

Toggle Button

A Toggle Button can be used to group related options.

Ad Premium Themes - Kickstart your application development with a ready-made theme. [Get started](#)
ad by MUI

To emphasize groups of related Toggle buttons, a group should share a common container. The `ToggleButtonGroup` controls the selected state of its child buttons when given its own `value` prop.

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

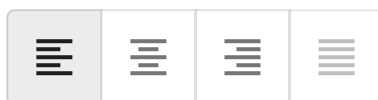
Exclusive selection



With exclusive selection, selecting one option deselects any other.

In this example, text justification toggle buttons present options for left, center, right, and fully justified text (disabled), with only one item available for selection at a time.

Note: Exclusive selection does not enforce that a button must be active. For that effect see [enforce value set](#).

[Edit in Chat](#)

JS

TS

[Hide code](#)

```
import * as React from 'react';
import FormatAlignLeftIcon from '@mui/icons-material/FormatAlignLeft';
import FormatAlignCenterIcon from '@mui/icons-material/FormatAlignCenter';
import FormatAlignRightIcon from '@mui/icons-material/FormatAlignRight';
import FormatAlignJustifyIcon from '@mui/icons-material/FormatAlignJustify';
import ToggleButton from '@mui/material/ToggleButton';
import ToggleButtonGroup from '@mui/material/ToggleButtonGroup';

export default function ToggleButtons() {
  const [alignment, setAlignment] = React.useState<string | null>('left');
```

```
const handleAlignment = (
  event: React.MouseEvent<HTMLElement>,
  newAlignment: string | null,
) => {
  setAlignment(newAlignment);
};

return (
  <ToggleButtonGroup
    value={alignment}
    exclusive
    onChange={handleAlignment}
    aria-label="text alignment"
  >
    <ToggleButton value="left" aria-label="left aligned">
      <FormatAlignLeftIcon />
    </ToggleButton>
    <ToggleButton value="center" aria-label="centered">
      <FormatAlignCenterIcon />
    </ToggleButton>
    <ToggleButton value="right" aria-label="right aligned">
      <FormatAlignRightIcon />
    </ToggleButton>
    <ToggleButton value="justify" aria-label="justified" disabled>
      <FormatAlignJustifyIcon />
    </ToggleButton>
  </ToggleButtonGroup>
);
```

Multiple selection



Multiple selection allows for logically-grouped options, like bold, italic, and underline, to have multiple options selected.



 Edit in Chat

JS

TS

Hide code



```
import * as React from 'react';
import FormatBoldIcon from '@mui/icons-material/FormatBold';
import FormatItalicIcon from '@mui/icons-material/FormatItalic';
import FormatUnderlinedIcon from '@mui/icons-material/FormatUnderlined';
import FormatColorFillIcon from '@mui/icons-material/FormatColorFill';
import ArrowDropDownIcon from '@mui/icons-material/ArrowDropDown';
import ToggleButton from '@mui/material/ToggleButton';
import ToggleButtonGroup from '@mui/material/ToggleButtonGroup';

export default function ToggleButtonsMultiple() {
  const [formats, setFormats] = React.useState(() => ['bold', 'italic']);
```

```
const handleFormat = (
  event: React.MouseEvent<HTMLElement>,
  newFormats: string[],
) => {
  setFormats(newFormats);
};

return (
  <ToggleButtonGroup
    value={formats}
    onChange={handleFormat}
    aria-label="text formatting"
  >
    <ToggleButton value="bold" aria-label="bold">
      <FormatBoldIcon />
    </ToggleButton>
    <ToggleButton value="italic" aria-label="italic">
      <FormatItalicIcon />
    </ToggleButton>
    <ToggleButton value="underlined" aria-label="underlined">
      <FormatUnderlinedIcon />
    </ToggleButton>
    <ToggleButton value="color" aria-label="color" disabled>
      <FormatColorFillIcon />
      <ArrowDropDownIcon />
    </ToggleButton>
  </ToggleButtonGroup>
);
```

Size



For larger or smaller buttons, use the `size` prop.


