何时使用 {#when-to-use}

- 需要从一组相关联的数据集合进行选择，例如省市区，公司层级，事物分类等。
- 从一个较大的数据集合中进行选择时，用多级分类进行分隔，方便选择。
- 比起 Select 组件，可以在同一个浮层中完成选择，有较好的体验。

代码演示

基本

```
import React from 'react';
import type { CascaderProps } from 'antd';
import { Cascader } from 'antd';

interface Option {
  value: string;
  label: string;
  children?: Option[];
}

const options: Option[] = [
  {
    value: 'zhejiang',
    label: 'Zhejiang',
    children: [
      {
        value: 'hangzhou',
        label: 'Hangzhou',
        children: [
          {
            value: 'xihu',
            label: 'West Lake',
          },
        ],
      },
    ],
  },
  {
    value: 'jiangsu',
    label: 'Jiangsu',
    children: [
      {
        value: 'nanjing',
        label: 'Nanjing',
        children: [
          {
            value: 'zhonghuamen',
```

```
            label: 'Zhong Hua Men',
          },
        ],
      },
    ],
  },
];

const onChange: CascaderProps<Option>['onChange'] = (value) => {
  console.log(value);
};

const App: React.FC = () => (
  <Cascader options={options} onChange={onChange} placeholder="Please
select" />
);

export default App;
```

默认值

```
import React from 'react';
import type { CascaderProps } from 'antd';
import { Cascader } from 'antd';

interface Option {
  value: string;
  label: string;
  children?: Option[];
}

const options: Option[] = [
  {
    value: 'zhejiang',
    label: 'Zhejiang',
    children: [
      {
        value: 'hangzhou',
        label: 'Hangzhou',
        children: [
          {
            value: 'xihu',
            label: 'West Lake',
          },
        ],
      },
```

```
    ],
  },
  {
    value: 'jiangsu',
    label: 'Jiangsu',
    children: [
      {
        value: 'nanjing',
        label: 'Nanjing',
        children: [
          {
            value: 'zhonghuamen',
            label: 'Zhong Hua Men',
          },
        ],
      },
    ],
  },
];

const onChange: CascaderProps<Option>['onChange'] = (value) => {
  console.log(value);
};

const App: React.FC = () => (
  <Cascader defaultValue={['zhejiang', 'hangzhou', 'xihu']} options=
{options} onChange={onChange} />
);

export default App;
```

可以自定义显示

```
import React, { useState } from 'react';
import type { CascaderProps } from 'antd';
import { Cascader } from 'antd';

interface Option {
  value: string;
  label: string;
  children?: Option[];
}

const options: Option[] = [
  {
    value: 'zhejiang',
```

```
    label: 'Zhejiang',
    children: [
      {
        value: 'hangzhou',
        label: 'Hangzhou',
      },
    ],
  },
  {
    value: 'jiangsu',
    label: 'Jiangsu',
    children: [
      {
        value: 'nanjing',
        label: 'Nanjing',
      },
    ],
  },
];

const App: React.FC = () => {
  const [text, setText] = useState('Unselect');

  const onChange: CascaderProps<Option>['onChange'] = (_, selectedOptions)
=> {
    setText(selectedOptions.map((o) => o.label).join(', '));
  };

  return (
    <span>
      {text}
       
      <Cascader options={options} onChange={onChange}>
        <a>Change city</a>
      </Cascader>
    </span>
  );
};

export default App;
```

移入展开

```
import React from 'react';
import type { CascaderProps } from 'antd';
import { Cascader } from 'antd';
```

```tsx
interface Option {
  value: string;
  label: string;
  children?: Option[];
}

const options: Option[] = [
  {
    value: 'zhejiang',
    label: 'Zhejiang',
    children: [
      {
        value: 'hangzhou',
        label: 'Hangzhou',
        children: [
          {
            value: 'xihu',
            label: 'West Lake',
          },
        ],
      },
    ],
  },
  {
    value: 'jiangsu',
    label: 'Jiangsu',
    children: [
      {
        value: 'nanjing',
        label: 'Nanjing',
        children: [
          {
            value: 'zhonghuamen',
            label: 'Zhong Hua Men',
          },
        ],
      },
    ],
  },
];

const onChange: CascaderProps<Option>['onChange'] = (value) => {
  console.log(value);
};
```

```
// Just show the latest item.
const displayRender = (labels: string[]) => labels[labels.length - 1];

const App: React.FC = () => (
  <Cascader
    options={options}
    expandTrigger="hover"
    displayRender={displayRender}
    onChange={onChange}
  />
);

export default App;
```

