Speed Dial

When pressed, a floating action button can display three to six related actions in the form of a Speed Dial.

If more than six actions are needed, something other than a FAB should be used to present them.

View as Markdown Feedback Bundle size Source WAI-ARIA Material Design

Figma Sketch

Basic speed dial

The floating action button can display related actions.

Edit in Chat JS TS Hide code

```
import Box from '@mui/material/Box';
import SpeedDial from '@mui/material/SpeedDial';
import SpeedDialIcon from '@mui/material/SpeedDialIcon';
import SpeedDialAction from '@mui/material/SpeedDialAction';
import FileCopyIcon from '@mui/icons-material/FileCopyOutlined';
import SaveIcon from '@mui/icons-material/Save';
import PrintIcon from '@mui/icons-material/Print';
import ShareIcon from '@mui/icons-material/Share';

const actions = [
  { icon: <FileCopyIcon />, name: 'Copy' },
  { icon: <SaveIcon />, name: 'Save' },
  { icon: <PrintIcon />, name: 'Print' },
  { icon: <ShareIcon />, name: 'Share' },
];

export default function BasicSpeedDial() {
  return (
    <Box sx={{ height: 320, transform: 'translateZ(0px)', flexGrow: 1 }}>
      <SpeedDial
        ariaLabel="SpeedDial basic example"
        sx={{ position: 'absolute', bottom: 16, right: 16 }}
        icon={<SpeedDialIcon />}
      >
        {actions.map((action) => (
          <SpeedDialAction
            key={action.name}
            icon={action.icon}
            slotProps={{
              tooltip: {
                title: action.name,
              },
            }}
          />
        ))}
      </SpeedDial>
    </Box>
  );
```

Playground

Hidden

Direction

● Up ○ Right ○ Down ○ Left

Edit in Chat JS TS Hide code

```
import * as React from 'react';
import { styled } from '@mui/material/styles';
import Box from '@mui/material/Box';
import FormControl from '@mui/material/FormControl';
import FormControlLabel from '@mui/material/FormControlLabel';
import FormLabel from '@mui/material/FormLabel';
import Radio from '@mui/material/Radio';
import RadioGroup from '@mui/material/RadioGroup';
import Switch from '@mui/material/Switch';
import SpeedDial, { SpeedDialProps } from '@mui/material/SpeedDial';
import SpeedDialIcon from '@mui/material/SpeedDialIcon';
import SpeedDialAction from '@mui/material/SpeedDialAction';
import FileCopyIcon from '@mui/icons-material/FileCopyOutlined';
import SaveIcon from '@mui/icons-material/Save';
import PrintIcon from '@mui/icons-material/Print';
import ShareIcon from '@mui/icons-material/Share';

const StyledSpeedDial = styled(SpeedDial)(({ theme }) => ({
  position: 'absolute',
  '&.MuiSpeedDial-directionUp, &.MuiSpeedDial-directionLeft': {
    bottom: theme.spacing(2),
    right: theme.spacing(2),
  },
  '&.MuiSpeedDial-directionDown, &.MuiSpeedDial-directionRight': {
    top: theme.spacing(2),
    left: theme.spacing(2),
  },
}));

const actions = [
  { icon: <FileCopyIcon />, name: 'Copy' },
  { icon: <SaveIcon />, name: 'Save' },
  { icon: <PrintIcon />, name: 'Print' },
  { icon: <ShareIcon />, name: 'Share' },
];

export default function PlaygroundSpeedDial() {
  const [direction, setDirection] =
```

Controlled speed dial

The open state of the component can be controlled with the `open`/`onOpen`/`onClose` props.

Edit in Chat JS TS Hide code

```
import * as React from 'react';
import Box from '@mui/material/Box';
import SpeedDial from '@mui/material/SpeedDial';
import SpeedDialIcon from '@mui/material/SpeedDialIcon';
import SpeedDialAction from '@mui/material/SpeedDialAction';
import FileCopyIcon from '@mui/icons-material/FileCopyOutlined';
import SaveIcon from '@mui/icons-material/Save';
import PrintIcon from '@mui/icons-material/Print';
import ShareIcon from '@mui/icons-material/Share';

const actions = [
  { icon: <FileCopyIcon />, name: 'Copy' },
  { icon: <SaveIcon />, name: 'Save' },
  { icon: <PrintIcon />, name: 'Print' },
  { icon: <ShareIcon />, name: 'Share' },
];

export default function ControlledOpenSpeedDial() {
  const [open, setOpen] = React.useState(false);
  const handleOpen = () => setOpen(true);
  const handleClose = () => setOpen(false);

  return (
    <Box sx={{ height: 320, transform: 'translateZ(0px)', flexGrow: 1 }}>
      <SpeedDial
        ariaLabel="SpeedDial controlled open example"
        sx={{ position: 'absolute', bottom: 16, right: 16 }}
        icon={<SpeedDialIcon />}
        onClose={handleClose}
        onOpen={handleOpen}
        open={open}
      >
        {actions.map((action) => (
          <SpeedDialAction
```

```
            key={action.name}
            icon={action.icon}
            slotProps={{
              tooltip: {
```

Custom close icon

You can provide an alternate icon for the closed and open states using the `icon` and `openIcon` props of the `SpeedDialIcon` component.