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


```
import * as React from 'react';
import FormatAlignLeftIcon from '@mui/icons-material/FormatAlignLeft';
import FormatAlignCenterIcon from '@mui/icons-material/FormatAlignCenter';
import FormatAlignRightIcon from '@mui/icons-material/FormatAlignRight';
import FormatAlignJustifyIcon from '@mui/icons-material/FormatAlignJustify';
import Stack from '@mui/material/Stack';
import ToggleButton from '@mui/material/ToggleButton';
```

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

export default function ToggleButtonSizes() {
  const [alignment, setAlignment] = React.useState('left');

  const handleChange = (
    event: React.MouseEvent<HTMLElement>,
    newAlignment: string,
  ) => {
    setAlignment(newAlignment);
  };

  const children = [
    <ToggleButton value="left" key="left">
      <FormatAlignLeftIcon />
    </ToggleButton>,
    <ToggleButton value="center" key="center">
      <FormatAlignCenterIcon />
    </ToggleButton>,
    <ToggleButton value="right" key="right">
      <FormatAlignRightIcon />
    </ToggleButton>,
    <ToggleButton value="justify" key="justify">
      <FormatAlignJustifyIcon />
    </ToggleButton>,
  ];

  const control = {
    value: alignment,
    onChange: handleChange,
    exclusive: true
  };
}
```

Color



WEB

ANDROID

IOS

Edit in Chat

JS

TS

Collapse code



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

export default function ColorToggleButton() {
  const [alignment, setAlignment] = React.useState('web');

  const handleChange = (
    event: React.MouseEvent<HTMLElement>,
    newAlignment: string,
  ) => {
    setAlignment(newAlignment);
  };

  const children = [
    <ToggleButton value="web" key="web">
      <FormatAlignWebIcon />
    </ToggleButton>,
    <ToggleButton value="center" key="center">
      <FormatAlignCenterIcon />
    </ToggleButton>,
    <ToggleButton value="right" key="right">
      <FormatAlignRightIcon />
    </ToggleButton>,
    <ToggleButton value="justify" key="justify">
      <FormatAlignJustifyIcon />
    </ToggleButton>,
  ];

  const control = {
    value: alignment,
    onChange: handleChange,
    exclusive: true
  };
}
```

```

    setAlignment(newAlignment);
  };

  return (
    <ToggleButtonGroup
      color="primary"
      value={alignment}
      exclusive
      onChange={handleChange}
      aria-label="Platform"
    >
      <ToggleButton value="web">Web</ToggleButton>
      <ToggleButton value="android">Android</ToggleButton>
      <ToggleButton value="ios">iOS</ToggleButton>
    </ToggleButtonGroup>
  );
}

```

Vertical buttons



The buttons can be stacked vertically with the `orientation` prop set to "vertical".


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


```

import * as React from 'react';
import ViewListIcon from '@mui/icons-material/ViewList';
import ViewModuleIcon from '@mui/icons-material/ViewModule';
import ViewQuiltIcon from '@mui/icons-material/ViewQuilt';
import ToggleButton from '@mui/material/ToggleButton';
import ToggleButtonGroup from '@mui/material/ToggleButtonGroup';

export default function VerticalToggleButtons() {
  const [view, setView] = React.useState('list');

  const handleChange = (event: React.MouseEvent<HTMLDivElement>, nextView: string) => {
    setView(nextView);
  };

  return (
    <ToggleButtonGroup
      orientation="vertical"

```

```

value={view}
exclusive
onChange={handleChange}
>
<ToggleButton value="list" aria-label="list">
  <ViewListIcon />
</ToggleButton>
<ToggleButton value="module" aria-label="module">
  <ViewModuleIcon />
</ToggleButton>
<ToggleButton value="quilt" aria-label="quilt">
  <ViewQuiltIcon />
</ToggleButton>
</ToggleButtonGroup>
);
}

```

Enforce value set



If you want to enforce that at least one button must be active, you can adapt your `handleChange` function.

```

const handleAlignment = (event, newAlignment) => {
  if (newAlignment !== null) {
    setAlignment(newAlignment);
  }
};

const handleDevices = (event, newDevices) => {
  if (newDevices.length) {
    setDevices(newDevices);
  }
};

