**TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

|  |
| --- |
|  |
|  | C:\Users\Admin\AppData\Local\Temp\ksohtml18948\wps1.png |

**BÁO CÁO MÔN HỌC**

**CÔNG NGHỆ WEB**

Chuyên ngành: Công nghệ thông tin

Giảng viên: Th.S. Nguyễn Thị Hạnh

Sinh viên thực hiện: Nhóm 15

Khuất Đăng Hiếu

Nguyễn Thị Ngọc

Tống Thị Vân

Lớp: 20CN5

Hà Nội, tháng năm 2023

**MỤC LỤC**

[1. Giới thiệu về HTML: 5](#_Toc146358659)

[1.1. Hướng dẫn tạo một tệp HTML cơ bản. 5](#_Toc146358660)

[1.2. Tạo trang web với tiêu đề, đoạn văn bản, hình ảnh và siêu liên kết. 5](#_Toc146358661)

[1.3. Hướng dẫn về cú pháp và các phần tử HTML cơ bản. 7](#_Toc146358662)

[2. Cơ bản về CSS: 8](#_Toc146358663)

[2.1. Hướng dẫn cách thêm kiểu cho trang web bằng CSS. 8](#_Toc146358664)

[2.2. Tạo các lớp và áp dụng kiểu bằng cách sử dụng lựa chọn. 10](#_Toc146358665)

[2.3. Thử nghiệm với các thuộc tính CSS cơ bản như font chữ, màu sắc, độ rộng và chiều cao. 13](#_Toc146358666)

[2.3.1. Font chữ: 13](#_Toc146358667)

[2.3.2. Màu sắc: 13](#_Toc146358668)

[2.3.3. Độ rộng và chiều cao: 13](#_Toc146358669)

[3. JavaScript Cơ bản: 15](#_Toc146358670)

[3.1. Giới thiệu cú pháp JavaScript cơ bản. 15](#_Toc146358671)

[3.2. Hướng dẫn tạo biến, hàm, và sử dụng sự kiện. 15](#_Toc146358672)

[3.3. Tạo các ví dụ đơn giản như hiển thị thông báo khi nút được nhấn 16](#_Toc146358673)

[4. HTML Forms và Input Elements: 17](#_Toc146358674)

[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. 17](#_Toc146358675)

[4.2. Tạo kiểu cho biểu mẫu bằng CSS. 18](#_Toc146358676)

[4.3. Sử dụng JavaScript để xử lý dữ liệu từ biểu mẫu (vd: kiểm tra dữ liệu hợp lệ). 19](#_Toc146358677)

[5. Quy trình phát triển web: 21](#_Toc146358678)

[5.1. Giới thiệu về quy trình phát triển web từ lập kế hoạch đến triển khai trang web. 21](#_Toc146358679)

[5.1.1. Khảo sát và thu thập yêu cầu: 21](#_Toc146358680)

[5.1.2. Lập kế hoạch: 21](#_Toc146358681)

[5.1.3. Thiết kế: 21](#_Toc146358682)

[5.1.4. Phát triển: 21](#_Toc146358683)

[5.1.5.Kiểm thử: 22](#_Toc146358684)

[5.1.6. Triển khai: 22](#_Toc146358685)

[5.1.7. Đánh giá và cải tiến: 22](#_Toc146358686)

[5.2. Xác định các bước quan trọng như phân tích yêu cầu, thiết kế giao diện, phát 22](#_Toc146358687)

[triển, kiểm thử, và triển khai. 22](#_Toc146358688)

[5.2.1. Phân tích yêu cầu: 22](#_Toc146358689)

[5.2.2. Thiết kế giao diện: 22](#_Toc146358690)

[5.2.3. Kiểm thử: 23](#_Toc146358691)

[5.2.4. Triển khai: 23](#_Toc146358692)

[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. 23](#_Toc146358693)

[5.3.1.Quản lý dự án: 24](#_Toc146358694)

[5.3.2. Xây dựng dự án: 24](#_Toc146358695)

**MỤC LỤC HÌNH ẢNH**

1. Giới thiệu về HTML

[Hình 1. 1. Tạo file HTML 4](#_Toc146315149)

[Hình 1. 2. Thực hiện thao tác 4](#_Toc146315150)

[Hình 1. 3. Tiêu đề 5](#_Toc146315151)

[Hình 1. 4. Đoạn văn bản 5](#_Toc146315152)

[Hình 1. 5. Kết quả tạo tiều đề, đoạn văn bản 5](#_Toc146315153)

[Hình 1. 6. Thêm hình ảnh và siêu liên kết 6](#_Toc146315154)

[Hình 1. 7. Hoàn thành tạo trang 6](#_Toc146315155)

1. Cơ bản về CSS

[Hình 2. 1. Tạo tệp CSS 7](#_Toc146315386)

[Hình 2. 2. Liên kết HTML và CSS 8](#_Toc146315387)

[Hình 2. 3. Tạo kiểu 8](#_Toc146315388)

[Hình 2. 4. Chạy HTML liên kết CSS 9](#_Toc146315389)

[Hình 2. 5. Tạo lớp trong CSS 9](#_Toc146315390)

[Hình 2. 6. Ví dụ tạo lớp 10](#_Toc146315391)

[Hình 2. 7. Tạo kiểu dựa trên lớp 10](#_Toc146315392)

[Hình 2. 8. Chạy kết quả 11](#_Toc146315393)

[Hình 2. 9. Tạo kiểu lựa chọn trong HTML 11](#_Toc146315394)

[Hình 2. 10. Chạy chương trình 12](#_Toc146315395)

[Hình 2. 11. Thử nghiệm thuộc tính tạo kiểu CSS 13](#_Toc146315396)

[Hình 2. 12. Chạy chương trình 14](#_Toc146315397)

1. JavaScipt cơ bản

[Hình 3. 1. Tạo biến, hàm và sử dụng sự kiện 16](#_Toc146387143)

[Hình 3. 2. Kết quả 17](#_Toc146387144)

