Button **2**

To trigger an operation.

When To Use

A button means an operation (or a series of operations). Clicking a button will trigger corresponding business logic.

In Ant Design we provide 5 types of button.

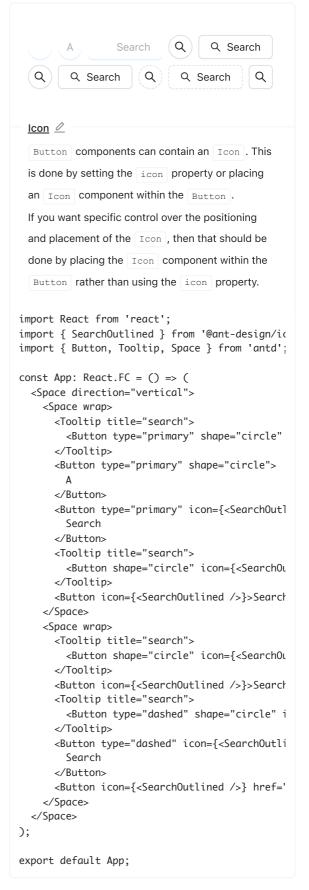
- $\circ\quad$ Primary button: indicate the main action, one primary button at most in one section.
- o Default button: indicate a series of actions without priority.
- o Dashed button: used for adding action commonly.
- Text button: used for the most secondary action.
- Link button: used for external links.

And 4 other properties additionally.

- o danger: used for actions of risk, like deletion or authorization.
- ghost: used in situations with complex background, home pages usually.
- o disabled: when actions are not available.
- o loading: add loading spinner in button, avoiding multiple submits too.

Examples

```
Primary Button
                      Default Button
   Dashed Button
                      Text Button
   Link Button
 Type 🖉
 There are primary button, default button,
  dashed button, text button and link button
 in antd.
import React from 'react';
import { Button, Space } from 'antd';
const App: React.FC = () \Rightarrow (
 <Space wrap>
   <Button type="primary">Primary Button</But
   <Button>Default Button/Button>
    <Button type="dashed">Dashed Button</Butto
    <Button type="text">Text Button/Button>
    <Button type="link">Link Button/Button>
 </Space>
);
export default App;
```



```
Large
             Default
                        Small
     Preview
                 Default
   Primary
                              Dashed
   Link
                              Download
       Download
 Size 🖉
 Ant Design supports a default button size as well
 as a large and small size.
 If a large or small button is desired, set the size
 property to either large or small respectively.
 Omit the size property for a button with the
 default size.
import React, { useState } from 'react';
import { DownloadOutlined } from '@ant-design/
import { Button, Radio, Space, Divider } from
import type { SizeType } from 'antd/es/config-
const App: React.FC = () => {
  const [size, setSize] = useState<SizeType>('
  return (
      <Radio.Group value={size} onChange={(e)</pre>
        <Radio.Button value="large">Large</Rac
        <Radio.Button value="default">Default
        <Radio.Button value="small">Small</Rac
      </Radio.Group>
      <Divider orientation="left" plain>
        Preview
      </Divider>
      <Space direction="vertical">
        <Space wrap>
           <Button type="primary" size={size}>
            Primary
           </Button>
           <Button size={size}>Default</Button>
           <Button type="dashed" size={size}>
            Dashed
           </Button>
        </Space>
        <Button type="link" size={size}>
          Link
        </Button>
        <Space wrap>
           <Button type="primary" icon={<Download</pre>
           <Button type="primary" shape="circle"</pre>
           <Button type="primary" shape="round"</pre>
           <Button type="primary" shape="round"</pre>
            Download
           </Button>
           <Button type="primary" icon={<Download</pre>
            Download
          </Button>
        </Space>
      </Space>
    </>
```

```
Primary
   Default
   Dashed
   Text
   Link
            Link(disabled)
   Href Primary
   Danger Default
   Danger Text
   Danger Link
                   Danger Link(disabled)
     Ghost
 Disabled 
 To mark a button as disabled, add the disabled
 property to the Button .
import React from 'react';
import { Button, Space } from 'antd';
const App: React.FC = () => (
 <Space direction="vertical">
    <Space>
      <Button type="primary">Primary/Button>
      <Button type="primary" disabled>
        Primary(disabled)
      </Button>
    </Space>
    <Space>
      <Button>Default</Button>
      <Button disabled>Default(disabled)</Butt
    </Space>
    <Space>
      <Button type="dashed">Dashed</Button>
      <Button type="dashed" disabled>
        Dashed(disabled)
      </Button>
    </Space>
      <Button type="text">Text</Button>
      <Button type="text" disabled>
        Text(disabled)
      </Button>
    </Space>
    <Space>
      <Button type="link">Link
      <Button type="link" disabled>
        Link(disabled)
      </Button>
    </Space>
    <Space>
      <Button type="primary" href='https://ant
        Href Primary
      </Button>
      <Button type="primary" href='https://ant</pre>
        Href Primary(disabled)
      </Button>
    </Space>
    <Space>
       Dutton denger Dancer Default / Dutton
```

```
Loading Loading

Click me! Click me!
```

```
Loading 2
 A loading indicator can be added to a button by
 setting the loading property on the Button .
import React, { useState } from 'react';
import { PoweroffOutlined } from '@ant-design/
import { Button, Space } from 'antd';
const App: React.FC = () => {
  const [loadings, setLoadings] = useState<box</pre>
  const enterLoading = (index: number) => {
    setLoadings((prevLoadings) => {
      const newLoadings = [...prevLoadings];
      newLoadings[index] = true;
      return newLoadings;
    });
    setTimeout(() => {
      setLoadings((prevLoadings) => {
        const newLoadings = [...prevLoadings];
        newLoadings[index] = false;
        return newLoadings;
      });
    }, 6000);
  };
  return (
    <Space direction="vertical">
      <Space wrap>
        <Button type="primary" loading>
          Loading
        </Button>
        <Button type="primary" size="small" lc
          Loading
        </Button>
        <Button type="primary" icon={<Poweroff</pre>
      </Space>
      <Space wrap>
        <Button type="primary" loading={loadir</pre>
          Click me!
        </Button>
        <Button
          type="primary"
          icon={<PoweroffOutlined />}
          loading={loadings[1]}
          onClick={() => enterLoading(1)}
          Click me!
        </Button>
        <Button
          type="primary"
          icon={<PoweroffOutlined />}
          loading={loadings[2]}
          onClick={() => enterLoading(2)}
        />
      </Space>
    </Space>
  );
};
export default App;
```

```
secondary

Actions ...
```

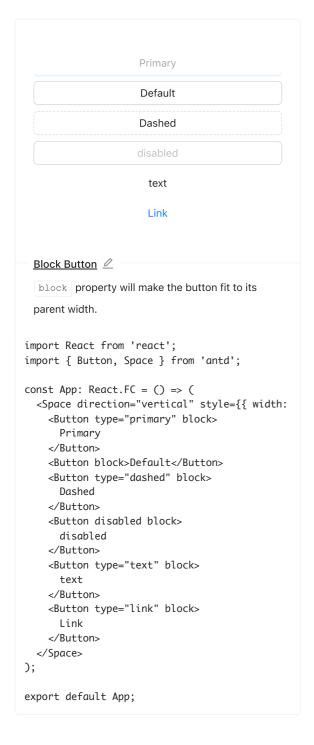
Multiple Buttons 🖉

If you need several buttons, we recommend that you use 1 primary button + n secondary buttons, and if there are more than three operations, you can group some of them into <u>Dropdown.Button</u>.

```
import React from 'react';
import type { MenuProps } from 'antd';
import { Button, Dropdown, Space } from 'antd'
const onMenuClick: MenuProps['onClick'] = (e)
  console.log('click', e);
};
const items = [
  {
    key: '1',
    label: '1st item',
  },
  {
    key: '2',
    label: '2nd item',
  },
  {
    key: '3',
    label: '3rd item',
];
const App: React.FC = () => (
  <Space direction="vertical">
    <Button type="primary">primary</Button>
    <Button>secondary</Button>
    <Dropdown.Button menu={{ items, onClick: </pre>
  </Space>
);
export default App;
```

```
Primary
                  Default
                             Dashed
     Danger
 Ghost Button
  ghost property will make button's background
 transparent, it is commonly used in colored
 background.
import React from 'react';
import { Button, Space } from 'antd';
const App: React.FC = () => (
  <Space className="site-button-ghost-wrapper"</pre>
    <Button type="primary" ghost>
     Primary
    </Button>
    <Button ghost>Default</Button>
    <Button type="dashed" ghost>
     Dashed
    </Button>
    <Button type="primary" danger ghost>
     Danger
    </Button>
  </Space>
);
export default App;
```

```
Dashed
    Primary
                Default
    Text
             Link
 Danger Buttons 🖉
  danger is a property of button after antd 4.0.
import React from 'react';
import { Button, Space } from 'antd';
const App: React.FC = () \Rightarrow (
  <Space wrap>
    <Button type="primary" danger>
      Primary
    </Button>
    <Button danger>Default</Button>
    <Button type="dashed" danger>
      Dashed
    </Button>
    <Button type="text" danger>
      Text
    </Button>
    <Button type="link" danger>
      Link
    </Button>
  </Space>
);
export default App;
```



API

Different button styles can be generated by setting Button properties. The recommended order is: type -> shape -> size -> loading -> disabled .

