**Nhiệm vụ 1: Giới thiệu về HTML**

Giới thiệu về HTML: HTML là viết tắt của cụm từ Hypertext Markup Language (tạm dịch là Ngôn ngữ đánh dấu siêu văn bản). HTML 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… và HTML không phải là ngôn ngữ lập trình.

**I, Hướng dẫn tạo một tệp HTML cơ bản:**

Để tạo một tệp HTML cơ bản, bạn cần một trình soạn thảo văn bản như Notepad (trên Windows), Visual Studio Code, Sublime Text, hoặc bất kỳ trình soạn thảo nào bạn ưa thích. Sau đó, làm theo các bước sau:

1. **Mở trình soạn thảo văn bản:** Mở trình soạn thảo văn bản trên máy tính của bạn.
2. **Tạo tệp mới:** Bấm vào "File" và chọn "New" để tạo tệp mới.
3. **Lưu tệp với phần mở rộng .html:** Bấm vào "File" sau đó chọn "Save As". Trong hộp thoại lưu tệp, bạn cần chỉ định tên cho tệp của bạn và đảm bảo rằng nó có phần mở rộng là .html. Ví dụ: index.html.

**II, Hướng dẫn về cú pháp và các phần tử HTML cơ bản**

**1, Cú pháp cho HTML :**

**<!DOCTYPE html>**

<html>

**<head>**

<title>Howkteam.com</title>

**</head>**

**<body>**

**</body>**

</html>

**2, Các phần tử HTML cơ bản:**

##### Thẻ tiêu đề

**<h1>** Nội dung tiêu đề **</h1>**

##### Thẻ định dạng đoạn văn bản

**<p>** Nội dung đoạn văn bản chứa bên trong **</p>**

##### Thẻ hình ảnh & liên kết

Thẻ **<img>** thẻ chứa nội dung là file ảnh có các thuộc tính:

* src=”” (dùng làm đường dẫn/địa chỉ của ảnh )
* alt=”” (văn bản thay thế hiển thị khi ảnh không tải lên được trên trang web )

**<img src**=”đường dẫn hình ảnh” **alt**=”nội dung thay thế”**>**

Thẻ**<a></a>** dẫn đến các trang web khác.

**<a href**=”đường dẫn/ liên kết”**>** nội dung gắn đường dẫn **</a>**

Thẻ **<form></form>** là nơi chứa các thẻ như **input**, **textarea**, **select** để làm mẫu điền các thông tin trong trang web

Thẻ **<input></input>** là chỗ điền thông tin theo dạng chữ, hoặc dạng email, chúng ta sẽ nói kĩ về input ở các bài sau

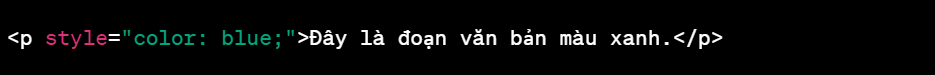
Thẻ **<textarea></textarea>**điền thông tin dạng văn bản

**Nhiệm vụ 2: Cơ bản về CSS**

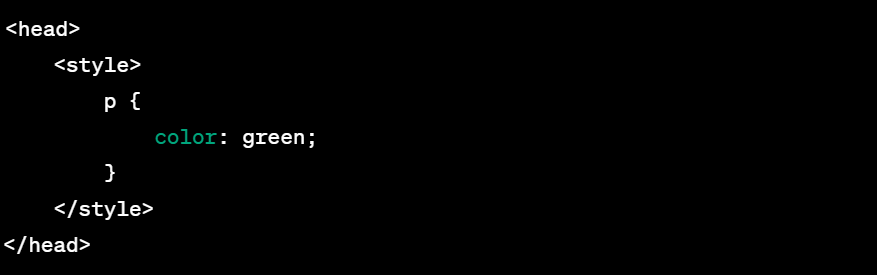
**I, Thêm kiểu cho trang web bằng CSS:**

Để thêm kiểu cho trang web bằng CSS, bạn có thể sử dụng một trong các cách sau:

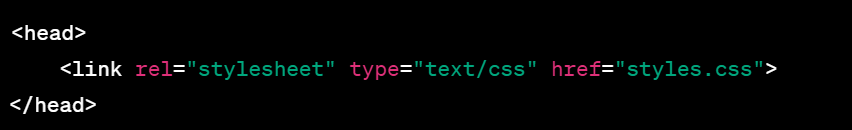
**Inline CSS**: Đặt thuộc tính **style** trực tiếp trong các phần tử HTML. Ví dụ:

****

**Internal CSS**: Đặt các quy tắc CSS trong một phần tử **<style>** bên trong phần **<head>** của tài liệu HTML. Ví dụ:

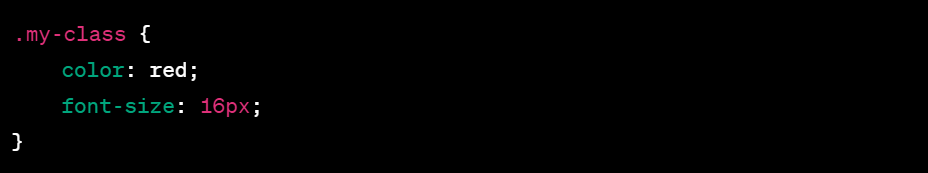


**External CSS**: Tạo một tệp CSS riêng biệt và sau đó liên kết nó với tài liệu HTML bằng thẻ **<link>** trong phần **<head>.** Ví dụ:

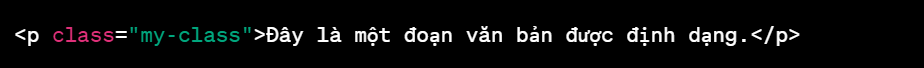


**II. Tạo lớp và áp dụng kiểu bằng cách sử dụng lựa chọn:**

**Tạo lớp CSS**: Định nghĩa lớp CSS trong tệp CSS bằng cách sử dụng dấu chấm (.) trước tên lớp. Ví dụ:

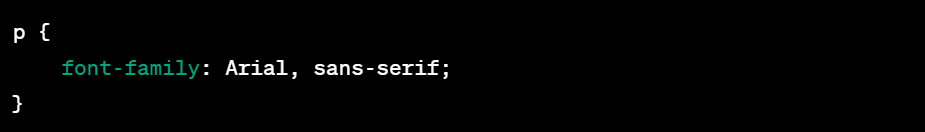


**Áp dụng lớp CSS**: Trong HTML, bạn sử dụng thuộc tính **class** để áp dụng lớp CSS cho một hoặc nhiều phần tử. Ví dụ:

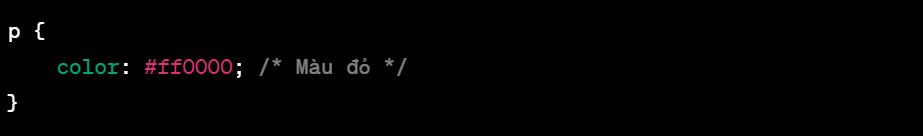


**III. Thử nghiệm với các thuộc tính CSS cơ bản:**

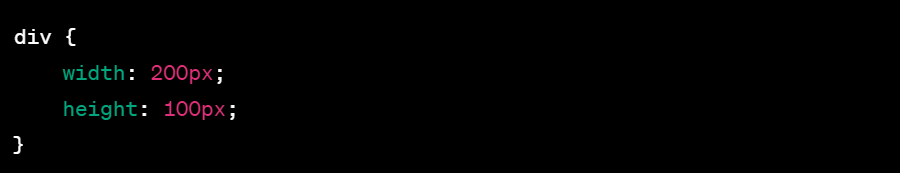
**font-family:** Định rõ kiểu font chữ. Ví dụ:



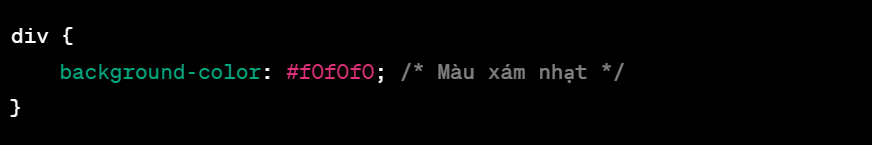
**color:** Định màu sắc cho văn bản. Ví dụ:



**width** và **height**: Đặt chiều rộng và chiều cao của phần tử. Ví dụ:



**background-color:** Đặt màu nền cho phần tử. Ví dụ:



**Nhiệm vụ 3: JavaScrip Cơ bản**

**I. Giới thiệu cú pháp JavaScript cơ bản:**

JavaScript sử dụng các dòng mã bên trong thẻ **<script>** trong trang HTML hoặc được lưu trong các tệp JavaScript có phần mở rộng **.js**.

