何时使用 {#when-to-use}

- 弹出一个下拉菜单给用户选择操作，用于代替原生的选择器，或者需要一个更优雅的多选器时。
- 当选项少时（少于 5 项），建议直接将选项平铺，使用 Radio 是更好的选择。
- 如果你在寻找一个可输可选的输入框，那你可能需要 AutoComplete。

用法升级 5.11.0+

:::warning{title="升级提示"} 在 5.11.0 版本后，我们提供了 `<Select options={[...]} />` 的简写方式，有更好的性能和更方便的数据组织方式，开发者不再需要自行拼接 JSX。 同时我们废弃了原先的写法，你还是可以在 5.x 继续使用，但会在控制台看到警告，并会在 6.0 后移除。 :::

```
// >=5.11.0 可用，推荐的写法 ✅
return <Select options={[{ value: 'sample', label: <span>sample</span> }]}
/>;

// 5.x 都可用，>=5.11.0 时不推荐 🙅🏻‍♀️
return (
  <Select onChange={onChange}>
    <Select.Option value="sample">Sample</Select.Option>
  </Select>
);
```

代码演示

基本使用

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

const handleChange = (value: string) => {
  console.log(`selected ${value}`);
};

const App: React.FC = () => (
  <Space wrap>
    <Select
      defaultValue="lucy"
      style={{ width: 120 }}
      onChange={handleChange}
      options={[
        { value: 'jack', label: 'Jack' },
        { value: 'lucy', label: 'Lucy' },
        { value: 'Yiminghe', label: 'yiminghe' },
        { value: 'disabled', label: 'Disabled', disabled: true },
      ]}
    />
    <Select
      defaultValue="lucy"
      style={{ width: 120 }}
      disabled
      options={[{ value: 'lucy', label: 'Lucy' }]}
    />
    <Select
      defaultValue="lucy"
      style={{ width: 120 }}
      loading
      options={[{ value: 'lucy', label: 'Lucy' }]}
    />
    <Select
      defaultValue="lucy"
      style={{ width: 120 }}
      allowClear
      options={[{ value: 'lucy', label: 'Lucy' }]}
      placeholder="select it"
    />
  </Space>
);

export default App;
```

带搜索框

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

const onChange = (value: string) => {
  console.log(`selected ${value}`);
};

const onSearch = (value: string) => {
  console.log('search:', value);
};

const App: React.FC = () => (
  <Select
    showSearch
    placeholder="Select a person"
    optionFilterProp="label"
    onChange={onChange}
    onSearch={onSearch}
    options={[
      {
        value: 'jack',
        label: 'Jack',
      },
      {
        value: 'lucy',
        label: 'Lucy',
      },
      {
        value: 'tom',
        label: 'Tom',
      },
    ]}
  />
);

export default App;
```

自定义搜索

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

const App: React.FC = () => (
  <Select
    showSearch
    placeholder="Select a person"
    filterOption={(input, option) =>
      (option?.label ?? '').toLowerCase().includes(input.toLowerCase())
    }
    options={[
      { value: '1', label: 'Jack' },
      { value: '2', label: 'Lucy' },
      { value: '3', label: 'Tom' },
    ]}
  />
);

export default App;
```

多选

```
import React from 'react';
import { Select, Space } from 'antd';
import type { SelectProps } from 'antd';
```

```
const options: SelectProps['options'] = [];

for (let i = 10; i < 36; i++) {
  options.push({
    label: i.toString(36) + i,
    value: i.toString(36) + i,
  });
}

const handleChange = (value: string[]) => {
  console.log(`selected ${value}`);
};

const App: React.FC = () => (
  <Space style={{ width: '100%' }} direction="vertical">
    <Select
      mode="multiple"
      allowClear
      style={{ width: '100%' }}
      placeholder="Please select"
      defaultValue={['a10', 'c12']}
      onChange={handleChange}
      options={options}
    />
    <Select
      mode="multiple"
      disabled
      style={{ width: '100%' }}
      placeholder="Please select"
      defaultValue={['a10', 'c12']}
      onChange={handleChange}
      options={options}
    />
  </Space>
);

export default App;
```

三种大小

```
import React, { useState } from 'react';
import { Radio, Select, Space } from 'antd';
import type { ConfigProviderProps, RadioChangeEvent, SelectProps } from
'antd';

type SizeType = ConfigProviderProps['componentSize'];
```

```
const options: SelectProps['options'] = [];

for (let i = 10; i < 36; i++) {
  options.push({
    value: i.toString(36) + i,
    label: i.toString(36) + i,
  });
}

const handleChange = (value: string | string[]) => {
  console.log(`Selected: ${value}`);
};

const App: React.FC = () => {
  const [size, setSize] = useState<SizeType>('middle');

  const handleSizeChange = (e: RadioChangeEvent) => {
    setSize(e.target.value);
  };

  return (
    <>
      <Radio.Group value={size} onChange={handleSizeChange}>
        <Radio.Button value="large">Large</Radio.Button>
        <Radio.Button value="middle">Default</Radio.Button>
        <Radio.Button value="small">Small</Radio.Button>
      </Radio.Group>
      <br />
      <br />
      <Space direction="vertical" style={{ width: '100%' }}>
        <Select
          size={size}
          defaultValue="a1"
          onChange={handleChange}
          style={{ width: 200 }}
          options={options}
        />
        <Select
          mode="multiple"
          size={size}
          placeholder="Please select"
          defaultValue={['a10', 'c12']}
          onChange={handleChange}
          style={{ width: '100%' }}
          options={options}
        />
        <Select
          mode="tags"
          size={size}
          placeholder="Please select"
          defaultValue={['a10', 'c12']}
          onChange={handleChange}
          style={{ width: '100%' }}
          options={options}
        />
      </Space>
    </>
  );
};

export default App;
```

自定义下拉选项

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

const handleChange = (value: string[]) => {
  console.log(`selected ${value}`);
};

const options = [
  {
    label: 'China',
    value: 'china',
    emoji: '🇨🇳',
    desc: 'China (中国)',
  },
  {
    label: 'USA',
    value: 'usa',
    emoji: '🇺🇸',
    desc: 'USA (美国)',
  },
  {
    label: 'Japan',
    value: 'japan',
    emoji: '🇯🇵',
    desc: 'Japan (日本)',
  },
  {
    label: 'Korea',
    value: 'korea',
    emoji: '🇰🇷',
    desc: 'Korea (韩国)',
  },
];

const App: React.FC = () => (
  <Select
    mode="multiple"
    style={{ width: '100%' }}
    placeholder="select one country"
    defaultValue={['china']}
    onChange={handleChange}
    options={options}
    optionRender={(option) => (
      <Space>
        <span role="img" aria-label={option.data.label}>
          {option.data.emoji}
        </span>
        {option.data.desc}
      </Space>
    )}
  />
);

export default App;
```

