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

Mục lục

[**1.** **Giới thiệu về HTML** 3](#_Toc149116737)

[**1.1.** **Khái niệm** 3](#_Toc149116738)

[**1.2.** **Ưu điểm và nhược điểm** 3](#_Toc149116739)

[**1.3.** **Hướng dẫn tạo một tệp HTML cơ bản** 3](#_Toc149116740)

[**1.4.** **Tạo trang web với tiêu đề, đoạn văn bản, hình ảnh và siêu liên kết** 4](#_Toc149116741)

[**1.5.** Hướng dẫn về cú pháp và các phần tử HTML cơ bản 6](#_Toc149116742)

[**2.** **Cơ bản về CSS** 8](#_Toc149116743)

[**2.1.** **Khái niệm** 8](#_Toc149116744)

[**2.2.** **Ưu điểm và nhược điểm** 8](#_Toc149116745)

[**2.3.** **Hướng dẫn cách thêm kiểu cho trang web bằng CSS** 8](#_Toc149116746)

[**2.4.** **Tạo các lớp và áp dụng kiểu bằng cách sử dụng lựa chọn** 9](#_Toc149116747)

[**2.5.** **Thử nghiệm với các thuộc tính CSS cơ bản như font chữ, màu sắc, độ rộng và chiều cao** 10](#_Toc149116748)

[**3.** **JavaScript cơ bản** 11](#_Toc149116749)

[**3.1.** **Khái niệm:** 11](#_Toc149116750)

[**3.2.** **Ưu điểm và nhược điểm** 11](#_Toc149116751)

[**3.3.** **Hướng dẫn tạo biến, hàm và giới thiệu cú pháp JavaScript cơ bản** 12](#_Toc149116752)

[**3.3.1.** **Biến** 12](#_Toc149116753)

[**3.3.2.** **Kiểu dữ liệu** 12](#_Toc149116754)

[**3.3.3.** **Toán tử** 12](#_Toc149116755)

[**3.3.4.** **Câu điều kiện** 13](#_Toc149116756)

[**3.3.5.** **Vòng lặp** 13](#_Toc149116757)

[**3.3.6.** **Hàm** 13](#_Toc149116758)

[**3.3.7.** **Sự kiện:** 14](#_Toc149116759)

[**3.3.8.** **Mảng** 14](#_Toc149116760)

[**3.3.9.** **Đối tượng** 15](#_Toc149116761)

[**3.4.** **Ví dụ về việc hiển thị thông báo khi nút được nhấn** 15](#_Toc149116762)

[**4.** **HTML Forms và Input Elements** 15](#_Toc149116763)

[**4.1.** **Hướng dẫn tạo biểu mẫu HTML đơn giản với các phần tử nhập liệu như ô văn bản, ô chọn và nút gửi** 16](#_Toc149116764)

[**4.2.** **Tạo kiểu cho biểu mẫu bằng CSS** 17](#_Toc149116765)

[**4.3.** **Sử dụng JavaScript để xử lý dữ liệu từ biểu mẫu** 18](#_Toc149116766)

[**5.** **Quy trình phát triển web** 19](#_Toc149116767)

[**5.1.** **Giới thiệu về quy trình phát triển web từ lập kế hoạch đến triển khai trang web** 19](#_Toc149116768)

[**5.2.** **Xác định các bước quan trọng** 19](#_Toc149116769)

[**5.2.1.** **Lập kế hoạch và phân tích yêu cầu** 19](#_Toc149116770)

[**5.2.2.** **Thiết kế giao diện và cơ sở dữ liệu** 20](#_Toc149116771)

[**5.2.3.** **Triển khai dự án** 20](#_Toc149116772)

[**5.2.4.** **Kiểm thử** 20](#_Toc149116773)

[**5.3.** **Hướng dẫn về vai trò và trách nhiệm của từng thành viên trong nhóm** 20](#_Toc149116774)

[**6.** **Thực hành xây dựng trang web đơn giản** 20](#_Toc149116775)

[**6.1.** **Xây dựng trang web đơn giản từ đầu bằng việc kết hợp HTML, CSS và JavaScript** 20](#_Toc149116776)

[**6.2.** **Xây dựng 1 website hoàn thiện (Web động có kết nối CSDL)** 23](#_Toc149116777)

[**7.** **Reponsive Web Design:** 25](#_Toc149116778)

[**7.1.** **Định nghĩa:** 25](#_Toc149116779)

[**8.** **HTML5 và CSS3** 27](#_Toc149116780)

[**8.1.** **Giới thiệu về các tính năng mới trong HTML5 và CSS3** 27](#_Toc149116781)

[**8.1.1.** **HTML 5** 27](#_Toc149116782)

[**8.1.2.** **CSS3** 28](#_Toc149116783)

[**8.2.** **Hướng dẫn cách sử dụng các phần tử HTML5** 29](#_Toc149116784)

[**8.2.1.** **Video và Âm thanh** 29](#_Toc149116785)

[**8.2.2.** **Canvas** 30](#_Toc149116786)

[**8.2.3.** **Đồ họa vector** 30](#_Toc149116787)