Một lệnh JavaScript kết thúc bằng dấu chấm phẩy **;**.

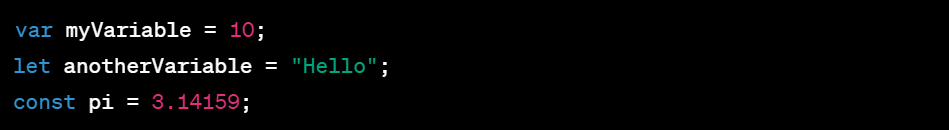
Một số biến và tên hàm có phân biệt chữ hoa và chữ thường (ví dụ: **myVariable** và **myFunction**).

Sử dụng các cặp dấu ngoặc đơn **' '** hoặc dấu ngoặc kép **" "** để đặt văn bản (chuỗi) trong JavaScript.

**II. Tạo biến, hàm và sử dụng sự kiện:**

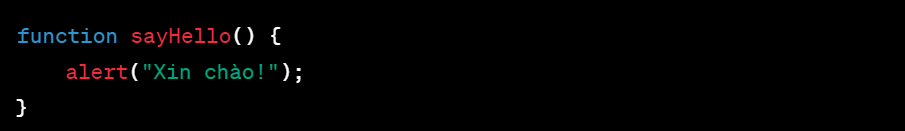
**a. Tạo biến:**

Biến trong JavaScript được tạo bằng từ khóa **var**, **let**, hoặc **const**. Ví dụ:



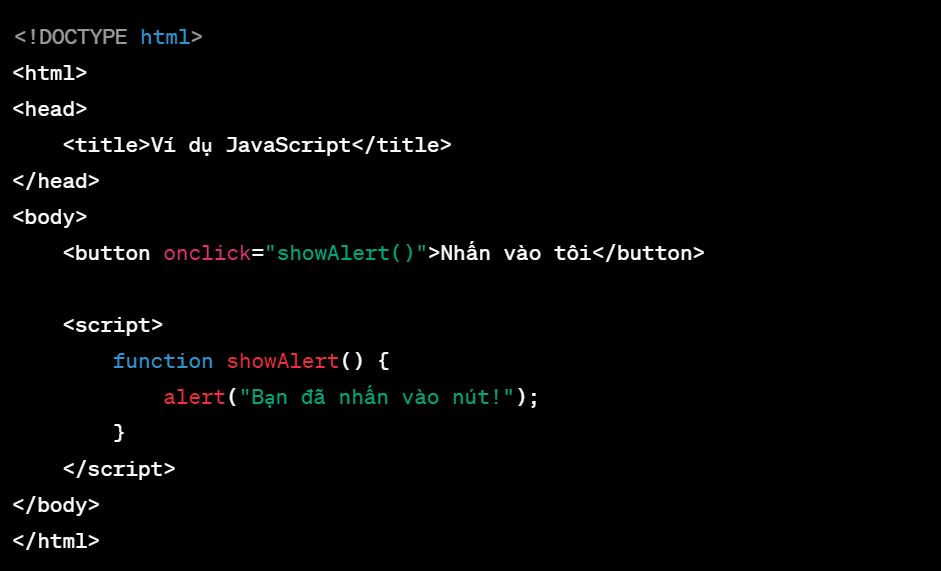
**b. Tạo hàm:**

Hàm trong JavaScript được định nghĩa bằng từ khóa **function**. Ví dụ:



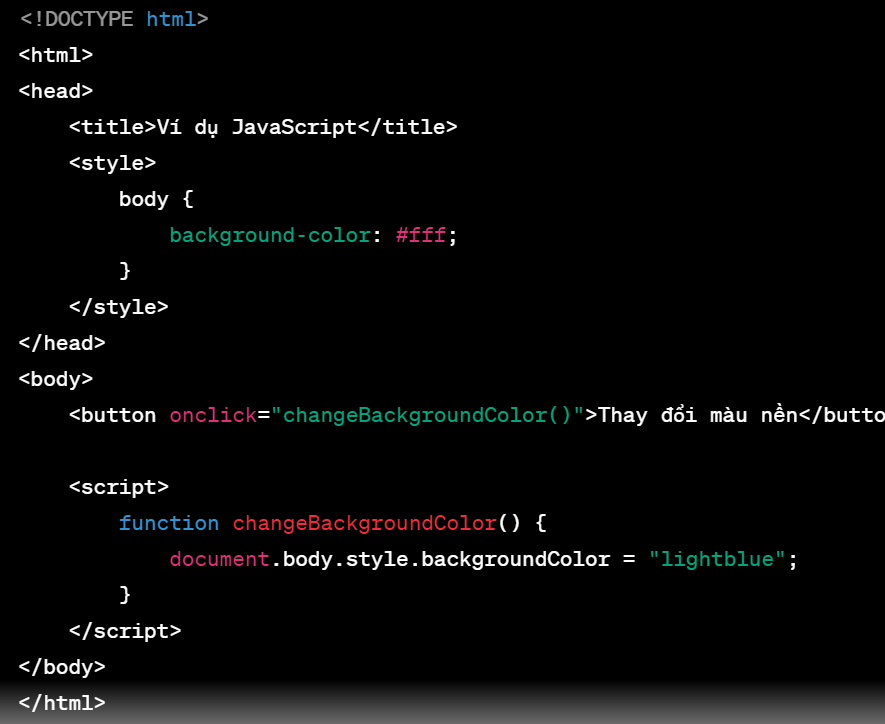
**c. Sử dụng sự kiện:**

Sự kiện là các tương tác của người dùng trên trang web, ví dụ như nhấn nút hoặc di chuột. Ví dụ dưới đây hiển thị một thông báo khi nút được nhấn:



**III. Ví dụ đơn giản:**

Dưới đây là một ví dụ đơn giản: một trang web có một nút, và khi bạn nhấn vào nút, nó sẽ thay đổi màu nền của trang.



**Nhiệm vụ 4: HTML Forms và Input Elements**

**I. Tạo biểu mẫu HTML đơn giản:**

Dưới đây là một ví dụ về biểu mẫu HTML đơn giản:

<!DOCTYPE html>

<html>

<head>

<title>Biểu mẫu đơn giản</title>

</head>

<body>

<h1>Biểu mẫu đơn giản</h1>

<form id="myForm">

<label for="name">Họ và tên:</label>

<input type="text" id="name" name="name" required><br>

<label for="email">Email:</label>

<input type="email" id="email" name="email" required><br>

<label for="gender">Giới tính:</label>

<select id="gender" name="gender">

<option value="male">Nam</option>

<option value="female">Nữ</option>

</select><br>

<input type="submit" value="Gửi">

</form>

</body>

</html>

II**. Tạo kiểu cho biểu mẫu bằng CSS:**

ví dụ đơn giản sử dụng CSS để tạo kiểu cho biểu mẫu

/\* CSS \*/

form {

max-width: 400px;

margin: 0 auto;

padding: 20px;

border: 1px solid #ccc;

border-radius: 5px;

}

label {

display: block;

margin-bottom: 10px;

}

input[type="text"],

input[type="email"],

select {

width: 100%;

padding: 10px;

margin-bottom: 15px;

border: 1px solid #ccc;

border-radius: 5px;

}

input[type="submit"] {

background-color: #007bff;

color: #fff;

padding: 10px 20px;

border: none;

border-radius: 5px;

cursor: pointer;

}

input[type="submit"]:hover {

background-color: #0056b3;

}

**III. Sử dụng JavaScript để xử lý dữ liệu từ biểu mẫu:**

Dưới đây là một ví dụ cụ thể về cách sử dụng JavaScript để kiểm tra dữ liệu hợp lệ trong một biểu mẫu. Trong ví dụ này, chúng ta sẽ kiểm tra xem một trường email đã nhập đúng định dạng hay không.