带排序的搜索

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

const App: React.FC = () => (
  <Select
    showSearch
    style={{ width: 200 }}
    placeholder="Search to Select"
    optionFilterProp="label"
    filterSort={(optionA, optionB) =>
      (optionA?.label ?? '').toLowerCase().localeCompare((optionB?.label ??
'').toLowerCase())
    }
    options={[
      {
        value: '1',
        label: 'Not Identified',
      },
      {
        value: '2',
        label: 'Closed',
      },
      {
        value: '3',
        label: 'Communicated',
      },
      {
        value: '4',
        label: 'Identified',
      },
      {
        value: '5',
        label: 'Resolved',
      },
      {
        value: '6',
        label: 'Cancelled',
      },
    ]}
  />
);

export default App;
```

标签

```
import React from 'react';
import { Select } from 'antd';
import type { SelectProps } from 'antd';

const options: SelectProps['options'] = [];

for (let i = 10; i < 36; i++) {
  options.push({
    value: i.toString(36) + i,
    label: i.toString(36) + i,
  });
}

const handleChange = (value: string[]) => {
  console.log(`selected ${value}`);
};
```

```
const App: React.FC = () => (
  <Select
    mode="tags"
    style={{ width: '100%' }}
    placeholder="Tags Mode"
    onChange={handleChange}
    options={options}
  />
);

export default App;
```

分组

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

const handleChange = (value: string) => {
  console.log(`selected ${value}`);
};

const App: React.FC = () => (
  <Select
    defaultValue="lucy"
    style={{ width: 200 }}
    onChange={handleChange}
    options={[
      {
        label: <span>manager</span>,
        title: 'manager',
        options: [
          { label: <span>Jack</span>, value: 'Jack' },
          { label: <span>Lucy</span>, value: 'Lucy' },
        ],
      },
      {
        label: <span>engineer</span>,
        title: 'engineer',
        options: [
          { label: <span>Chloe</span>, value: 'Chloe' },
          { label: <span>Lucas</span>, value: 'Lucas' },
        ],
      },
    ]}
  />
);

export default App;
```

联动

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

const cityData = {
  Zhejiang: ['Hangzhou', 'Ningbo', 'Wenzhou'],
  Jiangsu: ['Nanjing', 'Suzhou', 'Zhenjiang'],
};

type CityName = keyof typeof cityData;

const provinceData: CityName[] = ['Zhejiang', 'Jiangsu'];

const App: React.FC = () => {
  const [cities, setCities] = useState(cityData[provinceData[0] as
CityName]);
  const [secondCity, setSecondCity] = useState(cityData[provinceData[0]][0]
as CityName);

  const handleProvinceChange = (value: CityName) => {
    setCities(cityData[value]);
    setSecondCity(cityData[value][0] as CityName);
  };

  const onSecondCityChange = (value: CityName) => {
    setSecondCity(value);
  };

  return (
    <Space wrap>
      <Select
        defaultValue={provinceData[0]}
        style={{ width: 120 }}
        onChange={handleProvinceChange}
        options={provinceData.map((province) => ({ label: province, value:
province }))}
      />
      <Select
        style={{ width: 120 }}
        value={secondCity}
        onChange={onSecondCityChange}
        options={cities.map((city) => ({ label: city, value: city }))}
      />
    </Space>
  );
};

export default App;
```

搜索框

```
/* eslint-disable compat/compat */
import React, { useState } from 'react';
import { Select } from 'antd';
import type { SelectProps } from 'antd';
import type { AnyObject } from 'antd/es/_util/type';

let timeout: ReturnType<typeof setTimeout> | null;
let currentValue: string;

const toURLSearchParams = <T extends AnyObject>(record: T) => {
  const params = new URLSearchParams();
  for (const [key, value] of Object.entries(record)) {
    params.append(key, value);
  }
  return params;
};

const fetchData = (value: string, callback: (data: { value: string; text:
string }[]) => void) => {
  if (timeout) {
    clearTimeout(timeout);
    timeout = null;
  }
  currentValue = value;

  const params = toURLSearchParams({ code: 'utf-8', q: value });

  const fake = () => {
    fetch(`https://suggest.taobao.com/sug?${params.toString()}`)
      .then((response) => response.json())
      .then(({ result }) => {
        if (currentValue === value) {
          const data = result.map((item: any) => ({ value: item[0], text:
item[0] }));
          callback(data);
        }
      });
  };
  if (value) {
    timeout = setTimeout(fake, 300);
  } else {
    callback([]);
  }
};

const SearchInput: React.FC<{ placeholder: string; style:
React.CSSProperties }> = (props) => {
  const [data, setData] = useState<SelectProps['options']>([]);
  const [value, setValue] = useState<string>();

  const handleSearch = (newValue: string) => {
    fetchData(newValue, setData);
  };

  const handleChange = (newValue: string) => {
    setValue(newValue);
  };

  return (
    <Select
      showSearch
      value={value}
      placeholder={props.placeholder}
      style={props.style}
      defaultActiveFirstOption={false}
      suffixIcon={null}
      filterOption={false}
      onSearch={handleSearch}
      onChange={handleChange}
      notFoundContent={null}
      options={(data || []).map((d) => ({
        value: d.value,
        label: d.text,
      }))}
    />
  );
};

const App: React.FC = () => <SearchInput placeholder="input search text"
style={{ width: 200 }} />;

export default App;
```

获得选项的文本

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

const handleChange = (value: { value: string; label: React.ReactNode }) =>
{
  console.log(value); // { value: "lucy", key: "lucy", label: "Lucy (101)"
}
};

const App: React.FC = () => (
  <Select
    labelInValue
    defaultValue={{ value: 'lucy', label: 'Lucy (101)' }}
    style={{ width: 120 }}
    onChange={handleChange}
    options={[
      {
        value: 'jack',
        label: 'Jack (100)',
      },
      {
        value: 'lucy',
        label: 'Lucy (101)',
      },
    ]}
  />
);

export default App;
```

