

Button使用

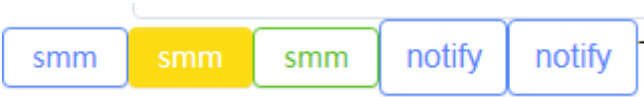
通过设置 Button 的属性来产生不同的按钮样式。

按钮的属性说明如下：

属性	说明	类型	默认值	版本
disabled	按钮失效状态	boolean	false	1.0
href	点击跳转的地址，指定此属性 button 的行为和 a 链接一致	string	-	1.0
size	设置按钮大小	lg	md	sm
btnType	设置按钮类型	primary	info	success
onClick	点击按钮时的回调	(event) => void	-	1.0

支持原生 button 的其他所有属性。

```
<Button className="btn-class" btnType="warning" size="ssm"></Button>
```



Menu菜单

参数	说明	类型	默认值	版本
description	警告提示的辅助性文字介绍	ReactNode	-	1.0
message	警告提示内容	ReactNode	-	1.0
showIcon	是否显示辅助图标	boolean	false	1.0
type	指定警告提示的样式，有四种选择 success、info、warning、error	string	info， banner 模式下默认值为 warning	1.0
onClose	关闭时触发的回调函数	() => void	-	1.0

```
<Alert
  title="警告"
  type="error"
  message="删除账号前请先解散或转移你的团队和工程。 该操作不可恢复！删除账号仅对创立EDA上的
  账号与数据进行删除， 如果需要删除创立创商城/嘉立创账号请联系创立创商城或嘉立创。"></Alert>

<Alert
  type="error"
  showIcon
  close
  message="当前账号尚未激活邮箱，无法开启邮件提醒功能">
</Alert>
```



Alert提醒框使用

参数	说明	类型	默认值	版本
description	警告提示的辅助性文字介绍	ReactNode	-	1.0
message	警告提示内容	ReactNode	-	1.0
showIcon	是否显示辅助图标	boolean	false	1.0
type	指定警告提示的样式，有四种选择 success、info、warning、error	string	info，banner 模式下默认值为 warning	1.0
onClose	关闭时触发的回调函数	() => void	-	1.0

```
<Alert
  title="警告"
  type="error"
  message="删除账号前请先解散或转移你的团队和工程。 该操作不可恢复！删除账号仅对创立EDA上的
  账号与数据进行删除， 如果需要删除创立创商城/嘉立创账号请联系创立创商城或嘉立创。"></Alert>

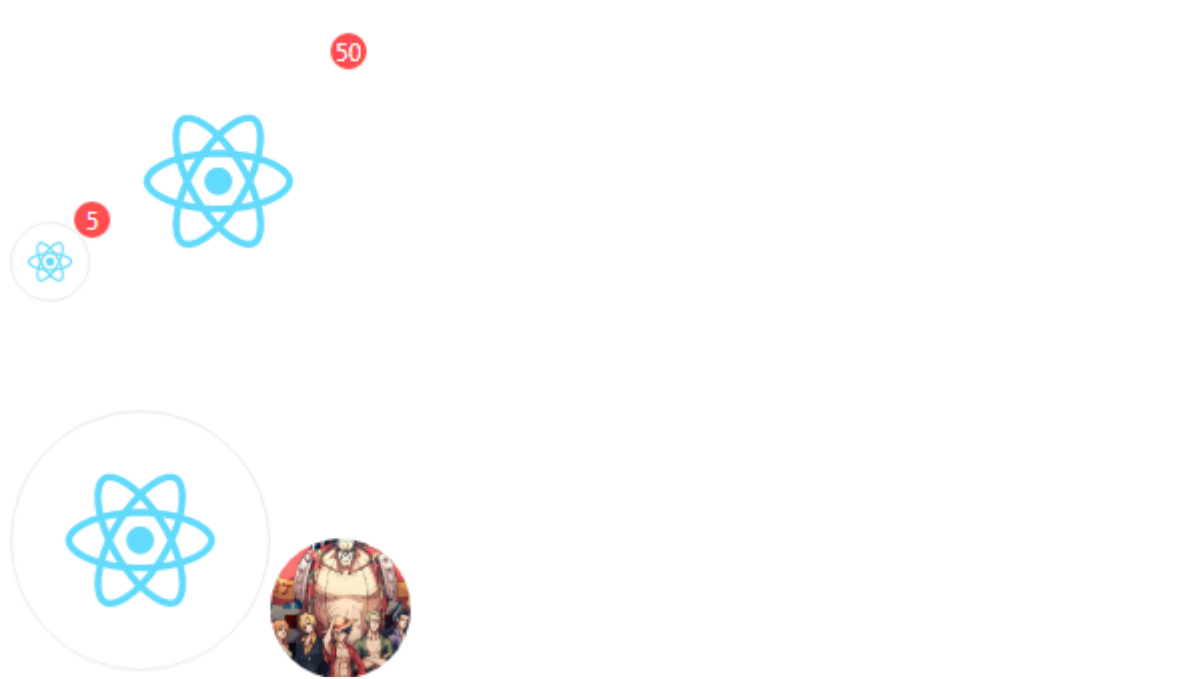
<Alert
  type="error"
  showIcon
  close
  message="当前账号尚未激活邮箱，无法开启邮件提醒功能">
</Alert>
```



