Methods

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

Activates your content as an offcanvas element. Accepts an optional options object.

You can create an offcanvas instance with the constructor, for example:

```
var myOffcanvas = document.getElementById('myOffcanvas')
var bsOffcanvas = new bootstrap.Offcanvas(myOffcanvas)
```

<code>{{< bs-table “table” >}}</code>	Method	Description
<code> — — — — </code>	<code>toggle</code>	Toggles an offcanvas element to shown or hidden. Returns to the caller before the offcanvas element has actually been shown or hidden (i.e. before the <code>shown.bs.offcanvas</code> or <code>hidden.bs.offcanvas</code> event occurs).
<code> — — — — </code>	<code>show</code>	Shows an offcanvas element. Returns to the caller before the offcanvas element has actually been shown (i.e. before the <code>shown.bs.offcanvas</code> event occurs).
<code> — — — — </code>	<code>hide</code>	Hides an offcanvas element. Returns to the caller before the offcanvas element has actually been hidden (i.e. before the <code>hidden.bs.offcanvas</code> event occurs).
<code> — — — — </code>	<code>getInstance</code>	<i>Static</i> method which allows you to get the offcanvas instance associated with a DOM element

`getOrCreateInstance` | *Static* method which allows you to get the `offcanvas` instance associated with a DOM element, or create a new one in case it wasn't initialized | {{< /bs-table >}}

Events

Bootstrap's `offcanvas` class exposes a few events for hooking into `offcanvas` functionality.

{{< bs-table "table" >}}	Event type	Description
<code>show.bs.offcanvas</code>		This event fires immediately when the <code>show</code> instance method is called.
<code>shown.bs.offcanvas</code>		This event is fired when an <code>offcanvas</code> element has been made visible to the user (will wait for CSS transitions to complete).
<code>hide.bs.offcanvas</code>		This event is fired immediately when the <code>hide</code> method has been called.
<code>hidden.bs.offcanvas</code>		This event is fired when an <code>offcanvas</code> element has been hidden from the user (will wait for CSS transitions to complete).

```
var myOffcanvas = document.getElementById('myOffcanvas')
myOffcanvas.addEventListener('hidden.bs.offcanvas', function () {
  // do something...
})
```