**CSS (11) - Bộ chọn theo trạng thái & thuộc tính**

**1.1.5 Bài tập và thực hành**

**Bài tập 1. Viết lại các đoạn mã trong phần lý thuyết.**

**1.1 Bộ chọn theo trạng thái & thuộc tính**

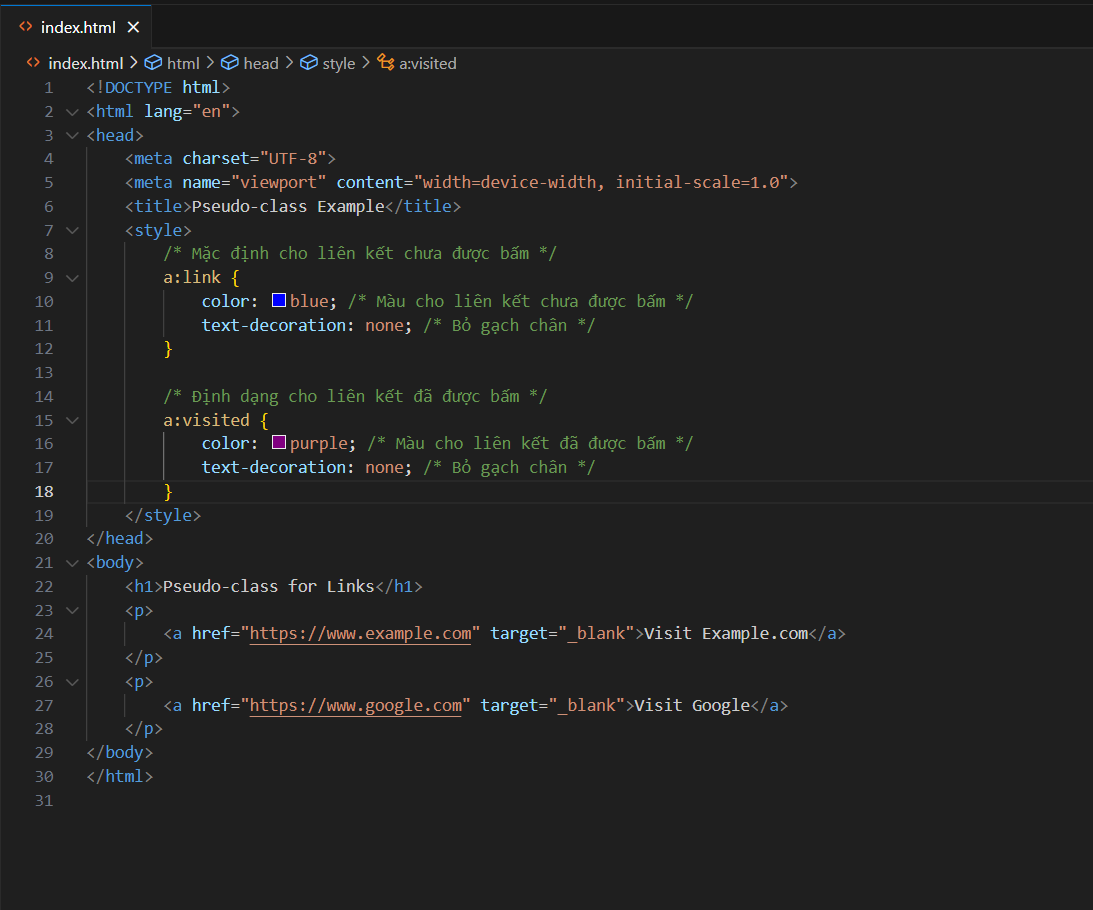
Bộ chọn pseudo-class cho liên kết

Bộ chọn pseudo-class cho liên kết được sử dụng để định dạng các liên kết dựa trên trạng thái của nó. Cụ thể,

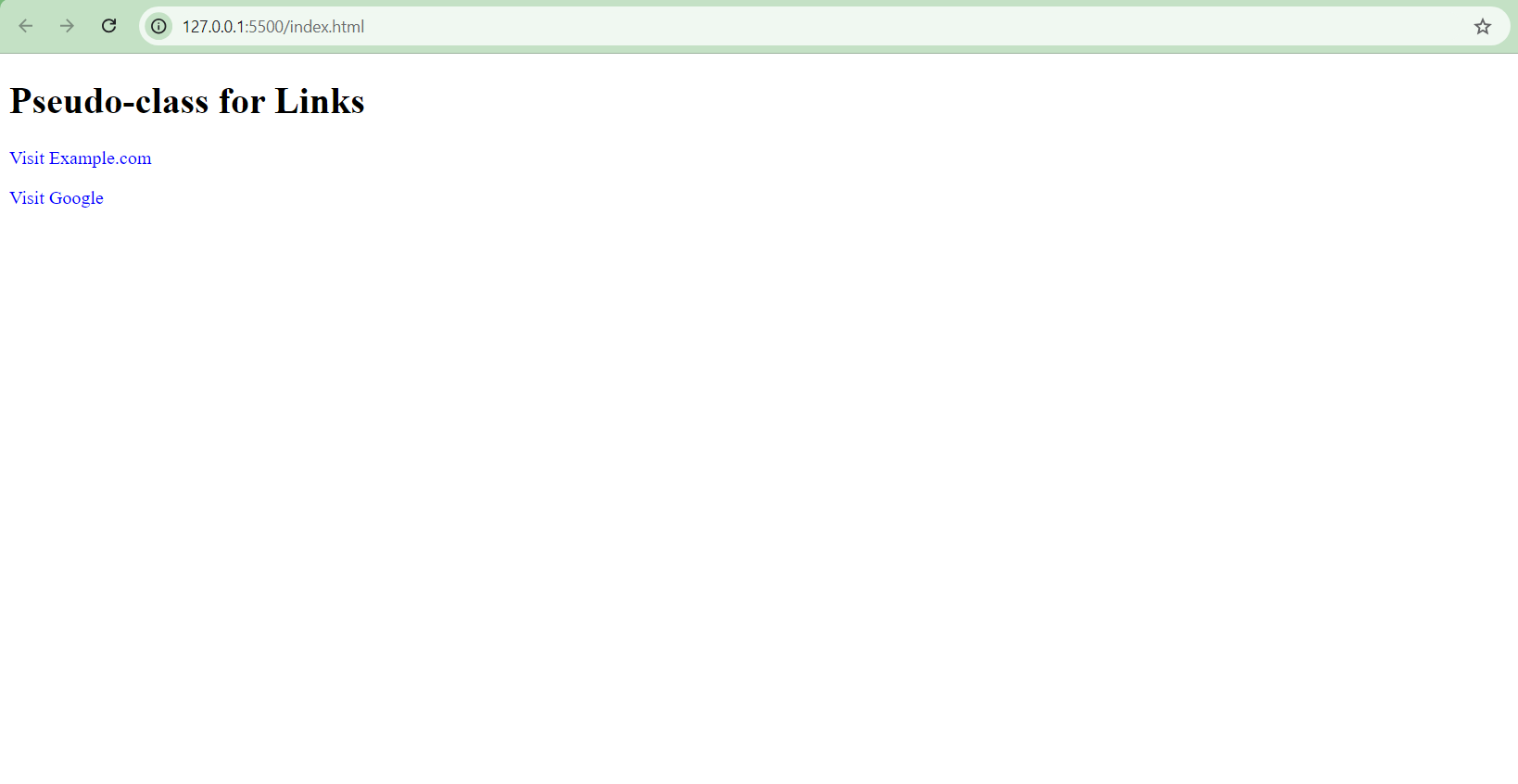
– :link dùng để định dạng cho liên kết chưa được người dùng bấm vào

