RadioGroup

Multi-Platform Component | Figma UI KIT

First item

Second item

Third item

```
<RadioGroup
  items={[
    {
      label: 'First item',
      id: 'first'
    },
    {
      label: 'Second item',
      id: 'second'
    },
    {
      label: 'Third item',
      id: 'third'
    }
  ]}
/>
```

Introduction

A RadioGroup renders a group of Radio buttons in a list, with a text label for each radio button. Exactly one radio button may be selected at a time.

Follow the appropriate instructions to add this component in to your app.

Guidance

Use RadioGroup where the user needs to select one item from multiple options, and where those options can be explained simply with one short phrase

(generally, up to around 5 words).

Alternatives

- If the options require information beyond one short phrase, consider RadioCardGroup
- If the options are very simple and can be described in one or two words or numbers, consider ButtonGroup
- If it should be possible to select more than one item at a time, consider CheckboxGroup

Labeling the group

Three props are available for labelling the group to users:

- `legend` sets the main title for the group and should be a short, simple name for what the user is to choose.
- `hint` is an optional short sentence clarifying the `legend` to help users understand what choice they are being asked to make and what the result of their choice will be.
- `tooltip` allows further clarifying information to be included but not shown by default, only shown on pressing a TooltipButton. This should be used for information that will help many users make a choice, but which is not needed by all users.

Data allowance Choose a monthly internet data limit

- 20 GB, with 10 GB first month bonus
- 30 GB, with 20 GB first month bonus
- 40 GB, with 30 GB first month bonus

```
<RadioGroup
  legend="Data allowance"
  hint="Choose a monthly internet data limit"
  tooltip="Amounts are listed in GB (gigabytes). 1 GB is roughly
40 minutes of Blu-ray quality video."
  items={[
    {
      label: '20 GB, with 10 GB first month bonus',
      id: '20'
```

```
    },
    {
      label: '30 GB, with 20 GB first month bonus',
      id: '30'
    },
    {
      label: '40 GB, with 30 GB first month bonus',
      id: '40'
    }
  ]}
/>
```

Items

Use the `items` prop to pass an array of objects describing each Radio in the group:

- `label`: text displayed alongside the radio button
- `id`: identifier used to store which Radio is selected (uses index if undefined). This is written into HTML as an id attr on web, so should be unique and not match IDs used elsewhere on a page.
- `onChange`: optional function called on selection, in addition to updating the group's selection state

First item

Second item

Third item

```
<RadioGroup
  items={[
    {
      label: 'First item',
      id: 'first'
    },
    {
      label: 'Second item',
      id: 'second'
    },
    {
      label: 'Third item',
      id: 'third',
```

```
      onChange: (value, event) =>
        console.log('Third item selected', value,
event.currentTarget, event)
    }
  ]}
/>
```

Controlled

The selection state may be controlled by a parent by passing `checkedId` and an `onChange` function that updates the parent's state. If the component is controlled in this way, `checkedId` must never be undefined; pass `null` or an empty string if there is no selection.

Currently selected: ""

- First item
- Second item
- Third item

```
function ControlledRadioButtonGroup() {
  const [checkedId, setCheckedId] = useState('')
  return (
    <StackView space={3}>
      <Typography>Currently selected: "{checkedId}"</Typography>
      <RadioGroup
        checkedId={checkedId}
        onChange={setCheckedId}
        items={[
          { label: 'First item', id: 'first' },
          { label: 'Second item', id: 'second' },
          { label: 'Third item', id: 'third' }
        ]}
      />
    </StackView>
  )
}
```

Uncontrolled

If `checkedId` is not provided, RadioGroup manages its own state. An initial selection may be provided by passing `initialCheckedId`.

First item

Second item

Third item

```
<RadioGroup
  initialCheckedId="Second item"
  items={[
    { label: 'First item', id: 'first' },
    { label: 'Second item', id: 'second' },
    { label: 'Third item', id: 'third' }
  ]}
/>
```

`onChange` functions

The `onChange` handler for RadioGroup may be used in both controlled and uncontrolled usage and takes two arguments: `value` (string) and `event` (React SyntheticEvent object).

The `event` object shape varies between web and native applications, and type of user interaction. It is usually more stable to have application logic respond to the `value` and not `event` properties.

