










When To Use

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

In Ant Design we provide 5 types of button.

-  Primary button: used for the main action, there can be at most one primary button in a section.
-  Default button: used for a series of actions without priority.
-  Dashed button: commonly used for adding more actions.
-  Text button: used for the most secondary action.
-  Link button: used for external links.

And 4 other properties additionally.

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

Examples

Syntactic sugar

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

const App: React.FC = () => (
  <Flex gap="small" wrap>
    <Button type="primary">Primary Button</Button>
    <Button>Default Button</Button>
    <Button type="dashed">Dashed Button</Button>
    <Button type="text">Text Button</Button>
    <Button type="link">Link Button</Button>
  </Flex>
);

export default App;
```

Color & Variant

v5.21.0

```
import React from 'react';
import { Button, ConfigProvider, Flex } from 'antd';
import { useResponsive } from 'antd-style';
```

```
const App: React.FC = () => {
  const { xxl } = useResponsive();

  return (
    <ConfigProvider componentSize={xxl ? 'middle' : 'small'}>
      <Flex vertical gap="small">
        <Flex gap="small" wrap>
          <Button color="default" variant="solid">
            Solid
          </Button>
          <Button color="default" variant="outlined">
            Outlined
          </Button>
          <Button color="default" variant="dashed">
            Dashed
          </Button>
          <Button color="default" variant="filled">
            Filled
          </Button>
          <Button color="default" variant="text">
            Text
          </Button>
          <Button color="default" variant="link">
            Link
          </Button>
        </Flex>
        <Flex gap="small" wrap>
          <Button color="primary" variant="solid">
            Solid
          </Button>
          <Button color="primary" variant="outlined">
            Outlined
          </Button>
          <Button color="primary" variant="dashed">
            Dashed
          </Button>
          <Button color="primary" variant="filled">
            Filled
          </Button>
          <Button color="primary" variant="text">
            Text
          </Button>
          <Button color="primary" variant="link">
            Link
          </Button>
        </Flex>
      </Flex>
    </ConfigProvider>
  );
}
```

```
<Flex gap="small" wrap>
  <Button color="danger" variant="solid">
    Solid
  </Button>
  <Button color="danger" variant="outlined">
    Outlined
  </Button>
  <Button color="danger" variant="dashed">
    Dashed
  </Button>
  <Button color="danger" variant="filled">
    Filled
  </Button>
  <Button color="danger" variant="text">
    Text
  </Button>
  <Button color="danger" variant="link">
    Link
  </Button>
</Flex>
<Flex gap="small" wrap>
  <Button color="pink" variant="solid">
    Solid
  </Button>
  <Button color="pink" variant="outlined">
    Outlined
  </Button>
  <Button color="pink" variant="dashed">
    Dashed
  </Button>
  <Button color="pink" variant="filled">
    Filled
  </Button>
  <Button color="pink" variant="text">
    Text
  </Button>
  <Button color="pink" variant="link">
    Link
  </Button>
</Flex>
<Flex gap="small" wrap>
  <Button color="purple" variant="solid">
    Solid
  </Button>
  <Button color="purple" variant="outlined">
    Outlined
```

```

    </Button>
    <Button color="purple" variant="dashed">
      Dashed
    </Button>
    <Button color="purple" variant="filled">
      Filled
    </Button>
    <Button color="purple" variant="text">
      Text
    </Button>
    <Button color="purple" variant="link">
      Link
    </Button>
  </Flex>
  <Flex gap="small" wrap>
    <Button color="cyan" variant="solid">
      Solid
    </Button>
    <Button color="cyan" variant="outlined">
      Outlined
    </Button>
    <Button color="cyan" variant="dashed">
      Dashed
    </Button>
    <Button color="cyan" variant="filled">
      Filled
    </Button>
    <Button color="cyan" variant="text">
      Text
    </Button>
    <Button color="cyan" variant="link">
      Link
    </Button>
  </Flex>
</Flex>
</ConfigProvider>
);
};

export default App;

```

Icon

```

import React from 'react';
import { SearchOutlined } from '@ant-design/icons';
import { Button, Flex, Tooltip } from 'antd';

```

```

const App: React.FC = () => (
  <Flex gap="small" vertical>
    <Flex wrap gap="small">
      <Tooltip title="search">
        <Button type="primary" shape="circle" icon={<SearchOutlined />} />
      </Tooltip>
      <Button type="primary" shape="circle">
        A
      </Button>
      <Button type="primary" icon={<SearchOutlined />}>
        Search
      </Button>
      <Tooltip title="search">
        <Button shape="circle" icon={<SearchOutlined />} />
      </Tooltip>
      <Button icon={<SearchOutlined />}>Search</Button>
    </Flex>
    <Flex wrap gap="small">
      <Tooltip title="search">
        <Button shape="circle" icon={<SearchOutlined />} />
      </Tooltip>
      <Button icon={<SearchOutlined />}>Search</Button>
      <Tooltip title="search">
        <Button type="dashed" shape="circle" icon={<SearchOutlined />} />
      </Tooltip>
      <Button type="dashed" icon={<SearchOutlined />}>
        Search
      </Button>
      <Button icon={<SearchOutlined />} href="https://www.google.com"
target="_blank" />
    </Flex>
  </Flex>
);

export default App;

```

Icon Position

v5.17.0

```

import React, { useState } from 'react';
import { SearchOutlined } from '@ant-design/icons';
import { Button, Divider, Flex, Radio, Space, Tooltip } from 'antd';

const App: React.FC = () => {

```

```

const [position, setPosition] = useState<'start' | 'end'>('end');

return (
  <>
    <Space>
      <Radio.Group value={position} onChange={(e) =>
setPosition(e.target.value)}>
        <Radio.Button value="start">start</Radio.Button>
        <Radio.Button value="end">end</Radio.Button>
      </Radio.Group>
    </Space>
    <Divider orientation="left" plain>
      Preview
    </Divider>
    <Flex gap="small" vertical>
      <Flex wrap gap="small">
        <Tooltip title="search">
          <Button type="primary" shape="circle" icon={<SearchOutlined />}
/>

          </Tooltip>
          <Button type="primary" shape="circle">
            A
          </Button>
          <Button type="primary" icon={<SearchOutlined />} iconPosition=
{position}>
            Search
          </Button>
          <Tooltip title="search">
            <Button shape="circle" icon={<SearchOutlined />} />
          </Tooltip>
          <Button icon={<SearchOutlined />} iconPosition={position}>
            Search
          </Button>
        </Flex>
        <Flex wrap gap="small">
          <Tooltip title="search">
            <Button shape="circle" icon={<SearchOutlined />} />
          </Tooltip>
          <Button icon={<SearchOutlined />} type="text" iconPosition=
{position}>
            Search
          </Button>
          <Tooltip title="search">
            <Button type="dashed" shape="circle" icon={<SearchOutlined />}
/>

            </Tooltip>

```

```

        <Button type="dashed" icon={<SearchOutlined />} iconPosition=
{position}>
          Search
        </Button>
        <Button
          icon={<SearchOutlined />}
          href="https://www.google.com"
          target="_blank"
          iconPosition={position}
        />
        <Button type="primary" loading iconPosition={position}>
          Loading
        </Button>
      </Flex>
    </Flex>
  </>
);
};

export default App;

```

Debug Icon

Debug

```

import React from 'react';
import { MinusSquareOutlined, SearchOutlined } from '@ant-design/icons';
import { Button, ConfigProvider, Divider, Flex, Radio, Tooltip, Input }
from 'antd';
import type { ConfigProviderProps } from 'antd';
import { FiColumns } from 'react-icons/fi';

type SizeType = ConfigProviderProps['componentSize'];