自动分词

```
import React from 'react';
import { Select } from 'antd';
import type { SelectProps } from 'antd';

const options: SelectProps['options'] = [];

for (let i = 10; i < 36; i++) {
  options.push({
    value: i.toString(36) + i,
    label: i.toString(36) + i,
  });
}

const handleChange = (value: string[]) => {
  console.log(`selected ${value}`);
};

const App: React.FC = () => (
  <Select
    mode="tags"
    style={{ width: '100%' }}
    onChange={handleChange}
    tokenSeparators={[',']}
    options={options}
  />
);

export default App;
```

搜索用户

```
import React, { useMemo, useRef, useState } from 'react';
import { Select, Spin } from 'antd';
import type { SelectProps } from 'antd';
import debounce from 'lodash/debounce';

export interface DebounceSelectProps<ValueType = any>
  extends Omit<SelectProps<ValueType | ValueType[]>, 'options' |
'children'> {
  fetchOptions: (search: string) => Promise<ValueType[]>;
  debounceTimeout?: number;
}

function DebounceSelect<
  ValueType extends { key?: string; label: React.ReactNode; value: string |
number } = any,
>({ fetchOptions, debounceTimeout = 800, ...props }:
DebounceSelectProps<ValueType>) {
  const [fetching, setFetching] = useState(false);
  const [options, setOptions] = useState<ValueType[]>([]);
  const fetchRef = useRef(0);

  const debounceFetcher = useMemo(() => {
    const loadOptions = (value: string) => {
      fetchRef.current += 1;
      const fetchId = fetchRef.current;
      setOptions([]);
      setFetching(true);

      fetchOptions(value).then((newOptions) => {
        if (fetchId !== fetchRef.current) {
          // for fetch callback order
          return;
        }

        setOptions(newOptions);
        setFetching(false);
      });
    };

    return debounce(loadOptions, debounceTimeout);
  }, [fetchOptions, debounceTimeout]);

  return (
    <Select
      labelInValue
      filterOption={false}
      onSearch={debounceFetcher}
      notFoundContent={fetching ? <Spin size="small" /> : null}
      {...props}
      options={options}
    />
  );
}

// Usage of DebounceSelect
interface UserValue {
  label: string;
  value: string;
}

async function fetchUserList(username: string): Promise<UserValue[]> {
  console.log('fetching user', username);

  return fetch('https://randomuser.me/api/?results=5')
    .then((response) => response.json())
    .then((body) =>
      body.results.map(
        (user: { name: { first: string; last: string }; login: { username:
string } }) => ({
          label: `${user.name.first} ${user.name.last}`,
          value: user.login.username,
        }),
      ),
    );
}

const App: React.FC = () => {
  const [value, setValue] = useState<UserValue[]>([]);

  return (
    <DebounceSelect
      mode="multiple"
      value={value}
      placeholder="Select users"
      fetchOptions={fetchUserList}
      style={{ width: '100%' }}
      onChange={(newValue) => {
        if (Array.isArray(newValue)) {
          setValue(newValue);
        }
      }}
    />
  );
};

export default App;
```

前后缀

v5.22.0

```
import React from 'react';
import { MehOutlined, SmileOutlined } from '@ant-design/icons';
import { Select, Space } from 'antd';

const smileIcon = <SmileOutlined />;
const mehIcon = <MehOutlined />;

const handleChange = (value: string | string[]) => {
  console.log(`selected ${value}`);
};

const App: React.FC = () => (
  <Space wrap>
    <Select
      prefix="User"
      defaultValue="lucy"
      style={{ width: 200 }}
      onChange={handleChange}
      options={[
        { value: 'jack', label: 'Jack' },
        { value: 'lucy', label: 'Lucy' },
        { value: 'Yiminghe', label: 'yiminghe' },
        { value: 'disabled', label: 'Disabled', disabled: true },
      ]}
    />
    <Select
      suffixIcon={smileIcon}
      defaultValue="lucy"
      style={{ width: 120 }}
      onChange={handleChange}
      options={[
        { value: 'jack', label: 'Jack' },
        { value: 'lucy', label: 'Lucy' },
        { value: 'Yiminghe', label: 'yiminghe' },
        { value: 'disabled', label: 'Disabled', disabled: true },
      ]}
    />
    <Select
      suffixIcon={mehIcon}
      defaultValue="lucy"
      style={{ width: 120 }}
      disabled
      options={[{ value: 'lucy', label: 'Lucy' }]}
    />
    <br />
    <Select
      prefix="User"
      defaultValue={['lucy']}
      mode="multiple"
      style={{ width: 200 }}
      onChange={handleChange}
      options={[
        { value: 'jack', label: 'Jack' },
        { value: 'lucy', label: 'Lucy' },
        { value: 'Yiminghe', label: 'yiminghe' },
        { value: 'disabled', label: 'Disabled', disabled: true },
      ]}
    />
    <Select
      suffixIcon={smileIcon}
      defaultValue={['lucy']}
      mode="multiple"
      style={{ width: 120 }}
      onChange={handleChange}
      options={[
        { value: 'jack', label: 'Jack' },
        { value: 'lucy', label: 'Lucy' },
        { value: 'Yiminghe', label: 'yiminghe' },
        { value: 'disabled', label: 'Disabled', disabled: true },
      ]}
    />
    <Select
      suffixIcon={mehIcon}
      defaultValue={['lucy']}
      mode="multiple"
      style={{ width: 120 }}
      disabled
      options={[{ value: 'lucy', label: 'Lucy' }]}
    />
  </Space>
);

export default App;
```