– :visited dùng để định dạng cho liên kết đã được người dùng bấm vào

Ví dụ: mặc định, liên kết người dùng chưa bấm vào sẽ có màu xanh, bấm vào rồi sẽ có màu tím. Tuy nhiên, chúng ta có thể sử dụng CSS để thay đổi màu mặc định của liên kết thành các màu khác,



Kết quả:



Bộ chọn pseudo-class dựa trên thao tác người dùng

– :hover áp dụng khi người dùng rê con trỏ chuột trên một phần tử, mặc dù có thể áp dụng trên mọi phần tử HTML, tuy nhiên, bộ chọn này hay được áp dụng trên các liên kết, nhằm gây chú ý cho người sử dụng. Ví dụ, liên kết sẽ đổi màu nền khi người dùng rê chuột lên,

a:hover {

color: maroon;

background-color: #ffd9d9;

}

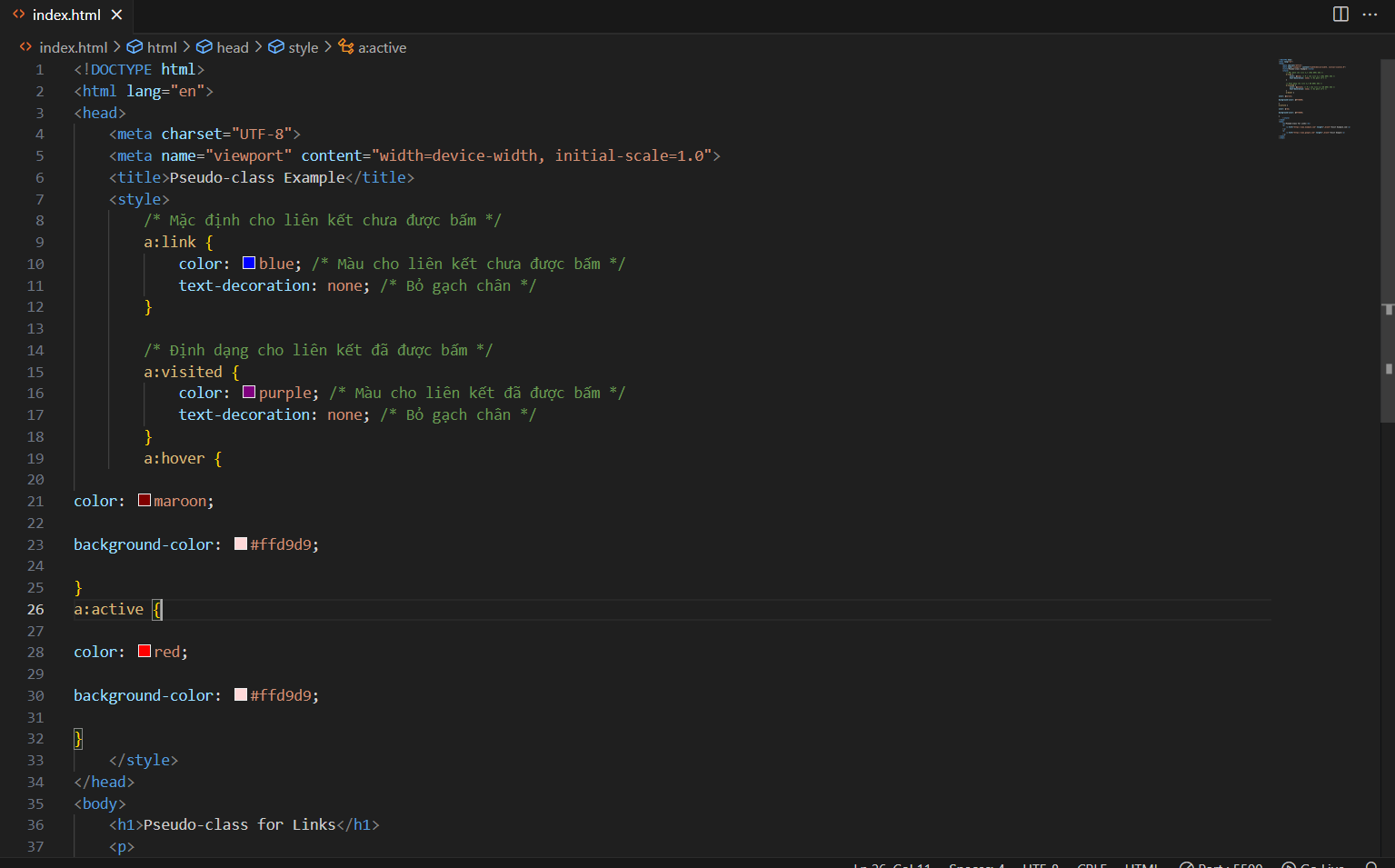
– :active áp dụng khi một phần tử (nút hoặc liên kết) đang được bấm chuột. Ví dụ, liên kết sẽ đổi màu nền khi người dùng đang bấm chuột lên,

a:active {

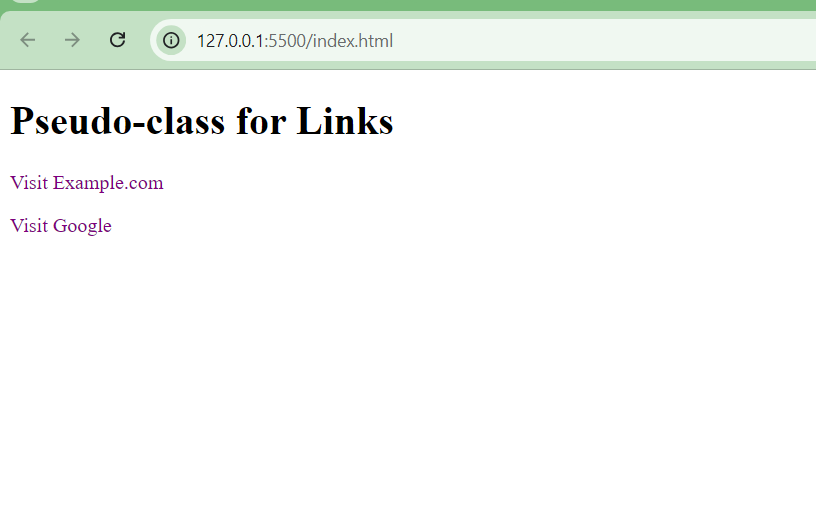
color: red;

background-color: # ffd9d9;

