何时使用 {#when-to-use}

- 需要用户输入表单域内容时。
- 提供组合型输入框，带搜索的输入框，还可以进行大小选择。

代码演示

基本使用

```
import React from 'react';
import { Input } from 'antd';

const App: React.FC = () => <Input placeholder="Basic usage" />;

export default App;
```

三种大小

```
import React from 'react';
import { UserOutlined } from '@ant-design/icons';
import { Input } from 'antd';

const App: React.FC = () => (
  <>
    <Input size="large" placeholder="large size" prefix={<UserOutlined />}
/>
    <br />
    <br />
    <Input placeholder="default size" prefix={<UserOutlined />} />
    <br />
    <br />
    <Input size="small" placeholder="small size" prefix={<UserOutlined />}
/>
  </>
);

export default App;
```

形态变体

v5.13.0

```
import React from 'react';
import { Flex, Input } from 'antd';

const App: React.FC = () => (
  <Flex vertical gap={12}>
    <Input placeholder="Outlined" />
    <Input placeholder="Filled" variant="filled" />
    <Input placeholder="Borderless" variant="borderless" />
    <Input placeholder="Underlined" variant="underlined" />
  </Flex>
);

export default App;
```

面性变体 Debug

Debug

```
import React from 'react';
import { Flex, Input } from 'antd';

const { TextArea } = Input;

const App: React.FC = () => (
  <Flex vertical gap={20}>
    <Flex gap={12}>
      <Input placeholder="Filled" variant="filled" />
      <Input placeholder="Filled" variant="filled" disabled />
      <Input placeholder="Filled" variant="filled" status="error"
value="Filled Error" />
    </Flex>
    <Flex gap={12}>
      <Input prefix="$" placeholder="Filled" variant="filled" />
      <Input prefix="$" placeholder="Filled" variant="filled" disabled />
      <Input prefix="$" placeholder="Filled" variant="filled"
status="error" value="Filled Error" />
    </Flex>
    <Flex gap={12}>
      <Input addonBefore="http://" addonAfter=".com" placeholder="Filled"
variant="filled" />
      <Input
        addonBefore="http://"
        addonAfter=".com"
        placeholder="Filled"
        variant="filled"
        disabled
      />
      <Input
        addonBefore="http://"
        addonAfter=".com"
        placeholder="Filled"
        variant="filled"
        status="error"
        value="Filled Error"
      />
    </Flex>
    <Flex gap={12}>
      <Input addonAfter=".com" placeholder="Filled" variant="filled" />
      <Input addonAfter=".com" placeholder="Filled" variant="filled"
disabled />
      <Input
        addonAfter=".com"
        placeholder="Filled"
        variant="filled"
        status="error"
        value="Filled Error"
      />
    </Flex>
    <Flex gap={12}>
      <Input addonBefore="http://" placeholder="Filled" variant="filled" />
      <Input addonBefore="http://" placeholder="Filled" variant="filled"
disabled />
      <Input
        addonBefore="http://"
        placeholder="Filled"
        variant="filled"
        status="error"
        value="Filled Error"
      />
    </Flex>
    <TextArea variant="filled" placeholder="Basic" />
    <TextArea variant="filled" placeholder="Basic" status="error"
value="Filled Error" />
    <TextArea variant="filled" placeholder="Allow Clear" allowClear />
    <TextArea variant="filled" placeholder="Show Count" showCount />
    <TextArea
      variant="filled"
      placeholder="Show Count"
      showCount
      status="error"
      value="Filled Error"
    />
  </Flex>
);

export default App;
```

前置/后置标签

```
import React from 'react';
import { SettingOutlined } from '@ant-design/icons';
import { Cascader, Input, Select, Space } from 'antd';

const { Option } = Select;

const selectBefore = (
  <Select defaultValue="http://">
    <Option value="http://">http://</Option>
    <Option value="https://">https://</Option>
  </Select>
);
const selectAfter = (
  <Select defaultValue=".com">
    <Option value=".com">.com</Option>
    <Option value=".jp">.jp</Option>
    <Option value=".cn">.cn</Option>
    <Option value=".org">.org</Option>
  </Select>
);

const App: React.FC = () => (
  <Space direction="vertical">
    <Input addonBefore="http://" addonAfter=".com" defaultValue="mysite" />
    <Input addonBefore={selectBefore} addonAfter={selectAfter}
defaultValue="mysite" />
    <Input addonAfter={<SettingOutlined />} defaultValue="mysite" />
    <Input addonBefore="http://" suffix=".com" defaultValue="mysite" />
    <Input
      addonBefore={<Cascader placeholder="cascader" style={{ width: 150 }}
/>}
      defaultValue="mysite"
    />
  </Space>
);

export default App;
```

