

**TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

A blue and white logo

Description automatically generated with low confidence

**CÔNG NGHỆ WEB**

**Đề tài : Thiết Kế Website Bán Ô Tô**

**Nhóm 14 \_ 20CN5**

**Giảng viên hướng dẫn : Nguyễn Thị Hạnh**

**Nhóm Sinh viên thực hiện:**

1. **Trịnh Thị Lữ**
2. **Nguyễn Thị Phương**
3. **Trần Thị Thu Chang**

**Hà Nội \_2023**

**Lớp : 20VL**

**Giảng viên giảng dạy:**

Giảng viên giảng dạy : Lương Thị Bích Phượng

HÀ NỘI, 10/2023

# 1. Giới thiệu về HTML

**a. Khái niệm HTML**

- HTML (Hypertext Markup Language) là ngôn ngữ đánh dấu siêu văn bản được sử dụng để tạo và cấu trúc các thành phần trong trang web hoặc ứng dụng, phân chia các đoạn văn, heading, titles, blockquotes.v.v.

- HTML không phải là một ngôn ngữ lập trình, vì vậy không thể tạo ra các chức năng “động” được. Mục đích sử dụng tương tự như Microsoft Word, chỉ dùng để xây dựng bố cục và định dạng văn bản.

**b. Ưu và nhược điểm**

|  |  |
| --- | --- |
| Ưu điểm | Nhược điểm |
| Không giới hạn tài nguyên hỗ trợ và cộng đồng người dùng. | HTML chỉ được sử dụng cho những trang web tĩnh. Nếu muốn tạo thêm những tính năng động thì lập trình viên phải sử dụng thêm các ngôn ngữ khác như javaScript hoặc PHP. |
| Khả năng hoạt động mượt mà trên hầu hết các trình duyệt web phổ biến hiện nay. | Có khá nhiều các yếu tố trùng lặp như header và footer nếu trang HTML không được tạo riêng biệt. |
| Đơn giản trong việc đọc HTML. | Cách đọc và hiển thị file HTML của trình duyệt khó kiểm soát. |
| Các markup sử dụng trong HTML thường có độ đồng nhất cao, ngắn gọn, dễ nhìn, dễ hiểu. | Tồn tại một số trình duyệt cập nhật chậm, không có khả năng hỗ trợ những tính năng mới của HTML. |
| Các mã nguồn mở được sử dụng hoàn toàn miễn phí. |  |
| Được vận hành bởi W3C, HTML chuẩn web. |  |
| Tích hợp các loại ngôn ngữ backend khác như PHP, Node.js,… một cách dễ dàng. |  |

**Bảng 1: Ưu và nhược điểm của HTML**

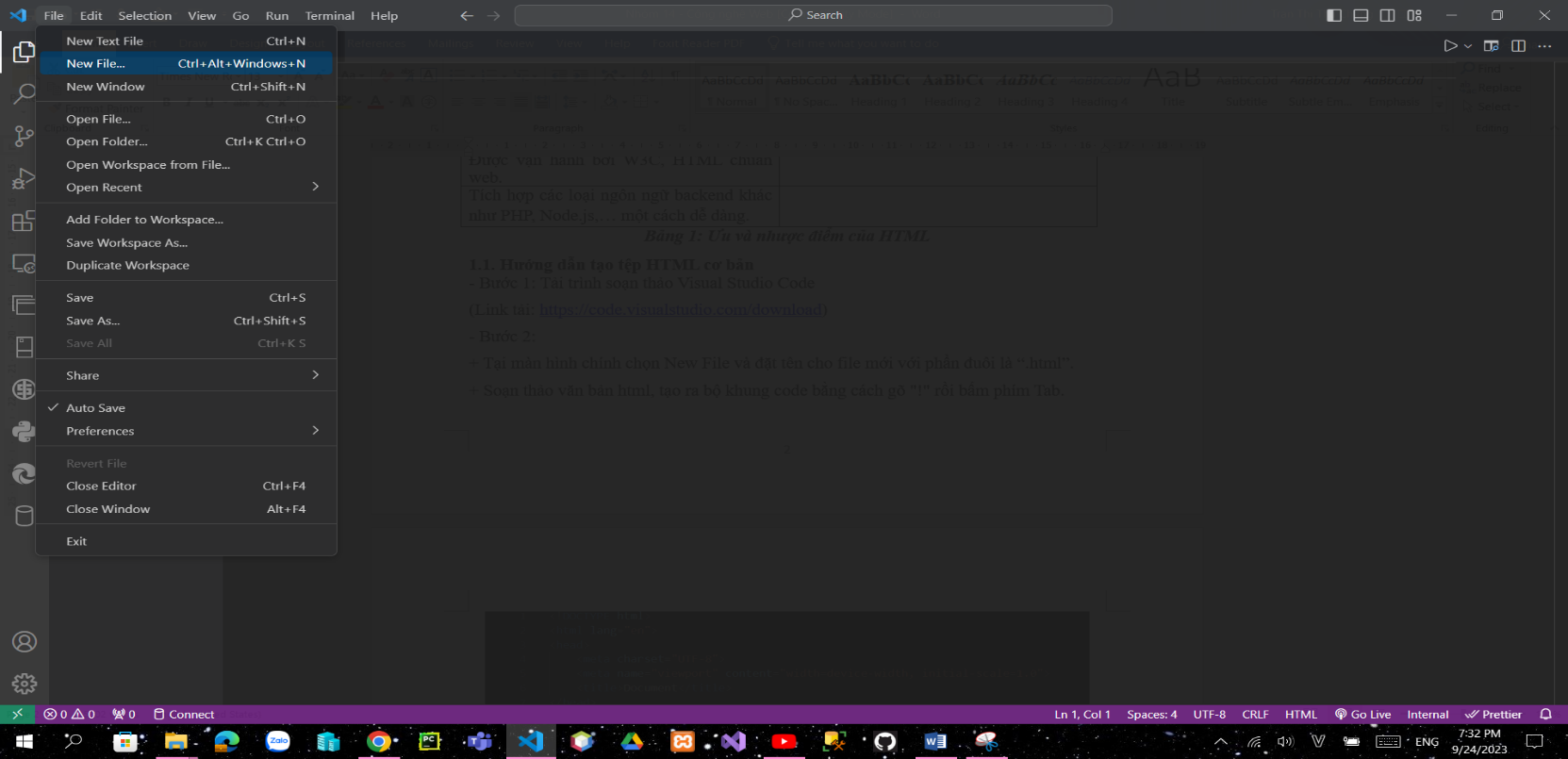
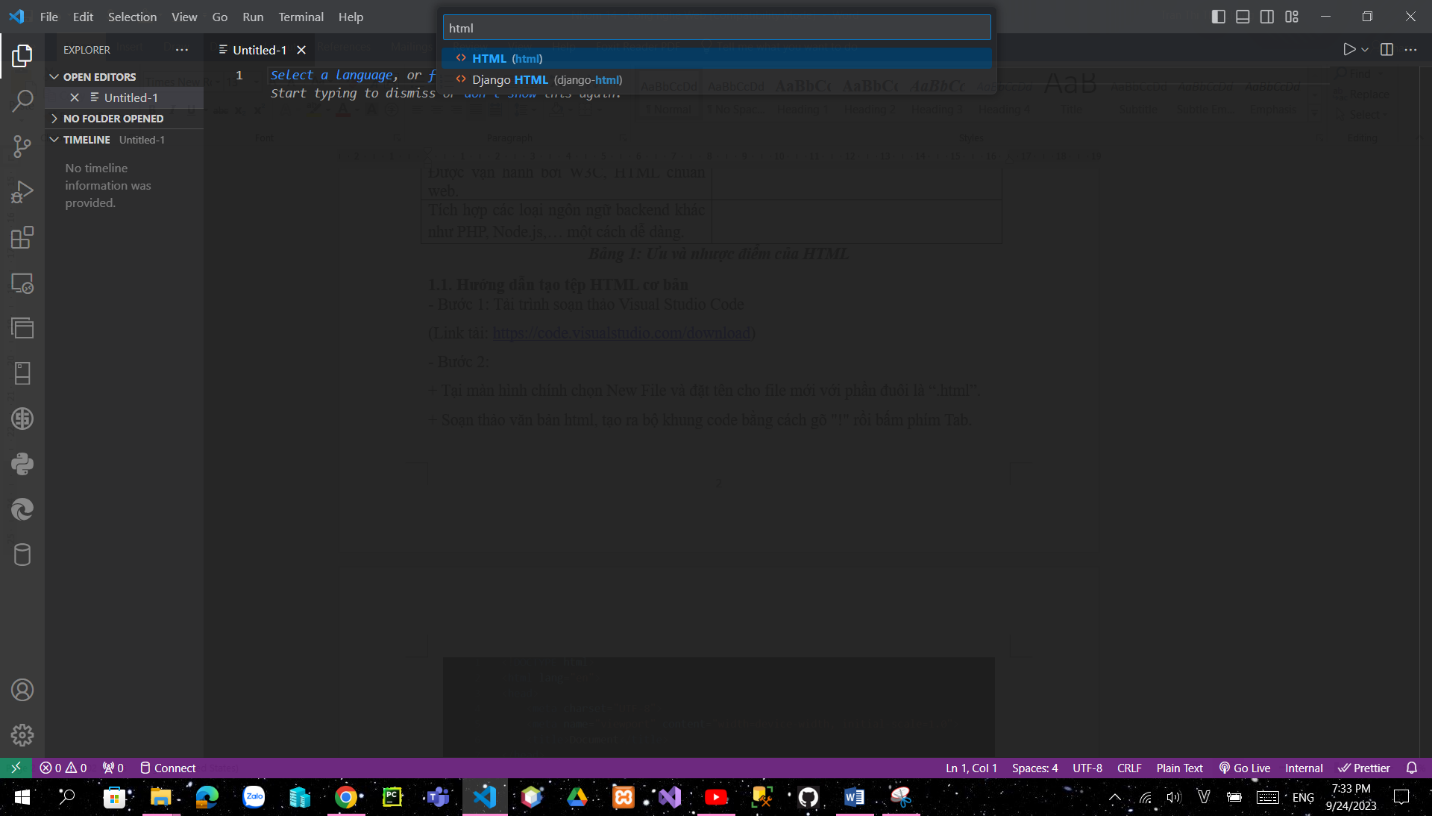
## 1.1. Hướng dẫn tạo tệp HTML cơ bản

- Bước 1: Tải trình soạn thảo Visual Studio Code

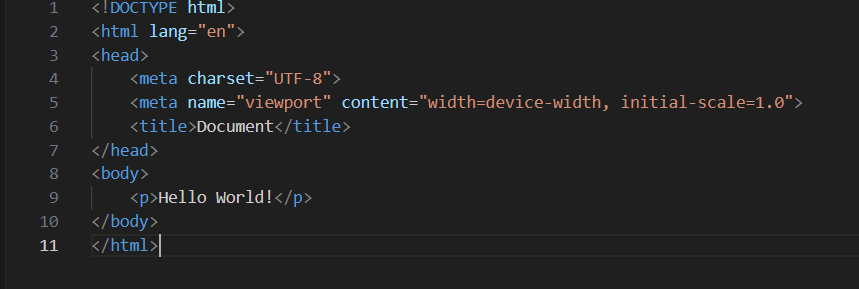
(Link tải: <https://code.visualstudio.com/download>)

- Bước 2:

+ Tại màn hình chính chọn New File và đặt tên cho file mới với phần đuôi là “.html”.