<!DOCTYPE html>

<html>

<head>

<title>Biểu mẫu kiểm tra email</title>

</head>

<body>

<h1>Biểu mẫu kiểm tra email</h1>

<form id="myForm">

<label for="email">Email:</label>

<input type="email" id="email" name="email" required><br>

<span id="emailError" style="color: red;"></span><br>

<input type="submit" value="Gửi">

</form>

<script>

document.getElementById("myForm").addEventListener("submit", function(event) {

var emailField = document.getElementById("email");

var emailError = document.getElementById("emailError");

var emailPattern = /^[a-zA-Z0-9.\_-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$/;

if (!emailPattern.test(emailField.value)) {

emailError.textContent = "Email không hợp lệ.";

event.preventDefault(); // Ngăn form được gửi nếu email không hợp lệ

} else {

emailError.textContent = ""; // Xóa thông báo lỗi nếu email hợp lệ

}

});

</script>

</body>

</html>

**Nhiệm vụ 5: Quy trình phát triển web:**

***1. Lập kế hoạch (Planning):***

* **Mục tiêu dự án**: Xác định mục tiêu và mục đích của trang web hoặc ứng dụng web.
* **Nghiên cứu thị trường:** Điều tra thị trường và người dùng để hiểu nhu cầu và yêu cầu của họ.
* **Xác định yêu cầu:** Thu thập thông tin từ khách hàng hoặc khách hàng tiềm năng để xác định yêu cầu cụ thể của dự án.

***2. Phân tích yêu cầu (Requirement Analysis):***

* **Phân tích yêu cầu:** Xác định các tính năng và chức năng cần thiết cho trang web.
* **Lập kế hoạch dự án**: Xác định tài liệu dự án, thời gian và ngân sách cần thiết.

***3. Thiết kế (Design):***

* **Thiết kế giao diện (UI/UX Design):** Tạo ra giao diện người dùng dựa trên yêu cầu và nghiên cứu thị trường. Điều này bao gồm thiết kế trang web, trải nghiệm người dùng, và cách trang web hoạt động.
* **Thiết kế cơ sở dữ liệu:** Xác định cấu trúc dữ liệu và cách dữ liệu sẽ được lưu trữ và quản lý.

***4. Phát triển (Development):***

* **Phát triển trang web:** Xây dựng trang web hoặc ứng dụng web bằng cách sử dụng các ngôn ngữ lập trình như HTML, CSS, JavaScript, và các framework phía máy chủ nếu cần.
* **Phát triển cơ sở dữ liệu:** Tạo và quản lý cơ sở dữ liệu theo thiết kế cơ sở dữ liệu.

***5. Kiểm thử (Testing):***

* **Kiểm tra tích hợp (Integration Testing):** Kiểm tra tích hợp giữa các thành phần của trang web để đảm bảo chúng hoạt động cùng nhau.
* **Kiểm thử giao diện (UI Testing):** Kiểm tra giao diện người dùng để đảm bảo tính nhất quán và đáp ứng.
* **Kiểm thử chức năng (Functional Testing):** Kiểm tra các tính năng và chức năng để đảm bảo chúng hoạt động theo đúng yêu cầu.

***6. Triển khai (Deployment):***

* **Triển khai trực tiếp (Live Deployment):** Đưa trang web hoặc ứng dụng web vào hoạt động trên máy chủ và địa chỉ truy cập thực tế.
* **Quản lý phiên bản (Version Management):** Quản lý phiên bản của ứng dụng để theo dõi các thay đổi và dự án cơ bản.

***7. Quản lý dự án (Project Management):***

* **Quản lý dự án:** Điều hành và quản lý toàn bộ dự án, bao gồm việc xác định nguồn lực, lên lịch làm việc, và theo dõi tiến độ.
* **Nhóm phát triển:** Các thành viên trong nhóm phát triển gồm các lập trình viên, thiết kế viên giao diện, chuyên gia cơ sở dữ liệu, và các vai trò khác.
* **Kiểm tra và duyệt (Quality Assurance):** Thực hiện kiểm tra chất lượng để đảm bảo trang web hoặc ứng dụng web hoạt động đúng cách.
* **Hỗ trợ và bảo trì:** Cung cấp hỗ trợ sau triển khai và bảo trì trang web hoặc ứng dụng web.