紧凑模式

```
import React from 'react';
import { SearchOutlined } from '@ant-design/icons';
import { Button, Input, Select, Space } from 'antd';
```

```
const { Search } = Input;

const options = [
  {
    value: 'zhejiang',
    label: 'Zhejiang',
  },
  {
    value: 'jiangsu',
    label: 'Jiangsu',
  },
];

const App: React.FC = () => (
  <Space direction="vertical" size="middle">
    <Space.Compact>
      <Input defaultValue="26888888" />
    </Space.Compact>
    <Space.Compact>
      <Input style={{ width: '20%' }} defaultValue="0571" />
      <Input style={{ width: '80%' }} defaultValue="26888888" />
    </Space.Compact>
    <Space.Compact>
      <Search addonBefore="https://" placeholder="input search text"
allowClear />
    </Space.Compact>
    <Space.Compact style={{ width: '100%' }}>
      <Input defaultValue="Combine input and button" />
      <Button type="primary">Submit</Button>
    </Space.Compact>
    <Space.Compact>
      <Select defaultValue="Zhejiang" options={options} />
      <Input defaultValue="Xihu District, Hangzhou" />
    </Space.Compact>
    <Space.Compact size="large">
      <Input addonBefore={<SearchOutlined />} placeholder="large size" />
      <Input placeholder="another input" />
    </Space.Compact>
  </Space>
);

export default App;
```

输入框组合

Debug

```
import React from 'react';
import { SearchOutlined } from '@ant-design/icons';
import {
  AutoComplete,
  Button,
  Cascader,
  Col,
  DatePicker,
  Input,
  InputNumber,
  Row,
  Select,
  Tooltip,
} from 'antd';

const { Option } = Select;

const options = [
  {
    value: 'zhejiang',
    label: 'Zhejiang',
    children: [
      {
        value: 'hangzhou',
        label: 'Hangzhou',
        children: [
          {
            value: 'xihu',
            label: 'West Lake',
          },
        ],
      },
    ],
  },
  {
    value: 'jiangsu',
    label: 'Jiangsu',
    children: [
      {
        value: 'nanjing',
        label: 'Nanjing',
        children: [
          {
            value: 'zhonghuamen',
            label: 'Zhong Hua Men',
          },
        ],
      },
    ],
  },
];

const App: React.FC = () => (
  <div className="site-input-group-wrapper">
    <Input.Group size="large">
      <Row gutter={8}>
        <Col span={5}>
          <Input defaultValue="0571" />
        </Col>
        <Col span={8}>
          <Input defaultValue="26888888" />
        </Col>
      </Row>
    </Input.Group>
    <br />
    <Input.Group compact>
      <Input style={{ width: '20%' }} defaultValue="0571" />
      <Input style={{ width: '30%' }} defaultValue="26888888" />
    </Input.Group>
    <br />
    <Input.Group compact>
      <Input style={{ width: 'calc(100% - 200px)' }}
defaultValue="https://ant.design" />
      <Button type="primary">Submit</Button>
    </Input.Group>
    <br />
    <Input.Group compact>
      <Input
        style={{ width: 'calc(100% - 200px)' }}
        defaultValue="git@github.com:ant-design/ant-design.git"
      />
      <Tooltip title="search git url">
        <Button icon={<SearchOutlined />} />
      </Tooltip>
    </Input.Group>
    <br />
    <Input.Group compact>
      <Select defaultValue="Zhejiang">
        <Option value="Zhejiang">Zhejiang</Option>
        <Option value="Jiangsu">Jiangsu</Option>
      </Select>
      <Input style={{ width: '50%' }} defaultValue="Xihu District, Hangzhou" />
    </Input.Group>
    <br />
    <Input.Group compact>
      <Input.Search allowClear style={{ width: '40%' }} defaultValue="0571" />
      <Input.Search allowClear style={{ width: '40%' }} defaultValue="26888888" />
    </Input.Group>
    <br />
    <Input.Group compact>
      <Select defaultValue="Option1">
        <Option value="Option1">Option1</Option>
        <Option value="Option2">Option2</Option>
      </Select>
      <Input style={{ width: '50%' }} defaultValue="input content" />
      <InputNumber prefix="@" />
    </Input.Group>
    <br />
    <Input.Group compact>
      <Input style={{ width: '50%' }} defaultValue="input content" />
      <DatePicker style={{ width: '50%' }} />
    </Input.Group>
    <br />
    <Input.Group compact>
      <Input style={{ width: '30%' }} defaultValue="input content" />
      <DatePicker.RangePicker style={{ width: '70%' }} />
    </Input.Group>
    <br />
    <Input.Group compact>
      <Select defaultValue="Option1-1">
        <Option value="Option1-1">Option1-1</Option>
        <Option value="Option1-2">Option1-2</Option>
      </Select>
      <Select defaultValue="Option2-2">
        <Option value="Option2-1">Option2-1</Option>
        <Option value="Option2-2">Option2-2</Option>
      </Select>
    </Input.Group>
    <br />
    <Input.Group compact>
      <Select defaultValue="1">
        <Option value="1">Between</Option>
        <Option value="2">Except</Option>
      </Select>
      <Input style={{ width: 100, textAlign: 'center' }}
placeholder="Minimum" />
      <Input
        className="site-input-split"
        style={{
          width: 30,
          borderLeft: 0,
          borderRight: 0,
          pointerEvents: 'none',
        }}
        placeholder="~"
        disabled
      />
      <Input
        className="site-input-right"
        style={{
          width: 100,
          textAlign: 'center',
        }}
        placeholder="Maximum"
      />
    </Input.Group>
    <br />
    <Input.Group compact>
      <Select defaultValue="Sign Up" style={{ width: '30%' }}>
        <Option value="Sign Up">Sign Up</Option>
        <Option value="Sign In">Sign In</Option>
      </Select>
      <AutoComplete
        style={{ width: '70%' }}
        placeholder="Email"
        options={[{ value: 'text 1' }, { value: 'text 2' }]}
      />
    </Input.Group>
    <br />
    <Input.Group compact>
      <Select style={{ width: '30%' }} defaultValue="Home">
        <Option value="Home">Home</Option>
        <Option value="Company">Company</Option>
      </Select>
      <Cascader style={{ width: '70%' }} options={options}
placeholder="Select Address" />
    </Input.Group>
  </div>
);
```

```
export default App;
```

