Floating Action Button

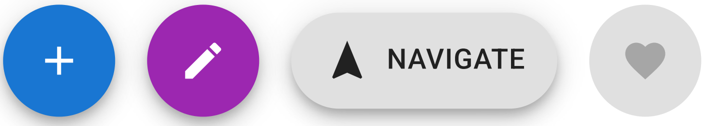
A Floating Action Button (FAB) performs the primary, or most common, action on a screen.

A floating action button appears in front of all screen content, typically as a circular shape with an icon in its center. FABs come in two types: regular, and extended.

Only use a FAB if it is the most suitable way to present a screen's primary action. Only one component is recommended per screen to represent the most common action.

View as Markdown   Feedback   Bundle size   Source   Material Design   Figma   Sketch

Basic FAB

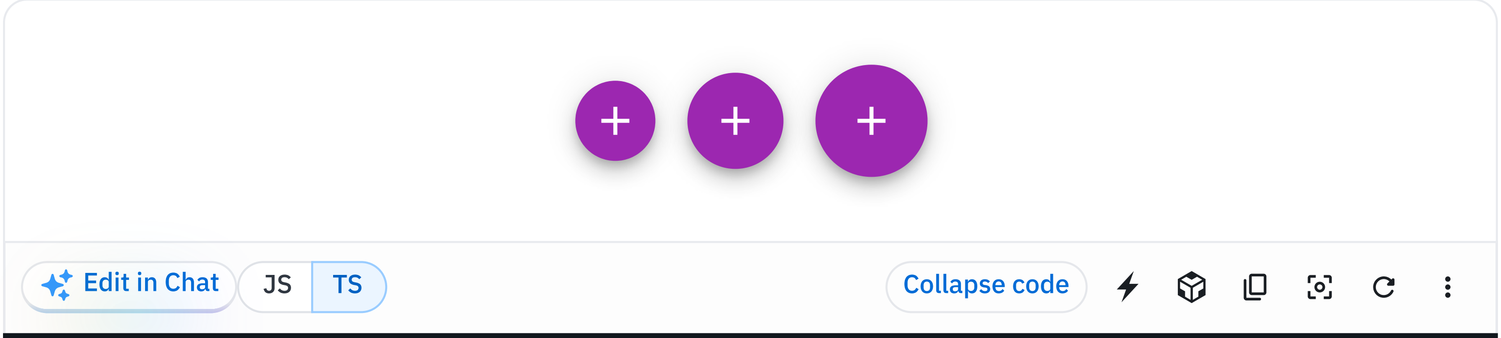


Edit in Chat   JS   TS   Collapse code

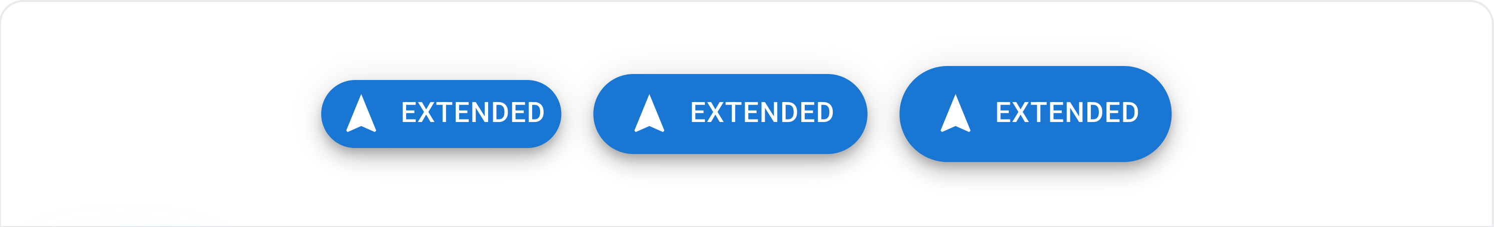
```
import Box from '@mui/material/Box';
import Fab from '@mui/material/Fab';
import AddIcon from '@mui/icons-material/Add';
import EditIcon from '@mui/icons-material/Edit';
import FavoriteIcon from '@mui/icons-material/Favorite';
import NavigationIcon from '@mui/icons-material/Navigation';

export default function FloatingActionButtons() {
  return (
    <Box sx={{ '& > :not(style)': { m: 1 } }}>
      <Fab color="primary" aria-label="add">
        <AddIcon />
      </Fab>
      <Fab color="secondary" aria-label="edit">
        <EditIcon />
      </Fab>
      <Fab variant="extended">
        <NavigationIcon sx={{ mr: 1 }} />
        Navigate
      </Fab>
      <Fab disabled aria-label="like">
        <FavoriteIcon />
      </Fab>
    </Box>
  );
}
```

Size

By default, the size is `large`. Use the `size` prop for smaller floating action buttons.



```
import Box from '@mui/material/Box';
import Fab from '@mui/material/Fab';
import AddIcon from '@mui/icons-material/Add';

export default function FloatingActionButtonSize() {
  return (
    <Box sx={{ '& > :not(style)': { m: 1 } }}>
      <Fab size="small" color="secondary" aria-label="add">
        <AddIcon />
      </Fab>
      <Fab size="medium" color="secondary" aria-label="add">
        <AddIcon />
      </Fab>
      <Fab color="secondary" aria-label="add">
        <AddIcon />
      </Fab>
    </Box>
  );
}
```