[**8.3.** **Sử dụng CSS3 để thêm các hiệu ứng và chuyển động vào trang web** 31](#_Toc149116788)

[**8.3.1.** **Transitions** 31](#_Toc149116789)

[**8.3.2.** **Animations** 32](#_Toc149116790)

[**9.** **Thiết kế trang web thân thiện với SEO** 33](#_Toc149116791)

[**9.1.** **Giới thiệu về SEO (Search Engine Optimization) và tại sao nó quan trọng đối với trang web** 33](#_Toc149116792)

[**9.2.** **Cách tối ưu hóa tiêu đề trang, thẻ meta và nội dung để cải thiện thứ hạng tìm kiếm** 34](#_Toc149116793)

[**9.2.1.** **Tiêu đề trang** 34](#_Toc149116794)

[**9.2.2.** **Thẻ meta** 34](#_Toc149116795)

[**9.2.3.** **Nội dung** 34](#_Toc149116796)

[**9.3.** **Sử dụng công cụ tối ưu hóa SEO để theo dõi hiệu suất và tối ưu hóa trang web** 35](#_Toc149116797)

[**10.** **Quản lý phiên bản và Git** 35](#_Toc149116798)

[**10.1.** **Giới thiệu về hệ thống quản lý mã nguồn và Git** 35](#_Toc149116799)

[**10.2.** **Hướng dẫn tạo kho lưu trữ (reponsitory) Git và làm việc với các lệnh cơ bản như git clone, git pull, git push và git commit** 36](#_Toc149116800)

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>

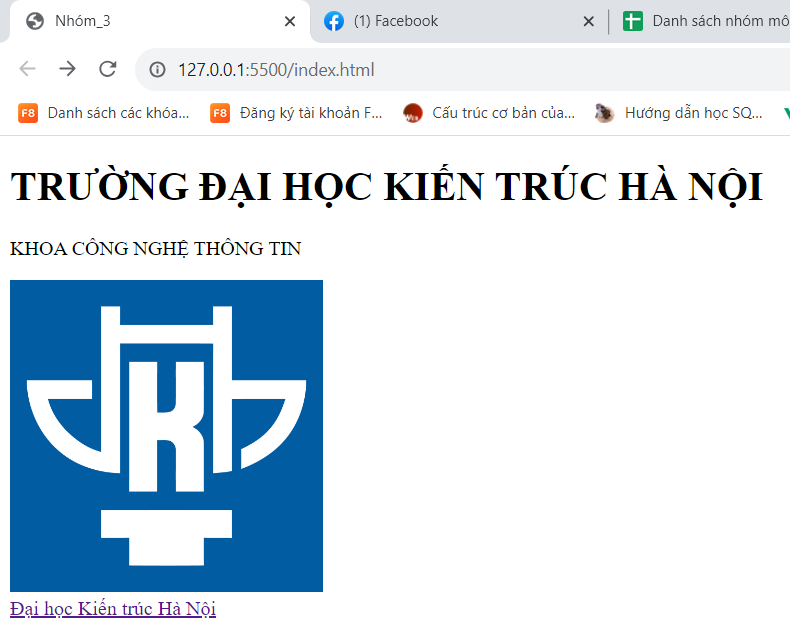
        <a href="https://www.facebook.com/DHKIENTRUCHN">Đại học Kiến trúc Hà Nội</a> <!-- siêu liên kết -->

    </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:

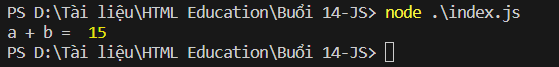
let a = 5;

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;

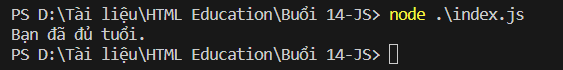
if (age < 18){

    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

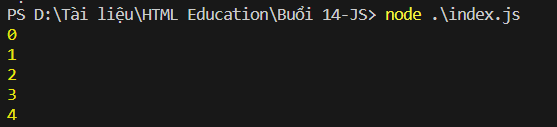
let i = 0;

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

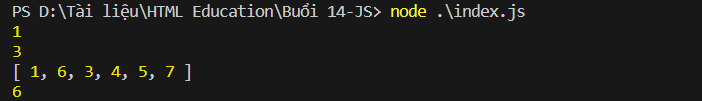
numbers[1] = 6;

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!");

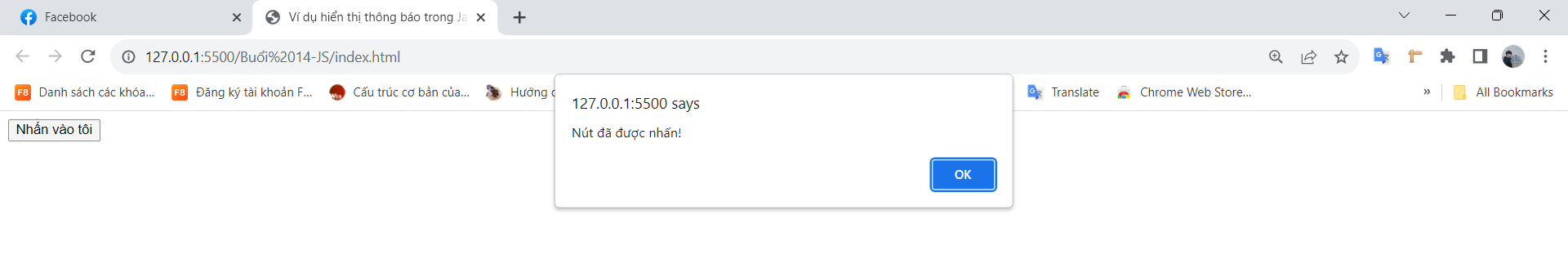
    });

