



链滴

React Visual - 带波浪效果的按钮

作者: [Vanessa](#)

原文链接: <https://ld246.com/article/1570836908451>

来源网站: [链滴](#)

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

2019-10-12

描述

渲染一个当点击后有波浪动画效果的按钮。

- 为波浪效果定义一些适当的 CSS 样式和动画
- 使用 `React.useState()` hook 为坐标和按钮的动画状态创建变量
- 使用 `React.useEffect()` hook 来更新动画状态，每当坐标状态变量修改时，就开始进行动画
- 在动画播放结束后，使用前一个 hook 中的 `Use setTimeout()` 来对动画进行清除
- 每当 `isRippling` 状态变量为 `false` 时，再次使用 `React.useEffect()` hook 对 `coords` 进行重置
- `onClick` 事件中需要更新 `coords` 状态变量并调用传递的回调方法
- 最后渲染一个包含一个或两个 `` 元素的 `<button>` 元素，基于 `coords` 状态变量设置 `.ripple` 元素的位置

实现

```
.ripple-button {  
  border-radius: 4px;  
  border: none;  
  margin: 8px;  
  padding: 14px 24px;  
  background: #1976d2;  
  color: #fff;  
  overflow: hidden;  
  position: relative;  
  cursor: pointer;  
}  
  
.ripple-button > .ripple {  
  width: 20px;  
  height: 20px;  
  position: absolute;  
  background: #63a4ff;  
  display: block;  
  content: "";  
  border-radius: 9999px;  
  opacity: 1;  
  animation: 1.2s ease 1 forwards ripple-effect;  
}  
  
@keyframes ripple-effect {  
  0% {  
    transform: scale(1);  
    opacity: 1;  
  }  
  50% {  
    transform: scale(10);  
    opacity: 0.375;  
  }  
}
```

```
}

100% {
  transform: scale(35);
  opacity: 0;
}

.ripple-button > .content {
  position: relative;
  z-index: 2;
}

function RippleButton({ children, onClick }) {
  const [coords, setCoords] = React.useState({ x: -1, y: -1 });
  const [isRippling, setIsRippling] = React.useState(false);

  React.useEffect(
    () => {
      if (coords.x !== -1 && coords.y !== -1) {
        setIsRippling(true);
        setTimeout(() => setIsRippling(false), 1200);
      } else setIsRippling(false);
    },
    [coords]
  );

  React.useEffect(
    () => {
      if (!isRippling) setCoords({ x: -1, y: -1 });
    },
    [isRippling]
  );

  return (
    <button
      className="ripple-button"
      onClick={e => {
        var rect = e.target.getBoundingClientRect();
        var x = e.clientX - rect.left;
        var y = e.clientY - rect.top;
        setCoords({ x, y });
        onClick && onClick(e);
      }}
    >
    {isRippling ? (
      <span
        className="ripple"
        style={{
          left: coords.x + 10,
          top: coords.y
        }}
      />
    ) : (
      ""
    )}
  );
}
```

```
        )}
      <span className="content">{children}</span>
    </button>
  );
}
```

使用

```
ReactDOM.render(
  <RippleButton onClick={e => console.log(e)}>Click me</RippleButton>,
  document.getElementById('root')
);
```

返回总目录

[每天 30 秒系列之 React](#)