

zh-CN

这里列出了支持 rtl 方向的组件，您可以在演示中切换方向。

en-US

Components which support rtl direction are listed here, you can toggle the direction in the demo.

```
import {
  Input,
  Col,
  Row,
  Select,
  InputNumber,
  ConfigProvider,
  Cascader,
  Radio,
  Switch,
  Tree,
  TreeSelect,
  Modal,
  Button,
  Pagination,
  Steps,
  Rate,
  Badge,
  Divider,
} from 'antd';

import {
  SearchOutlined as SearchIcon,
  SmileOutlined,
  DownloadOutlined,
  LeftOutlined,
  RightOutlined,
  MinusOutlined,
  PlusOutlined,
} from '@ant-design/icons';

const InputGroup = Input.Group;
const ButtonGroup = Button.Group;
const { Option } = Select;
const { TreeNode } = Tree;
const { Search } = Input;
const { Step } = Steps;

const cascaderOptions = [
  {
    value: 'tehran',
    label: 'تهران',
    children: [
```

```

    {
      value: 'tehran-c',
      label: 'تهران',
      children: [
        {
          value: 'saadat-abad',
          label: 'سعادت آباد',
        },
      ],
    },
  ],
},
{
  value: 'ardabil',
  label: 'اردبیل',
  children: [
    {
      value: 'ardabil-c',
      label: 'اردبیل',
      children: [
        {
          value: 'primadar',
          label: 'پیرمادر',
        },
      ],
    },
  ],
},
{
  value: 'gilan',
  label: 'گیلان',
  children: [
    {
      value: 'rasht',
      label: 'رشت',
      children: [
        {
          value: 'district-3',
          label: 'منطقه ۳',
        },
      ],
    },
  ],
},
],
};

class Page extends React.Component {
  state = {
    currentStep: 0,
    modalVisible: false,

    badgeCount: 5,
  };
}

```

```

    showBadge: true,
  };

  selectBefore = (
    <Select defaultValue="Http://" style={{ width: 90 }}>
      <Option value="Http://">Http://</Option>
      <Option value="Https://">Https://</Option>
    </Select>
  );

  selectAfter = (
    <Select defaultValue=".com" style={{ width: 80 }}>
      <Option value=".com">.com</Option>
      <Option value=".jp">.jp</Option>
      <Option value=".cn">.cn</Option>
      <Option value=".org">.org</Option>
    </Select>
  );

  // ==== Cascader ====
  cascaderFilter = (inputValue, path) =>
    path.some(option => option.label.toLowerCase().indexOf(inputValue.toLowerCase())
  > -1);

  onCascaderChange = value => {
    console.log(value);
  };
  // ==== End Cascader ====

  // ==== Modal ====
  showModal = () => {
    this.setState({
      modalVisible: true,
    });
  };

  handleOk = e => {
    console.log(e);
    this.setState({
      modalVisible: false,
    });
  };

  handleCancel = e => {
    console.log(e);
    this.setState({
      modalVisible: false,
    });
  };
  // ==== End Modal ====

  onStepsChange = currentStep => {

```

[illegible]

```

<Divider orientation="left">Switch example</Divider>
 
<Switch defaultChecked />
 
<Switch loading defaultChecked />
 
<Switch size="small" loading />
</Col>
<Col span={12}>
  <Divider orientation="left">Radio Group example</Divider>

  <Radio.Group defaultValue="c" buttonStyle="solid">
    <Radio.Button value="a">تهران</Radio.Button>
    <Radio.Button value="b" disabled>
      اصفهان
    </Radio.Button>
    <Radio.Button value="c">فارس</Radio.Button>
    <Radio.Button value="d">خوزستان</Radio.Button>
  </Radio.Group>
</Col>
</Row>
<br />
<Row>
  <Col span={12}>
    <Divider orientation="left">Button example</Divider>
    <div className="button-demo">
      <Button type="primary" icon={<DownloadOutlined />} />
      <Button type="primary" shape="circle" icon={<DownloadOutlined />} />
      <Button type="primary" shape="round" icon={<DownloadOutlined />} />
      <Button type="primary" shape="round" icon={<DownloadOutlined />}>
        Download
      </Button>
      <Button type="primary" icon={<DownloadOutlined />}>
        Download
      </Button>
    <br />
    <Button.Group>
      <Button type="primary">
        <LeftOutlined />
        Backward
      </Button>
      <Button type="primary">
        Forward
        <RightOutlined />
      </Button>
    </Button.Group>
    <Button type="primary" loading>
      Loading
    </Button>
    <Button type="primary" size="small" loading>
      Loading
    </Button>
  </Col>
</Row>

