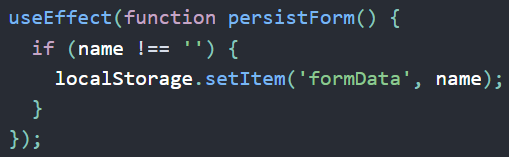
* Được thêm ở phiên bản React 16.8
* Cho phép Functional Component sử dụng lifecycle như ở Class Component

Chỉ hoạt động ở functional component

* Chia các component thành các subComponent dựa trên sự liên quan (chứ không phải nhóm vào từng method như trong lifecycle)
* **Tính năng**
  + Không bắt buộc phải dùng
  + 100% tương thích phiên bản cũ
  + Tái sử dụng code được
  + Các component được trình bày linh hoạt hơn
  + Viết code ngắn hơn
  + File build ra nhẹ hơn
* **Built-in Hook**
  + State hook
  + Effect hook
  + Context hook
  + Memo hook
  + Callback hook
  + Reducer hook
  + ...
  + Custom Hook
* **Note**
  + Có keyword **use** ở đầu (Coding convention)
  + Luôn gọi Hook ở trên cùng để đảm bảo hook được gọi theo đúng thứ tự trong các lần render
  + Chỉ gọi hook trong function của React (không gọi trong function của javascript)
  + Không gọi Hook trong loop, conditional, nested function

Nếu muốn gọi hook khi có điều kiện thì sẽ làm như sau



[*https://reactjs.org/docs/hooks-rules.html*](https://reactjs.org/docs/hooks-rules.html)

* **Tổng hợp hook**
  + State
    - Để quản lý 1 state
  + Reducer
    - Để quản lý 1 state có các action phức tạp
  + Effect
    - Thay thế cho componentDidMount(), componentDidUpdate(), componentWillUnmount()
    - Clean up & do not clean up
  + Context
    - Thay thế cho consumer trong context
  + Memo
    - Tối ưu function tính toán tính toán
  + Callback
    - Tối ưu callback bị khởi tạo lại khi component re-render
  + Custom hook
    - Tái sử dụng logic
* React Hook Form API

TODO

* **Reference**

<https://vi.reactjs.org/docs/hooks-intro.html>