TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI

KHOA CÔNG NGHỆ THÔNG TIN

Logo

Description automatically generated

**BÀI TẬP LỚN**

**MÔN: CÔNG NGHỆ WEB**

**NHÓM 3: XÂY DỰNG TRANG WEB BLOG CÁ NHÂN**

|  |  |
| --- | --- |
| Giảng viên hướng dẫn: | Nguyễn Thị Hạnh |
| Thành viên nhóm: | Dương Hữu Quý - 2055010222  Trịnh Minh Quang - 2055010299 |

Hà Nội, 9/2023

1. **Giới thiệu về HTML**
   1. **Khái niệm**

* **HTML** là ngôn ngữ đánh dấu siêu văn bản – viết tắt của **Hypertext Markup Language**. Nó giúp người dùng tạo và cấu trúc các thành phần trong trang web hoặc ứng dụng, phân chia các đoạn văn, heading (tiêu đề), links (liên kết), … và được hiển thị trên một chương trình đặc biệt ta gọi là Browser.
* Hiện nay có khá nhiều Browser như Firefox, Chrome, Cốc Cốc, …Tất cả đều có điểm chung là giúp người dùng có thể thao tác với website và nó đều có khả năng biên dịch những đoạn mã HTML, CSS và JavaScript.
  1. **Ưu điểm và nhược điểm**
* **Ưu điểm:**
* Dễ học và sử dụng: HTML là một ngôn ngữ đánh dấu đơn giản và dễ hiểu, nên nó thích hợp cho cả người mới học và các nhà phát triển kỳ cựu.
* Khả năng tích hợp: HTML có thể tích hợp dễ dàng với các ngôn ngữ lập trình khác như JavaScript và CSS để tạo ra các trang web động và tương tác.
* Hỗ trợ rộng rãi: Mọi trình duyệt web hiện đại đều hỗ trợ HTML, giúp đảm bảo tính tương thích trên nhiều nền tảng.
* SEO tốt: HTML cung cấp cơ hội tốt để tối ưu hóa trang web cho công cụ tìm kiếm (SEO), giúp trang web dễ dàng được tìm thấy trên các công cụ tìm kiếm.
* **Nhược điểm:**
* Hạn chế trong việc thiết kế giao diện: HTML tập trung vào cấu trúc nội dung và không thể tạo ra các giao diện phức tạp mà CSS phải được sử dụng để làm điều đó.
* Không thể tạo được tương tác: HTML không thể tạo ra các chức năng tương tác trên trang web, ví dụ như biểu đồ động hoặc các ứng dụng web phức tạp mà cần JavaScript để làm.
  1. **Hướng dẫn tạo một tệp HTML cơ bản**
* **Bước 1:** Mở trình soạn thảo văn bản: Có thể sử dụng bất kỳ trình soạn thảo văn bản nào yêu thích và phù hợp với dòng máy tính mà mình sử dụng. Ở đây nhóm mình sẽ sử dụng Visual Studio Code.
* **Bước 2:** Tạo file HTML mới: Tạo một tệp mới và lưu nó với phần mở rộng “.html”. Tệp này dùng để viết những nội dung người dùng mong muốn đưa lên website. VD: 
* **Bước 3:** Bắt đầu viết mã HTML: Viết những nội dung cần việt và tệp mở rộng .html vừa tạo. VD:

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

<title>The First</title>

</head>

<body>

    <h1>TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI</h1>

    <p>Đây là nhóm 3, lớp 20CN5</p>

</body>

</html>

Đoạn mã trên là cấu trúc 1 tệp HTML cơ bản, bao gồm các thẻ: meta, title, h1, p

* **Bước 4**: Sau khi viết xong đoạn mã theo ý muốn, lưu tệp HTML vào vị trí phù hợp của máy tính. Sử dụng cú pháp Ctrl + S để lưu lại tệp.
* **Bước 5:** Mở tệp HTML trong trình duyệt:
* Cách 1: double kích vào tệp HTML vừa lưu hoặc kéo và thả tệp vào của sổ trình duyệt
* Cách 2: Trên trình Visual Studio Code truy cập vào Extentions (phím tắt: Ctrl + Shift + X) 🡪 Tải Live Server 🡪 Sau khi tải xong, phía góc dưới màn hình bên phải sẽ hiện nút “Go Live”. Khi muốn mở tệp chạy chương trình trên web ta sẽ kích vào nút đó, chương trình sẽ được chạy.
  1. **Tạo trang web với tiêu đề, đoạn văn bản, hình ảnh và siêu liên kết**
* Nội dung trong tệp HTML:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Nhóm\_3</title>  <!-- tiêu đề -->

