何时使用 {#when-to-use}

当需要将文本转换成为二维码时使用。

代码演示

基本使用

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

const App: React.FC = () => {
  const [text, setText] = React.useState('https://ant.design/');

  return (
    <Space direction="vertical" align="center">
      <QRCode value={text || '-'} />
      <Input
        placeholder="-"
        maxLength={60}
        value={text}
        onChange={(e) => setText(e.target.value)}
      />
    </Space>
  );
};

export default App;
```

带 Icon 的例子

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

const App: React.FC = () => (
  <QRCode
    errorLevel="H"
    value="https://ant.design/"

icon="https://gw.alipayobjects.com/zos/rmsportal/KDpgvguMpGfqaHPjicRK.svg"
  />
);

export default App;
```

不同的状态

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

const value = 'https://ant.design';

const App: React.FC = () => (
  <Flex gap="middle" wrap>
    <QRCode value={value} status="loading" />
    <QRCode value={value} status="expired" onRefresh={() =>
console.log('refresh')} />
    <QRCode value={value} status="scanned" />
  </Flex>
);

export default App;
```

自定义状态渲染器

v5.20.0

```
import React from 'react';
import { CheckCircleFilled, CloseCircleFilled, ReloadOutlined } from '@ant-
design/icons';
import type { QRCodeProps } from 'antd';
import { Button, Flex, QRCode, Space, Spin } from 'antd';

const value = 'https://ant.design';

const customStatusRender: QRCodeProps['statusRender'] = (info) => {
  switch (info.status) {
    case 'expired':
      return (
        <div>
          <CloseCircleFilled style={{ color: 'red' }} />
{info.locale?.expired}
          <p>
            <Button type="link" onClick={info.onRefresh}>
              <ReloadOutlined /> {info.locale?.refresh}
            </Button>
          </p>
        </div>
      );
    case 'loading':
      return (
```

```
        <Space direction="vertical">
          <Spin />
          <p>Loading...</p>
        </Space>
      );
    case 'scanned':
      return (
        <div>
          <CheckCircleFilled style={{ color: 'green' }} />
{info.locale?.scanned}
        </div>
      );
    default:
      return null;
  }
};

const App: React.FC = () => (
  <Flex gap="middle" wrap>
    <QRCode value={value} status="loading" statusRender=
{customStatusRender} />
    <QRCode
      value={value}
      status="expired"
      onRefresh={() => console.log('refresh')}
      statusRender={customStatusRender}
    />
    <QRCode value={value} status="scanned" statusRender=
{customStatusRender} />
  </Flex>
);

export default App;
```

自定义渲染类型

```
import React from 'react';
import { QRCode, Space } from 'antd';

const App: React.FC = () => (
  <Space>
    <QRCode type="canvas" value="https://ant.design/" />
    <QRCode type="svg" value="https://ant.design/" />
  </Space>
);
```

```
export default App;
```

自定义尺寸

```
import React, { useState } from 'react';
import { MinusOutlined, PlusOutlined } from '@ant-design/icons';
import { Button, QRCode, Space } from 'antd';

const MIN_SIZE = 48;
const MAX_SIZE = 300;

const App: React.FC = () => {
  const [size, setSize] = useState<number>(160);

  const increase = () => {
    setSize((prevSize) => {
      const newSize = prevSize + 10;
      if (newSize >= MAX_SIZE) {
        return MAX_SIZE;
      }
      return newSize;
    });
  };

  const decline = () => {
    setSize((prevSize) => {
      const newSize = prevSize - 10;
      if (newSize <= MIN_SIZE) {
        return MIN_SIZE;
      }
      return newSize;
    });
  };

  return (
    <>
      <Space.Compact style={{ marginBottom: 16 }}>
        <Button onClick={decline} disabled={size <= MIN_SIZE} icon=
{<MinusOutlined />}>
          Smaller
        </Button>
        <Button onClick={increase} disabled={size >= MAX_SIZE} icon=
{<PlusOutlined />}>
          Larger
        </Button>
```

```
      </Space.Compact>
      <QRCode
        errorLevel="H"
        size={size}
        iconSize={size / 4}
        value="https://ant.design/"

icon="https://gw.alipayobjects.com/zos/rmsportal/KDpgvguMpGfqaHPjicRK.svg"
      />
    </>
  );
};

export default App;
```

自定义颜色

```
import React from 'react';
import { QRCode, Space, theme } from 'antd';

const { useToken } = theme;

const App: React.FC = () => {
  const { token } = useToken();
  return (
    <Space>
      <QRCode value="https://ant.design/" color={token.colorSuccessText} />
      <QRCode
        value="https://ant.design/"
        color={token.colorInfoText}
        bgColor={token.colorBgLayout}
      />
    </Space>
  );
};

export default App;
```

下载二维码

```
import React from 'react';
import { Button, QRCode, Segmented, Space } from 'antd';
import type { QRCodeProps } from 'antd';