Avatar头像

参数	说明	类型	默认值	版本
alt	图像无法显示时的替代文本	string	-	1.0
size	设置头像的大小	lg ssm	sm	default
href	图片头像的资源地址	string	-	1.0
border	边框	boolean	false	1.0

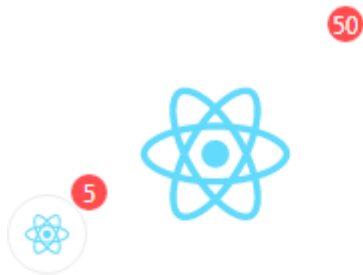
```
<Avatar href="xxxxxxxxxxxxx" border="true" size="ssm"></Avatar>
```



Badge微标数

参数	说明	类型	默认值	版本
bgColor	字体颜色	string	-	1.0
count	展示的数字，大于 overflowCount 时显示为 <code>`\${overflowCount}+`</code> ，为 0 时隐藏	number	0	1.0
overflowCount	展示封顶的数字值	number	99	1.0

```
<Badge count={5} overflowCount={99}>
  <Avatar href={sv} border="true" size="ssm"></Avatar>
</Badge>
```



Breadcrumb面包屑

参数	说明	类型	默认值	版本
separator	分隔符	string	"/"	1.0

Breadcrumb.Item

参数	说明	类型	默认值	版本
to	路由地址，有传递则设置为可跳转	string		1.0

```
<Breadcrumb>
  <Breadcrumb.Item to="/aaa">Ant Design</Breadcrumb.Item>
  <Breadcrumb.Item to="/aaa">Component</Breadcrumb.Item>
  <Breadcrumb.Item>General</Breadcrumb.Item>
</Breadcrumb>
```

Ant Design / Component / General

DropDown下拉列表

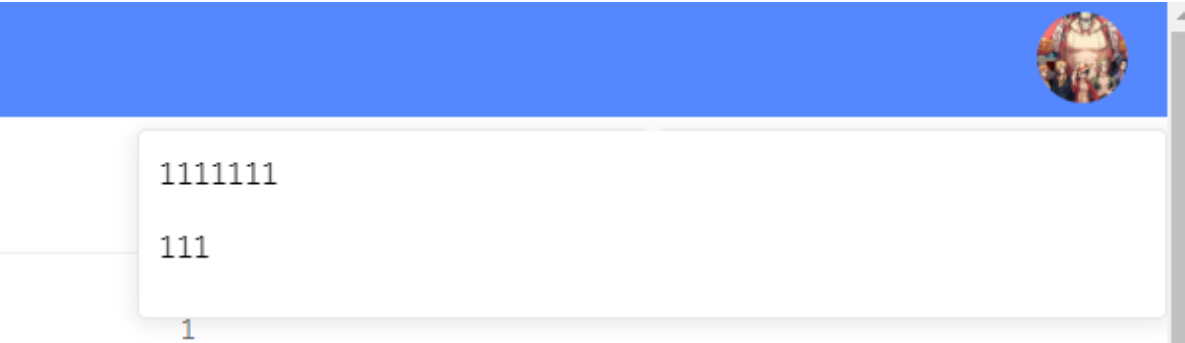
参数	说明	类型	默认值	版本
overlay	menu项，暂定包裹全部元素	DropDownMenu		1.0
position	menu位置	IPosition	center	1.0
positionClass	下拉菜单定位类名			1.0

