

zh-CN

可以简单配置几种等分栅格和间距。

en-US

A simple playground for column count and gutter.

```
import { Row, Col, Slider } from 'antd';

const gutters = {};
const vgutters = {};
const colCounts = {};

[8, 16, 24, 32, 40, 48].forEach((value, i) => {
  gutters[i] = value;
});
[8, 16, 24, 32, 40, 48].forEach((value, i) => {
  vgutters[i] = value;
});
[2, 3, 4, 6, 8, 12].forEach((value, i) => {
  colCounts[i] = value;
});

class App extends React.Component {
  state = {
    gutterKey: 1,
    vgutterKey: 1,
    colCountKey: 2,
  };

  onGutterChange = gutterKey => {
    this.setState({ gutterKey });
  };

  onVGutterChange = vgutterKey => {
    this.setState({ vgutterKey });
  };

  onColCountChange = colCountKey => {
    this.setState({ colCountKey });
  };

  render() {
    const { gutterKey, vgutterKey, colCountKey } = this.state;
    const cols = [];
    const colCount = colCounts[colCountKey];
    let colCode = '';
    for (let i = 0; i < colCount; i++) {
      cols.push(
        <Col key={i.toString()} span={24 / colCount}>
```

```

        <div>Column</div>
    </Col>,
);
colCode += `    <Col span=${24 / colCount} />\n`;
}
return (
    <>
        <span>Horizontal Gutter (px): </span>
        <div style={{ width: '50%' }}>
            <Slider
                min={0}
                max={Object.keys(gutters).length - 1}
                value={gutterKey}
                onChange={this.onGutterChange}
                marks={gutters}
                step={null}
                tipFormatter={value => gutters[value]}
            />
        </div>
        <span>Vertical Gutter (px): </span>
        <div style={{ width: '50%' }}>
            <Slider
                min={0}
                max={Object.keys(vgutters).length - 1}
                value={vgutterKey}
                onChange={this.onVGutterChange}
                marks={vgutters}
                step={null}
                tipFormatter={value => vgutters[value]}
            />
        </div>
        <span>Column Count:</span>
        <div style={{ width: '50%', marginBottom: 48 }}>
            <Slider
                min={0}
                max={Object.keys(colCounts).length - 1}
                value={colCountKey}
                onChange={this.onColCountChange}
                marks={colCounts}
                step={null}
                tipFormatter={value => colCounts[value]}
            />
        </div>
        <Row gutter={[gutters[gutterKey], vgutters[vgutterKey]]}>
            {cols}
            {cols}
        </Row>
        Another Row:
        <Row gutter={[gutters[gutterKey], vgutters[vgutterKey]]}>{cols}</Row>
        <pre className="demo-code">{`<Row gutter=[${gutters[gutterKey]},
    ${vgutters[vgutterKey]}]}>\n${colCode}\n${colCode}</Row>`}</pre>
        <pre className="demo-code">{`<Row gutter=[${gutters[gutterKey]},

```

```

    ${vgutters[vgutterKey]}}}}>\n${colCode}</Row>`}</pre>
    </>
  );
}
}

export default () => <App />;

```

```

#components-grid-demo-playground [class~='ant-col'] {
  background: transparent;
  border: 0;
}
#components-grid-demo-playground [class~='ant-col'] > div {
  height: 120px;
  font-size: 14px;
  line-height: 120px;
  background: #0092ff;
  border-radius: 4px;
}
#components-grid-demo-playground pre {
  padding: 8px 16px;
  font-size: 13px;
  background: #f9f9f9;
  border-radius: 6px;
}
#components-grid-demo-playground pre.demo-code {
  direction: ltr;
}
#components-grid-demo-playground .ant-col {
  padding: 0;
}

```