```

```

    </div>
</Col>
<Col span={12}>
  <Divider orientation="left">Tree example</Divider>
  <Tree
    showLine
    checkable
    defaultExpandedKeys={['0-0-0', '0-0-1']}
    defaultSelectedKeys={['0-0-0', '0-0-1']}
    defaultCheckedKeys={['0-0-0', '0-0-1']}
  >
    <TreeNode title="parent 1" key="0-0">
      <TreeNode title="parent 1-0" key="0-0-0" disabled>
        <TreeNode title="leaf" key="0-0-0-0" disableCheckbox />
        <TreeNode title="leaf" key="0-0-0-1" />
      </TreeNode>
      <TreeNode title="parent 1-1" key="0-0-1">
        <TreeNode title={<span style={{ color: '#1890ff' }}>sss</span>}
key="0-0-1-0" />
      </TreeNode>
    </TreeNode>
  </Tree>
</Col>
</Row>
<br />
<Row>
  <Col span={24}>
    <Divider orientation="left">Input (Input Group) example</Divider>
    <InputGroup size="large">
      <Row gutter={8}>
        <Col span={5}>
          <Input defaultValue="0571" />
        </Col>
        <Col span={8}>
          <Input defaultValue="26888888" />
        </Col>
      </Row>
    </InputGroup>
    <br />
    <InputGroup compact>
      <Input style={{ width: '20%' }} defaultValue="0571" />
      <Input style={{ width: '30%' }} defaultValue="26888888" />
    </InputGroup>
    <br />
    <InputGroup compact>
      <Select defaultValue="Option1">
        <Option value="Option1">Option1</Option>
        <Option value="Option2">Option2</Option>
      </Select>
      <Input style={{ width: '50%' }} defaultValue="input content" />
      <InputNumber />
    </InputGroup>
  </Col>
</Row>

```

```

        <br />
        <Search placeholder="input search text" enterButton="Search"
size="large" />
        <br />
        <br />
        <div style={{ marginBottom: 16 }}>
            <Input
                addonBefore={this.selectBefore}
                addonAfter={this.selectAfter}
                defaultValue="mysite"
            />
        </div>
        <br />
        <Row>
            <Col span={12}>
                <Divider orientation="left">Select example</Divider>
                <Select mode="multiple" defaultValue="مورچه" style={{ width: 120 }}>
                    <Option value="jack">Jack</Option>
                    <Option value="مورچه">مورچه</Option>
                    <Option value="disabled" disabled>
                        Disabled
                    </Option>
                    <Option value="Yiminghe">yiminghe</Option>
                </Select>
                <Select defaultValue="مورچه" style={{ width: 120 }} disabled>
                    <Option value="مورچه">مورچه</Option>
                </Select>
                <Select defaultValue="مورچه" style={{ width: 120 }} loading>
                    <Option value="مورچه">مورچه</Option>
                </Select>
                <Select
                    showSearch
                    style={{ width: 200 }}
                    placeholder="Select a person"
                    optionFilterProp="children"
                    filterOption={(input, option) =>
                        option.props.children.toLowerCase().indexOf(input.toLowerCase())
                    }
                >
                    <Option value="jack">Jack</Option>
                    <Option value="سعید">سعید</Option>
                    <Option value="tom">Tom</Option>
                </Select>
            </Col>
            <Col span={12}>
                <Divider orientation="left">TreeSelect example</Divider>
                <div>
                    <TreeSelect
                        showSearch
                        style={{ width: '100%' }}
                        dropdownStyle={{ maxHeight: 400, overflow: 'auto' }}

```

>= 0

```

placeholder="Please select"
allowClear
treeDefaultExpandAll
>
<TreeNode value="parent 1" title="parent 1" key="0-1">
  <TreeNode value="parent 1-0" title="parent 1-0" key="0-1-1">
    <TreeNode value="leaf1" title="my leaf" key="random" />
    <TreeNode value="leaf2" title="your leaf" key="random1" />
  </TreeNode>
  <TreeNode value="parent 1-1" title="parent 1-1" key="random2">
    <TreeNode
      value="sss"
      title={<b style={{ color: '#08c' }}>sss</b>}
      key="random3"
    />
  </TreeNode>
</TreeNode>
</TreeSelect>
</div>
</Col>
</Row>
<br />
<Row>
  <Col span={24}>
    <Divider orientation="left">Modal example</Divider>
    <div>
      <Button type="primary" onClick={this.showModal}>
        Open Modal
      </Button>
      <Modal
        title="پنجره ساده"
        visible={this.state.modalVisible}
        onOk={this.handleOk}
        onCancel={this.handleCancel}
      >
        <p>نگاشته های خود را اینجا قرار دهید</p>
        <p>نگاشته های خود را اینجا قرار دهید</p>
        <p>نگاشته های خود را اینجا قرار دهید</p>
      </Modal>
    </div>
  </Col>
</Row>
<br />
<Row>
  <Col span={24}>
    <Divider orientation="left">Steps example</Divider>
    <div>
      <Steps progressDot current={currentStep}>
        <Step title="Finished" description="This is a description." />
        <Step title="In Progress" description="This is a description."

        <Step title="Waiting" description="This is a description." />
      </Steps>
    </div>
  </Col>
</Row>
</div>