搜索框

```
import React from 'react';
import { AudioOutlined } from '@ant-design/icons';
import { Input, Space } from 'antd';
import type { GetProps } from 'antd';

type SearchProps = GetProps<typeof Input.Search>;

const { Search } = Input;

const suffix = (
  <AudioOutlined
    style={{
      fontSize: 16,
      color: '#1677ff',
    }}
  />
);

const onSearch: SearchProps['onSearch'] = (value, _e, info) =>
console.log(info?.source, value);

const App: React.FC = () => (
  <Space direction="vertical">
    <Search placeholder="input search text" onSearch={onSearch} style={{
width: 200 }} />
    <Search placeholder="input search text" allowClear onSearch={onSearch}
style={{ width: 200 }} />
    <Search
      addonBefore="https://"
      placeholder="input search text"
      allowClear
      onSearch={onSearch}
      style={{ width: 304 }}
    />
    <Search placeholder="input search text" onSearch={onSearch} enterButton
/>
    <Search
      placeholder="input search text"
      allowClear
      enterButton="Search"
      size="large"
      onSearch={onSearch}
    />
    <Search
      placeholder="input search text"
      enterButton="Search"
      size="large"
      suffix={suffix}
      onSearch={onSearch}
    />
  </Space>
);

export default App;
```

搜索框 loading

```
import React from 'react';
import { Input } from 'antd';

const { Search } = Input;

const App: React.FC = () => (
  <>
    <Search placeholder="input search loading default" loading />
    <br />
    <br />
    <Search placeholder="input search loading with enterButton" loading
enterButton />
    <br />
    <br />
    <Search placeholder="input search text" enterButton="Search"
size="large" loading />
  </>
);

export default App;
```

文本域

```
import React from 'react';
import { Input } from 'antd';

const { TextArea } = Input;

const App: React.FC = () => (
  <>
    <TextArea rows={4} />
    <br />
    <br />
    <TextArea rows={4} placeholder="maxLength is 6" maxLength={6} />
  </>
);

export default App;
```

适应文本高度的文本域

```
import React, { useState } from 'react';
import { Input } from 'antd';

const { TextArea } = Input;

const App: React.FC = () => {
  const [value, setValue] = useState('');

  return (
    <>
      <TextArea placeholder="Autosize height based on content lines"
autoSize />
      <div style={{ margin: '24px 0' }} />
      <TextArea
        placeholder="Autosize height with minimum and maximum number of
lines"
        autoSize={{ minRows: 2, maxRows: 6 }}
      />
      <div style={{ margin: '24px 0' }} />
      <TextArea
        value={value}
        onChange={(e) => setValue(e.target.value)}
        placeholder="Controlled autosize"
        autoSize={{ minRows: 3, maxRows: 5 }}
      />
    </>
  );
};

export default App;
```

