Chip

Chips are compact elements that represent an input, attribute, or action.

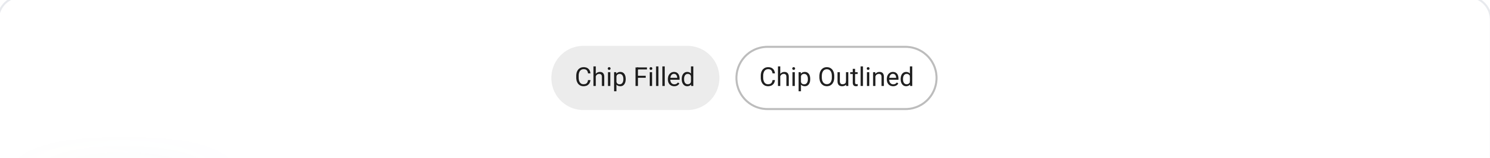
Chips allow users to enter information, make selections, filter content, or trigger actions.

While included here as a standalone component, the most common use will be in some form of input, so some of the behavior demonstrated here is not shown in context.

View as Markdown
Feedback
Bundle size
Source
Material Design
Figma
Sketch

Basic chip

The `Chip` component supports outlined and filled styling.



Edit in Chat
JS
TS
Collapse code

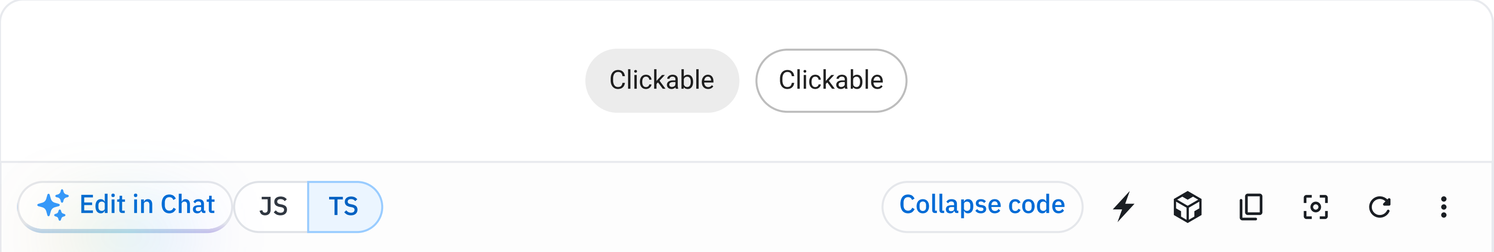
```
import Chip from '@mui/material/Chip';
import Stack from '@mui/material/Stack';

export default function BasicChips() {
  return (
    <Stack direction="row" spacing={1}>
      <Chip label="Chip Filled" />
      <Chip label="Chip Outlined" variant="outlined" />
    </Stack>
  );
}
```

Chip actions

You can use the following actions.

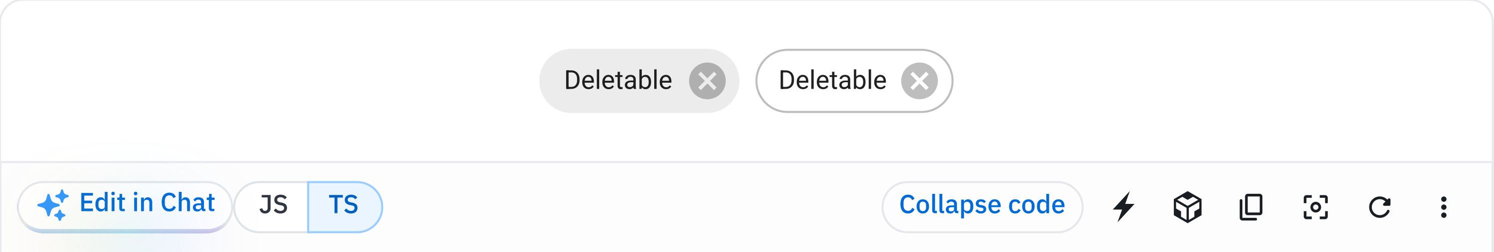
- Chips with the `onClick` prop defined change appearance on focus, hover, and click.
- Chips with the `onDelete` prop defined will display a delete icon which changes appearance on hover.

