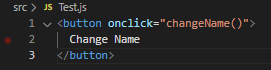
Xử lý sự kiện trong ReactJS. Trong một website việc tương tác giữa người dùng là điều không thể thiếu như click, nhập form,..chúng ta có thể thực hiện bắt các sự kiện này trong React một cách dễ dàng.

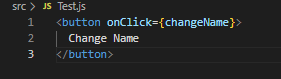
Xử lý các sự kiện trong React rất giống với xử lý các sự kiện trên các phần tử DOM. Có một số khác biệt về cú pháp:

* Các sự kiện React được đặt tên bằng camelCase, thay vì chữ thường. Ví dụ: *onclick -> onClick, onchange -> onChange*
* Với JSX, bạn truyền một hàm để bắt sự kiện, thay vì một chuỗi như HTML thông thường.

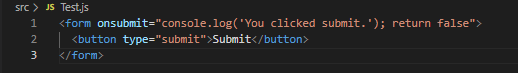
Ở đây là ví dụ với HTML:



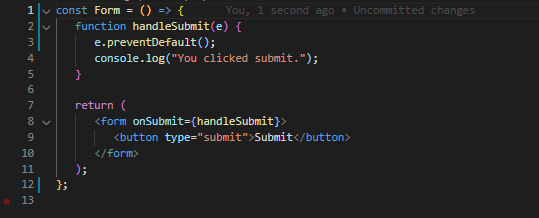
khi làm việc với JSX trong React thì có chút khác biệt:



Một điểm khác biệt nữa là bạn không thể sử dụng return false để chặn các hành động mặc định được (prevent default), trong React bạn cần phải sử dụng preventDefault(). Giả sử trong HTML mình muốn dừng hành động mặc định của một form:



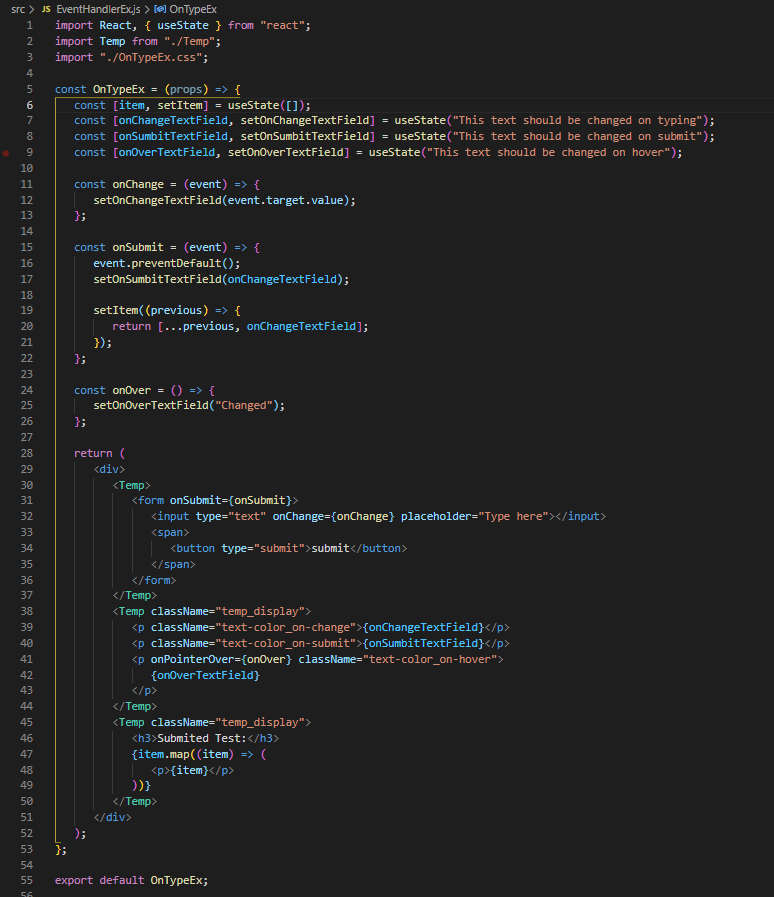
Còn trong Reactjs ta phải sử dụng event.preventDefault():



Ở đây, e là một object chứa tất cá event. React định nghĩa object này theo [W3C spec](https://freetuts.net/handling-events-trong-reactjs-2393.html), vì vậy chúng ta không cần phải quan tâm về khả năng tương thích giữa các trình duyệt với nhau.