一次性密码框

v5.16.0

```
import React from 'react';
import { Flex, Input, Typography } from 'antd';
import type { GetProps } from 'antd';

type OTPProps = GetProps<typeof Input.OTP>;

const { Title } = Typography;

const App: React.FC = () => {
  const onChange: OTPProps['onChange'] = (text) => {
    console.log('onChange:', text);
  };

  const onInput: OTPProps['onInput'] = (value) => {
    console.log('onInput:', value);
  };

  const sharedProps: OTPProps = {
    onChange,
    onInput,
  };

  return (
    <Flex gap="middle" align="flex-start" vertical>
      <Title level={5}>With formatter (Upcase)</Title>
      <Input.OTP formatter={(str) => str.toUpperCase()} {...sharedProps} />
      <Title level={5}>With Disabled</Title>
      <Input.OTP disabled {...sharedProps} />
      <Title level={5}>With Length (8)</Title>
      <Input.OTP length={8} {...sharedProps} />
      <Title level={5}>With variant</Title>
      <Input.OTP variant="filled" {...sharedProps} />
      <Title level={5}>With custom display character</Title>
      <Input.OTP mask="🔒" {...sharedProps} />
      <Title level={5}>With custom ReactNode separator</Title>
      <Input.OTP separator={<span>/</span>} {...sharedProps} />
      <Title level={5}>With custom function separator</Title>
      <Input.OTP
        separator={(i) => <span style={{ color: i & 1 ? 'red' : 'blue' }}>—</span>}
        {...sharedProps}
      />
    </Flex>
  );
};
```

```
export default App;
```

输入时格式化展示

```
import React, { useState } from 'react';
import { Input, Tooltip } from 'antd';

interface NumericInputProps {
  style: React.CSSProperties;
  value: string;
  onChange: (value: string) => void;
}

const formatNumber = (value: number) => new
Intl.NumberFormat().format(value);

const NumericInput = (props: NumericInputProps) => {
  const { value, onChange } = props;

  const handleChange = (e: React.ChangeEvent<HTMLInputElement>) => {
    const { value: inputValue } = e.target;
    const reg = /^-?\d*(\.\d*)?$/;
    if (reg.test(inputValue) || inputValue === '' || inputValue === '-') {
      onChange(inputValue);
    }
  };

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

  const title = value ? (
    <span className="numeric-input-title">{value !== '-' ?
formatNumber(Number(value)) : '-'}</span>
  ) : (
    'Input a number'
  );

  return (
    <Tooltip
```

```
      trigger={['focus']}
      title={title}
      placement="topLeft"
      classNames={{ root: 'numeric-input' }}
    >
      <Input
        {...props}
        onChange={handleChange}
        onBlur={handleBlur}
        placeholder="Input a number"
        maxLength={16}
      />
    </Tooltip>
  );
};

const App: React.FC = () => {
  const [value, setValue] = useState('');

  return <NumericInput style={{ width: 120 }} value={value} onChange={setValue} />;
};

export default App;
```

前缀和后缀

```
import React from 'react';
import { InfoCircleOutlined, UserOutlined } from '@ant-design/icons';
import { Input, Tooltip } from 'antd';

const App: React.FC = () => (
  <>
    <Input
      placeholder="Enter your username"
      prefix={<UserOutlined style={{ color: 'rgba(0,0,0,.25)' }} />}
      suffix={
        <Tooltip title="Extra information">
          <InfoCircleOutlined style={{ color: 'rgba(0,0,0,.45)' }} />
        </Tooltip>
      }
    />
    <br />
    <br />
    <Input prefix="￥" suffix="RMB" />
    <br />
    <br />
    <Input prefix="￥" suffix="RMB" disabled />
  </>
);

export default App;
```

密码框

```
import React from 'react';
import { EyeInvisibleOutlined, EyeTwoTone } from '@ant-design/icons';
import { Button, Input, Space } from 'antd';

const App: React.FC = () => {
  const [passwordVisible, setPasswordVisible] = React.useState(false);

  return (
    <Space direction="vertical">
      <Input.Password placeholder="input password" />
      <Input.Password
        placeholder="input password"
        iconRender={(visible) => (visible ? <EyeTwoTone /> :
<EyeInvisibleOutlined />)}
      />
      <Space direction="horizontal">
        <Input.Password
          placeholder="input password"
          visibilityToggle={{ visible: passwordVisible, onVisibleChange:
setPasswordVisible }}
        />
        <Button style={{ width: 80 }} onClick={() =>
setPasswordVisible((prevState) => !prevState)}>
          {passwordVisible ? 'Hide' : 'Show'}
        </Button>
      </Space>
      <Input.Password disabled placeholder="disabled input password" />
    </Space>
  );
};

export default App;
```

