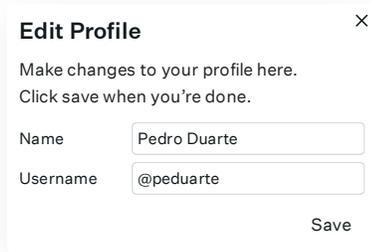


Unstyled, accessible components for building high-quality design systems and web apps in React.

### Install Primitives



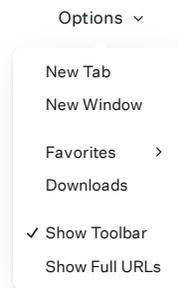
**Edit Profile** ✕

Make changes to your profile here.  
Click save when you're done.

Name

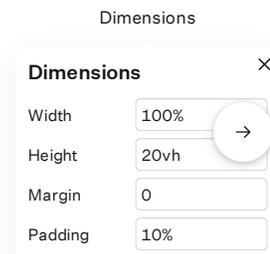
Username

Save



Options ▾

- New Tab
- New Window
- Favorites >
- Downloads
- ✓ Show Toolbar
- Show Full URLs



Dimensions ✕

Width	<input type="text" value="100%"/>
Height	<input type="text" value="20vh"/>
Margin	<input type="text" value="0"/>
Padding	<input type="text" value="10%"/>

### Dialog

With modal and non-modal modes, fine-grained focus control, accessible to screen readers.

### Dropdown Menu

With submenus, checkable items, collision handling, arrow key navigation, and typeahead support.

### Popover

With fine-grained focus control, collision handling, origin-aware and collision-aware animations.

### Case studies

## World-class teams use Radix to power their products



“We’ve been able to focus on building solid user experiences on top of Radix Primitives. With UI components, there are just too many angles and rabbit holes to cover for product teams that wish to move quickly.”



Rauno Freiberg, UI Engineer at Vercel



“Radix has significantly improved the accessibility standard in our components. We spend far less time discussing and implementing keyboard navigation, focus traps, and researching appropriate ARIA techniques for our components.”



And dozens more companies →



Basedash

basement.



Linear



liveblocks



supabase



UNIVERSITY OF AMSTERDAM

### Why Radix Primitives

## Spend less time on undifferentiated work

Saved

5M+

Monthly downloads

3000+

Discord members

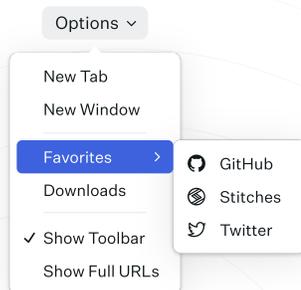
5k+

GitHub stars

### Case in point

## So, you think you can build a dropdown?

We agonise over API design, performance, and accessibility so you don't need to.



✓ Full keyboard navigation

✓ Modal and non-modal modes

Navigate the menu using arrow keys, Escape, and Enter hotkeys, or even via typeahead.

Configure whether the dropdown menu allows or blocks outside interactions.

✓ **Supports submenus**

Nest another level of fully functional submenus inside the dropdown menu.

✓ **Supports assistive technology**

Implements correct semantics and behaviors, so it's accessible to people using assistive technology.

✓ **Collision and origin-aware animations**

Create open and close animations that take the dropdown menu's actual position into account.

✓ **Control alignment and collisions**

Position the menu anywhere relative to the trigger, taking collisions with the screen into account.

✓ **Fully managed focus**

Granularly control focus behavior when user closes the dropdown menu or focuses an outside element.

✓ **Supports checkable items**

Items can be used to perform actions, as well as act as checkbox or radiobutton controls.

**Supports assistive technology**

## Accessibility out of the box

Screen reader support

Typeahead support

Keyboard navigation

RTL support

Screen reader ⓘ



Radix component </>

Navigation ▾

Show Minimap

Go to Symbol

Go to Definition

Go to References



### WAI-ARIA compliant

Radix Primitives follow the WAI-ARIA guidelines, implementing correct semantics and behaviors for our components.



### Keyboard navigation

Primitives provide full keyboard support for components where users expect to use a keyboard or other input devices.



### Focus management

Out of the box, Primitives provide sensible focus management defaults, which can be further customized in your code.



### Screen reader tested

We test Primitives with common assistive technologies, looking out for practical issues that people may experience.

## Developer experience to love

# Develop with an open, thought-out API

One of our main goals is to provide the best possible developer experience. Radix Primitives provides a fully-typed API. All components share a similar API, creating a consistent and predictable experience.

```
// Add styles with your preferred CSS technology
const TooltipContent = styled(Tooltip.Content, {
  backgroundColor: 'black',
  borderRadius: '3px',
  padding: '5px'
});

const PopoverContent = styled(Popover.Content, {
  backgroundColor: 'white',
  boxShadow: '0 2px 10px -3px rgb(0 0 0 / 20%)',
  borderRadius: '3px',
});

const DialogContent = styled(Dialog.Content, {
  backgroundColor: 'white',
  boxShadow: '0 3px 15px -4px rgb(0 0 0 / 30%)',
  borderRadius: '5px',
});
```

```
// Compose a custom Tooltip component
export const StatusTooltip = ({ state, label
return (
  <Tooltip.Root>
    <Tooltip.Trigger asChild>
      <Text>
        <Status variant={state} />
      </Text>
    </Tooltip.Trigger>
    <Tooltip.Portal>
      <TooltipContent>
        <Tooltip.Arrow />
        {label}
      </TooltipContent>
    </Tooltip.Portal>
  </Tooltip.Root>
);
```

### Unstyled

No need to override styles, no specificity wars.

### Composable

Granular access to each component part.

## Transition to Radix Primitives

# Adoption made easy

[Go to docs](#)

### Incremental adoption

Each component is its own independently versioned package, so new components can be added alongside your existing code. No need to disrupt feature work with a huge rewrite—you can start small and add more components one by one.

### Detailed docs and TypeScript support

Radix documentation contains real-world examples, extensive API references, accessibility details, and full TypeScript support. All components share a similar API, creating a consistent developer experience. You will love working with Radix Primitives.

An active and friendly community 🙌

## Join our fast-growing community



### Twitter

For announcements, blog posts, and general Radix tips.



### Discord

To get involved in the community, ask questions, and share tips.



### GitHub

To file issues, request features, and contribute, check out our GitHub.

More from the Radix team

## Complete suite of design system tools



### Stitches

CSS-in-JS with near-zero runtime, SSR support, and a best-in-class DX.



### Colors

Beautiful, thought-out palettes with auto dark mode.



### Icons

A crisp set of 15×15 icons, balanced and consistent.



A project by WorkOS.

#### Products

[Primitives](#)

[Colors](#)

#### Docs

[Introduction](#)

[Styling](#)

#### Community

[GitHub](#) ↗

[Twitter](#) ↗

[Icons ↗](#)

[Accessibility](#)

[Discord ↗](#)

[Stitches ↗](#)

[Releases](#)