Radio buttons React component

Radio

Radio buttons allow the user to select one option from a set.

Use radio buttons when the user needs to see all available options. If available options can be collapsed, consider using a Select component because it uses less space.

Radio buttons should have the most commonly used option selected by default.

{{“component”: “modules/components/ComponentLinkHeader.js”}}

Radio group

`RadioGroup` is a helpful wrapper used to group `Radio` components that provides an easier API, and proper keyboard accessibility to the group.

{{“demo”: “RadioButtonsGroup.js”}}

Direction

To lay out the buttons horizontally, set the `row` prop:

{{“demo”: “RowRadioButtonsGroup.js”}}

Controlled

You can control the radio with the `value` and `onChange` props:

{{“demo”: “ControlledRadioButtonsGroup.js”}}

Standalone radio buttons

`Radio` can also be used standalone, without the RadioGroup wrapper.

{{“demo”: “RadioButtons.js”}}

Size

Use the `size` prop or customize the font size of the svg icons to change the size of the radios.

{{"demo": "SizeRadioButtons.js"}}

Color

{{"demo": "ColorRadioButtons.js"}}

Label placement

You can change the placement of the label with the `FormControlLabel` component's `labelPlacement` prop:

{{"demo": "FormControlLabelPlacement.js"}}

Show error

In general, radio buttons should have a value selected by default. If this is not the case, you can display an error if no value is selected when the form is submitted:

{{"demo": "ErrorRadios.js"}}

Customization

Here is an example of customizing the component. You can learn more about this in the overrides documentation page.

{{"demo": "CustomizedRadios.js"}}

`useRadioGroup`

For advanced customization use cases, a `useRadioGroup()` hook is exposed. It returns the context value of the parent radio group. The Radio component uses this hook internally.

API

```
import { useRadioGroup } from '@mui/material/RadioGroup';
```

Returns `value` (*object*):

- `value.name` (*string* [optional]): The name used to reference the value of the control.
- `value.onChange` (*func* [optional]): Callback fired when a radio button is selected.
- `value.value` (*any* [optional]): Value of the selected radio button.

Example {{“demo”: “UseRadioGroup.js”}}

When to use

- Checkboxes vs. Radio Buttons

Accessibility

(WAI-ARIA: https://www.w3.org/TR/wai-aria-practices/#radiobutton)

- All form controls should have labels, and this includes radio buttons, checkboxes, and switches. In most cases, this is done by using the `<label>` element (FormControlLabel).
- When a label can’t be used, it’s necessary to add an attribute directly to the input component. In this case, you can apply the additional attribute (e.g. `aria-label`, `aria-labelledby`, `title`) via the `inputProps` property.

```
<Radio
  value="radioA"
  inputProps={{
    'aria-label': 'Radio A',
  }}
/>
```