扩展菜单

```
import React, { useRef, useState } from 'react';
import { PlusOutlined } from '@ant-design/icons';
import { Button, Divider, Input, Select, Space } from 'antd';
import type { InputRef } from 'antd';

let index = 0;

const App: React.FC = () => {
  const [items, setItems] = useState(['jack', 'lucy']);
  const [name, setName] = useState('');
  const inputRef = useRef<InputRef>(null);

  const onNameChange = (event: React.ChangeEvent<HTMLInputElement>) => {
    setName(event.target.value);
  };

  const addItem = (e: React.MouseEvent<HTMLButtonElement |
HTMLAnchorElement>) => {
    e.preventDefault();
    setItems([...items, name || `New item ${index++}`]);
    setName('');
    setTimeout(() => {
      inputRef.current?.focus();
    }, 0);
  };

  return (
    <Select
      style={{ width: 300 }}
      placeholder="custom dropdown render"
      dropdownRender={(menu) => (
        <>
          {menu}
          <Divider style={{ margin: '8px 0' }} />
          <Space style={{ padding: '0 8px 4px' }}>
            <Input
              placeholder="Please enter item"
              ref={inputRef}
              value={name}
              onChange={onNameChange}
              onKeyDown={(e) => e.stopPropagation()}
            />
            <Button type="text" icon={<PlusOutlined />} onClick={addItem}>
              Add item
            </Button>
          </Space>
        </>
      )}
      options={items.map((item) => ({ label: item, value: item }))}
    />
  );
};

export default App;
```

隐藏已选择选项

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

const OPTIONS = ['Apples', 'Nails', 'Bananas', 'Helicopters'];

const App: React.FC = () => {
  const [selectedItems, setSelectedItems] = useState<string[]>([]);

  const filteredOptions = OPTIONS.filter((o) =>
!selectedItems.includes(o));

  return (
    <Select
      mode="multiple"
      placeholder="Inserted are removed"
      value={selectedItems}
      onChange={setSelectedItems}
      style={{ width: '100%' }}
      options={filteredOptions.map((item) => ({
        value: item,
        label: item,
      }))}
    />
  );
};

export default App;
```

形态变体

v5.13.0

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

const App: React.FC = () => (
  <Flex gap={12} vertical>
    <Flex gap={8}>
      <Select
        placeholder="Outlined"
        style={{ flex: 1 }}
        options={[
          { value: 'jack', label: 'Jack' },
          { value: 'lucy', label: 'Lucy' },
          { value: 'Yiminghe', label: 'yiminghe' },
        ]}
      />
      <Select
        mode="multiple"
        defaultValue={['lucy']}
        placeholder="Outlined"
        style={{ flex: 1 }}
        options={[
          { value: 'jack', label: 'Jack' },
          { value: 'lucy', label: 'Lucy' },
          { value: 'Yiminghe', label: 'yiminghe' },
        ]}
      />
    </Flex>
    <Flex gap={8}>
      <Select
        placeholder="Filled"
        variant="filled"
        style={{ flex: 1 }}
        options={[
          { value: 'jack', label: 'Jack' },
          { value: 'lucy', label: 'Lucy' },
          { value: 'Yiminghe', label: 'yiminghe' },
        ]}
      />
      <Select
        mode="multiple"
        defaultValue={['lucy']}
        placeholder="Filled"
        variant="filled"
        style={{ flex: 1 }}
        options={[
          { value: 'jack', label: 'Jack' },
          { value: 'lucy', label: 'Lucy' },
          { value: 'Yiminghe', label: 'yiminghe' },
        ]}
      />
    </Flex>
    <Flex gap={8}>
      <Select
        placeholder="Borderless"
        variant="borderless"
        style={{ flex: 1 }}
        options={[
          { value: 'jack', label: 'Jack' },
          { value: 'lucy', label: 'Lucy' },
          { value: 'Yiminghe', label: 'yiminghe' },
        ]}
      />
      <Select
        mode="multiple"
        defaultValue={['lucy']}
        placeholder="Borderless"
        variant="borderless"
        style={{ flex: 1 }}
        options={[
          { value: 'jack', label: 'Jack' },
          { value: 'lucy', label: 'Lucy' },
          { value: 'Yiminghe', label: 'yiminghe' },
        ]}
      />
    </Flex>
    <Flex gap={8}>
      <Select
        placeholder="Underlined"
        variant="underlined"
        style={{ flex: 1 }}
        options={[
          { value: 'jack', label: 'Jack' },
          { value: 'lucy', label: 'Lucy' },
          { value: 'Yiminghe', label: 'yiminghe' },
        ]}
      />
      <Select
        mode="multiple"
        defaultValue={['lucy']}
        placeholder="Underlined"
        variant="underlined"
        style={{ flex: 1 }}
        options={[
          { value: 'jack', label: 'Jack' },
          { value: 'lucy', label: 'Lucy' },
          { value: 'Yiminghe', label: 'yiminghe' },
        ]}
      />
    </Flex>
  </Flex>
);

export default App;
```

Filled debug

Debug

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

const App: React.FC = () => (
  <Flex gap={12} vertical>
    <Flex gap={8}>
      <Select
        value="lucy"
        disabled
        variant="filled"
        style={{ flex: 1 }}
        options={[
          { value: 'jack', label: 'Jack' },
          { value: 'lucy', label: 'Lucy' },
          { value: 'Yiminghe', label: 'yiminghe' },
        ]}
      />
      <Select
        value="lucy"
        disabled
        mode="multiple"
        variant="filled"
        placeholder="Outlined"
        style={{ flex: 1 }}
        options={[
          { value: 'jack', label: 'Jack' },
          { value: 'lucy', label: 'Lucy' },
          { value: 'Yiminghe', label: 'yiminghe' },
        ]}
      />
    </Flex>
    <Flex gap={8}>
      <Select
        value="lucy"
        status="error"
        variant="filled"
        style={{ flex: 1 }}
        options={[
          { value: 'jack', label: 'Jack' },
          { value: 'lucy', label: 'Lucy' },
          { value: 'Yiminghe', label: 'yiminghe' },
        ]}
      />
      <Select
        value="lucy"
        status="error"
        mode="multiple"
        variant="filled"
        placeholder="Outlined"
        style={{ flex: 1 }}
        options={[
          { value: 'jack', label: 'Jack' },
          { value: 'lucy', label: 'Lucy' },
          { value: 'Yiminghe', label: 'yiminghe' },
        ]}
      />
    </Flex>
    <Flex gap={8}>
      <Select
        disabled
        value="lucy"
        status="error"
        variant="filled"
        style={{ flex: 1 }}
        options={[
          { value: 'jack', label: 'Jack' },
          { value: 'lucy', label: 'Lucy' },
          { value: 'Yiminghe', label: 'yiminghe' },
        ]}
      />
      <Select
        disabled
        value="lucy"
        status="error"
        mode="multiple"
        variant="filled"
        placeholder="Outlined"
        style={{ flex: 1 }}
        options={[
          { value: 'jack', label: 'Jack' },
          { value: 'lucy', label: 'Lucy' },
          { value: 'Yiminghe', label: 'yiminghe' },
        ]}
      />
    </Flex>
  </Flex>
);