</head>

<body>

    <h1>TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI</h1>

    <p>KHOA CÔNG NGHỆ THÔNG TIN</p> <!-- Đoạn văn bản -->

    <img src="https://cdn.haitrieu.com/wp-content/uploads/2022/01/Logo-DH-Kien-Truc-Ha-Noi-HAU-BL.png" alt="" style="width: 250px;"> <!-- hình ảnh -->

    <div>

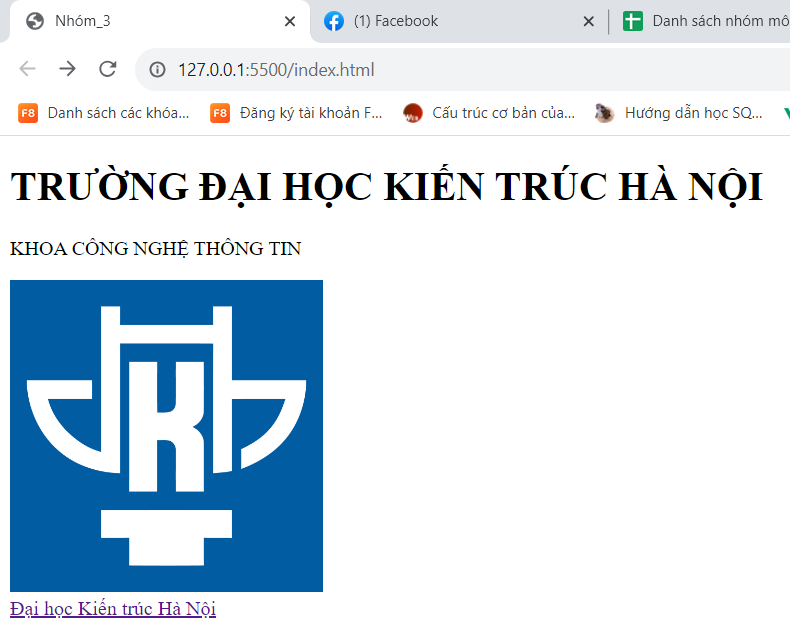
        <a href="https://www.facebook.com/DHKIENTRUCHN">Đại học Kiến trúc Hà Nội</a> <!-- siêu liên kết -->

    </div>

</body>

</html>

* Kết quả:



* 1. Hướng dẫn về cú pháp và các phần tử HTML cơ bản

|  |  |
| --- | --- |
| p | Thẻ đoạn văn |
| h1 | Thẻ tiêu đề |
| u | Thẻ gạch chân |
| i | Thẻ nghiêng |
| b | Thẻ in đậm |
| center | Căn nội dung ra giữa giao diện |
| br | Thẻ ngắt dòng |
| hr | Kẻ dòng |
| img  src  alt | Thẻ hình ảnh  link ảnh  mô tả hình ảnh khi link bị lỗi |
| ul  li | Danh sách không quan tâm đến thứ tự |
| ol  li | Danh sách quan tâm đến thứ tự |
| input  type  radio | Nếu để cùng tên thì sẽ chỉ chọn được 1 |
| textaria | Ô trống người dùng có thể điền nội dung text vào đó |
| form | Quy định phạm vi để reset hoặc thao tác với các ô input |
| caption | Văn bản mô tả mục đích của bảng |
| table  tr  td | Bảng  Dòng  Cột  HTML Table: Cách tạo bảng trong HTML |
| th | Tiêu đề của bảng |
| colspan - rowspan | Gộp 2 hàng và 2 cột với nhau |
| border | Viền của bảng |
| placeholder | Dùng để hiển thị nội dung ẩn trên text |

* Lưu ý: Khi chúng ta tạo một cái danh sách không có thứ tự thì chỉ mục của các danh mục bên trong danh sách sẽ có kiểu mặc định là dấu chấm tròn màu đen. Nếu muốn thiết lập lại kiểu chỉ mục cho các danh mục thì chúng ta cần phải sử dụng thuộc tính **type**.
* Dưới đây là những giá trị có thể áp dụng cho thuộc tính **type:**

|  |  |
| --- | --- |
| disc | - Chỉ mục sẽ có kiểu là một dấu chấm tròn màu đen (mặc định) |
| circle | - Chỉ mục sẽ có kiểu là một dấu chấm tròn màu trắng viền đen. |
| square | - Chỉ mục sẽ có kiểu là một ô vuông màu đen. |
| none | - Không hiển thị chỉ mục. |

