**BẢN THẢO TUẦN 4**

**Mục tiêu**: Thực hiện tiếp chương 1(phần thuộc về chương 2 đã được gộp trước đó nhằm đảm bảo tính cân đối giữa các chương)

**Dự kiến tuần tiếp theo**: Hoàn thiện các trang về mặt giao diện. Bắt đầu thực hiện chương 2.

**CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI**

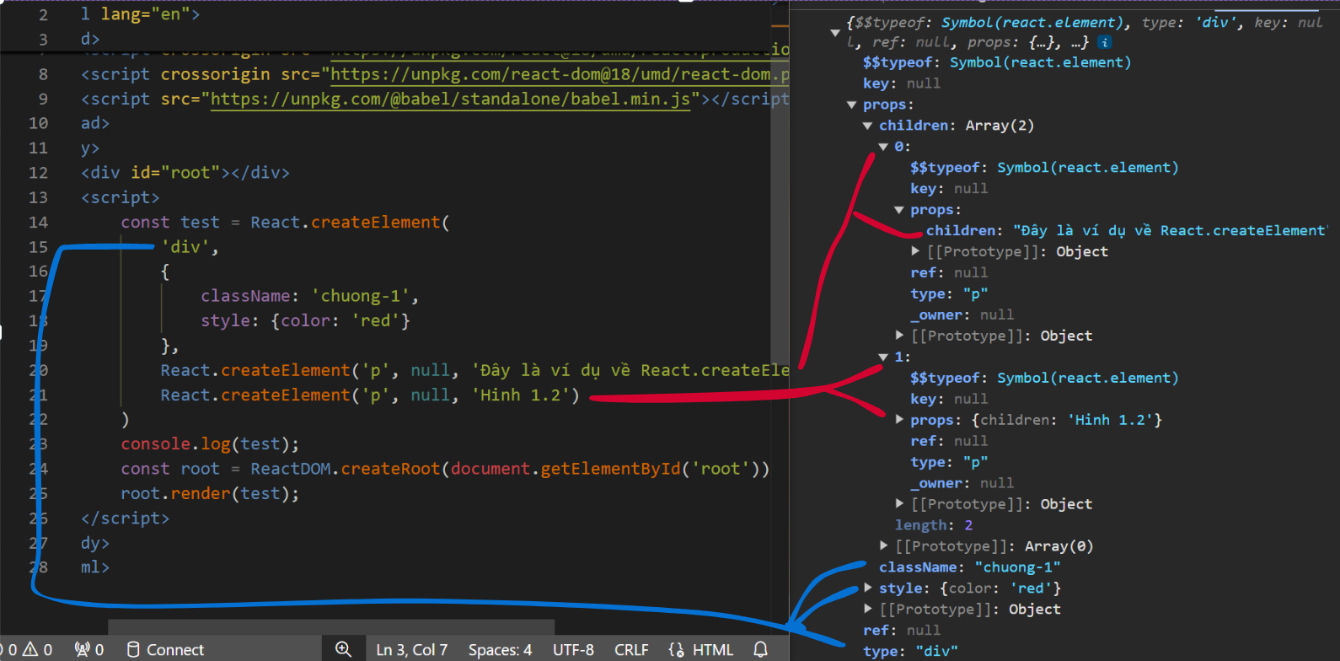
* 1. **Kết cấu chung của trang web**
  2. **Cơ sở lý thuyết**
     1. ***Lập trình giao diện người dùng***
     2. ***Lập trình phía server***
  3. **Tìm hiểu về ReactJS và các vấn đề liên quan**
     1. ***React và ReactDOM***

Tạo phần tử trong react bằng cách sử dụng phương thức: React.createElement(typeof, props, children1, children2,… chidrenN)