[Hình 3. 3. Thêm Button thao tác trên web 18](#_Toc146387145)

[Hình 3. 4. Các câu lệnh trong JavaScipt 18](#_Toc146387146)

[Hình 3. 5. Kết quả 19](#_Toc146387147)

1. HTML Forms và Input Elements

[Hình 4. 1. Ví dụ thẻ lựa chọn 19](#_Toc146387165)

[Hình 4. 2. Ví dụ HTML 20](#_Toc146387166)

[Hình 4. 3. Tạo kiểu CSS 21](#_Toc146387167)

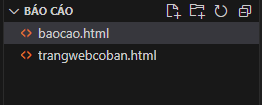
[Hình 4. 4. Chạy chương trình 21](#_Toc146387168)

[Hình 4. 5. Xử lý dữ liệu trong JavaScript 22](#_Toc146387169)

[Hình 4. 6. Chưa điền thông tin hợp lệ 23](#_Toc146387170)

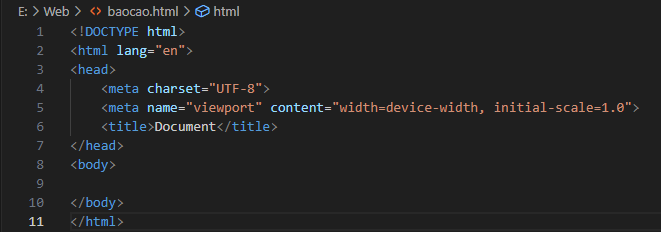
1. **Giới thiệu về HTML:** 
   1. **Hướng dẫn tạo một tệp HTML cơ bản.**

* Bước 1: Mở công cụ lập trình tạo một tệp bằng cách gõ tên file.html.
* Ví dụ tạo file html có tên baocao.html

****

**Hình 1. 1. Tạo file HTML**

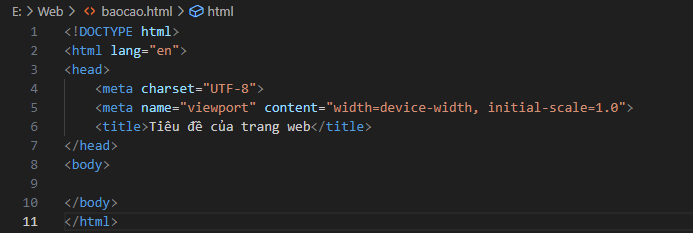
* Bước 2: Sau khi tạo một tệp HTML cơ bản. Bắt đầu tập tin bằng cách gõ ký tự ! tệp sẽ hiện ra các dòng code
* Sau khi tạo thành công một tệp tin HTML cơ bản. Bạn có thể tiếp tục thêm các thẻ và nội dung khác vào tệp tin của bạn để tạo nên trang web đầy đủ.

****

**Hình 1. 2. Thực hiện thao tác**

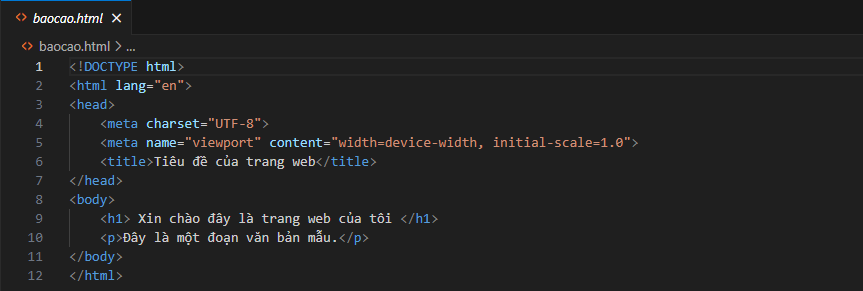
* 1. **Tạo trang web với tiêu đề, đoạn văn bản, hình ảnh và siêu liên kết.**
* Cấu trúc cơ bản của một trang HTML có dạng như sau, thường gồm 3 phần:
* <!DOCTYPE html>: Phần khai báo chuẩn của html hay xhtml.
* <head></head>: Phần khai báo ban đầu, khai báo về [meta](https://hocwebchuan.com/reference/tag/tag_meta.php), [title](https://hocwebchuan.com/reference/tag/tag_title.php), [css](https://hocwebchuan.com/tutorial/tut_css.php), javascript…
* <body></body>: Phần chứa nội dung của trang web, nơi hiển thị nội dung.
* Trong thẻ `<html>`, hai phần chính là `<head>` và `<body>`.
* Phần `<head>` chứa các thông tin đặc tả về tài liệu, như tiêu đề, tập tin CSS, JavaScript, và các thẻ meta khác. Ví dụ:

Ở đây Document = Tiêu đề của trang web

****

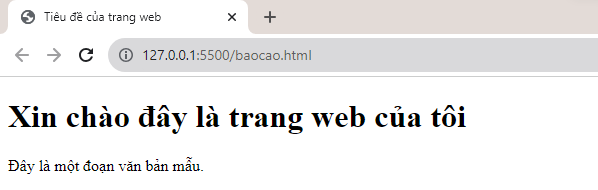
**Hình 1. 3. Tiêu đề**

* Phần `<body>` chứa nội dung thực sự của trang web. Ví dụ:

****

**Hình 1. 4. Đoạn văn bản**

* Sau khi hoàn thành ví dụ ta lưu lại nội dung và chạy chương trình thì được kết quả như hình