DropDownMenu

参数	说明	类型	默认值	版本
children	menu内容	ReactNode		1.0

```
type IPosition = "center" | "left" | "right"

<DropDown overlay={
  <DropDownMenu className="drop-menu">
    1111111
    <p>111</p>
  </DropDownMenu>}>
  <div className="nav-notify">
    <span><Icon type="nc-u-notice-header"></Icon></span>
    <Avatar size="sm"
href="https://image.lceda.cn/avatars/2020/10/e36x5ZmcEpm2ibm91N30rs2CzoHeGz8Ew0eR6dKD.jpeg"></Avatar>
  </div>
</DropDown>
```



Icon

参数	说明	类型	默认值	版本
type	图标类型，参考原网站图标类型	string		1.0
iconType	引入iconfont图标类型	string		1.0
color	图标颜色	string		1.0
style	样式	CSSProperties		1.0
onClick	点击事件	()=>void		1.0

```
<Icon type="nc-u-search"></Icon>
```

Input输入框

参数	说明	类型	默认值	版本
type	声明 input 类型，同原生 input 标签的 type 属性，请直接使用 <code>TextArea</code> 代替 <code>type="textarea"</code>	string		1.0
value	输入框内容	string		1.0
disabled	是否禁用	false		1.0
onChangeInput	点击事件	(value) =>void		1.0

```
<Input className="myClass" placeholder="name" onChangeInput=
{(val:string)=>{console.log(val);
}}></Input>
```

Message提示消息

参数	说明	类型	默认值	版本
type	信息类型	string	default	1.0
message	消息内容	string		1.0
close	关闭按钮	boolean	false	1.0
style	样式	CSSProperties		1.0
closeCallback	点击事件	()=>void		1.0

```
<Message
  type="error"
  close
  message="当前账号尚未当前账号箱"
></Message>
```