  </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**
* Nhóm trưởng : Xác định đề tài , xây dựng các chức năng của đề tài , phân chia công việc , quản lý công việc 1 cách hiệu quả
* Thành viên nhóm : Thực hiện các phần công việc được giao 1 cách hoàn chỉnh , báo cáo lại những gì đã làm được

1. **Thực hành xây dựng trang web đơn giản**
   1. **Xây dựng trang web đơn giản từ đầu bằng việc kết hợp HTML, CSS và JavaScript**

6.1.1: Xây dựng trang web bằng HTML:

A screenshot of a computer

Description automatically generated

6.1.2: Kết hợp thêm với CSS:

- Để kết hợp thêm CSS ở đây ra dùng thẻ link để link file CSS vào , sau khi link thì ta được web như sau

A screenshot of a computer

Description automatically generated

6.1.3: Kết hợp với JavaScript :

- Tạo modal thêm mới:

A screenshot of a computer

Description automatically generated

-Dùng logic để in ra các phần tử sau khi thêm :

A screenshot of a computer

Description automatically generated

-Sửa Blog:  
A screenshot of a computer

Description automatically generated

- Xóa Blog:

A screenshot of a computer

Description automatically generated

* 1. **Xây dựng 1 website hoàn thiện (Web động có kết nối CSDL)**

Node.js và MongoDB là hai công nghệ quan trọng trong lĩnh vực phát triển phần mềm và ứng dụng web hiện đại. Hãy xem xét từng công nghệ một:

1.NodeJS:   
Node.js là một môi trường chạy mã JavaScript ở phía máy chủ, được xây dựng dựa trên Chrome V8 JavaScript Engine. Điều này cho phép bạn thực hiện mã JavaScript trên máy chủ, thay vì chỉ trên trình duyệt. Dưới đây là một số điểm quan trọng về Node.js:

**Hiệu suất cao**: Node.js được thiết kế để hoạt động với cơ chế xử lý không đồng bộ (asynchronous) giúp tối ưu hóa hiệu suất ứng dụng. Điều này làm cho nó trở thành một lựa chọn phù hợp cho các ứng dụng yêu cầu xử lý đồng thời nhiều kết nối như ứng dụng thời gian thực.

**Khả năng mở rộng**: Node.js có sự hỗ trợ mạnh mẽ cho việc mở rộng ứng dụng. Bạn có thể dễ dàng thêm các module bên ngoài để tạo ra ứng dụng phức tạp và mở rộng nó theo nhu cầu.

**Cộng đồng phát triển lớn:** Cộng đồng phát triển Node.js rất lớn và nhiệt tình. Điều này đồng nghĩa với việc có nhiều thư viện và framework hữu ích để giúp bạn xây dựng ứng dụng nhanh chóng và hiệu quả.

2.MongoDB:

MongoDB là một hệ thống quản lý cơ sở dữ liệu NoSQL, dựa trên nguyên tắc lưu trữ dữ liệu dưới dạng JSON-like (BSON) và sử dụng các cặp khóa-giá trị để truy xuất dữ liệu. Dưới đây là một số điểm quan trọng về MongoDB:

**Cấu trúc dữ liệu linh hoạt:** MongoDB không yêu cầu cấu trúc dữ liệu cố định và cho phép bạn lưu trữ dữ liệu với cấu trúc linh hoạt. Điều này làm cho nó phù hợp cho các ứng dụng yêu cầu mô hình dữ liệu thay đổi thường xuyên.

**Mã nguồn mở và miễn phí:** MongoDB là một hệ thống mã nguồn mở, có nghĩa là bạn có thể sử dụng nó miễn phí và thậm chí tùy chỉnh mã nguồn theo nhu cầu của bạn.

**Khả năng mở rộng dọc:** MongoDB được thiết kế để dễ dàng mở rộng theo chiều ngang bằng cách thêm các máy chủ. Điều này giúp bạn xây dựng các hệ thống có khả năng xử lý lượng dữ liệu lớn.

Dưới đây là trang web sau khi kết nối thêm CSDL:

A screenshot of a computer

Description automatically generated

1. **Reponsive Web Design:**
   1. **Định nghĩa:**

Responsive web, hay còn gọi là "web responsive," là một phong cách thiết kế web và phát triển trang web mà mục tiêu chính là tạo ra các trang web có khả năng thích nghi với nhiều loại thiết bị và kích thước màn hình khác nhau. Mục tiêu của responsive web là đảm bảo rằng trang web hiển thị và hoạt động tốt trên các thiết bị như máy tính, điện thoại di động, máy tính bảng và các thiết bị khác mà người dùng có thể sử dụng để truy cập trang web.