+ Soạn thảo văn bản html, tạo ra bộ khung code bằng cách gõ "!" rồi bấm phím Tab.



- Bước 3: Xem hiển thị trang.

+ Lưu tập tin vừa tạo bằng phím tắt Ctrl+S.

+ Vào vị trí lưu file, click chuột phải để xem trang trên trình duyệt máy tính đang sử dụng (Safari, Firefox, Chrome, Edge, Opera).



## 1.2. Tạo trang web với tiêu đề ,đoạn văn ,hình ảnh và siêu liên kết

**\* Tiêu đề:**

-Thẻ <title>: dùng để xác định tiêu đề tab trình duyệt cho trang web và được đặt bên trong phần tử <head>.

-Thẻ tiêu đề (Heading): được xác định bằng thẻ đánh dấu từ <h1> tới <h6>, trong đó <h1> là tiêu đề quan trọng nhất còn <h6> là ít quan trọng nhất.

**\* Đoạn văn bản:**

- Thẻ văn bản <p> : được dùng để thêm đoạn văn cho trang.

**\* Hình ảnh:**

- Thẻ hình ảnh <img>: là một thẻ trống và chỉ có thể chứa danh sách các thuộc tính và không có thẻ đóng.

- Xác định rõ tên file ảnh trong thuộc tính src.

- Thuộc tính alt là một thuộc tính ủy quyền, nếu hình ảnh không hiển thị, nó sẽ xác định một văn bản thay thế cho hình ảnh đó.

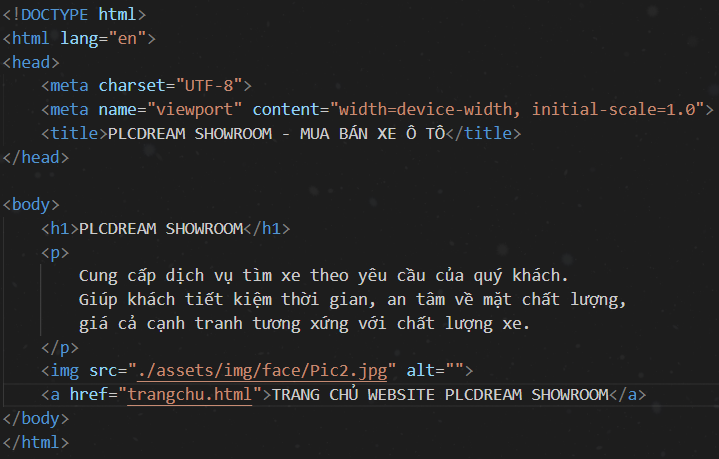
- Cấu trúc: <img src="đường dẫn url chứa hình ảnh" ... alt=" "/>

**\* Siêu liên kết: chứa liên kết điều hướng**

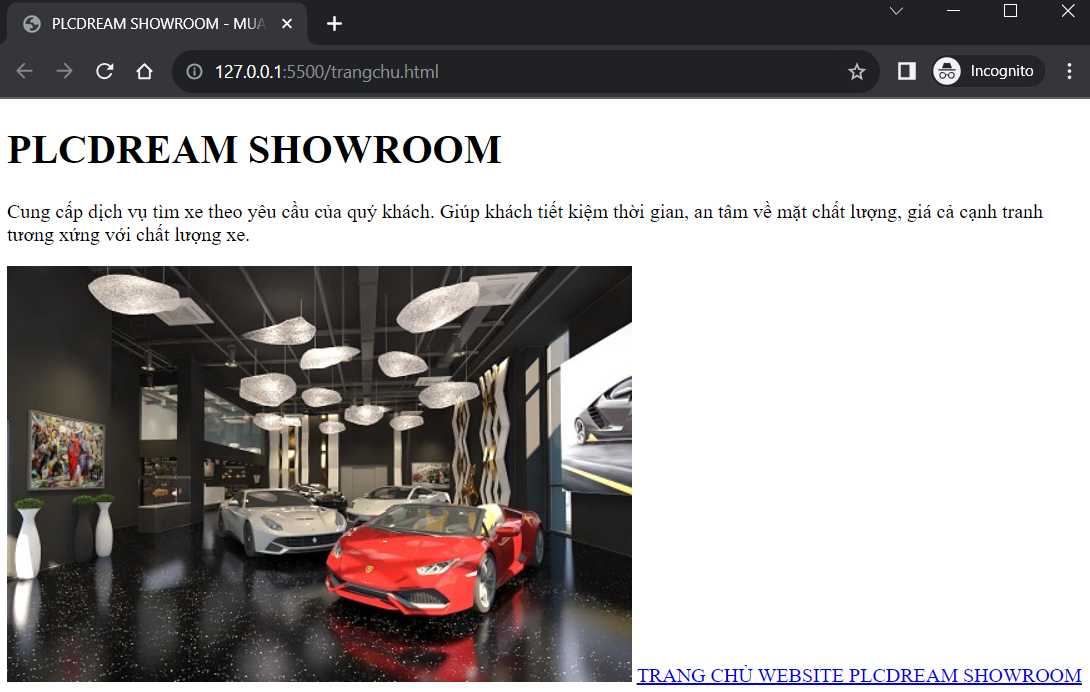
- Khái niệm:Siêu liên kết cho phép liên kết tài liệu với các tài liệu hoặc tài nguyên khác, liên kết đến các phần cụ thể của tài liệu hoặc cung cấp ứng dụng tại một địa chỉ web. Hầu như bất kỳ nội dung web nào cũng có thể được chuyển đổi thành một liên kết để khi được nhấp hoặc kích hoạt, trình duyệt web sẽ chuyển đến một địa chỉ web khác.

- Cấu trúc: <a href="duong-dan-lien-ket.html">Siêu liên kết của bạn</a>

=> Tóm lại, ví dụ trang web với tiêu đề, đoạn văn, hình ảnh và siêu liên kết:



=> Kết quả khi chạy chương trình:



## 1.3. Hướng dẫn về cú pháp và các phần tử HTML cơ bản

**a. Cú pháp comment trong HTML** : **<!---**Nội dung comment**-->**

Ví dụ :

<!--Nhóm 14 -->

<!--Đề tài: thiết kế website bán ô tô -->

**b. Cấu trúc của file HTML cơ bản**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title> Nhóm 14 – Công Nghê Web </title>**

**</head>**

**<body>**

**</body>**

**</html>**

* **<!DOCTYPE html>** dùng để xác định rằng code bạn đang dùng là html.
* **<html> </html>** chứa toàn bộ các dòng lệnh html bên – thẻ định dạng tiêu đề.
* **<head></head>**chứa thông tin của webpage - thẻ đầu trang và đặc trưng bởi **<title></title>** mặc định sẽ không được hiển thị trên web - thẻ tiêu đề.
* **<body></body>** chứa toàn bộ nội dung html sẽ hiển thị lên trên trang web - thẻ nội dung.

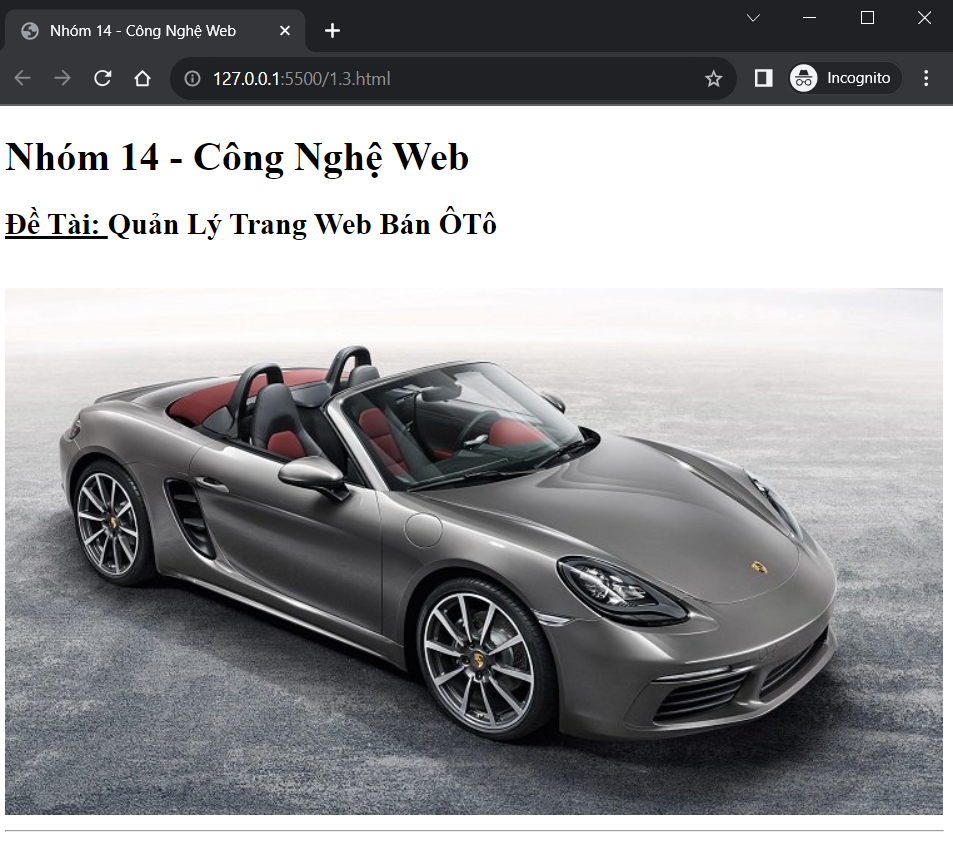
**c. Các thẻ thường gặp trong HTML :**

|  |  |
| --- | --- |
| **Các thẻ** | **Cấu trúc + Chức năng** |
| Thẻ tiêu đề | <h1> Nội dung </h1>  <h2> Nội dung </h2> |
| Thẻ văn bản | <p> </p> |
| Thẻ hình ảnh | <img src= “ảnh” > |
| Thẻ link | <a href= “https://www.nhom4.com” |
| Thẻ định dạng | <u> bị gạch chân </u> |
|  | <b> In đậm </b> |
|  | <i> Chữ in nghiêng </i> |
| Thẻ căn lề | <p align= “left”> Căn trái </p> |
|  | <p align= “right”> Căn phải </p> |
|  | <p align= “center”> Căn giữa </p> |
|  | <p align= “justify”> Căn đều </p> |
| Thẻ nhãn | <label></label> |
| Thẻ ngang | <hr> thẻ ngang </hr> |
| Thẻ table | <table> </table> |
|  | <th> Tiêu đề bảng </th> |
|  | <tr> Tạo hàng </tr> |
|  | <td> Tạo ô </td> |
| Xuống dòng | <br> |
| Thẻ danh sách | <ul> </ul> |
|  | <ol> </ol> |
|  | <li> </li> |
| Thẻ video | <video > </video> |
| ... | ... |

d. Ví dụ một vài thẻ HTML :



=>Kết quả :