****

**Hình 1. 5. Kết quả tạo tiều đề, đoạn văn bản**

* Để tạo hình ảnh ta sử dụng dòng lệnh sau

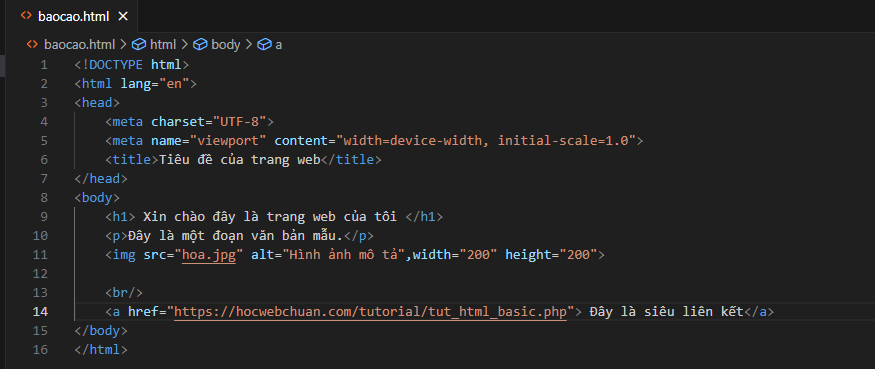
***<img src="duong-dan-cua-hinh-anh.jpg" alt="Hình ảnh mô tả">***

* Sử dụng dòng lệnh sau để tạo siêu liên kết

***<a href="https://www.example.com">Đây là siêu liên kết</a>***

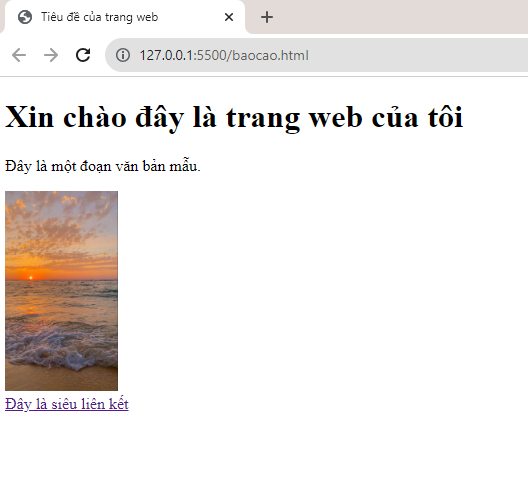
Hãy nhớ thay thế "duong-dan-cua-hinh-anh.jpg" bằng đường dẫn thực sự tới file hình ảnh muốn sử dụng. Ngoài ra, có thể tùy chỉnh và bổ sung thêm nội dung và thẻ HTML khác vào tệp tin để tạo ra trang web phù hợp với yêu cầu.

* Như vậy ta có chương trình hoàn chỉnh theo yêu cầu

****

**Hình 1. 6. Thêm hình ảnh và siêu liên kết**

* Kết quả khi chạy chương trình

****

**Hình 1. 7. Hoàn thành tạo trang**

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

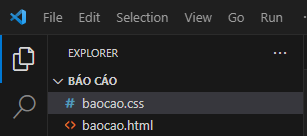
Cú pháp HTML cơ bản:

* Một tệp HTML bắt đầu bằng thẻ `<!DOCTYPE html>` để định dạng tài liệu.
* Một trang HTML được bao bọc bởi thẻ `<html>` và kết thúc bằng thẻ `</html>`.
* Thẻ `<head>` chứa các thông tin mô tả và thuộc tính của trang web.
* Thẻ `<body>` chứa nội dung thực sự của trang web.
* Các phần tử HTML thường được bao bọc bởi cặp thẻ mở `<tag>` và thẻ đóng `</tag>`.
* Một số phần tử có thể tự đóng mà không cần thẻ đóng, như `<br>` hoặc `<img>`.
* Các phần tử HTML cơ bản:
* `<h1>` đến `<h6>`: tạo tiêu đề với mức độ ưu tiên khác nhau.
* `<p>`: tạo đoạn văn bản.
* `<a>`: tạo siêu liên kết đến một trang web hoặc địa chỉ khác.
* `<img>`: chèn hình ảnh vào trang web.
* `<ul>` và `<li>`: tạo danh sách không sắp xếp.
* `<ol>` và `<li>`: tạo danh sách có sắp xếp.
* `<table>`, `<tr>`, `<th>`, `<td>`: tạo bảng.
* `<div>` và `<span>`: Được sử dụng để nhóm và phân loại các phần tử khác.
* `<form>`, `<input>`, `<label>`, `<select>`, `<textarea>`: tạo biểu mẫu và các phần tử nhập liệu.

1. **Cơ bản về CSS:**
   1. **Hướng dẫn cách thêm kiểu cho trang web bằng CSS.**

Để thêm kiểu cho trang web của bạn bằng CSS, bạn có thể làm theo các bước sau:

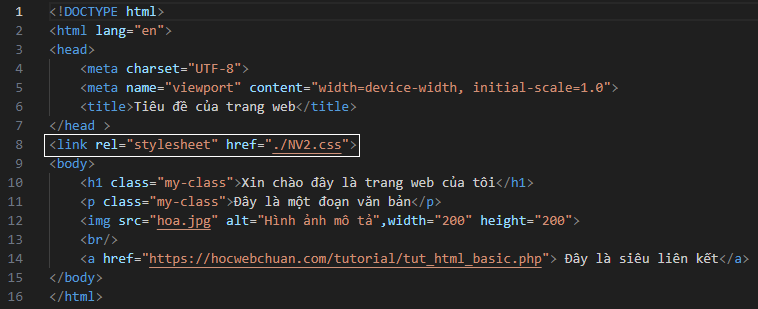
* Bước 1: Tạo một tệp tin CSS
* Để thực hiện cách tốt nhất, tạo một tệp tin CSS riêng và liên kết với tệp HTML bằng cách tạo một file chứa cả 2 tệp tin này. Ví dụ:

****

**Hình 2. 1. Tạo tệp CSS**

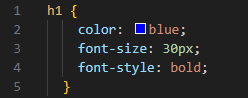
Cách tạo một file chứa cả 2 tệp tin CSS và HTML sẽ nhanh và tiện dụng hơn.

