DropdownMenu

- Anatomy
- Design guidelines
  - Usage
    - When to use a DropdownMenu
    - Behavior
    - Placement
- Development guidelines
  - Usage
  - Props
    - icon: string | null
    - label: string
    - controls: DropdownOption[] | DropdownOption[][]
    - children: (callbackProps: DropdownCallbackProps) => ReactNode
    - className: string
    - popoverProps: DropdownProps['popoverProps']
    - toggleProps: ToggleProps
    - menuProps: NavigableContainerProps
    - disableOpenOnArrowDown: boolean
  - defaultOpen: boolean
  - open: boolean
  - onToggle: (willOpen: boolean) => void

The DropdownMenu displays a list of actions (each contained in a MenuItem, MenuItemsChoice, or MenuGroup) in a compact way. It appears in a Popover after the user has interacted with an element (a button or icon) or when they perform a specific action.

When building for the Gutenberg repo, use `Menu` from `@wordpress/ui` instead. Otherwise, continue using `DropdownMenu` for now.

An expanded DropdownMenu, containing a list of MenuItems.

Anatomy

Anatomy of a DropdownMenu.

1. Popover: a container component in which the DropdownMenu is wrapped.
2. Parent button: the icon or button that is used to toggle the display of the Popover containing the DropdownMenu.
3. MenuItem: the list items within the DropdownMenu.

Design guidelines

Usage

When to use a DropdownMenu

Use a DropdownMenu when you want users to:

- Choose an action or change a setting from a list, AND
- Only see the available choices contextually.

If you need to display all the available options at all times, consider using a Toolbar instead.

Use a DropdownMenu to display a list of actions after the user interacts with an icon.

Do
Use a DropdownMenu to display a list of actions after the user interacts with an icon.

Don’t use a DropdownMenu for important actions that should always be visible. Use a Toolbar instea

Don’t
Don’t use a DropdownMenu for frequently used actions. Use a Toolbar instead.

Behavior

Generally, the parent button should have a triangular icon to the right of the icon or text to indicate that interacting with it will show a DropdownMenu. In rare cases where the parent button directly indicates that there’ll be more content (through the use of an ellipsis or “More” label), this can be omitted.

The parent button should retain the same visual styling regardless of whether the DropdownMenu is displayed or not.

Opening the menu with a pointer focuses the menu container without highlighting an item. Arrow keys then move focus into the menu. Opening it with Enter, Space, or ArrowDown focuses the first item. Set `popoverProps.focusOnMount` to override this default focus behavior.

Placement

The DropdownMenu should typically appear directly below, or below and to the left of, the parent button. If there isn't enough space below to display the full DropdownMenu, it can be displayed instead above the parent button.

Development guidelines

DropdownMenu is a React component to render an expandable menu of buttons. It is similar in purpose to a `<select>` element, with the distinction that it does not maintain a value. Instead, each option behaves as an action button.

Usage

Render a Dropdown Menu with a set of controls:

```
import { DropdownMenu } from '@wordpress/components';
import {
    more,
    arrowLeft,
    arrowRight,
    arrowUp,
    arrowDown,
} from '@wordpress/icons';

const MyDropdownMenu = () => (
    <DropdownMenu
        icon={ more }
        label="Select a direction"
        controls={ [
            {
                title: 'Up',
                icon: arrowUp,
                onClick: () => console.log( 'up' ),
            },
```

```
            {
                title: 'Right',
                icon: arrowRight,
                onClick: () => console.log( 'right' ),
            },
            {
                title: 'Down',
                icon: arrowDown,
                onClick: () => console.log( 'down' ),
            },
            {
                title: 'Left',
                icon: arrowLeft,
                onClick: () => console.log( 'left' ),
            },
        ] }
    />
);
```