带移除图标

```
import React from 'react';
import { Input } from 'antd';

const { TextArea } = Input;

const onChange = (e: React.ChangeEvent<HTMLInputElement |
HTMLTextAreaElement>) => {
  console.log(e);
};

const App: React.FC = () => (
  <>
    <Input placeholder="input with clear icon" allowClear onChange=
{onChange} />
    <br />
    <br />
    <TextArea placeholder="textarea with clear icon" allowClear onChange=
{onChange} />
  </>
);

export default App;
```

带字数提示

```
import React from 'react';
import { Flex, Input } from 'antd';

const { TextArea } = Input;

const onChange = (e: React.ChangeEvent<HTMLInputElement |
HTMLTextAreaElement>) => {
  console.log('Change:', e.target.value);
};

const App: React.FC = () => (
  <Flex vertical gap={32}>
    <Input showCount maxLength={20} onChange={onChange} />
    <TextArea showCount maxLength={100} onChange={onChange}
placeholder="can resize" />
    <TextArea
      showCount
      maxLength={100}
      onChange={onChange}
      placeholder="disable resize"
      style={{ height: 120, resize: 'none' }}
    />
  </Flex>
);

export default App;
```

= 5.10.0">定制计数能力

```
import React from 'react';
import { Flex, Input, Typography } from 'antd';
import { runes } from 'runes2';

const App: React.FC = () => (
  <Flex vertical gap={16}>
    <div>
      <Typography.Title level={5}>Exceed Max</Typography.Title>
      <Input
        count={{
          show: true,
          max: 10,
        }}
        defaultValue="Hello, antd!"
      />
    </div>

    <div>
      <Typography.Title level={5}>Emoji count as length
1</Typography.Title>
      <Input
        count={{
          show: true,
          strategy: (txt) => runes(txt).length,
        }}
        defaultValue="🔥🔥🔥"
      />
    </div>

    <div>
      <Typography.Title level={5}>Not exceed max</Typography.Title>
      <Input
        count={{
          show: true,
          max: 6,
          strategy: (txt) => runes(txt).length,
          exceedFormatter: (txt, { max }) => runes(txt).slice(0,
max).join(''),
        }}
        defaultValue="🔥 antd"
      />
    </div>
  </Flex>
);

export default App;
```

自定义状态

```
import React from 'react';
import ClockCircleOutlined from '@ant-design/icons/ClockCircleOutlined';
import { Input, Space } from 'antd';

const App: React.FC = () => (
  <Space direction="vertical" style={{ width: '100%' }}>
    <Input status="error" placeholder="Error" />
    <Input status="warning" placeholder="Warning" />
    <Input status="error" prefix={<ClockCircleOutlined />}
placeholder="Error with prefix" />
    <Input status="warning" prefix={<ClockCircleOutlined />}
placeholder="Warning with prefix" />
  </Space>
);

export default App;
```

聚焦

```
import React, { useRef, useState } from 'react';
import type { InputRef } from 'antd';
import { Button, Input, Space, Switch } from 'antd';

const App: React.FC = () => {
  const inputRef = useRef<InputRef>(null);
  const [input, setInput] = useState(true);

  const sharedProps = {
    style: { width: '100%' },
    defaultValue: 'Ant Design love you!',
    ref: inputRef,
  };
  return (
    <Space direction="vertical" style={{ width: '100%' }}>
      <Space wrap>
        <Button
          onClick={() => {
            inputRef.current!.focus({
              cursor: 'start',
            });
          }}
        >
          Focus at first
        </Button>
        <Button
          onClick={() => {
            inputRef.current!.focus({
              cursor: 'end',
            });
          }}
        >
          Focus at last
        </Button>
        <Button
          onClick={() => {
            inputRef.current!.focus({
              cursor: 'all',
            });
          }}
        >
          Focus to select all
        </Button>
        <Button
          onClick={() => {
            inputRef.current!.focus({
              preventScroll: true,
            });
          }}
        >
          Focus prevent scroll
        </Button>
        <Switch
          checked={input}
          checkedChildren="Input"
          unCheckedChildren="TextArea"
          onChange={() => {
            setInput(!input);
          }}
        />
      </Space>
      <br />
      {input ? <Input {...sharedProps} /> : <Input.TextArea
{...sharedProps} />}
    </Space>
  );
};

export default App;
```

Style Debug

Debug

```
import React from 'react';
import { Input } from 'antd';

const { TextArea } = Input;

const App: React.FC = () => (
  <div style={{ backgroundColor: 'rgba(0, 0, 128, .2)' }}>
    <Input placeholder="Unbordered" variant="borderless" />
    <Input placeholder="Unbordered" variant="borderless" size="large" />
    <TextArea placeholder="Unbordered" variant="borderless" />
    <TextArea placeholder="Unbordered" variant="borderless" allowClear />
    <Input placeholder="Unbordered" variant="borderless" allowClear />
    <Input prefix="￥" suffix="RMB" variant="borderless" />
    <Input prefix="￥" suffix="RMB" disabled variant="borderless" />
    <TextArea allowClear style={{ border: '2px solid #000' }} />

    {/* status */}
    <Input defaultValue="error" variant="borderless" status="error" />
    <Input defaultValue="warning" variant="borderless" status="warning" />
    <Input prefix="$" defaultValue="error" variant="borderless"
status="error" />
    <Input prefix="$" defaultValue="warning" variant="borderless"
status="warning" />
  </div>
);

export default App;
```

