

 GitHub

Home

Usage

Customizing

Other tiny libs

Extensions

 Components

All components

Typography

Accordion

Alert

Badge

Button

Card

Dialog

Dropdown WC

Form elements

Meter

Progress

Spinner

Skeleton

Sidebar

Switch

Table

Tabs WC

Tooltip

Toast

Grid

.....

 Switch theme

Components

Oat is an ultra-lightweight HTML + CSS + minimal JS, semantic UI component library with zero dependencies. No framework or build or dev dependencies of any kind. Just include the tiny CSS and JS bundles.

Semantic tags and attributes are styled contextually out of the box without classes, thereby forcing best practices. A few dynamic components are WebComponents.

Typography

Base text elements are styled automatically. No classes needed.

 Preview Code

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

This is a paragraph with **bold text**, *italic text*, and [a link](#).

Here's some inline code and a code block:

This is a blockquote. It's styled automatically.

Ordered list item 1

- Unordered list item 2
- Unordered list item 3

1. Ordered list item 1
2. Ordered list item 2
3. Ordered list item 3

[Link](#)

Accordion

Use native `<details>` and `<summary>` for collapsible content.

Preview Code

What is Oat?

How do I use it?

Is it accessible?

[Link](#)

Alert

Use `role="alert"` for alert styling. Set `data-variant` for success, warning, or error.

Default Alert This is a default alert message.

Success! Your changes have been saved.

Warning! Please review before continuing.

Error! Something went wrong.

[Link](#)

Badge

Use `.badge` class with variant modifiers.

Preview

Code

Default

Secondary

Outline

Success

Warning

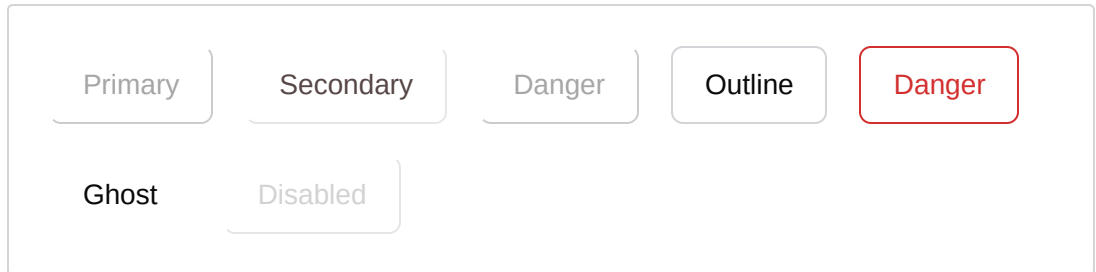
Danger

[Link](#)

Button

classes for visual styles.

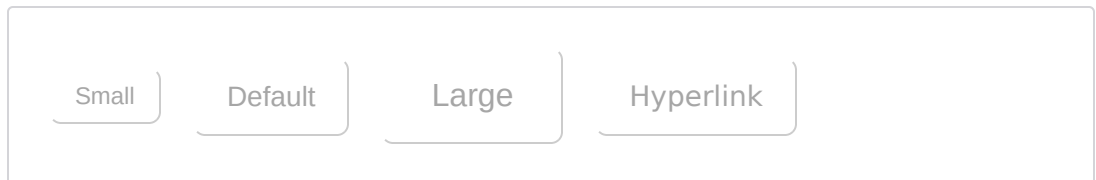
Preview `}` Code



Sizes

Use `.small` or `.large` for size variants.

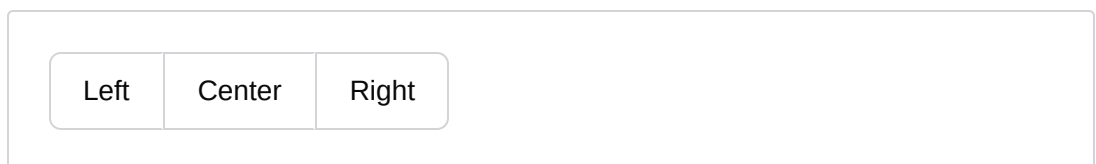
Preview `}` Code



Button group

Wrap buttons in `<menu class="buttons">` for connected buttons.