function doDownload(url: string, fileName: string) {
```

```
  const a = document.createElement('a');
  a.download = fileName;
  a.href = url;
  document.body.appendChild(a);
  a.click();
  document.body.removeChild(a);
}

const downloadCanvasQRCode = () => {
  const canvas =
document.getElementById('myqrcode')?.querySelector<HTMLCanvasElement>
('canvas');
  if (canvas) {
    const url = canvas.toDataURL();
    doDownload(url, 'QRCode.png');
  }
};

const downloadSvgQRCode = () => {
  const svg =
document.getElementById('myqrcode')?.querySelector<SVGElement>('svg');
  const svgData = new XMLSerializer().serializeToString(svg!);
  const blob = new Blob([svgData], { type: 'image/svg+xml;charset=utf-8'
});
  const url = URL.createObjectURL(blob);

  doDownload(url, 'QRCode.svg');
};

const App: React.FC = () => {
  const [renderType, setRenderType] = React.useState<QRCodeProps['type']>
('canvas');
  return (
    <Space id="myqrcode" direction="vertical">
      <Segmented options={['canvas', 'svg']} value={renderType} onChange=
{setRenderType} />
      <div>
        <QRCode
          type={renderType}
          value="https://ant.design/"
          bgColor="#fff"
          style={{ marginBottom: 16 }}

icon="https://gw.alipayobjects.com/zos/rmsportal/KDpgvguMpGfqaHPjicRK.svg"
        />
        <Button
```

```
          type="primary"
          onClick={renderType === 'canvas' ? downloadCanvasQRCode :
downloadSvgQRCode}
        >
          Download
        </Button>
      </div>
    </Space>
  );
};

export default App;
```

纠错比例

```
import React, { useState } from 'react';
import type { QRCodeProps } from 'antd';
import { QRCode, Segmented } from 'antd';

const App: React.FC = () => {
  const [level, setLevel] = useState<QRCodeProps['errorLevel']>('L');
  return (
    <>
      <QRCode
        style={{ marginBottom: 16 }}
        errorLevel={level}

value="https://gw.alipayobjects.com/zos/rmsportal/KDpgvguMpGfqaHPjicRK.svg"
      />
      <Segmented options={['L', 'M', 'Q', 'H']} value={level} onChange=
{setLevel} />
    </>
  );
};

export default App;
```

高级用法

```
import React from 'react';
import { Button, Popover, QRCode } from 'antd';

const App: React.FC = () => (
  <Popover content={<QRCode value="https://ant.design" bordered={false}
/>}>
```

```
    <Button type="primary">Hover me</Button>
  </Popover>
);

export default App;
```

API

通用属性参考：通用属性

> *自 `antd@5.1.0` 版本开始提供该组件。*

参数	说明	类型	默认值	版本
value	扫描后的文本	string	-	
type	渲染类型	`canvas | svg`	`canvas`	5.6.0
icon	二维码中图片的地址（目前只支持图片地址）	string	-	
size	二维码大小	number	160	
iconSize	二维码中图片的大小	number | { width: number; height: number }	40	5.19.0
color	二维码颜色	string	`#000`	
bgColor	二维码背景颜色	string	`transparent`	5.5.0
bordered	是否有边框	boolean	`true`	
errorLevel	二维码纠错等级	`'L' | 'M' | 'Q' | 'H'`	M	
status	二维码状态	`active | expired | loading | scanned`	`active`	scanned: 5.13.0
statusRender	自定义状态渲染器	(info: StatusRenderInfo) => React.ReactNode	-	5.20.0
onRefresh	点击"点击刷新"的回调	`() => void`	-	

StatusRenderInfo

```
type StatusRenderInfo = {
  status: QRStatus;
  locale: Locale['QRCode'];
```

```
  onRefresh?: () => void;
};
```

主题变量（Design Token）

FAQ

关于二维码纠错等级

纠错等级也叫纠错率，就是指二维码可以被遮挡后还能正常扫描，而这个能被遮挡的最大面积就是纠错率。

通常情况下二维码分为 4 个纠错级别：`L级` 可纠正约 `7%` 错误、`M级` 可纠正约 `15%` 错误、`Q级` 可纠正约 `25%` 错误、`H级` 可纠正约 `30%` 错误。并不是所有位置都可以缺损，像最明显的三个角上的方框，直接影响初始定位。中间零散的部分是内容编码，可以容忍缺损。当二维码的内容编码携带信息比较少的时候，也就是链接比较短的时候，设置不同的纠错等级，生成的图片不会发生变化。

> *有关更多信息，可参阅相关资料：https://www.qrcode.com/zh/about/error_correction*

⚠️⚠️⚠️ 二维码无法扫描?

若二维码无法扫码识别，可能是因为链接地址过长导致像素过于密集，可以通过 size 配置二维码更大，或者通过短链接服务等方式将链接变短。