WEB AWESOME *(/)* 3.0.0-beta.4 **BETA**   ✏ Default ⌄   ⚙ System ⌄ | ◯ ☆ | Login

Dialog

`<wa-dialog>`   Since 2.0   stable

Dialogs, sometimes called "modals", appear above the page and require the user's immediate attention.

Examples # (#examples)

Dialog without Header

Headers are enabled by default. To render a dialog without a header, add the `without-header` attribute.

Dialog with Footer

Footers can be used to display titles and more. Use the `footer` slot to add a footer to the dialog.

Opening and Closing Dialogs Declaratively

You can open and close dialogs with JavaScript by toggling the `open` attribute, but you can also do it declaratively. Add the `data-dialog="open id"` to any button on the page, where `id` is the ID of the dialog you want to open.

Similarly, you can add `data-dialog="close"` to a button *inside* of a dialog to tell it to close.

Custom Width

Just use the `--width` custom property to set the dialog's width.

Scrolling

By design, a dialog's height will never exceed that of the viewport. As such, dialogs will not scroll with the page ensuring the header and footer are always accessible to the user.

Header Actions

The header shows a functional close button by default. You can use the `header-actions` slot to add additional [buttons (/docs/components/button)](/docs/components/button) if needed.

Light Dismissal

If you want the dialog to close when the user clicks on the overlay, add the `light-dismiss` attribute.

Preventing the Dialog from Closing

By default, dialogs will close when the user clicks the close button, clicks the overlay, or presses the `Escape` key. In most cases, the default behavior is the best behavior in terms of UX. However, there are situations where this may be undesirable, such as when data loss will occur.

To keep the dialog open in such cases, you can cancel the `wa-hide` event. When canceled, the dialog will remain open and pulse briefly to draw the user's attention to it.

You can use `event.detail.source` to determine which element triggered the request to close. This example prevents the dialog from closing when the overlay is clicked, but allows the close button or `Escape` to dismiss it.

Setting Initial Focus

To give focus to a specific element when the dialog opens, use the `autofocus` attribute.

Slots

Learn more about [using slots (/docs/usage/#slots).](/docs/usage/#slots)

Name	Description
(default)	The dialog's main content.
label	The dialog's label. Alternatively, you can use the `label` attribute.
header-actions	Optional actions to add to the header. Works best with `<wa-button>`.
footer	The dialog's footer, usually one or more buttons representing various options.

Attributes & Properties

Learn more about [attributes and properties (/docs/usage/#attributes-and-properties)](/docs/usage/#attributes-and-properties).

Name	Description	Reflects
open open	Indicates whether or not the dialog is open. Toggle this attribute to show and hide the dialog. **Type** `boolean` **Default** `false`	✓
label label	The dialog's label as displayed in the header. You should always include a relevant label, as it is required for proper accessibility. If you need to display HTML, use the `label` slot instead. **Type** `string` **Default** `''`	✓
withoutHeader without-header	Disables the header. This will also remove the default close button. **Type** `boolean` **Default** `false`	✓
lightDismiss light-dismiss	When enabled, the dialog will be closed when the user clicks outside of it. **Type** `boolean` **Default** `false`	

Events

Learn more about [events (/docs/usage/#events)](/docs/usage/#events).

Name	Description
wa-show	Emitted when the dialog opens.
wa-after-show	Emitted after the dialog opens and all animations are complete.
wa-hide	Emitted when the dialog is requested to close. Calling `event.preventDefault()` will prevent the dialog from closing. You can inspect `event.detail.source` to see which element caused the dialog to close. If the source is the dialog element itself, the user has pressed `Escape` or the dialog has been closed programmatically. Avoid using this unless closing the dialog will result in destructive behavior such as data loss.
wa-after-hide	Emitted after the dialog closes and all animations are complete.

CSS custom properties

Learn more about [CSS custom properties (/docs/customizing/#custom-properties)](/docs/customizing/#custom-properties).

Name	Description
--spacing	The amount of space around and between the dialog's content.
--width	The preferred width of the dialog. Note that the dialog will shrink to accommodate smaller screens.
--show-duration	The animation duration when showing the dialog. **Default** `200ms`

Name	Description
`--hide-duration`	The animation duration when hiding the dialog. **Default** `200ms`

CSS parts

Learn more about [CSS parts (/docs/customizing/#css-parts)](/docs/customizing/#css-parts).

Name	Description
`header`	The dialog's header. This element wraps the title and header actions.
`header-actions`	Optional actions to add to the header. Works best with `<wa-button>`.
`title`	The dialog's title.
`close-button`	The close button, a `<wa-button>`.
`close-button__base`	The close button's exported `base` part.
`body`	The dialog's body.
`footer`	The dialog's footer.

Dependencies

This component automatically imports the following elements. Sub-dependencies, if any exist, will also be included in this list.

- [`<wa-button>` (/docs/components/button)](/docs/components/button)
- [`<wa-icon>` (/docs/components/icon)](/docs/components/icon)
- [`<wa-spinner>` (/docs/components/spinner)](/docs/components/spinner)

Importing

The [autoloader (/docs/#quick-start-autoloading-via-cdn)](/docs/#quick-start-autoloading-via-cdn) is the recommended way to import components. If you prefer to do it manually, use one of the following code snippets.

CDN   npm   React

Need a hand?   Report a bug   Ask for help