**CSS (3) - Định chuẩn CSS, box model và đơn vị đo**

**Một số cách định chuẩn CSS**

Chúng ta có thể sử dụng một số cách sau để định chuẩn CSS:

– Cách đầu tiên, đơn giản nhất là đặt lại giá trị của thuộc tính căn lề (margin), và dãn biên (padding) bằng 0 cho mọi phần tử (kí hiệu \*). Cách làm này không được khuyến khích vì nó ảnh hưởng đến tốc độ xử lý, trong khi còn nhiều thiết lập mặc định khác lại không được thực hiện.

<style>

        \* {

            margin: 0;

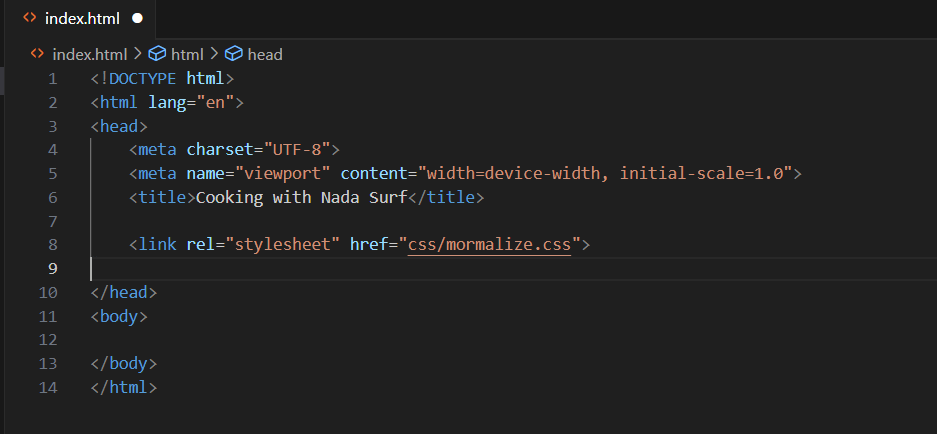
            padding: 0;

        }

…

    </style>

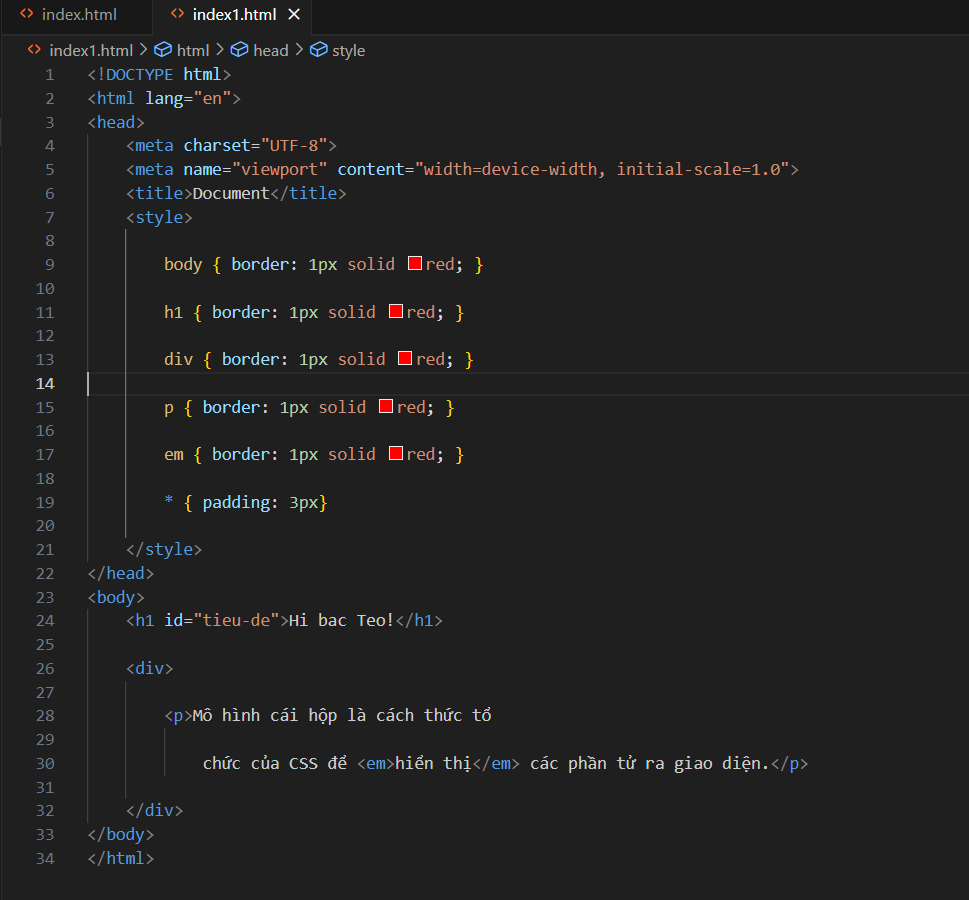
– Trong tập tin HTML, khai báo để luôn thực thi đoạn mã trong normalize.css đầu tiên.