* Bước 2: Liên kết tệp CSS với trang HTML:
* Thêm `<link rel="stylesheet" href="NV2.css">` vào trong thẻ `<head>` của trang HTML như ảnh sau:
* Bước 3: Chọn phần tử HTML
* Sử dụng các thẻ CSS selector để chọn phần tử HTML mà bạn muốn kiểu (ví dụ: `<h1>`, `<p>`, `<div>`, v.v.)
* Định nghĩa các kiểu CSS: Sử dụng các thuộc tính CSS để định nghĩa kiểu cho phần tử đã chọn:  `color`, `font-size`, `background-color`, `margin`, `padding`, v.v.
* Trong tệp tin CSS có thể sử dụng các lựa chọn để chọn các phần tử muốn áp dụng.
* Ví dụ:
* Cách thêm kiểu cho một phần tử h1 trong CSS:
* Tạo tệp CSS: Tạo một tệp có tên là "NV2.css".

****

**Hình 2. 2. Liên kết HTML và CSS**

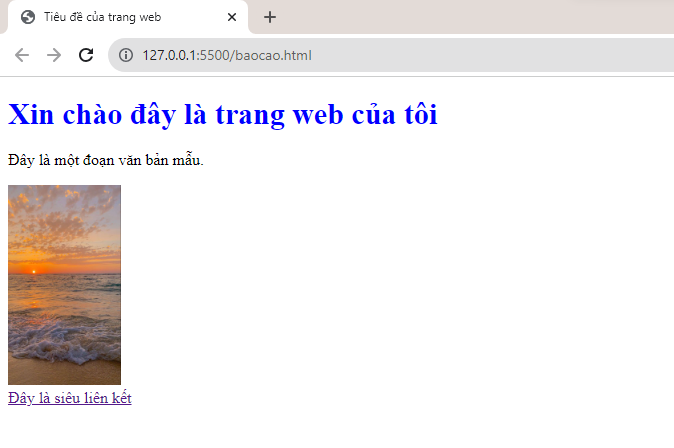
Chọn phần tử HTML muốn kiểu: Sử dụng h1 trong tệp styles.css như sau:

****

**Hình 2. 3. Tạo kiểu**

Trong ví dụ trên, chúng ta đã chọn các phần tử `<h1>`  để áp dụng kiểu.

* Hoàn thành các bước trên ta chạy chương trình

****

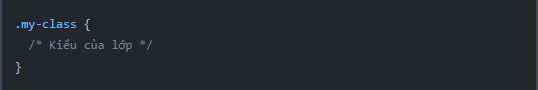
**Hình 2. 4. Chạy HTML liên kết CSS**

* Trong ví dụ trên, chúng ta đã sử dụng thuộc tính `color` để đặt màu chữ cho tiêu đề `<h1>`,
* Các thuộc tính CSS khác như `background`, `margin`, `padding`, `border`, và nhiều thuộc tính khác để tùy chỉnh kiểu cho các phần tử trong trang web.
* Nhớ rằng CSS của bạn có thể trực tiếp được viết trong phần `<style>` của tệp HTML hoặc sau đó được chèn qua tệp CSS riêng.

Tuy nhiên, việc tách riêng CSS ra tệp tin riêng giúp tạo ra sự phân tách giữa kiểu và nội dung, giúp quản lý kiểu được dễ dàng hơn và tăng tính tái sử dụng mã.

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

* Đầu tiên, trong tệp tin CSS riêng hoặc phần `<style>` của tệp tin HTML, tạo một lớp bằng cách sử dụng dấu chấm "." trước tên lớp của bạn. Ví dụ:

****

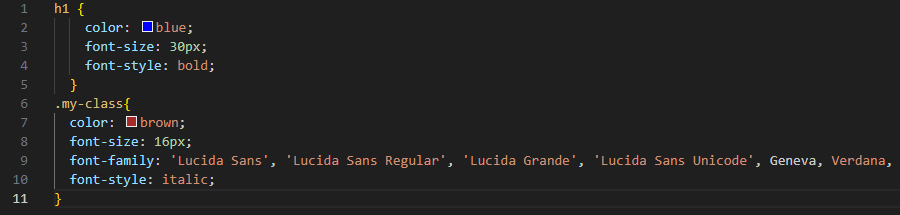
**Hình 2. 5. Tạo lớp trong CSS**

* Sau đó, trong phần tử HTML mà bạn muốn áp dụng kiểu cho, thêm thuộc tính "class" với giá trị là tên lớp mà bạn đã tạo. Ví dụ:

****

**Hình 2. 6. Ví dụ tạo lớp**

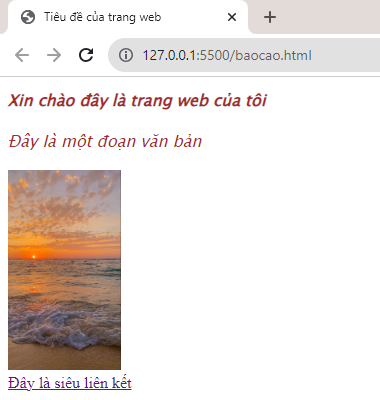
* Bây giờ, trong phần CSS có thể sử dụng lựa chọn `.` kết hợp với tên lớp để áp dụng kiểu cho các phần tử có lớp tương ứng. Ví dụ:

****

**Hình 2. 7. Tạo kiểu dựa trên lớp**

* Trong ví dụ trên, chúng ta đã tạo một lớp có tên là "my-class" và áp dụng kiểu màu chữ, kích thước chữ, phông chữ và kiểu phông chữ cho mọi phần tử có lớp đó.
* Có thể áp dụng một lớp cho nhiều phần tử khác nhau trong tệp HTML bằng cách thêm
* <p class="my-class">Đây là một đoạn văn bản</p>
* <h1 class="my-class">Xin chào đây là trang web của tôi</h1>

Như vậy, tất cả đoạn văn bản và tiêu đề có lớp "my-class" sẽ được áp dụng vào cả 2 kết quả như sau

****

**Hình 2. 8. Chạy kết quả**

Thông qua cách này, ta có thể tạo các lớp tùy chỉnh và áp dụng kiểu khác nhau cho từng lớp một cách linh hoạt trong trang web của bạn.

