



链滴

React Visual - 时钟倒数计时器 (advanced)

作者: [Vanessa](#)

原文链接: <https://ld246.com/article/1569721187385>

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

渲染一个倒数计时器，当其为 0 时打印一条消息。

- 使用对象解构设置默认的 `hours`、`minutes` 和 `seconds` 属性
- 使用 `React.useState()` hook 创建 `time`、`paused` 和 `over` 状态变量，他们的默认值分别为传入属性、`false` 和 `false`
- 创建一个 `tick` 方法，基于当前的值更新 `time` 值（如：每次减少 1 秒）
- 如果 `paused` 或者 `over` 的值为 `true`，`tick` 方法立即返回
- 创建一个 `reset` 方法，重置所有的状态变量为他们的初始默认值
- 在 `React.useEffect()` hook 中，通过使用 `setInterval()`，每隔一秒对 `tick` 方法进行一次调用。当件卸载时，使用 `clearInterval()` 进行清除
- 将用于展示组件 `time` 状态变量的文本 `<p>` 元素，用于暂停/恢复的按钮，用于重置的按钮，用展示计时器为 0 时使用的 `<div>` 元素都使用 `<div>` 进行包裹
- 如果 `over` 的值为 `true`，计时器将显示一条消息

实现

```
function Countdown({ hours = 0, minutes = 0, seconds = 0 }) {
  const [paused, setPaused] = React.useState(false);
  const [over, setOver] = React.useState(false);
  const [time, setTime] = React.useState({
    hours: parseInt(hours),
    minutes: parseInt(minutes),
    seconds: parseInt(seconds)
  });

  const tick = () => {
    if (paused || over) return;
    if (time.hours === 0 && time.minutes === 0 && time.seconds === 0) setOver(true);
    else if (time.minutes === 0 && time.seconds === 0)
      setTime({
        hours: time.hours - 1,
        minutes: 59,
        seconds: 59
      });
    else if (time.seconds === 0)
      setTime({
        hours: time.hours,
        minutes: time.minutes - 1,
        seconds: 59
      });
    else
      setTime({
        hours: time.hours,
        minutes: time.minutes,
        seconds: time.seconds - 1
      });
  };
}
```

```
    });  
  };  
  
  const reset = () => {  
    setTime({  
      hours: parseInt(hours),  
      minutes: parseInt(minutes),  
      seconds: parseInt(seconds)  
    });  
    setPaused(false);  
    setOver(false);  
  };  
  
  React.useEffect(() => {  
    let timerID = setInterval(() => tick(), 1000);  
    return () => clearInterval(timerID);  
  });  
  
  return (  
    <div>  
      <p>`${time.hours.toString().padStart(2, '0')}:${time.minutes  
        .toString()  
        .padStart(2, '0')}:${time.seconds.toString().padStart(2, '0')}`</p>  
      <div>{over ? "Time's up!" : ""}</div>  
      <button onClick={() => setPaused(!paused)}>{paused ? 'Resume' : 'Pause'}</button>  
      <button onClick={() => reset()}>Restart</button>  
    </div>  
  );  
}
```

使用

```
ReactDOM.render(<CountDown hours="1" minutes="45" />, document.getElementById('root'  
));
```

返回总目录

[每天 30 秒系列之 React](#)