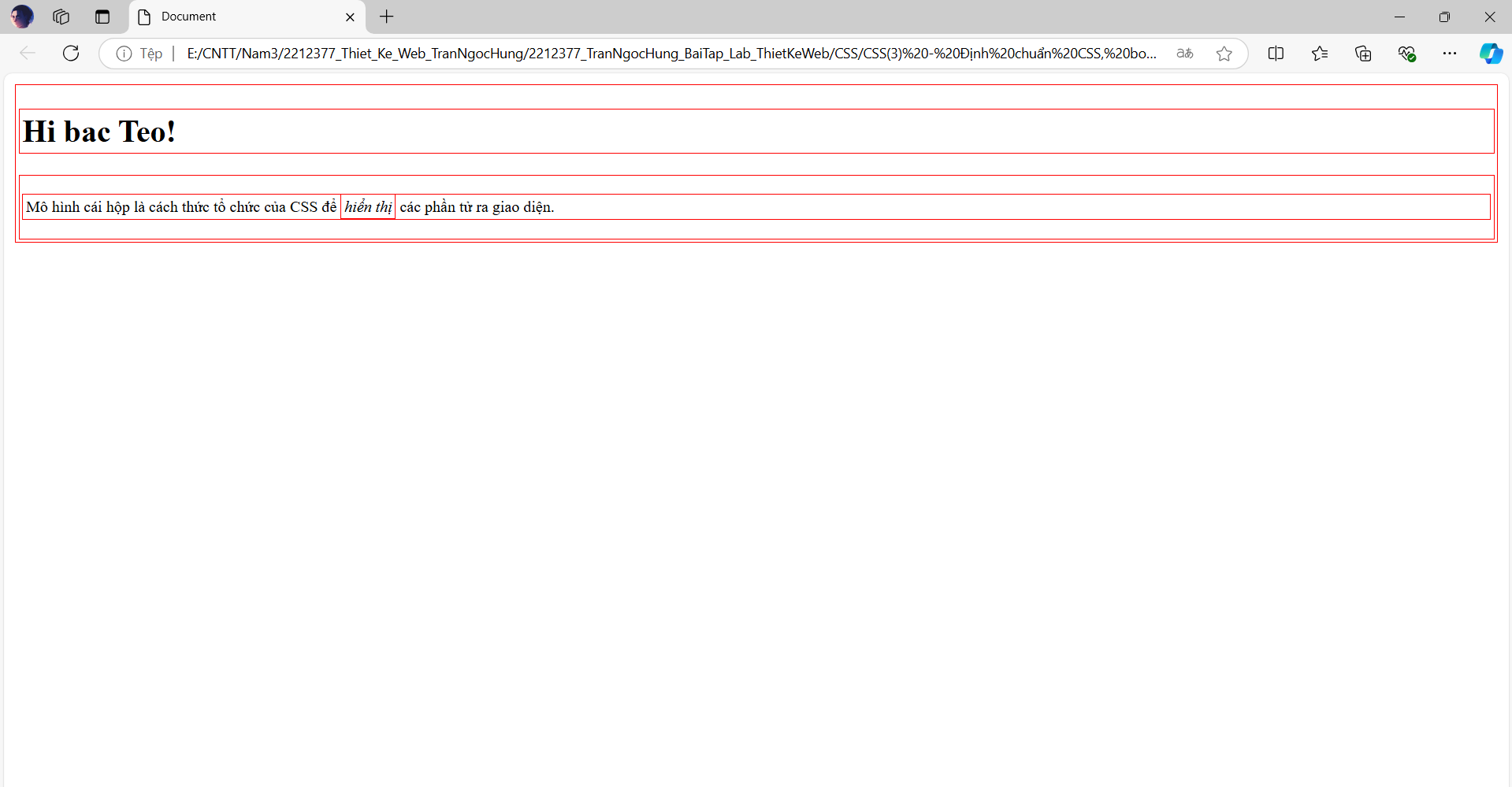
**Box model**

Trong việc định dạng hiển thị, chúng ta sẽ phải đặt các phần tử HTML ở một ví trí cụ thể trên giao diện, với chiều rộng, chiều dài xác định, hoặc đổ màu