

# Anchor

Hyperlinks to scroll on one page.

## When To Use

For displaying anchor hyperlinks on page and jumping between them.

Notes for developers

After version `4.24.0`, we rewrite Anchor use FC, Some methods of obtaining `ref` and calling internal instance methods will invalid.

## Examples

## Basic

The simplest usage.

```
import React from 'react';
import { Anchor, Row, Col } from 'antd';

const App: React.FC = () => (
  <Row>
    <Col span={16}>
      <div id="part-1" style={{ height: '100vh', background: 'rgba(255,0,0,0.02)' }} />
      <div id="part-2" style={{ height: '100vh', background: 'rgba(0,255,0,0.02)' }} />
      <div id="part-3" style={{ height: '100vh', background: 'rgba(0,0,255,0.02)' }} />
    </Col>
    <Col span={8}>
      <Anchor
        items={[
          {
            key: 'part-1',
            href: '#part-1',
            title: 'Part 1',
          },
          {
            key: 'part-2',
            href: '#part-2',
            title: 'Part 2',
          },
          {
            key: 'part-3',
            href: '#part-3',
            title: 'Part 3',
          },
        ]}
      />
    </Col>
  </Row>
);

export default App;
```

## Horizontal Anchor

Horizontally aligned anchors

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

const App: React.FC = () => (
  <>
    <div style={{ padding: '20px' }}>
      <Anchor
        direction="horizontal"
        items={[
          {
            key: 'part-1',
            href: '#part-1',
            title: 'Part 1',
          },
          {
            key: 'part-2',
            href: '#part-2',
            title: 'Part 2',
          },
          {
            key: 'part-3',
            href: '#part-3',
            title: 'Part 3',
          },
          {
            key: 'part-4',
            href: '#part-4',
            title: 'Part 4',
          },
          {
            key: 'part-5',
            href: '#part-5',
            title: 'Part 5',
          },
          {
            key: 'part-6',
            href: '#part-6',
            title: 'Part 6',
          },
        ]}
      />
    </div>
    <div>
      <div
        id="part-1"
        style={{
          width: '100vw',
          height: '100vh',
          textAlign: 'center',
          background: 'rgba(0,255,0,0.02)',
        }}
      />
      <div
        id="part-2"
        style={{
          width: '100vw',
          height: '100vh',
          textAlign: 'center',
          backaround: 'raba(0.0.255.0.02)',
        }}
      />
    </div>
  </>
);
```

- Basic demo
- Static demo
- API
  - Anchor Props
  - Link Props

## Static Anchor [✎](#)

Do not change state when page is scrolling.

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

const App: React.FC = () => (
  <Anchor
    affix={false}
    items={[
      {
        key: '1',
        href: '#components-anchor-demo-basic',
        title: 'Basic demo',
      },
      {
        key: '2',
        href: '#components-anchor-demo-static',
        title: 'Static demo',
      },
      {
        key: '3',
        href: '#api',
        title: 'API',
        children: [
          {
            key: '4',
            href: '#anchor-props',
            title: 'Anchor Props',
          },
          {
            key: '5',
            href: '#link-props',
            title: 'Link Props',
          },
        ],
      },
    ]},
  />
);

export default App;
```

- Basic demo
- Static demo
- API
  - Anchor Props
  - Link Props

## Customize the onClick event

Clicking on an anchor does not record history.