export default App;
```

自定义选择标签

```
import React from 'react';
import { Select, Tag } from 'antd';
import type { SelectProps } from 'antd';

type TagRender = SelectProps['tagRender'];

const options: SelectProps['options'] = [
  { value: 'gold' },
  { value: 'lime' },
  { value: 'green' },
  { value: 'cyan' },
];

const tagRender: TagRender = (props) => {
  const { label, value, closable, onClose } = props;
  const onPreventMouseDown = (event: React.MouseEvent<HTMLSpanElement>) =>
{
    event.preventDefault();
    event.stopPropagation();
  };
  return (
    <Tag
      color={value}
      onMouseDown={onPreventMouseDown}
      closable={closable}
      onClose={onClose}
      style={{ marginInlineEnd: 4 }}
    >
      {label}
    </Tag>
  );
};

const App: React.FC = () => (
  <Select
    mode="multiple"
    tagRender={tagRender}
    defaultValue={['gold', 'cyan']}
    style={{ width: '100%' }}
    options={options}
  />
);

export default App;
```

自定义选中 label

```
import React from 'react';
import { Select } from 'antd';
import type { SelectProps } from 'antd';

type LabelRender = SelectProps['labelRender'];

const options = [
  { label: 'gold', value: 'gold' },
  { label: 'lime', value: 'lime' },
  { label: 'green', value: 'green' },
  { label: 'cyan', value: 'cyan' },
];

const labelRender: LabelRender = (props) => {
  const { label, value } = props;

  if (label) {
    return value;
  }
  return <span>No option match</span>;
};

const App: React.FC = () => (
  <Select labelRender={labelRender} defaultValue="1" style={{ width: '100%'
}} options={options} />
);

export default App;
```

响应式 maxTagCount

```
import React, { useState } from 'react';
import type { SelectProps } from 'antd';
import { Select, Space, Tooltip } from 'antd';

interface ItemProps {
  label: string;
  value: string;
}

const options: ItemProps[] = [];

for (let i = 10; i < 36; i++) {
  const value = i.toString(36) + i;
  options.push({
    label: `Long Label: ${value}`,
    value,
  });
}

const sharedProps: SelectProps = {
  mode: 'multiple',
  style: { width: '100%' },
  options,
  placeholder: 'Select Item...',
  maxTagCount: 'responsive',
};

const App: React.FC = () => {
  const [value, setValue] = useState(['a10', 'c12', 'h17', 'j19', 'k20']);

  const selectProps: SelectProps = {
    value,
    onChange: setValue,
  };

  return (
    <Space direction="vertical" style={{ width: '100%' }}>
      <Select {...sharedProps} {...selectProps} />
      <Select {...sharedProps} disabled />
      <Select
        {...sharedProps}
        {...selectProps}
        maxTagPlaceholder={(omittedValues) => (
          <Tooltip
            styles={{ root: { pointerEvents: 'none' } }}
            title={omittedValues.map(({ label }) => label).join(', ')}
          >
            <span>Hover Me</span>
          </Tooltip>
        )}
      />
    </Space>
  );
};

export default App;
```

大数据

```
import React from 'react';
import type { SelectProps } from 'antd';
import { Select, Typography } from 'antd';

const { Title } = Typography;

const options: SelectProps['options'] = [];

for (let i = 0; i < 100000; i++) {
  const value = `${i.toString(36)}${i}`;
  options.push({
    label: value,
    value,
    disabled: i === 10,
  });
}

const handleChange = (value: string[]) => {
  console.log(`selected ${value}`);
};

const App: React.FC = () => (
  <>
    <Title level={4}>{options.length} Items</Title>
    <Select
      mode="multiple"
      style={{ width: '100%' }}
      placeholder="Please select"
      defaultValue={['a10', 'c12']}
      onChange={handleChange}
      options={options}
    />
  </>
);

export default App;
```

自定义状态

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

const App: React.FC = () => (
  <Space direction="vertical" style={{ width: '100%' }}>
    <Select status="error" style={{ width: '100%' }} />
    <Select status="warning" style={{ width: '100%' }} />
  </Space>
);

export default App;
```

弹出位置

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

```tsx
type SelectCommonPlacement = SelectProps['placement'];

const App: React.FC = () => {
  const [placement, SetPlacement] = useState<SelectCommonPlacement>
('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 />
      <Select
        defaultValue="HangZhou"
        style={{ width: 120 }}
        popupMatchSelectWidth={false}
        placement={placement}
        options={[
          {
            value: 'HangZhou',
            label: 'HangZhou #310000',
          },
          {
            value: 'NingBo',
            label: 'NingBo #315000',
          },
          {
            value: 'WenZhou',
            label: 'WenZhou #325000',
          },
        ]}
      />
    </>
  );
};

export default App;
```

动态高度

Debug

```
import React, { useState } from 'react';
import type { RadioChangeEvent, SelectProps } from 'antd';
import { Button, Radio, Select, Space, Switch } from 'antd';

type SelectCommonPlacement = SelectProps['placement'];

const randomOptions = (count?: number) => {
  const length = count ?? Math.floor(Math.random() * 5) + 1;

  // Random 1 ~ 5 options
  return Array.from({ length }).map((_, index) => ({
    value: index,
    label: `Option ${index}`,
  }));
};

const App: React.FC = () => {
  const [placement, SetPlacement] = useState<SelectCommonPlacement>
('topLeft');
  const [open, setOpen] = useState(false);
  const [options, setOptions] = useState(() => randomOptions(3));

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

  return (
    <div
      style={{
        height: '100%',
        minHeight: 500,
        display: 'flex',
        flexDirection: 'column',
        justifyContent: 'center',
        alignItems: 'center',
        position: 'relative',
      }}
    >
      <Space
        style={{
          position: 'absolute',
          top: 0,
          insetInlineStart: '50%',
          transform: 'translateX(-50%)',
        }}
      >
        <Radio.Group value={placement} onChange={placementChange}>
          <Radio.Button value="topLeft">TL</Radio.Button>
          <Radio.Button value="topRight">TR</Radio.Button>
          <Radio.Button value="bottomLeft">BL</Radio.Button>
          <Radio.Button value="bottomRight">BR</Radio.Button>
        </Radio.Group>

        <Switch checked={open} onChange={() => setOpen((o) => !o)} />
        <Button onClick={() => setOptions(randomOptions())}>Random</Button>
      </Space>
      <Select
        open={open}
        style={{ width: 120 }}
        placement={placement}
        options={options}
        popupMatchSelectWidth={200}
      />
    </div>
  );
};

export default App;
```