}



Kết quả:



Code <!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Link States Example</title>

    <style>

        /\* Bỏ dấu gạch chân cho tất cả các liên kết \*/

        a {

            text-decoration: none;

        }

        /\* Định dạng cho liên kết chưa được bấm \*/

        a:link {

            color: maroon; /\* Màu cho liên kết chưa được bấm \*/

        }

        /\* Định dạng cho liên kết đã được bấm \*/

        a:visited {

            color: gray; /\* Màu cho liên kết đã được bấm \*/

        }

        /\* Định dạng cho liên kết khi được chọn (focus) \*/

        a:focus {

            color: maroon; /\* Màu khi chọn \*/

            background-color: #ffd9d9; /\* Màu nền khi chọn \*/

        }

        /\* Định dạng cho liên kết khi di chuột qua (hover) \*/

        a:hover {

            color: maroon; /\* Màu khi di chuột qua \*/

            background-color: #ffd9d9; /\* Màu nền khi di chuột qua \*/

        }

        /\* Định dạng cho liên kết khi nhấn vào (active) \*/

        a:active {

            color: red; /\* Màu khi nhấn vào \*/

            background-color: #ffd9d9; /\* Màu nền khi nhấn vào \*/

        }

    </style>

</head>

<body>

    <h1>Link States Example</h1>

    <p>

        <a href="https://www.example.com" target="\_blank">Visit Example.com</a>

    </p>

    <p>

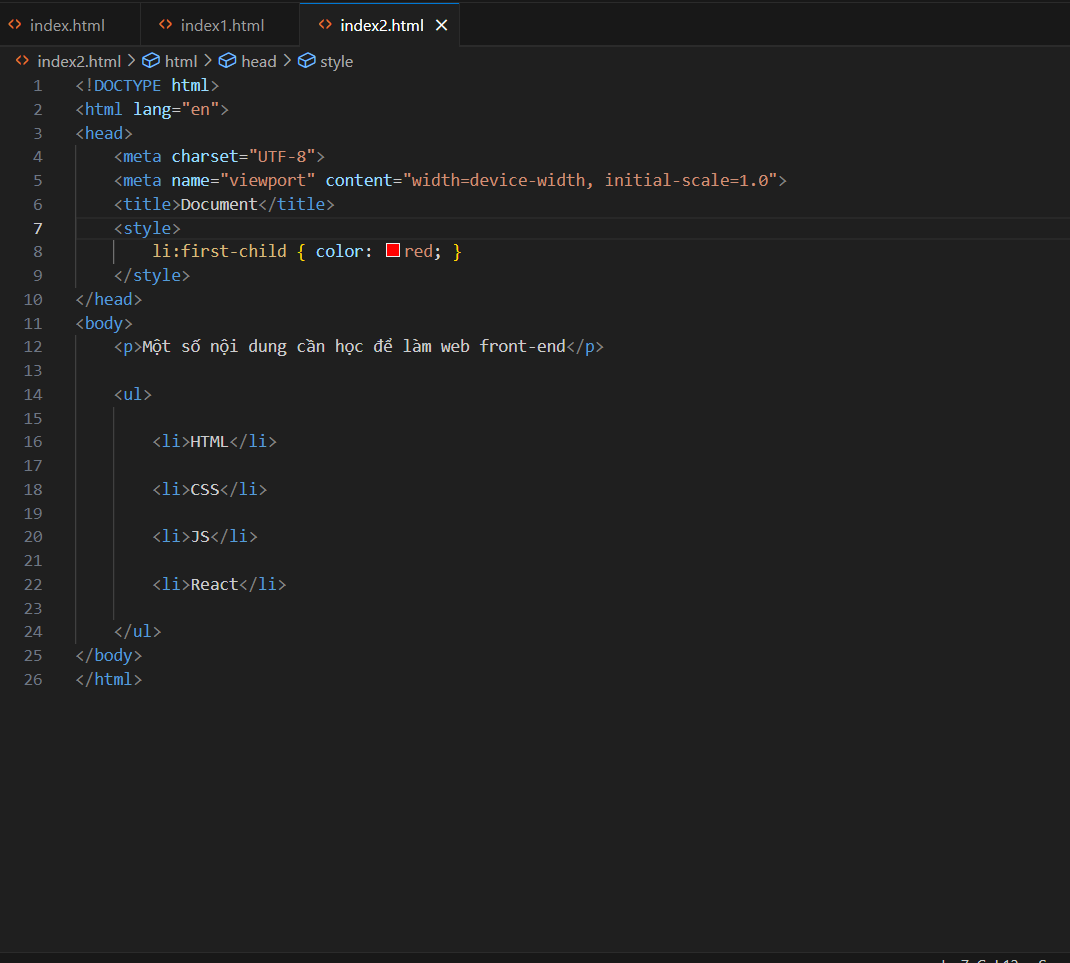
        <a href="https://www.google.com" target="\_blank">Visit Google</a>

    </p>

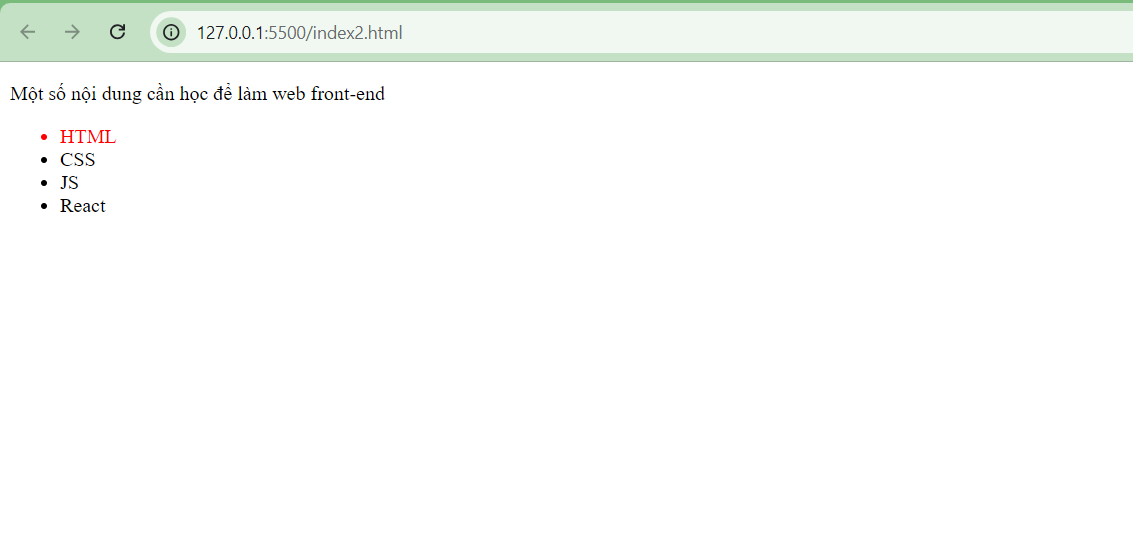
</body>

</html>

Kết quả:

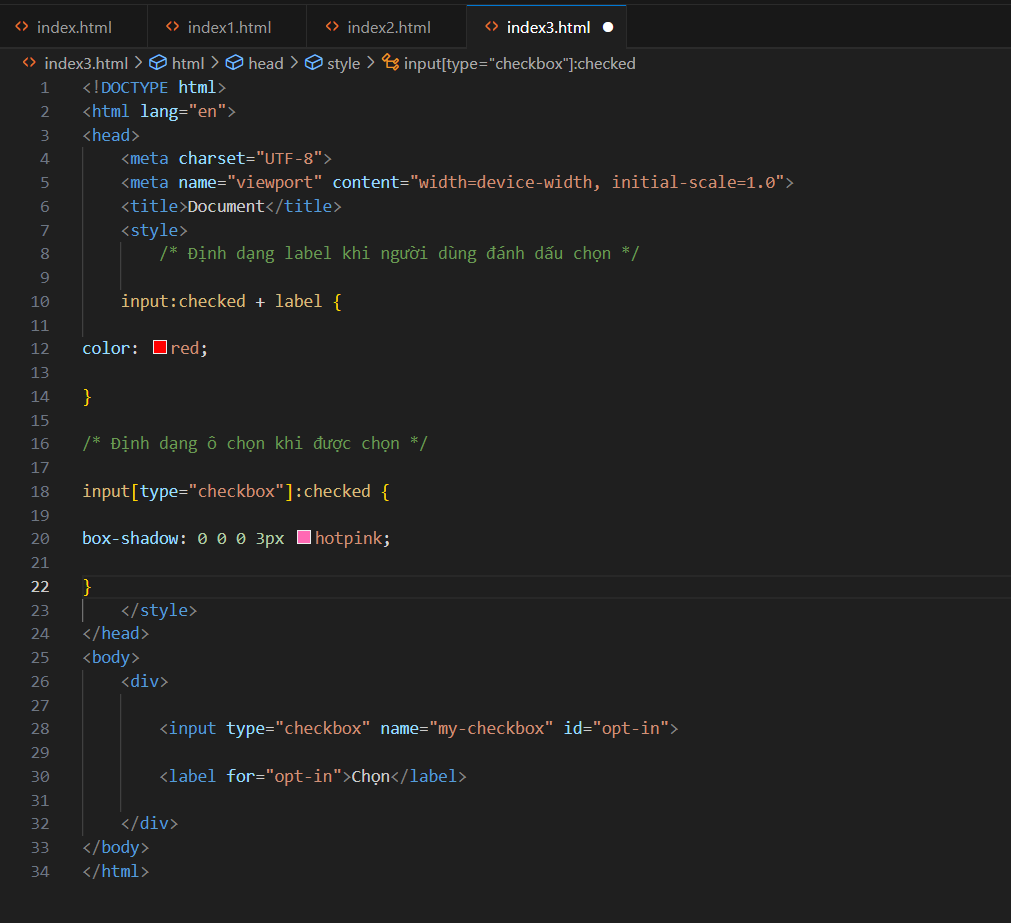


Kết quả:

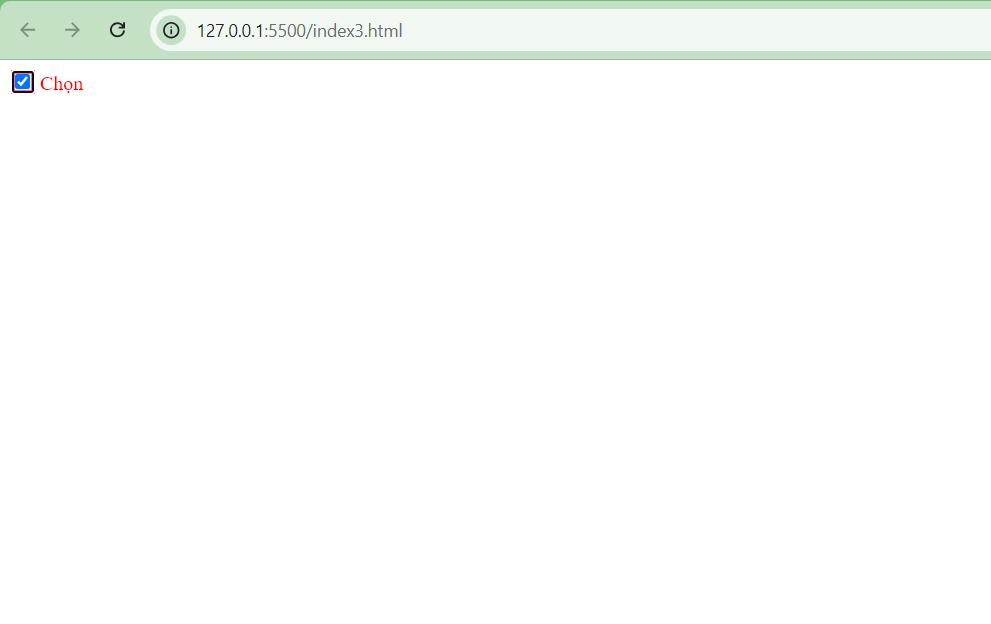


**Bộ chọn pseudo-class cho các input của form**

Bộ chọn pseudo-class cho các input của form sẽ chọn ra các phần tử dựa vào thuộc tính HTML và trạng thái của các input trước và sau khi tương tác. Ví dụ,



Kết quả:



**1.1.2 Bộ chọn phần tử giả (pseudo-element)**

Trong CSS, không chỉ có bộ-chọn-lớp-giả (pseudo-class), mà nó còn có bộ-chọn-phần-tử-giả (pseudo-element). Bộ chọn pseudo-element không chọn các phần tử trong tài liệu HTML, cũng không chọn phần tử dựa vào trạng thái của nó, mà sẽ chọn các đối tượng “giả”, dựa vào vị trí hiển thị của các đối tượng trên cửa sổ trình duyệt. Có nhiều bộ chọn pseudo-element, phần này chúng ta sẽ tìm hiểu một số bộ chọn gồm: ::first-line, ::first-letter, ::before và ::after.