Modal模态框

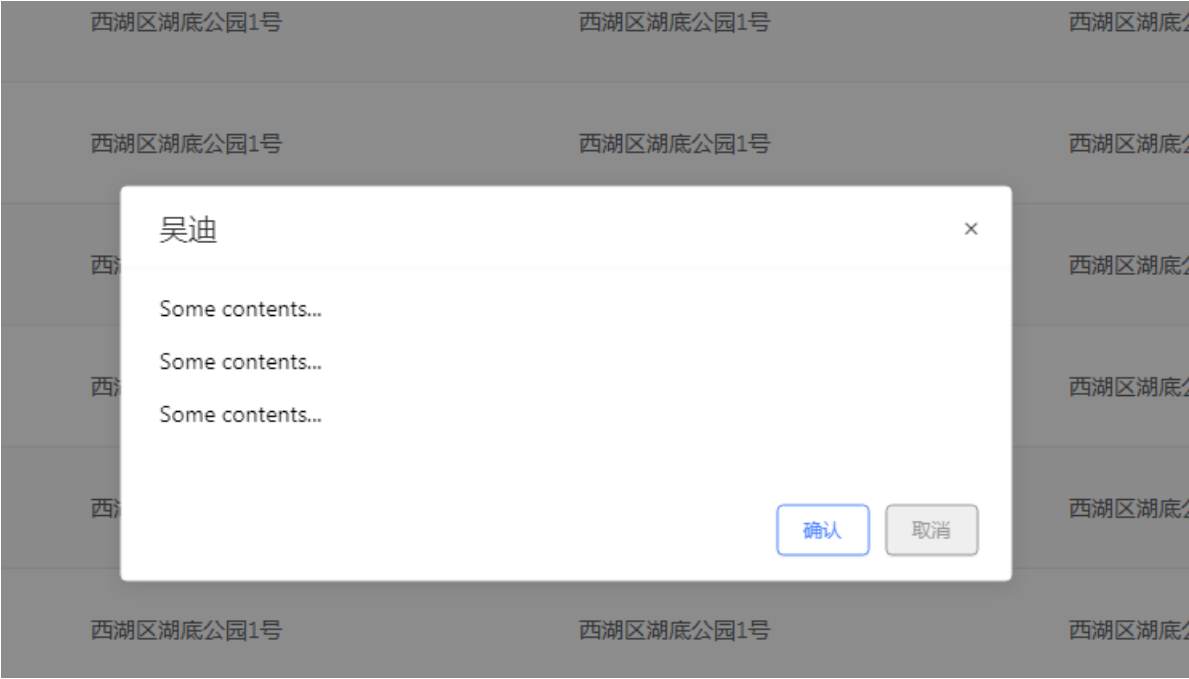
参数	说明	类型	默认值	版本
visible	模态框消失隐藏	boolean	false	1.0
title	模态框标题	ReactNode		1.0
showClose	是否显示关闭按钮	boolean	false	1.0
onCancel	取消回调	() => void		1.0
onOk	确认回调	() => void		1.0
onCancelText	取消按钮信息	string	取消	1.0
onOkText	确认按钮信息	string	确定	1.0

```
<Modal visible={vis}
  className="modal-big"
  showClose
  title="吴迪" onOk={() => {
    console.log("ok");
    setVis(false)
  }}
  onCancel={() => {
    setVis(false)
    console.log("onCancel");
  }} >
  <p>Some contents...</p>
  <p>Some contents...</p>
  <p>Some contents...</p>
</Modal>
```

Notification.open通知消息

参数	说明	类型	默认值	版本
type	通知类型	boolean	false	1.0
message	通知信息	string		1.0
duration	通知消失时间	number（单位秒）	3	1.0

```
<Button onClick={() => {
  Notification
    .open({
      message: "error" + Math.random(),
      type: "error",
      duration: 3
    })
}}>
  notify
</Button>
```



Pagination分页

参数	说明	类型	默认值	版本
defaultCurrent	默认页	number	1	1.0
current	当前页	number	1	1.0
defaultPageSize	默认页面大小	number	20	1.0
pageSize	页面数据大小	number	20	1.0
total	数据总数	number	0	1.0
groupCount	显示页码大小	number	7	
onChange	页码改变的回调	(page: number, pageSize: number) => void		

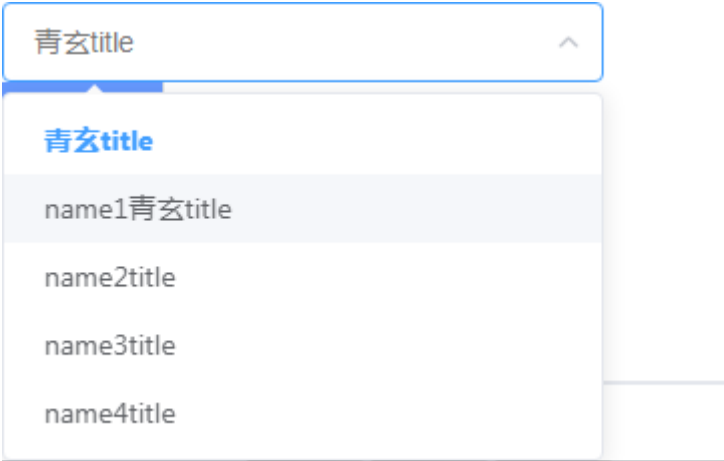
```
<Pagination total={400} onChange={(page, pageSize) => {
  console.log(page, pageSize);
}}></Pagination>
```