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

const handleClick = (
  e: React.MouseEvent<HTMLDivElement>,
  link: {
    title: React.ReactNode;
    href: string;
  },
) => {
  e.preventDefault();
  console.log(link);
};

const App: React.FC = () => (
  <Anchor
    affix={false}
    onClick={handleClick}
    items={[
      {
        key: '1',
        href: '#components-anchor-demo-basic',
        title: 'Basic demo',
      },
      {
        key: '2',
        href: '#components-anchor-demo-static',
        title: 'Static demo',
      },
      {
        key: '3',
        href: '#api',
        title: 'API',
        children: [
          {
            key: '4',
            href: '#anchor-props',
            title: 'Anchor Props',
          },
          {
            key: '5',
            href: '#link-props',
            title: 'Link Props',
          },
        ],
      },
    ]},
  </>
);

export default App;
```

- Basic demo
- Static demo
- API
  - Anchor Props
  - Link Props

## Customize the anchor highlight

Customize the anchor highlight.

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

const getCurrentAnchor = () => '#components-anchor-demo-static';

const App: React.FC = () => (
  <Anchor
    affix={false}
    getCurrentAnchor={getCurrentAnchor}
    items={[
      {
        key: '1',
        href: '#components-anchor-demo-basic',
        title: 'Basic demo',
      },
      {
        key: '2',
        href: '#components-anchor-demo-static',
        title: 'Static demo',
      },
      {
        key: '3',
        href: '#api',
        title: 'API',
        children: [
          {
            key: '4',
            href: '#anchor-props',
            title: 'Anchor Props',
          },
          {
            key: '5',
            href: '#link-props',
            title: 'Link Props',
          },
        ],
      },
    ]},
  </>
);

export default App;
```

## Set Anchor scroll offset

Anchor target scroll to screen center.

```
import React, { useEffect, useState } from 'react';
import { Anchor, Row, Col } from 'antd';

const App: React.FC = () => {
  const topRef = React.useRef<HTMLDivElement>(null);
  const [targetOffset, setTargetOffset] = useState<number>();

  useEffect(() => {
    setTargetOffset(topRef.current?.clientHeight);
  }, []);

  return (
    <div>
      <Row>
        <Col span={18}>
          <div
            id="part-1"
            style={{ height: '100vh', background: 'rgba(255,0,0,0.02)', marginTop: '30vh' }}
          >
            Part 1
          </div>
          <div id="part-2" style={{ height: '100vh', background: 'rgba(0,255,0,0.02)' }}>
            Part 2
          </div>
          <div id="part-3" style={{ height: '100vh', background: 'rgba(0,0,255,0.02)' }}>
            Part 3
          </div>
        </Col>
        <Col span={6}>
          <Anchor
            targetOffset={targetOffset}
            items={[
              {
                key: 'part-1',
                href: '#part-1',
                title: 'Part 1',
              },
              {
                key: 'part-2',
                href: '#part-2',
                title: 'Part 2',
              },
              {
                key: 'part-3',
                href: '#part-3',
                title: 'Part 3',
              },
            ]}
          />
        </Col>
      </Row>

      <div
        style={{
          height: '30vh',
          background: 'rgba(0,0,0,0.85)',
          position: 'fixed',
          top: 0,
          left: 0,
        }}
      >
    </div>
  )
}
```

- Basic demo
- Static demo
- API
  - Anchor Props
  - Link Props

## Listening for anchor link change

Listening for anchor link change.

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

const onChange = (link: string) => {
  console.log('Anchor:OnChange', link);
};

const App: React.FC = () => (
  <Anchor
    affix={false}
    onChange={onChange}
    items={[
      {
        key: '1',
        href: '#components-anchor-demo-basic',
        title: 'Basic demo',
      },
      {
        key: '2',
        href: '#components-anchor-demo-static',
        title: 'Static demo',
      },
      {
        key: '3',
        href: '#api',
        title: 'API',
        children: [
          {
            key: '4',
            href: '#anchor-props',
            title: 'Anchor Props',
          },
          {
            key: '5',
            href: '#link-props',
            title: 'Link Props',
          },
        ],
      },
    ]}
  />
);