禁用选项

```
import React from 'react';
import type { CascaderProps } from 'antd';
import { Cascader } from 'antd';

interface Option {
  value: string;
  label: string;
  disabled?: boolean;
  children?: Option[];
}

const options: Option[] = [
  {
    value: 'zhejiang',
    label: 'Zhejiang',
    children: [
      {
        value: 'hangzhou',
        label: 'Hangzhou',
        children: [
          {
            value: 'xihu',
            label: 'West Lake',
          },
        ],
      },
    ],
  },
  {
    value: 'jiangsu',
```

```
    label: 'Jiangsu',
    disabled: true,
    children: [
      {
        value: 'nanjing',
        label: 'Nanjing',
        children: [
          {
            value: 'zhonghuamen',
            label: 'Zhong Hua Men',
          },
        ],
      },
    ],
  },
];

const onChange: CascaderProps<Option>['onChange'] = (value) => {
  console.log(value);
};

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

export default App;
```

选择即改变

```
import React from 'react';
import type { CascaderProps } from 'antd';
import { Cascader } from 'antd';

interface Option {
  value: string;
  label: string;
  children?: Option[];
}

const options: Option[] = [
  {
    value: 'zhejiang',
    label: 'Zhejiang',
    children: [
      {
        value: 'hangzhou',
        label: 'Hanzhou',
```

```
        children: [
          {
            value: 'xihu',
            label: 'West Lake',
          },
        ],
      },
    ],
  },
  {
    value: 'jiangsu',
    label: 'Jiangsu',
    children: [
      {
        value: 'nanjing',
        label: 'Nanjing',
        children: [
          {
            value: 'zhonghuamen',
            label: 'Zhong Hua Men',
          },
        ],
      },
    ],
  },
];

const onChange: CascaderProps<Option>['onChange'] = (value) => {
  console.log(value);
};

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

export default App;
```

多选

```
import React from 'react';
import type { CascaderProps } from 'antd';
import { Cascader } from 'antd';

interface Option {
  value: string | number;
  label: string;
  children?: Option[];
```

```
  disableCheckbox?: boolean;
}

const options: Option[] = [
  {
    label: 'Light',
    value: 'light',
    children: Array.from({ length: 20 }).map((_, index) => ({
      label: `Number ${index}`,
      value: index,
    })),
  },
  {
    label: 'Bamboo',
    value: 'bamboo',
    children: [
      {
        label: 'Little',
        value: 'little',
        children: [
          {
            label: 'Toy Fish',
            value: 'fish',
            disableCheckbox: true,
          },
          {
            label: 'Toy Cards',
            value: 'cards',
          },
          {
            label: 'Toy Bird',
            value: 'bird',
          },
        ],
      },
    ],
  },
];

const onChange: CascaderProps<Option, 'value', true>['onChange'] = (value)
=> {
  console.log(value);
};

const App: React.FC = () => (
  <Cascader
```

```
    style={{ width: '100%' }}
    options={options}
    onChange={onChange}
    multiple
    maxTagCount="responsive"
  />
);

export default App;
```

自定义回填方式

```
import React from 'react';
import type { CascaderProps } from 'antd';
import { Cascader } from 'antd';

const { SHOW_CHILD } = Cascader;

interface Option {
  value: string | number;
  label: string;
  children?: Option[];
}
const options: Option[] = [
  {
    label: 'Light',
    value: 'light',
    children: Array.from({ length: 20 }).map((_, index) => ({
      label: `Number ${index}`,
      value: index,
    })),
  },
  {
    label: 'Bamboo',
    value: 'bamboo',
    children: [
      {
        label: 'Little',
        value: 'little',
        children: [
          {
            label: 'Toy Fish',
            value: 'fish',
          },
          {
            label: 'Toy Cards',
```

```
            value: 'cards',
          },
          {
            label: 'Toy Bird',
            value: 'bird',
          },
        ],
      },
    ],
  },
];

const App: React.FC = () => {
  const onChange: CascaderProps<Option, 'value', true>['onChange'] =
(value) => {
    console.log(value);
  };
  return (
    <>
      <Cascader
        style={{ width: '100%' }}
        options={options}
        onChange={onChange}
        multiple
        maxTagCount="responsive"
        showCheckedStrategy={SHOW_CHILD}
        defaultValue={[
          ['bamboo', 'little', 'fish'],
          ['bamboo', 'little', 'cards'],
          ['bamboo', 'little', 'bird'],
        ]}
      />
      <br />
      <br />
      <Cascader
        style={{ width: '100%' }}
        options={options}
        onChange={onChange}
        multiple
        maxTagCount="responsive"
        defaultValue={[['bamboo']]}
      />
    </>
  );
};
```

```
export default App;
```

