

When To Use

- Used for selecting multiple values from several options.
- If you use only one checkbox, it is the same as using Switch to toggle between two states.
The difference is that Switch will trigger the state change directly, but Checkbox just marks the state as changed and this needs to be submitted.

Examples

Basic

```
import React from 'react';
import { Checkbox } from 'antd';
import type { CheckboxProps } from 'antd';

const onChange: CheckboxProps['onChange'] = (e) => {
  console.log(`checked = ${e.target.checked}`);
};

const App: React.FC = () => <Checkbox onChange=
{onChange}>Checkbox</Checkbox>;

export default App;
```

Disabled

```
import React from 'react';
import { Checkbox } from 'antd';

const App: React.FC = () => (
  <>
    <Checkbox defaultChecked={false} disabled />
    <br />
    <Checkbox indeterminate disabled />
    <br />
    <Checkbox defaultChecked disabled />
  </>
);

export default App;
```

Controlled Checkbox

```
import React, { useState } from 'react';
import { Button, Checkbox } from 'antd';
```

```

import type { CheckboxProps } from 'antd';

const App: React.FC = () => {
  const [checked, setChecked] = useState(true);
  const [disabled, setDisabled] = useState(false);

  const toggleChecked = () => {
    setChecked(!checked);
  };

  const toggleDisable = () => {
    setDisabled(!disabled);
  };

  const onChange: CheckboxProps['onChange'] = (e) => {
    console.log('checked = ', e.target.checked);
    setChecked(e.target.checked);
  };

  const label = `${checked ? 'Checked' : 'Unchecked'}-${disabled ?
'Disabled' : 'Enabled'}`;

  return (
    <>
      <p style={{ marginBottom: '20px' }}>
        <Checkbox checked={checked} disabled={disabled} onChange=
{onChange}>
          {label}
        </Checkbox>
      </p>
      <p>
        <Button type="primary" size="small" onClick={toggleChecked}>
          {!checked ? 'Check' : 'Uncheck'}
        </Button>
        <Button style={{ margin: '0 10px' }} type="primary" size="small"
onClick={toggleDisable}>
          {!disabled ? 'Disable' : 'Enable'}
        </Button>
      </p>
    </>
  );
};

export default App;

```

Checkbox Group

```

import React from 'react';
import { Checkbox } from 'antd';
import type { GetProp } from 'antd';

const onChange: GetProp<typeof Checkbox.Group, 'onChange'> =
  (checkedValues) => {
    console.log('checked = ', checkedValues);
  };

const plainOptions = ['Apple', 'Pear', 'Orange'];

const options = [
  { label: 'Apple', value: 'Apple' },
  { label: 'Pear', value: 'Pear' },
  { label: 'Orange', value: 'Orange' },
];

const optionsWithDisabled = [
  { label: 'Apple', value: 'Apple' },
  { label: 'Pear', value: 'Pear' },
  { label: 'Orange', value: 'Orange', disabled: false },
];

const App: React.FC = () => (
  <>
    <Checkbox.Group options={plainOptions} defaultValue={['Apple']}
    onChange={onChange} />
    <br />
    <br />
    <Checkbox.Group options={options} defaultValue={['Pear']} onChange=
    {onChange} />
    <br />
    <br />
    <Checkbox.Group
      options={optionsWithDisabled}
      disabled
      defaultValue={['Apple']}
      onChange={onChange}
    />
  </>
);

export default App;

```

Check all

```

import React, { useState } from 'react';
import { Checkbox, Divider } from 'antd';
import type { CheckboxProps } from 'antd';

const CheckboxGroup = Checkbox.Group;

const plainOptions = ['Apple', 'Pear', 'Orange'];
const defaultCheckedList = ['Apple', 'Orange'];

const App: React.FC = () => {
  const [checkedList, setCheckedList] = useState<string[]>(
    defaultCheckedList);

  const checkAll = plainOptions.length === checkedList.length;
  const indeterminate = checkedList.length > 0 && checkedList.length <
    plainOptions.length;

  const onChange = (list: string[]) => {
    setCheckedList(list);
  };

  const onCheckAllChange: CheckboxProps['onChange'] = (e) => {
    setCheckedList(e.target.checked ? plainOptions : []);
  };

  return (
    <>
      <Checkbox indeterminate={indeterminate} onChange={onCheckAllChange}
checked={checkAll}>
        Check all
      </Checkbox>
      <Divider />
      <CheckboxGroup options={plainOptions} value={checkedList} onChange=
{onCheckAllChange} />
    </>
  );
};

export default App;

```

Use with Grid

```

import React from 'react';
import { Checkbox, Col, Row } from 'antd';
import type { GetProp } from 'antd';

