Trường Đại Học Kiến Trúc Hà Nội

Khoa Công Nghệ Thông Tin

# 

Môn: Công nghệ WEB

Shop bán điện thoại

Giáo viên hướng dẫn: Nguyễn Thị Hạnh

Thành viên nhóm 19:

1. Vũ Thanh Tùng
2. Vũ Minh Thưởng

Năm - 2023

## I.Khái niệm

### 1.1. HTML

HTML (HyperText Markup Language) là ngôn ngữ đánh dấu sử dụng để tạo và định dạng nội dung trên trang web. Nó là một phần quan trọng của phát triển web và cho phép bạn tạo cấu trúc trang web bằng cách sử dụng các thẻ và các phần tử khác.

- Sau đây là các bước cơ bản để làm quen với html

+ Tạo một tệp HTML cơ bản

Để bắt đầu tạo một trang web HTML cơ bản, bạn cần một trình soạn thảo văn bản như Notepad, Visual Studio Code, hoặc bất kỳ trình soạn thảo nào bạn ưa thích. Sau đó, bạn tạo một tệp văn bản mới và lưu nó với phần mở rộng .html. Ví dụ: index.html.

+ Đây là ví dụ về cấu trúc cơ bản của html



Ở đây có những điều sau đây:

<!DOCTYPE html>—  . Đó là một lời mở đầu cần thiết.

<html></html>- [<html>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/html) phần tử. Phần tử này bao bọc tất cả nội dung trên toàn bộ trang và đôi khi được gọi là phần tử gốc. Nó cũng bao gồm lang thuộc tính, thiết lập ngôn ngữ chính của tài liệu.

<head></head>- [<head>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/head) phần tử. Phần tử này hoạt động như một nơi chứa tất cả nội dung muốn đưa vào trang HTML không phải là nội dung bạn đang hiển thị cho người xem trang của mình. Điều này bao gồm những thứ như [từ khóa](https://developer.mozilla.org/en-US/docs/Glossary/Keyword) và mô tả trang mà bạn muốn xuất hiện trong kết quả tìm kiếm, CSS để tạo kiểu cho nội dung của chúng tôi, khai báo bộ ký tự, v.v.

<meta charset="utf-8">— Phần tử này đặt bộ ký tự nên sử dụng thành UTF-8, bao gồm hầu hết các ký tự từ phần lớn các ngôn ngữ viết. Về cơ bản, giờ đây nó có thể xử lý bất kỳ nội dung văn bản nào bạn có thể đưa vào. Không có lý do gì để không thiết lập điều này và nó có thể giúp tránh một số vấn đề sau này.

<meta name="viewport" content="width=device-width">— [Phần tử khung nhìn](https://developer.mozilla.org/en-US/docs/Web/CSS/Viewport_concepts" \l "mobile_viewports) này đảm bảo trang hiển thị ở độ rộng của khung nhìn, ngăn trình duyệt trên thiết bị di động hiển thị các trang rộng hơn khung nhìn và sau đó thu nhỏ chúng xuống.

<title></title>- [<title>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/title)phần tử. Điều này đặt tiêu đề cho trang, là tiêu đề xuất hiện trong tab trình duyệt mà trang được tải vào. Nó cũng được sử dụng để mô tả trang khi bạn đánh dấu trang/yêu thích nó.

<body></body>- [<body>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/body)phần tử. Phần này chứa tất cả nội dung muốn hiển thị cho người dùng web khi họ truy cập trang của bạn, cho dù đó là văn bản, hình ảnh, video, trò chơi, bản âm thanh có thể phát hay bất kỳ nội dung nào khác.

Mở tệp HTML trên trình duyệt web bằng cách kéo và thả nó vào trình duyệt hoặc bằng cách nhấp đúp vào tệp.

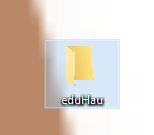
Với bài báo cáo này chúng em không chỉ muốn giới thiệu và hướng dẫn cách tạo 1 file html mà muốn tạo 1 project theo hướng chuyên nghiệp để dễ quản lý hơn bằng công cụ viết code Visual studio code

#### 1.1.1. Các bước khâu chuẩn bị:

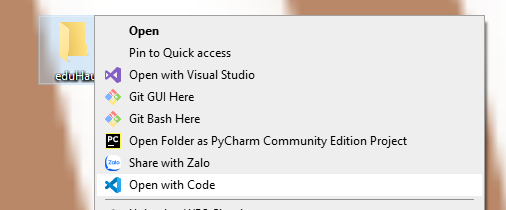
- Cài đặt Visual studio code , cài thêm các tool hỗ trợ trong phần Extensions như : Live Sever,elins,…

- Các bước tạo (có nhiều cách nhưng cách này là cách thông dụng nhất) :

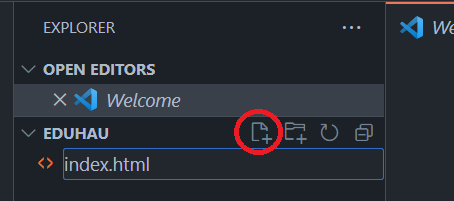
+ B1 : Tạo folder có tên theo project của mình