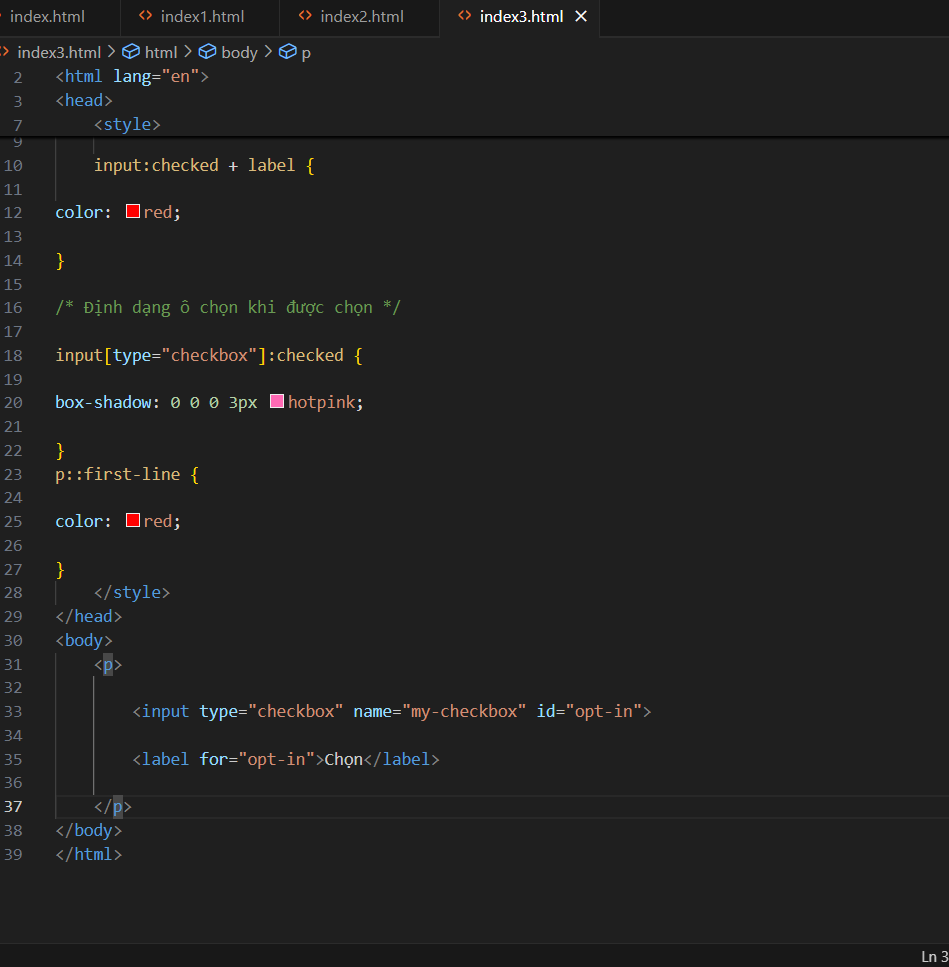
Bộ chọn ::first-line

Bộ chọn ::first-line được sử dụng để chọn hàng đầu tiên của một phần tử văn bản (khi văn bản hiển thị trên trình duyệt). Tuy nhiên, với bộ chọn này, chỉ có thể áp dụng được các thuộc tính sau,

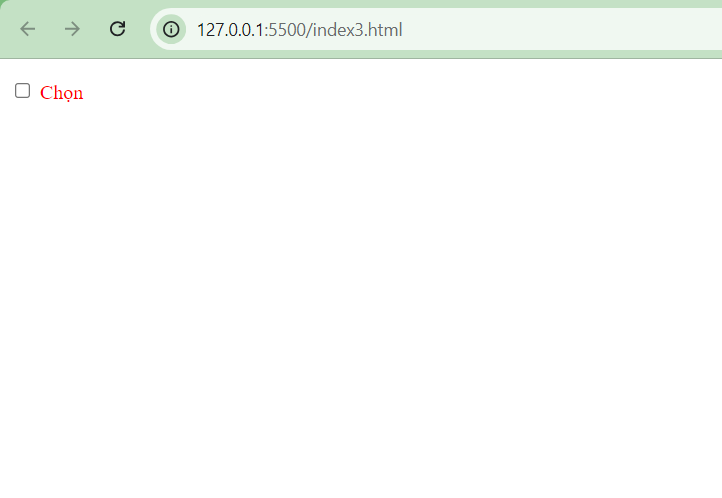
color, font, background

word-spacing, letter-spacing, text-decoration

vertical-align, text-transform, line-height



Kết quả:



**Bộ chọn ::first-letter**

Bộ chọn ::first-letter được sử dụng để chọn kí tự đầu tiên của một phần tử văn bản (khi văn bản hiển thị trên trình duyệt). Tuy nhiên, với bộ chọn này, chỉ có thể áp dụng được các thuộc tính sau,

