

何时使用 {#when-to-use}

当内容区域比较长，需要滚动页面时，这部分内容对应的操作或者导航需要在滚动范围内始终展现。常用于侧边菜单和按钮组合。

页面可视范围过小时，慎用此功能以免出现遮挡页面内容的情况。

开发者注意事项：

自 5.10.0 起，由于 Affix 组件由 class 重构为 FC，之前获取 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 { Affix, Button } from 'antd';

const App: React.FC = () => {
  const [top, setTop] = React.useState<number>(100);
  const [bottom, setBottom] = React.useState<number>(100);
  return (
    <>
      <Affix offsetTop={top}>
        <Button type="primary" onClick={() => setTop(top + 10)}>
          Affix top
        </Button>
      </Affix>
      <br />
      <Affix offsetBottom={bottom}>
        <Button type="primary" onClick={() => setBottom(bottom + 10)}>
          Affix bottom
        </Button>
      </Affix>
    </>
  );
};

export default App;
```

固定状态改变的回调

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

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

```

    <Affix offsetTop={120} onChange={({affixed}) => console.log(affixed)}>
      <Button>120px to affix top</Button>
    </Affix>
  );

export default App;

```

滚动容器

```

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

const containerStyle: React.CSSProperties = {
  width: '100%',
  height: 100,
  overflow: 'auto',
  boxShadow: '0 0 0 1px #1677ff',
  scrollbarWidth: 'thin',
  scrollbarGutter: 'stable',
};

const style: React.CSSProperties = {
  width: '100%',
  height: 1000,
};

const App: React.FC = () => {
  const [container, setContainer] = React.useState<HTMLDivElement | null>(null);
  return (
    <div style={containerStyle} ref={setContainer}>
      <div style={style}>
        <Affix target={() => container}>
          <Button type="primary">Fixed at the top of container</Button>
        </Affix>
      </div>
    </div>
  );
};

export default App;

```

调整浏览器大小，观察 Affix 容器是否发生变化。跟随变化为正常。#17678

Debug

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

const App: React.FC = () => {
  const [top, setTop] = useState(10);

  return (
    <div style={{ height: 10000 }}>
      <div>Top</div>
      <Affix offsetTop={top}>
        <div style={{ background: 'red' }}>
          <Button type="primary" onClick={() => setTop(top + 10)}>
            Affix top
          </Button>
        </div>
      </Affix>
      <div>Bottom</div>
    </div>
  );
};

export default App;
```

API

通用属性参考：[通用属性](#)

参数	说明	类型	默认值
offsetBottom	距离窗口底部达到指定偏移量后触发	number	-
offsetTop	距离窗口顶部达到指定偏移量后触发	number	0
target	设置 Affix 需要监听其滚动事件的元素， 值为一个返回对应 DOM 元素的函数	() => HTMLElement	() => window
onChange	固定状态改变时触发的回调函数	(affixed?: boolean) => void	-

注意： Affix 内的元素不要使用绝对定位，如需要绝对定位的效果，可以直接设置 Affix 为绝对定位：

```
<Affix style={{ position: 'absolute', top: y, left: x }}>...</Affix>
```

FAQ

Affix 使用 `target` 绑定容器时，元素会跑到容器外。

从性能角度考虑，我们只监听容器滚动事件。如果希望任意滚动，你可以在窗体添加滚动监听：

<https://codesandbox.io/s/stupefied-maxwell-ophqnm?file=/index.js>

相关 issue: [#3938](#) [#5642](#) [#16120](#)

Affix 在水平滚动容器中使用时，元素 `left` 位置不正确。

Affix 一般只适用于单向滚动的区域，只支持在垂直滚动容器中使用。如果希望在水平容器中使用，你可以考虑使用 原生 `position: sticky` 实现。

相关 issue: [#29108](#)