nền. Để tiện lợi cho việc này, CSS đưa ra khái niệm mô hình cái hộp (box model).

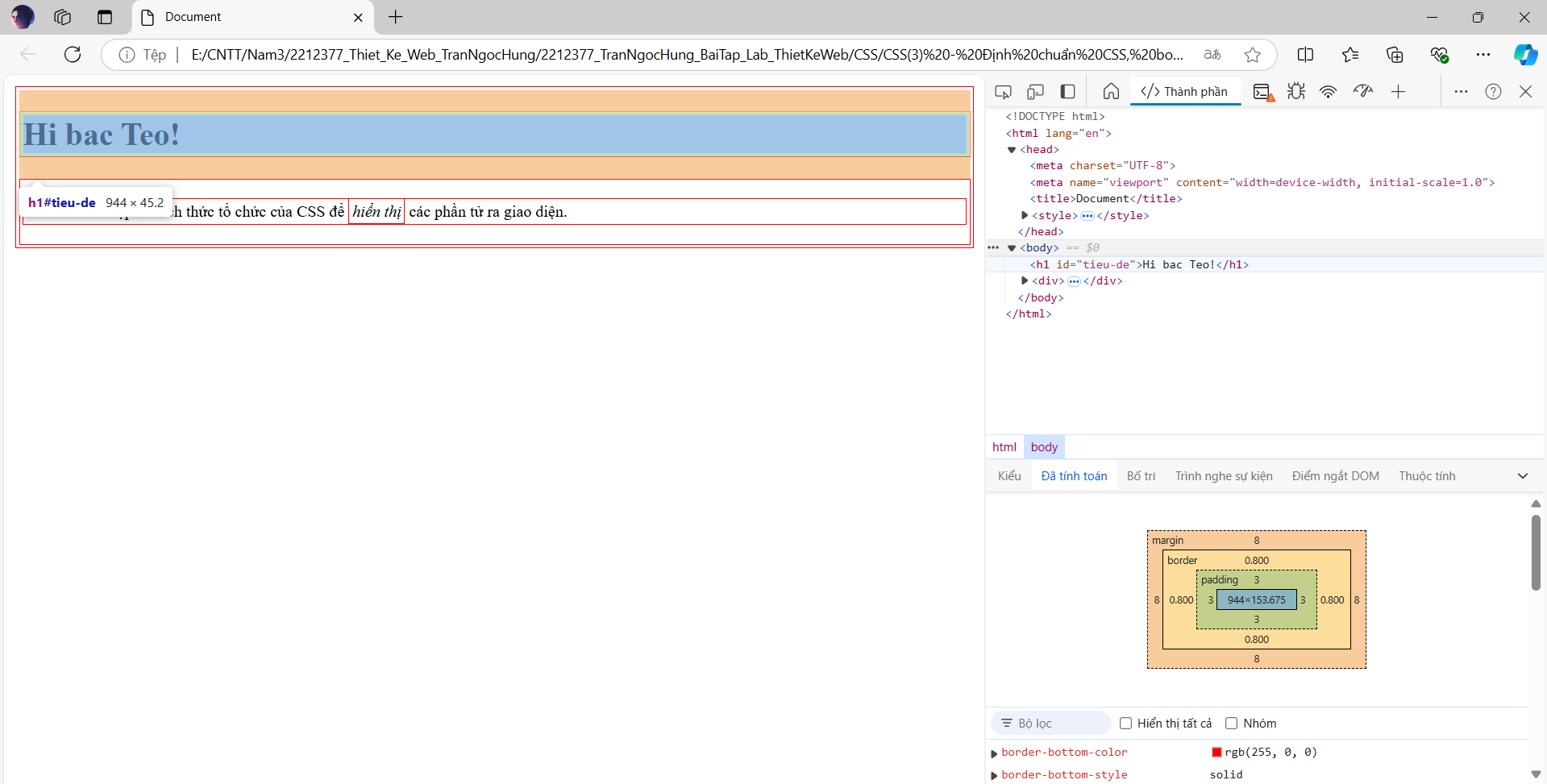
Ví dụ, đoạn mã sau minh họa hình ảnh của box model, gồm 5 cái hộp, với cấu trúc lồng nhau.