**Xây dựng ví dụ**

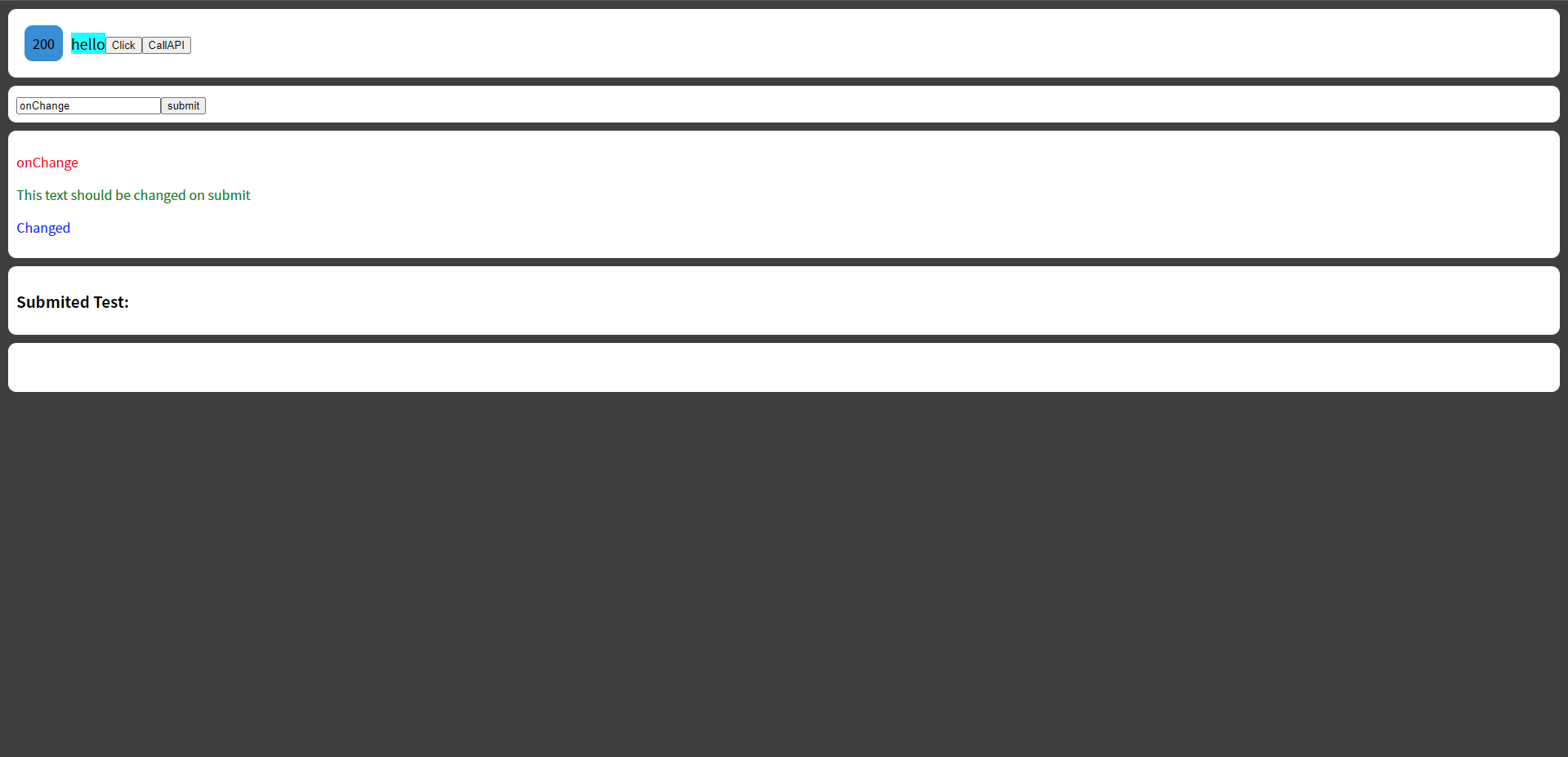
Phần này sẽ là phần đi xây dựng ví dụ về bắt sự kiện trong React và một vài cách gọi hàm trong React. Trong file src/EventHandlerEx.js chúng ta có ví dụ:

****

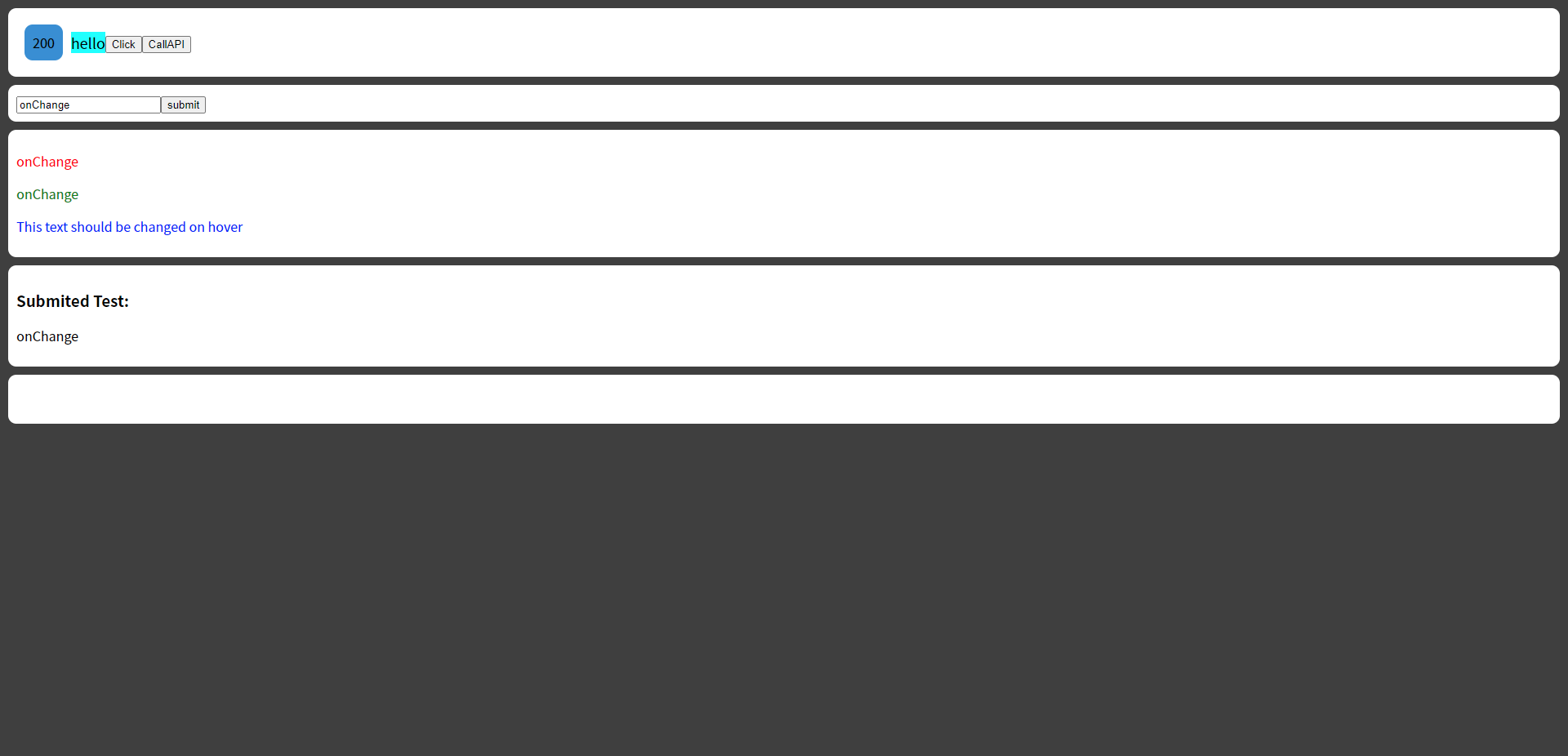
Trong ví dụ trên có những hàm như onSumbit(),onOver(),onChange() lưu ý các tên hàm này chỉ phục vụ cho ví dụ không nhất thiết phải đặt tên hàm như trên.

Đây là phần giao diện của ví dụ trên: 

Khi nhập vào input thì hàm onChange() sẽ được thực thi và đổi dòng chữ đỏ ban đầu thành input mỗi khi có sự thay đổi của input.



Khi chọn vào nút submit thì hàm onSubmit() được thực thi và lấy dữ liệu từ input và thay thế cho dòng chữ xanh lá ban đầu.



Khi di chuột qua dòng xanh dương thì hàm onOver() được thực thi thay đổi dòng chữ thành Changed.