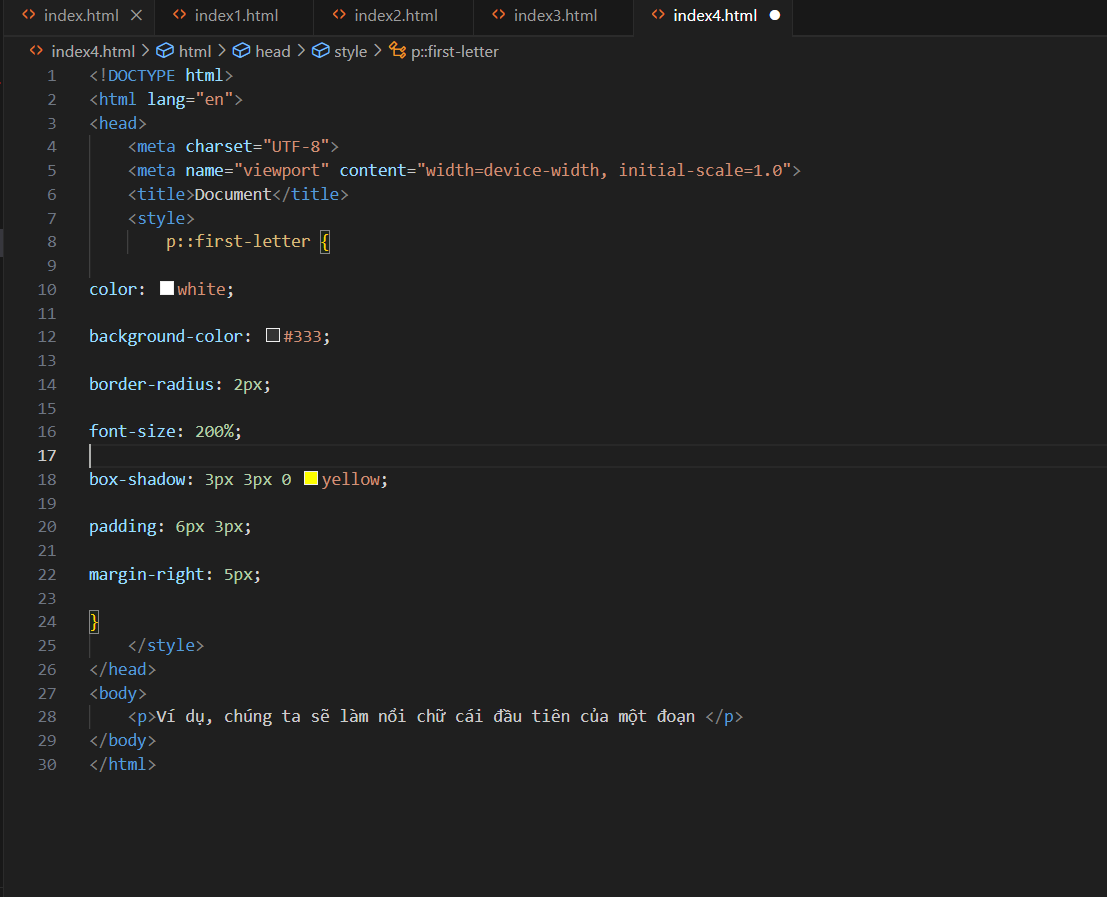
color, font, text-decoration

text-transform, vertical-align, padding

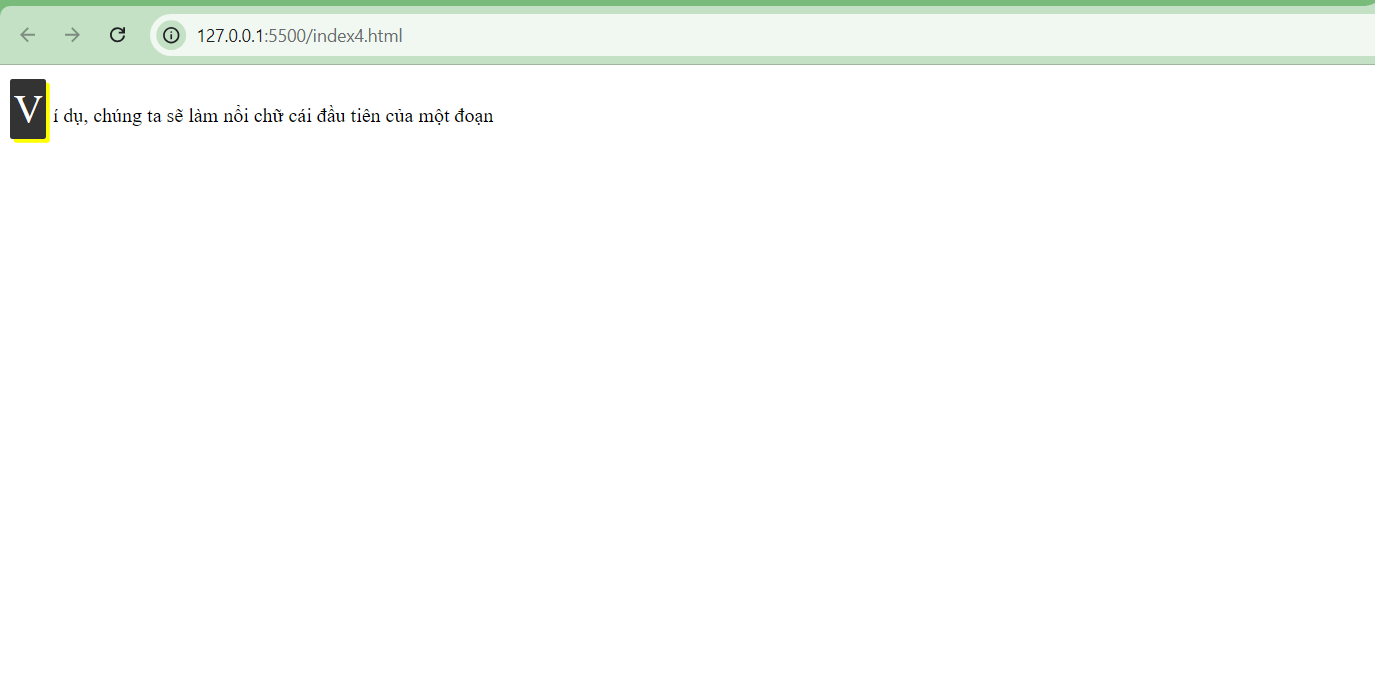
background, margin, line-height

border, float, letter-spacing, word-spacing

Ví dụ, chúng ta sẽ làm nổi chữ cái đầu tiên của một đoạn (tạo drop cap):



Kết quả:

**Bộ chọn ::before**

Bộ chọn ::before được sử dụng để chèn thêm nội dung vào trước một phần tử mà không cần chỉnh sửa mã nguồn HTML (cái này được gọi là thêm nội dung bằng CSS).

Ví dụ, đoạn mã CSS sau sẽ thêm cụm từ “Đọc thêm: ” vào trước phần tử p,

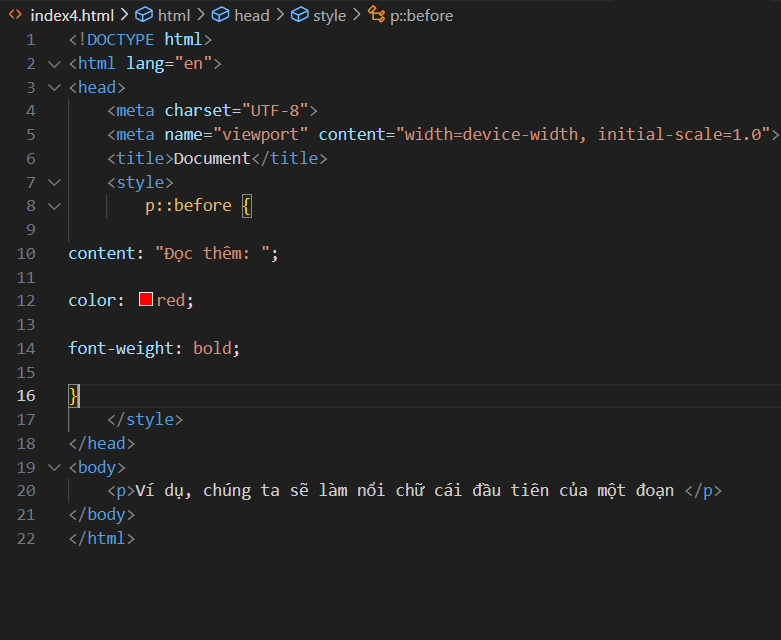
p::before {

content: "Đọc thêm: ";

color: red;

font-weight: bold;

}



Kết quả:



**Bộ chọn ::after**

Bộ chọn ::after được sử dụng để chèn thêm nội dung vào sau một phần tử mà không cần chỉnh sửa mã nguồn HTML (cái này được gọi là thêm nội dung bằng CSS).

