

何时使用 {#when-to-use}

- 需要表示开关状态/两种状态之间的切换时；
- 和 `checkbox` 的区别是，切换 `switch` 会直接触发状态改变，而 `checkbox` 一般用于状态标记，需要和提交操作配合。

代码演示

基本

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

const onChange = (checked: boolean) => {
  console.log(`switch to ${checked}`);
};

const App: React.FC = () => <Switch defaultChecked onChange={onChange} />;

export default App;
```

不可用

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

const App: React.FC = () => {
  const [disabled, setDisabled] = useState(true);

  const toggle = () => {
    setDisabled(!disabled);
  };

  return (
    <Space direction="vertical">
      <Switch disabled={disabled} defaultChecked />
      <Button type="primary" onClick={toggle}>
        Toggle disabled
      </Button>
    </Space>
  );
};

export default App;
```

文字和图标

```
import React from 'react';
import { CheckOutlined, CloseOutlined } from '@ant-design/icons';
import { Space, Switch } from 'antd';

const App: React.FC = () => (
  <Space direction="vertical">
    <Switch checkedChildren="开启" uncheckedChildren="关闭" defaultChecked
  />
    <Switch checkedChildren="1" uncheckedChildren="0" />
    <Switch
      checkedChildren={<CheckOutlined />}
      uncheckedChildren={<CloseOutlined />}
      defaultChecked
    />
  </Space>
);

export default App;
```

两种大小

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

const App: React.FC = () => (
  <>
    <Switch defaultChecked />
    <br />
    <Switch size="small" defaultChecked />
  </>
);

export default App;
```

加载中

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

const App: React.FC = () => (
  <>
    <Switch loading defaultChecked />
  </>
);

export default App;
```

```

    <br />
    <Switch size="small" loading />
  </>
);

export default App;

```

自定义组件 Token

Debug

```

import React from 'react';
import { ConfigProvider, Space, Switch } from 'antd';

const App: React.FC = () => (
  <ConfigProvider
    theme={{
      components: {
        Switch: {
          trackHeight: 14,
          trackMinWidth: 32,
          // opacityLoading: 0.1,
          colorPrimary: 'rgb(25, 118, 210, 0.5)',
          trackPadding: -3,
          handleSize: 20,
          handleBg: 'rgb(25, 118, 210)',
          handleShadow:
            'rgba(0, 0, 0, 0.2) 0px 2px 1px -1px, rgba(0, 0, 0, 0.14) 0px
1px 1px 0px, rgba(0, 0, 0, 0.12) 0px 1px 3px 0px',
          // innerMinMargin: 4,
          // innerMaxMargin: 8,
        },
      },
    }}
  >
    <Space>
      <Switch defaultChecked />
    </Space>
  </ConfigProvider>
);

export default App;

```

API

通用属性参考: [通用属性](#)

参数	说明	类型	默认值	版本
autoFocus	组件自动获取焦点	boolean	false	
checked	指定当前是否选中	boolean	false	
checkedChildren	选中时的内容	ReactNode	-	
className	Switch 器类名	string	-	
defaultChecked	初始是否选中	boolean	false	
defaultValue	defaultChecked 的别名	boolean	-	5.12.0
disabled	是否禁用	boolean	false	
loading	加载中的开关	boolean	false	
size	开关大小，可选值：default small	string	default	
unCheckedChildren	非选中时的内容	ReactNode	-	
value	checked 的别名	boolean	-	5.12.0
onChange	变化时的回调函数	function(checked: boolean, event: Event)	-	
onClick	点击时的回调函数	function(checked: boolean, event: Event)	-	

方法

名称	描述
blur()	移除焦点
focus()	获取焦点

主题变量（Design Token）

FAQ

为什么在 Form.Item 下不能绑定数据？

Form.Item 默认绑定值属性到 value 上，而 Switch 的值属性为 checked 。你可以通过 valuePropName 来修改绑定的值属性。

```
<Form.Item name="fieldA" valuePropName="checked">
  <Switch />
</Form.Item>
```

```
</Form.Item>
```