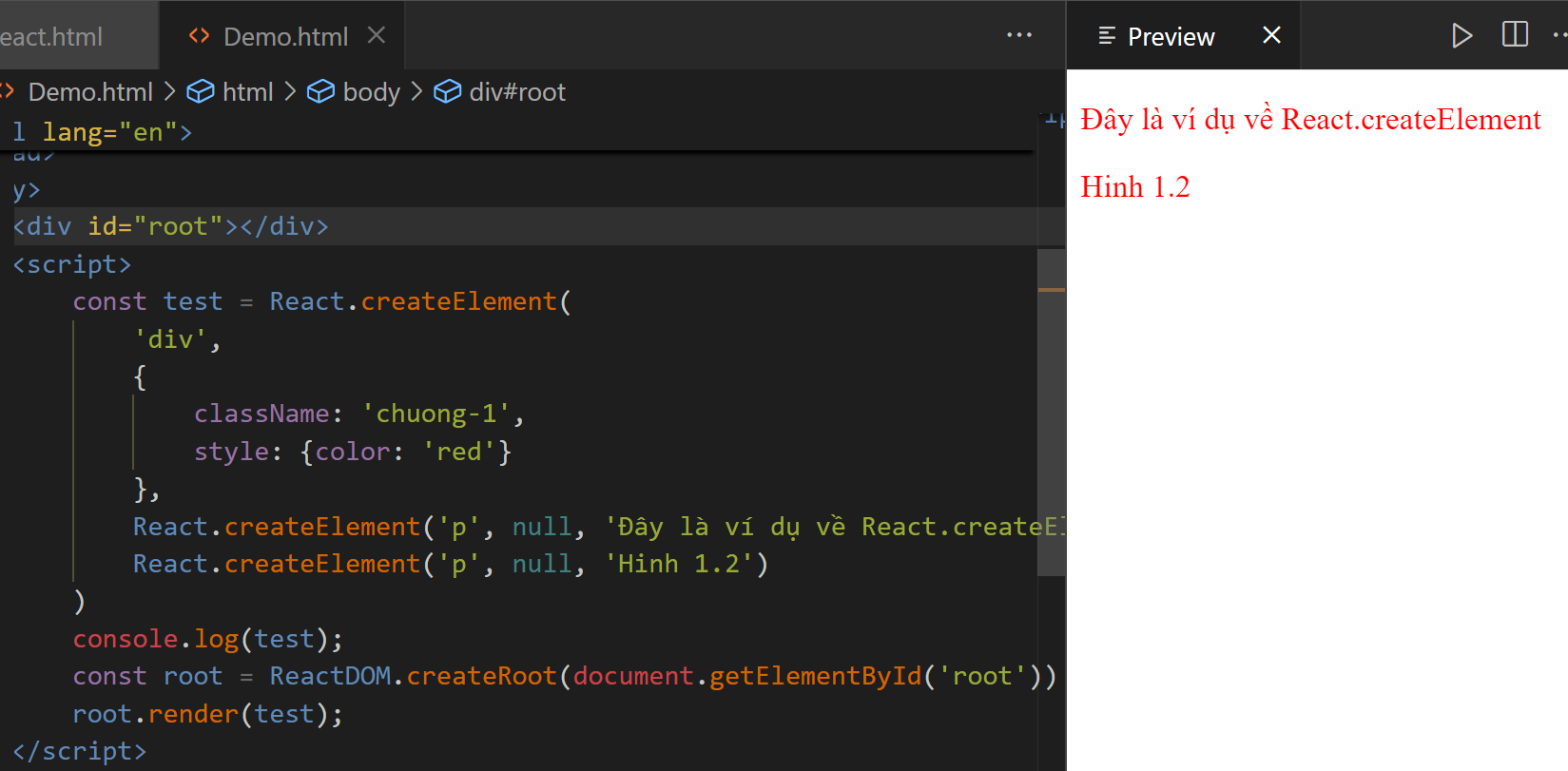
Trong đó:

* + typeof có thể là chuỗi (từ chuỗi có thể tạo ra các thẻ như h1, span,…), đối tượng, function, class,…
  + props: là đối tượng giúp truyền các đặc tính vào typeof, props có thể rỗng(null) hoặc chứa các key value như className, id, style,…)
  + children: các phần tử con của typeof

Hình 1.2 mô tả về một phần tử div được khởi tạo thông qua phương thức React.createElement với typeof là thẻ div có các đặc tính như className là chuong-1, style chứa đối tượng mang thuộc tính màu đỏ. Phần tử div này chứa hai phần tử con là hai thẻ p với phần props rỗng.

