

# Những điều cần biết về useState() hook

---

Presented by: Hau Nguyen | Easy Frontend

1. Giới thiệu hook `useState()`
2. Array destructuring syntax
3. So sánh `State` giữa class và functional component
4. Thêm ví dụ dùng `useState()`
5. Những điều lưu ý khi dùng `useState()`
6. Next step

## 1. Giới thiệu hook `useState()`

- Là một hook cơ bản.
- Giúp mình có thể dùng state trong functional component.
- Input: `initialState` (giá trị hoặc function)
- Output: một mảng có 2 phần tử tương ứng cho `state` và `setState`
- Cách dùng: `const [name, setName] = useState('Hau');`

## 2. Array destructuring syntax

```
// You know before
const array = ['Easy', 'Frontend'];
const a = array[0]; // Easy
const b = array[1]; // Frontend

// Now we use array destructuring
const [a, b] = ['Easy', 'Frontend'];
// a = 'Easy'
// b = 'Frontend'
```

```
// Fake useState() 🤖
function useState(initialState) {
  // do some mystery stuffs ...
  return [state, setState];
}

function ColorBox() {
  // Because useState() return an array of 2 items
  const [color, setColor] = useState('deeppink');
  return ...;
}
```

### 3. So sánh State giữa class và functional component

#### CLASS COMPONENT

```
class ColorBox extends PureComponent {
  constructor(props) {
    super(props);

    this.state = {
      shape: 'square',
      color: 'deeppink',
    };
  }

  handleBoxClick() {
    this.setState({ color: 'green' });
  };

  render() {
    const {color} = this.state;
    return (
      <div
        className="color-box"
        style={{ backgroundColor: color }}
        onClick={this.handleBoxClick}
      ></div>
    )
  }
}
```

#### FUNCTIONAL COMPONENT

```
function ColorBox() {
  const [shape, setShape] = useState('square');
  const [color, setColor] = useState('deeppink');

  function handleBoxClick() {
    setColor('green');
  }

  return (
    <div
      className="color-box"
      style={{ backgroundColor: color }}
      onClick={handleBoxClick}
    ></div>
  );
}
```

## 4. Thêm ví dụ dùng `useState()`

```
function TodoList() {
  const [todoList, setTodoList] = useState(['love', 'easy', 'frontend']);

  function removeTodo(index) {
    // Remember to clone into a new array
    const newTodoList = [...todoList];

    newTodoList.splice(index, 1);
    setTodoList(newTodoList);
  }

  return (
    <ul className="todo-list">
      {todoList.map((todo, index) => (
        <li
          key={todo.id}
          onClick={() => removeTodo(index)}
        >
          {todo.title}
        </li>
      ))}
    </ul>
  )
}
```

## 5. Những điều lưu ý khi dùng `useState()`

- `useState()` use REPLACING instead of MERGING.

```
// setState() in class component: MERGING
this.state = { name: 'Hau', color: 'red' };
this.setState({ color: 'green' });
// --> { name: 'Hau', color: 'green' }
```

```
// useState() in functional component: REPLACING
const [person, setPerson] = useState({ name: 'Hau', color: 'red' });
setPerson({ color: 'green' });
// --> { color: 'green' }
```

### SOLUTION

```
// useState() in functional component: REPLACING
const [person, setPerson] = useState({ name: 'Hau', color: 'red' });
setPerson({ ...person, color: 'green' });
// --> { name: 'Hau', color: 'green' }
```

- Initial state chỉ dùng cho lần đầu, những lần sau nó bị bỏ rơi.

```
function ColorBox() {
  // redundant code for sub-sequent re-render
  const initColor = getComplicatedColor();
  const [color, setColor] = useState(initColor);

  function handleBoxClick() {
    setColor('green');
  }

  return (
    <div
      className="color-box"
      style={{ backgroundColor: color }}
      onClick={handleBoxClick}
    ></div>
  );
}
```

- Initial state callback chỉ chạy một lần.

```
function ColorBox() {
  const [color, setColor] = useState(() => {
    // You're safe here
    // This function will be called once
    const initColor = getComplicatedColor();
    return initColor;
  });

  function handleBoxClick() {
    setColor('green');
  }

  return (
    <div
      className="color-box"
      style={{ backgroundColor: color }}
      onClick={handleBoxClick}
    ></div>
  );
}
```

## 6. Next step: useState() lab - ColorBox



Các bạn nhớ nè

- useState() giúp functional component có thể dùng state.
- useState() trả về một mảng 2 phần tử [name, setName].
- useState() áp dụng replacing thay vì merging như bên class component.
- Initial state callback chỉ thực thi 1 lần đầu.



Link tham khảo

- Introduction to react hooks: <https://reactjs.org/docs/hooks-intro.html>
- React hooks API reference: <https://reactjs.org/docs/hooks-reference.html>
- React hooks FAQ: <https://reactjs.org/docs/hooks-faq.html>