1. **Cơ bản về CSS**
   1. **Khái niệm**

**CSS** là ngôn ngữ tạo phong cách cho trang web – viết tắt của **Cascading Style Sheet** language. Nó dùng để tạo phong cách và định kiểu cho những yếu tố được viết dưới dạng ngôn ngữ đánh dấu, như là HTML.

* 1. **Ưu điểm và nhược điểm**
* **Ưu điểm:**
* Tách biệt cấu trúc và giao diện: CSS cho phép bạn tách biệt phần cấu trúc nội dung (HTML) và giao diện, giúp dễ dàng duy trì và cải thiện trang web.
* Tùy chỉnh linh hoạt: CSS cho phép bạn tùy chỉnh màu sắc, font chữ, khoảng cách và nhiều tính năng thiết kế khác một cách linh hoạt.
* Responsive design: CSS cho phép bạn tạo ra thiết kế phản hồi, tức là trang web có thể thích nghi với các kích thước màn hình khác nhau.
* Tiết kiệm thời gian: Sử dụng CSS, bạn có thể áp dụng các kiểu mẫu chung cho nhiều phần tử trên trang web, giúp tiết kiệm thời gian và công sức.
* **Nhược điểm:**
* Học và sử dụng khó khăn hơn so với HTML: CSS có một học cấu trúc phức tạp hơn và có thể gặp khó khăn khi cần xử lý các hiệu ứng phức tạp.
* Tương thích trình duyệt: Một số tính năng CSS có thể không hoạt động một cách đồng nhất trên tất cả các trình duyệt web, do đó cần kiểm tra và điều chỉnh cho từng trình duyệt khác nhau.
  1. **Hướng dẫn cách thêm kiểu cho trang web bằng CSS**
* **Bước 1:** Tạo file CSS mới: Tạo một tệp mới và lưu nó với phần mở rộng “.css”. Tệp này dùng để tạo kiểu hay chỉnh sửa những nội dung người dùng mong muốn đưa lên website. VD: 
* **Bước 2:** Trong file HTML cùng cấp với file CSS, ta sử dụng thẻ “link” để liên kết file HTML với file CSS. Để có thể liên kết được ta điền cú pháp “./ + tên\_file .css” vào thẻ link. VD:

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

Lưu ý: cú pháp “./ + tên\_file .css” chỉ được sử dụng khi file HTML và CSS cùng cấp với nhau.

* **Bước 3:** Sau khi thực hiện xong 2 bước trên thì ta có thể tạo kiểu tùy ý trong file .css để có thể hoàn thiện được mong muốn.
  1. **Tạo các lớp và áp dụng kiểu bằng cách sử dụng lựa chọn**

|  |  |
| --- | --- |
| id  #ten\_id | * Dùng để gọi tên các thẻ * Mỗi id chỉ có duy nhất, không được trùng tên |
| class  .ten\_class | * Dùng để gọi tên các thẻ * Có thể trùng tên với nhau |

* Sử dụng ID:

Để tạp một id trong CSS và áp dụng kiểu cho phần tử cụ thể và duy nhất ta sử dụng dấu “#” trước tên id trong CSS. VD:

HTML:

<p id="text\_1">KHOA CÔNG NGHỆ THÔNG TIN</p>

CSS:

#text\_1{

    color: blue;

    font-size: 16px;

}

* Sử dụng lớp (Class):

Để tạo một lớp trong CSS và áp dụng kiểu cho các phần tử sử dụng lớp đó ta sử dụng dấu “.” trước tên class trong CSS.