Preview `}` Code



[Link](#)

Use `class="card"` for a visual box-like card look.

Preview

Code

Card Title

Card description goes here.

This is the card content. It can contain any HTML.

Cancel

Save

[Link](#)

Dialog

Fully semantic, zero-Javascript, dynamic dialog with `<dialog>`. Use `commandfor` and `command="show-modal"` attributes on an element to open a target dialog. Focus trapping, z placement, keyboard shortcuts all work out of the box.

Preview

Code

Open dialog

With form fields

Forms inside dialogs work naturally. Use `command="close"` on cancel buttons to close.

Open form dialog

Handling return value

Listen to the native `close` event to get the button value:

```
const dialog = document.querySelector("#demo-dialog");
dialog.addEventListener('close', (e) => {
  console.log(dialog.returnValue); // "confirm"
});
```

or use `onclose` inline:

```
<dialog id="my-dialog" onclose="console.log(this.returnValue)">
```

[Link](#)

Dropdown

Wrap in `<ot-dropdown>`. Use `popovertarget` on the trigger and `popover` on the target. If a dropdown `<menu>`, items use `role="menuitem"`.

Preview Code

Options ▾

Popover

Preview Code

Confirm

Link

Form elements

Form elements are styled automatically. Wrap inputs in `<label>` for proper association.

Preview Code

Name

Enter your name

Email

you@example.com

Password

Password

This is a small hint

Select

Select an option

Message

Your message...

File

Choose File No file chosen

Date and time

mm/dd/yyyy, --:-- --

Date

mm/dd/yyyy

I agree to the terms

Preference

Option A Option B Option C

Volume

Submit

Input group

Use `.group` on a `<fieldset>` to combine inputs with buttons or labels.

Preview Code

https:// subdomain .example.com Go

Search Go

Validation error

Preview Code

Email

invalid-email

Please enter a valid email address.

[Link](#)

Meter

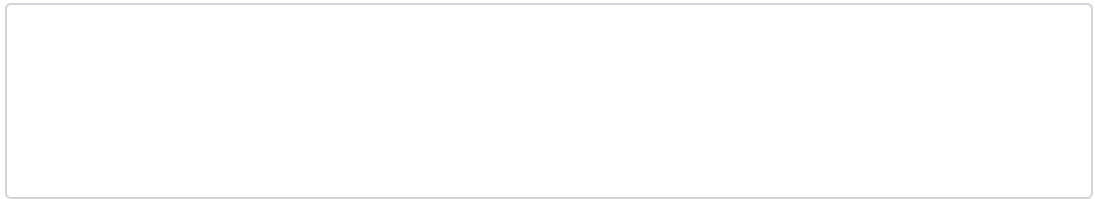
Use `<meter>` for values within a known range. Browser shows colors based on low/high/optimum attributes.

Preview Code

[Link](#)

Progress

Use the native `<progress>` element.

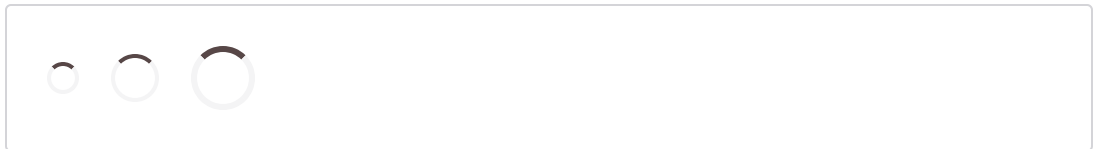


[Link](#)

Spinner

Use `.spinner` with `role="status"` for loading indicators. Size with `.small` or `.large`.

Preview Code

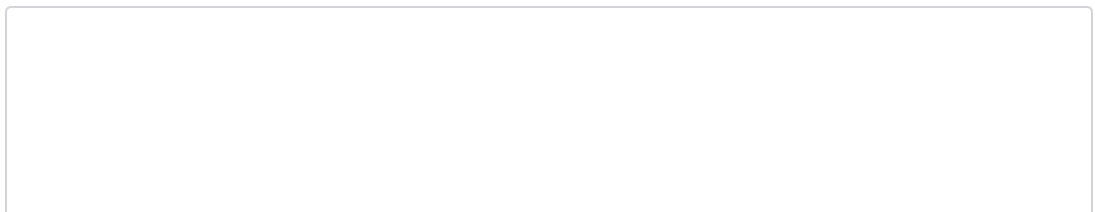


[Link](#)

Skeleton

Use `.skeleton` with `role="status"` for loading placeholders. Add `.line` for text or `.box` for images.