Ví dụ, đoạn mã CSS sau sẽ thêm cụm từ “(Kết thúc)” vào sau phần tử p,

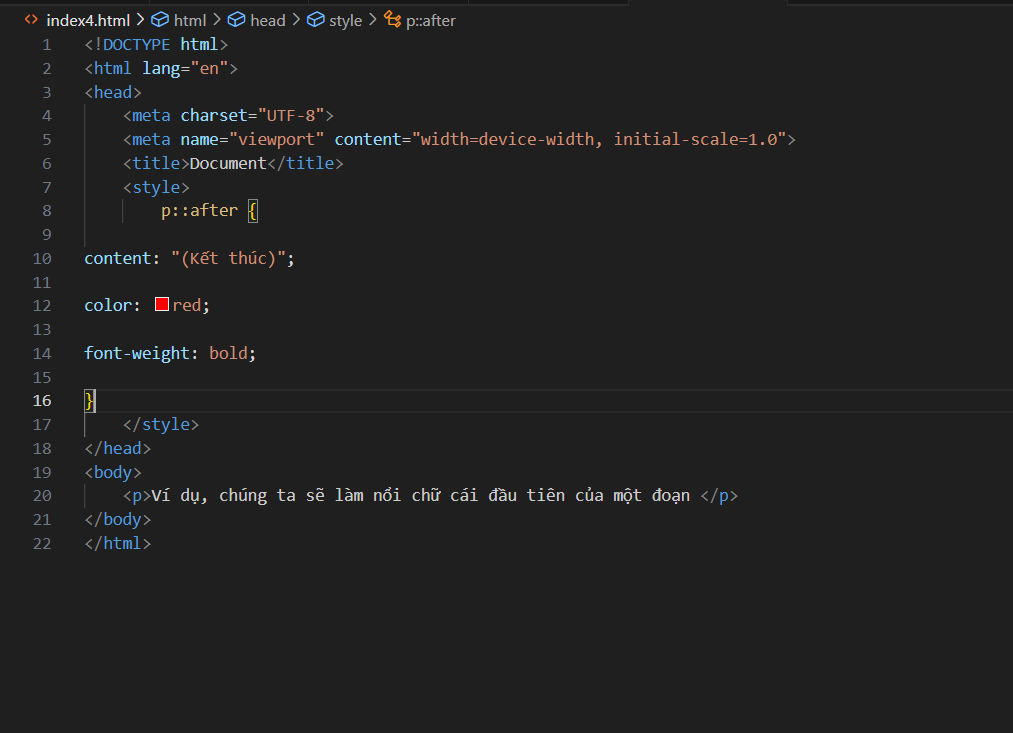
p::after {

content: "(Kết thúc)";

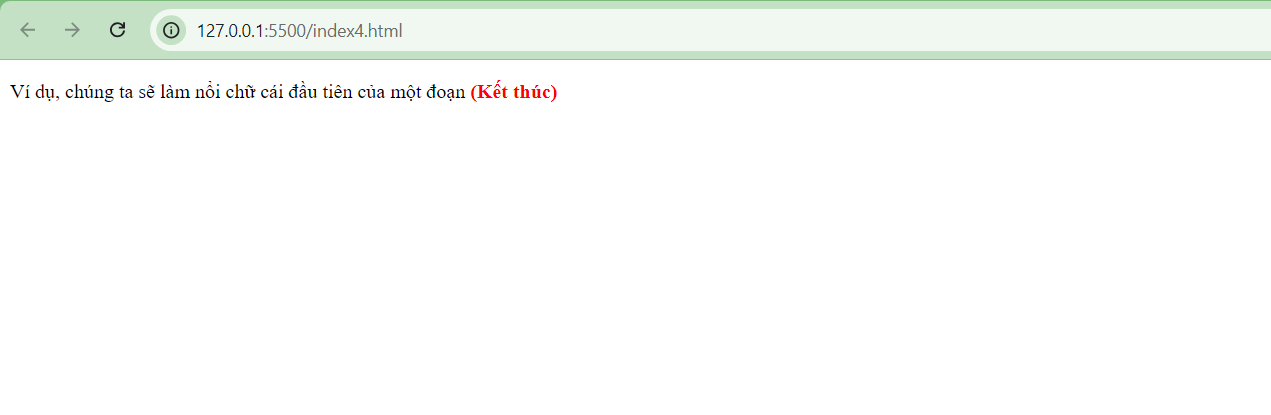
color: red;

font-weight: bold;

}



Kết quả:



**1.1.3 Bộ chọn dựa vào thuộc tính**

Chúng ta có thể tạo bộ chọn dựa vào tên và giá trị thuộc tính của phần tử. Cách làm này giúp tạo ra các bộ chọn rất linh hoạt mà không cần phải thêm thuộc tính class hoặc id vào các phần tử.

element[attribute]

Đây là bộ chọn đơn giản, nó sẽ chọn ra các phần tử với tên thuộc tính cụ thể, không quan tâm tới giá trị của thuộc tính.

Ví dụ, bộ chọn sau sẽ chọn ra tất cả các phần tử img, có thuộc tính title

img[title] {border: 3px solid;}

element[attribute=“giá trị”]

Bộ chọn này sẽ chọn ra các phần tử với tên và giá trị thuộc tính cụ thể. Vì giá trị có phân biệt chữ hoa/chữ thường nên cần xác định giá trị chính xác.

Ví dụ, bộ chọn sau sẽ chọn ra tất cả các phần tử img, có tên thuộc tính là title và giá trị là “con vịt”,

img[title="con vịt"] {border: 3px solid;}

element[attribute~=“giá trị”]

Bộ chọn này sẽ chọn ra các phần tử với tên thuộc tính và một phần giá trị của thuộc tính.

Ví dụ, bộ chọn sau sẽ chọn ra tất cả các phần tử img, mà trong giá trị thuộc tính title có chứa từ “con”, như vậy, các phần tử img với title có giá trị là “con gà” hay “ vịt con” đều được chọn.

img[title~="con"] {border: 3px solid;}

element[attribute|=“giá trị”]

Bộ chọn này sẽ chọn ra các phần tử với tên thuộc tính và giá trị thuộc tính sẽ bắt đầu bằng một từ cụ thể, sau đó là dấu gạch nối (hyphen) (-).

Ví dụ, bộ chọn sau sẽ chọn ra tất cả các phần tử img, mà trong giá trị thuộc tính title bắt đầu là từ “con”, sau đó là dấu (-), như vậy các phần tử img với title có giá trị là “con-gà” hay “con-vịt” đều được chọn.

img[title|="con"] {border: 3px solid;}

element[attribute^=“phần đầu của giá trị”]

Bộ chọn này sẽ chọn ra các phần tử, với tên thuộc tính và phần đầu của giá trị được xác định trước.

Ví dụ, bộ chọn sau sẽ chọn ra tất cả các phần tử img, mà trong giá trị thuộc tính src bắt đầu là chuỗi “/images/icons”.

img[src^="/images/icons"] {border: 3px solid;}

element[attribute$=“phần cuối của giá trị”]

Bộ chọn này sẽ chọn ra các phần tử, với tên thuộc tính và phần cuối của giá trị được xác định trước.

Ví dụ, bộ chọn sau sẽ chọn ra tất cả các phần tử img, mà trong giá trị thuộc tính src kết thúc là chuỗi “.jpg”.

img[src^=".jpg"] {border: 3px solid;}

element[attribute\*=“một phần giá trị bất kì”]

Bộ chọn này sẽ chọn ra các phần tử, với tên thuộc tính và một phần bất kì trong giá trị thuộc tính.