Edit in Chat JS TS Collapse code

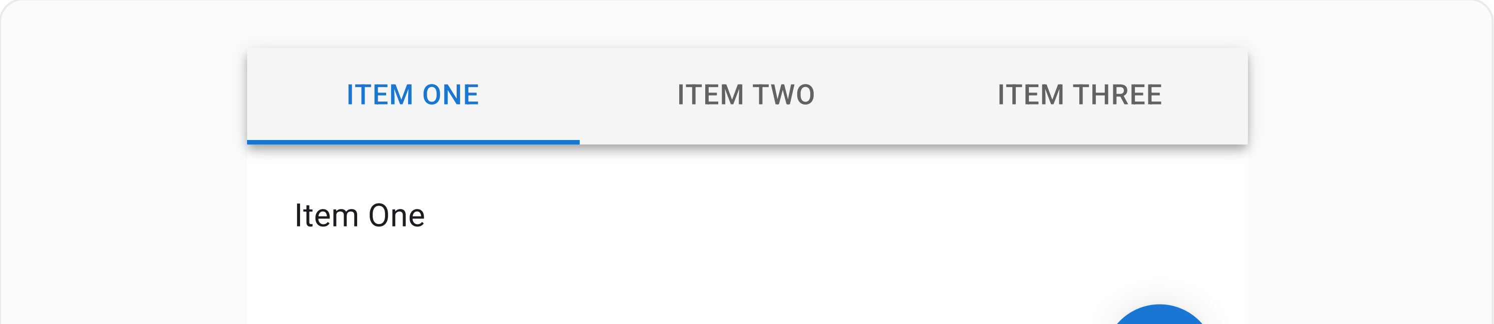
```
import Box from '@mui/material/Box';
import Fab from '@mui/material/Fab';
import NavigationIcon from '@mui/icons-material/Navigation';

export default function FloatingActionButtonExtendedSize() {
  return (
    <Box sx={{ '& > :not(style)': { m: 1 } }}>
      <Fab variant="extended" size="small" color="primary">
        <NavigationIcon sx={{ mr: 1 }} />
        Extended
      </Fab>
      <Fab variant="extended" size="medium" color="primary">
        <NavigationIcon sx={{ mr: 1 }} />
        Extended
      </Fab>
      <Fab variant="extended" color="primary">
        <NavigationIcon sx={{ mr: 1 }} />
        Extended
      </Fab>
    </Box>
  );
}
```

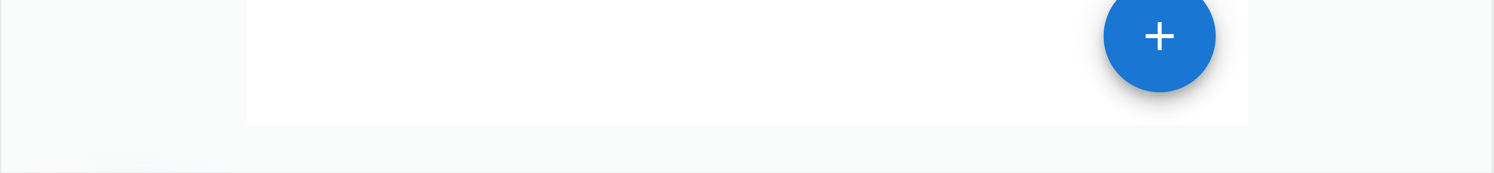
Animation

The floating action button animates onto the screen as an expanding piece of material, by default.

A floating action button that spans multiple lateral screens (such as tabbed screens) should briefly disappear, then reappear if its action changes.

The Zoom transition can be used to achieve this. Note that since both the exiting and entering animations are triggered at the same time, we use `enterDelay` to allow the outgoing Floating Action Button's animation to finish before the new one enters.





Edit in Chat JS TS Hide code

```
import * as React from 'react';
import { useTheme } from '@mui/material/styles';
import AppBar from '@mui/material/AppBar';
import Tabs from '@mui/material/Tabs';
import Tab from '@mui/material/Tab';
import Typography from '@mui/material/Typography';
import Zoom from '@mui/material/Zoom';
import Fab from '@mui/material/Fab';
import AddIcon from '@mui/icons-material/Add';
import EditIcon from '@mui/icons-material/Edit';
import UpIcon from '@mui/icons-material/KeyboardArrowUp';
import { green } from '@mui/material/colors';
import Box from '@mui/material/Box';
import { SxProps } from '@mui/system';

interface TabPanelProps {
  children?: React.ReactNode;
  dir?: string;
  index: number;
  value: number;
}

function TabPanel(props: TabPanelProps) {
  const { children, value, index, ...other } = props;

  return (
    <Typography
      component="div"
      role="tabpanel"
      hidden={value !== index}
      id={`action-tabpanel-${index}`}
      aria-labelledby={`action-tab-${index}`}
      {...other}
    >
      {value === index && <Box sx={{ p: 3 }}>{children}</Box>}
    </Typography>
  );
}
```

API

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

- `<Fab />`

Edit this page

Was this page helpful?

‹ Checkbox

Number Field ›