大小

```
import React from 'react';
import type { CascaderProps } from 'antd';
import { Cascader } from 'antd';

interface Option {
  value: string;
  label: string;
  children?: Option[];
}

const options: Option[] = [
  {
    value: 'zhejiang',
    label: 'Zhejiang',
    children: [
      {
        value: 'hangzhou',
        label: 'Hangzhou',
        children: [
          {
            value: 'xihu',
            label: 'West Lake',
          },
        ],
      },
    ],
  },
  {
    value: 'jiangsu',
    label: 'Jiangsu',
    children: [
      {
        value: 'nanjing',
        label: 'Nanjing',
        children: [
          {
            value: 'zhonghuamen',
            label: 'Zhong Hua Men',
          },
        ],
      },
```

```
    ],
  },
];

const onChange: CascaderProps<Option>['onChange'] = (value) => {
  console.log(value);
};

const App: React.FC = () => (
  <>
    <Cascader size="large" options={options} onChange={onChange} />
    <br />
    <br />
    <Cascader options={options} onChange={onChange} />
    <br />
    <br />
    <Cascader size="small" options={options} onChange={onChange} />
    <br />
    <br />
  </>
);

export default App;
```

自定义已选项

```
import React from 'react';
import { Cascader } from 'antd';
import type { CascaderProps, GetProp } from 'antd';

type DefaultOptionType = GetProp<CascaderProps, 'options'>[number];

interface Option {
  value: string;
  label: string;
  children?: Option[];
  code?: number;
}

const options: Option[] = [
  {
    value: 'zhejiang',
    label: 'Zhejiang',
    children: [
      {
        value: 'hangzhou',
```

```
        label: 'Hangzhou',
        children: [
          {
            value: 'xihu',
            label: 'West Lake',
            code: 752100,
          },
        ],
      },
    ],
  },
  {
    value: 'jiangsu',
    label: 'Jiangsu',
    children: [
      {
        value: 'nanjing',
        label: 'Nanjing',
        children: [
          {
            value: 'zhonghuamen',
            label: 'Zhong Hua Men',
            code: 453400,
          },
        ],
      },
    ],
  },
];

const handleAreaClick = (
  e: React.MouseEvent<HTMLAnchorElement>,
  label: string,
  option: DefaultOptionType,
) => {
  e.stopPropagation();
  console.log('clicked', label, option);
};

const displayRender: CascaderProps<Option>['displayRender'] = (labels,
selectedOptions = []) =>
  labels.map((label, i) => {
    const option = selectedOptions[i];
    if (i === labels.length - 1) {
      return (
        <span key={option.value}>
```

```
          {label} (<a onClick={(e) => handleAreaClick(e, label, option)}>
{option.code}</a>)
        </span>
      );
    }
    return <span key={option.value}>{label} / </span>;
  });

const App: React.FC = () => (
  <Cascader
    options={options}
    defaultValue={['zhejiang', 'hangzhou', 'xihu']}
    displayRender={displayRender}
    style={{ width: '100%' }}
  />
);

export default App;
```

搜索

```
import React from 'react';
import { Cascader } from 'antd';
import type { CascaderProps, GetProp } from 'antd';

type DefaultOptionType = GetProp<CascaderProps, 'options'>[number];

interface Option {
  value: string;
  label: string;
  children?: Option[];
  disabled?: boolean;
}

const options: Option[] = [
  {
    value: 'zhejiang',
    label: 'Zhejiang',
    children: [
      {
        value: 'hangzhou',
        label: 'Hangzhou',
        children: [
          {
            value: 'xihu',
            label: 'West Lake',
```

```
          },
          {
            value: 'xiasha',
            label: 'Xia Sha',
            disabled: true,
          },
        ],
      },
    ],
  },
  {
    value: 'jiangsu',
    label: 'Jiangsu',
    children: [
      {
        value: 'nanjing',
        label: 'Nanjing',
        children: [
          {
            value: 'zhonghuamen',
            label: 'Zhong Hua men',
          },
        ],
      },
    ],
  },
];

const onChange: CascaderProps<Option>['onChange'] = (value,
selectedOptions) => {
  console.log(value, selectedOptions);
};

const filter = (inputValue: string, path: DefaultOptionType[]) =>
  path.some(
    (option) => (option.label as
string).toLowerCase().indexOf(inputValue.toLowerCase()) > -1,
  );

const App: React.FC = () => (
  <Cascader
    options={options}
    onChange={onChange}
    placeholder="Please select"
    showSearch={{ filter }}
    onSearch={(value) => console.log(value)}
```

```
  />
);

export default App;
```

动态加载选项

```
import React, { useState } from 'react';
import type { CascaderProps } from 'antd';
import { Cascader } from 'antd';

interface Option {
  value?: string | number | null;
  label: React.ReactNode;
  children?: Option[];
  isLeaf?: boolean;
}

const optionLists: Option[] = [
  {
    value: 'zhejiang',
    label: 'Zhejiang',
    isLeaf: false,
  },
  {
    value: 'jiangsu',
    label: 'Jiangsu',
    isLeaf: false,
  },
];

const App: React.FC = () => {
  const [options, setOptions] = useState<Option[]>(optionLists);

  const onChange: CascaderProps<Option>['onChange'] = (value,
selectedOptions) => {
    console.log(value, selectedOptions);
  };

  const loadData = (selectedOptions: Option[]) => {
    const targetOption = selectedOptions[selectedOptions.length - 1];

    // load options lazily
    setTimeout(() => {
      targetOption.children = [
        {
```

```
          label: `${targetOption.label} Dynamic 1`,
          value: 'dynamic1',
        },
        {
          label: `${targetOption.label} Dynamic 2`,
          value: 'dynamic2',
        },
      ];
      setOptions([...options]);
    }, 1000);
  };

  return <Cascader options={options} loadData={loadData} onChange=
{onChange} changeOnSelect />;
};

export default App;
```

自定义字段名

```
import React from 'react';
import type { CascaderProps } from 'antd';
import { Cascader } from 'antd';

interface Option {
  code: string;
  name: string;
  items?: Option[];
}

const options: Option[] = [
  {
    code: 'zhejiang',
    name: 'Zhejiang',
    items: [
      {
        code: 'hangzhou',
        name: 'Hangzhou',
        items: [
          {
            code: 'xihu',
            name: 'West Lake',
          },
        ],
      },
    ],
```

```
  },
  {
    code: 'jiangsu',
    name: 'Jiangsu',
    items: [
      {
        code: 'nanjing',
        name: 'Nanjing',
        items: [
          {
            code: 'zhonghuamen',
            name: 'Zhong Hua Men',
          },
        ],
      },
    ],
  },
];

const onChange: CascaderProps<Option>['onChange'] = (value) => {
  console.log(value);
};

const App: React.FC = () => (
  <Cascader
    fieldNames={{ label: 'name', value: 'code', children: 'items' }}
    options={options}
    onChange={onChange}
    placeholder="Please select"
  />
);

export default App;
```

