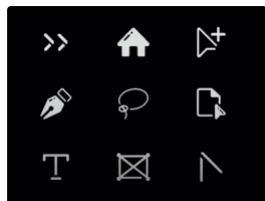


Switch

Switches toggle the state of a single setting on or off.



Design and Development tips in your inbox. Every weekday.

ads via Carbon

Switches are the preferred way to adjust settings on mobile. The option that the switch controls, as well as the state it's in, should be made clear from the corresponding inline label.

[View as Markdown](#)[Feedback](#)[Bundle size](#)[Source](#)[Material Design](#)[Figma](#)[Sketch](#)

Basic switches

[Edit in Chat](#)

JS

TS

[Collapse code](#)

```
import Switch from '@mui/material/Switch';

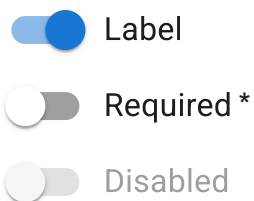
const label = { inputProps: { 'aria-label': 'Switch demo' } };

export default function BasicSwitches() {
  return (
    <div>
      <Switch {...label} defaultChecked />
      <Switch {...label} />
      <Switch {...label} disabled defaultChecked />
      <Switch {...label} disabled />
    </div>
  );
}
```

Label



You can provide a label to the `Switch` thanks to the `FormControlLabel` component.

[Edit in Chat](#)[JS](#)[TS](#)[Collapse code](#)

```
import FormGroup from '@mui/material/FormGroup';
import FormControlLabel from '@mui/material/FormControlLabel';
import Switch from '@mui/material/Switch';

export default function SwitchLabels() {
  return (
    <FormGroup>
      <FormControlLabel control={<Switch defaultChecked />} label="Label" />
      <FormControlLabel required control={<Switch />} label="Required" />
      <FormControlLabel disabled control={<Switch />} label="Disabled" />
    </FormGroup>
  );
}
```

Size



Use the `size` prop to change the size of the switch.

[Edit in Chat](#)[JS](#)[TS](#)[Collapse code](#)

```
import Switch from '@mui/material/Switch';

const label = { inputProps: { 'aria-label': 'Size switch demo' } };

export default function SwitchesSize() {
  return (
    <div>
      <Switch {...label} defaultChecked size="small" />
      <Switch {...label} defaultChecked />
    </div>
  );
}
```

```
);  
}
```

Color

[Edit in Chat](#)

JS

TS

[Collapse code](#)

```
import { alpha, styled } from '@mui/material/styles';  
import { pink } from '@mui/material/colors';  
import Switch from '@mui/material/Switch';  
  
const PinkSwitch = styled(Switch)(({ theme }) => ({  
  '& .MuiSwitch-switchBase.Mui-checked': {  
    color: pink[600],  
    '&:hover': {  
      backgroundColor: alpha(pink[600], theme.palette.action.hoverOpacity),  
    },  
  },  
  '& .MuiSwitch-switchBase.Mui-checked + .MuiSwitch-track': {  
    backgroundColor: pink[600],  
  },  
}));  
  
const label = { inputProps: { 'aria-label': 'Color switch demo' } };  
  
export default function ColorSwitches() {  
  return (  
    <div>  
      <Switch {...label} defaultChecked />  
      <Switch {...label} defaultChecked color="secondary" />  
      <Switch {...label} defaultChecked color="warning" />  
      <Switch {...label} defaultChecked color="default" />  
      <PinkSwitch {...label} defaultChecked />  
    </div>  
  );  
}
```

Controlled



You can control the switch with the `checked` and `onChange` props:

[Edit in Chat](#)

JS

TS

[Collapse code](#)

```
import * as React from 'react';
import Switch from '@mui/material/Switch';

export default function ControlledSwitches() {
  const [checked, setChecked] = React.useState(true);

  const handleChange = (event: React.ChangeEvent<HTMLInputElement>) => {
    setChecked(event.target.checked);
  };

  return (
    <Switch
      checked={checked}
      onChange={handleChange}
      slotProps={{ input: { 'aria-label': 'controlled' } }}
    />
  );
}
```

Switches with FormGroup



`FormGroup` is a helpful wrapper used to group selection controls components that provides an easier API. However, you are encouraged to use [Checkboxes](#) instead if multiple related controls are required. (See: [When to use](#)).

