CircularOptionPicker

- Usage
- Props
  - className: string
  - actions: ReactNode
  - options: ReactNode
  - children: ReactNode
  - presentation: 'listbox' | 'toggle-buttons' | 'command-buttons'
  - loop: boolean
  - aria-labelledby: string
  - aria-label: string
- Subcomponents
  - CircularOptionPicker.ButtonAction
    - Props
      - className: string
      - children: ReactNode
      - Inherited props
  - CircularOptionPicker.DropdownLinkAction
    - Props
      - className: string
      - linkText: string
      - dropdownProps: object
      - buttonProps: object

This component is not exported, and therefore can only be used internally to the `@wordpress/components` package.

`CircularOptionPicker` is a component that displays a set of options as circular buttons.

Usage

```
import { useState } from 'react';
import { CircularOptionPicker } from '../circular-option-picker';

const Example = () => {
    const [ currentColor, setCurrentColor ] = useState();
    const colors = [
        { color: '#f00', name: 'Red' },
        { color: '#0f0', name: 'Green' },
        { color: '#00f', name: 'Blue' },
    ];
    const colorOptions = (
        <>
            { colors.map( ( { color, name }, index ) => {
                return (
                    <CircularOptionPicker.Option
                        key={ `${ color }-${ index }` }
                        tooltipText={ name }
                        style={ { backgroundColor: color, color } }
                        isSelected={ index === currentColor }
                        onClick={ () => setCurrentColor( index ) }
                    />
                );
            } ) }
        </>
    );
    return (
        <CircularOptionPicker
                options={ colorOptions }
                actions={
                    <CircularOptionPicker.ButtonAction
                        onClick={ () => setCurrentColor( undefined ) }
                    >
                        { 'Clear' }
                    </CircularOptionPicker.ButtonAction>
                }
            />
```

```
    );
};
```

Props

`className`: `string`

A CSS class to apply to the wrapper element.

- Required: No

`actions`: `ReactNode`

The action(s) to be rendered after the options, such as a ‘clear’ button as seen in `ColorPalette`.

Usually a `CircularOptionPicker.ButtonAction` or `CircularOptionPicker.DropdownLinkAction` component.

- Required: No

`options`: `ReactNode`

The options to be rendered, such as color swatches.

Usually a `CircularOptionPicker.Option` component.

- Required: No

`children`: `ReactNode`

The child elements.

- Required: No

`presentation`: `'listbox' | 'toggle-buttons' | 'command-buttons'`

Controls the swatches' interaction and accessibility semantics.

- `listbox` uses one tab stop for the picker. Arrow keys move between options. Options expose selection with `aria-selected`.
- `toggle-buttons` uses a tab stop for each swatch. Swatches expose selection with `aria-pressed`.
- `command-buttons` uses a tab stop for each swatch. Swatches do not expose selection. Use this presentation when each swatch runs a command, such as opening an editor.
- Required: No
- Default: `listbox`

`loop`: `boolean`

Prevents arrow-key navigation from wrapping around. Only used with the `listbox` presentation.

- Required: No
- Default: `true`

`aria-labelledby`: `string`

The ID reference list of one or more elements that label the wrapper element.

- Required: No

`aria-label`: `string`

The label for the wrapper element. Not used if an 'aria-labelledby' is provided.

- Required: No
- Default: `Custom color picker`

Subcomponents

`CircularOptionPicker.ButtonAction`

A `ButtonAction` is an action that is rendered as a button alongside the options themselves.

A common use case is a 'clear' button to deselect the currently selected option.

Props

`className`: `string`

A CSS class to apply to the underlying `Button` component.

- Required: No

`children`: `ReactNode`

The button’s children.

- Required: No

Inherited props

`CircularOptionPicker.ButtonAction` also inherits all of the `Button` props, except for `href` and `target`.

`CircularOptionPicker.DropdownLinkAction`

`CircularOptionPicker.DropdownLinkAction` is an action that’s hidden behind a dropdown toggle. The button is formatted as a link and rendered as an `anchor` element.

Props

`className`: `string`

A CSS class to apply to the underlying `Dropdown` component.

- Required: No

`linkText`: `string`

The text to be displayed on the button.

- Required: Yes

`dropdownProps`: `object`

The props for the underlying `Dropdown` component.

Inherits all of the `Dropdown` props, except for `className` and `renderToggle`.

- Required: Yes

`buttonProps` : `object`

Props for the underlying `Button` component.

Inherits all of the `Button` props, except for `href`, `target`, and `children`.

- Required: No

□□□ #2
□ □□□ □□□ 2026-10-31 18:16:52 UTC
□ □□□ □□□ 2026-10-31 18:16:54 UTC