Edit in Chat JS TS Hide code

```
import Box from '@mui/material/Box';
import SpeedDial from '@mui/material/SpeedDial';
import SpeedDialIcon from '@mui/material/SpeedDialIcon';
import SpeedDialAction from '@mui/material/SpeedDialAction';
import FileCopyIcon from '@mui/icons-material/FileCopyOutlined';
import SaveIcon from '@mui/icons-material/Save';
import PrintIcon from '@mui/icons-material/Print';
import ShareIcon from '@mui/icons-material/Share';
import EditIcon from '@mui/icons-material/Edit';

const actions = [
  { icon: <FileCopyIcon />, name: 'Copy' },
  { icon: <SaveIcon />, name: 'Save' },
  { icon: <PrintIcon />, name: 'Print' },
  { icon: <ShareIcon />, name: 'Share' },
];

export default function OpenIconSpeedDial() {
  return (
    <Box sx={{ height: 320, transform: 'translateZ(0px)', flexGrow: 1 }}>
```

```
      <SpeedDial
        ariaLabel="SpeedDial openIcon example"
        sx={{ position: 'absolute', bottom: 16, right: 16 }}
        icon={<SpeedDialIcon openIcon={<EditIcon />} />}
      >
        {actions.map((action) => (
          <SpeedDialAction
            key={action.name}
            icon={action.icon}
            slotProps={{
              tooltip: {
                title: action.name,
              },
            }}
          />
        ))}
      </SpeedDial>
```

Persistent action tooltips

The SpeedDialActions tooltips can be displayed persistently so that users don't have to long-press to see the tooltip on touch devices.

It is enabled here across all devices for demo purposes, but in production it could use the `isTouch` logic to conditionally set the prop.

Edit in Chat JS TS Hide code

```
import * as React from 'react';
import Box from '@mui/material/Box';
import Backdrop from '@mui/material/Backdrop';
import SpeedDial from '@mui/material/SpeedDial';
import SpeedDialIcon from '@mui/material/SpeedDialIcon';
import SpeedDialAction from '@mui/material/SpeedDialAction';
import FileCopyIcon from '@mui/icons-material/FileCopyOutlined';
import SaveIcon from '@mui/icons-material/Save';
import PrintIcon from '@mui/icons-material/Print';
import ShareIcon from '@mui/icons-material/Share';

const actions = [
  { icon: <FileCopyIcon />, name: 'Copy' },
  { icon: <SaveIcon />, name: 'Save' },
  { icon: <PrintIcon />, name: 'Print' },
  { icon: <ShareIcon />, name: 'Share' },
];

export default function SpeedDialTooltipOpen() {
  const [open, setOpen] = React.useState(false);
  const handleOpen = () => setOpen(true);
  const handleClose = () => setOpen(false);

  return (
    <Box sx={{ height: 330, transform: 'translateZ(0px)', flexGrow: 1 }}>
      <Backdrop open={open} />
      <SpeedDial
        ariaLabel="SpeedDial tooltip example"
        sx={{ position: 'absolute', bottom: 16, right: 16 }}
        icon={<SpeedDialIcon />}
        onClose={handleClose}
        onOpen={handleOpen}
        open={open}
      >
        {actions.map((action) => (
          <SpeedDialAction
            key={action.name}
            icon={action.icon}
```

Accessibility

ARIA

Required

- You should provide an `ariaLabel` for the speed dial component.
- You should provide a tooltip title using `slotProps.tooltip.title` for each speed dial action.

Provided

- The Fab has `aria-haspopup`, `aria-expanded` and `aria-controls` attributes.
- The speed dial actions container has `role="menu"` and `aria-orientation` set according to the direction.

- The speed dial actions have `role="menuitem"`, and an `aria-describedby` attribute that references the associated tooltip.

Keyboard

- The speed dial opens on focus.
- The Space and Enter keys trigger the selected speed dial action, and toggle the speed dial open state.
- The cursor keys move focus to the next or previous speed dial action. (Note that any cursor direction can be used initially to open the speed dial. This enables the expected behavior for the actual or perceived orientation of the speed dial, for example for a screen reader user who perceives the speed dial as a drop-down menu.)
- The Escape key closes the speed dial and, if a speed dial action was focused, returns focus to the Fab.

API

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

- `<SpeedDial />`
- `<SpeedDialAction />`
- `<SpeedDialIcon />`