On web, `event.currentTarget` will point to the accessible container that holds the radio role and state, and should be used if accessing the raw HTML element is required (`event.target` will vary on mouse clicks depending on where exactly the click occured).

First item

Second item

Third item

```
<RadioGroup
  onChange={(value, event) => console.log(value,
event.currentTarget, event)}
```

```
  items={[
    { label: 'First item', id: 'first' },
    { label: 'Second item', id: 'second' },
    { label: 'Third item', id: 'third' }
  ]}
/>
```

Use in forms

For web forms, the `name` prop may be used to define the name of the group's `<fieldset>` and input elements, and `legend` may be used to provide a title with `<legend>` type for the fieldset.

Choose a product

- First product
- Second product
- Third product

```
<RadioGroup
  legend="Choose a product"
  name="product-choice"
  items={[
    { label: 'First product', id: 'first' },
    { label: 'Second product', id: 'second' },
    { label: 'Third product', id: 'third' }
  ]}
/>
```

Validation

Validation state may be set by passing 'error' or 'success' to the `validation` prop.

Choose 'a' or 'c'

- Option 'a'
- Option 'b'
- Option 'c'

◯ Option 'd'

```
function ValidationExample() {
  const [validation, setValidation] = useState(null)
  const validate = (id) => setValidation(['a', 'c'].includes(id)
? 'success' : 'error')
  return (
    <RadioGroup
      legend="Choose 'a' or 'c'"
      onChange={validate}
      validation={validation}
      feedback={validation && `${validation === 'success' ?
'Valid' : 'Invalid'} item chosen.`}
      items={[
        { label: "Option 'a'", id: 'a' },
        { label: "Option 'b'", id: 'b' },
        { label: "Option 'c'", id: 'c' },
        { label: "Option 'd'", id: 'd' }
      ]}
    />
  )
}
```

A11y guidelines

RadioGroup accepts all the common accessibility props, but also sets accessibility role `'radiogroup'` and controls the accessibility state of children like Radio and Feedback based on current state.

Platform considerations

The component is available on both native platforms and web.

Props

Name	Type	Platform	Default	Description
copy	'en' | 'fr'	standard		Whether the English or

Name	Type	Platform	Default	Description
				French copy will be used (e.g. for accessibility labels).
tokens	tokens	standard		System tokens prop, see tokens for more details
radioTokens	custom	standard		Optional theme token overrides for each inner Radio component
variant	variant	standard		System variant prop, see variants for more details
items	arrayOf	standard		Array of objects containing specifics for each Radio to be rendered in the group.
legend	string	standard		Main text used to describe this group, used in Fieldset's Legend element.

Name	Type	Platform	Default	Description
hint	string	standard		Optional additional text giving more detail to help a user make a choice.
hintPosition	'inline' | 'below'	standard		Position of the hint relative to label. Use `below` to display a larger hint below the label.
tooltip	string	standard		Optional tooltip text content to include alongside the legend and hint.
validation	'error' | 'success'	standard		Current validation status of the group, passed to the feedback element if there is one.
feedback	string	standard		If provided, a Feedback element is rendered containing this text.

Name	Type	Platform	Default	Description
initialCheckedId	string	standard		If provided, the radio with this id is selected on first render.
checkedId	string	standard		If not undefined, the radio with this id is selected (or none is selected if `null`), and the element's selection state will be controlled by its parent using the `onChange` function.
onChange	func	standard		Function to call on change in selection state. Is required if the selection state is controlled by a parent using checkedId and the input is not readOnly.
readOnly	bool	standard		If true, the radios cannot be selected by the user and simply show

Name	Type	Platform	Default	Description
				their current state.
inactive	bool	standard		If true, the radios cannot be interacted with, elements are set as `disabled` and if the theme supports `inactive` appearances rules, these are applied.
name	string	standard		On Web, this is passed to the `name` attribute of the fieldset and each radio input.

Tokens

In exceptional circumstances, the following tokens can be passed to this component to override its default styles. **Do not do this unless absolutely necessary.** Read more about overriding styles.

▶ View Tokens

Variants

This component does not have any stylistic variants.

Feedback

- Spotted a problem with this component? Raise an issue on GitHub
- See any existing issues for this component
- Contact the team on slack in #ds-support