Từ ví dụ trên, nhóm áp dụng kiểu bằng cách sử dụng lựa chọn

****

**Hình 2. 9. Tạo kiểu lựa chọn trong HTML**

* Sau khi chạy chương trình, kết quả là

****

**Hình 2. 10. Chạy chương trình**

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

Để thử nghiệm các thuộc tính CSS cơ bản như font chữ, màu sắc, độ rộng và chiều cao, bạn có thể làm như sau:

### **2.3.1. Font chữ:**

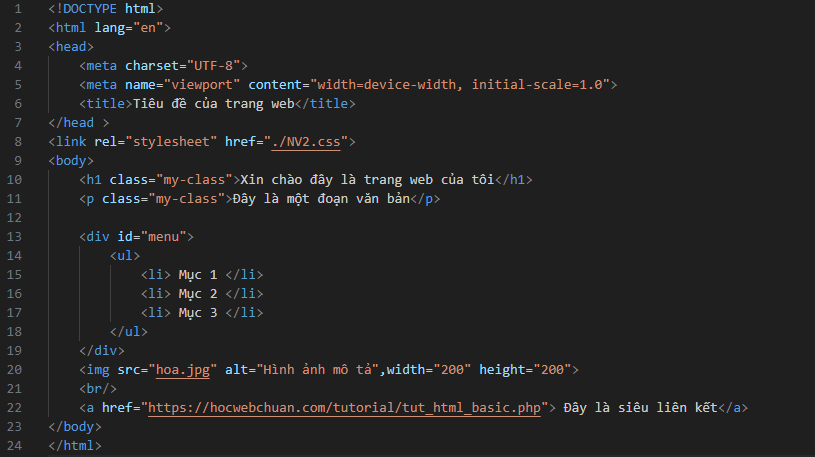
* Sử dụng thuộc tính `font-family` để đặt font chữ cho một phần tử. Ví dụ: `font-family: Arial, sans-serif;`
* Sử dụng thuộc tính `font-size` để đặt kích thước font chữ. Ví dụ: `font-size: 16px; `

### **2.3.2. Màu sắc:**

* Sử dụng thuộc tính `color` để đặt màu chữ cho một phần tử. Ví dụ:`color: blue;`
* Sử dụng thuộc tính `background-color` để đặt màu nền cho một phần tử. Ví dụ: `background-color: yellow;`

**2.3.3. Độ rộng và chiều cao:**

* Sử dụng thuộc tính `width` để đặt độ rộng cho một phần tử. Ví dụ: `width: 300px;`
* Sử dụng thuộc tính `height` để đặt chiều cao cho một phần tử. Ví dụ: `height: 50px;`
* Thử nghiệm các thuộc tính này bằng cách thêm chúng vào phần CSS của trang web của bạn. Ví dụ:

****

**Hình 2. 11. Thử nghiệm thuộc tính tạo kiểu CSS**

* Chạy chương trình

****

**Hình 2. 12. Chạy chương trình**

Ngoài ra, CSS có các thuộc tính khác như **`padding`, `margin`, `border`**, và nhiều thuộc tính khác để tùy chỉnh kiểu cho phần tử trong trang web của bạn

* **`padding`**: Được sử dụng để tạo khoảng cách giữa nội dung bên trong phần tử và viền của phần tử.

Có thể đặt giá trị padding cho các cạnh riêng biệt hoặc sử dụng thuộc tính **`padding-top`, `padding-right`, `padding-bottom`**và **`padding-left`** để chỉ định giá trị cho từng cạnh một cách riêng lẻ.

* **`margin`**: Được sử dụng để tạo khoảng cách giữa các phần tử.

Khác với `padding`, **`margin`** không ảnh hưởng đến nội dung bên trong phần tử. Cũng giống như `padding`, có thể đặt giá trị margin cho các cạnh riêng biệt hoặc sử dụng thuộc tính **`margin-top`, `margin-right`**, **`margin-bottom`** và **`margin-left`** để chỉ định giá trị cho từng cạnh một cách riêng lẻ.

* **`border`**: Được sử dụng để tạo đường viền xung quanh một phần tử. Có thể thiết lập độ dày, kiểu và màu sắc của đường viền bằng cách sử dụng các thuộc tính **`border-width`, `border-style`**và**`border-color`**.

Cũng giống như `padding` và `margin`, có thể thiết lập giá trị cho từng cạnh bằng cách sử dụng các thuộc tính có tiền tố là **`border-top`, `border-right`, `border-bottom`**và**`border-left`.**

# **3. JavaScript Cơ bản:**

## **3.1. Giới thiệu cú pháp JavaScript cơ bản.**

Cú pháp JavaScript cơ bản:

* Tạo biến:
* Sử dụng từ khóa "var", "let" hoặc "const" để tạo biến.

var name = "John";

let age = 25;

const PI = 3.14;

* Biến `var` có phạm vi toàn cục hoặc hàm, biến `let` và `const` có phạm vi block-scoped (phạm vi của một khối mã lệnh).
* Biến`let` cho phép bạn gán lại giá trị cho biến, trong khi `const` không cho phép gán lại giá trị sau khi đã được khởi tạo.
* JavaScript phân biệt chữ hoa chữ thường. Điều này có nghĩa là các biến, từ khóa ngôn ngữ, tên hàm và các mã định danh khác phải luôn được nhập với chữ viết nhất quán.
* Ngoài tạo 1 tệp JavaScript, có thể đặt các thẻ <script>, chứa JavaScript bất cứ đâu trong tệp HTML, nhưng được đề nghị là nên giữ trong các thẻ <head>.
* Thẻ script nhận hai thuộc tính quan trọng sau:
* Language − Thuộc tính này xác định ngôn ngữ scripting nào bạn đang sử dụng. Giá trị của nó sẽ là javascript. Mặc dù các phiên bản gần đây của HTML (và XHTML) đã thôi dần không sử dụng thuộc tính này nữa.
* Type − Thuộc tính này là những gì bây giờ được đề nghị sử dụng và giá trị của nó nên được thiết lập là "text/javascript". Ví dụ
* <script language="javascript" type="text/javascript">
* JavaScript code
* </script>