Clickable



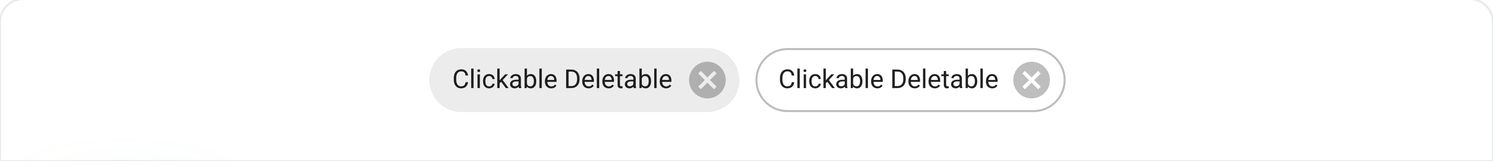
```
import Chip from '@mui/material/Chip';
import Stack from '@mui/material/Stack';

export default function ClickableChips() {
  const handleClick = () => {
    console.info('You clicked the Chip.');
  };

  return (
    <Stack direction="row" spacing={1}>
      <Chip label="Clickable" onClick={handleClick} />
      <Chip label="Clickable" variant="outlined" onClick={handleClick} />
    </Stack>
  );
}
```

Deletable



```
import Chip from '@mui/material/Chip';
import Stack from '@mui/material/Stack';

export default function DeletableChips() {
  const handleDelete = () => {
    console.info('You clicked the delete icon.');
  };

  return (
    <Stack direction="row" spacing={1}>
      <Chip label="Deletable" onDelete={handleDelete} />
      <Chip label="Deletable" variant="outlined" onDelete={handleDelete} />
    </Stack>
  );
}
```

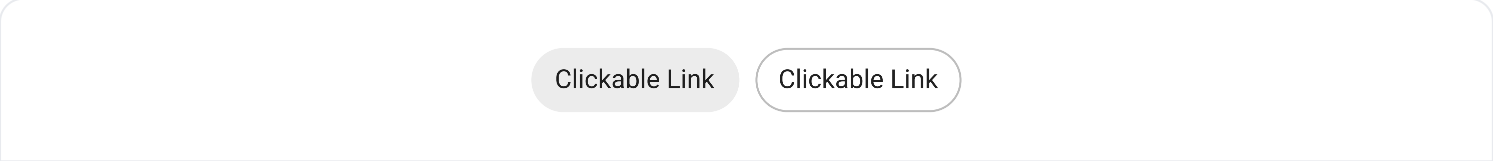
Clickable and deletable



Edit in Chat JS TS Collapse code

```
import Chip from '@mui/material/Chip';
import Stack from '@mui/material/Stack';

export default function ClickableAndDeletableChips() {
  const handleClick = () => {
    console.info('You clicked the Chip.');
  };

  const handleDelete = () => {
    console.info('You clicked the delete icon.');
  };

  return (
    <Stack direction="row" spacing={1}>
      <Chip
        label="Clickable Deletable"
        onClick={handleClick}
        onDelete={handleDelete}
      />
      <Chip
        label="Clickable Deletable"
        variant="outlined"
        onClick={handleClick}
        onDelete={handleDelete}
      />
    </Stack>
  );
}
```

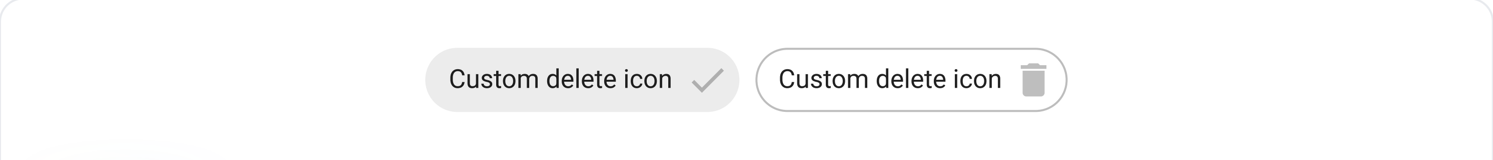
Clickable link



Edit in Chat JS TS Collapse code

```
import Chip from '@mui/material/Chip';
import Stack from '@mui/material/Stack';

export default function ClickableLinkChips() {
  return (
    <Stack direction="row" spacing={1}>
      <Chip label="Clickable Link" component="a" href="#basic-chip" clickable />
      <Chip
        label="Clickable Link"
        component="a"
        href="#basic-chip"
        variant="outlined"
        clickable
      />
    </Stack>
  );
}
```

