



链滴

React Visual - 轮播组件

作者: [Vanessa](#)

原文链接: <https://ld246.com/article/1569646759044>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

描述

渲染一个轮播组件。

- 使用 `React.useState()` hook 创建一个 `active` 状态变量，设置默认值为 `0` (第一个元素的索引)。
- 使用一个 `style` 对象分别为组件保存各自的样式
- 使用 `React.useEffect()` hook 更新 `active` 值为下一个元素的索引，可以使用 `setInterval`
- 解构 `props`，根据 `active` 和 `index` 对轮播组件中的每一项进行可见样式的设置，并将其和对应样式进行组合后赋值给轮播组件中对应的每一项
- 使用 `React.cloneElement()` 渲染轮播组件中的每一项，并传递剩余的 `props` 和计算好的样式

实现

```
function Carousel(props) {
  const [active, setActive] = React.useState(0);
  let scrollInterval = null;
  const style = {
    carousel: {
      position: 'relative'
    },
    carouselItem: {
      position: 'absolute',
      visibility: 'hidden'
    },
    visible: {
      visibility: 'visible'
    }
  };
  React.useEffect(() => {
    scrollInterval = setInterval(() => {
      const { carouselItems } = props;
      setActive((active + 1) % carouselItems.length);
    }, 2000);
    return () => clearInterval(scrollInterval);
  });
  const { carouselItems, ...rest } = props;
  return (
    <div style={style.carousel}>
      {carouselItems.map((item, index) => {
        const activeStyle = active === index ? style.visible : {};
        return React.cloneElement(item, {
          ...rest,
          style: {
            ...style.carouselItem,
            ...activeStyle
          }
        });
      })}
    </div>
  );
}
```

```
    </div>
  );
}
```

使用

```
ReactDOM.render(
  <Carousel
    carouselItems={[
      <div>carousel item 1</div>,
      <div>carousel item 2</div>,
      <div>carousel item 3</div>
    ]}
  />,
  document.getElementById('root')
);
```

返回总目录

[每天 30 秒系列之 React](#)