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

* Tạo biến và xử lý sự kiện :
* Sử dụng từ khoá `function` để định nghĩa hàm.
* Để gán một hàm xử lý sự kiện cho một phần tử HTML, ta sử dụng cú pháp sau:
* Ví dụ:

document.getElementById("myButton").onclick = function() {

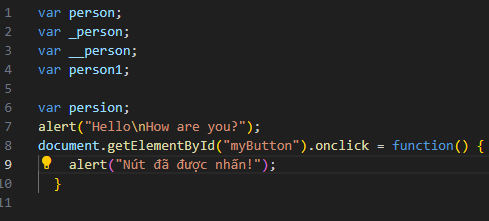
alert("Nút đã được nhấn!");

}

* Dưới đây là một ví dụ đơn giản về hiển thị thông báo khi nút được nhấn: document.getElementById("myButton").onclick = function() {

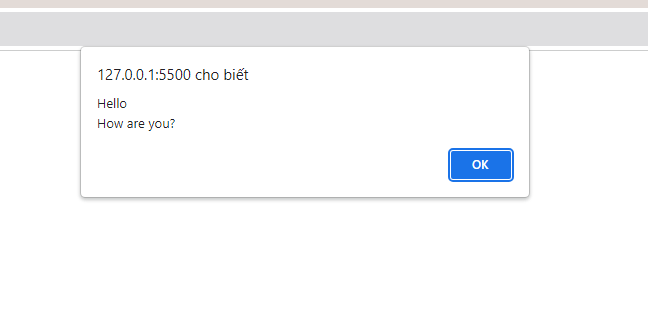
alert("Nút đã được nhấn!");

}



**Hình 3. 1. Tạo biến, hàm và sử dụng sự kiện**

* Trong ví dụ trên, khi người dùng nhấn vào nút có id là "myButton", một thông báo sẽ được hiển thị trên trình duyệt.



**Hình 3. 2. Kết quả**

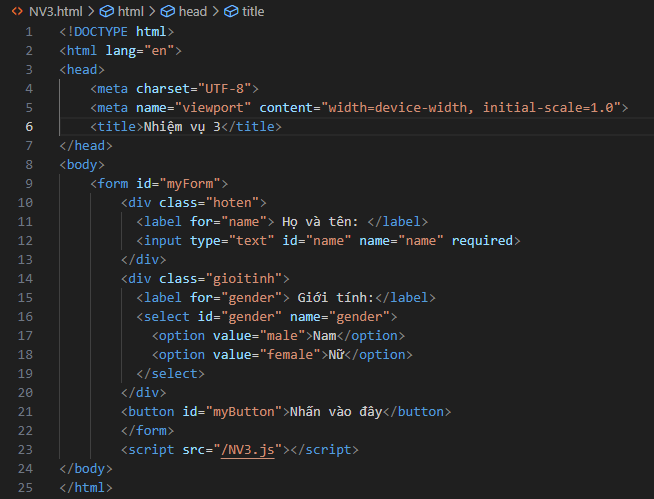
## **3.3. Tạo các ví dụ đơn giản như hiển thị thông báo khi nút được nhấn**

* Sử dụng sự kiện:
* Sử dụng phương thức `addEventListener()` để thêm một chức năng xử lý cho một sự kiện cụ thể của một phần tử HTML.

Trong phần tham số thứ hai của `addEventListener()`, bạn có thể tạo một hàm xử lý sự kiện riêng, hoặc sử dụng một hàm đã được định nghĩa trước đó.

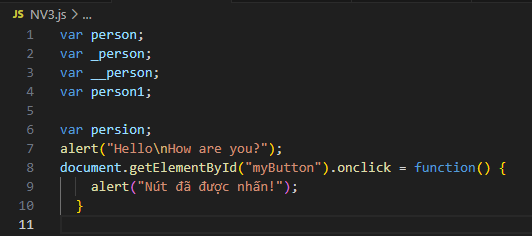
* Ví dụ đơn giản hiển thị thông báo khi nút được nhấn:
* HTML:

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

****

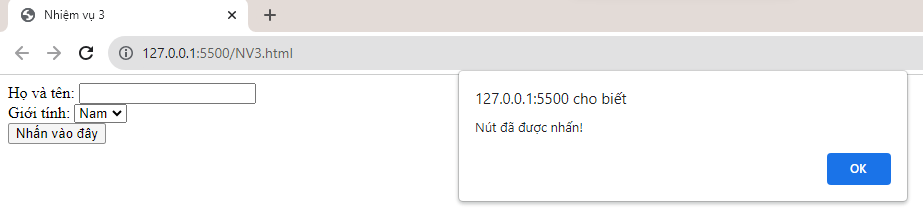
**Hình 3. 3. Thêm Button thao tác trên web**

* JavaScript:

****

**Hình 3. 4. Các câu lệnh trong JavaScipt**

Trong ví dụ trên, chúng ta đã sử dụng JavaScript để thêm một hàm xử lý sự kiện cho nút. Khi nút được nhấn, sự kiện `click` sẽ được kích hoạt và thông báo "Bạn đã nhấn nút!" sẽ hiển thị. Có thể sửa đổi thông báo, thay đổi phần tử mà sự kiện áp dụng và tùy chỉnh hành vi cho các sự kiện khác nhau.