Property	Description	Туре	Default	Version
block	Option to fit button width to its parent width	boolean	false	
danger	Set the danger status of button	boolean	false	
disabled	Disabled state of button	boolean	false	
ghost	Make background transparent and invert text and border colors	boolean	false	

Property	Description	Туре	Default	Version
href	Redirect url of link button	string	_	
htmlType	Set the original html type of button, see: MDN	string	button	
icon	Set the icon component of button	ReactNode	-	
loading	Set the loading status of button	<pre>boolean { delay: number }</pre>	false	
shape	Can be set button shape	default circle round	default	
size	Set the size of button	large middle small	middle	
target	Same as target attribute of a, works when href is specified	string	-	
type	Can be set to primary ghost dashed link text default	string	default	
onClick	Set the handler to handle click event	<pre>(event: MouseEvent) => void</pre>	-	

It accepts all props which native buttons support.

Design Token

▼ Global Token

Token Name	Description	Туре	Default Value	
colorBgContainer	Container background color, e.g: default button, input box, etc. Be sure not to confuse this with `colorBgElevated`.	string	#ffffff	
colorBgContainerDisabled	Control the background color of container in disabled state.	string	rgba(0, 0, 0, 0, 0.04)	
colorBgTextActive	Control the background color of text in active state.	string	rgba(0, 0, 0, 0, 0.15)	
colorBgTextHover	Control the background color of text in hover state.	string	rgba(0, 0, 0, 0, 0.06)	
colorBorder	Default border color, used to separate different elements, such as: form separator, card separator, etc.	string	□ #d9d9d9	
colorError	Used to represent the visual elements of the operation failure, such as the error Button, error Result component, etc.	string	□ #ff4d4f	

Token Name	Description	Туре	Default Value
colorErrorActive	The active state of the error color.	string	□ #d9363e
colorErrorBg	The background color of the error state.	string	□ #fff2f0
colorErrorBorderHover	The hover state border color of the error state.	string	□ #ffa39e
colorErrorHover	The hover state of the error color.	string	□ #ff7875
colorErrorOutline	Control the outline color of input component in error state.	string	□ rgba(255, 38, 5, 0.06)
colorLink	Control the color of hyperlink.	string	□ #1677ff
colorLinkActive	Control the color of hyperlink when clicked.	string	□ #0958d9
colorLinkHover	Control the color of hyperlink when hovering.	string	□ #69b1ff
colorPrimary	Brand color is one of the most direct visual elements to reflect the characteristics and communication of the product. After you have selected the brand color, we will automatically generate a complete color palette and assign it effective design semantics.	string	□ #1677ff
colorPrimaryActive	Dark active state under the main color gradient.	string	□ #0958d9
colorPrimaryBorder	The stroke color under the main color gradient, used on the stroke of components such as Slider.	string	#91caff
colorPrimaryHover	Hover state under the main color gradient.	string	□ #4096ff
colorText	Default text color which comply with W3C standards, and this color is also the darkest neutral color.	string	rgba(0, 0, 0, 0, 0.88)
colorTextDisabled	Control the color of text in disabled state.	string	rgba(0, 0, 0, 0, 0.25)
colorTextLightSolid	Control the highlight color of text with background color, such as the text in Primary Button components.	string	#fff
borderRadius	Border radius of base components	number	6
borderRadiusLG	LG size border radius, used in some large border radius components, such as Card, Modal and other components.	number	8
borderRadiusSM	SM size border radius, used in small size components, such as Button, Input, Select and other input components in small size	number	4
controlHeight	The height of the basic controls such as buttons and input boxes in Ant Design	number	32
controlHeightLG	LG component height	number	40
controlHeightSM	SM component height	number	24

Token Name	Description	Туре	Default Value
controlOutline	Control the outline color of input component.	string	□ rgba(5, 145, 255, 0.1)
controlOutlineWidth	Control the outline width of input component.	number	2
controlTmpOutline	Default style outline color.	string	□rgba(0, 0, 0, 0, 0.02)
fontSize	The most widely used font size in the design system, from which the text gradient will be derived.	number	14
fontSizeLG	Large font size	number	16
lineHeight	Line height of text.	number	1.5714285714285714
lineType	Border style of base components	string	solid
lineWidth	Border width of base components	number	1
lineWidthFocus	Control the width of the line when the component is in focus state.	number	4
marginXS	Control the margin of an element, with a small size.	number	8
motionDurationMid	Motion speed, medium speed. Used for medium element animation interaction.	string	0.2s
motionDurationSlow	Motion speed, slow speed. Used for large element animation interaction.	string	0.3s
motionEaseInOut	Preset motion curve.	string	cubic-bezier(0.645, 0.045, 0.355, 1)
opacityLoading	Control the opacity of the loading state.	number	0.65
paddingContentHorizontal	Control the horizontal padding of content element.	number	16
paddingXS	Control the extra small padding of the element.	number	8

FAQ

How to remove space between 2 chinese characters?

Following the Ant Design specification, we will add one space between if Button (exclude Text button and Link button) contains two Chinese characters only. If you don't need that, you can use <u>ConfigProvider</u> to set

autoInsertSpaceInButton as false.