# 2. Cơ bản CSS

## 2.1. Hướng dẫn cách thêm kiểu cho trang web bằng CSS

**- CSS (Cascading Style Sheets** ) là ngôn ngữ [biểu định kiểu](https://developer.mozilla.org/en-US/docs/Web/API/StyleSheet) được sử dụng để mô tả cách trình bày tài liệu được viết bằng [HTML](https://developer.mozilla.org/en-US/docs/Web/HTML) hoặc [XML](https://developer.mozilla.org/en-US/docs/Web/XML/XML_introduction) . CSS mô tả cách các phần tử được hiển thị trên màn hình, trên giấy, trong lời nói hoặc trên các phương tiện khác. CSS là một trong những ngôn ngữ cốt lõi của **web mở** và được tiêu chuẩn hóa trên các trình duyệt Web theo [thông số kỹ thuật của W3C](https://www.w3.org/Style/CSS/#specs)

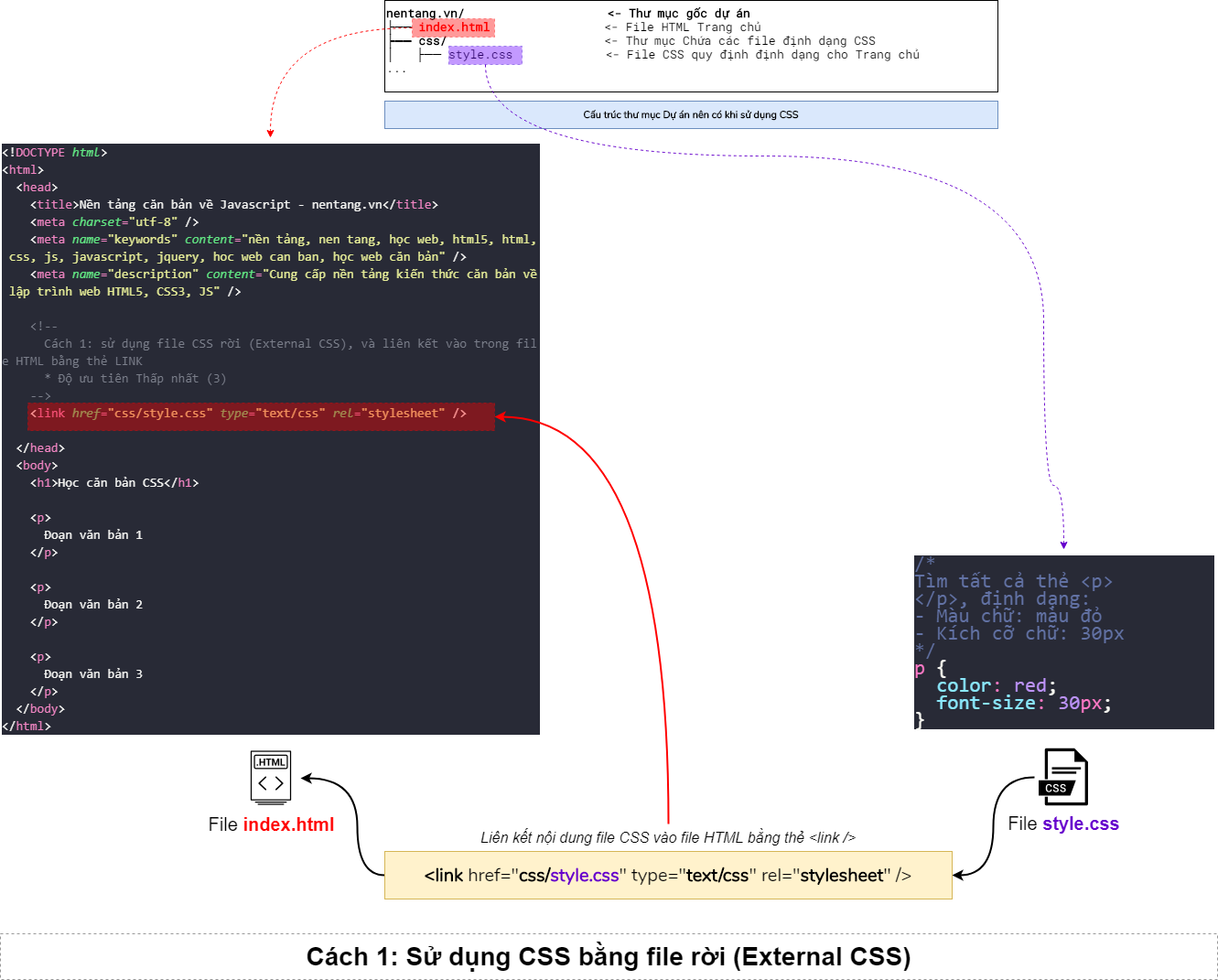
- Các cách nhúng CSS vào HTML : 4 cách

### 2.1.1. Sử dụng External CSS (Tập tin CSS rời)

- Một file CSS

- Một file HTML

🡺Nhiều ưu điểm và thường được dùng trong các dự án



### 2.1.2. Sử dụng Internal CSS

- Cách này sử dụng đoạn code CSS bên trong file html bằng cách sử dụng thẻ

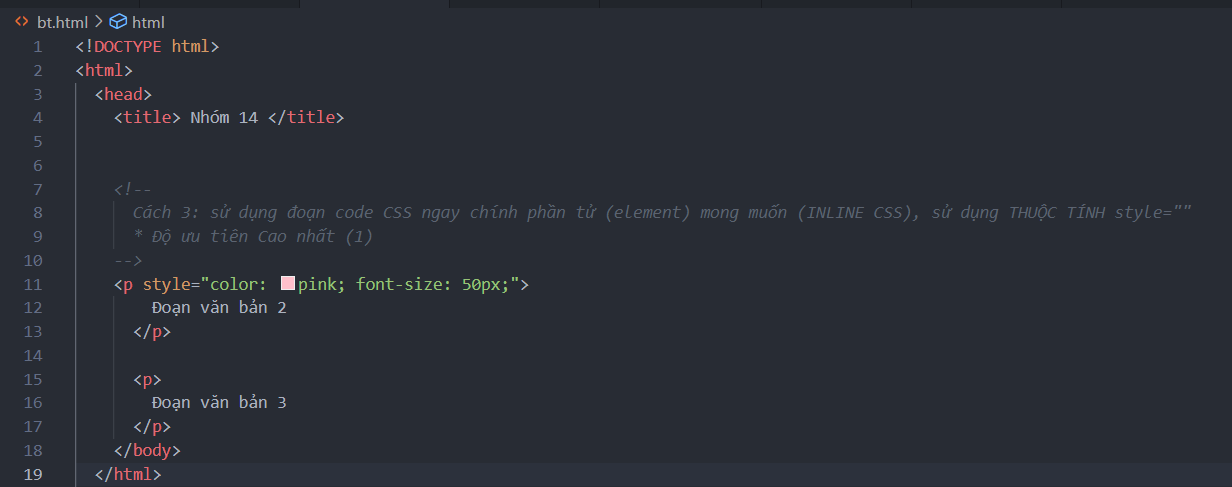
<style> Cú pháp CSS </style>





### 2.1.3. Sử dụng Inline CSS

- Sử dụng để áp dụng trực tiếp CSS vào thẻ HTML bằng thuộc tính Style



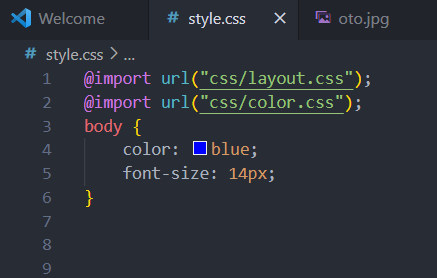
### 2.1.4. Sử dụng Import CSS

@import CSS là một cách khác để tải một file CSS bên ngoài vào trong HTML. Câu lệnh @import hướng dẫn trình duyệt tải một file CSS bên ngoài và sử dụng . Hai cách