Custom delete icon



Edit in Chat JS TS Collapse code

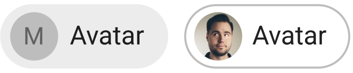
```
import Chip from '@mui/material/Chip';
import Stack from '@mui/material/Stack';
import DoneIcon from '@mui/icons-material/Done';
import DeleteIcon from '@mui/icons-material/Delete';

export default function CustomDeleteIconChips() {
  const handleClick = () => {
    console.info('You clicked the Chip.');
  };

  const handleDelete = () => {
    console.info('You clicked the delete icon.');
  };

  return (
    <Stack direction="row" spacing={1}>
      <Chip
        label="Custom delete icon"
        onClick={handleClick}
        onDelete={handleDelete}
        deleteIcon={<DoneIcon />}
      />
      <Chip
        label="Custom delete icon"
        onClick={handleClick}
        onDelete={handleDelete}
        deleteIcon={<DeleteIcon />}
        variant="outlined"
      />
    </Stack>
  );
}
```

Chip adornments

You can add ornaments to the beginning of the component.

Use the `avatar` prop to add an avatar or use the `icon` prop to add an icon.

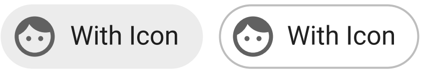
Avatar chip



Edit in Chat JS TS Collapse code

```
import Avatar from '@mui/material/Avatar';
import Chip from '@mui/material/Chip';
import Stack from '@mui/material/Stack';

export default function AvatarChips() {
  return (
    <Stack direction="row" spacing={1}>
      <Chip avatar={<Avatar>M</Avatar>} label="Avatar" />
      <Chip
        avatar={<Avatar alt="Natacha" src="/static/images/avatar/1.jpg" />}
        label="Avatar"
        variant="outlined"
      />
    </Stack>
  );
}
```

Icon chip

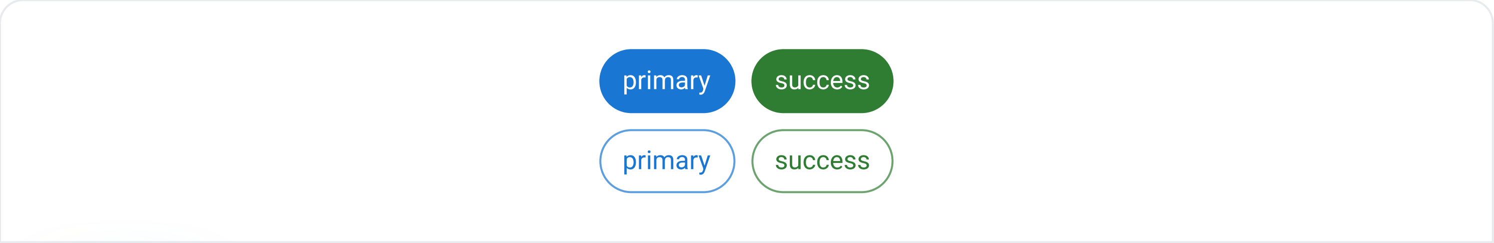


Edit in Chat JS TS Collapse code

```
import Chip from '@mui/material/Chip';
import Stack from '@mui/material/Stack';
import FaceIcon from '@mui/icons-material/Face';

export default function IconChips() {
  return (
    <Stack direction="row" spacing={1}>
      <Chip icon={<FaceIcon />} label="With Icon" />
      <Chip icon={<FaceIcon />} label="With Icon" variant="outlined" />
    </Stack>
  );
}
```

Color chip

You can use the `color` prop to define a color from theme palette.