Kết quả:



Chúng ta cũng có thể sử dụng Developer tools để quan sát box model một cách trực quan. Như hình sau:

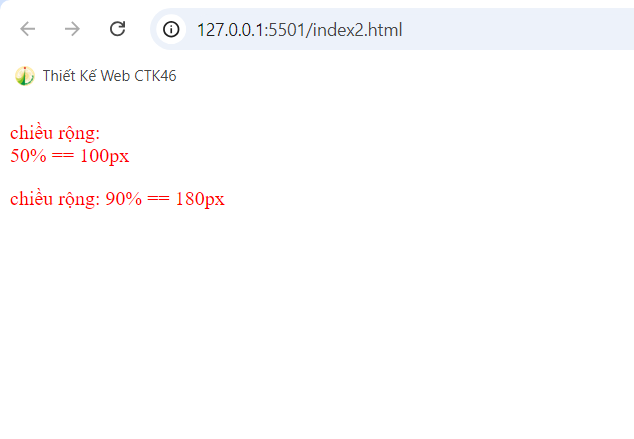


Trong thiết kế bố cục (layout), đơn vị đo này giúp cho trang web luôn giữ được tỉ lệ cố định, sự cân đối giữa các thành phần, khi người dùng thay đổi kích thước cửa sổ trình duyệt.