4.0 Debug

Debug

```
import React from 'react';
import { Button, Input, Select, Space } from 'antd';

const style: React.CSSProperties = {
  width: 500,
  position: 'relative',
  zIndex: 1,
  border: '1px solid red',
  backgroundColor: '#fff',
};

const handleChange = (value: string | string[]) => {
  console.log(`selected ${value}`);
};
```

```
const App: React.FC = () => (
  <Space style={style} wrap>
    <Input style={{ width: 100 }} value="222" />
    <Select
      style={{ width: 120 }}
      onChange={handleChange}
      showSearch
      placeholder="233"
      options={[
        { value: 'jack', label: 'Jack' },
        { value: 'lucy', label: 'Lucy' },
        { value: 'disabled', disabled: true, label: 'Disabled' },
        { value: 'Yiminghe', label: 'yiminghe' },
        { value: 'long', label: 'I am super super long!' },
      ]}
    />
    <Select
      mode="multiple"
      style={{ width: 120 }}
      defaultValue={['lucy']}
      onChange={handleChange}
      showSearch
      placeholder="233"
      options={[
        { value: 'jack', label: 'Jack' },
        { value: 'lucy', label: 'Lucy' },
        { value: 'disabled', disabled: true, label: 'Disabled' },
        { value: 'Yiminghe', label: 'yiminghe' },
        { value: 'long', label: 'I am super super long!' },
      ]}
    />
    <span className="debug-align">AntDesign</span>
    <Button>222</Button>
  </Space>
);

export default App;
```

_InternalPanelDoNotUseOrYouWillBeFired

Debug

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

const { _InternalPanelDoNotUseOrYouWillBeFired: InternalSelect } = Select;
const App: React.FC = () => {
  const [open, setOpen] = React.useState(true);

  return (
    <Space direction="vertical" style={{ display: 'flex' }}>
      <Switch checked={open} onChange={() => setOpen(!open)} />
      <InternalSelect
        defaultValue="lucy"
        style={{ width: 120 }}
        open={open}
        options={[
          { label: 'Jack', value: 'jack' },
          { label: 'Lucy', value: 'lucy' },
          { label: 'Disabled', value: 'disabled' },
          { label: 'Bamboo', value: 'bamboo' },
        ]}
        virtual={false}
      />
    </Space>
  );
};

export default App;
```

选项文本居中

Debug

```
import React from 'react';
import {
  AutoComplete,
  Cascader,
  Flex,
  Form,
  Input,
  Select,
  Space,
  TreeSelect,
  Typography,
} from 'antd';

const options = [
  { value: 'long', label: <Typography>long, long, long piece of
text</Typography> },
  { value: 'short', label: <Typography>short</Typography> },
  { value: 'normal', label: <div>normal</div> },
];

const App: React.FC = () => (
  <>
    <Space wrap>
      <Select
        defaultValue="long, long, long piece of text"
        style={{ width: 120 }}
        allowClear
        options={options}
      />

      <Select
        placeholder="Select a option"
        style={{ width: 120, height: 60 }}
        allowClear
        options={options}
      />

      <Select
        defaultValue="normal"
        placeholder="Select a option"
        style={{ width: 120 }}
        allowClear
        options={options}
      />

      <Select
        defaultValue={['normal']}
        mode="multiple"
        placeholder="Select a option"
        style={{ width: 120 }}
        allowClear
        options={options}
      />

      <Select
        mode="multiple"
        placeholder="Select a option"
        style={{ width: 120, height: 60 }}
        allowClear
        options={options}
      />

      <Cascader
        placeholder="Select a option"
        style={{ width: 120, height: 60 }}
        allowClear
        options={options}
      />

      <TreeSelect
        showSearch
        style={{ width: 120, height: 60 }}
        placeholder="Please select"
        allowClear
        popupMatchSelectWidth={false}
        treeDefaultExpandAll
        treeData={[
          {
            value: 'parent 1',
            title: 'parent 1',
            children: options,
          },
        ]}
      />
      <Select
        prefix="Hello World"
        mode="multiple"
        allowClear
        placeholder="Select"
        style={{ minWidth: 200, height: 200 }}
        defaultValue={['long']}
        options={options}
      />
      <Select
        mode="multiple"
        style={{ width: 200 }}
        placeholder="请选择"
        maxTagCount="responsive"
        prefix="城市"
        options={options}
      />
      <Select
        style={{ width: 200 }}
        placeholder="请选择"
        prefix="城市"
        options={options}
        showSearch
        allowClear
        status="error"
      />
      <Select
        style={{ width: 100 }}
        prefix="Hi"
        options={options}
        showSearch
        allowClear
        status="warning"
        variant="filled"
        defaultValue="Bamboo"
      />
      <Select
        style={{ width: 100 }}
        prefix="Hi"
        options={options}
        showSearch
        allowClear
        status="error"
        variant="borderless"
        defaultValue="Bamboo"
      />
      <Form style={{ width: 200 }} layout="vertical">
        <Form.Item
          label="Label"
          name="bamboo"
          initialValue="Bamboo"
          style={{
            boxShadow: '0 0 0 1px red',
          }}
        >
          <Select options={options} allowClear placeholder="bamboo" />
        </Form.Item>
        <Form.Item
          label="Label"
          name="bamboo"
          initialValue="Bamboo"
          style={{
            boxShadow: '0 0 0 1px red',
          }}
        >
          <AutoComplete options={options} allowClear placeholder="bamboo"
/>
        </Form.Item>
      </Form>
    </Space>
    {/* Test for align */}
    <Flex vertical style={{ width: 200 }}>
      {/* Single */}
      <Input prefix="Hi" placeholder="Input" allowClear />
      <Select prefix="Hi" placeholder="Single" options={options} allowClear
showSearch />
      <Select
        prefix="Hi"
        placeholder="Single"
        options={options}
        allowClear
        showSearch
        defaultValue="Bamboo"
      />
      {/* Multiple */}
      <Select placeholder="Multiple" options={options} allowClear
mode="multiple" />
      <Select prefix="Hi" placeholder="Multiple" options={options}
allowClear mode="multiple" />
      <Select
        prefix="Hi"
        placeholder="Multiple"
        options={options}
        allowClear
        mode="multiple"
        defaultValue={['Bamboo']}
      />
      <Select
        placeholder="Multiple"
        options={options}
        allowClear
        mode="multiple"
        defaultValue={['Bamboo']}
      />
    </Flex>
  </>
);