-Cách 1 : Nhập CSS vào HTML



-Cách 2 : Nhập một file CSS trong một file CSS khác

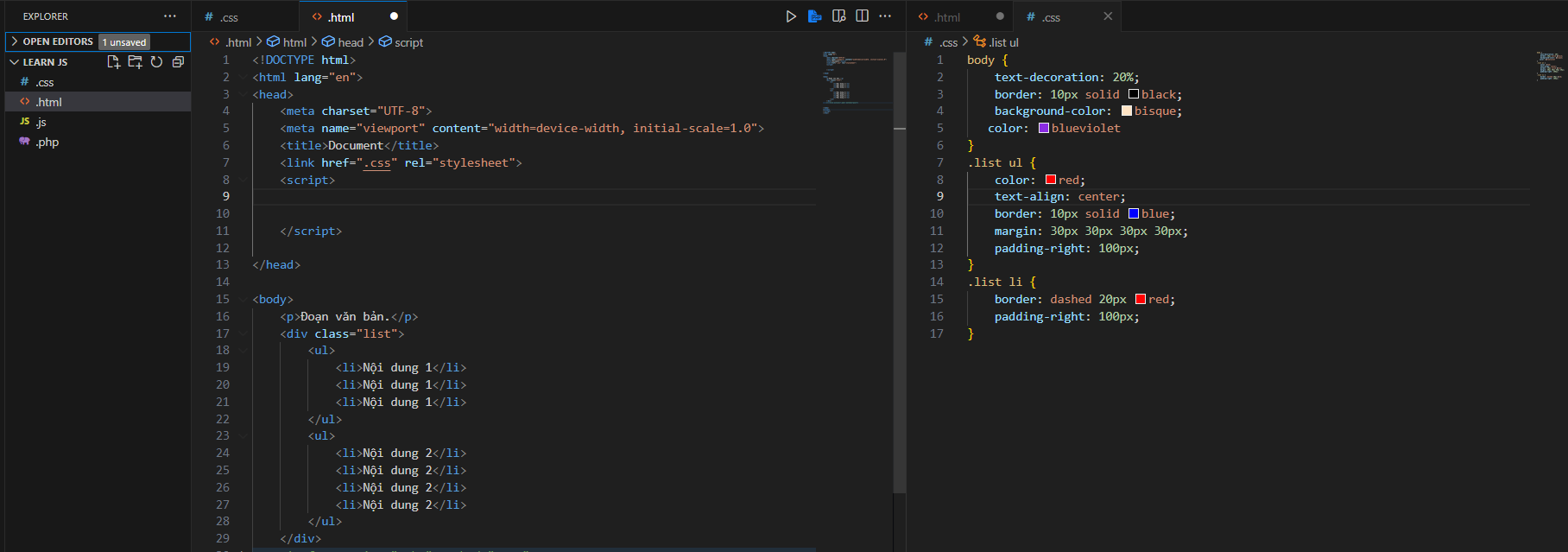


## 2.2. Tạo các lớp và áp dụng kiểu bằng cách sử dụng lựa chọn.

**\* Cú pháp CSS**

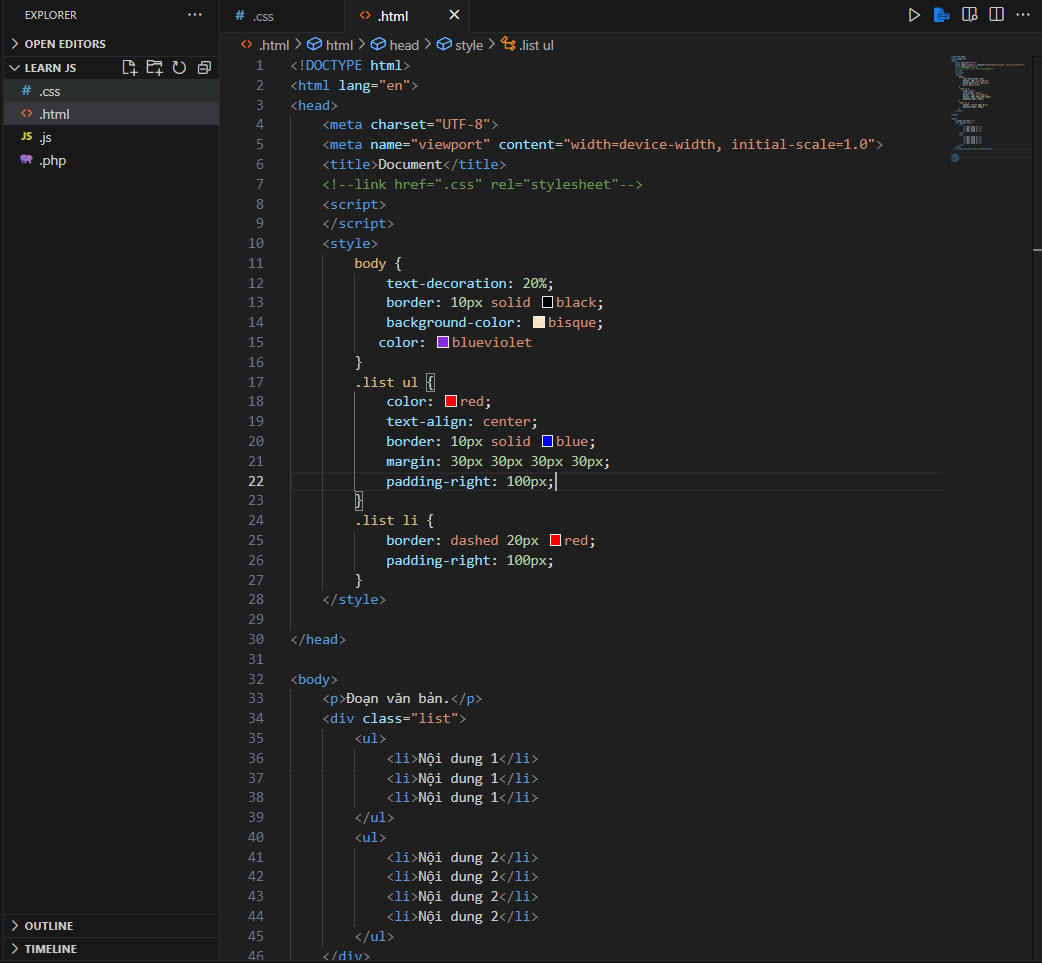


### 2.2.1. Sử dụng External CSS (Tập tin CSS rời)



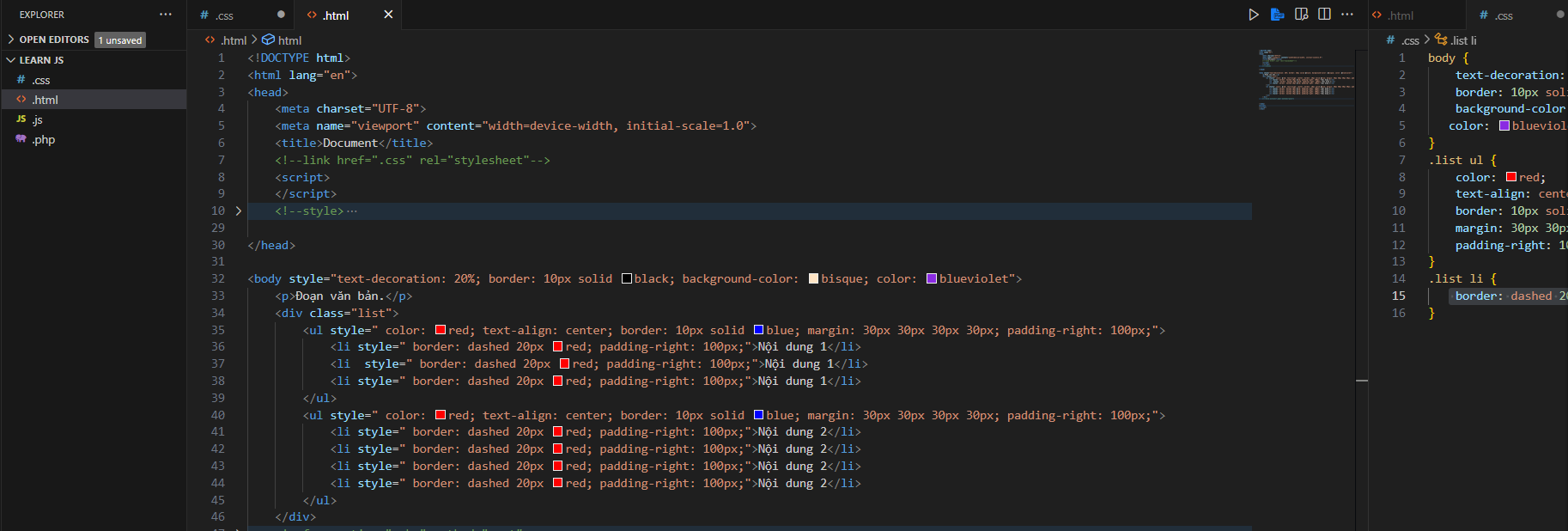
Kết nối tập tin rời .css với .html bằng dòng <link href=”.css”>.

### 2.2.2. Sử dụng Internal CSS

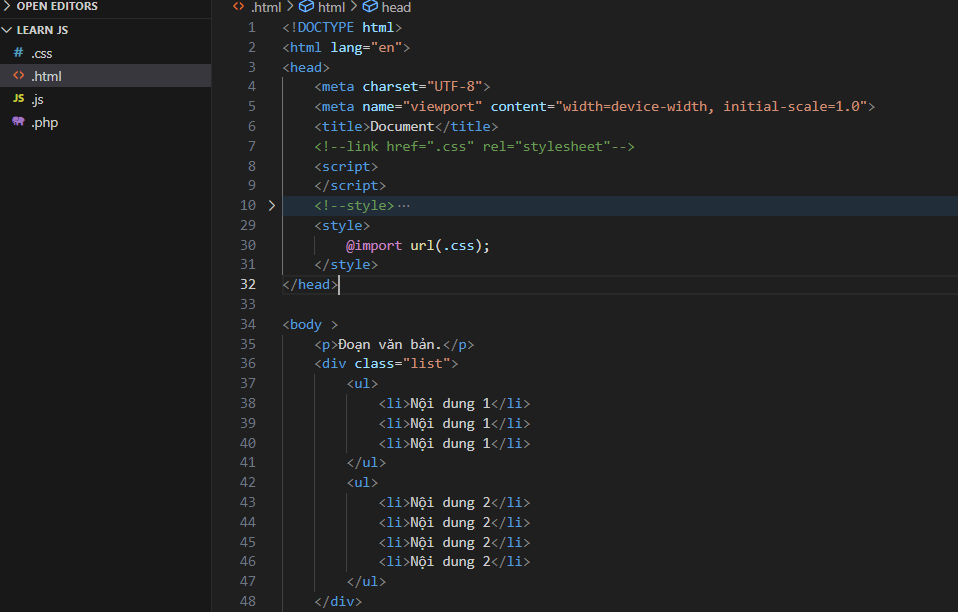


Ta viết cs strong thẻ <style></style>.

### 2.2.3. Sử dụng Inline CSS



### 2.2.4. Sử dụng Import CSS



## 2.3. Thử nghiệm với các thuộc tính CSS cơ bản như font chữ, màu sắc, độ rộng và chiều cao.

### ***2.3.1. Thuộc tính font chữ***

- Font: thiết lập tất cả thuộc tính font chữ trong khai báo.

- Font-Family: xác định nhiều tên font chữ khác nhau dưới dạng "dự phòng".

- Font-Style: sử dụng để chỉ định kiểu phông chữ cho text. (normal-bình thường, italic-in nghiêng, oblique-gần giống italic)

- Font-Size: xác định kích thước phông chữ (% hoặc px).

- Font-Weight: xác định mức độ đậm nhạt của phông chữ.

- Responsive Font-Size: thiết lập kích thước phông chữ bằng đơn vị vw (viewport width-chiều rộng của viewport).

- Font-Variant: xác định văn bản hiện thị dưới dạng chữ in hoa nhỏ(small-cap) hay không.

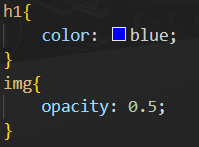


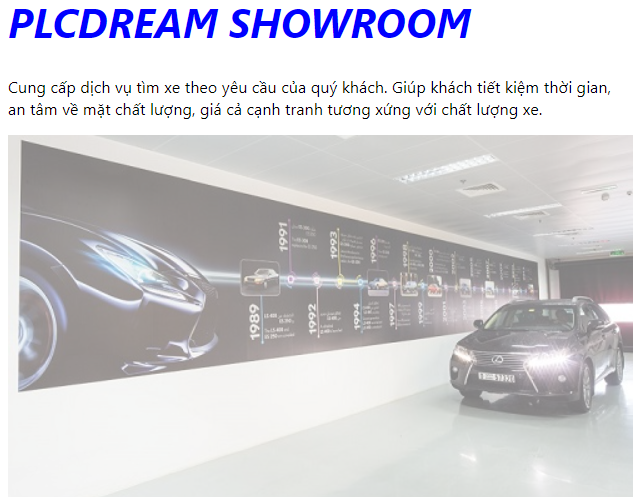


### 2.3.2. Thuộc tính màu sắc

- Color: dùng để thiết lập màu sắc cho văn bản.

- Opacity: dùng để thiết lập mức độ trong suốt của 1 phần tử.



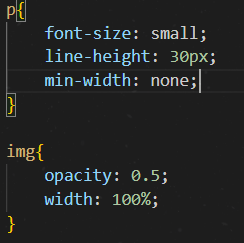


### 2.3.3. Thuộc tính độ rộng (width)

- Width: thiết lập chiều rộng (đơn vị đo lường: pixel(px), cm, %, auto,…)

- Max-width: thiết lập chiều rộng tối đa

- Min-width: thiết lập chiều rộng tối thiểu



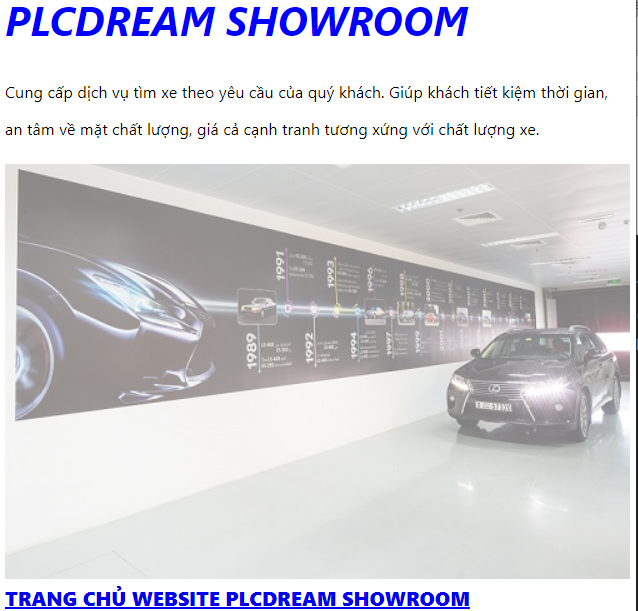
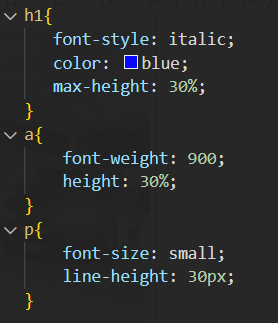
### 2.3.4. Thuộc tính chiều cao (height)

- Height: thiết lập chiều cao

- Line-height: xác định chiều cao của 1 dòng văn bản

- Max-height: thiết lập chiều cao tối đa

- Min-height: thiết lập chiều cao tối thiểu



# 3. JavaScript Cơ bản

## 3.1. Giới thiệu cú pháp JavaScript cơ bản

**a. Khái niệm JavaScript**

- Javascript là ngôn ngữ lập trình web và có thể thêm mới và update cả HTML và CSS.