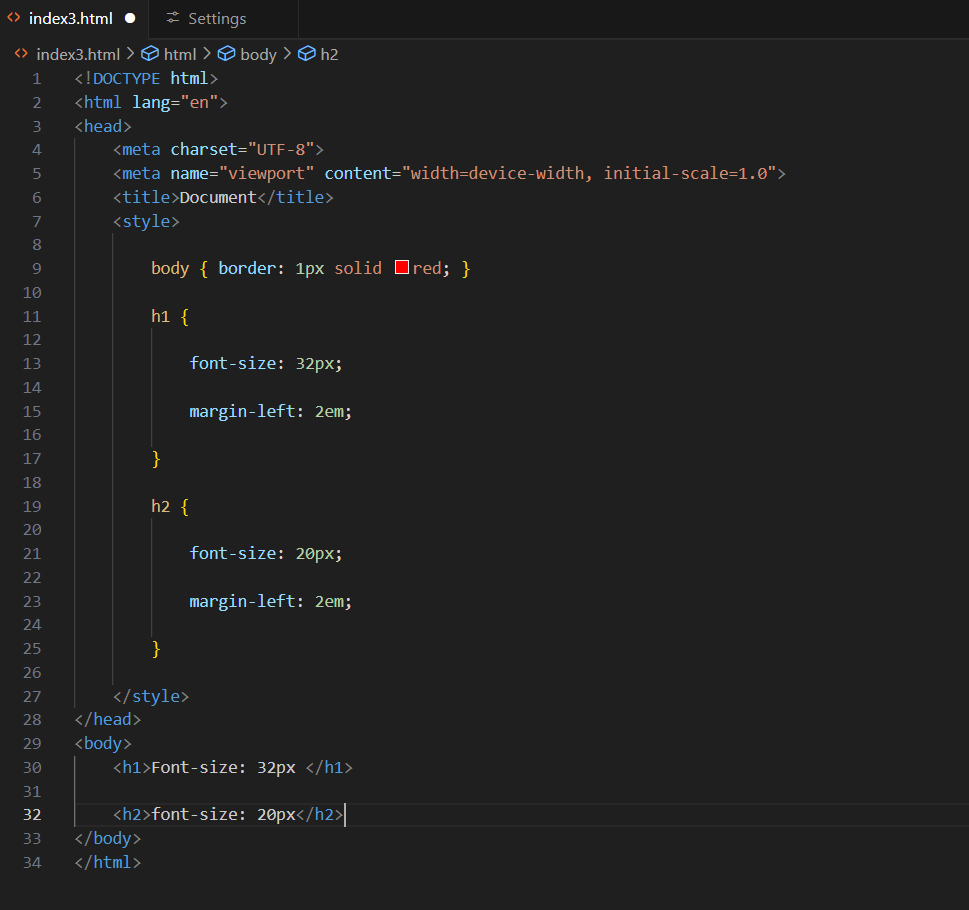
Ví dụ:



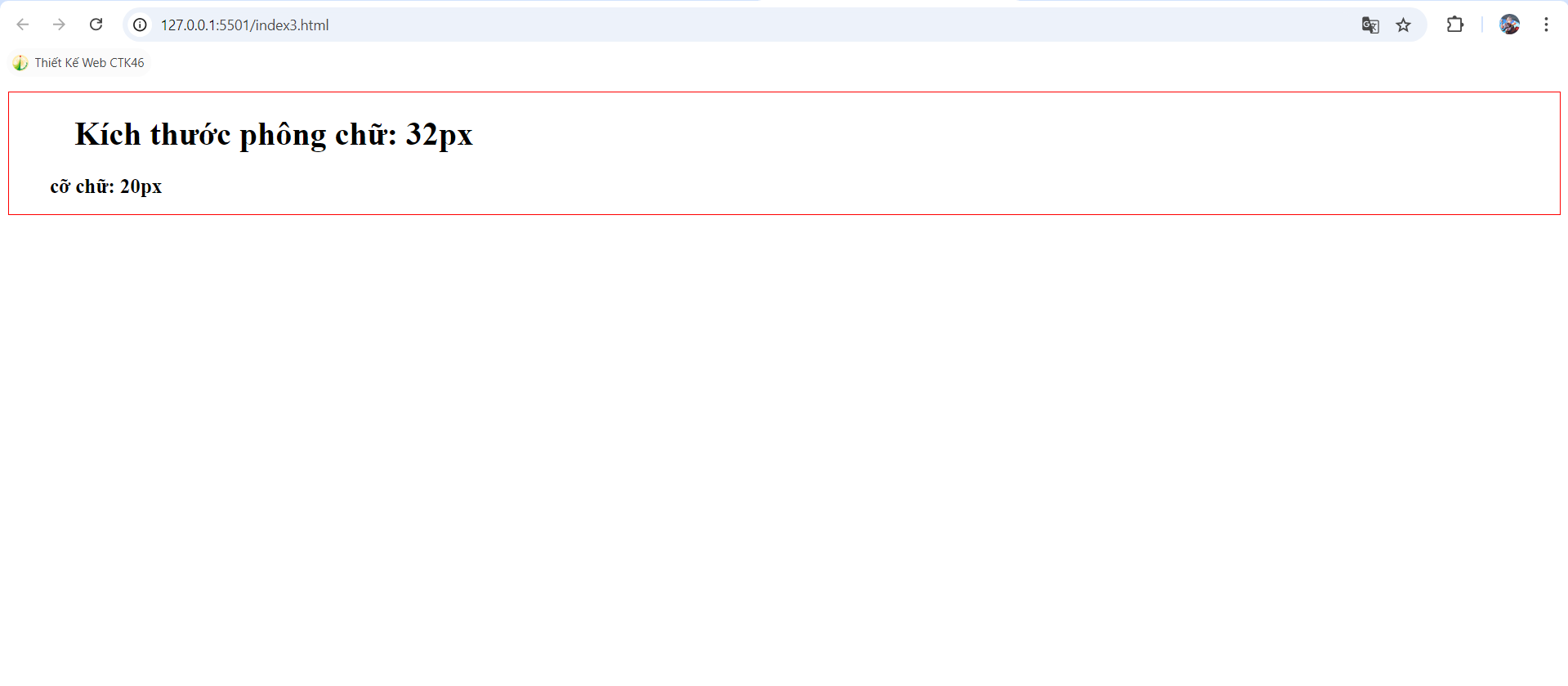
Kết quả:



**EM**

****

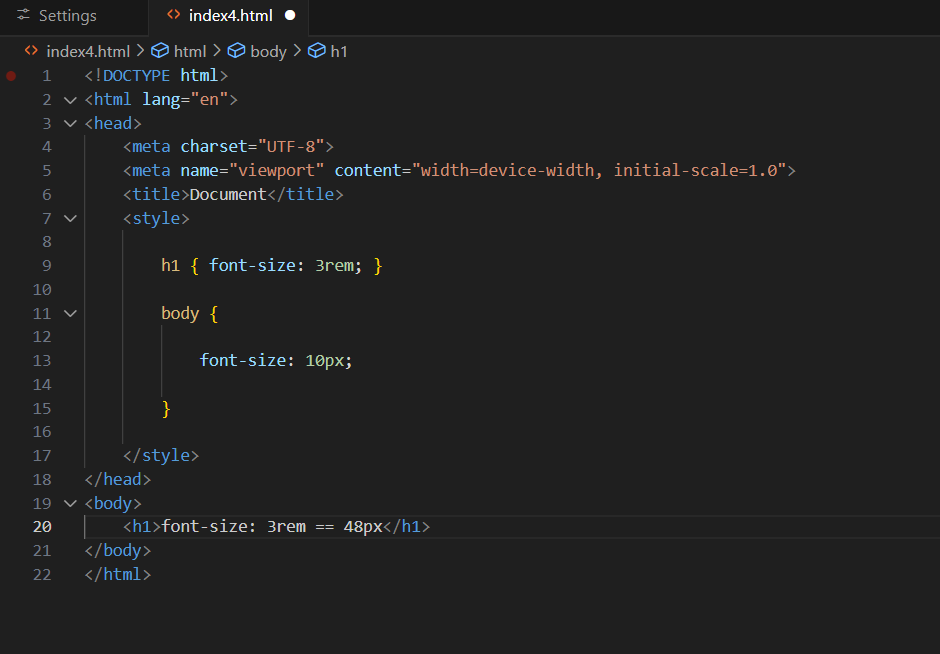
**Kết quả:**

****

**Rem**

Sử dụng đơn vị đo là em có khá nhiều ưu điểm, tuy nhiên trong một số trường hợp, do việc tính giá trị em phải dựa vào phần tử cha, điều này gây ra bất tiện cho lập trình viên khi tài liệu HTML gồm nhiều cấp. Để khắc phục nhược điểm này, có thể sử dụng đơn vị đo khác là rem.

Rem là viết tắt của “root em”. Một rem sẽ tương đương với kích thước phông chữ của phần tử html. Như vậy, nếu thiết lập kích thước cho một phần tử bất kì là rem, nghĩa là kích thước của phần tử đó sẽ được tính theo kích thước của phần tử html chứ không phải phần tử cha của nó.



Kết quả:

