

# Skeleton

Provide a placeholder while you wait for content to load, or to visualize content that doesn't exist yet.

## When To Use

- When a resource needs long time to load.
- When the component contains lots of information, such as List or Card.
- Only works when loading data for the first time.
- Could be replaced by Spin in any situation, but can provide a better user experience.

## Examples



### Basic

Simplest Skeleton usage.

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

const App: React.FC = () => <Skeleton />;

export default App;
```

### Complex combination

Complex combination with avatar and multiple paragraphs.

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

const App: React.FC = () => <Skeleton avatar paragraph={{ rows: 4 }} />;

export default App;
```

### Active Animation

Display active animation.

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

const App: React.FC = () => <Skeleton active />;

export default App;
```



Active: ☐

Button and Input Block: ☐

Size: Default Large Small

Button Shape: Default Square Round Circle

Avatar Shape: Square Circle

### Button/Avatar/Input/Image/Node

Skeleton Button, Avatar, Input, Image and Node.

```
import { DotChartOutlined } from '@ant-design/icons';
import type { RadioChangeEvent } from 'antd';
import { Divider, Form, Radio, Skeleton, Space, Switch } from 'antd';
import React, { useState } from 'react';

type SizeType = 'default' | 'small' | 'large';
type ButtonShapeType = 'circle' | 'square' | 'round' | 'default';
type AvatarShapeType = 'circle' | 'square';

const App: React.FC = () => {
  const [active, setActive] = useState(false);
  const [block, setBlock] = useState(false);
  const [size, setSize] = useState<SizeType>('default');
  const [buttonShape, setButtonShape] = useState<ButtonShapeType>('default');
  const [avatarShape, setAvatarShape] = useState<AvatarShapeType>('circle');

  const handleActiveChange = (checked: boolean) => {
    setActive(checked);
  };

  const handleBlockChange = (checked: boolean) => {
    setBlock(checked);
  };

  const handleSizeChange = (e: RadioChangeEvent) => {
    setSize(e.target.value);
  };

  const handleShapeButton = (e: RadioChangeEvent) => {
    setButtonShape(e.target.value);
  };

  const handleAvatarShape = (e: RadioChangeEvent) => {
    setAvatarShape(e.target.value);
  };
}
```

## Ant Design, a design language

We supply a series of design principles, practical patterns and high quality design resources (Sketch and Axure), to help people create their product prototypes beautifully and efficiently.

Show Skeleton

### Contains sub component

Skeleton contains sub component.

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

const App: React.FC = () => {
  const [loading, setLoading] = useState<boolean>(false);

  const showSkeleton = () => {
    setLoading(true);
    setTimeout(() => {
      setLoading(false);
    }, 3000);
  };

  return (
    <Space direction="vertical" style={{ width: '100%' }} size={16}>
      <Skeleton loading={loading}>
        <h4 style={{ marginBottom: 16 }}>Ant Design, a design language</h4>
        <p>
          We supply a series of design principles, practical patterns and high quality design
          resources (Sketch and Axure), to help people create their product prototypes beautiful
          and efficiently.
        </p>
      </Skeleton>
      <Button onClick={showSkeleton} disabled={loading}>
        Show Skeleton
      </Button>
    </Space>
  );
};

export default App;
```

```
      <Radio.Button value="square">Square</Radio.Button>
      <Radio.Button value="circle">Circle</Radio.Button>
    </Radio.Group>
  </Form.Item>
  <Form.Item label="Avatar Shape">
    <Radio.Group value={avatarShape} onChange={handleAvatarShape}>
      <Radio.Button value="square">Square</Radio.Button>
      <Radio.Button value="circle">Circle</Radio.Button>
    </Radio.Group>
  </Form.Item>
</Space>
</Form>
```



## List

Use skeleton in list component.

```
import type Icon from '@ant-design/icons';
import { LikeOutlined, MessageOutlined, StarOutlined } from '@ant-design/icons';
import { Avatar, List, Skeleton, Switch } from 'antd';
import React, { useState } from 'react';

interface IconTextProps {
  icon: typeof Icon;
  text: React.ReactNode;
}

const listData = Array.from({ length: 3 }).map((_, i) => ({
  href: 'https://ant.design',
  title: `ant design part ${i + 1}`,
  avatar: `https://xsgames.co/randomusers/avatar.php?g=pixel&key=${i}`,
  description:
    'Ant Design, a design language for background applications, is refined by Ant UED Team.',
  content:
    'We supply a series of design principles, practical patterns and high quality design resource
}));

const IconText: React.FC<IconTextProps> = ({ icon, text }) => (
  <>
    {React.createElement(icon, { style: { marginRight: 8 } })}
    {text}
  </>
);

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

  const onChange = (checked: boolean) => {
    setLoading(!checked);
  }
}
```

```
return (
  <Switch checked={!loading} onChange={onChange} style={{ marginBottom: 16 }} />
  <List
    itemLayout="vertical"
    size="large"
    dataSource={listData}
    renderItem={(item) => (
      <List.Item
        key={item.title}
        actions={
          !loading
        }
      />
    )}
  />
)
```

```
>
<Skeleton loading={loading} active avatar>
  <List.Item.Meta
    avatar=<Avatar src={item.avatar}>/>

```

```
},  
  
export default App;
```

## SkeletonParagraphProps

## SkeletonParagraphProps

Property	Description	Type	Default
rows	Set the row count of paragraph	number	-
width	Set the width of paragraph. When width is an Array, it can set the width of each row. Otherwise only set the last row width	number   string   Array<number   string>	-

### SkeletonButtonProps

Property	Description	Type	Default	Version
active	Show animation effect	boolean	false	
block	Option to fit button width to its parent width	boolean	false	4.17.0
shape	Set the shape of button	circle   round   square   default	-	
size	Set the size of button	large   small   default	-	

### SkeletonInputProps

Property	Description	Type	Default
active	Show animation effect	boolean	false
size	Set the size of input	large   small   default	-

## Design Token

#### ▼ Global Token

Token Name	Description	Type	Default Value
colorFill	The darkest fill color is used to distinguish between the second and third level of fill color, and is currently only used in the hover effect of Slider.	string	<code>rgba(0, 0, 0, 0.15)</code>
colorFillContent	Control the background color of content area.	string	<code>rgba(0, 0, 0, 0.06)</code>
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

Token Name	Description	Type	Default Value
controlHeightSM	SM component height	number	24
controlHeightXS	XS component height	number	16
marginLG	Control the margin of an element, with a large size.	number	24
marginSM	Control the margin of an element, with a medium-small size.	number	12
marginXXS	Control the margin of an element, with the smallest size.	number	4
padding	Control the padding of the element.	number	16