- Javascript không chỉ gói gọn trong các ứng dụng front end trên trình duyệt mà còn có thể phát triển cả ứng dụng cho desktop, mobile hay server.

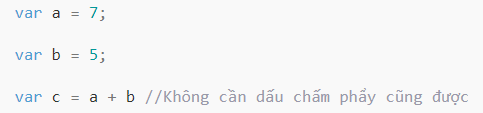
- Javascript được hỗ trợ trên tất cả các trình duyệt như Firefox, Chrome, IE… Thậm chí hỗ trợ các trình duyệt trên thiết bị di động.

* jQuery – thư viện Javascript để đơn giản hóa việc xử lý DOM.
* ReactJS – thư viện phát triển ứng dụng web hiện đại.
* NodeJS – là nền tảng phát triển ứng dụng phía server bằng javascript.

**b. Cú pháp JavaScript cơ bản**

- Một chương trình máy tính là một tập danh sách các hướng dẫn (statements) để máy tính thực thi. Và Javascript cũng vậy nhưng khác một điều là các tập lệnh này cho trình duyệt thực hiện thay vì máy tính.

- Các câu lệnh Javascript không cần phải đặt trong dấu ngoặc đơn, hay ngoặc nhọn và cuối mỗi câu lệnh cũng không cần dấu chấm phẩy.



**\* Cách đặt tên biến, hàm**

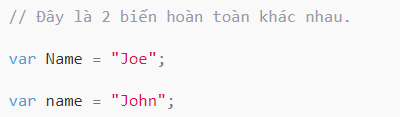
- Trong Javascript, định danh (identifiers) được sử dụng để đặt tên biến, từ khóa, hàm và labels.

- Việc định danh nói chung là theo quy tắc sau:

* Ký tự đầu tiên của định danh phải là chữ cái ( a-z, A-Z) hoặc ký tự $ hoặc dấu gạch dưới (\_)
* Không được sử dụng chữ số làm ký tự đầu tiên của định danh.



- Các định danh có phân biệt chữ hoa với chữ thường.



- Về quy tắc đặt tên các định danh điển hình có 3 loại:

* Dùng dấu gạch ngang: Ví dụ: first-name
* Dùng dấu gạch dưới. Ví dụ: first\_name
* Và trường phái **camel case**, tức là viết chữ in hoa với mỗi từ đơn. Ví dụ: FirstName

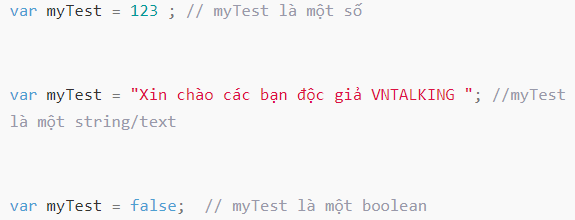
**\* Cách khai báo biến**

- Javascript hỗ trợ 3 cách để khai báo một biến, đó là sử dụng các từ khóa: let, var và const.

* Const: được dùng để khai báo hằng số, giá trị của nó không thay đổi trong suốt chương trình. Nó giống với từ khóa *final* của Java vậy.
* var: biến được khai báo với từ khóa *var* sẽ được truy cập ở bất kể đâu trong chương trình.
* let: từ khóa này giúp cho biến chỉ có thể được truy xuất bên trong phạm vi block quanh nó (block được định nghĩa bởi dấu ngoặc nhọn {} ).

**\* Kiểu dữ liệu của biến**

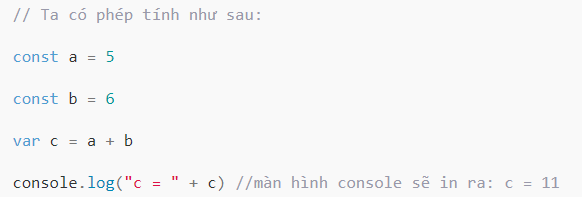
- Kiểu dữ liệu sẽ được xác định một cách tự động khi chương trình thực thi. Điều này có nghĩa là một biến có thể có nhiều kiểu dữ liệu ở những thời điểm khác nhau.



- Theo tiêu chuẩn ES7, có 7 kiểu dữ liệu:

* Kiểu boolean
* Kiểu null
* Kiểu undefined
* Kiểu số (int)
* Kiểu chuỗi (String)
* Kiểu Symbol (mới trong ECMAScript 6)
* Kiểu đối tượng (Object)

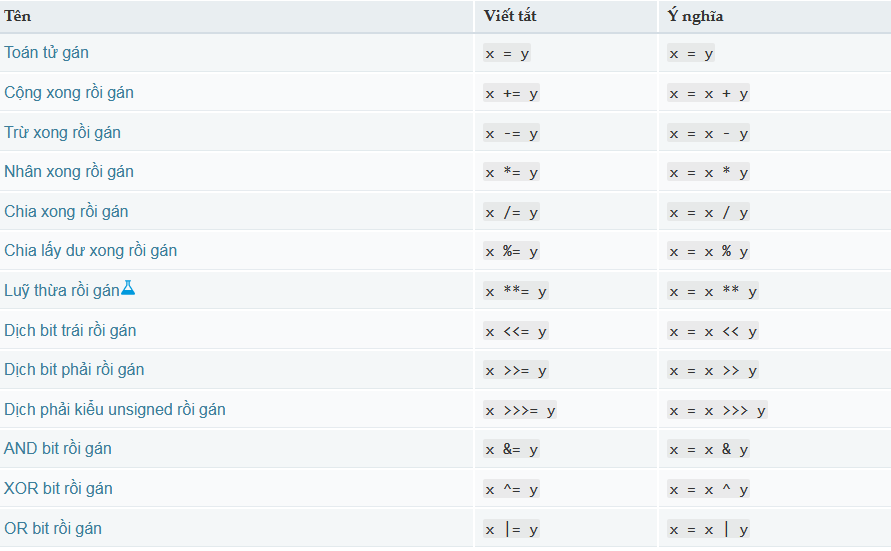
**\* Toán tử trong JavaScript**



- Trong ví dụ trên, phép tính a + b, thì phép tính cộng được gọi là toán tử (operator).

- Javascript hỗ trợ đầy đủ các loại toán tử tính toán và kết hợp.

- Bảng các toán tử được hỗ trợ:



**\* Comment trong Javascript**

- Javascript hỗ trợ 2 kiểu comment: **Single Line Comments** (comment một dòng) và **Multi-line Comments** (comment nhiều dòng).





- Chú ý:

* Nên Comment để giải thích “**Tại sao**” lại viết code đó thay vì **đoạn code làm như thế nào**. Vì người đọc code là đủ hiểu code làm như nào thế nào rồi.
* Không comment những đoạn code mà đọc code đó đã quá rõ ràng rồi.
* Không comment để lấp liếm vấn đề của đoạn code. Tức là người code sau cần phải đọc comment sử dụng đúng đoạn code nếu không sẽ bị lỗi. Thay vì đó, bạn nên viết code lại cho chặt chẽ hơn.
* Xóa ngay comment khi đoạn code đi kèm đã bị xóa để tránh dư thừa comment.

## 3.2. Hướng dẫn tạo biến ,hàm và sử dụng sự kiện

**a. Tạo biến JavaScript**

- Khai báo và sử dụng biến trong JavaScript với var và let ,quy tắc đặt tên biến và phép toán gán ,khai báo hằng số với const trong js

-Biến dùng để lưu trữ giá trị các dữ liệu hay các đối tượng

-Giá trị của biến có thể được thay đổi cập nhật trong quá trình ứng dụng hoạt động (tên biến,khai báo và khởi tạo biến …)

- Quy tắc đặt tên biến :

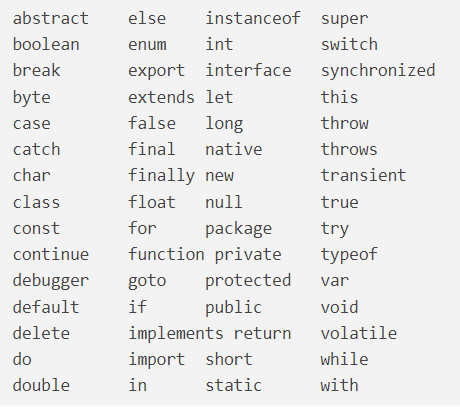
+ Tên biến được tạo ra bởi các ký tự chữ , số , \_ và $

+ Tên không được phép bắt đầu bằng số ( chỉ bắt đầu bằng ký tự chữ , $ , \_ )

+ Không được chứa các ký hiệu đặc biệt như toán học , logic (như +,-,\*,>,<,…)

+ Không chứa khoảng trắng

+ Không được đặt tên biến trùng với các từ khóa dành riêng cho ngôn ngữ JavaScript liệt kê ở bảng sau



- Khái báo và khởi tạo biến

**var** varName1 = 10; // khai báo và khởi tạo biến với từ khóa var

**let** varName2; // khai báo và không khởi tạo giá trị ban đầu

varName2 = 10; // gán giá trị

**const** varName3 = 10; // khai báo và khởi tạo biến với từ khóa const

**b. Tạo hàm JavaScript**

// Khai báo hàm

function isEven(num) {

return num % 2 === 0;