Cách thiết kế responsive web thường bao gồm sử dụng các phương pháp linh hoạt như sử dụng media queries để thay đổi cách hiển thị nội dung dựa trên kích thước màn hình và sử dụng đơn vị đo lường linh hoạt như phần trăm thay vì pixel cố định. Điều này giúp trang web tự động thích nghi với mọi kích thước màn hình mà không cần phải tạo các phiên bản riêng biệt cho từng loại thiết bị.

Responsive web đóng vai trò quan trọng trong thế giới kỹ thuật số hiện đại, bởi vì người dùng truy cập internet thông qua nhiều loại thiết bị khác nhau. Trang web responsive giúp cung cấp trải nghiệm người dùng tốt hơn và giảm nguy cơ mất khách hàng do việc hiển thị không đúng trên các thiết bị khác nhau.

* 1. Áp dụng vào trong trang web:
* Giao diện dành cho các thiết bị máy tính :

A screenshot of a computer

Description automatically generated

* Giao diện dành cho Tablet (Máy tính bảng):

A screenshot of a computer

Description automatically generated

* Giao diện dành cho điện thoại:

A screenshot of a phone

Description automatically generated

1. **HTML5 và CSS3**
   1. **Giới thiệu về các tính năng mới trong HTML5 và CSS3**
      1. **HTML 5**

HTML5 là phiên bản mới nhất của ngôn ngữ đánh dấu HyperText Markup Language (HTML). HTML là mã lệnh dùng để tạo ra các trang web trên Internet. HTML5 cung cấp nhiều tính năng mới và cải tiến so với phiên bản trước, bao gồm các thẻ mới, tính năng trình diễn âm thanh và video, cải tiến về bố cục và bố trí trang web, và hỗ trợ cho các thiết bị di động. HTML5 cũng được thiết kế để đáp ứng nhu cầu về việc xây dựng trang web đa nền tảng, đồng thời giúp giảm sự phụ thuộc vào các công cụ bổ sung như Flash.

**Một số tính năng nổi bật trong HTML5:**

* Thẻ sematic mới: HTML5 cung cấp nhiều thẻ sematic mới như <header>, <footer>, <article>, <nav> và <section> giúp tối ưu hóa cấu trúc và nội dung trên trang web.
* Multimedia (Đa phương tiện): HTML5 cung cấp các thẻ như <video> và <audio> để dễ dàng nhúng video và âm thanh mà không cần sử dụng các plugin như Flash.
* Canvas: HTML5 cung cấp một công cụ đồ họa 2D mới gọi là Canvas, cho phép bạn vẽ và tạo nội dung đồ họa trực tiếp trên trang web.
* Hỗ trợ cho Web Storage: HTML5 cung cấp hỗ trợ cho Web Storage, cho phép lưu trữ dữ liệu trên trình duyệt của người dùng mà không phải lưu trữ trên máy chủ.
* Form Validation: HTML5 cung cấp hỗ trợ cho xác thực dữ liệu trên form một cách tự động và dễ dàng hơn so với phiên bản cũ hơn.
* Geolocation: HTML5 cung cấp hỗ trợ cho Geolocation, cho phép trang web xác định vị trí của người dùng và sử dụng thông tin này để cung cấp nội dung tùy chỉnh cho người dùng.
  + 1. **CSS3**

Cascading Style Sheets (CSS) là một ngôn ngữ định dạng được sử dụng để tùy chỉnh cách trang web được hiển thị. CSS3 là phiên bản mới nhất của ngôn ngữ này với nhiều tính năng mạnh mẽ hơn so với các phiên bản cũ hơn của CSS. Những tính năng mới bao gồm các tùy chọn trong việc tạo các hiệu ứng, sử dụng các font chữ đẹp hơn và sử dụng các trình tự hóa để tối ưu hóa trang web.

CSS3 giúp tăng tính độc đáo và tính trải nghiệm người dùng trong việc sử dụng các trang web, và cung cấp nhiều lựa chọn cho những ai muốn tạo ra các trang web tốt hơn**.**

**Một số tính năng nổi bật trong CSS3**

* Box Shadow: Cho phép bạn thêm bóng rực rỡ vào các hình chữ nhật trong trang web của bạn.
* Border Radius: Cho phép bạn tạo các góc tròn cho các hình chữ nhật trên trang web.
* Text Shadow: Cho phép bạn thêm bóng rực rỡ vào văn bản trên trang web.
* Gradients: Cho phép bạn tạo các gradient cho các phần tử trong trang web.
* Multiple Columns: Cho phép bạn chia nội dung trang web thành nhiều cột.
* Transitions: Cho phép bạn tạo các hiệu ứng chuyển đổi giữa các trạng thái của phần tử.
* Animations: Cho phép bạn tạo các hiệu ứng hoạt hình trong trang web.
* Flexbox Layout: Cho phép bạn dễ dàng bố trí các phần tử trong trang web theo một cách linh hoạt.
* Media Queries: Cho phép bạn tùy chỉnh giao diện trang web dựa trên kích thước màn hình của người dùng.
* Rounded Corners:Cho phép bạn tạo các góc tròn cho các hình chữ nhật trên trang web.
  1. **Hướng dẫn cách sử dụng các phần tử HTML5**
     1. **Video và Âm thanh**

Thẻ <video> và <audio> cho phép nhúng video và âm thanh vào trang web

VD:

<body>

    <video controls>

        <source src="video.mp4" type="video/mp4">

        Your browser does not support the video tag.