export default App;
```

翻转+偏移

Debug

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

```
const App: React.FC = () => (
  <Select
    style={{ width: 120, marginTop: '50vh' }}
    open
    options={Array.from({ length: 100 }).map((_, index) => ({
      value: index,
    }))}
  />
);

export default App;
```

组件 Token

Debug

```
import React from 'react';
import { ConfigProvider, Select, Space } from 'antd';
import type { SelectProps } from 'antd';

const options: SelectProps['options'] = [];

for (let i = 10; i < 36; i++) {
  options.push({
    label: i.toString(36) + i,
    value: i.toString(36) + i,
  });
}

const App: React.FC = () => (
  <Space direction="vertical">
    <ConfigProvider
      theme={{
        components: {
          Select: {
            multipleItemBorderColor: 'rgba(0,0,0,0.06)',
            multipleItemBorderColorDisabled: 'rgba(0,0,0,0.06)',
            optionSelectedColor: '#1677ff',
            hoverBorderColor: 'red',
            activeBorderColor: 'green',
            activeOutlineColor: 'pink',
          },
        },
      }}
    >
      <Space style={{ width: '100%' }} direction="vertical">
        <Select
          mode="multiple"
          allowClear
          style={{ width: '100%' }}
          placeholder="Please select"
          defaultValue={['a10', 'c12']}
          options={options}
        />
        <Select
          mode="multiple"
          disabled
          style={{ width: '100%' }}
          placeholder="Please select"
          defaultValue={['a10', 'c12']}
          options={options}
        />
      </Space>
    </ConfigProvider>
    <ConfigProvider
      theme={{
        token: {
          controlHeightSM: 28,
        },
      }}
    >
      <Space style={{ width: '100%' }} direction="vertical">
        <Select
          mode="multiple"
          allowClear
          size="small"
          style={{ width: '100%' }}
          placeholder="Please select"
          defaultValue={['a10', 'c12']}
          options={options}
        />
        <Select
          mode="multiple"
          allowClear
          style={{ width: '100%' }}
          placeholder="Please select"
          defaultValue={['a10', 'c12']}
          options={options}
        />
      </Space>
    </ConfigProvider>
```

```
    <ConfigProvider
      theme={{
        components: {
          Select: {
            paddingXXS: 0,
            controlHeight: 28,
          },
        },
      }}
    >
      <Space style={{ width: '100%' }} direction="vertical">
        <Select style={{ width: '100%' }} defaultValue="a10" options=
{options} />
        <Select
          mode="multiple"
          style={{ width: '100%' }}
          defaultValue={['a10', 'c12']}
          options={options}
        />
      </Space>
    </ConfigProvider>
  </Space>
);

export default App;
```

最大选中数量

v5.13.0

```
import React from 'react';
import { DownOutlined } from '@ant-design/icons';
import { Select } from 'antd';

const MAX_COUNT = 3;

const App: React.FC = () => {
  const [value, setValue] = React.useState<string[]>(['Ava Swift']);

  const suffix = (
    <>
      <span>
        {value.length} / {MAX_COUNT}
      </span>
      <DownOutlined />
    </>
  );

  return (
    <Select
      mode="multiple"
      maxCount={MAX_COUNT}
      value={value}
      style={{ width: '100%' }}
      onChange={setValue}
      suffixIcon={suffix}
      placeholder="Please select"
      options={[
        { value: 'Ava Swift', label: 'Ava Swift' },
        { value: 'Cole Reed', label: 'Cole Reed' },
        { value: 'Mia Blake', label: 'Mia Blake' },
        { value: 'Jake Stone', label: 'Jake Stone' },
        { value: 'Lily Lane', label: 'Lily Lane' },
        { value: 'Ryan Chase', label: 'Ryan Chase' },
        { value: 'Zoe Fox', label: 'Zoe Fox' },
        { value: 'Alex Grey', label: 'Alex Grey' },
        { value: 'Elle Blair', label: 'Elle Blair' },
      ]}
    />
  );
};