```



```

        </Steps>
        <br />
        <Steps current={currentStep} onChange={this.onStepsChange}>
          <Step title="Step 1" description="This is a description." />
          <Step title="Step 2" description="This is a description." />
          <Step title="Step 3" description="This is a description." />
        </Steps>
      </div>
    </Col>
  </Row>
  <br />
  <Row>
    <Col span={12}>
      <Divider orientation="left">Rate example</Divider>
      <div>
        <Rate defaultValue={2.5} />
        <br />
        <strong>* Note:</strong> Half star not implemented in RTL
direction, it will be
        supported after <a href="https://github.com/react-
component/rate">rc-rate</a>{' '}
        implement rtl support.
      </div>
    </Col>
    <Col span={12}>
      <Divider orientation="left">Badge example</Divider>
      <div>
        <div>
          <Badge count={this.state.badgeCount}>
            <a href="#" className="head-example" />
          </Badge>
          <ButtonGroup>
            <Button onClick={this.declineBadge}>
              <MinusOutlined />
            </Button>
            <Button onClick={this.increaseBadge}>
              <PlusOutlined />
            </Button>
          </ButtonGroup>
        </div>
        <div style={{ marginTop: 10 }}>
          <Badge dot={this.state.showBadge}>
            <a href="#" className="head-example" />
          </Badge>
          <Switch onChange={this.onChangeBadge} checked=
{this.state.showBadge} />
        </div>
      </div>
    </Col>
  </Row>
</Col>
</Row>

```

```

<br />
<br />
<Row>
  <Col span={24}>
    <Divider orientation="left">Pagination example</Divider>
    <Pagination showSizeChanger defaultCurrent={3} total={500} />
  </Col>
</Row>
<br />
<Row>
  <Col span={24}>
    <Divider orientation="left">Grid System example</Divider>
    <div className="grid-demo">
      <div className="code-box-demo">
        <p>
          <strong>* Note:</strong> Every calculation in RTL grid system is
from right side
          (offset, push, etc.)
        </p>
        <Row>
          <Col span={8}>col-8</Col>
          <Col span={8} offset={8}>
            col-8
          </Col>
        </Row>
        <Row>
          <Col span={6} offset={6}>
            col-6 col-offset-6
          </Col>
          <Col span={6} offset={6}>
            col-6 col-offset-6
          </Col>
        </Row>
        <Row>
          <Col span={12} offset={6}>
            col-12 col-offset-6
          </Col>
        </Row>
        <Row>
          <Col span={18} push={6}>
            col-18 col-push-6
          </Col>
          <Col span={6} pull={18}>
            col-6 col-pull-18
          </Col>
        </Row>
      </div>
    </div>
  </Col>
</Row>
</div>

```

```

    );
  }
}

class App extends React.Component {
  state = {
    direction: 'ltr',
    popupPlacement: 'bottomLeft',
  };

  changeDirection = e => {
    const directionValue = e.target.value;
    this.setState({ direction: directionValue });
    if (directionValue === 'rtl') {
      this.setState({ popupPlacement: 'bottomRight' });
    } else {
      this.setState({ popupPlacement: 'bottomLeft' });
    }
  };

  render() {
    const { direction, popupPlacement } = this.state;
    return (
      <>
        <div style={{ marginBottom: 16 }}>
          <span style={{ marginRight: 16 }}>Change direction of components: </span>
          <Radio.Group defaultValue="ltr" onChange={this.changeDirection}>
            <Radio.Button key="ltr" value="ltr">
              LTR
            </Radio.Button>
            <Radio.Button key="rtl" value="rtl">
              RTL
            </Radio.Button>
          </Radio.Group>
        </div>
        <ConfigProvider direction={direction}>
          <Page className={direction} popupPlacement={popupPlacement} />
        </ConfigProvider>
      </>
    );
  }
}

export default () => <App />;

```

```

.button-demo .ant-btn,
.button-demo .ant-btn-group {
  margin-right: 8px;
  margin-bottom: 12px;
}
.button-demo .ant-btn-group > .ant-btn,

```

```
.button-demo .ant-btn-group > span > .ant-btn {  
  margin-right: 0;  
  margin-left: 0;  
}  
  
.head-example {  
  display: inline-block;  
  width: 42px;  
  height: 42px;  
  vertical-align: middle;  
  background: #eee;  
  border-radius: 4px;  
}  
  
.ant-badge:not(.ant-badge-not-a-wrapper) {  
  margin-right: 20px;  
}  
  
.ant-badge-rtl:not(.ant-badge-not-a-wrapper) {  
  margin-right: 0;  
  margin-left: 20px;  
}
```