      </video>

      <audio controls>

        <source src="audio.mp3" type="audio/mp3">

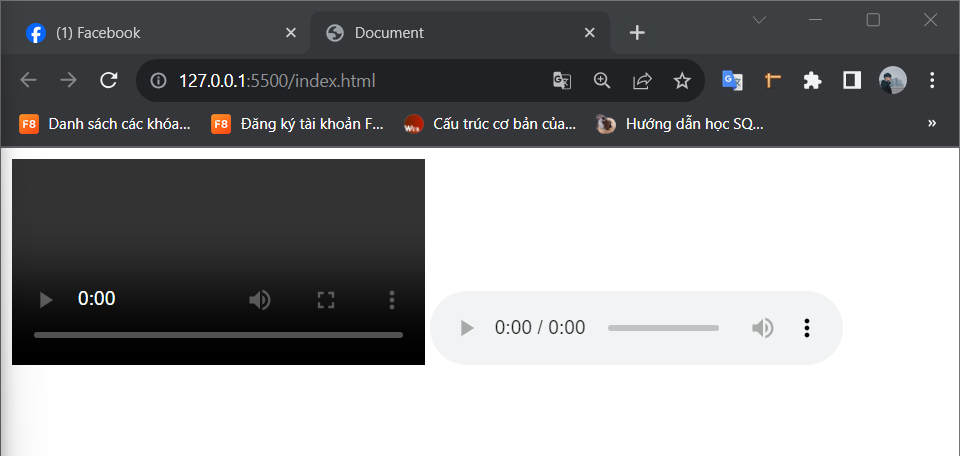
        Your browser does not support the audio tag.

      </audio>

</body>

* Thuộc tính <controls>: thêm giao diện người dùng cho việc điều khiển video và âm thanh.
* <source>: là nơi xác định đường dẫn và loại phương tiện.

Kết quả:



* + 1. **Canvas**

Thẻ <canvas> cho phép vẽ đồ họa trên trang web bằng JavaScript

VD:

* HTML:

<canvas id="myCanvas" width="400" height="200"></canvas>

* JS:

// Lấy tham chiếu đến phần tử canvas

var canvas = document.getElementById('myCanvas');

var context = canvas.getContext('2d');

// Vẽ hình tròn

context.beginPath();

context.arc(100, 100, 50, 0, 2 \* Math.PI);

context.fillStyle = 'blue';

context.fill();

Ví dụ này tạo ra một phần tử canvas có kích thước 400x200 và sau đó vẽ một hình tròn màu xanh

* + 1. **Đồ họa vector**

Thẻ <svg> cho phép vẽ đồ họa vector trên trang web.

VD:

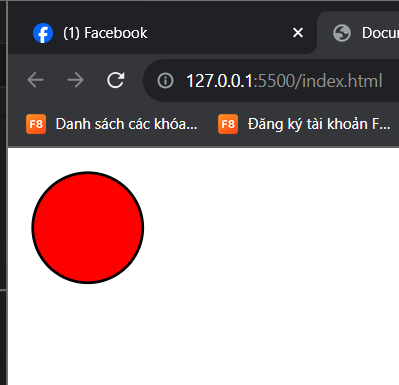
    <svg width="100" height="100">

        <circle cx="50" cy="50" r="40" stroke="black" stroke-width="2" fill="red" />

    </svg>

* Ví dụ này tạo một hình tròn đỏ bằng cách sử dụng thẻ <circle> bên trong thẻ <svg>
* Các phần tử như: cx, cy, r, stroke, stroke-width và fill dùng để điều chỉnh hình dạng và màu sắc của hình

Kết quả:



* 1. **Sử dụng CSS3 để thêm các hiệu ứng và chuyển động vào trang web**
     1. **Transitions**

CSS transitions cho phép bạn tạo các hiệu ứng chuyển đổi mượt mà khi một phần tử thay đổi trạng thái

VD:

* HTML:

<body>

    <div class="box"></div>

</body>

* CSS:

.box {

    width: 100px;

    height: 100px;

    background-color: blue;

    transition: width 0.5s, height 0.5s, background-color 0.5s;

}

/\* CSS chuyển đổi khi trỏ chuột vào \*/

.box:hover {

    width: 150px;

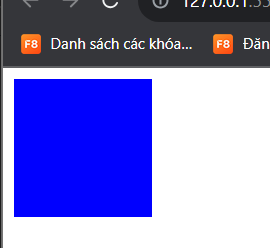
    height: 150px;

    background-color: red;

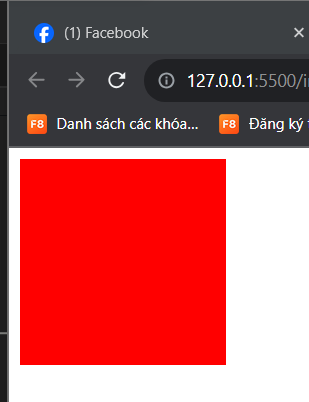
}

Kết quả:

* Trước khi hover:



* Sau khi hover:



* + 1. **Animations**

CSS animations cho phép tạo các hoạt ảnh phức tạp hơn và điều khiển chuyển động một cách chi tiết

VD:

* HTML:

<body>

    <div class="box"></div>

</body>

* CSS:

.box {

    width: 100px;

    height: 100px;

    background-color: blue;

    animation: move 2s ease-in-out infinite alternate;

}

/\* Định nghĩa hoạt ảnh \*/

@keyframes move {

    0% {

      transform: translateX(0);

    }

    100% {

      transform: translateX(200px);

    }

}

Kết quả thu được là hình vuông nền màu xanh di chuyển từ trái sang phải, từ vị trí ban đầu đến vị trí mới và ngược lại sau 2 giây

1. **Thiết kế trang web thân thiện với SEO**
   1. **Giới thiệu về SEO (Search Engine Optimization) và tại sao nó quan trọng đối với trang web**

* SEO - Search Engine Optimization (Tối ưu hóa công cụ tìm kiếm), là một loạt các chiến lược và kỹ thuật được sử dụng để tối ưu hóa một trang web để nó hiển thị ở vị trí cao trên kết quả tìm kiếm của các công cụ tìm kiếm như Google, Bing, và Yahoo.
* Mục tiêu của SEO là làm cho trang web dễ dàng được tìm thấy và hiển thị trên trang đầu của kết quả tìm kiếm khi người dùng tìm kiếm các từ khóa liên quan đến nội dung trang web đó.
* Một số lý do tại sao SEO quan trọng đối với trang web:
* **Tăng Lư lượng truy cập:** SEO giúp tăng lư lượng truy cập đến trang web. Khi trang web xếp hạng cao trên kết quả tìm kiếm, nó có cơ hội thu hút nhiều người dùng hơn.
* **Tăng Sự Tin cậy:** Khi một trang web xuất hiện ở vị trí cao trên kết quả tìm kiếm, nó thường được coi là đáng tin cậy hơn. Người dùng thường tin tưởng và nhấp vào các kết quả ở trang đầu hơn.
* **Nâng Cao Trải nghiệm Người dùng:** SEO không chỉ liên quan đến việc tối ưu hóa cho công cụ tìm kiếm mà còn tạo ra trải nghiệm tốt hơn cho người dùng. Điều này bao gồm cải thiện tốc độ tải trang, thiết kế thân thiện với thiết bị di động và cung cấp nội dung chất lượng.
* **Cạnh tranh Hiệu quả hơn:** Trong môi trường trực tuyến cạnh tranh, SEO giúp cạnh tranh với các đối thủ cùng ngành. Nếu không có chiến lược SEO, nguy cơ bị đánh bại bởi các đối thủ có tối ưu hóa tốt hơn sẽ tăng lên.
* **Tiết kiệm Chi phí Quảng cáo:** SEO là một phần quan trọng của tiếp thị trực tuyến, và nó có thể giúp giảm chi phí quảng cáo trả cho quảng cáo trả tiền mỗi lần nhấp.
* **Phân tích Dữ liệu và Cải thiện:** SEO cung cấp các công cụ để theo dõi và phân tích dữ liệu về lư lượng truy cập, hành vi người dùng và hiệu suất trang web. Điều này cho phép cải thiện liên tục trang web của mình dựa trên thông tin thống kê.
* **Tạo Tương tác với Khách hàng:** SEO cung cấp cơ hội để tạo tương tác với khách hàng thông qua nội dung cung cấp và tương tác trên mạng xã hội, giúp tạo dự án thương hiệu mạnh mẽ.
  1. **Cách tối ưu hóa tiêu đề trang, thẻ meta và nội dung để cải thiện thứ hạng tìm kiếm**
     1. **Tiêu đề trang**
* Tiêu đề trang là một trong những yếu tố quan trọng nhất cho SEO. Nó nên chứa từ khóa chính mà bạn muốn trang web của bạn xếp hạng cho. Đảm bảo tiêu đề thể hiện nội dung của trang.
* Hạn chế độ dài của tiêu đề, khoảng 50-60 ký tự là lý tưởng để hiển thị trên kết quả tìm kiếm.
  + 1. **Thẻ meta**
* Thẻ meta description không ảnh hưởng trực tiếp đến thứ hạng tìm kiếm, nhưng nó quan trọng để làm cho người dùng quyết định có nhấp vào kết quả tìm kiếm của bạn hay không. Nó nên chứa mô tả ngắn gọn và hấp dẫn về nội dung của trang.
* Hạn chế độ dài của thẻ meta description, khoảng 150-160 ký tự.
  + 1. **Nội dung**
* Tạo nội dung chất lượng và hữu ích cho người đọc. Nội dung nên cung cấp giá trị và giải quyết vấn đề hoặc nhu cầu của người dùng.
* Tối ưu hóa hình ảnh và multimedia để giảm thời gian tải trang.
* Tạo liên kết nội bộ để kết nối các trang trong trang web của bạn.
  1. **Sử dụng công cụ tối ưu hóa SEO để theo dõi hiệu suất và tối ưu hóa trang web**

Sử dụng công cụ tối ưu hóa SEO là một phần quan trọng trong việc theo dõi và cải thiện hiệu suất trang web của bạn.