前后缀

v5.22.0

```
import React from 'react';
import { SmileOutlined } from '@ant-design/icons';
import type { CascaderProps } from 'antd';
import { Cascader } from 'antd';

interface Option {
  value: string;
  label: string;
```

```tsx
  children?: Option[];
}

const options: Option[] = [
  {
    value: 'zhejiang',
    label: 'Zhejiang',
    children: [
      {
        value: 'hangzhou',
        label: 'Hangzhou',
        children: [
          {
            value: 'xihu',
            label: 'West Lake',
          },
        ],
      },
    ],
  },
  {
    value: 'jiangsu',
    label: 'Jiangsu',
    children: [
      {
        value: 'nanjing',
        label: 'Nanjing',
        children: [
          {
            value: 'zhonghuamen',
            label: 'Zhong Hua Men',
          },
        ],
      },
    ],
  },
];

const onChange: CascaderProps<Option>['onChange'] = (value) => {
  console.log(value);
};

const App: React.FC = () => (
  <>
    <Cascader
      suffixIcon={<SmileOutlined />}
```

```
      options={options}
      onChange={onChange}
      placeholder="Please select"
    />
    <br />
    <br />
    <Cascader suffixIcon="ab" options={options} onChange={onChange}
placeholder="Please select" />
    <br />
    <br />
    <Cascader
      expandIcon={<SmileOutlined />}
      options={options}
      onChange={onChange}
      placeholder="Please select"
    />
    <br />
    <br />
    <Cascader expandIcon="ab" options={options} onChange={onChange}
placeholder="Please select" />
    <br />
    <br />
    <Cascader
      prefix={<SmileOutlined />}
      options={options}
      onChange={onChange}
      placeholder="Please select"
    />
  </>
);

export default App;
```

扩展菜单

```
import React from 'react';
import { Cascader, Divider } from 'antd';

interface Option {
  value: string;
  label: string;
  children?: Option[];
}

const options: Option[] = [
  {
```

```
    value: 'zhejiang',
    label: 'Zhejiang',
    children: [
      {
        value: 'hangzhou',
        label: 'Hangzhou',
        children: [
          {
            value: 'xihu',
            label: 'West Lake',
          },
        ],
      },
    ],
  },
  {
    value: 'jiangsu',
    label: 'Jiangsu',
    children: [
      {
        value: 'nanjing',
        label: 'Nanjing',
        children: [
          {
            value: 'zhonghuamen',
            label: 'Zhong Hua Men',
          },
        ],
      },
    ],
  },
];

const dropdownRender = (menus: React.ReactNode) => (
  <div>
    {menus}
    <Divider style={{ margin: 0 }} />
    <div style={{ padding: 8 }}>The footer is not very short.</div>
  </div>
);

const App: React.FC = () => (
  <Cascader options={options} dropdownRender={dropdownRender}
placeholder="Please select" />
);
```

```
export default App;
```

弹出位置

```
import React, { useState } from 'react';
import type { RadioChangeEvent } from 'antd';
import { Cascader, Radio } from 'antd';

interface Option {
  value: string;
  label: string;
  children?: Option[];
}

const options: Option[] = [
  {
    value: 'zhejiang',
    label: 'Zhejiang',
    children: [
      {
        value: 'hangzhou',
        label: 'Hangzhou',
        children: [
          {
            value: 'xihu',
            label: 'West Lake',
          },
        ],
      },
    ],
  },
  {
    value: 'jiangsu',
    label: 'Jiangsu',
    children: [
      {
        value: 'nanjing',
        label: 'Nanjing',
        children: [
          {
            value: 'zhonghuamen',
            label: 'Zhong Hua Men',
          },
        ],
      },
```

```
    ],
  },
];

const App: React.FC = () => {
  const [placement, SetPlacement] = useState<'bottomLeft' | 'bottomRight' |
'topLeft' | 'topRight'>(
    'topLeft',
  );

  const placementChange = (e: RadioChangeEvent) => {
    SetPlacement(e.target.value);
  };

  return (
    <>
      <Radio.Group value={placement} onChange={placementChange}>
        <Radio.Button value="topLeft">topLeft</Radio.Button>
        <Radio.Button value="topRight">topRight</Radio.Button>
        <Radio.Button value="bottomLeft">bottomLeft</Radio.Button>
        <Radio.Button value="bottomRight">bottomRight</Radio.Button>
      </Radio.Group>
      <br />
      <br />
      <Cascader options={options} placeholder="Please select" placement=
{placement} />
    </>
  );
};

export default App;
```

形态变体

v5.13.0

```
import React from 'react';
import { Cascader, Flex } from 'antd';

const App: React.FC = () => (
  <Flex vertical gap="middle">
    <Cascader placeholder="Please select" variant="borderless" />
    <Cascader placeholder="Please select" variant="filled" />
    <Cascader placeholder="Please select" variant="outlined" />
    <Cascader placeholder="Please select" variant="underlined" />
  </Flex>
```

```
);

export default App;
```

自定义状态

```
import React from 'react';
import { Cascader, Space } from 'antd';

const App: React.FC = () => (
  <Space direction="vertical">
    <Cascader status="error" placeholder="Error" />
    <Cascader status="warning" multiple placeholder="Warning multiple" />
  </Space>
);

export default App;
```

