

zh-CN

结合 [Tooltip](#) 组件，实现一个数值输入框，方便内容超长时的全量展现。

en-US

You can use the Input in conjunction with [Tooltip](#) component to create a Numeric Input, which can provide a good experience for extra-long content display.

```
import { Input, Tooltip } from 'antd';

function formatNumber(value) {
  value += '';
  const list = value.split('.');
  const prefix = list[0].charAt(0) === '-' ? '-' : '';
  let num = prefix ? list[0].slice(1) : list[0];
  let result = '';
  while (num.length > 3) {
    result = `,${num.slice(-3)}${result}`;
    num = num.slice(0, num.length - 3);
  }
  if (num) {
    result = num + result;
  }
  return `${prefix}${result}${list[1] ? `.${list[1]}` : ''}`;
}

class NumericInput extends React.Component {
  onChange = e => {
    const { value } = e.target;
    const reg = /^-?\d*(\.\d*)?$/;
    if ((!isNaN(value) && reg.test(value)) || value === '' || value === '-') {
      this.props.onChange(value);
    }
  };

  // '.' at the end or only '-' in the input box.
  onBlur = () => {
    const { value, onBlur, onChange } = this.props;
    let valueTemp = value;
    if (value.charAt(value.length - 1) === '.' || value === '-') {
      valueTemp = value.slice(0, -1);
    }
    onChange(valueTemp.replace(/0*(\d+)/, '$1'));
    if (onBlur) {
      onBlur();
    }
  };

  render() {
    const { value } = this.props;
```

```

    const title = value ? (
      <span className="numeric-input-title">{value !== '-' ? formatNumber(value) :
'-'}</span>
    ) : (
      'Input a number'
    );
    return (
      <Tooltip
        trigger={['focus']}
        title={title}
        placement="topLeft"
        overlayClassName="numeric-input"
      >
        <Input
          {...this.props}
          onChange={this.onChange}
          onBlur={this.onBlur}
          placeholder="Input a number"
          maxLength={25}
        />
      </Tooltip>
    );
  }
}

```

```

class NumericInputDemo extends React.Component {
  constructor(props) {
    super(props);
    this.state = { value: '' };
  }

  onChange = value => {
    this.setState({ value });
  };

  render() {
    return (
      <NumericInput style={{ width: 120 }} value={this.state.value} onChange=
{this.onChange} />
    );
  }
}

export default () => <NumericInputDemo />;

```

```

/* to prevent the arrow overflow the popup container,
or the height is not enough when content is empty */
.numeric-input .ant-tooltip-inner {
  min-width: 32px;
  min-height: 37px;
}

```

```
.numeric-input .numeric-input-title {  
  font-size: 14px;  
}
```