HTML:

<p class="text\_2">KHOA CÔNG NGHỆ THÔNG TIN</p>

CSS:

.text\_2{

    color: blue;

    font-size: 16px;

}

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

HTML:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Nhóm\_3</title>  <!-- tiêu đề -->

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

</head>

<body>

    <h1 id="text\_1">TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI</h1>

    <p class="text\_2">KHOA CÔNG NGHỆ THÔNG TIN</p> <!-- Đoạn văn bản -->

    <img src="https://cdn.haitrieu.com/wp-content/uploads/2022/01/Logo-DH-Kien-Truc-Ha-Noi-HAU-BL.png" alt=""> <!-- hình ảnh -->

    <div>

        <a href="https://www.facebook.com/DHKIENTRUCHN">Đại học Kiến trúc Hà Nội</a> <!-- siêu liên kết -->

    </div>

</body>

</html>

CSS:

#text\_1{

    color: blue;

}

.text\_2{

    color: red;

    font-size: 20px;

}

img{

    width: 250px;

    height: 250px;

}

1. **JavaScript cơ bản**
   1. **Khái niệm:**

JavaScript là một ngôn ngữ lập trình dựa trên văn bản (scripting language) thường được sử dụng trong phát triển web. Nó là một phần quan trọng của các ứng dụng web hiện đại và cho phép thêm tính năng tương tác vào các trang web.

* 1. **Ưu điểm và nhược điểm**
* **Ưu điểm:**
* Tích hợp dễ dàng: JavaScript được hỗ trợ bởi tất cả các trình duyệt web phổ biến, có thể hoạt động trên nhiều nền tảng không giống nhau, từ Windows, macOS cho đến các hệ điều hành trên mobile nên không cần cài đặt thêm phần mềm nào để chạy mã JavaScript trên trình duyệt của người dùng.
* Tính tương tác: JavaScript cho phép bạn thêm tính năng tương tác vào trang web. Bạn có thể thực hiện các hành động như xử lý biểu mẫu, thay đổi nội dung trang mà không cần tải lại trang.
* Phát triển nhanh chóng: JavaScript là một ngôn ngữ dễ học và nhanh chóng để phát triển. Nó cho phép bạn thực hiện nhiều công việc trong thời gian ngắn và có rất nhiều tài liệu và cộng đồng hỗ trợ.
* Đa năng: JavaScript không chỉ sử dụng cho phía máy chủ mà còn cho phía khách hàng (client-side) và có thể được sử dụng để xây dựng ứng dụng web đầy đủ và đa dạng.
* **Nhược điểm:**
* Bảo mật: JavaScript chạy trên máy khách, điều này có nghĩa là mã JavaScript có thể bị truy cập và sửa đổi bởi người dùng. Điều này có thể tạo ra các vấn đề về bảo mật nếu không kiểm tra và xử lý cẩn thận.
* Phụ thuộc vào trình duyệt: JavaScript có thể hoạt động khác nhau trên các trình duyệt khác nhau, và có thể cần viết mã khác nhau cho từng trình duyệt để đảm bảo tương thích.
* Xử lý lỗi: Quản lý lỗi trong JavaScript có thể khá khó khăn và dễ dẫn đến sự cố nếu không được quản lý cẩn thận.
  1. **Hướng dẫn tạo biến, hàm và giới thiệu cú pháp JavaScript cơ bản**
     1. **Biến**
* Để khai báo biến ta có thể dùng 3 cách sau:

|  |  |
| --- | --- |
| var | Cho phép được khai báo trùng |
| let | Biến được khai báo là duy nhất và không trùng nhau |
| const | Kiểu hằng số, là duy nhất |

VD:

var name = "Quang";

let age = 30;

const PI = 3.14;

* + 1. **Kiểu dữ liệu**
* Có các kiểu dữ liệu cơ bản như: số (number), chuỗi (string), boolean, mảng (array) và đối tượng (object)

VD:

let number = 6;

let greeting = "Hello, world!";

let isTrue = true;

let colors = ["red", "green", "blue"];

let person = {name: "Quang", age: 21};

* + 1. **Toán tử**
* Các toán tử cơ bản:

|  |  |
| --- | --- |
| + | Phép cộng |
| - | Phép trừ |
| \* | Phép nhân |
| / | Phép chia lấy số dư |
| % | Phép chia lấy số nguyên |
| === | So sánh bằng |
| != | So sánh khác |

VD:

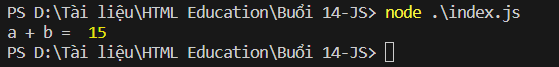
let a = 5;

let b = 10;

let c = a + b;

console.log("a + b = ",c);

Kết quả:



* + 1. **Câu điều kiện**
* Sử dụng if, else if, và else để kiểm tra điều kiện và thực hiện các hành động tương ứng.