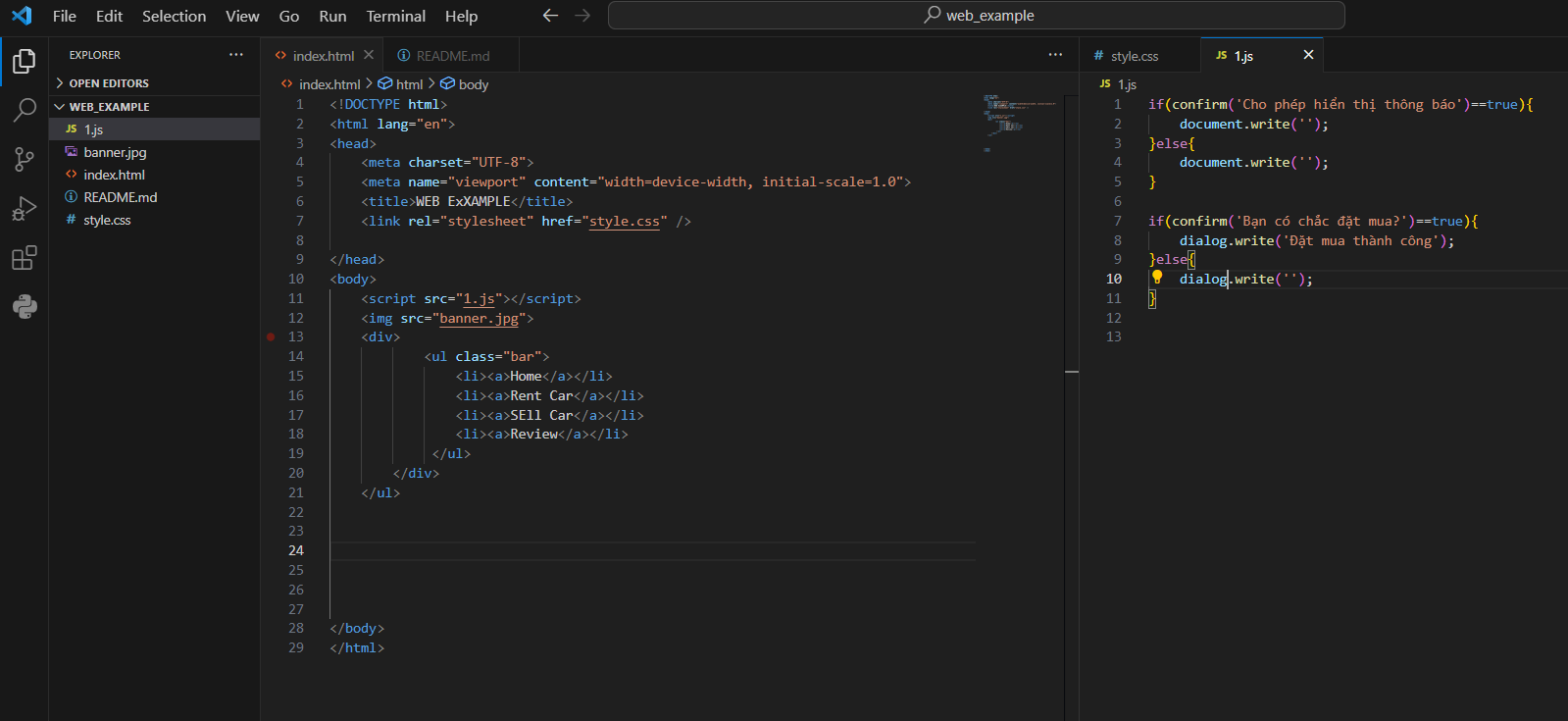
}

isEven(24); // => true

isEven(11); // => false

* Khai báo tạo một hàm trong phạm vi hiện tại với mã định nghĩa bằng hàm tên. Biến này chứa hàm object.
* Đồng thời, biến này sẽ được **nâng lên** trong phạm vi hiện tại, nhờ đó ta có thể gọi và sử dụng hàm trước khi khai báo.
* Hàm được tạo ra có nghĩa là tên thuộc tính của hàm chứa tên đối tượng. Nó rất hữu ích khi xem cuộc gọi: trong quá trình gỡ lỗi hoặc đọc thông báo lỗi.

## 3.3. Tạo các ví dụ đơn giản như hiển thị thông báo khi nút được nhấn.

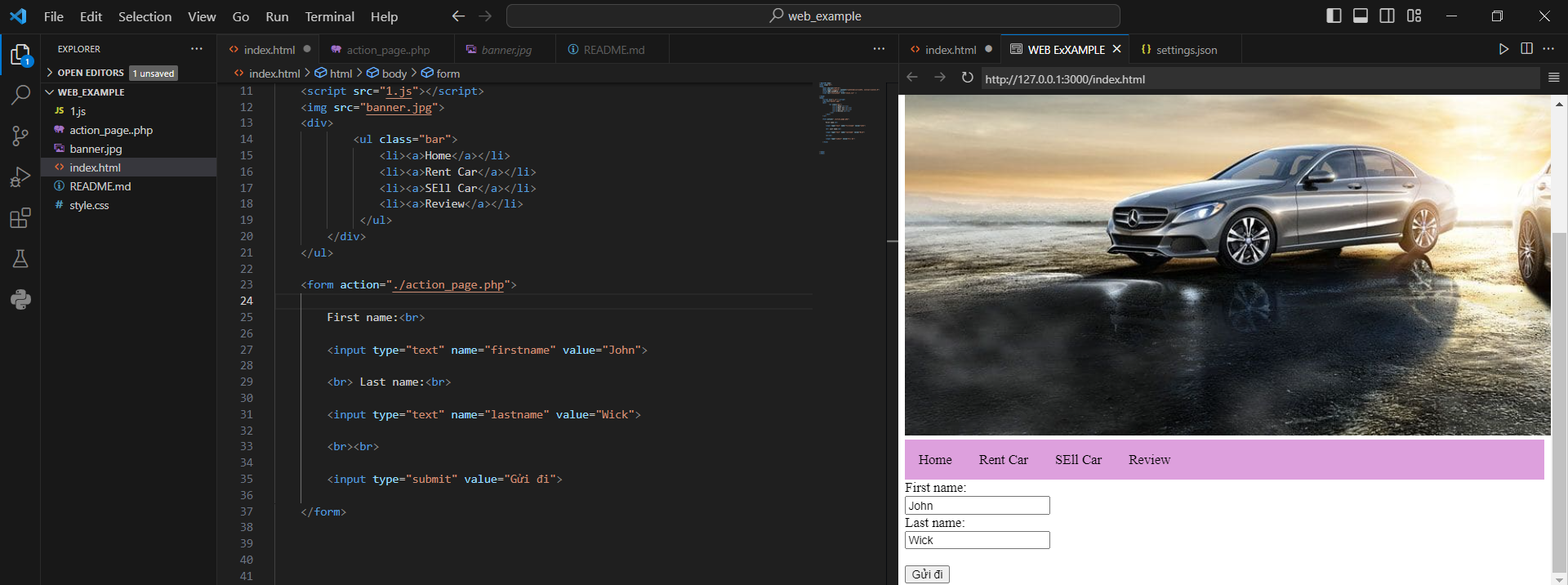


- Ví dụ: sử dụng js để kiểm tra form và hiện thông báo



# 4. HTML , Forms và Input Elements

## 4.1. Hướng dẫn cách tạo biểu mẫu HTML đơn giản với các phần tử nhập liệu như ô văn bản, ô chọn, và nút gửi.



- Phần tử HTML <form> xác định một biểu mẫu dùng để thu thập thông tin người dùng nhập vào. Các phần tử form là các kiểu phần tử nhập khác nhau như trường nhập văn bản, hộp kiểm, nút radio, nút gửi,v.v…

- Phần tử input có thể được hiển thị bằng nhiều cách khác nhau, tùy thuộc vào thuộc tính type.

|  |  |
| --- | --- |
| **Type** | **Description** |
| **<input type="text">** | Xác định trường nhập văn bản trên một dòng |
| **<input type="radio">** | Xác định các nút radio (để chọn một trong nhiều lựa chọn) |
| **<input type="submit">** | Xác định nút gửi (để gửi biểu mẫu) |

**\* Trường nhập nút radio:**

- <input type="radio"> - Xác định một nút radio.

- Các nút radio cho phép người dùng chọn MỘT trong số các lựa chọn giới hạn.

**\* Nút Submit (Gửi đi):**

- <input type="submit"> - xác định một nút để gửi dữ liệu biểu mẫu tới bộ xử lý biểu mẫu.

- Bộ xử lý biểu mẫu thường là một trang server với script để xử lý dữ liệu đầu vào.

- Bộ xử lý biểu mẫu được xác định trong thuộc tính action của form.

## 4.2. Tạo kiểu cho biểu mẫu bằng CSS.

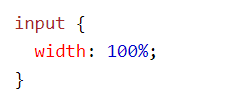
**a. Khái niệm**

- Biểu mẫu (Form) là đối tượng giúp ích cho việc nhập và hiển thị dữ liệu, thông tin một cách trực quan, có hệ thống, có thể thống kê dễ thuận tiện cho việc điều khiển một ứng dụng, hoặc thực hiện một chức năng nào đó.

**b. Tạo kiểu cho biểu mẫu HTML bằng CSS**

- Sử dụng thuộc tính *width* để xác định rõ chiều rộng của trường đầu vào:

+ Ví dụ: áp dụng cho tất cả các phần tử <input> :



+ Nếu chỉ muốn tạo kiểu cho một loại đầu vào cụ thể thì sử dụng bộ chọn thuộc tính sau:

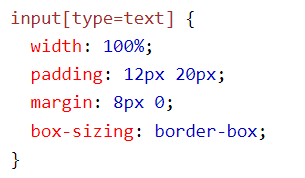
* input[type=text] – Sẽ chỉ chọn các trường văn bản
* input[type=password] – Sẽ chỉ chọn các trường mật khẩu
* input[type=number] – Sẽ chỉ chọn các trường số
* v.v.v…

- Định dạng *padding* cho trường đầu vào:

+ Sử dụng thuộc tính padding để thêm khoảng trống bên trong trường văn bản.

+ Khi có nhiều trường đầu vào khác nhau có thể thêm margin, để thêm khoảng trống bên ngoài cho trường đầu vào.

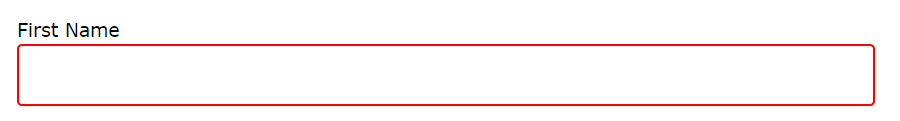
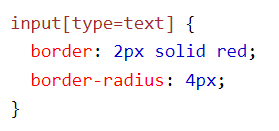
+ Đặt thuộc tính box-sizing là border-box để đảm bảo phần padding và border cuối cùng được đặt trong tổng chiều rộng và chiều cao của các phần tử.



