

## zh-CN

对某一列数据进行筛选，使用列的 `filters` 属性来指定需要筛选菜单的列，`onFilter` 用于筛选当前数据，`filterMultiple` 用于指定多选和单选。

对某一列数据进行排序，通过指定列的 `sorter` 函数即可启动排序按钮。 `sorter: function(rowA, rowB) { ... }`，`rowA`、`rowB` 为比较的两个行数据。

`sortDirections: ['ascend' | 'descend']` 改变每列可用的排序方式，切换排序时按数组内容依次切换，设置在 `table props` 上时对所有列生效。你可以通过设置 `['ascend', 'descend', 'ascend']` 禁止排序恢复到默认状态。

使用 `defaultSortOrder` 属性，设置列的默认排序顺序。

## en-US

Use `filters` to generate filter menu in columns, `onFilter` to determine filtered result, and `filterMultiple` to indicate whether it's multiple or single selection.

Uses `defaultFilteredValue` to make a column filtered by default.

Use `sorter` to make a column sortable. `sorter` can be a function of the type `function(a, b) { ... }` for sorting data locally.

`sortDirections: ['ascend' | 'descend']` defines available sort methods for each columns, effective for all columns when set on table props. You can set as `['ascend', 'descend', 'ascend']` to prevent sorter back to default status.

Uses `defaultSortOrder` to make a column sorted by default.

If a `sortOrder` or `defaultSortOrder` is specified with the value `ascend` or `descend`, you can access this value from within the function passed to the `sorter` as explained above. Such a function can take the form:

```
function(a, b, sortOrder) { ... }.
```

```
import { Table } from 'antd';

const columns = [
  {
    title: 'Name',
    dataIndex: 'name',
    filters: [
      {
        text: 'Joe',
        value: 'Joe',
      },
      {
        text: 'Jim',
        value: 'Jim',
      },
      {
        text: 'Submenu',
        value: 'Submenu',
      },
    ],
  },
];
```

```

        children: [
          {
            text: 'Green',
            value: 'Green',
          },
          {
            text: 'Black',
            value: 'Black',
          },
        ],
      },
    ],
    // specify the condition of filtering result
    // here is that finding the name started with `value`
    onFilter: (value, record) => record.name.indexOf(value) === 0,
    sorter: (a, b) => a.name.length - b.name.length,
    sortDirections: ['descend'],
  },
  {
    title: 'Age',
    dataIndex: 'age',
    defaultSortOrder: 'descend',
    sorter: (a, b) => a.age - b.age,
  },
  {
    title: 'Address',
    dataIndex: 'address',
    filters: [
      {
        text: 'London',
        value: 'London',
      },
      {
        text: 'New York',
        value: 'New York',
      },
    ],
    onFilter: (value, record) => record.address.indexOf(value) === 0,
  },
];

const data = [
  {
    key: '1',
    name: 'John Brown',
    age: 32,
    address: 'New York No. 1 Lake Park',
  },
  {
    key: '2',
    name: 'Jim Green',
    age: 42,
  },
];

```

```
    address: 'London No. 1 Lake Park',
  },
  {
    key: '3',
    name: 'Joe Black',
    age: 32,
    address: 'Sidney No. 1 Lake Park',
  },
  {
    key: '4',
    name: 'Jim Red',
    age: 32,
    address: 'London No. 2 Lake Park',
  },
];

function onChange(pagination, filters, sorter, extra) {
  console.log('params', pagination, filters, sorter, extra);
}

export default () => <Table columns={columns} dataSource={data} onChange={onChange}
/>;
```