```

Copy



Edit in Chat

JS

TS

Hide code



```

import * as React from 'react';
import FormatAlignLeftIcon from '@mui/icons-material/FormatAlignLeft';
import FormatAlignCenterIcon from '@mui/icons-material/FormatAlignCenter';
import FormatAlignRightIcon from '@mui/icons-material/FormatAlignRight';
import LaptopIcon from '@mui/icons-material/Laptop';
import TvIcon from '@mui/icons-material/Tv';
import PhoneAndroidIcon from '@mui/icons-material/PhoneAndroid';
import Stack from '@mui/material/Stack';

```

```
import ToggleButton from '@mui/material/ToggleButton';
import ToggleButtonGroup from '@mui/material/ToggleButtonGroup';

export default function ToggleButtonNotEmpty() {
  const [alignment, setAlignment] = React.useState('left');
  const [devices, setDevices] = React.useState(() => ['phone']);

  const handleAlignment = (
    event: React.MouseEvent<HTMLInputElement>,
    newAlignment: string | null,
  ) => {
    if (newAlignment !== null) {
      setAlignment(newAlignment);
    }
  };

  const handleDevices = (
    event: React.MouseEvent<HTMLInputElement>,
    newDevices: string[],
  ) => {
    if (newDevices.length) {
      setDevices(newDevices);
    }
  };

  return (
    <Stack direction="row" spacing={4}>
      <ToggleButtonGroup
        value={alignment}
        exclusive
```

Standalone toggle button


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


```
import * as React from 'react';
import CheckIcon from '@mui/icons-material/Check';
import ToggleButton from '@mui/material/ToggleButton';

export default function StandaloneToggleButton() {
  const [selected, setSelected] = React.useState(false);

  return (
    <ToggleButton
      value="check"
      selected={selected}
      onChange={() => setSelected((prevSelected) => !prevSelected)}
```

```

>
<CheckIcon />
</ToggleButton>
);
}

```

Customization



Here is an example of customizing the component. You can learn more about this in the [overrides documentation page](#).



 Edit in Chat

JS

TS

Hide code



```

import * as React from 'react';
import { styled } from '@mui/material/styles';
import FormatAlignLeftIcon from '@mui/icons-material/FormatAlignLeft';
import FormatAlignCenterIcon from '@mui/icons-material/FormatAlignCenter';
import FormatAlignRightIcon from '@mui/icons-material/FormatAlignRight';
import FormatAlignJustifyIcon from '@mui/icons-material/FormatAlignJustify';
import FormatBoldIcon from '@mui/icons-material/FormatBold';
import FormatItalicIcon from '@mui/icons-material/FormatItalic';
import FormatUnderlinedIcon from '@mui/icons-material/FormatUnderlined';
import FormatColorFillIcon from '@mui/icons-material/FormatColorFill';
import ArrowDropDownIcon from '@mui/icons-material/ArrowDropDown';
import Divider from '@mui/material/Divider';
import Paper from '@mui/material/Paper';
import ToggleButton from '@mui/material/ToggleButton';
import ToggleButtonGroup, {
  toggleButtonGroupClasses,
} from '@mui/material/ToggleButtonGroup';

const StyledToggleButtonGroup = styled(ToggleButtonGroup)(({ theme }) => ({
  [`&.${toggleButtonGroupClasses.grouped}`]: {
    margin: theme.spacing(0.5),
    border: 0,
    borderRadius: theme.shape.borderRadius,
    [`&.${toggleButtonGroupClasses.disabled}`]: {
      border: 0,
    },
  },
},
),
[`${toggleButtonGroupClasses.middleButton}&.${toggleButtonGroupClasses.lastButton}`]:
  {
    marginLeft: -1,
    borderLeft: '1px solid transparent',
  },
]));

```



```
export default function CustomizedDividers() {
  const [alignment, setAlignment] = React.useState('left');
  const [formats, setFormats] = React.useState(() => ['italic']);
}
```

Spacing



The demos below show how to adjust spacing between toggle buttons in horizontal and vertical orientations.