Edit in Chat JS TS Collapse code

```
import Chip from '@mui/material/Chip';
import Stack from '@mui/material/Stack';

export default function ColorChips() {
  return (
    <Stack spacing={1} sx={{ alignItems: 'center' }}>
      <Stack direction="row" spacing={1}>
        <Chip label="primary" color="primary" />
        <Chip label="success" color="success" />
      </Stack>
      <Stack direction="row" spacing={1}>
        <Chip label="primary" color="primary" variant="outlined" />
        <Chip label="success" color="success" variant="outlined" />
      </Stack>
    </Stack>
  );
}
```

Sizes chip

You can use the `size` prop to define a small Chip.

Small Small

Edit in Chat JS TS Collapse code

```
import Chip from '@mui/material/Chip';
import Stack from '@mui/material/Stack';

export default function SizesChips() {
  return (
    <Stack direction="row" spacing={1}>
      <Chip label="Small" size="small" />
      <Chip label="Small" size="small" variant="outlined" />
    </Stack>
  );
}
```

Multiline chip

By default, Chips displays labels only in a single line. To have them support multiline content, use the `sx` prop to add `height:auto` to the Chip component, and `whiteSpace: normal` to the `label` styles.

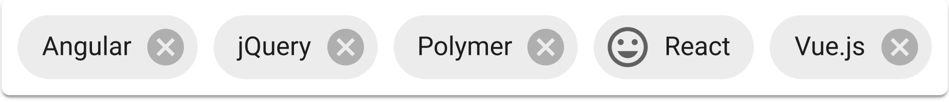
This is a chip that has multiple lines.

Edit in Chat JS TS Collapse code

```
import Chip from '@mui/material/Chip';
import Box from '@mui/material/Box';

export default function MultilineChips() {
  return (
    <Box sx={{ width: 100 }}>
      <Chip
        sx={{
          height: 'auto',
          '& .MuiChip-label': {
            display: 'block',
            whiteSpace: 'normal',
          },
        }}
        label="This is a chip that has multiple lines."
      />
    </Box>
  );
}
```

Chip array

An example of rendering multiple chips from an array of values. Deleting a chip removes it from the array. Note that since no `onClick` prop is defined, the `Chip` can be focused, but does not gain depth while clicked or touched.



Edit in Chat JS TS Hide code

```
import * as React from 'react';
import { styled } from '@mui/material/styles';
import Chip from '@mui/material/Chip';
import Paper from '@mui/material/Paper';
import TagFacesIcon from '@mui/icons-material/TagFaces';

interface ChipData {
  key: number;
  label: string;
}

const ListItem = styled('li')(({ theme }) => ({
  margin: theme.spacing(0.5),
}));

export default function ChipsArray() {
  const [chipData, setChipData] = React.useState<readonly ChipData[]>([
    { key: 0, label: 'Angular' },
    { key: 1, label: 'jQuery' },
    { key: 2, label: 'Polymer' },
    { key: 3, label: 'React' },
    { key: 4, label: 'Vue.js' },
  ]);

  const handleDelete = (chipToDelete: ChipData) => () => {
    setChipData((chips) => chips.filter((chip) => chip.key !== chipToDelete.key));
  };

  return (
    <Paper
      sx={{
        display: 'flex',
        justifyContent: 'center',
        flexWrap: 'wrap',
        listStyle: 'none',
        p: 0.5,
        m: 0,
      }}
```

Chip playground

Chip Component

variant

● filled ○ outlined

color

● default ○ primary ○ secondary ○ error ○ info ○ success ○ warning

size

● medium ○ small

icon

● none ○ icon

avatar

● none ○ letter ○ img

onDelete

● none ○ default ○ custom

```
<Chip />
```

Copy

Accessibility

If the Chip is deletable or clickable then it is a button in tab order. When the Chip is focused (for example when tabbing) releasing (`keyup` event) `Backspace` or `Delete` will call the `onDelete` handler while releasing `Escape` will blur the Chip.

API

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

- `<Chip />`

Edit this page

Was this page helpful?

‹ Badge

Divider ›