= 5.10.0">面板使用

```
import React, { useState } from 'react';
import type { CascaderProps } from 'antd';
import { Cascader, Flex, Switch } from 'antd';

interface Option {
  value: string | number;
  label: string;
  children?: Option[];
}

const options: Option[] = [
  {
    value: 'zhejiang',
    label: 'Zhejiang',
    children: [
      {
        value: 'hangzhou',
        label: 'Hangzhou',
        children: [
          {
            value: 'xihu',
            label: 'West Lake',
          },
        ],
      },
```

```
    ],
  },
  {
    value: 'jiangsu',
    label: 'Jiangsu',
    children: [
      {
        value: 'nanjing',
        label: 'Nanjing',
        children: [
          {
            value: 'zhonghuamen',
            label: 'Zhong Hua Men',
          },
        ],
      },
    ],
  },
];

const onChange: CascaderProps<Option>['onChange'] = (value) => {
  console.log(value);
};

const onMultipleChange: CascaderProps<Option, 'value', true>['onChange'] =
(value) => {
  console.log(value);
};

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

  return (
    <Flex vertical gap="small" align="flex-start">
      <Switch
        checked={disabled}
        checkedChildren="Enabled"
        unCheckedChildren="Disabled"
        onChange={setDisabled}
        aria-label="disabled switch"
      />
      <Cascader.Panel options={options} onChange={onChange} disabled=
{disabled} />
      <Cascader.Panel multiple options={options} onChange=
{onMultipleChange} disabled={disabled} />
      <Cascader.Panel />
```

```
    </Flex>
  );
};

export default App;
```

_InternalPanelDoNotUseOrYouWillBeFired

Debug

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

const { _InternalPanelDoNotUseOrYouWillBeFired: InternalCascader } =
Cascader;

interface Option {
  value: string | number;
  label: string;
  children?: Option[];
}

const options: Option[] = [
  {
    value: 'zhejiang',
    label: 'Zhejiang',
    children: [
      {
        value: 'hangzhou',
        label: 'Hangzhou',
        children: [
          {
            value: 'xihu',
            label: 'West Lake',
          },
        ],
      },
    ],
  },
  {
    value: 'jiangsu',
    label: 'Jiangsu',
    children: [
      {
        value: 'nanjing',
        label: 'Nanjing',
```

```
        children: [
          {
            value: 'zhonghuamen',
            label: 'Zhong Hua Men',
          },
        ],
      },
    ],
  },
];

const App: React.FC = () => <InternalCascader options={options}
placeholder="Please select" />;

export default App;
```

Component Token

Debug

```
import React from 'react';
import type { CascaderProps } from 'antd';
import { Cascader, ConfigProvider } from 'antd';

interface Option {
  value: string;
  label: string;
  children?: Option[];
}

const options: Option[] = [
  {
    value: 'zhejiang',
    label: 'Zhejiang',
    children: [
      {
        value: 'hangzhou',
        label: 'Hangzhou',
        children: [
          {
            value: 'xihu',
            label: 'West Lake',
          },
        ],
      },
    ],
```

```
  },
  {
    value: 'jiangsu',
    label: 'Jiangsu',
    children: [
      {
        value: 'nanjing',
        label: 'Nanjing',
        children: [
          {
            value: 'zhonghuamen',
            label: 'Zhong Hua Men',
          },
        ],
      },
    ],
  },
];

const onChange: CascaderProps<Option>['onChange'] = (value) => {
  console.log(value);
};

const App: React.FC = () => (
  <ConfigProvider
    theme={{
      components: {
        Cascader: {
          optionSelectedColor: 'red',
        },
      },
    }}
  >
    <Cascader options={options} onChange={onChange} placeholder="Please
select" />
  </ConfigProvider>
);