VD:

let age = 21;

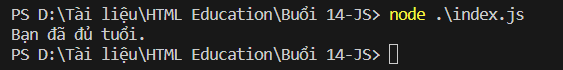
if (age < 18){

    console.log("Bạn chưa đủ tuổi.");}

else {

    console.log("Bạn đã đủ tuổi.");}

Kết quả:



* + 1. **Vòng lặp**
* JavaScript hỗ trợ vòng lặp for, while, và do...while để lặp qua các phần tử hoặc thực hiện mã nhiều lần.

VD:

for (let i = 0; i < 5; i++){

    console.log(i);}

// hoặc

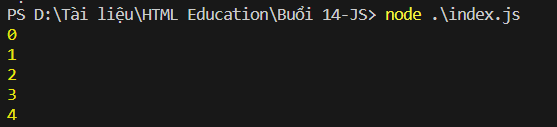
let i = 0;

while (i < 5){

    console.log(i);

    i++;}

Kết quả:



* + 1. **Hàm**
* Hàm (function) là một khối mã được đặt tên và sau đó gọi nó nhiều lần từ nhiều nơi khác nhau trong chương trình của bạn.

VD:

function tongHaiSo(a, b){

    var tong = a + b;

    return tong;

}

  var ketQua = tongHaiSo(5, 3); // Gọi hàm và lưu kết quả vào biến ketQua

  console.log(ketQua); // In ra 8

Kết quả:

****

* + 1. **Sự kiện:**
* Để sử dụng sự kiện ban đầu chúng ta sẽ tạo hàm bên file JavaScript
* Giờ gán vào 1 nút bất kì thì chúng ta có thể gọi lại tên hàm vào trong Attribute Onclick của nút ấy
* Hoặc chúng ra có thể sử dụng element.addEventListener("click", myFunction);
  + 1. **Mảng**
* Mảng (array) là một kiểu dữ liệu được sử dụng để lưu trữ nhiều giá trị khác nhau trong một biến duy nhất
* Một mảng trong JavaScript có thể chứa các loại dữ liệu khác nhau như số nguyên, chuỗi, đối tượng và thậm chí là các mảng khác
* JavaScript có nhiều phương thức để làm việc với mảng, như push, pop, shift, unshift, forEach, và nhiều phương thức khác.

VD:

var numbers = [1, 2, 3, 4, 5];

console.log(numbers[0]); // In ra 1

console.log(numbers[2]); // In ra 3

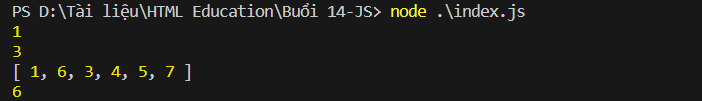
numbers[1] = 6;

numbers.push(7);

console.log(numbers);

console.log(numbers.length); // In ra 6

Kết quả:



* + 1. **Đối tượng**
* Đối tượng trong JavaScript là một tập hợp các thuộc tính và phương thức.

VD:

let person = { name: "Quý", age: 21 };

* 1. **Ví dụ về việc hiển thị thông báo khi nút được nhấn**

HTML:

<!DOCTYPE html>

<html>

<head>

  <title>Ví dụ hiển thị thông báo trong JavaScript</title>

</head>

<body>

  <button id="myButton">Nhấn vào tôi</button>

  <script>

    // Lấy tham chiếu đến nút bằng cách sử dụng ID

    var button = document.getElementById("myButton");

    // Thêm sự kiện "click" cho nút

    button.addEventListener("click", function() {

      alert("Nút đã được nhấn!");

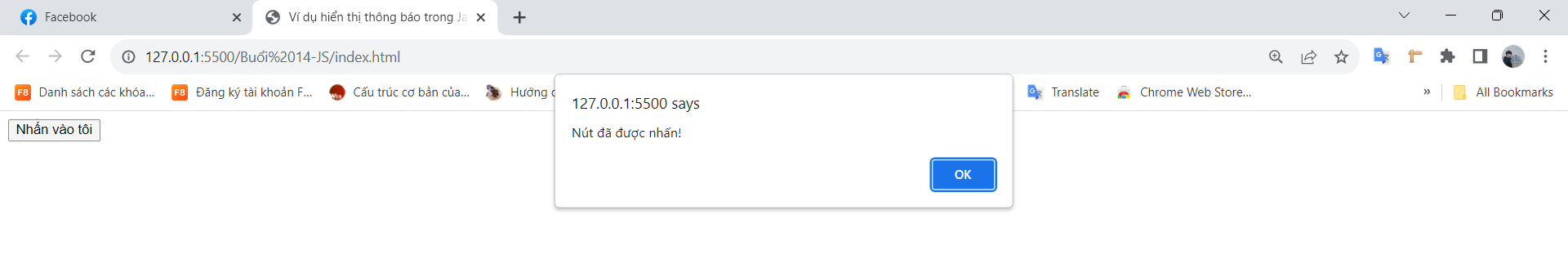
    });