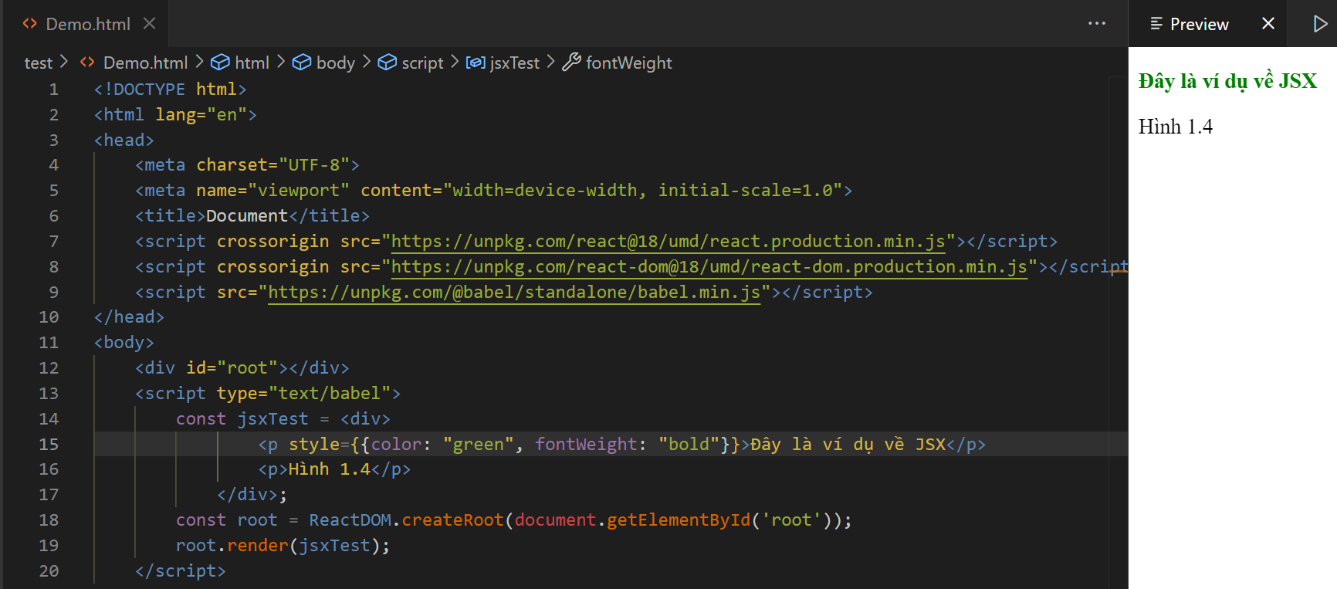
Để có thể hiển thị được các phần tử ra phía trình duyệt thì phải sử dụng ReactDOM. ReactDOM là trung gian giữa các phần tử React và Document Object Model (DOM). Cú pháp như sau:

ReactDOM.createRoot();

root.render();