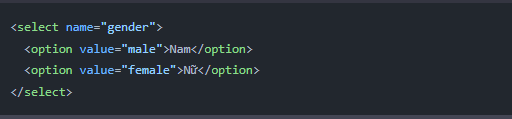
****

**Hình 3. 5. Kết quả**

**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.**

* Tạo biểu mẫu HTML:
* Sử dụng thẻ `<form>` để bao bọc các phần tử nhập liệu trong biểu mẫu.
* Sử dụng thẻ `<input>` với thuộc tính `type` là "text" để tạo ô văn bản. Ví dụ: `<input type="text" name="name">`
* Sử dụng thẻ `<select>` và các phần tử `<option>` để tạo ô chọn. Ví dụ:

****

**Hình 4. 1. Ví dụ thẻ lựa chọn**

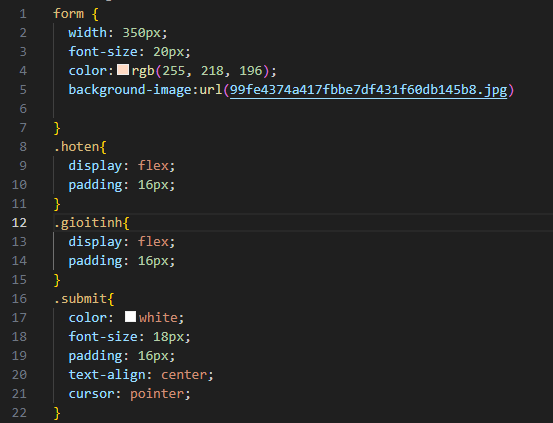
* Sử dụng thẻ `<button>` với thuộc tính `type` là "submit" để tạo nút gửi. Ví dụ: `<button type="submit">Gửi</button>`
* Ví dụ về biểu mẫu HTML đơn giản:



**Hình 4.2. Ví dụ HTML**

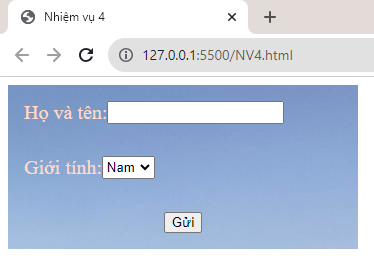
## **4.2. Tạo kiểu cho biểu mẫu bằng CSS.**

* Tạo kiểu cho biểu mẫu bằng CSS:
* Sử dụng lớp hoặc ID để chọn các phần tử trong biểu mẫu.
* Sử dụng thuộc tính CSS như `color`, `background-color`, `padding`, và `margin` để tùy chỉnh kiểu cho các phần tử.
* Ví dụ:

****

**Hình 4. 3. Tạo kiểu CSS**

* Kết quả khi áp dụng

****

**Hình 4. 4. Chạy chương trình**

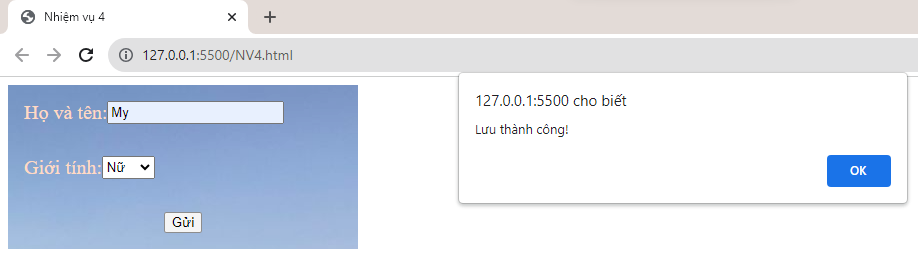
## **4.3. Sử dụng JavaScript để xử lý dữ liệu từ biểu mẫu (vd: kiểm tra dữ liệu hợp lệ).**

* Sử dụng JavaScript để xử lý dữ liệu từ biểu mẫu:
* Xử lý một sự kiện như khi biểu mẫu được gửi đi, bạn có thể sử dụng sự kiện `submit`.
* Sử dụng phương thức `addEventListener()` để thêm một chức năng xử lý cho sự kiện `submit`.
* Trong chức năng xử lý, bạn có thể sử dụng các phương thức như `preventDefault()` để ngăn chặn hành vi mặc định của trình duyệt và xử lý dữ liệu từ biểu mẫu.
* Ví dụ:

**Hình 4. 5. Xử lý dữ liệu trong JavaScript**

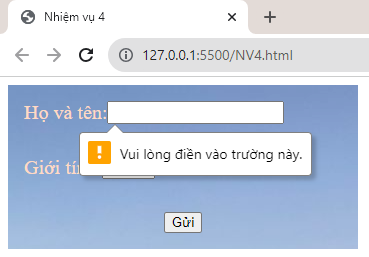
Trên đây là hướng dẫn cách tạo một biểu mẫu HTML đơn giản với các phần tử nhập liệu, tạo kiểu cho nó bằng CSS, và sử dụng JavaScript để xử lý dữ liệu từ biểu mẫu.

* Kết quả sau khi chạy chương trình
* Lưu thông tin vừa điền thành công

****

**Hình 4.6. Kết quả lưu thành công**

* Khi thao tác lưu bất thường, hiện ra thông báo



**Hình 4. 6. Chưa điền thông tin hợp lệ**

# **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.**

### **5.1.1. Khảo sát và thu thập yêu cầu:**

* Thu thập thông tin từ khách hàng và hiểu rõ yêu cầu của họ.
* Hiểu rõ nhu cầu và mong muốn của khách hàng để có thể thiết kế một website phù hợp.

### **5.1.2. Lập kế hoạch:**

* Đặt ra mục tiêu, tiến độ, ngân sách và tài nguyên cần thiết.

### **5.1.3. Thiết kế:**

* Xây dựng sơ đồ trang web để xác định cấu trúc và liên kết giữa các trang.
* Xác định giao diện người dùng (UI) và trải nghiệm người dùng (UX).
* Thiết kế giao diện người dùng, bao gồm màu sắc, hình ảnh, thông điệp và định dạng.
* Xác định các yếu tố và phần tử cần thiết như các trường nhập liệu, nút, công cụ tìm kiếm, hình ảnh và video.

### **5.1.4. Phát triển:**

* Xây dựng các trang web hoặc ứng dụng web sử dụng các ngôn ngữ lập trình như HTML, CSS và JavaScript.
* Tạo cơ sở dữ liệu và xử lý dữ liệu bên phía máy chủ sử dụng ngôn ngữ như PHP, Python hoặc Ruby.
* Tối ưu hóa mã nguồn, đảm bảo code gọn gàng, dễ đọc và dễ bảo trì.

