

# Watermark

Add specific text or patterns to the page.

## When To Use

- Use when the page needs to be watermarked to identify the copyright.
- Suitable for preventing information theft.

## Examples

## Basic

The most basic usage.

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

const App: React.FC = () => (
  <Watermark content="Ant Design">
    <div style={{ height: 500 }} />
  </Watermark>
);

export default App;
```

### Multi-line watermark

Use 'content' to set a string array to specify multi-line text watermark content.

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

const App: React.FC = () => (
  <Watermark content={['Ant Design', 'Happy Working']}>
    <div style={{ height: 500 }} />
  </Watermark>
);

export default App;
```

### Image watermark

Specify the image address via 'image'. To ensure that the image is high definition and not stretched, set the width and height, and upload at least twice the width and height of the logo image address.

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

const App: React.FC = () => (
  <Watermark
    height={30}
    width={130}
    image="https://mdn.alipayobjects.com/huamei_7uahnr/afts/img/A*lkAoRbywo0oAAAAAAAAAAAAADrJ8AQ
  >
    <div style={{ height: 500 }} />
  </Watermark>
);

export default App;
```

The light-speed iteration of the digital world makes products more complex. However, human consciousness and attention resources are limited. Facing this design contradiction, the pursuit of natural interaction will be the consistent direction of Ant Design.

**Natural user cognition:** According to cognitive psychology, about 80% of external information is obtained through visual channels. The most important visual elements in the interface design, including layout, colors, illustrations, icons, etc., should fully absorb the laws of nature, thereby reducing the user's cognitive cost and bringing authentic and smooth feelings. In some scenarios, opportunely adding other sensory channels such as hearing, touch can create a richer and more natural product experience.

**Natural user behavior:** In the interaction with the system, the designer should fully understand the relationship between users, system roles, and task objectives, and also contextually organize system functions and services. At the same time, a series of methods such as behavior analysis, artificial intelligence and sensors could be applied to assist users to make effective decisions and reduce extra operations of users, to save users' mental and physical resources and make human-computer interaction more natural.



## Content

Ant Design

Color

FontSize

zIndex

Rotate

Gap

100

100

Offset

offsetLeft

offsetTop

## Custom configuration [✎](#)

Preview the watermark effect by configuring custom parameters.

```
import React, { useMemo, useState } from 'react';
import { Watermark, Popover, Typography, Form, Input, Slider, Space, InputNumber } from 'antd';
import { SketchPicker } from 'react-color';
import type { RGBColor } from 'react-color';
```

```
const { Paragraph } = Typography;
```

```
interface ColorPickerProps {
  value?: RGBColor;
  onChange?: (value: RGBColor) => void;
}
```

```
const ColorPicker: React.FC<ColorPickerProps> = ({ value, onChange }) => {
  const switchStyle = {
    padding: 4,
    background: '#fff',
    borderRadius: 2,
    border: '1px solid #dedede',
    display: 'inline-block',
    cursor: 'pointer',
  };
};
```

```
const colorStyle = {
  width: 36,
  height: 14,
  borderRadius: 2,
  background: `rgba(${value?.r}, ${value?.g}, ${value?.b}, ${value?.a})`,
};
```

```
};