文本对齐

Debug

```
import React from 'react';
import {
  AutoComplete,
  Button,
  Cascader,
  DatePicker,
  Input,
  InputNumber,
  Mentions,
  Radio,
  Select,
  TimePicker,
  TreeSelect,
  Typography,
} from 'antd';

const { Text } = Typography;
const { RangePicker } = DatePicker;

const narrowStyle: React.CSSProperties = {
  width: 50,
};

const options = [
  {
    value: 'zhejiang',
    label: 'Zhejiang',
    children: [
      {
        value: 'hangzhou',
        label: 'Hangzhou',
        children: [
          {
            value: 'xihu',
            label: 'West Lake',
          },
        ],
      },
    ],
  },
  {
    value: 'jiangsu',
    label: 'Jiangsu',
    children: [
      {
        value: 'nanjing',
        label: 'Nanjing',
        children: [
          {
            value: 'zhonghuamen',
            label: 'Zhong Hua Men',
          },
        ],
      },
    ],
  },
];

const selectOptions = [
  { value: 'jack', label: 'Jack' },
  { value: 'lucy', label: 'Lucy' },
];

const App: React.FC = () => (
  <>
    <Mentions style={{ width: 100 }} rows={1} />
    <Input.TextArea rows={1} style={{ width: 100 }} />
    <Button type="primary">Button</Button>
    <Input style={{ width: 100 }} />
    <Text copyable>Ant Design</Text>
    <Input prefix="1" suffix="2" style={{ width: 100 }} />
    <Input addonBefore="1" addonAfter="2" style={{ width: 100 }} />
    <InputNumber style={{ width: 100 }} />
    <DatePicker style={{ width: 100 }} />
    <TimePicker style={{ width: 100 }} />
    <Select style={{ width: 100 }} defaultValue="jack" options=
{selectOptions} />
    <Select style={{ width: 100 }} defaultValue="" options={selectOptions}
/>
    <Select style={{ width: 100 }} options={selectOptions} />
    <TreeSelect style={{ width: 100 }} />
    <Cascader defaultValue={['zhejiang', 'hangzhou', 'xihu']} options=
{options} />
    <RangePicker />
    <DatePicker picker="month" />
    <Radio.Group defaultValue="a">
      <Radio.Button value="a">Hangzhou</Radio.Button>
      <Radio.Button value="b">Shanghai</Radio.Button>
    </Radio.Group>
    <AutoComplete style={{ width: 100 }} placeholder="input here" />
    <br />
    <Input prefix="$" addonBefore="Http://" addonAfter=".com"
```

```
defaultValue="mysite" />
    <Input style={narrowStyle} suffix="Y" />
    <Input style={narrowStyle} />
    <Input style={narrowStyle} defaultValue="1" suffix="Y" />
  </>
);

export default App;
```

文本域

Debug

```
import React, { useState } from 'react';
import { Button, Input } from 'antd';

const { TextArea } = Input;

const defaultValue =
  'The autoSize property applies to textarea nodes, and only the height
changes automatically. In addition, autoSize can be set to an object,
specifying the minimum number of rows and the maximum number of rows. The
autoSize property applies to textarea nodes, and only the height changes
automatically. In addition, autoSize can be set to an object, specifying
the minimum number of rows and the maximum number of rows.';

const App: React.FC = () => {
  const [autoResize, setAutoResize] = useState(false);

  return (
    <>
      <Button onClick={() => setAutoResize(!autoResize)} style={{
marginBottom: 16 }}>
        Auto Resize: {String(autoResize)}
      </Button>
      <TextArea rows={4} autoSize={autoResize} defaultValue={defaultValue}
/>
      <TextArea allowClear style={{ width: 93 }} />
      <br />
      <TextArea
        style={{
          resize: 'both',
        }}
        showCount
      />
    </>
  );
};

export default App;
```

debug 前置/后置标签

Debug

```
import React from 'react';
import { SettingOutlined } from '@ant-design/icons';
import { Button, Input, Space } from 'antd';

const App: React.FC = () => (
  <Space direction="vertical">
    Input addon Button:
    <Input addonAfter={<Button type="primary">Submit</Button>}
defaultValue="mysite" />
    <Input addonAfter={<Button>Submit</Button>} defaultValue="mysite" />
    <br />
    <br />
    Input addon Button icon:
    <Input
      addonAfter={
        <Button>
          <SettingOutlined />
        </Button>
      }
      defaultValue="mysite"
    />
  </Space>
);

export default App;
```