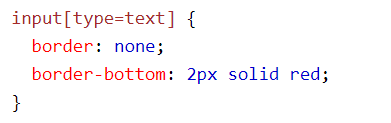
- Định dạng đường viền cho trường đầu vào:

+ Sử dụng thuộc tính *border* để thay đổi kích thước và màu sắc đường viền

+ Sử dụng thuộc tính *border-radius* để bo góc tròn

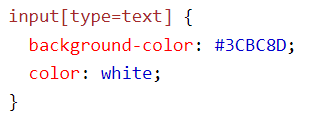


+ Nếu chỉ muốn thêm đường viền dưới cùng thì sử dụng thuộc tính *border-bottom:*



- Định dạng màu cho trường đầu vào:

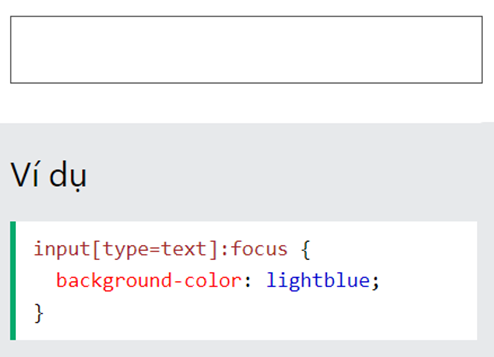
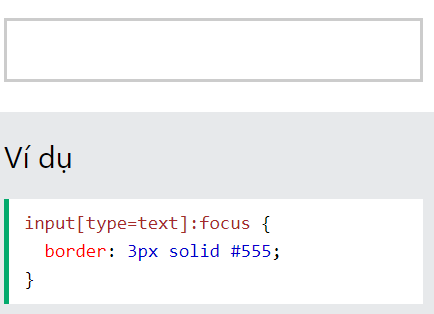
+ Sử dụng thuộc tính *background-color* để thêm màu nền cho trường đầu vào, và thuộc tính *color* thay đổi màu văn bản:



- Tạo ấn tượng cho trường đầu vào:

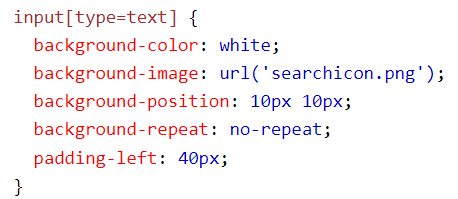
+ Một số trình duyệt mặc định sẽ thêm đường viền màu xanh xung quanh trường đầu vào khi người dùng click chuột để nhập thông tin và có thể xóa trạng thái này bằng cách nhập *outline: none;* cho trường đầu vào.

+ Sử dụng bộ chọn *:focus* để thực hiện điều gì đó với trường đầu vào khi click chuột để nhập thông tin:

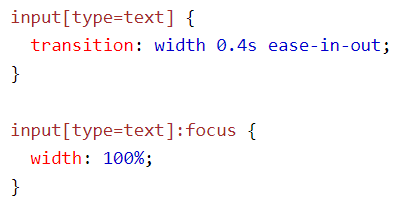
- Thêm biểu tượng/ảnh vào bên trong trường đầu vào:

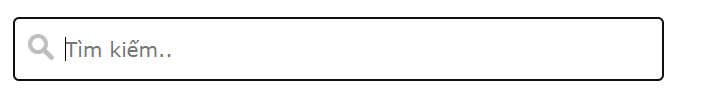
+ Sử dụng thuộc tính *background-image* và thuộc tính vị trí *background-position* để thêm icon vào trong trường đầu vào. Chú ý: thêm *padding-left* để dành khoảng trống cho biểu tượng:



- Thêm hiệu ứng tìm kiếm cho trường đầu vào:

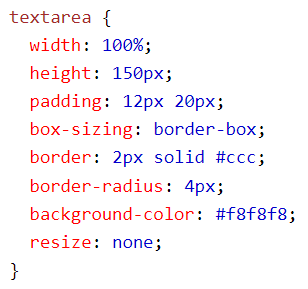
+ Sử dụng thuộc tính *transition* trong CSS để tạo hiệu ứng chiều rộng cho trường tìm kiếm khi người dùng click chuột vào trường nhập văn bản.



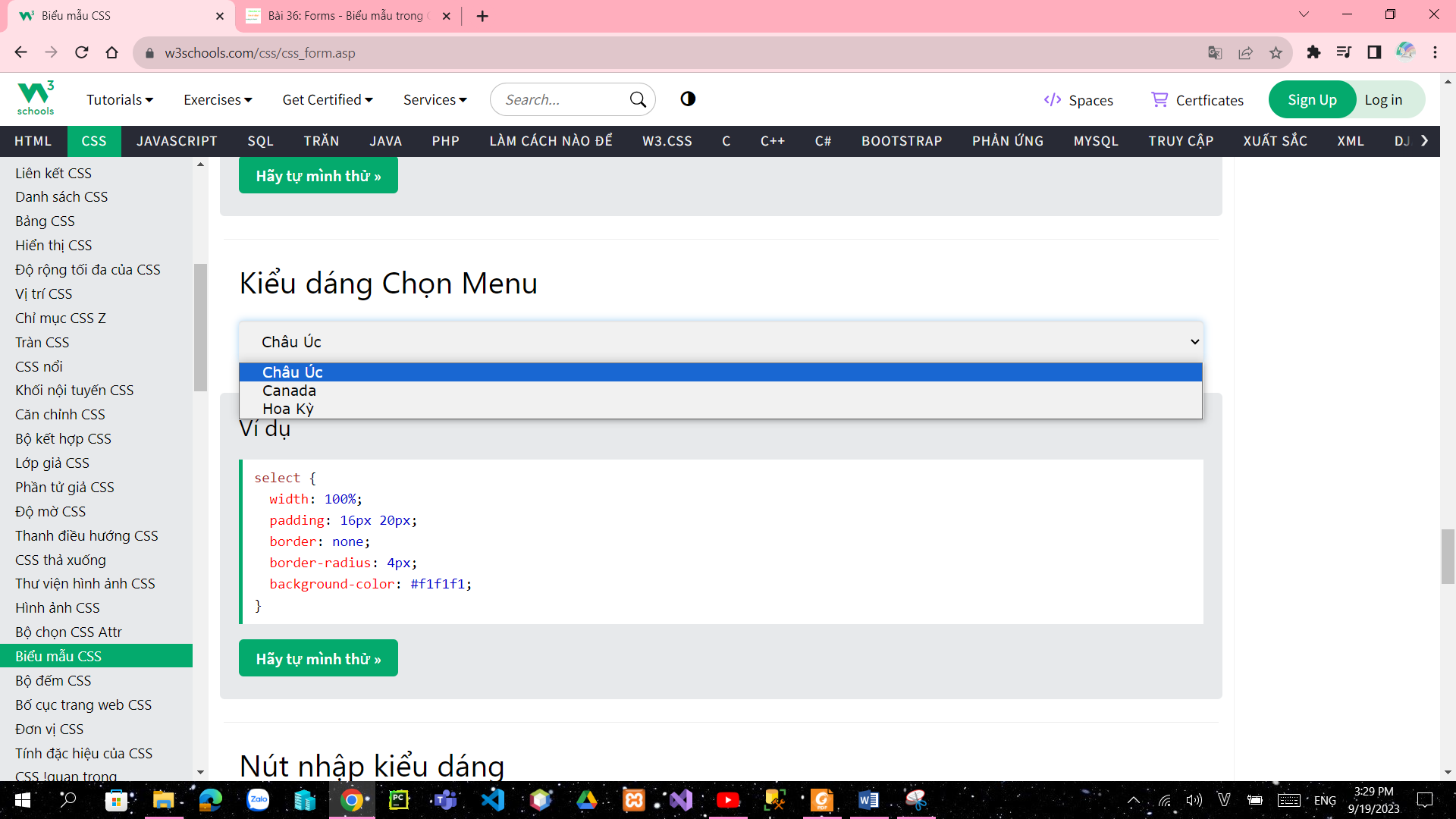
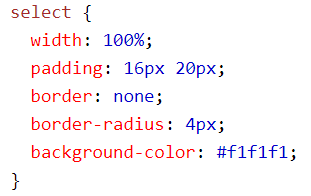


- Tạo kiểu cho Textareas:

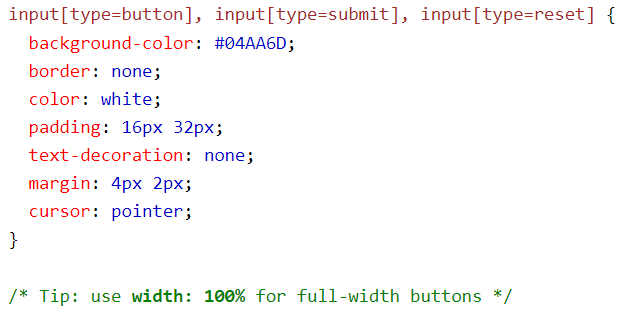
+ Sử dụng thuộc tính *resize* để ngăn các vùng văn bản bị thay đổi kích thước ( tắt “grabber” ở góc dưới cùng bên phải):

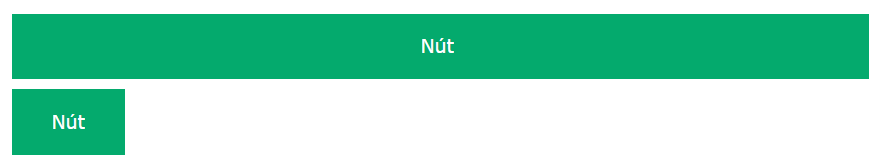


- Định dạng menu chọn:



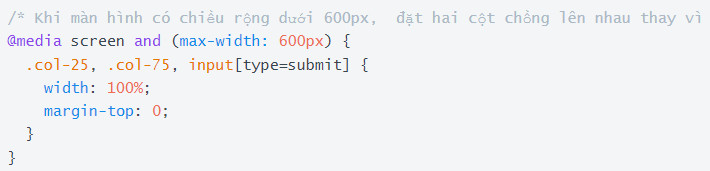
- Định dạng button:



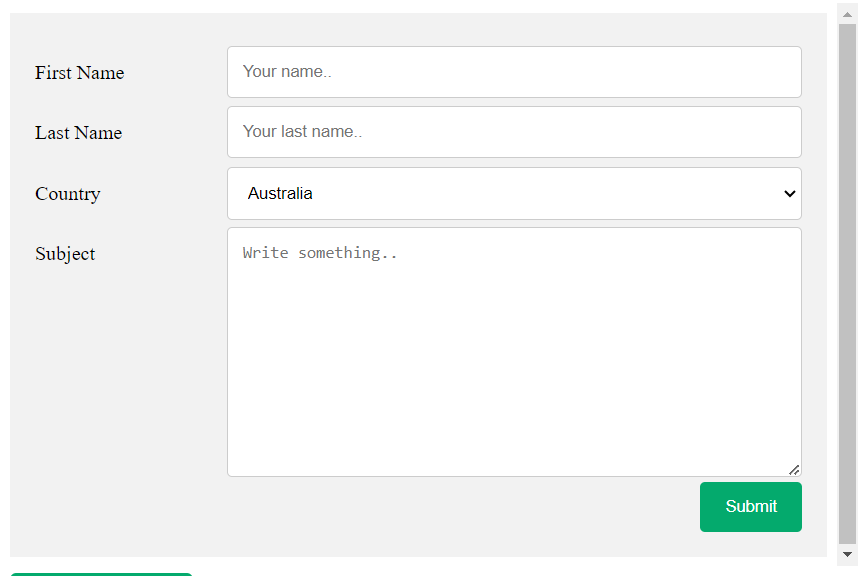


- Responsive Form:

Thay đổi kích thước cửa sổ trình duyệt để xem hiệu ứng. Khi màn hình có chiều rộng dưới 600px, đặt hai cột chồng lên nhau thay vì cạnh nhau.



