何时使用 {#when-to-use}

需要展现当前页面上可供跳转的锚点链接，以及快速在锚点之间跳转。

> *开发者注意事项：*
>
> *自 `4.24.0` 起，由于组件从 class 重构成 FC，之前一些获取 `ref` 并调用内部实例方法的写法都会失效*

代码演示

基本

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

横向 Anchor

```
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',
          },
        ]}
      />
    </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',
          background: 'rgba(0,0,255,0.02)',
        }}
      />
      <div
        id="part-3"
        style={{ width: '100vw', height: '100vh', textAlign: 'center',
background: '#FFFBE9' }}
      />
    </div>
  </>
);

export default App;
```

静态位置

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

const App: React.FC = () => (
  <Anchor
    affix={false}
    items={[
      {
        key: '1',
        href: '#anchor-demo-basic',
        title: 'Basic demo',
      },
      {
        key: '2',
        href: '#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;
```

自定义 onClick 事件

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

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

const App: React.FC = () => (
  <Anchor
    affix={false}
    onClick={handleClick}
    items={[
      {
        key: '1',
        href: '#anchor-demo-basic',
        title: 'Basic demo',
      },
      {
        key: '2',
        href: '#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;
```

自定义锚点高亮

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

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

const App: React.FC = () => (
  <Anchor
    affix={false}
    getCurrentAnchor={getCurrentAnchor}
    items={[
      {
        key: '1',
        href: '#anchor-demo-basic',
        title: 'Basic demo',
      },
      {
        key: '2',
        href: '#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;
```

设置锚点滚动偏移量

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

const style: React.CSSProperties = {
  height: '30vh',
  backgroundColor: 'rgba(0, 0, 0, 0.85)',
  position: 'fixed',
  top: 0,
  insetInlineStart: 0,
  width: '75%',
  color: '#fff',
};

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={style} ref={topRef}>
        <div>Fixed Top Block</div>
      </div>
    </div>
  );
};

export default App;
```

监听锚点链接改变

```
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: '#anchor-demo-basic',
        title: 'Basic demo',
      },
      {
        key: '2',
        href: '#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;
```

替换历史中的 href

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

废弃的 JSX 示例

Debug

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

const { Link } = Anchor;

const App: React.FC = () => (
  <Anchor affix={false}>
```

```
    <Link href="#anchor-demo-basic" title="Basic demo" />
    <Link href="#anchor-demo-static" title="Static demo" />
    <Link href="#api" title="API">
      <Link href="#anchor-props" title="Anchor Props" />
      <Link href="#link-props" title="Link Props" />
    </Link>
  </Anchor>
);

export default App;
```

组件 Token

Debug

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

/** Test usage. Do not use in your production. */

export default () => (
  <ConfigProvider
    theme={{
      components: {
        Anchor: {
          linkPaddingBlock: 100,
          linkPaddingInlineStart: 50,
        },
      },
    }}
  >
    <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>
  </ConfigProvider>
);
```

API

通用属性参考：通用属性

Anchor Props

参数	说明	类型	默认值	版本
affix	固定模式	boolean | Omit<AffixProps, 'offsetTop' | 'target' | 'children'>	true	object: 5.19.0
bounds	锚点区域边界	number	5	
getContainer	指定滚动的容器	() => HTMLElement	() => window	
getCurrentAnchor	自定义高亮的锚点	(activeLink: string) => string	-	
offsetTop	距离窗口顶部达到指定偏移量后触发	number		
showInkInFixed	`affix={false}` 时是否显示小方块	boolean	false	

targetOffset	锚点滚动偏移量，默认与 offsetTop 相同，例子	number	-	
onChange	监听锚点链接改变	(currentActiveLink: string) => void	-	
onClick	`click` 事件的 handler	(e: MouseEvent, link: object) => void	-	
items	数据化配置选项内容，支持通过 children 嵌套	{ key, href, title, target, children }[] 具体见	-	5.1.0
direction	设置导航方向	`vertical` | `horizontal`	`vertical`	5.2.0
replace	替换浏览器历史记录中项目的 href 而不是推送它	boolean	false	5.7.0

AnchorItem

参数	说明	类型	默认值	版本
key	唯一标志	string | number	-	
href	锚点链接	string	-	
target	该属性指定在何处显示链接的资源	string	-	
title	文字内容	ReactNode	-	
children	嵌套的 Anchor Link，注意：水平方向该属性不支持	AnchorItem[]	-	
replace	替换浏览器历史记录中的项目 href 而不是推送它	boolean	false	5.7.0

Link Props

建议使用 items 形式。

参数	说明	类型	默认值	版本
href	锚点链接	string	-	
target	该属性指定在何处显示链接的资源	string	-	

title	文字内容	ReactNode	-	

主题变量（Design Token）