debug token

Debug

```
import React from 'react';
import { UserOutlined } from '@ant-design/icons';
import { ConfigProvider, Input } from 'antd';

const App: React.FC = () => (
  <>
    <ConfigProvider theme={{ token: { controlHeight: 28 } }}>
```

```
      <Input placeholder="Basic usage" />
    </ConfigProvider>
    <ConfigProvider
      componentSize="small"
      theme={{ token: {}, components: { Input: { inputFontSizeSM: 12 } } }}
    >
      <Input placeholder="Basic usage" />
    </ConfigProvider>
    <ConfigProvider theme={{ components: { Input: { inputFontSize: 10 } }
}}>
      <Input placeholder="With prefix" prefix={<UserOutlined />} />
    </ConfigProvider>
  </>
);

export default App;
```

API

通用属性参考：通用属性

Input

参数	说明	类型	默认值	版本
addonAfter	带标签的 input，设置后置标签	ReactNode	-	
addonBefore	带标签的 input，设置前置标签	ReactNode	-	
allowClear	可以点击清除图标删除内容	boolean | { clearIcon: ReactNode }	-	
~~bordered~~	是否有边框, 请使用 `variant` 替换	boolean	true	4.5.0
classNames	语义化结构 class	Record<SemanticDOM, string>	-	5.4.0
count	字符计数配置	CountConfig	-	5.10.0
defaultValue	输入框默认内容	string	-	
disabled	是否禁用状态，默认为 false	boolean	false	
id	输入框的 id	string	-	
maxLength	最大长度	number	-	

prefix	带有前缀图标的 input	ReactNode	-	
showCount	是否展示字数	boolean | { formatter: (info: { value: string, count: number, maxLength?: number }) => ReactNode }	false	4.18.0 info.value: 4.23.0
status	设置校验状态	'error' | 'warning'	-	4.19.0
styles	语义化结构 style	Record<[SemanticDOM](), CSSProperties>	-	5.4.0
size	控件大小。注：标准表单内的输入框大小限制为 `middle`	`large` | `middle` | `small`	-	
suffix	带有后缀图标的 input	ReactNode	-	
type	声明 input 类型，同原生 input 标签的 type 属性，见：[MDN]()(请直接使用 `Input.TextArea` 代替 `type="textarea"`)	string	`text`	
value	输入框内容	string	-	
variant	形态变体	`outlined` | `borderless` | `filled` | `underlined`	`outlined`	5.13.0 | `underlined`: 5.24.0
onChange	输入框内容变化时的回调	function(e)	-	
onPressEnter	按下回车的回调	function(e)	-	
onClear	按下清除按钮的回调	() => void	-	5.20.0

> *如果 `Input` 在 `Form.Item` 内，并且 `Form.Item` 设置了 `id` 属性，则 `value` `defaultValue` 和 `id` 属性会被自动设置。*

Input 的其他属性和 React 自带的 [input]() 一致。

CountConfig

```
interface CountConfig {
  // 最大字符数，不同于原生 `maxLength`，超出后标红但不会截断
  max?: number;
  // 自定义字符计数，例如标准 emoji 长度大于 1，可以自定义计数策略将其改为 1
  strategy?: (value: string) => number;
  // 同 `showCount`
  show?: boolean | ((args: { value: string; count: number; maxLength?:
number }) => ReactNode);
  // 当字符数超出 `count.max` 时的自定义裁剪逻辑，不配置时不进行裁剪
  exceedFormatter?: (value: string, config: { max: number }) => string;
}
```

Input.TextArea

同 Input 属性，外加：

参数	说明	类型	默认值	版本
autoSize	自适应内容高度，可设置为 true | false 或对象：{ minRows: 2, maxRows: 6 }	boolean | object	false	
classNames	语义化结构 class	Record<SemanticDOM, string>	-	5.4.0
styles	语义化结构 style	Record<SemanticDOM, CSSProperties>	-	5.4.0

`Input.TextArea` 的其他属性和浏览器自带的 textarea 一致。

Input.Search

参数	说明	类型	默认值
enterButton	是否有确认按钮，可设为按钮文字。该属性会与 `addonAfter` 冲突。	ReactNode	false
loading	搜索 loading	boolean	false
onSearch	点击搜索图标、清除图标，或按下回车键时的回调	function(value, event, { source: "input" | "clear" })	-