Alternatively, specify a `children` function which returns elements valid for use in a DropdownMenu: `MenuItem`, `MenuItemsChoice`, or `MenuGroup`.

```
import { DropdownMenu, MenuGroup, MenuItem } from '@wordpress/components';
import { more, arrowUp, arrowDown, trash } from '@wordpress/icons';

const MyDropdownMenu = () => (
    <DropdownMenu icon={ more } label="Select a direction">
        { ( { onClose } ) => (
            <>
                <MenuGroup>
                    <MenuItem icon={ arrowUp } onClick={ onClose }>
                        Move Up
                    </MenuItem>
                    <MenuItem icon={ arrowDown } onClick={ onClose }>
                        Move Down
                    </MenuItem>
                </MenuGroup>
                <MenuGroup>
                    <MenuItem icon={ trash } onClick={ onClose }>
                        Remove
```

```
                    </MenuItem>
                </MenuGroup>
            </>
        ) }
    </DropdownMenu>
);
```

Props

The component accepts the following props:

`icon`: `string | null`

The Dashicon icon slug to be shown in the collapsed menu button.

- Required: No
- Default: `"menu"`

See also: https://developer.wordpress.org/resource/dashicons/

`label`: `string`

A human-readable label to present as accessibility text on the focused collapsed menu button.

- Required: Yes

`controls`: `DropdownOption[] | DropdownOption[][]`

An array or nested array of objects describing the options to be shown in the expanded menu.

Each object should include an `icon` Dashicon slug string, a human-readable `title` string, `isDisabled` boolean flag and an `onClick` function callback to invoke when the option is selected.

A valid DropdownMenu must specify a `controls` or `children` prop, or both.
– Required: No

`children`: `( callbackProps: DropdownCallbackProps ) => ReactNode`

A function render prop which should return an element or elements valid for use in a DropdownMenu: `MenuItem`, `MenuItemsChoice`, or `MenuGroup`. Its first argument is a props object including the same values as given to a `Dropdown`'s `renderContent` (`isOpen`, `onToggle`, `onClose`).

A valid DropdownMenu must specify a `controls` or `children` prop, or both.

- Required: No

See also: https://developer.wordpress.org/resource/dashicons/

`className`: `string`

A class name to apply to the dropdown menu's toggle element wrapper.

- Required: No

`popoverProps`: `DropdownProps[ 'popoverProps' ]`

Properties of `popoverProps` object will be passed as props to the nested `Popover` component.
Use this object to modify props available for the `Popover` component that are not already exposed in the `DropdownMenu` component, e.g.: the direction in which the popover should open relative to its parent node set with `position` prop.

- Required: No

`toggleProps`: `ToggleProps`

Properties of `toggleProps` object will be passed as props to the nested `Button` component in the `renderToggle` implementation of the `Dropdown` component used internally.
Use this object to modify props available for the `Button` component that are not already exposed in the `DropdownMenu` component, e.g.: the tooltip text displayed on hover set with `tooltip` prop.

- Required: No

`menuProps`: `NavigableContainerProps`

Properties of `menuProps` object will be passed as props to the nested `NavigableMenu` component in the `renderContent` implementation of the `Dropdown` component used internally.
Use this object to modify props available for the `NavigableMenu` component that are not already exposed in the `DropdownMenu` component, e.g.: the orientation of the menu set with `orientation` prop.

- Required: No

`disableOpenOnArrowDown`: `boolean`

In some contexts, the arrow down key used to open the dropdown menu might need to be disabled—for example when that key is used to perform another action.

- Required: No

- Default: `false`

`defaultOpen`: `boolean`

The open state of the dropdown menu when initially rendered. Use when you do not need to control its open state. It will be overridden by the `open` prop if it is specified on the component's first render.

- Required: No

`open`: `boolean`

The controlled open state of the dropdown menu. Must be used in conjunction with `onToggle`.

- Required: No

`onToggle`: `( willOpen: boolean ) => void`

A callback invoked when the state of the dropdown changes from open to closed and vice versa.

- Required: No

□□□ #2
□ □□□ □□□ 2026-10-31 18:12:57 UTC
□ □□□ □□□ 2026-10-31 18:12:58 UTC