export default App;
```

## API

### Anchor Props



| Property                      | Description   | Type   | Default                      | Version |
|-------------------------------|---|--|------------------------------|---------|
| <code>affix</code>            | Fixed mode of Anchor  | <code>boolean</code>   | <code>true</code>            |         |
| <code>bounds</code>           | Bounding distance of anchor area  | <code>number</code>  | <code>5</code>               |         |
| <code>getContainer</code>     | Scrolling container   | <code>() =&gt; HTMLElement</code>  | <code>() =&gt; window</code> |         |
| <code>getCurrentAnchor</code> | Customize the anchor highlight  | <code>(activeLink: string) =&gt; string</code>                             | <code>-</code>               |         |
| <code>offsetTop</code>        | Pixels to offset from top when calculating position of scroll                     | <code>number</code>  | <code>0</code>               |         |
| <code>showInkInFixed</code>   | Whether show ink-square when <code>affix={false}</code>                           | <code>boolean</code>   | <code>false</code>           |         |
| <code>targetOffset</code>     | Anchor scroll offset, default as <code>offsetTop</code> , <a href="#">example</a> | <code>number</code>  | <code>-</code>               |         |
| <code>onChange</code>         | Listening for anchor link change  | <code>(currentActiveLink: string) =&gt; void</code>                        |                              |         |
| <code>onClick</code>          | Set the handler to handle <code>click</code> event                                | <code>(e: MouseEvent, link: object) =&gt; void</code>                      | <code>-</code>               |         |
| <code>items</code>            | Data configuration option content, support nesting through children               | <code>{ key, href, title, target, children } []</code> <a href="#">see</a> | <code>-</code>               | 5.1.0   |
| <code>direction</code>        | Set Anchor direction  | <code>vertical</code>   <code>horizontal</code>                            | <code>vertical</code>        | 5.2.0   |

## AnchorItem

| Property              | Description  | Type                                      | Default        | Version |
|-----------------------|--|---|----------------|---------|
| <code>key</code>      | The unique identifier of the Anchor Link   | <code>string</code>   <code>number</code> | <code>-</code> |         |
| <code>href</code>     | The target of hyperlink  | <code>string</code>                       |                |         |
| <code>target</code>   | Specifies where to display the linked URL  | <code>string</code>                       |                |         |
| <code>title</code>    | The content of hyperlink   | <code>ReactNode</code>                    |                |         |
| <code>children</code> | Nested Anchor Link, <code>Attention:</code> This attribute does not support horizontal orientation | <a href="#">AnchorItem</a> []             | <code>-</code> |         |

## Link Props

We recommend using the items form instead.

| Property | Description                               | Type      | Default | Version |
|----------|---|-----------|---------|---------|
| href     | The target of hyperlink                   | string    |         |         |
| target   | Specifies where to display the linked URL | string    |         |         |
| title    | The content of hyperlink                  | ReactNode |         |         |

## Design Token

### ▼ Global Token

| Token Name         | Description   | Type   | Default Value   |
|--------------------|---|--------|---|
| 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   |
| colorSplit         | Used as the color of separator, this color is the same as colorBorderSecondary but with transparency.   | string | rgba(5, 5, 5, 0.06)   |
| colorText          | Default text color which comply with W3C standards, and this color is also the darkest neutral color.   | string | rgba(0, 0, 0, 0.88)   |
| fontFamily         | The font family of Ant Design prioritizes the default interface font of the system, and provides a set of alternative font libraries that are suitable for screen display to maintain the readability and readability of the font under different platforms and browsers, reflecting the friendly, stable and professional characteristics. | string | -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, 'Helvetica Neue', Arial, 'Noto Sans', sans-serif, 'Apple Color Emoji', 'Segoe UI Emoji', 'Segoe UI Symbol', 'Noto Color Emoji' |
| 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   |
| lineWidthBold      | The default line width of the outline class components, such as Button, Input, Select, etc.   | number | 2   |
| motionDurationSlow | Motion speed, slow speed. Used for large element animation interaction.   | string | 0.3s  |
| padding            | Control the padding of the element.   | number | 16  |

| Token Name | Description   | Type              | Default Value |
|------------|---|-------------------|---------------|
| paddingXXS | Control the extra extra small padding of the element. | <div>number</div> | 4             |