Preview Code

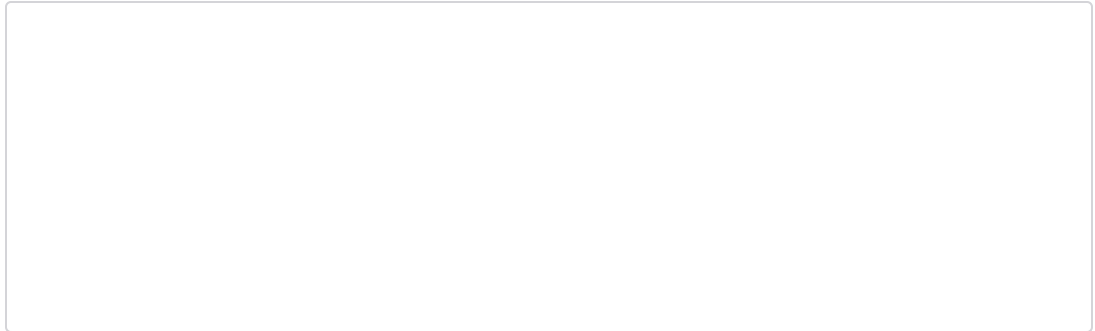


Skeleton card

Put skeleton loader inside `<article>` to get a card layout.

Preview

Code



[Link](#)

Sidebar

Use `data-sidebar-layout` on a container (typically `<body>`) with `<aside data-sidebar>` for the sidebar and `<main>` for content. The sidebar stays sticky while the main content scrolls. On mobile, the sidebar becomes a slide-out overlay toggled by a `[data-sidebar-toggle]` button. To make the sidebar collapsible at all widths, set `data-sidebar-layout="always"`.

Preview

Code

Home

Users

Settings

General

Security

Main content area. Scrolls with the page body.

Always-collapsible

Set `data-sidebar-layout="always"` to keep the toggle visible on all screen sizes.

```
<body data-sidebar-layout="always">
  ...
</body>
```

With top sticky nav

Add `data-topnav` to a header element for a full-width top navigation bar. The sidebar will adjust to sit below it.

```
<body data-sidebar-layout>
  <nav data-topnav>
    <button data-sidebar-toggle aria-label="Toggle menu" class=
    <span>App Name</span>
  </nav>

  <aside data-sidebar>
    <header>Logo</header>
    <nav>...navigation...</nav>
    <footer>Actions</footer>
  </aside>

  <main>
    Main page content.
  </main>
</body>
```

Structure

| Attribute | Element | |
|----------------------------------|-----------|---|
| <code>data-sidebar-layout</code> | Container | Grid layout wrapper (sidebar + main), typically <code><body></code> |

| | | |
|---|-----------------------------|---|
| <code>data-sidebar-layout="always"</code> | Container | Always-collapsible sidebar (toggle visible on screen sizes) |
| <code>data-topnav</code> | <code><header></code> | Full-width top nav (optional, spans full width) |
| <code>data-sidebar</code> | <code><aside></code> | Sticky sidebar element |
| <code>data-sidebar-toggle</code> | <code><button></code> | Toggles sidebar (mobile) and collapse (always mode) |
| <code>data-sidebar-open</code> | Layout | Applied to layout when sidebar is open |

[Link](#)

Switch

Add `role="switch"` to a checkbox for toggle switch styling.

Preview Code

Notifications Confabulation

Disabled

Preview Code

Disabled off Disabled on

Table

Tables are styled by default. Use `<thead>` and `<tbody>` tags.

Preview Code

| Name | Email | Role | Status |
|---------------|-------------------|--------|---------|
| Alice Johnson | alice@example.com | Admin | Active |
| Bob Smith | bob@example.com | Editor | Active |
| Carol White | carol@example.com | Viewer | Pending |

[Link](#)

Tabs

Wrap tab buttons and panels in `<ot-tabs>`. Use `role="tablist"`, `role="tab"`, and `role="tabpanel"`.