export default App;
```

API

通用属性参考：通用属性

Select props

参数	说明	类型
allowClear	自定义清除按钮	boolean | { clearIcon?: ReactNod
autoClearSearchValue	是否在选中项后清空搜索框，只在 `mode` 为 `multiple` 或 `tags` 时有效	boolean
autoFocus	默认获取焦点	boolean
defaultActiveFirstOption	是否默认高亮第一个选项	boolean
defaultOpen	是否默认展开下拉菜单	boolean

defaultValue	指定默认选中的条目	string | string[] | number | number[] | LabeledValue | LabeledValue[]
disabled	是否禁用	boolean
popupClassName	下拉菜单的 className 属性	string
popupMatchSelectWidth	下拉菜单和选择器同宽。默认将设置 `min-width`，当值小于选择框宽度时会被忽略。false 时会关闭虚拟滚动	boolean | number
dropdownRender	自定义下拉框内容	(originNode: ReactNode) => ReactNode
dropdownStyle	下拉菜单的 style 属性	CSSProperties
fieldNames	自定义节点 label、value、options、groupLabel 的字段	object
filterOption	是否根据输入项进行筛选。当其为一个函数时，会接收 `inputValue` `option` 两个参数，当 `option` 符合筛选条件时，应返回 true，反之则返回 false。示例	boolean | function(inputValue, option)
filterSort	搜索时对筛选结果项的排序函数，类似Array.sort里的 compareFunction	(optionA: Option, optionB: Optio info: { searchValue: string }) => number
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)
labelInValue	是否把每个选项的 label 包装到 value 中，会把 Select 的 value 类型从 `string` 变为 { value: string, label: ReactNode } 的格式	boolean
listHeight	设置弹窗滚动高度	number
loading	加载中状态	boolean

maxCount	指定可选中的最多 items 数量，仅在 `mode` 为 `multiple` 或 `tags` 时生效	number
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(omittedVa
maxTagTextLength	最大显示的 tag 文本长度	number
menuItemSelectedIcon	自定义多选时当前选中的条目图标	ReactNode
mode	设置 Select 的模式为多选或标签	`multiple` | `tags`
notFoundContent	当下拉列表为空时显示的内容	ReactNode
open	是否展开下拉菜单	boolean
optionFilterProp	搜索时过滤对应的 `option` 属性，如设置为 `children` 表示对内嵌内容进行搜索。若通过 `options` 属性配置选项内容，建议设置 `optionFilterProp="label"` 来对内容进行搜索。	string
optionLabelProp	回填到选择框的 Option 的属性值，默认是 Option 的子元素。比如在子元素需要高亮效果时，此值可以设为 `value`。示例	string
options	数据化配置选项内容，相比 jsx 定义会获得更好的渲染性能	{ label, value }[]
optionRender	自定义渲染下拉选项	(option: FlattenOptionData<BaseOptionT , info: { index: number }) => React.ReactNode
placeholder	选择框默认文本	string
placement	选择框弹出的位置	`bottomLeft` `bottomRight` `topL` `topRight`
prefix	自定义前缀	ReactNode
removeIcon	自定义的多选框清除图标	ReactNode

searchValue	控制搜索文本	string
showSearch	配置是否可搜索	boolean
size	选择框大小	`large` | `middle` | `small`
status	设置校验状态	'error' | 'warning'
suffixIcon	自定义的选择框后缀图标。以防止图标被用于其他交互，替换的图标默认不会响应展开、收缩事件，可以通过添加 `pointer-events: none` 样式透传。	ReactNode
tagRender	自定义 tag 内容 render，仅在 `mode` 为 `multiple` 或 `tags` 时生效	(props) => ReactNode
labelRender	自定义当前选中的 label 内容 render （LabelInValueType的定义见 LabelInValueType）	(props: LabelInValueType) => ReactNode
tokenSeparators	自动分词的分隔符，仅在 `mode="tags"` 时生效	string[]
value	指定当前选中的条目，多选时为一个数组。（value 数组引用未变化时，Select 不会更新）	string | string[] | number | number[] | LabeledValue | LabeledValue[]
variant	形态变体	`outlined` | `borderless` | `fille` `underlined`
virtual	设置 false 时关闭虚拟滚动	boolean
onBlur	失去焦点时回调	function
onChange	选中 option，或 input 的 value 变化时，调用此函数	function(value, option:Option | Array<Option>)
onClear	清除内容时回调	function
onDeselect	取消选中时调用，参数为选中项的 value (或 key) 值，仅在 `multiple` 或 `tags` 模式下生效	function(value: string | number | LabeledValue)
onDropdownVisibleChange	展开下拉菜单的回调	(open: boolean) => void
onFocus	获得焦点时回调	(event: FocusEvent) => void
onInputKeyDown	按键按下时回调	(event: KeyboardEvent) => void

onPopupScroll	下拉列表滚动时的回调	(event: UIEvent) => void
onSearch	文本框值变化时回调	function(value: string)
onSelect	被选中时调用，参数为选中项的 value (或 key) 值	function(value: string | number | LabeledValue, option: Option)

> *注意，如果发现下拉菜单跟随页面滚动，或者需要在其他弹层中触发 Select，请尝试使用 `getPopupContainer={triggerNode => triggerNode.parentElement}` 将下拉弹层渲染节点固定在触发器的父元素中。*

Select Methods

名称	说明	版本
blur()	取消焦点	
focus()	获取焦点	

Option props

参数	说明	类型	默认值	版本
className	Option 器类名	string	-	
disabled	是否禁用	boolean	false	
title	选项上的原生 title 提示	string	-	
value	默认根据此属性值进行筛选	string | number	-	

OptGroup props

参数	说明	类型	默认值	版本
key	Key	string	-	
label	组名	React.ReactNode	-	
className	Option 器类名	string	-	
title	选项上的原生 title 提示	string	-	

主题变量（Design Token）

FAQ

`mode="tags"` 模式下为何搜索有时会出现两个相同选项?

这一般是 `options` 中的 `label` 和 `value` 不同导致的，你可以通过 `optionFilterProp="label"` 将过滤设置为展示值以避免这种情况。

点击 `dropdownRender` 里的元素，下拉菜单不会自动消失？

你可以使用受控模式，手动设置 `open` 属性：codesandbox。

反过来希望点击 `dropdownRender` 里元素不消失该怎么办？

Select 当失去焦点时会关闭下拉框，如果你可以通过阻止默认行为为避免丢失焦点导致的关闭：

```
<Select
  dropdownRender={() => (
    <div
      onMouseDown={(e) => {
        e.preventDefault();
        e.stopPropagation();
      }}
    >
      Some Content
    </div>
  )}
/>
```

自定义 Option 样式导致滚动异常怎么办？

这是由于虚拟滚动默认选项高度为 `24px`，如果你的选项高度小于该值则需要通过 `listItemHeight` 属性调整，而 `listHeight` 用于设置滚动容器高度：

```
<Select listItemHeight={10} listHeight={250} />
```

注意： `listItemHeight` 和 `listHeight` 为内部属性，如无必要，请勿修改该值。

为何无障碍测试会报缺失 `aria-` 属性？

Select 无障碍辅助元素仅在弹窗展开时创建，因而当你在进行无障碍检测时请先打开下拉后再进行测试。对于 `aria-label` 与 `aria-labelledby` 属性缺失警告，请自行为 Select 组件添加相应无障碍属性。

Select 虚拟滚动会模拟无障碍绑定元素。如果需要读屏器完整获取全部列表，你可以设置 `virtual={false}` 关闭虚拟滚动，无障碍选项将会绑定到真实元素上。