Assign responsibility



Gilad Gray



Jason Killian



Antoine Llorca

Be careful

[Edit in Chat](#)

JS

TS

[Hide code](#)

```
import * as React from 'react';
import FormLabel from '@mui/material/FormLabel';
import FormControl from '@mui/material/FormControl';
import FormGroup from '@mui/material/FormGroup';
import FormControlLabel from '@mui/material/FormControlLabel';
```

```
import FormHelperText from '@mui/material/FormHelperText';
import Switch from '@mui/material/Switch';

export default function SwitchesGroup() {
  const [state, setState] = React.useState({
    gilad: true,
    jason: false,
    antoine: true,
  });

  const handleChange = (event: React.ChangeEvent<HTMLInputElement>) => {
    setState({
      ...state,
      [event.target.name]: event.target.checked,
    });
  };

  return (
    <FormControl component="fieldset" variant="standard">
      <FormLabel component="legend">Assign responsibility</FormLabel>
      <FormGroup>
        <FormControllLabel
          control={
            <Switch checked={state.gilad} onChange={handleChange} name="gilad" />
          }
          label="Gilad Gray"
        />
        <FormControllLabel
          control={
            <Switch checked={state.jason} onChange={handleChange} name="jason" />
          }
          label="Jason Killian"
        />
      </FormGroup>
    </FormControl>
  );
}
```

Customization



Here are some examples of customizing the component. You can learn more about this in the [overrides documentation page](#).

 MUI switch

 Android 12

 iOS style

Off  On

 Edit in Chat

JS

TS

Hide code



```
import FormGroup from '@mui/material/FormGroup';
import FormControlLabel from '@mui/material/FormControlLabel';
import Switch, { SwitchProps } from '@mui/material/Switch';
import Stack from '@mui/material/Stack';
import Typography from '@mui/material/Typography';

const MaterialUISwitch = styled(Switch)(({ theme }) => ({
  width: 62,
  height: 34,
  padding: 7,
  '& .MuiSwitch-switchBase': {
    margin: 1,
    padding: 0,
    transform: 'translateX(6px)',
    '& .Mui-checked': {
      color: '#fff',
      transform: 'translateX(22px)',
      '& .MuiSwitch-thumb:before': {
        backgroundImage: `url('data:image/svg+xml;utf8,<svg xmlns="http://www.w3.org/2000/svg" height=
          '#fff',
        )} d="M4.2 2.51-.7 1.8-1.8.7 1.8.6-1.8L6.7 5.1-1.9-.7-.6-1.8zm15 8.3a6.7 6.7 0 11-6.6-
      },
      '& + .MuiSwitch-track': {
        opacity: 1,
        backgroundColor: '#aab4be',
        ...theme.applyStyles('dark', {
          backgroundColor: '#8796A5',
        }),
      },
    },
  },
  '& .MuiSwitch-thumb': {
    backgroundColor: '#001e3c',
    width: 32,
    height: 32,
    '&::before': {
```

 If you are looking for inspiration, you can check [MUI Treasury's customization examples](#).

You can change the placement of the label:

Label placement

☒ Bottom ☐ End

```
import Switch from '@mui/material/Switch';
import FormGroup from '@mui/material/FormGroup';
import FormControlLabel from '@mui/material/FormControlLabel';
import FormControl from '@mui/material/FormControl';
import FormLabel from '@mui/material/FormLabel';

export default function FormControlLabelPosition() {
  return (
    <FormControl component="fieldset">
      <FormLabel component="legend">Label placement</FormLabel>
      <FormGroup aria-label="position" row>
        <FormControlLabel
          value="bottom"
          control={<Switch color="primary" />}
          label="Bottom"
          labelPlacement="bottom"
        />
        <FormControlLabel
          value="end"
          control={<Switch color="primary" />}
          label="End"
          labelPlacement="end"
        />
      </FormGroup>
    </FormControl>
  );
}
```

When to use

- [Checkboxes vs. Switches](#)

Accessibility

- 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 (for example `aria-label`, `aria-labelledby`, `title`) via the `inputProps` prop.


```
<Switch value="checkedA" inputProps={{ 'aria-label': 'Switch A' }} />
```



Copy

API

See the documentation below for a complete reference to all of the props and classes available to the components mentioned here.

- `<FormControl />`
- `<FormControlLabel />`
- `<FormGroup />`
- `<FormLabel />`
- `<Switch />`

 [Edit this page](#)

Was this page helpful?  

[< Slider](#)

[Text Field >](#)