return (
  <Popover
    trigger="click"
    placement="bottomLeft"
    overlayInnerStyle={{ padding: 0 }}
    content={<SketchPicker color={value} onChange={({color}) => onChange?.(color.rgb)} />}
  />
);
```

Property	Description	Type	Default	Version
<pre>width &lt;/div&gt; &lt;/Popover&gt; ); };</pre>	The width of the watermark, the default value of <code>content</code> is its own width	number	120	
<pre>const App: React.FC = () =&gt; {   const [form] = Form.useForm();   const [config, setConfig] = useState({     content: 'Ant Design',     color: { r: 0, g: 0, b: 0, a: 0.15 },     rotate: 0,     fontSize: 16,     zIndex: 11,     rotate: -22,     gap: [100, 100],     offset: undefined,   });   const { content, color, fontSize, zIndex, rotate, gap, offset } = config;   const watermarkProps = {     content,     color,     fontSize,     zIndex,     rotate,     gap,     offset,   };   return (     &lt;div style={{ gap: 20, padding: 10 }}&gt;       &lt;Watermark {...watermarkProps}&gt;         &lt;Typography&gt;           &lt;Paragraph&gt;             The light-speed iteration of the digital world makes products more complex. However, human consciousness and attention resources are limited. Facing this design           &lt;/Paragraph&gt;           &lt;Paragraph&gt;             Natural user cognition: According to cognitive psychology, about 80% of external information is obtained through visual channels. The most important visual elements of the interface design, including layout, colors, illustrations, icons, etc., should follow the laws of nature, thereby reducing the user's cognitive cost and bring authentic and smooth feelings. In some cases, opportunely adding other sensory channels such as hearing, touch can create a richer and more natural product experience.           &lt;/Paragraph&gt;           &lt;Paragraph&gt;             Natural user behavior: In the interaction with the system, the designer should fully understand the relationship between users, system roles, and task objectives, and contextually organize system functions and services. At the same time, a series of methods such as behavior analysis, artificial intelligence and sensors could be applied to assist users to make effective decisions and reduce extra operations of users, to save users' mental and physical resources and make human-computer interaction more natural.           &lt;/Paragraph&gt;           &lt;Typography&gt;             &lt;img alt="Watermark logo" data-bbox="100 100 200 200" style={{ width: 100px; height: 100px; }} /&gt;           &lt;/Typography&gt;         &lt;/Watermark&gt;       &lt;/div&gt;     );   }; };</pre>	The height of the watermark, the default value of <code>content</code> is its own height	number	64	
<pre>rotate</pre>	When the watermark is drawn, the rotation Angle, unit $^{\circ}$	number	-22	
<pre>zIndex</pre>	The z-index of the appended watermark element	number	9	
<pre>color</pre>	recommended to export 2x or 3x image, use <code>rgba</code> (priority (support base64 format))	string	-	
<pre>fontSize</pre>	Text style	Font	Font	
<pre>gap</pre>	The spacing between watermarks	[number, number]	[100, 100]	
<pre>offset</pre>	The offset of the watermark from the upper left corner of the container. The default is	[number, number]	[gap[0]/2, gap[1]/2]	
<pre>color</pre>	Font color	string	rgba(0,0,0,.15)	
<pre>fontSize</pre>	font size	number	16	
<pre>fontWeight</pre>	font weight	normal   light   weight	normal	
<pre>fontFamily</pre>	font family	string	sans-serif	
<pre>fontStyle</pre>	font style	normal   italic   oblique	normal	

```

      zIndex: 10,
      width: '100%',
      maxWidth: 800,
      position: 'relative',
    }}
    src="https://gw.alipayobjects.com/mdn/rms_08e378/afts/img/A*zx7LTI_ECSAAAAAAAAAAAAABkARl
    alt="示例图片"
  />
</Watermark>
<Form
  style={{
    width: 280,
    flexShrink: 0,

<Watermark
  height={30}
  width={130}
  content="Ant Design"
  image="https://mdn.alipayobjects.com/huamei_7uahnr/afts/img/A*lAoRbywo0oAAAAAAAAAAAAADrJ8AQ/or
>
  <div style={{ height: 500 }} />
</Watermark>

  <Input placeholder="请输入" />
</Form.Item>
<Form.Item name="color" label="Color">
  <ColorPicker />
</Form.Item>
<Form.Item name="fontSize" label="FontSize">
  <Slider step={1} min={0} max={100} />
</Form.Item>
<Form.Item name="zIndex" label="zIndex">
  <Slider step={1} min={0} max={100} />
</Form.Item>
<Form.Item name="rotate" label="Rotate">
  <Slider step={1} min={-180} max={180} />
</Form.Item>
<Form.Item label="Gap" style={{ marginBottom: 0 }}>
  <Space style={{ display: 'flex' }} align="baseline">
    <Form.Item name={['gap', 0]}>
      <InputNumber placeholder="gapX" style={{ width: '100%' }} />
    </Form.Item>
    <Form.Item name={['gap', 1]}>
      <InputNumber placeholder="gapY" style={{ width: '100%' }} />
    </Form.Item>
  </Space>
</Form.Item>
<Form.Item label="Offset" style={{ marginBottom: 0 }}>
  <Space style={{ display: 'flex' }} align="baseline">
    <Form.Item name={['offset', 0]}>
      <InputNumber placeholder="offsetLeft" style={{ width: '100%' }} />
    </Form.Item>
    <Form.Item name={['offset', 1]}>
      <InputNumber placeholder="offsetTop" style={{ width: '100%' }} />
    </Form.Item>
  </Space>
</Form.Item>
</Form>
</div>
);
};

export default App;

```