Affix

Wrap Affix around another component to make it stick the viewport.

When To Use

On longer web pages, it's helpful to stick component into the viewport. This is common for menus and actions.

Please note that Affix should not cover other content on the page, especially when the size of the viewport is small.

Examples

Basic

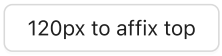
The simplest usage.

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

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

  return (
    <>
      <Affix offsetTop={top}>
        <Button type="primary" onClick={() =>
          Affix top
        </Button>
      </Affix>
      <br />
      <Affix offsetBottom={bottom}>
        <Button type="primary" onClick={() =>
          Affix bottom
        </Button>
      </Affix>
    </>
  );
};

export default App;
```



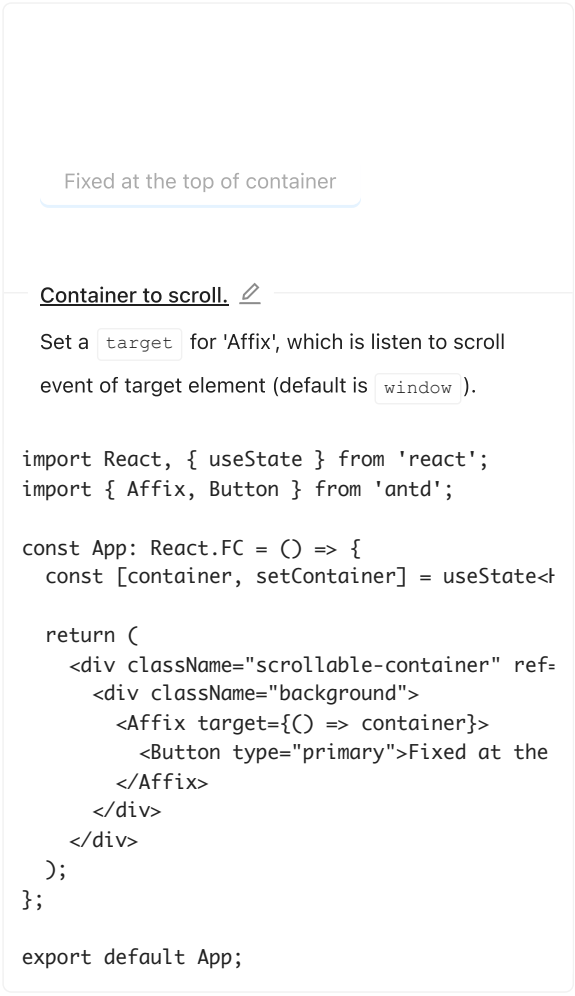
Callback

Callback with affixed state.

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

const App: React.FC = () => (
  <Affix offsetTop={120} onChange={(affixed) =
    <Button>120px to affix top</Button>
  </Affix>
);

export default App;
```



Container to scroll.

Set a `target` for 'Affix', which is listen to scroll event of target element (default is `window`).

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

const App: React.FC = () => {
  const [container, setContainer] = useState<H

  return (
    <div className="scrollable-container" ref=
      <div className="background">
        <Affix target={() => container}>
          <Button type="primary">Fixed at the
        </Affix>
      </div>
    </div>
  );
};

export default App;
```

API

Property	Description	Type	Default
offsetBottom	Offset from the bottom of the viewport (in pixels)	number	-
offsetTop	Offset from the top of the viewport (in pixels)	number	0
target	Specifies the scrollable area DOM node	() => HTMLElement	() => window
onChange	Callback for when Affix state is changed	(affixed?: boolean) => void	-

Note: Children of `Affix` must not have the property `position: absolute`, but you can set `position: absolute` on `Affix` itself:

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

FAQ

When binding container with `target` in Affix, elements sometimes move out of the container.

We only listen to container scroll events for performance consideration. You can add custom listeners if you still want to: https://codesandbox.io/s/2xyj5zr85p

Related issues：#3938 #5642 #16120

When Affix is used in a horizontal scroll container, the position of the element `left` is incorrect.

Affix is generally only applicable to areas with one-way scrolling, and only supports usage in vertical scrolling containers. If you want to use it in a horizontal container, you can consider implementing with the native `position: sticky` property.

Related issues：#29108