Horizontal Spacing


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


```
import * as React from 'react';
import FormatAlignLeftIcon from '@mui/icons-material/FormatAlignLeft';
import FormatAlignCenterIcon from '@mui/icons-material/FormatAlignCenter';
import FormatAlignRightIcon from '@mui/icons-material/FormatAlignRight';
import FormatAlignJustifyIcon from '@mui/icons-material/FormatAlignJustify';
import ToggleButton, { toggleButtonClasses } from '@mui/material/ToggleButton';
import ToggleButtonGroup, {
  toggleButtonGroupClasses,
} from '@mui/material/ToggleButtonGroup';
import { styled } from '@mui/material/styles';

const StyledToggleButtonGroup = styled(ToggleButtonGroup)(({ theme }) => ({
  gap: '2rem',
  [`& .${toggleButtonGroupClasses.firstButton}, & .${toggleButtonGroupClasses.middleButton}`]:
    {
      borderTopRightRadius: (theme.vars || theme).shape.borderRadius,
      borderBottomRightRadius: (theme.vars || theme).shape.borderRadius,
    },
  [`& .${toggleButtonGroupClasses.lastButton}, & .${toggleButtonGroupClasses.middleButton}`]:
    {
      borderTopLeftRadius: (theme.vars || theme).shape.borderRadius,
      borderBottomLeftRadius: (theme.vars || theme).shape.borderRadius,
      borderLeft: `1px solid ${theme.vars || theme}.palette.divider`,
    },
  [`& .${toggleButtonGroupClasses.lastButton}.${toggleButtonClasses.disabled}, & .${toggleButtonGroupC
    {
      borderLeft: `1px solid ${theme.vars || theme}.palette.action.disabledBackground`,
    },
})));

export default function HorizontalSpacingToggleButton() {
  const [alignment, setAlignment] = React.useState<string | null>('left');
```

```
const handleAlignment = (
  event: React.MouseEvent<HTMLElement>,
  newAlignment: string | null,
) => {
  setAlignment(newAlignment);
}
```

Vertical Spacing


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


```
import * as React from 'react';
import FormatAlignLeftIcon from '@mui/icons-material/FormatAlignLeft';
import FormatAlignCenterIcon from '@mui/icons-material/FormatAlignCenter';
import FormatAlignRightIcon from '@mui/icons-material/FormatAlignRight';
import FormatAlignJustifyIcon from '@mui/icons-material/FormatAlignJustify';
import ToggleButton, { toggleButtonClasses } from '@mui/material/ToggleButton';
import ToggleButtonGroup, {
  toggleButtonGroupClasses,
} from '@mui/material/ToggleButtonGroup';
import { styled } from '@mui/material/styles';

const StyledToggleButtonGroup = styled(ToggleButtonGroup)(({ theme }) => ({
  gap: '2rem',
  [`& .${toggleButtonGroupClasses.firstButton}, & .${toggleButtonGroupClasses.middleButton}`]:
  {
    borderBottomRightRadius: (theme.vars || theme).shape.borderRadius,
    borderBottomLeftRadius: (theme.vars || theme).shape.borderRadius,
  },
  [`& .${toggleButtonGroupClasses.lastButton}, & .${toggleButtonGroupClasses.middleButton}`]:
  {
    borderTopRightRadius: (theme.vars || theme).shape.borderRadius,
    borderTopLeftRadius: (theme.vars || theme).shape.borderRadius,
    borderTop: `1px solid ${theme.vars || theme}.palette.divider`,
  },
  [`& .${toggleButtonGroupClasses.lastButton}.${toggleButtonClasses.disabled}, & .${toggleButtonGroupC
```

```
borderTop: `1px solid ${theme.vars || theme}.palette.action.disabledBackground`,
  },
}));

export default function VerticalSpacingToggleButton() {
  const [alignment, setAlignment] = React.useState<string | null>('left');

  const handleAlignment = (
    event: React.MouseEvent<HTMLInputElement>,
    newAlignment: string | null,
  ) => {
```

Accessibility

ARIA

- ToggleButtonGroup has `role="group"`. You should provide an accessible label with `aria-label="label"`, `aria-labelledby="id"` or `<label>`.
- ToggleButton sets `aria-pressed="<bool>"` according to the button state. You should label each button with `aria-label`.


Keyboard

At present, toggle buttons are in DOM order. Navigate between them with the tab key. The button behavior follows standard keyboard semantics.

API

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

- [<ToggleButton />](#)
- [<ToggleButtonGroup />](#)

 [Edit this page](#)

Was this page helpful?  

[< Transfer List](#)

[Avatar >](#)