  </script>

</body>

</html>

Kết quả:



1. **HTML Forms và Input Elements** 
   1. **Hướng dẫn tạo biểu mẫu HTML đơn giản với các phần tử nhập liệu như ô văn bản, ô chọn và nút gửi**

Đây là một ví dụ về cách tạo một biểu mẫu HTML đơn giản với các phần tử nhập liệu như ô văn bản, ô chọn và nút gửi:

<!DOCTYPE *html*>

<html>

<head>

    <title>Biểu mẫu đơn giản</title>

</head>

<body>

    <h2>Biểu mẫu đơn giản</h2>

    <form *action*="xuly.php" *method*="POST">

        <!-- Ô văn bản -->

        <label *for*="ten">Tên:</label>

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

        <!-- Ô chọn -->

        <label *for*="gioitinh">Giới tính:</label>

        <select *id*="gioitinh" *name*="gioitinh">

            <option *value*="nam">Nam</option>

            <option *value*="nu">Nữ</option>

        </select><br><br>

        <!-- Nút gửi -->

        <input *type*="submit" *value*="Gửi">

    </form>

</body>

</html>

Giải thích:

* <!DOCTYPE html>: Khai báo kiểu tài liệu HTML5.
* <html>: Thẻ mở của tài liệu HTML.
* <head>: Phần đầu của tài liệu HTML, chứa thông tin về tài liệu.
* <title>: Tiêu đề của trang.
* <body>: Phần thân của tài liệu HTML, chứa nội dung hiển thị trên trình

duyệt.

* <h2>: Tiêu đề của biểu mẫu.
* <form>: Thẻ mở của biểu mẫu. action chỉ định URL mà dữ liệu biểu mẫu sẽ được gửi đến khi người dùng nhấn nút gửi. method chỉ định phương thức gửi dữ liệu, ở đây là "POST".
* <label>: Nhãn cho các phần tử nhập liệu để mô tả chức năng của chúng.
* <input>: Phần tử nhập liệu kiểu ô văn bản.
* type="text": Kiểu là ô văn bản.
* id và name: Định danh của phần tử để tham chiếu trong mã JavaScript hoặc xử lý dữ liệu ở phía máy chủ.
* required: Đánh dấu ô văn bản là bắt buộc nhập.
* <select>: Phần tử chọn.
* id và name: Định danh của phần tử.
* <option>: Các tùy chọn trong ô chọn.
* <input>: Phần tử nhập liệu kiểu nút gửi để gửi biểu mẫu.
* type="submit": Kiểu nút gửi.

Với biểu mẫu này, khi người dùng điền thông tin và nhấn nút "Gửi," dữ liệu sẽ được gửi đến một trang xử lý (ở đây là "xuly.php") qua phương thức POST. Bạn cần tạo một tệp xử lý để lấy và xử lý dữ liệu từ biểu mẫu sau khi nó được gửi đi.

* 1. **Tạo kiểu cho biểu mẫu bằng CSS**

<style>

    /\* Định dạng nền và khoảng cách của trang \*/

    body {

        font-family: Arial, sans-serif;

        background-color: #f2f2f2;

        margin: 0;

        padding: 0;

    }

    /\* Định dạng của biểu mẫu \*/

    form {

        max-width: 400px;

        margin: 20px auto;

        background-color: #ffffff;

        padding: 20px;

        border-radius: 5px;

        box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

    }

    /\* Định dạng của tiêu đề \*/

    h2 {

        text-align: center;

    }

    /\* Định dạng của nhãn và ô nhập liệu \*/

    label {

        display: block;

        margin-bottom: 8px;