export default App;
```

API

通用属性参考：通用属性

```
<Cascader options={options} onChange={onChange} />
```

参数	说明	类型	
allowClear	支持清除	boolean | { clearIcon?: ReactNode }	true
autoClearSearchValue	是否在选中项后清空搜索框，只在 `multiple` 为 `true` 时有效	boolean	true
autoFocus	自动获取焦点	boolean	false
changeOnSelect	单选时生效（multiple 下始终都可以选择），点选每级菜单选项值都会发生变化。	boolean	false
className	自定义类名	string	-
defaultOpen	是否默认展示浮层	boolean	-
defaultValue	默认的选中项	string[] | number[]	[]
disabled	禁用	boolean	false
displayRender	选择后展示的渲染函数	(label, selectedOptions) => ReactNode	label =
tagRender	自定义 tag 内容 render，仅在多选时生效	({ label: string, onClose: function, value: string }) => ReactNode	-
popupClassName	自定义浮层类名	string	-
dropdownRender	自定义下拉框内容	(menus: ReactNode) => ReactNode	-
expandIcon	自定义次级菜单展开图标	ReactNode	-
expandTrigger	次级菜单的展开方式，可选 'click' 和 'hover'	string	`click`
fieldNames	自定义 options 中 label value children 的字段	object	{ label `value` `child`
getPopupContainer	菜单渲染父节点。默认渲染到 body 上，如果你遇到菜单滚动定位问题，试试修改为滚动的区域，并相对其定位。[示例]	function(triggerNode)	() =>
loadData	用于动态加载选项，无法与 `showSearch` 一起使用	(selectedOptions) => void	-

maxTagCount	最多显示多少个 tag，响应式模式会对性能产生损耗	number | `responsive`	-
maxTagPlaceholder	隐藏 tag 时显示的内容	ReactNode | function(omittedValues)	-
maxTagTextLength	最大显示的 tag 文本长度	number	-
notFoundContent	当下拉列表为空时显示的内容	ReactNode	`Not F`
open	控制浮层显隐	boolean	-
options	可选项数据源	Option[]	-
placeholder	输入框占位文本	string	-
placement	浮层预设位置	`bottomLeft` `bottomRight` `topLeft` `topRight`	`botto`
prefix	自定义前缀	ReactNode	-
showSearch	在选择框中显示搜索框	boolean | Object	false
size	输入框大小	`large` | `middle` | `small`	-
status	设置校验状态	'error' | 'warning'	-
style	自定义样式	CSSProperties	-
suffixIcon	自定义的选择框后缀图标	ReactNode	-
value	指定选中项	string[] | number[]	-
variant	形态变体	`outlined` | `borderless` | `filled` | `underlined`	`outli`
onChange	选择完成后的回调	(value, selectedOptions) => void	-
onDropdownVisibleChange	显示/隐藏浮层的回调	(value) => void	-
multiple	支持多选节点	boolean	-
showCheckedStrategy	定义选中项回填的方式。`Cascader.SHOW_CHILD`: 只显示选中的子节点。`Cascader.SHOW_PARENT`: 只显示父节点（当父节点下所有子节点都选中时）。	`Cascader.SHOW_PARENT` | `Cascader.SHOW_CHILD`	`Casca`

removeIcon	自定义的多选框清除图标	ReactNode	-
searchValue	设置搜索的值，需要与 `showSearch` 配合使用	string	-
onSearch	监听搜索，返回输入的值	(search: string) => void	-
dropdownMenuColumnStyle	下拉菜单列的样式	CSSProperties	-
optionRender	自定义渲染下拉选项	(option: Option) => React.ReactNode	-

showSearch

`showSearch` 为对象时，其中的字段：

参数	说明	类型	默认值	版本
filter	接收 `inputValue` `path` 两个参数，当 `path` 符合筛选条件时，应返回 true，反之则返回 false	function(inputValue, path): boolean	-	
limit	搜索结果展示数量	number | false	50	
matchInputWidth	搜索结果列表是否与输入框同宽（效果）	boolean	true	
render	用于渲染 filter 后的选项	function(inputValue, path): ReactNode	-	
sort	用于排序 filter 后的选项	function(a, b, inputValue)	-	

Option

```tsx
interface Option {
  value: string | number;
  label?: React.ReactNode;
  disabled?: boolean;
  children?: Option[];
  // 标记是否为叶子节点，设置了 `loadData` 时有效
  // 设为 `false` 时会强制标记为父节点，即使当前节点没有 children，也会显示展开图标
  isLeaf?: boolean;
}
```

方法

名称	描述	版本
blur()	移除焦点	
focus()	获取焦点	

> *注意，如果需要获得中国省市区数据，可以参考 china-division。*

主题变量（Design Token）