

# Thực hành sử dụng thuộc tính props và state trong React

## props

Trong React, **props** là một cách để truyền dữ liệu từ component cha đến component con.

Hãy tạo một ứng dụng React đơn giản để hiển thị danh sách các sản phẩm điện thoại di động. Mỗi sản phẩm sẽ có tên, giá và mô tả. Sử dụng **props** để truyền thông tin sản phẩm từ component cha đến component con.

### Hướng dẫn: Hoàn thiện các component và tạo thành một ứng dụng hoàn chỉnh từ các hướng dẫn sau đây

**1. Tạo Component Con (Product.js):** Tạo một component con để hiển thị thông tin của sản phẩm. Component này nhận thông tin sản phẩm thông qua **props** và hiển thị tên, giá và mô tả của sản phẩm.

```
const Product = (props) => {  
  return (  
    <div className="product">  
      <h2>{props.name}</h2>  
      <p>Giá: {props.price}</p>  
      <p>{props.description}</p>  
    </div>  
  );  
}
```

**2. Tạo Component Cha (ProductList.js):** Tạo một component cha để chứa danh sách các sản phẩm. Trong component này, bạn sẽ tạo một mảng chứa thông tin của các sản phẩm và truyền thông tin này đến component con **Product** thông qua **props**.

```
const ProductList = () => {
```

```

const products = [
  { name: 'Điện thoại A', price: 1000000, description: 'Mô tả sản phẩm A' },
  { name: 'Điện thoại B', price: 1500000, description: 'Mô tả sản phẩm B' },
  { name: 'Điện thoại C', price: 1200000, description: 'Mô tả sản phẩm C' }
];
return (
  <div className="product-list">
    {products.map((product, index) => (
      <Product key={index} name={product.name}
price={product.price} description={product.description} />
    ))}
  </div>
);
}

```

### 3. Tạo Component Gốc (App.js): Tạo một component gốc để render ProductList.

```

const App = () => {
  return (
    <div className="app">
      <h1>Danh sách sản phẩm</h1>
      <ProductList />
    </div>
  );
}

```

### 4. CSS (style.css): Tạo một tệp CSS để trang trí giao diện của ứng dụng.

```
.product {
  border: 1px solid #ccc;
  margin: 10px;
  padding: 10px;
}

.product-list {
  display: flex;
  flex-direction: column;
  align-items: center;
}
```

## state

Trong React, **state** là một trong những khái niệm cơ bản và quan trọng nhất. **state** là một đối tượng chứa dữ liệu và trạng thái của một component. Trạng thái này có thể thay đổi theo thời gian dựa trên các sự kiện hoặc tương tác của người dùng. Khi **state** của một component thay đổi, React sẽ tự động render lại component đó để hiển thị trạng thái mới.

Hãy tạo một ứng dụng React đơn giản để quản lý danh sách công việc. Ứng dụng sẽ cho phép người dùng thêm mới công việc, đánh dấu công việc đã hoàn thành và xóa công việc khỏi danh sách.

### **Hướng dẫn: Hoàn thiện các component và tạo thành một ứng dụng hoàn chỉnh từ các hướng dẫn sau đây**

**1. Tạo Component Gốc (App.js):** Tạo một component gốc để quản lý danh sách công việc. Sử dụng state để lưu trữ danh sách công việc.

```
const App = () => {
  const [tasks, setTasks] = useState([]);
  const [newTask, setNewTask] = useState('');

  const addTask = () => {
```

```
    if (newTask) {
      setTasks([...tasks, newTask]);
      setNewTask('');
    }
  };

  const removeTask = (index) => {
    const updatedTasks = tasks.filter((_, i) => i !== index);
    setTasks(updatedTasks);
  };

  return (
    <div className="app">
      <h1>Quản lý công việc</h1>
      <div>
        <input
          type="text"
          value={newTask}
          onChange={(e) => setNewTask(e.target.value)}
          placeholder="Nhập công việc mới"
        />
        <button onClick={addTask}>Thêm</button>
      </div>
      <ul>
        {tasks.map((task, index) => (
          <li key={index}>
```

```
        {task}
        <button onClick={() =>
removeTask(index)}>Xóa</button>
        </li>
    )})
</ul>
</div>
);
}
```

**2. CSS (style.css):** Tạo một tệp CSS để trang trí giao diện của ứng dụng.

```
.app {
  text-align: center;
  margin-top: 50px;
}

input {
  margin-right: 10px;
}
```

## Bài tập sử dụng props, state và các vấn đề khác trong React

### Bài Tập 1: Ứng Dụng Đếm Số

*Yêu Cầu Bài Tập:*

Hãy tạo một ứng dụng React đơn giản cho phép người dùng nhấn nút để tăng số lượng lên 1. Hiển thị số lượng đó lên giao diện.

*Hướng Dẫn:*

- Tạo một component React có một state để lưu trữ số lượng.
- Hiển thị số lượng từ state trên giao diện.
- Tạo một nút nhấn, khi người dùng nhấn vào nó, tăng giá trị trong state lên 1.

## **Bài Tập 2: Ứng Dụng Danh Bạ**

### *Yêu Cầu Bài Tập:*

Hãy xây dựng một ứng dụng danh bạ. Ứng dụng này nên cho phép người dùng thêm mới, xem danh sách, sửa đổi và xóa liên hệ.

### *Hướng Dẫn:*

- Tạo một component để hiển thị danh sách liên hệ.
- Tạo một form để thêm mới liên hệ (sử dụng state để lưu trữ dữ liệu nhập vào).
- Cho phép người dùng chỉnh sửa và xóa các liên hệ.

## **Bài Tập 3: Ứng Dụng Xem Phim**

### *Yêu Cầu Bài Tập:*

Hãy xây dựng một ứng dụng xem phim đơn giản. Hiển thị danh sách các bộ phim với tên, hình ảnh và mô tả. Khi người dùng chọn một bộ phim, hiển thị thông tin chi tiết về bộ phim đó.

### *Hướng Dẫn:*

- Tạo một danh sách các bộ phim với các thuộc tính như tên, hình ảnh, và mô tả.
- Hiển thị danh sách bộ phim trên trang chính của ứng dụng.
- Khi người dùng nhấp vào một bộ phim, hiển thị thông tin chi tiết về bộ phim đó (sử dụng state hoặc props để truyền thông tin).