    }

    input[*type*="text"], select {

        width: 100%;

        padding: 10px;

        margin-bottom: 15px;

        border: 1px solid #ccc;

        border-radius: 3px;

    }

    /\* Định dạng của nút gửi \*/

    input[*type*="submit"] {

        background-color: #007BFF;

        color: #fff;

        padding: 10px 20px;

        border: none;

        border-radius: 3px;

        cursor: pointer;

    }

    /\* Định dạng của nút gửi khi rê chuột vào \*/

    input[*type*="submit"]*:hover* {

        background-color: #0056b3;

    }

</style>

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

<!DOCTYPE *html*>

<html>

<head>

    <title>Biểu mẫu đơn giản</title>

    <style>

        /\* CSS ở đây \*/

    </style>

</head>

<body>

    <h2>Biểu mẫu đơn giản</h2>

    <form *id*="myForm">

        <!-- Ô văn bản -->

        <label *for*="ten">Tên:</label>

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

        <!-- Ô chọn -->

        <label *for*="gioitinh">Giới tính:</label>

        <select *id*="gioitinh" *name*="gioitinh">

            <option *value*="nam">Nam</option>

            <option *value*="nu">Nữ</option>

        </select><br><br>

        <!-- Nút gửi -->

        <input *type*="submit" *value*="Gửi">

    </form>

    <script>

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

*e*.preventDefault(); // Ngăn chặn biểu mẫu gửi dữ liệu mặc định

            // Lấy giá trị từ các trường nhập liệu

            var ten = document.getElementById("ten").value;

            var gioitinh = document.getElementById("gioitinh").value;

            // Thực hiện xử lý dữ liệu (ở đây chỉ là ví dụ)

            alert("Dữ liệu đã được gửi:\nTên: " + ten + "\nGiới tính: " + gioitinh);

            // Sau khi xử lý xong, bạn có thể thực hiện các thao tác khác, chẳng hạn như gửi dữ liệu đến máy chủ.

            // Xóa dữ liệu trong biểu mẫu hoặc thực hiện các thao tác khác

            document.getElementById("ten").value = "";

            document.getElementById("gioitinh").value = "";

        });

    </script>

</body>

</html>

1. **Quy trình phát triển web**
   1. **Giới thiệu về quy trình phát triển web từ lập kế hoạch đến triển khai trang web**

Quy trình gồm các bước sau đây:

* Lập kế hoạch và phân tích yêu cầu.
* Thiết kế giao diện và cở sở dữ liệu.
* Triển khai dự án.
* Kiểm thử.
* Triển khai và vận hành.
  1. **Xác định các bước quan trọng**
     1. **Lập kế hoạch và phân tích yêu cầu**
* Hiểu rõ và xác định rõ yêu cầu của dự án. Xác định chức năng, giao diện, và hệ thống cần thiết.
  + 1. **Thiết kế giao diện và cơ sở dữ liệu**
* Thiết kế giao diện: Tập trung xây dựng giao diện thân thiện với người dùng.
* Thiết kế cở sở dữ liệu: Thiết kế cơ sở dữ liệu cần thiết, xác định cấu trúc, bảng, quan hệ, và tối ưu hóa.
  + 1. **Triển khai dự án**
* Giao diện: Xây dựng giao diện người dùng sử dụng HTML, CSS, và JavaScript. Tập trung vào việc làm cho giao diện thân thiện và tương tác tốt.
* BackEnd: Xây dựng hệ thống và xử lý dữ liệu bằng các ngôn ngữ và framework như Node.js
  + 1. **Kiểm thử**
* Đảm bảo rằng tất cả các chức năng hoạt động đúng theo yêu cầu.
* Xác định rằng giao diện người dùng hoạt động đúng cách và đẹp mắt trên mọi thiết bị.
* Đảm bảo rằng tương tác của người dùng với ứng dụng diễn ra một cách trơn tru và thân thiện
  1. **Hướng dẫn về vai trò và trách nhiệm của từng thành viên trong nhóm**

1. **Thực hành xây dựng trang web đơn giản**
   1. **Xây dựng trang web đơn giản từ ddaaauf bằng việc kết hợp HTML, CSS và JavaScript**
   2. **Xây dựng 1 website hoàn thiện (Web động có kết nối CSDL)**
2. **Reponsive Web Design**
3. **HTML5 và CSS3**
4. **Thiết kế trang web thân thiện với SEO**
5. **Quản lý phiên bản và Git**