Select选择器

参数	说明	类型	默认值	版本
defaultValue	默认option 的value, 输入框显示option 对应的值	string		1.0
placeholder	输入框提示信息	string		1.0
disabled	是否禁用	boolean	false	1.0
onFocus	获得焦点事件,可以获得焦点后发起请求	() => void		1.0
onBlur	失去焦点事件	() => void		1.0
onChange	option改变事件.结束option中value的值	(val) => void		1.0
suffix	输入框后端图标,需传递图标,来表示输入框是否可输入	React.ReactNode		1.0
onInput	输入框变化事件, 接受输入框的值	(val) => void		1.0

```
<Select defaultValue="青玄"
  onChange={(value: string) => {
    console.log(value);
  }}>
  <Option value="青玄">青玄title</Option>
  <Option value="name1">name1青玄title</Option>
  <Option value="name2">name2title</Option>
  <Option value="name3">name3title</Option>
  <Option value="name4">name4title</Option>
</Select>
```



Cascader级联选择器

参数	说明	类型	默认值	版本
defaultValue	默认选择的value, 输入框显示value对应的值	string[]	[]	1.0
placeholder	输入框提示信息	string		1.0
separator	连接分隔符	string		1.0
onFocus	获得焦点事件,可以获得焦点后发起请求	() => void		1.0
onChange	选中改变事件.结束选中value对应的值展示在输入框中	(ret:IChangeResult) => void		1.0
onInput	输入框变化事件, 接受输入框的值(待定)	(val) => void		1.0

```
let option = [{
  value: 'zhinan',
  label: '指南',
  children: [{
    value: 'shejiyuanze',
    label: '设计原则',
    children: [{
      value: 'yizhi',
      label: '一致'
    }, {
      value: 'fankui',
      label: '反馈'
    }, {
      value: 'xiaolv',
      label: '效率'
    }, {
      value: 'kekong',
      label: '可控'
    }
  ]
}, {
  value: 'daohang',
  label: '导航',
  children: [{
    value: 'cexiangdaohang',
    label: '侧向导航'
  }, {
    value: 'dingbudaohang',
    label: '顶部导航'
  }
]
}, {
  value: 'zujian',
  label: '组件',
  children: [{
    value: 'basic',
```

```
label: 'Basic',
children: [{
  value: 'layout',
  label: 'Layout 布局'
}, {
  value: 'color',
  label: 'Color 色彩'
}, {
  value: 'typography',
  label: 'Typography 字体'
}, {
  value: 'icon',
  label: 'Icon 图标'
}, {
  value: 'button',
  label: 'Button 按钮'
}]
}, {
  value: 'form',
  label: 'Form',
  children: [{
    value: 'radio',
    label: 'Radio 单选框'
  }, {
    value: 'checkbox',
    label: 'Checkbox 多选框'
  }, {
    value: 'input',
    label: 'Input 输入框'
  }, {
    value: 'input-number',
    label: 'InputNumber 计数器'
  }, {
    value: 'select',
    label: 'Select 选择器'
  }, {
    value: 'cascader',
    label: 'Cascader 级联选择器'
  }, {
    value: 'switch',
    label: 'Switch 开关'
  }, {
    value: 'slider',
    label: 'Slider 滑块'
  }, {
    value: 'time-picker',
    label: 'TimePicker 时间选择器'
  }, {
    value: 'date-picker',
    label: 'DatePicker 日期选择器'
  }, {
    value: 'datetime-picker',
    label: 'DateTimePicker 日期时间选择器'
  }, {
    value: 'upload',
    label: 'Upload 上传'
  }, {
    value: 'rate',
```

```
    label: 'Rate 评分'
  }, {
    value: 'form',
    label: 'Form 表单'
  }]
}, {
  value: 'data',
  label: 'Data',
  children: [{
    value: 'table',
    label: 'Table 表格'
  }, {
    value: 'tag',
    label: 'Tag 标签'
  }, {
    value: 'progress',
    label: 'Progress 进度条'
  }, {
    value: 'tree',
    label: 'Tree 树形控件'
  }, {
    value: 'pagination',
    label: 'Pagination 分页'
  }, {
    value: 'badge',
    label: 'Badge 标记'
  }]
}, {
  value: 'notice',
  label: 'Notice',
  children: [{
    value: 'alert',
    label: 'Alert 警告'
  }, {
    value: 'loading',
    label: 'Loading 加载'
  }, {
    value: 'message',
    label: 'Message 消息提示'
  }, {
    value: 'message-box',
    label: 'MessageBox 弹框'
  }, {
    value: 'notification',
    label: 'Notification 通知'
  }]
}, {
  value: 'navigation',
  label: 'Navigation',
  children: [{
    value: 'menu',
    label: 'NavMenu 导航菜单'
  }, {
    value: 'tabs',
    label: 'Tabs 标签页'
  }, {
    value: 'breadcrumb',
    label: 'Breadcrumb 面包屑'
  }
}
```

```

    }, {
      value: 'dropdown',
      label: 'Dropdown 下拉菜单'
    }, {
      value: 'steps',
      label: 'Steps 步骤条'
    }
  ]
}, {
  value: 'others',
  label: 'Others',
  children: [{
    value: 'dialog',
    label: 'Dialog 对话框'
  }, {
    value: 'tooltip',
    label: 'Tooltip 文字提示'
  }, {
    value: 'popover',
    label: 'Popover 弹出框'
  }, {
    value: 'card',
    label: 'Card 卡片'
  }, {
    value: 'carousel',
    label: 'Carousel 走马灯'
  }, {
    value: 'collapse',
    label: 'Collapse 折叠面板'
  }
  ]
}]
}, {
  value: 'ziyuan',
  label: '资源',
  children: [{
    value: 'axure',
    label: 'Axure Components'
  }, {
    value: 'sketch',
    label: 'Sketch Templates'
  }, {
    value: 'jiaohu',
    label: '组件交互文档'
  }
  ]
}]
}]

