When To Use

- A user input in a form field is needed.
- A search input is required.

Examples

Basic usage

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

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

export default App;
```

Three sizes of Input

```
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;
```

Variants

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;
```

Filled 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;
```

Pre / Post tab

```
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;
```

Compact Style

```
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;
```

Input Group

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;
```

Search box

```
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;
```

Search box with 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;
```

TextArea

```
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;
```

Autosizing the height to fit the content

```
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;
```

OTP

v5.16.0

```tsx
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;
```

Format Tooltip Input

```
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;
```

prefix and suffix

```
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;
```

Password box

```
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;
```

With clear icon

```
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;
```

With character counting

```
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">Custom count logic

```
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;
```

Status

```
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;
```

Focus

```
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;
```

Text Align

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;
```

TextArea

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 Pre / Post tab

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

Common props ref： Common props

Input

Property	Description	Type	Default	Version
addonAfter	The label text displayed after (on the right side of) the input field	ReactNode	-	
addonBefore	The label text displayed before (on the left side of) the input field	ReactNode	-	
allowClear	If allow to remove input content with clear icon	boolean | { clearIcon: ReactNode }	false	
~~bordered~~	Whether has border style, please use `variant` instead	boolean	true	4.5.0
classNames	Semantic DOM class	Record<SemanticDOM, string>	-	5.4.0

count	Character count config	CountConfig	-	5.10.0
defaultValue	The initial input content	string	-	
disabled	Whether the input is disabled	boolean	false	
id	The ID for input	string	-	
maxLength	The maximum number of characters in Input	number	-	
prefix	The prefix icon for the Input	ReactNode	-	
showCount	Whether to show character count	boolean | { formatter: (info: { value: string, count: number, maxLength?: number }) => ReactNode }	false	4.18.0 info.value: 4.23.0
status	Set validation status	'error' | 'warning'	-	4.19.0
styles	Semantic DOM style	Record<SemanticDOM, CSSProperties>	-	5.4.0
size	The size of the input box. Note: in the context of a form, the `middle` size is used	`large` | `middle` | `small`	-	
suffix	The suffix icon for the Input	ReactNode	-	
type	The type of input, see: MDN(use `Input.TextArea` instead of `type="textarea"`)	string	`text`	
value	The input content value	string	-	

variant	Variants of Input	`outlined` | `borderless` | `filled` | `underlined`	`outlined`	5.13.0 | `underlined`: 5.24.0
onChange	Callback when user input	function(e)	-	
onPressEnter	The callback function that is triggered when Enter key is pressed	function(e)	-	
onClear	Callback when click the clear button	() => void	-	5.20.0

> *When `Input` is used in a `Form.Item` context, if the `Form.Item` has the `id` props defined then `value`, `defaultValue`, and `id` props of `Input` are automatically set.*

The rest of the props of Input are exactly the same as the original input.

CountConfig

```
interface CountConfig {
  // Max character count. Different from the native `maxLength`, it will be marked warning but not truncated
  max?: number;
  // Custom character count, for example, the standard emoji length is greater than 1, you can customize the counting strategy to change it to 1
  strategy?: (value: string) => number;
  // Same as `showCount`
  show?: boolean | ((args: { value: string; count: number; maxLength?: number }) => ReactNode);
  // Custom clipping logic when the number of characters exceeds `count.max`, no clipping when not configured
  exceedFormatter?: (value: string, config: { max: number }) => string;
}
```

Input.TextArea

Same as Input, and more:

Property	Description	Type	Default	Version
autoSize	Height auto size feature, can be set to true | false or an object { minRows: 2, maxRows: 6 }	boolean | object	false	

classNames	Semantic DOM class	Record<SemanticDOM, string>	-	5.4.0
styles	Semantic DOM style	Record<SemanticDOM, CSSProperties>	-	5.4.0

The rest of the props of `Input.TextArea` are the same as the original textarea.

Input.Search

Property	Description	Type	Default
enterButton	false displays the default button color, true uses the primary color, or you can provide a custom button. Conflicts with addonAfter.	ReactNode	false
loading	Search box with loading	boolean	false
onSearch	The callback function triggered when you click on the search-icon, the clear-icon or press the Enter key	function(value, event, { source: "input" | "clear" })	-

Supports all props of `Input`.

Input.Password

Property	Description	Type	Default	Version
iconRender	Custom toggle button	(visible) => ReactNode	(visible) => (visible ? <EyeOutlined /> : <EyeInvisibleOutlined />)	4.3.0
visibilityToggle	Whether show toggle button or control password visible	boolean | VisibilityToggle	true	

Input.OTP

Added in `5.16.0`.

> *Notes for developers*
>
> *When the `mask` prop is string, we recommend receiving a single character or a single emoji. If multiple characters or multiple emoji are passed, a warning will be thrown.*

Property	Description	Type	Default	Version
defaultValue	Default value	string	-	

disabled	Whether the input is disabled	boolean	false	
formatter	Format display, blank fields will be filled with	(value: string) => string	-	
separator	render the separator after the input box of the specified index	ReactNode |((i: number) => ReactNode)	-	5.24.0
mask	Custom display, the original value will not be modified	boolean | string	`false`	`5.17.0`
length	The number of input elements	number	6	
status	Set validation status	'error' | 'warning'	-	
size	The size of the input box	`small` | `middle` | `large`	`middle`	
variant	Variants of Input	`outlined` | `borderless` | `filled` | `underlined`	`outlined`	`underlined`: 5.24.0
value	The input content value	string	-	
onChange	Trigger when all the fields are filled	(value: string) => void	-	
onInput	Trigger when the input value changes	(value: string[]) => void	-	`5.22.0`

VisibilityToggle

Property	Description	Type	Default	Version
visible	Whether the password is show or hide	boolean	false	4.24.0
onVisibleChange	Callback executed when visibility of the password is changed	(visible) => void	-	4.24.0

Input Methods

Name	Description	Parameters	Version

blur	Remove focus	-	
focus	Get focus	(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;
```

Design Token

FAQ

Why Input lose focus when change `prefix/suffix/showCount`

When Input dynamic add or remove `prefix/suffix/showCount` will make React recreate the dom structure and new input will be not focused. You can set an empty `<span />` element to keep the dom structure:

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

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

Why TextArea in control can make `value` exceed `maxLength` ?

When in control, component should show as what it set to avoid submit value not align with store value in Form.