Một số công cụ phổ biến sử dụng để tối ưu hóa SEO và theo dõi trang web:

* Google Search Console: là một công cụ miễn phí từ Google giúp theo dõi hiệu suất trang web trên Google Search.
* Google Analytics: là một công cụ miễn phí giúp theo dõi lư lượng truy cập trang web, hành vi người dùng, nguồn lư lượng, và nhiều dữ liệu khác.
* SEMrush: là một công cụ tối ưu hóa SEO mạnh mẽ cho phép theo dõi từ khóa, so sánh với đối thủ, kiểm tra backlinks, và cung cấp phân tích chi tiết về trang web của mình.
* Moz Pro: cung cấp nhiều công cụ tối ưu hóa SEO, bao gồm theo dõi từ khóa, phân tích backlinks, kiểm tra lỗi trang web, và phân tích kết quả tìm kiếm…

Khi bạn sử dụng các công cụ này, ta có thể kiểm tra và cải thiện các khía cạnh khác nhau của trang web để đảm bảo nó đáp ứng tốt hơn yêu cầu của công cụ tìm kiếm và cung cấp trải nghiệm tốt cho người dùng.

1. **Quản lý phiên bản và Git**
   1. **Giới thiệu về hệ thống quản lý mã nguồn và Git**

Hệ thống quản lý mã nguồn (Source Code Management System hay SCM) là một công cụ quan trọng trong quá trình phát triển phần mềm để theo dõi, quản lý và kiểm soát mã nguồn.

Trong lĩnh vực phát triển phần mềm, có nhiều SCM khác nhau, nhưng Git là một trong những hệ thống quản lý mã nguồn phổ biến nhất và mạnh mẽ. Dưới đây là một giới thiệu về Git và hệ thống quản lý mã nguồn.

Git là một hệ thống quản lý mã nguồn phân phối (distributed source code management system) được phát triển bởi Linus Torvalds vào năm 2005. Git được sử dụng rộng rãi trong việc quản lý mã nguồn mở và dự án phần mềm thương mại.

Hoạt động trên nguyên tắc phân phối (distributed), nghĩa là mỗi cá nhân hoặc nhóm làm việc với một bản sao đầy đủ của toàn bộ kho mã nguồn. Điều này cho phép làm việc ngoại tuyến, tạo ra nhánh (branch) riêng để phát triển tính năng hoặc sửa lỗi, và sau đó hợp nhất (merge) các thay đổi với nhau.

Git sử dụng cơ chế xác định (hash) để theo dõi sự thay đổi của tệp tin và cung cấp quyền kiểm soát và theo dõi chặt chẽ.

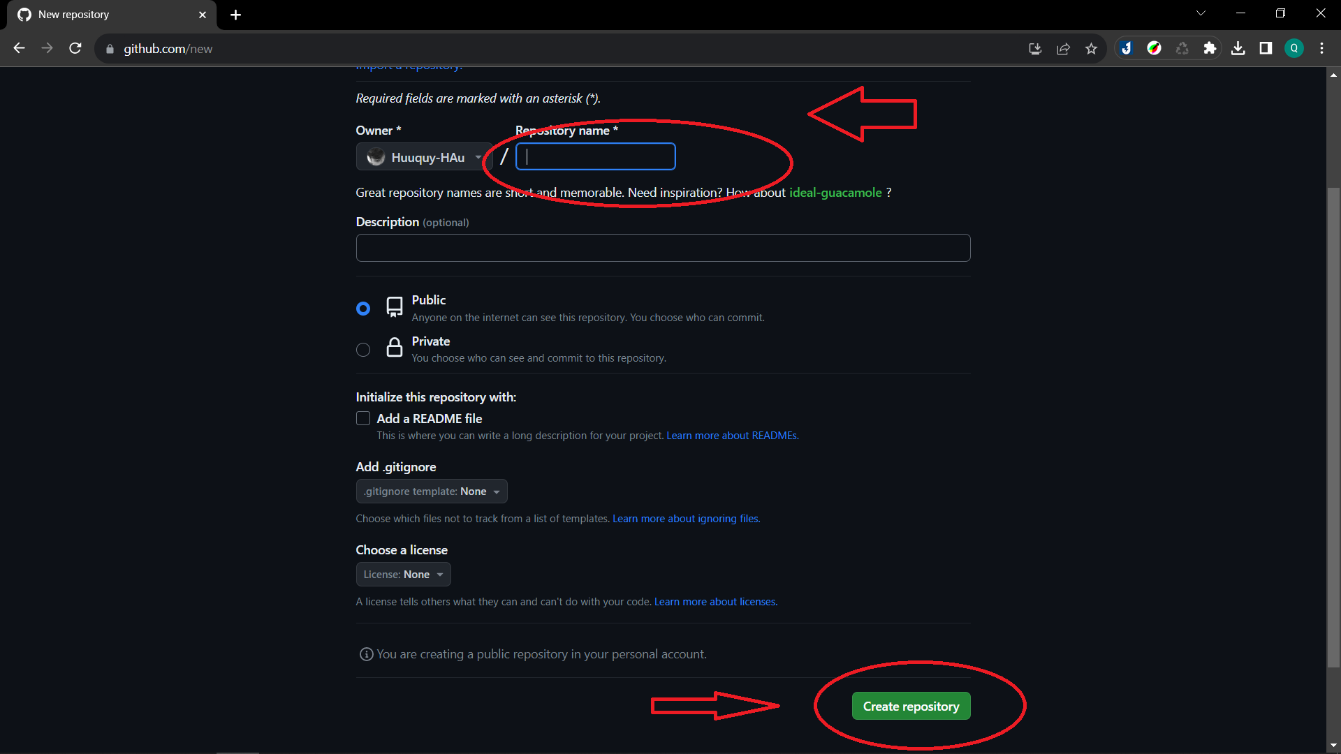
Các khái niệm cơ bản của Git:

* Repository (Kho): Kho Git là nơi lưu trữ tất cả tệp tin và lịch sử của dự án.
* Commit: Commit là một "chốt" của mã nguồn tại một thời điểm cụ thể. Mỗi commit bao gồm các thay đổi trong tệp tin, thông điệp mô tả thay đổi và một mã xác định (hash).
* Branch (Nhánh): Branch là một phiên bản riêng biệt của kho mã nguồn. Nhánh cho phép phát triển và thử nghiệm tính năng mà không ảnh hưởng đến nhánh chính.
* Merge: Merge là quá trình hợp nhất các thay đổi từ một nhánh vào nhánh khác.
* Pull Request (Yêu cầu kéo): Một yêu cầu kéo là cách để đề xuất thay đổi từ một nhánh đến nhánh chính và yêu cầu người khác xem xét và chấp nhận thay đổi.
* Clone: Clone là việc tạo một bản sao của kho mã nguồn từ một kho trên máy chủ từ xa.
* Push và Pull: Push là việc đẩy thay đổi từ máy tính cá nhân lên kho chính, trong khi pull là việc kéo thay đổi từ kho chính về máy tính cá nhân.
  1. **Hướng dẫn tạo kho lưu trữ (reponsitory) Git và làm việc với các lệnh cơ bản như git clone, git pull, git push và git commit**
* Bước 1: Tại trang chủ github của bạn , hãy tạo cho mình 1 dự án (**reponsitory**) mới :

A screenshot of a computer

Description automatically generated

* Bước 2 : Điền các thông tin cần thiết của dự án như tên , mô tả vào tiến hành tạo dự án mới :



* Bước 3: Sau khi tạo dự án thành công ta quay về dự án tại máy tính của chúng ta và tiến hành các lệnh như sau :

**-Git init** :Lệnh này tạo ra một kho lưu trữ Git mới trong thư mục làm việc hiện tại. Khi bạn thực hiện lệnh này, Git sẽ bắt đầu theo dõi tất cả các tệp trong thư mục và các thay đổi trong các tệp này.

-**git remote add origin** [**https://github.com/Huuquy-HAu/Project\_test-github.git**](https://github.com/Huuquy-HAu/Project_test-github.git)

Tạo liên kết với kho lưu trữ reponsitory git của bạn

-git add . : Nó sẽ thêm tất cả các tệp đã sửa đổi hoặc mới được tạo ra vào danh sách chờ (staging area) để sau đó bạn có thể commit chúng.

-**git commit -m “String”:** Lệnh này thực hiện việc tạo một commit với thông điệp "-m" được đặt trong dấu ngoặc kép. Commit là một bản ghi lịch sử của các thay đổi đã thêm vào danh sách chờ ở bước trước. Trong trường hợp này, thông điệp commit là "String" để mô tả mục đích của commit.

-**git push origin master :**Lệnh này đẩy (push) các commit từ kho lưu trữ cục bộ lên kho lưu trữ từ xa trên GitHub (tên từ xa là "origin"). "master" là tên của nhánh mà bạn muốn đẩy lên. Điều này có nghĩa rằng bạn đang đẩy các thay đổi của bạn từ nhánh "master" cục bộ lên kho lưu trữ từ xa "origin".

- **git clone <url\_kho\_luu\_tru\_tu\_xa> :** Git sẽ tải toàn bộ kho lưu trữ từ xa, bao gồm lịch sử và các nhánh. Sau khi hoàn thành, bạn sẽ có một bản sao của kho lưu trữ đó trên máy tính của bạn và bạn có thể bắt đầu làm việc trên nó. Các thay đổi mà bạn thực hiện trong thư mục này sau đó có thể được commit và đẩy (push) lên kho lưu trữ từ xa để chia sẻ với người khác.

-**git pull :** cập nhật mã nguồn của bạn từ kho lưu trữ từ xa (thường là một kho lưu trữ trên GitHub hoặc GitLab) và hợp nhất (merge) các thay đổi đó vào nhánh cục bộ của bạn