```

```

interface IOption {
  value: string,
  label: string,
  children?: IOption[]
}

interface IChangeResult {
  path: string[],
  val: string
}

let option: IOption[] = []
<div style={{width:500}}>
  <Cascader options={option}

```

指南-设计原则-一致

指南	>	设计原则	>	一致
组件	>	导航	>	反馈
资源	>			效率
				可控

Spin 加载Loading

参数	说明	类型	默认值	版本
tip	loading提示	string		1.0
spinning	是否显示	boolean	false	1.0

```
<div className="">
  <Spin spinning={true} ></Spin>
</div>
<Spin spinning={spin} tip="loading...">
  <div data-show="true" className="eda-alert ant-alert-info ant-alert-with-
description ant-alert-no-icon" role="alert">
    <div className="ant-alert-content">
      <div className="ant-alert-message">
        Alert message title</div>
      <div className="ant-alert-description">
        Further details about the context of this alert.
      </div>
    </div>
  </div>
</Spin>
```

Switch开关

参数	说明	类型	默认值	版本
defaultChecked	默认状态	boolean	false	1.0
disabled	是否禁用	boolean	false	1.0
onChange	状态改变回调	(value: boolean) => void		1.0

```
<Switch disabled defaultChecked></Switch>
<Switch defaultChecked onChange={ (checked) => {
  console.log(checked);
}}></Switch>
```



Table表格

参数	说明	类型	默认值	版本
columns	默认状态	boolean	false	1.0
dataSource	是否禁用	boolean	false	1.0
pagination	是否显示页码	IPagination		1.0
className	表格类名	string		1.0
rowClassName	行类名	string		1.0
stripe	是否显示样式,下边框，行背景，移入背景	boolean	true	1.0
scroll	是否溢出显示滚动条	{ x?: boolean, y?: boolean };	flase	1.0

列选项

```
interface IColumns {
  title: string,
  dataIndex: string,
  key: string,
  align?: string, 文字对齐
  className?: string, 类名
  ellipsis?: boolean, 溢出隐藏，需要配合宽度使用
  width?: number,列宽度
  render?: (item: string) => ReactNode 自定义渲染
```

```

}

let IColumns = [
  {
    title: '操作',
    dataIndex: 'action',
    key: 'action',
    className: "xs",
    ellipsis: true,
    width: 250,
    render: (items: any) => {
      return Array.isArray(items) ? items.map((item: any) => {
        return <Button size="sm" key={item} btnType="action">{item}</Button>
      }): <Button size="sm" btnType="action">{items}</Button>
    }
  },
]

const dataSource = [
  {
    key: '1',
    account: '1',
    action: ["兑换优惠券", "操作1", "操作2"],
    time: 'time',
    integralChange: '-1',
    integralValue: '1',
  },
]
//dataSource 数据中的key须与IColumns中的key的值相对应
<Table
  stripe={false}
  scroll={
    { x: true, y: false } 是否出现滚动条 可选，默认false
  }
  pagination={{
    total: 400, defaultPageSize: 50, onChange: (page, pageSize) => {
      let data = dataSource.filter((item, index) => {
        if ((index >= (page - 1) * pageSize) && (index < (page)
* pageSize))) {
          return true
        } else {
          return false
        }
      })
      console.log(data);
      setDataList(data)
    }
  }}
  columns={columns} dataSource={dataList}>
</Table>

```

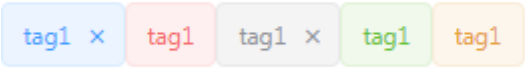

序号	操作	操作时间	积分变化	积分值
1	兑换优惠券	time	-1	1
2	32	西湖区湖底公园1号	西湖区湖底公园1号	西湖区湖底公园1号
2	32	西湖区湖底公园1号	西湖区湖底公园1号	西湖区湖底公园1号
2	32	西湖区湖底公园1号	西湖区湖底公园1号	西湖区湖底公园1号
2	32	西湖区湖底公园1号	西湖区湖底公园1号	西湖区湖底公园1号
2	32	西湖区湖底公园1号	西湖区湖底公园1号	西湖区湖底公园1号
2	32	西湖区湖底公园1号	西湖区湖底公园1号	西湖区湖底公园1号
2	32	西湖区湖底公园1号	西湖区湖底公园1号	西湖区湖底公园1号
2	32	西湖区湖底公园1号	西湖区湖底公园1号	西湖区湖底公园1号

每页显示数量：50 < 1 2 3 4 5 6 7 8 > 前往 1 页

Tabs标签

参数	说明	类型	默认值	版本
message	标签信息	string		1.0
type	标签类型	React.ReactNode		1.0
closable	是否显示关闭按钮	boolean	false	1.0
close	关闭回调	()=>void		1.0

```
<Tag closable>tag1</Tag>
<Tag type="danger">tag1</Tag>
<Tag type="info" closable>tag1</Tag>
<Tag type="success">tag1</Tag>
<Tag type="warning">tag1</Tag>
```



Textarea多行文本

参数	说明	类型	默认值	版本
value	标签信息	string		1.0
rows	行长度	number		1.0
cols	列长度	number	false	1.0
disabled	是否禁用	boolean	false	1.0
resize	是否可拉伸	boolean	true	1.0
onChangeInput	输入框改变的回调	(value:string)=>void		1.0

```
<Textarea resize={false} value="默认值" placeholder="请输入"></Textarea>
```

默认值

Radio单选框

参数	说明	类型	默认值	版本
buttonStyle	按钮类型 outline solid	buttonStyle	outline	1.0
defaultValue	默认传入的value值	any		1.0
value	value值,外部可通过此值进行动态改变选中状态	any		1.0
onChange	value改变事件，Item的value	(value: string) => void		1.0

```
<Radio defaultValue="child" onChange={(val:string)=>{
  console.log(val);
}}>
  <Radio.Item value={"child"}> child </Radio.Item>
  <Radio.Item value={"child1"}> child1 </Radio.Item>
  <Radio.Item value={"child2"}> child2 </Radio.Item>
</Radio>
```

- outline

childchild1child2

- solid

child	child1	child2
-------	--------	--------