```

```

const onChange: GetProp<typeof Checkbox.Group, 'onChange'> =
  (checkedValues) => {
    console.log('checked = ', checkedValues);
  };

const App: React.FC = () => (
  <Checkbox.Group style={{ width: '100%' }} onChange={onChange}>
    <Row>
      <Col span={8}>
        <Checkbox value="A">A</Checkbox>
      </Col>
      <Col span={8}>
        <Checkbox value="B">B</Checkbox>
      </Col>
      <Col span={8}>
        <Checkbox value="C">C</Checkbox>
      </Col>
      <Col span={8}>
        <Checkbox value="D">D</Checkbox>
      </Col>
      <Col span={8}>
        <Checkbox value="E">E</Checkbox>
      </Col>
    </Row>
  </Checkbox.Group>
);

export default App;

```

Same line

Debug

```

import React from 'react';
import { Checkbox, ConfigProvider, Radio, Space } from 'antd';

const sharedStyle: React.CSSProperties = {
  border: '1px solid red',
  marginBottom: 16,
};

const App: React.FC = () => (
  <div>
    <Space style={sharedStyle} align="center">
      <Checkbox value="light" />
    </Space>
  </div>
);

```

```

    <div>Bamboo</div>
    <Checkbox value="little">Little</Checkbox>
  </Space>

  <Space style={sharedStyle} align="center">
    <Radio value="light" />
    <div>Bamboo</div>
    <Radio value="little">Little</Radio>
  </Space>

  <div
    style={{
      ...sharedStyle,
      display: 'flex',
      alignItems: 'center',
    }}
  >
    <Checkbox value="light" />
    <div>Bamboo</div>
    <Checkbox value="little">Little</Checkbox>
  </div>

  <div
    style={{
      ...sharedStyle,
      display: 'flex',
      alignItems: 'center',
    }}
  >
    <Radio value="light" />
    <div>Bamboo</div>
    <Radio value="little">Little</Radio>
  </div>

  <div>
    <ConfigProvider
      theme={{
        token: {
          controlHeight: 48,
        },
      }}
    >
      <Checkbox>Aligned</Checkbox>
    </ConfigProvider>
  </div>

```

```

    <div>
      <Checkbox>
        <span style={{ fontSize: 32 }}>Aligned</span>
      </Checkbox>
    </div>
  </div>
);

export default App;

```

Disabled to show Tooltip

Debug

```

import React from 'react';
import { Checkbox, Popover } from 'antd';

const App: React.FC = () => (
  <div style={{ padding: 56 }}>
    <Popover content="xxxx" trigger="hover">
      <Checkbox disabled checked />
    </Popover>
  </div>
);

export default App;

```

customize lineWidth

Debug

```

import React from 'react';
import { Checkbox, ConfigProvider } from 'antd';

const App: React.FC = () => (
  <>
    <ConfigProvider
      theme={{
        components: {
          Checkbox: {
            lineWidth: 6,
          },
        },
      }}
    >
      <Checkbox checked />
    </ConfigProvider>
  </>
);

```

```

        <Checkbox />
      </ConfigProvider>
      <Checkbox checked />
      <Checkbox />
    </>
  );

  export default App;

```

API

Common props ref: [Common props](#)

Checkbox

Property	Description	Type	Default	Version
autoFocus	If get focus when component mounted	boolean	false	
checked	Specifies whether the checkbox is selected	boolean	false	
defaultChecked	Specifies the initial state: whether or not the checkbox is selected	boolean	false	
disabled	If disable checkbox	boolean	false	
indeterminate	The indeterminate checked state of checkbox	boolean	false	
onChange	The callback function that is triggered when the state changes	(e: CheckboxChangeEvent) => void	-	
onBlur	Called when leaving the component	function()	-	
onFocus	Called when entering the component	function()	-	

Checkbox Group

Property	Description	Type	Default	Version
defaultValue	Default selected value	(string number) []	[]	

disabled	If disable all checkboxes	boolean	false	
name	The name property of all input[type="checkbox"] children	string	-	
options	Specifies options	string[] number[] Option[]	[]	
value	Used for setting the currently selected value	(string number boolean)[]	[]	
onChange	The callback function that is triggered when the state changes	(checkedValue: T[]) => void	-	

Option

```
interface Option {
  label: string;
  value: string;
  disabled?: boolean;
}
```

Methods

Checkbox

Name	Description	Version
blur()	Remove focus	
focus()	Get focus	
nativeElement	Returns the DOM node of the Checkbox	5.17.3

Design Token

FAQ

Why not work in Form.Item?

Form.Item default bind value to `value` property, but Checkbox value property is `checked` . You can use `valuePropName` to change bind property.

```
<Form.Item name="fieldA" valuePropName="checked">
  <Checkbox />
</Form.Item>
```