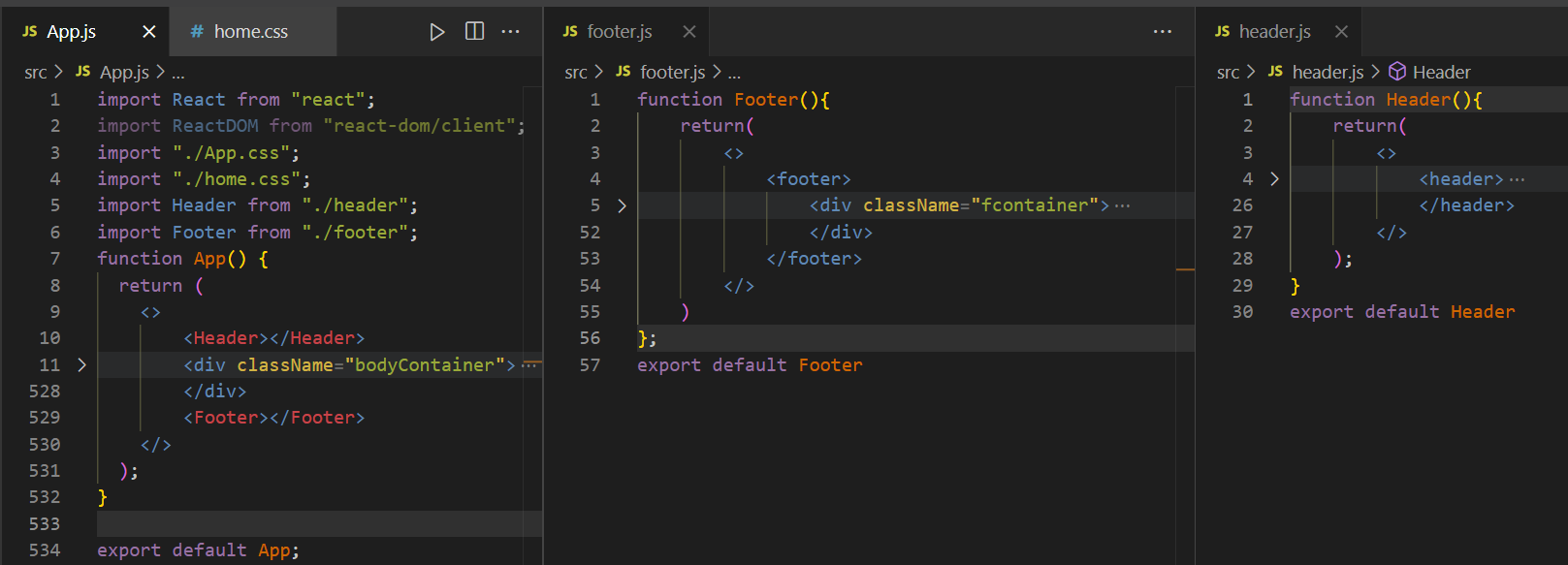
Hình 1.3 cho thấy phần tử div đã được xuất ra phía trình duyệt.

* + 1. ***Javascript XML (JSX)***

JSX hỗ trợ lập trình viên có thể đưa những đoạn mã HTML vào trong javascript để sử dụng. Trong thực tế với ReactJS, để tạo một phần tử phải sử dụng phương thức React.createElement() và khi cần khởi tạo nhiều phần tử trong trang web thì cần rất nhiều đoạn React.createElement(), do đó nhằm tạo sự thuận tiện thì lập trình viên sẽ sử dụng cú pháp của HTML và JSX sẽ đưa về dạng javascript. Để đưa các đoạn HTML vào trong javascript và trình duyệt có thể hiểu có thể sử dụng các transpilation như Babel để chuyển đổi HTML thành javascript.

Hình 1.4 minh hoạ về việc sử dụng trực tiếp thẻ div và thẻ p mà không cần khởi tạo phần tử bằng phương thức React.createElement(). Ở thẻ script sử dụng type=”text/babel” nhằm khả dụng hoá trình biên dịch babel được thêm ở dòng 9.

* + 1. ***Component***

Các phần tử của React có nhiều loại từ chuỗi, function đến class. Khi triển khai một website sẽ có nhiều thành phần nhỏ cấu thành, tiêu biểu có thể nhắc đến header và footer, hai thành phần này có mặt hầu hết ở mỗi trang nên việc sửa đổi sẽ diễn ra tại tất cả các header, footer ở tất cả các trang, nhằm tối ưu trong việc lập trình thì sẽ chia hai phần này thành các component. Ở các trang cần sử dụng (trang chủ, giỏ hàng,..) có thể import 2 component và khi export ra sẽ chứa đầy đủ nội dung trang kèm theo hai component là header và footer.

Hình 1.5 mô tả cách hoạt động của component, với đặc tính dễ nhìn và khả năng tái sử dụng cao, component trở thành một ưu điểm cực kỳ tốt của ReactJS.

Có thể băm nhỏ trang web thành những phần tử lớn như header và footer hoặc nhỏ như một nút, do đó khả năng bảo trì và phát triển ứng dụng cực kỳ cao. Thông thường một component sẽ được định nghĩa là function hoặc là class.

Cách dùng component:

* Tạo một file chứ component với đuôi js
* Tạo function với tên hàm viết hoa chữ cái đầu và xuất
* Nhập file chứa component vào file chính, lúc này tên function chính là tên component