Điều này bao gồm việc xử lý dữ liệu, tạo ra các trang web động, tích hợp các công cụ và plugin cần thiết, và thực hiện kiểm tra kỹ thuật để đảm bảo rằng website hoạt động một cách trơn tru.

### **5.1.5.Kiểm thử:**

* Kiểm tra chức năng của trang web để đảm bảo rằng mọi yêu cầu đã được đáp ứng.
* Kiểm tra sự tương thích trên nhiều trình duyệt và các nền tảng khác nhau.
* Xác minh tính bảo mật và bảo vệ của trang web.
* Đảm bảo trang web tương thích với nhiều thiết bị và kích thước màn hình khác nhau (phản hồi).

### **5.1.6. Triển khai:**

* Chuẩn bị môi trường triển khai, bao gồm máy chủ, tên miền, cấu hình hệ thống và giao thức truyền thông.
* Tải lên mã nguồn và cơ sở dữ liệu lên máy chủ.
* Kiểm tra lại trang web trước khi cho phép người dùng truy cập.
* Sửa lỗi và cải tiến trang web dựa trên phản hồi từ người dùng.

### **5.1.7. Đánh giá và cải tiến:**

* Cập nhật và bảo trì trang web để đảm bảo tính ổn định và an toàn.
* Thường xuyên kiểm tra và cập nhật các phiên bản mới của các công nghệ và thư viện sử dụng trong trang web.

Như vậy, quy trình này giúp đảm bảo rằng trang web được xây dựng và triển khai một cách có hệ thống, đáp ứng được yêu cầu của khách hàng và mang lại trải nghiệm người dùng tốt.

## **5.2. Xác định các bước quan trọng như phân tích yêu cầu, thiết kế giao diện, phát**

## **triển, kiểm thử, và triển khai.**

### **5.2.1. Phân tích yêu cầu:**

* Xác định yêu cầu chức năng và phi chức năng của trang web hoặc ứng dụng web.
* Thu thập thông tin từ khách hàng hoặc người dùng cuối để hiểu đúng những gì họ mong đợi từ trang web.
* Xác định kịch bản sử dụng, luồng công việc, và danh sách tính năng cần có.

### **5.2.2. Thiết kế giao diện:**

* Xây dựng sơ đồ trang web để định vị các trang và cấu trúc giao diện người dùng.
* Thiết kế giao diện người dùng, bao gồm màu sắc, kích thước, hình ảnh, văn bản và định dạng chung.
* Xác định bố cục và định vị các phần tử giao diện như trường nhập liệu, nút, menu, thanh điều hướng và vùng hiển thị nội dung.
* Phát triển:
* Xây dựng trang web hoặc ứng dụng web sử dụng các ngôn ngữ lập trình và công nghệ thích hợp (như HTML, CSS, JavaScript, PHP, Python, Ruby, v.v.).
* Tạo giao diện người dùng hoạt động và tương tác thông qua mã lập trình.
* Xây dựng hệ thống và cơ sở dữ liệu phù hợp với yêu cầu và thiết kế.

### **5.2.3. Kiểm thử:**

* Kiểm tra chức năng và hiệu suất của trang web hoặc ứng dụng để đảm bảo rằng nó hoạt động đúng, mượt mà và tương thích.
* Kiểm tra giao diện người dùng và trải nghiệm người dùng để đảm bảo tương thích trên nhiều trình duyệt và thiết bị khác nhau.
* Kiểm tra tính bảo mật và bảo vệ của trang web với việc kiểm tra lỗ hổng bảo mật, ứng phó với tấn công tiềm ẩn và xác minh tính bảo mật của dữ liệu.

### **5.2.4. Triển khai:**

* Cấu hình môi trường triển khai bao gồm máy chủ, tên miền và cấu hình hệ thống.
* Tải lên mã nguồn và cơ sở dữ liệu lên máy chủ hoặc hệ thống triển khai.
* Kiểm tra lại toàn bộ trang web hoặc ứng dụng trước khi cho phép người dùng cuối truy cập.
* Quy trình này giúp đảm bảo rằng trang web được phát triển và triển khai một cách có hệ thống và đúng yêu cầu của khách hàng. Các bước này được thực hiện tuần tự và có thể có sự chồng chéo và tương tác trong quá trình phát triển thực tế.

## **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.**

Trong một nhóm dự án, mỗi thành viên đóng vai trò quan trọng và có trách nhiệm riêng để đảm bảo công việc diễn ra suôn sẻ. Xác định các kỹ năng và chuyên môn cần thiết cho từng vai trò

### **5.3.1.Quản lý dự án:**

* Trưởng nhóm: Đóng vai trò lãnh đạo, phân phối công việc và giám sát tiến độ của dự án. Trưởng nhóm có trách nhiệm tạo ra và duy trì một môi trường làm việc tích cực trong nhóm.
* 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 và tiêu chuẩn đã đề ra trong dự án.

### **5.3.2. Xây dựng dự án:**

* 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.
* Chuyên gia cơ sở dữ liệu: việc quản lý cơ sở dữ liệu là rất quan trọng. Chuyên gia cơ sở dữ liệu cần có kiến thức sâu về hệ quản trị cơ sở dữ liệu. Sự hiểu biết về việc tối ưu hóa cơ sở dữ liệu và quản lý hiệu suất cũng là một kỹ năng quan trọng của vai trò này.
* Nhà phát triển: Tạo ra mã nguồn, triển khai các chức năng và tích hợp các yêu cầu kỹ thuật. 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: Phát triển và triển khai các kịch bản kiểm thử để đảm bảo tính ổn định và chất lượng của sản phẩm. Kiểm thử viên kiểm tra, ghi lại và báo cáo về các lỗi và vấn đề trong quá trình thử nghiệm.

Điều quan trọng là mỗi thành viên trong nhóm phải hiểu rõ vai trò và trách nhiệm của mình, cùng với cách hoạt động và giao tiếp hiệu quả với các thành viên khác trong nhóm. Điều này sẽ giúp tạo ra một môi trường làm việc tích cực và đảm bảo sự thành công của dự án.