Ví dụ, bộ chọn sau sẽ chọn ra tất cả các phần tử img, mà trong giá trị thuộc tính title có chứa chuỗi “on”. Như vậy các phần tử img với title có giá trị là “con-gà” hay “con-vịt” hay “anh-chon” đều được chọn.

img[title\*="on"] {border: 3px solid;}\

code html <!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Attribute Selectors Example</title>

    <style>

        /\* Bộ chọn dựa vào thuộc tính \*/

        img[title] {

            border: 3px solid blue; /\* Tất cả img có thuộc tính title \*/

        }

        img[title="con vịt"] {

            border: 3px solid green; /\* img có title là "con vịt" \*/

        }

        img[title~="con"] {

            border: 3px solid orange; /\* img có title chứa từ "con" \*/

        }

        img[title|="con"] {

            border: 3px solid purple; /\* img có title bắt đầu bằng "con-" \*/

        }

        img[src^="/images/icons"] {

            border: 3px solid red; /\* img có src bắt đầu bằng "/images/icons" \*/

        }

        img[src$=".jpg"] {

            border: 3px solid yellow; /\* img có src kết thúc bằng ".jpg" \*/

        }

        img[title\*="on"] {

            border: 3px solid pink; /\* img có title chứa "on" \*/

        }

    </style>

</head>

<body>

    <h1>Attribute Selectors Example</h1>

    <img src="/images/icons/icon1.png" title="con vịt" alt="Con vịt">

    <img src="/images/icons/icon2.png" title="con gà" alt="Con gà">

    <img src="/images/icons/icon3.jpg" title="anh chồn" alt="Anh chồn">

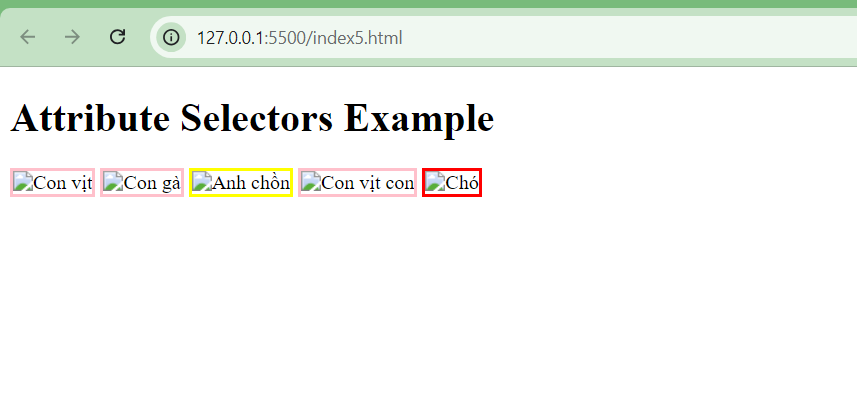
    <img src="/images/icons/icon4.jpg" title="con vịt con" alt="Con vịt con">

    <img src="/images/icons/icon5.png" title="chó" alt="Chó">

</body>

</html>

Kết quả:



**Bài tập 2. Làm một menu đơn giản như hình sau:**

****

**Code html**

<!DOCTYPE html >

<html>

<head>

    <meta charset="utf-8">

    <title>Menu</title>

    <link rel="stylesheet" href="css/style.css">

<style>

</style>

</head>

<body>

    <nav>

        <h2>Học làm web</h2>

        <ul>

            <li><a href="frontend.html">Front-end</a></li>

            <li><a href="backend.html">Back-end</a></li>

            <li><a href="devops.html">Devops</a></li>

        </ul>

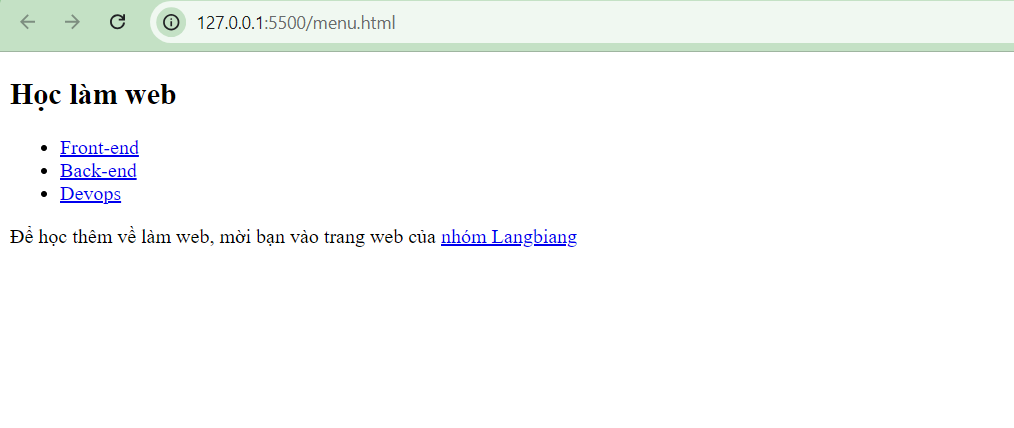
    </nav>

    <p>Để học thêm về làm web, mời bạn vào trang web của <a href="https://legiacong.blogspot.com">nhóm Langbiang</a></p>

</body>

</html>

**Kết quả:**

**1.1.6 Câu hỏi ôn tập**

**Câu 1. A \_\_\_\_\_\_\_ is a selector that selects elements that are in a specific state, e.g. they are the first element of their type, or they are being hovered over by the mouse pointer.**

A. pseudo-element

**B. pseudo-class**

C. id

D. class

**Câu 2. In CSS, a \_\_\_\_\_\_\_\_\_ selector applies styles to parts of your document content in scenarios where there isn't a specific HTML element to select.**

A. element

B. pseudo-class

C. id

**D. pseudo-element**

**Câu 3. Ví dụ, mã nguồn HTML có nhiều phần tử img, trong đó có 2 phần tử img với title có giá trị lần lượt là “con gà” và “ vịt con”. Bạn chỉ muốn chọn ra 2 phần tử img này, vậy bạn sẽ sử dụng bộ chọn nào dưới đây:**

A. img[title] {border: 3px solid;}

**B. img[title~="con"] {border: 3px solid;}**

C. img[title|="con"] {border: 3px solid;}

D. img[title="con"] {border: 3px solid;}

**Câu 4. Ví dụ, mã nguồn HTML có nhiều phần tử img, trong đó có một số phần tử img có thuộc tính title. Bạn chỉ muốn chọn ra các phần tử img với thuộc tính title đi kèm (mà không quan tâm tới giá trị của thuộc tính), thì bạn sẽ sử dụng bộ chọn nào dưới đây:**

**A. img[title] {border: 3px solid;}**

B. img[title~=\*] {border: 3px solid;}

C. img[title|="\*"] {border: 3px solid;}

D. img[title="\*"] {border: 3px solid;}