/**12px 图标 */
const Icon12Size = () => <div style={{ background: 'red', width: 12,
height: 12 }} />;
/**16px 图标 */
const Icon16Size = () => <div style={{ background: 'green', width: 16,
height: 16 }} />;
/**不规则宽高 */
const IconIrregularSize = () => <div style={{ background: 'blue', width:
14, height: 16 }} />;

const App: React.FC = () => {
  const [size, setSize] = React.useState<SizeType>('large');

```

```

return (
  <>
    <Radio.Group value={size} onChange={(e) => setSize(e.target.value)}>
      <Radio.Button value="large">Large</Radio.Button>
      <Radio.Button value="default">Default</Radio.Button>
      <Radio.Button value="small">Small</Radio.Button>
    </Radio.Group>
    <Divider orientation="left" plain>
      Preview
    </Divider>
    <ConfigProvider componentSize={size}>
      <Flex gap="small" vertical>
        <Flex gap="small" wrap>
          <Tooltip title="search">
            <Button type="primary" shape="circle" icon={<SearchOutlined
/>} />
            </Tooltip>
            <Button type="primary" shape="circle">
              A
            </Button>
            <Button type="primary" icon={<SearchOutlined />}>
              Search
            </Button>
            <Tooltip title="search">
              <Button shape="circle" icon={<SearchOutlined />} />
            </Tooltip>
            <Button icon={<SearchOutlined />}>Search</Button>
          </Flex>
          <Flex gap="small" wrap>
            <Tooltip title="search">
              <Button shape="circle" icon={<SearchOutlined />} />
            </Tooltip>
            <Button icon={<SearchOutlined />}>Search</Button>
            <Tooltip title="search">
              <Button type="dashed" shape="circle" icon={<SearchOutlined
/>} />
              </Tooltip>
              <Button type="dashed" icon={<SearchOutlined />}>
                Search
              </Button>
              <Button icon={<SearchOutlined />} href="https://www.google.com"
target="_blank" />
              <Button>
                <SearchOutlined />
                Search
              </Button>

```



```

    </Flex>
    <Divider plain> https://github.com/ant-design/ant-
design/issues/51811 </Divider>
    <div>
      <Button>without icon</Button>
      <Button icon={<SearchOutlined />}>with icon</Button>
    </div>
    <Divider plain> https://github.com/ant-design/ant-
design/issues/52124 </Divider>
    <div>
      <Button
        style={{
          height: 60,
        }}
      >
        without icon
      </Button>
      <Button
        icon={<SearchOutlined />}
        style={{
          height: 60,
        }}
      >
        with icon
      </Button>
    </div>
    <Divider plain> https://github.com/ant-design/ant-
design/issues/51380 </Divider>
    <div>
      <Button size="large" icon={<FiColumns className="my-class-name"
/>} />
      <Button size="large" icon={<FiColumns />}>
        custom icon
      </Button>
      <Button icon={<SearchOutlined />} />
      <Button icon={<SearchOutlined />}>with icon</Button>
      <Button size="large">without icon</Button>
      <Input.Search style={{ width: 100 }} />
    </div>
    <Divider plain> https://github.com/ant-design/ant-
design/issues/51380 </Divider>
    <Flex
      gap="small"
      style={{
        transform: 'scale(3)',
        transformOrigin: 'left top',

```

```

    }}
  >
    <Button icon={<MinusSquareOutlined />} />
    <Button icon={<Icon12Size />} />
    <Button icon={<Icon16Size />} />
    <Button icon={<IconIrregularSize />} />
  </Flex>
</Flex>
</ConfigProvider>
</>
);
};

export default App;

```

Debug Block

Debug

```

import React from 'react';
import { DownloadOutlined } from '@ant-design/icons';
import { Button, Form } from 'antd';

const App: React.FC = () => (
  <Form>
    <Form.Item>
      <Button size="large" shape="round" block style={{ marginBottom: 12
    }}>
        Submit
      </Button>
      <Button size="large" shape="round" icon={<DownloadOutlined />} />
    </Form.Item>
  </Form>
);

export default App;

```

Size

```

import React, { useState } from 'react';
import { DownloadOutlined } from '@ant-design/icons';
import { Button, Divider, Flex, Radio } from 'antd';
import type { ConfigProviderProps } from 'antd';