=> Kết quả khi chạy chương trình:



## 4.3. Sử dụng JavaScript để xử lý dữ liệu từ biểu mẫu (vd: kiểm tra dữ liệu hợp lệ).

### ***4.3.1. Truy cập đến form dùng javascript***

Bằng cách sau :

+ Truy xuất đến form theo chỉ số : document.form[ chỉ số]

+ Truy xuất đến form theo tên : document.form[‘tênform’]

+ Truy xuất đến form theo cú pháp ngắn gọn : document.tênform

+ Ví dụ :

<form name="timkiem">

<input name="tukhoa" value="Từ khóa" />

</form>

<form name="dangnhap">

<input name="un" value="teonv" />

<input name="pw" value="123" />

</form>

<script>

console.log(document.timkiem.tukhoa.value); //Từ khóa

console.log(document.forms[1].un.value); //teonv

console.log(document.forms['dangnhap'].pw.value); //123

</script>

### 4.3.2.Truy cập đến các control trong form để lấy value

+ Dùng hàm getElementById : documnet.getElementIdById(‘idcủaControl’).value;

+ Dùng hàm getElementsByName : documnet.getElementsByName(‘namecủaControl’)[chỉ số].value;

+ Truy cập thông qua form :

Document.tênForm.tênControl.value;

+ Ví dụ :

<form name="tk">

<input type="text" name="tukhoa" id="kw" value="Từ khóa">

<button type="submit">Tìm kiếm</button>

</form> <hr>

<script>

t = document.tk.tukhoa.value; //Từ khóa

t = document.forms[0].tukhoa.value; //Từ khóa

t = document.getElementById("kw").value; //Từ khóa

</script>

### 4.3.3. Kiểm tra dữ liệu trong Form với JavaScript

- Là kiểm tra xem thông tin có được điền vào form đúng yêu cầu hay chưa. Ví dụ tuổi hay số lượng sản phẩm phải là kiểu số nguyên, ngàysinh được được là chữ, mã sinh viên phải dài 7 ký tự …. Nhờ kiểm tra và báo lỗi, user được thông báo và nhập lại cho đúng, khi đó thông tin được xử lý và lưu trữ mới mới có giá trị

- Thực hiện kiểm tra dữ liệu trong form với javascript : Hàm gọi trong sự kiện onSubmit của tag form để kiểm tra dữ liệu nếu không hợp lệ thì trả về false ,còn nếu hợp lệ thì trả về true.

- Ví dụ :

<form name="frmlogin" action="xuly.php" **onsubmit="return kiemtra()"**>

<div>

<label>Username</label> <input type="text" name="u" id="un">

</div>

<div>

<label>Password</label> <input type="password" name="p" id="pw">

</div>

<div>

<button type="submit">Đăng nhập</button>

</div>

<h4 id="baoloi"></h4>

</form>

***\*DEMO KIỂM THỬ :***

<script>

function kiemtra(){

var u = document.frmlogin.u.value;

var p = document.frmlogin.p.value;

if (u=="") { *//quy tắc 1* document.getElementById("baoloi").innerHTML="Chưa nhập username mà";

**return false;** }

if (p=="") { **//quy tắc 2** document.getElementById("baoloi").innerHTML="Chưa nhập password mà";

**return false;** }

**return true;** }

</script>

# 5. Quy trình phát triển web

## 5.1. Giới thiệu về quy trình phát triển web từ lập kế hoạch đến triển khai trang web.

**\* Quy trình phát triển web :**

- B1: Thu thập thông tin

Giai đoạn này tập trung tìm hiểu và nghiên cứu về yêu cầu khách hàng để quyết định các bước tiếp theo sẽ thực hiện như thế nào. Nhiệm vụ quan trọng nhất trong bước này là hiểu rõ về mục đích của website, các chức năng website sẽ cung cấp, nhóm người dùng mục tiêu của website và thu thập đầy đủ các yêu cầu từ phía khách hàng, người dùng. Trả lời những câu hỏi trên sẽ giúp chúng ta lựa chọn được phương án tốt nhất để triển khai dự án

- B2:Lập kế hoạch

Ở bước này, lập trình viên sẽ đưa cho khách hàng cái nhìn tổng quan về website bằng các sitemap, wireframe hoặc bảng biểu dựa trên các thông tin đã thu thập được ở bước 1.

- B3: Thiết kế giao diện

Trong giai đoạn thiết kế, trang web sẽ trực quan hơn với các nút, hình ảnh, video,... sẽ được tạo , tất cả thông tin được thu thập trong giai đoạn đầu tiên là rất quan trọng. Các yêu cầu của khách hàng và trải nghiệm người dùng phải luôn được ghi nhớ trong khi thực hiện thiết kế.

- B4 : Viết tài liệu đặc tả SRS

SRS là tài liệu được sử dụng để mô tả chi tiết các yêu cầu chức năng và phi chức năng của hệ thống. Tài liệu này sẽ hỗ trợ đưa ra các tính năng của hệ thống hoặc dùng cho việc đọc hiểu hệ thống của phía khách hàng.

Đây là một tài liệu quan trọng cho đội phát triển (system analyst, business, analyst, code) và kiểm thử (tester). Ngoài ra, dựa vào các yêu cầu trong SRS, ta có thể đánh giá được số lượng scope, thời gian hoàn thành và các chi phí trong quá trình phát triển sản phẩm một cách nhanh chóng và dễ dàng hơn.

- B5: Viết mã nguồn

Bắt tay vào tạo trang web. Trước tiên khi bắt tay vào viết mã nguồn, cần lựa chọn công nghệ, ngôn ngữ lập trình phù hợp cho dự án.

- B6: Kiểm thử , đánh giá và ra mắt sản phẩm

Quá trình kiểm thử là quá trình thường xuyên nhất của một quy trình. Mọi liên kết nên được kiểm tra để đảm bảo rằng không có liên kết nào bị lỗi hay gián đoạn. Nên kiểm tra mọi biểu mẫu, mọi câu lệnh và kiểm tra lỗi chính tả của toàn bộ trang web. Sử dụng các tiêu chuẩn web chung để kiểm tra xem mã nguồn có đảm bảo sự tương thích giữa các trình duyệt hay không.

- B7 : Bảo trì

Điều quan trọng cần lưu ý, hay coi trang web là một dịch vụ hơn là một sản phẩm. Nên đảm bảo mọi thứ đều hoạt động tốt, mọi người đều hài lòng và luôn chuẩn bị sẵn sàng để thực hiện các thay đổi trong trường hợp khác.

## 5.2. Xác định các bước quan trọng như phân tích yêu cầu, thiết kế giao diện, phát triển, kiểm thử, và triển khai.

- Công nghệ web thường tuân theo quy trình phát triển phần mềm chung và các bước quan trọng trong quy trình phát triển công nghệ web là:

+ Phân tích yêu cầu: Đây là giai đoạn đầu tiên và quan trọng nhất trong quy trình phát triển. Tại giai đoạn này, cần hiểu rõ yêu cầu của khách hàng, xác định tính năng và chức năng cần có trong ứng dụng web.

+ Thiết kế giao diện: Sau khi có được yêu cầu của khách hàng cần thiết kế giao diện người dùng (UI/UX) cho ứng dụng web. Điều này bao gồm việc xác định các trang và luồng điều hướng giữa chúng.

+ Phát triển: Ở bước này cần triển khai mã nguồn để xây dựng chức năng của ứng dụng web. Có thể sử dụng các ngôn ngữ lập trình phù hợp như HTML, CSS, JavaScript, PHP, Ruby, Python, Java,… để phát triển ứng dụng.

+ Kiểm thử: Sau khi hoàn thành phát triển, chúng ta cần tiến hành kiểm thử ứng dụng. Kiểm thử bao gồm kiểm tra tính năng, tương thích, hiệu suất và bảo mật của ứng dụng web.

+ Triển khai: Khi kiểm thử thành công, triển khai ứng dụng web lên môi trường hoạt động thực tế. Điều này có thể là máy chủ chạy trên internet hoặc một môi trường đám mây như AWS, Google Cloud...

+ Bảo trì và nâng cấp: Sau khi triển khai, cần duy trì và cải tiến ứng dụng web theo yêu cầu của khách hàng. Bảo trì và nâng cấp bao gồm khắc phục lỗi, bảo mật và cập nhật tính năng mới cho ứng dụng.

- Ngoài ra, quy trình phát triển công nghệ web còn có thể bao gồm các bước khác như thu thập phản hồi từ người dùng, tối ưu hóa hiệu suất và tăng cường bảo mật. Các bước này có thể thay đổi phụ thuộc vào quy mô và phạm vi của dự án.

## 5.3. Hướng dẫn về vai trò và trách nhiệm của từng thành viên trong nhóm, bao gồm quản lý dự án và phát triển.

### 5.3.1. Quản lý dự án

- Trưởng nhóm: Đóng vai trò lãnh đạo, phân phối công việc, quản lý công việc và thiết lập các mốc thời gian quan trọng cho dự án.

- Quản lý dự án: Chịu trách nhiệm chủ động lập kế hoạch, định hướng và giám sát toàn bộ quá trình dự án, bao gồm quản lý nguồn lực, theo dõi tiến độ, và đảm bảo việc hoàn thành dự án đúng theo yêu cầu và thời gian.

- Quản lý chất lượng: Đảm bảo chất lượng của công việc được thực hiện đúng theo yêu cầu khách hàng và tiêu chuẩn đã đề ra trong dự án.

### 5.3.2. Phát triển

- Kiến trúc sư: Thiết kế, phân tích và xác định kiến trúc và cấu trúc tổ chức cho dự án. Kiến trúc sư tạo ra các sơ đồ, bản thiết kế và hướng dẫn cho các thành viên khác để phát triển sản phẩm, viết đặc tả, lựa chọn công nghệ, ngôn ngữ lập trình phù hợp cho dự án.

- Nhà phát triển: Làm Bố cục trang web - một bản phác họa đồ họa hoặc một thiết kế đồ họa thực tế. Chức năng chính của bố cục là thể hiện cấu trúc thông tin, trực quan hóa nội dung và các chức năng cơ bản. Nhà phát triển là người thực hiện trực tiếp công việc lập trình và đảm bảo mã nguồn được viết theo tiêu chuẩn và đúng mục tiêu của dự án.

- Kiểm thử viên: Kiểm tra mọi biểu mẫu, mọi câu lệnh và kiểm tra lỗi chính tả của toàn bộ trang web. Sử dụng các tiêu chuẩn web chung để kiểm tra xem mã nguồn có đảm bảo sự tương thích giữa các trình duyệt hay không. Sau khi đã kiểm thử xong, ta sẽ triển khai trang web lên máy chủ (server). Sau khi triển khai xong mã nguồn lên máy chủ, chúng ta nên chạy thêm một bài kiểm tra cuối cùng để đảm bảo rằng các mã nguồn đã được cài đặt chính xác.

# TỔNG KẾT

# TÀI LIỆU THAM KHẢO

- Link tài liệu: <https://timoday.edu.vn/bai-3-css-selectors/>

- Link tài liệu: [Tổng hợp các thuộc tính của CSS | Tìm ở đây (timoday.edu.vn)](https://timoday.edu.vn/tong-hop-cac-thuoc-tinh-cua-css/#text)

**-** Link tài liệu: [Javascript là gì? Cú pháp Javascript cơ bản - VNTALKING](https://vntalking.com/cu-phap-javascript-co-ban-nhung-buoc-di-dau-tien.html#javascript-la-gi)

- Link tài liệu: Tài liệu tham khảo: https://www.w3schools.com/

- Link : <https://www.codecademy.com/catalog>

- Tài liệu tham khảo: A Complete Guide to Flexbox (https://csstricks.com/snippets/css/a-guide-to-flexbox/), A Complete Guide to Grid