其余属性和 Input 一致。

Input.Password

参数	说明	类型	默认值	版本

iconRender	自定义切换按钮	(visible) => ReactNode	(visible) => (visible ? <EyeOutlined /> : <EyeInvisibleOutlined />)	4.3.0
visibilityToggle	是否显示切换按钮或者控制密码显隐	boolean | [VisibilityToggle]()	true	

Input.OTP

`5.16.0` 新增。

> *开发者注意事项：*
>
> *当 `mask` 属性的类型为 string 时，我们强烈推荐接收单个字符或单个 emoji，如果传入多个字符或多个 emoji，则会在控制台抛出警告。*

参数	说明	类型	默认值	版本
defaultValue	默认值	string	-	
disabled	是否禁用	boolean	false	
formatter	格式化展示，留空字段会被 填充	(value: string) => string	-	
separator	分隔符，在指定索引的输入框后渲染分隔符	ReactNode |((i: number) => ReactNode)	-	5.24.0
mask	自定义展示，和 `formatter` 的区别是不会修改原始值	boolean | string	`false`	`5.17.0`
length	输入元素数量	number	6	
status	设置校验状态	'error' | 'warning'	-	
size	输入框大小	`small` | `middle` | `large`	`middle`	
variant	形态变体	`outlined` | `borderless` | `filled` | `underlined`	`outlined`	`underlined`: 5.24.0
value	输入框内容	string	-	
onChange	当输入框内容全部填充时触发回调	(value: string) => void	-	

onInput	输入值变化时触发的回调	(value: string[]) => void	-	`5.22.0`

VisibilityToggle

Property	Description	Type	Default	Version
visible	用于手动控制密码显隐	boolean	false	4.24
onVisibleChange	显隐密码的回调	(visible) => void	-	4.24

Input Methods

名称	说明	参数	版本
blur	取消焦点	-	
focus	获取焦点	(option?: { preventScroll?: boolean, cursor?: 'start' | 'end' | 'all' })	option - 4.10.0

Semantic DOM

Input

演示

```
import React from 'react';
import { EditOutlined, UserOutlined } from '@ant-design/icons';
import { Input } from 'antd';

import SemanticPreview from '../../../.dumi/components/SemanticPreview';
import useLocale from '../../../.dumi/hooks/useLocale';

const locales = {
  cn: {
    input: '输入框元素',
    prefix: '前缀的包裹元素',
    suffix: '后缀的包裹元素',
    count: '文字计数元素',
  },
  en: {
    input: 'input element',
    prefix: 'prefix element',
    suffix: 'suffix element',
    count: 'count element',
  },
};
```

```
const App: React.FC = () => {
  const [locale] = useLocale(locales);
  return (
    <SemanticPreview
      componentName="Input"
      semantics={[
        { name: 'input', desc: locale.input, version: '5.4.0' },
        { name: 'prefix', desc: locale.prefix, version: '5.4.0' },
        { name: 'suffix', desc: locale.suffix, version: '5.4.0' },
        { name: 'count', desc: locale.count, version: '5.4.0' },
      ]}
    >
      <Input
        showCount
        prefix={<UserOutlined />}
        suffix={<EditOutlined />}
        defaultValue="Hello, Ant Design"
      />
    </SemanticPreview>
  );
};

export default App;
```

Input.TextArea

演示

```
import React from 'react';
import { Input } from 'antd';

import SemanticPreview from '../../../.dumi/components/SemanticPreview';
import useLocale from '../../../.dumi/hooks/useLocale';

const locales = {
  cn: {
    textarea: '输入框元素',
    count: '文字计数元素',
  },
  en: {
    textarea: 'textarea element',
    count: 'count element',
  },
};
```

```
const App: React.FC = () => {
  const [locale] = useLocale(locales);
  return (
    <SemanticPreview
      componentName="TextArea"
      semantics={[
        { name: 'textarea', desc: locale.textarea, version: '5.4.0' },
        { name: 'count', desc: locale.count, version: '5.4.0' },
      ]}
    >
      <Input.TextArea defaultValue="Hello, Ant Design" rows={3} count={{
max: 100, show: true }} />
    </SemanticPreview>
  );
};

export default App;
```

主题变量（Design Token）

FAQ

为什么我动态改变 `prefix/suffix/showCount` 时，Input 会失去焦点?

当 Input 动态添加或者删除 `prefix/suffix/showCount` 时，React 会重新创建 DOM 结构而新的 input 是没有焦点的。你可以预设一个空的 `<span />` 来保持 DOM 结构不变：

```
const suffix = condition ? <Icon type="smile" /> : <span />;

<Input suffix={suffix} />;
```

为何 TextArea 受控时， `value` 可以超过 `maxLength` ？

受控时，组件应该按照受控内容展示。以防止在表单组件内使用时显示值和提交值不同的问题。