type SizeType = ConfigProviderProps['componentSize'];

```

```

const App: React.FC = () => {
  const [size, setSize] = useState<SizeType>('large'); // default is
'middle'
  return (
    <>
      <Radio.Group value={size} onChange={(e) => setSize(e.target.value)}>
        <Radio.Button value="large">Large</Radio.Button>
        <Radio.Button value="default">Default</Radio.Button>
        <Radio.Button value="small">Small</Radio.Button>
      </Radio.Group>
      <Divider orientation="left" plain>
        Preview
      </Divider>
      <Flex gap="small" align="flex-start" vertical>
        <Flex gap="small" wrap>
          <Button type="primary" size={size}>
            Primary
          </Button>
          <Button size={size}>Default</Button>
          <Button type="dashed" size={size}>
            Dashed
          </Button>
        </Flex>
        <Button type="link" size={size}>
          Link
        </Button>
        <Flex gap="small" wrap>
          <Button type="primary" icon={<DownloadOutlined />} size={size} />
          <Button type="primary" shape="circle" icon={<DownloadOutlined />}
size={size} />
          <Button type="primary" shape="round" icon={<DownloadOutlined />}
size={size} />
          <Button type="primary" shape="round" icon={<DownloadOutlined />}
size={size}>
            Download
          </Button>
          <Button type="primary" icon={<DownloadOutlined />} size={size}>
            Download
          </Button>
        </Flex>
      </Flex>
    </>
  );
};

export default App;

```

Disabled

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

const App: React.FC = () => (
  <Flex gap="small" align="flex-start" vertical>
    <Flex gap="small">
      <Button type="primary">Primary</Button>
      <Button type="primary" disabled>
        Primary(disabled)
      </Button>
    </Flex>
    <Flex gap="small">
      <Button>Default</Button>
      <Button disabled>Default(disabled)</Button>
    </Flex>
    <Flex gap="small">
      <Button type="dashed">Dashed</Button>
      <Button type="dashed" disabled>
        Dashed(disabled)
      </Button>
    </Flex>
    <Flex gap="small">
      <Button type="text">Text</Button>
      <Button type="text" disabled>
        Text(disabled)
      </Button>
    </Flex>
    <Flex gap="small">
      <Button type="link">Link</Button>
      <Button type="link" disabled>
        Link(disabled)
      </Button>
    </Flex>
    <Flex gap="small">
      <Button type="primary" href="https://ant.design/index-cn">
        Href Primary
      </Button>
      <Button type="primary" href="https://ant.design/index-cn" disabled>
        Href Primary(disabled)
      </Button>
    </Flex>
    <Flex gap="small">
      <Button danger>Danger Default</Button>
      <Button danger disabled>

```

```

        Danger Default(disabled)
      </Button>
    </Flex>
    <Flex gap="small">
      <Button danger type="text">
        Danger Text
      </Button>
      <Button danger type="text" disabled>
        Danger Text(disabled)
      </Button>
    </Flex>
    <Flex gap="small">
      <Button type="link" danger>
        Danger Link
      </Button>
      <Button type="link" danger disabled>
        Danger Link(disabled)
      </Button>
    </Flex>
    <Flex gap="small" className="site-button-ghost-wrapper">
      <Button ghost>Ghost</Button>
      <Button ghost disabled>
        Ghost(disabled)
      </Button>
    </Flex>
  </Flex>
);

export default App;

```

Loading

```

import React, { useState } from 'react';
import { PoweroffOutlined, SyncOutlined } from '@ant-design/icons';
import { Button, Flex } from 'antd';

const App: React.FC = () => {
  const [loadings, setLoadings] = useState<boolean[]>([]);

  const enterLoading = (index: number) => {
    setLoadings((prevLoadings) => {
      const newLoadings = [...prevLoadings];
      newLoadings[index] = true;
      return newLoadings;
    });
  };

```

```

setTimeout(() => {
  setLoading((prevLoadings) => {
    const newLoadings = [...prevLoadings];
    newLoadings[index] = false;
    return newLoadings;
  });
}, 3000);
};

return (
  <Flex gap="small" vertical>
    <Flex gap="small" align="center" wrap>
      <Button type="primary" loading>
        Loading
      </Button>
      <Button type="primary" size="small" loading>
        Loading
      </Button>
      <Button type="primary" icon={<PoweroffOutlined />} loading />
      <Button type="primary" loading={{ icon: <SyncOutlined spin /> }}>
        Loading Icon
      </Button>
    </Flex>
    <Flex gap="small" wrap>
      <Button type="primary" loading={loadings[0]} onClick={() =>
enterLoading(0)}>
        Icon Start
      </Button>
      <Button
        type="primary"
        loading={loadings[2]}
        onClick={() => enterLoading(2)}
        iconPosition="end"
      >
        Icon End
      </Button>
      <Button
        type="primary"
        icon={<PoweroffOutlined />}
        loading={loadings[1]}
        onClick={() => enterLoading(1)}
      >
        Icon Replace
      </Button>
      <Button
        type="primary"

```

```

        icon={<PoweroffOutlined />}
        loading={loadings[3]}
        onClick={() => enterLoading(3)}
      />
      <Button
        type="primary"
        icon={<PoweroffOutlined />}
        loading={loadings[3] && { icon: <SyncOutlined spin /> }}
        onClick={() => enterLoading(3)}
      >
        Loading Icon
      </Button>
    </Flex>
  </Flex>
);
};

export default App;

```

Multiple Buttons

```

import React from 'react';
import type { MenuProps } from 'antd';
import { Button, Dropdown, Flex } 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 = () => (
  <Flex align="flex-start" gap="small" vertical>

```

```

    <Button type="primary">primary</Button>
    <Button>secondary</Button>
    <Dropdown.Button menu={{ items, onClick: onMenuClick
}}>Actions</Dropdown.Button>
  </Flex>
);

export default App;

```

Ghost Button

```

import React from 'react';
import { Button, Flex } from 'antd';

const App: React.FC = () => (
  <Flex wrap gap="small" className="site-button-ghost-wrapper">
    <Button type="primary" ghost>
      Primary
    </Button>
    <Button ghost>Default</Button>
    <Button type="dashed" ghost>
      Dashed
    </Button>
    <Button type="primary" danger ghost>
      Danger
    </Button>
  </Flex>
);

export default App;

```

Danger Buttons

```

import React from 'react';
import { Button, Flex } from 'antd';

const App: React.FC = () => (
  <Flex wrap gap="small">
    <Button type="primary" danger>
      Primary
    </Button>
    <Button danger>Default</Button>
    <Button type="dashed" danger>
      Dashed
    </Button>
  </Flex>
);

```



```

    <Button type="text" danger>
      Text
    </Button>
    <Button type="link" danger>
      Link
    </Button>
  </Flex>
);

export default App;

```

Block Button

```

import React from 'react';
import { Button, Flex } from 'antd';

const App: React.FC = () => (
  <Flex vertical gap="small" style={{ width: '100%' }}>
    <Button type="primary" block>
      Primary
    </Button>
    <Button block>Default</Button>
    <Button type="dashed" block>
      Dashed
    </Button>
    <Button disabled block>
      disabled
    </Button>
    <Button type="text" block>
      text
    </Button>
    <Button type="link" block>
      Link
    </Button>
  </Flex>
);

export default App;

```

Deprecated Button Group

Debug

```

import React from 'react';
import { DownloadOutlined } from '@ant-design/icons';
import { Button, Tooltip } from 'antd';

```

```
import type { GetProps } from 'antd';

type ButtonGroupProps = GetProps<typeof Button.Group>;

const CustomGroup: React.FC<ButtonGroupProps> = (props) => (
  <Button.Group {...props}>
    <Button type="primary">Button 1</Button>
    <Button type="primary">Button 2</Button>
    <Tooltip title="Tooltip">
      <Button type="primary" icon={<DownloadOutlined />} disabled />
    </Tooltip>
    <Tooltip title="Tooltip">
      <Button type="primary" icon={<DownloadOutlined />} />
    </Tooltip>
  </Button.Group>
);

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

export default App;
```

Loading style bug

Debug

```
import React from 'react';
import { PoweroffOutlined } from '@ant-design/icons';
import { Button, Flex } from 'antd';

const Text1 = () => <>部署</>;
const Text2 = () => <span>部署</span>;
const Text3 = () => <>Submit</>;

const App: React.FC = () => (
  <Flex wrap gap="small">
    <Button>
      <span>
        <span>部署</span>
      </span>
    </Button>
  </Flex>
);
```

```

        </span>
      </Button>
      <Button loading>部署</Button>
      <Button loading>
        <Text1 />
      </Button>
      <Button loading>
        <Text2 />
      </Button>
      <Button loading>
        <Text3 />
      </Button>
      <Button loading icon={<PoweroffOutlined />}>
        <Text1 />
      </Button>
      <Button loading>按钮</Button>
    </Flex>
  );

  export default App;

```

Component Token

Debug

```

import React from 'react';
import { Button, ConfigProvider, Flex } from 'antd';

const App: React.FC = () => (
  <Flex gap="small" vertical>
    <ConfigProvider
      theme={{
        components: {
          Button: {
            algorithm: true,
            colorPrimary: '#1976d2',
            controlHeight: 36,
            primaryShadow:
              '0 3px 1px -2px rgba(0,0,0,0.2), 0 2px 2px 0
              rgba(0,0,0,0.14), 0 1px 5px 0 rgba(0,0,0,0.12)',
            fontWeight: 500,
            defaultBorderColor: 'rgba(25, 118, 210, 0.5)',
            colorText: '#1976d2',
            defaultColor: '#1976d2',
            borderRadius: 4,
            colorTextDisabled: 'rgba(0, 0, 0, 0.26)',

```

```

        colorBgContainerDisabled: 'rgba(0, 0, 0, 0.12)',
        contentFontSizeSM: 12,
      },
    },
  }}
>
<Flex wrap gap="small">
  <Button type="text">TEXT</Button>
  <Button type="primary">CONTAINED</Button>
  <Button>OUTLINED</Button>
</Flex>
<Flex wrap gap="small">
  <Button type="text" disabled>
    TEXT
  </Button>
  <Button type="primary" disabled>
    CONTAINED
  </Button>
  <ConfigProvider
    theme={{
      components: {
        Button: {
          borderColorDisabled: 'rgba(0, 0, 0, 0.12)',
          colorBgContainerDisabled: 'transparent',
        },
      },
    }}
  >
    <Button disabled>OUTLINED</Button>
  </ConfigProvider>
</Flex>
<Flex wrap gap="small">
  <Button type="text" size="small">
    TEXT
  </Button>
  <Button type="primary" size="small">
    CONTAINED
  </Button>
  <Button size="small">OUTLINED</Button>
</Flex>
</ConfigProvider>
<ConfigProvider
  theme={{
    token: {
      colorPrimary: '#FF0000',
    },
  },

```

```

    }}
  >
    <Flex gap="small" wrap>
      <Button type="text">Text</Button>
      <Button type="link">Link</Button>
      <Button color="primary" variant="text">
        Primary Text
      </Button>
      <Button color="primary" variant="link">
        Primary Link
      </Button>
    </Flex>
  </ConfigProvider>
</Flex>
);

export default App;

```

Gradient Button

```

import React from 'react';
import { AntDesignOutlined } from '@ant-design/icons';
import { Button, ConfigProvider, Space } from 'antd';
import { createStyles } from 'antd-style';

const useStyle = createStyles(({ prefixCls, css }) => ({
  linearGradientButton: css`
    &.${prefixCls}-btn-primary:not([disabled]):not(&.${prefixCls}-btn-
dangerous) {
      > span {
        position: relative;
      }

      &::before {
        content: '';
        background: linear-gradient(135deg, #6253e1, #04befe);
        position: absolute;
        inset: -1px;
        opacity: 1;
        transition: all 0.3s;
        border-radius: inherit;
      }

      &:hover::before {
        opacity: 0;
      }
    }
  `
});

```

```

    }
    ` ,
  }));

const App: React.FC = () => {
  const { styles } = useStyles();

  return (
    <ConfigProvider
      button={{
        className: styles.linearGradientButton,
      }}
    >
      <Space>
        <Button type="primary" size="large" icon={<AntDesignOutlined />}>
          Gradient Button
        </Button>
        <Button size="large">Button</Button>
      </Space>
    </ConfigProvider>
  );
};

export default App;

```

API

Common props ref: [Common props](#)

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

Property	Description	Type	Default	
autoInsertSpace	We add a space between two Chinese characters by default, which removed by setting autoInsertSpace to false.	boolean	true	5.1
block	Option to fit button width to its parent width	boolean	false	

classNames	Semantic DOM class	Record<SemanticDOM, string>	-	5.4
color	Set button color	default primary danger PresetColors	-	de pr da Pr 5.2
danger	Syntactic sugar. Set the danger status of button. will follow color if provided	boolean	false	
disabled	Disabled state of button	boolean	false	
ghost	Make background transparent and invert text and border colors	boolean	false	
href	Redirect url of link button	string	-	
htmlType	Set the original html type of button, see: MDN	submit reset button	button	
icon	Set the icon component of button	ReactNode	-	
iconPosition	Set the icon position of button	start end	start	5.1
loading	Set the loading status of button	boolean { delay: number, icon: ReactNode }	false	icc
shape	Can be used to set button shape	default circle round	default	
size	Set the size of button	large middle small	middle	
styles	Semantic DOM style	Record<SemanticDOM, CSSProperties>	-	5.4

target	Same as target attribute of a, works when href is specified	string	-	
type	Syntactic sugar. Set button type. Will follow variant & color if provided	primary dashed link text default	default	
onClick	Set the handler to handle click event	(event: React.MouseEvent<HTMLElement, MouseEvent>) => void	-	
variant	Set button variant	outlined dashed solid filled text link	-	5.2

It accepts all props which native buttons support.

PresetColors

```
type PresetColors = 'blue' | 'purple' | 'cyan' | 'green' | 'magenta' | 'pink' | 'red' | 'orange' | 'yellow' | 'volcano' | 'geekblue' | 'lime' | 'gold';
```

Semantic DOM

演示

```
import React from 'react';
import { AntDesignOutlined } from '@ant-design/icons';
import { Button } from 'antd';

import SemanticPreview from '../../dumi/components/SemanticPreview';
import useLocale from '../../dumi/hooks/useLocale';

const locales = {
  cn: {
    icon: '图标元素',
  },
  en: {
    icon: 'Icon element',
  },
};

const App: React.FC = () => {
  const [locale] = useLocale(locales);
```



```

return (
  <SemanticPreview
    componentName="Button"
    semantics={[{ name: 'icon', desc: locale.icon, version: '5.5.0' }]}
  >
    <Button type="primary" icon={<AntDesignOutlined />}>
      Ant Design
    </Button>
  </SemanticPreview>
);
};

export default App;

```

Design Token

FAQ

How to choose type and color & variant? {#faq-type-color-variant}

Type is essentially syntactic sugar for colors and variants. It internally provides a set of mapping relationships between colors and variants for the type. If both exist at the same time, the colors and variants will be used first.

```
<Button type="primary">click</Button>
```

Equivalent

```

<Button color="primary" variant="solid">
  click
</Button>

```

How to close the click wave effect? {#faq-close-wave-effect}

If you don't need this feature, you can set `disabled` of `wave` in [ConfigProvider](#) as `true`.

```

<ConfigProvider wave={{ disabled: true }}>
  <Button>click</Button>
</ConfigProvider>

```