Preview Code

Account Password Notifications

Account Settings

[Link](#)

Tooltip

Use the standard `title` attribute on any element to render a tooltip with smooth transition.

Preview Code

Save

Delete

Profile

[Link](#)

Toast

Show toast notifications with `ot.toast(message, title?, options?)`.

Preview Code

Success

Danger

Warning

Info

Placement

```

ot.toast('Top center', '', { placement: 'top-center' })
ot.toast('Top right', '', { placement: 'top-right' }) // default
ot.toast('Bottom left', '', { placement: 'bottom-left' })
ot.toast('Bottom center', '', { placement: 'bottom-center' })
ot.toast('Bottom right', '', { placement: 'bottom-right' })

```

Options

| Option | Default | Description |
|-----------|-------------|-------------------------------------|
| variant | '' | 'success', 'danger', 'warning' |
| placement | 'top-right' | Position on screen |
| duration | 4000 | Auto-dismiss in ms (0 = persistent) |

Custom markup

Use `ot.toastEl(element, options?)` to show toasts with custom HTML content.

Preview Code

Toast with action

From a template:

```

ot.toastEl(document.querySelector('#my-template'))
ot.toastEl(document.querySelector('#my-template'), { duration:

```

Dynamic element:

```

const el = document.createElement('output');
el.className = 'toast';
el.setAttribute('data-variant', 'warning');

```

The element is cloned before display, so templates can be reused.

Clearing toasts

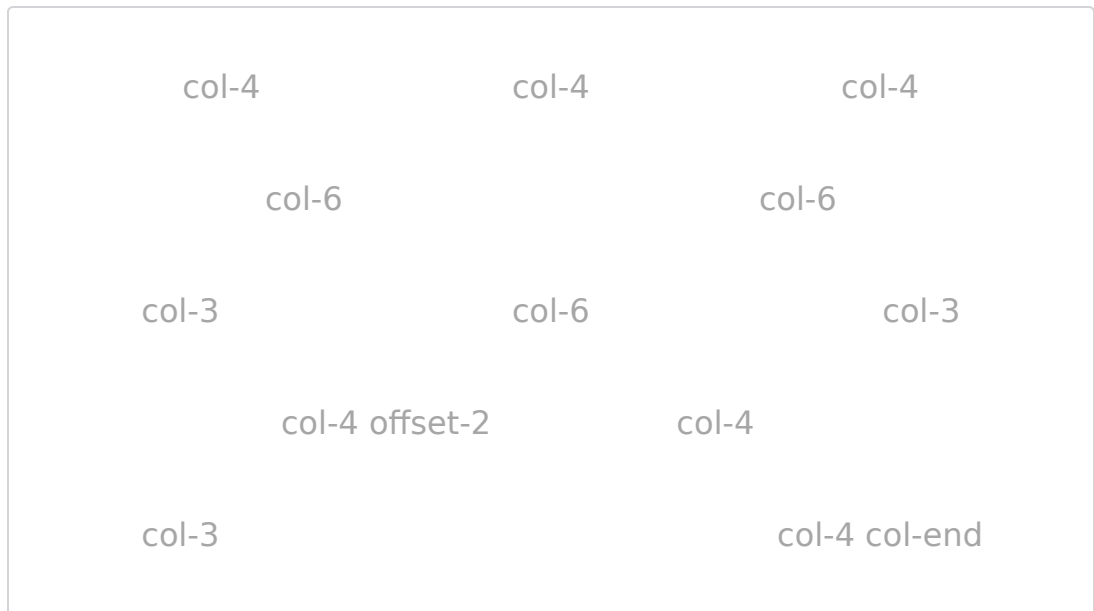
```
ot.toast.clear()           // Clear all
ot.toast.clear('top-right') // Clear specific placement
```

[Link](#)

Grid

A 12-column grid system using CSS grid. Use `.container`, `.row`, and `.col` classes. Column widths use `.col-{n}` where n is 1-12.

Preview Code



[Link](#)

See [utilities.css](#) for commonly used utility and helper classes.

[Link](#)

© [Kailash Nadh](#)