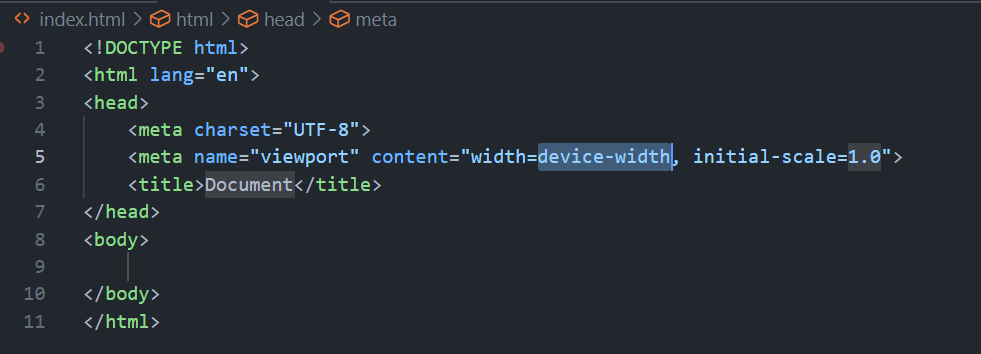
+ B2: Chuột phải open with Code



+ B3:Click vào biểu tượng thêm file và tạo một file .html là ta đã có 1 file html rồi

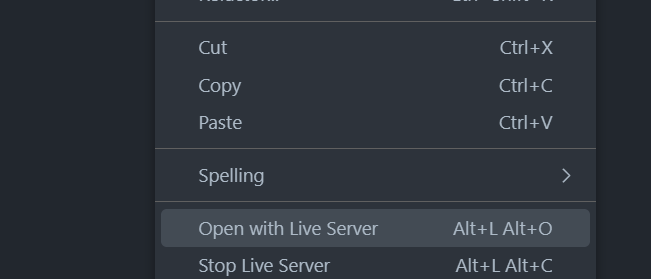


+ B4: Có thể viết bằng tay hoặc sử dụng tool có sẵn của visual code (ctrl +!) để có được một cấu trúc cơ bản của một file html



Cách chạy file html thông dụng, phổ biến nhất với live sever (trên visual code) :

Chuột phải chọn open with live serve



Ngoài ra trang web cơ bản sẽ hiển thị trên trình duyệt và có nội dung như trong đoạn mã HTML bạn đã tạo. Bạn có thể tùy chỉnh và mở rộng trang web bằng cách thêm thẻ HTML, CSS và JavaScript khác để làm cho nó trở nên phức tạp hơn và hấp dẫn hơn.

Vậy các thẻ cơ bản của html và cách dùng :

Thẻ <html>: Định nghĩa toàn bộ tài liệu HTML.

Thẻ <head>: Chứa thông tin về tài liệu như tiêu đề trang, ký tự mã hóa, và các liên kết tới các tệp CSS hoặc JavaScript.

Thẻ <title>: Định nghĩa tiêu đề của trang web, hiển thị trên thanh tiêu đề của trình duyệt.

Thẻ <meta>: Để cung cấp thông tin về trang web như mã ký tự, tác giả, và mô tả.

Thẻ <link>: Liên kết tới các tệp CSS hoặc các tài liệu khác.

Ví dụ link với file style.css

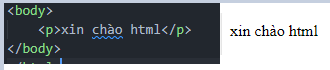


Thẻ <body>: Chứa nội dung chính của trang web như văn bản, hình ảnh, và các phần tử khác.

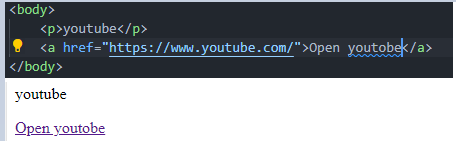
Thẻ <h1>, <h2>, <h3>, ... <h6>: Để tạo tiêu đề với độ ưu tiên giảm dần. <h1> là tiêu đề quan trọng nhất và <h6> là tiêu đề ít quan trọng nhất.



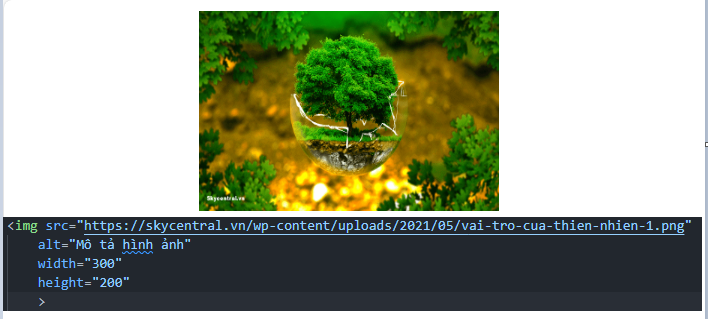
Thẻ <p>: Để định dạng đoạn văn bản.



Thẻ <a>: Để tạo liên kết (hyperlink) đến trang web khác hoặc các tài liệu trong trang web hiện tại.

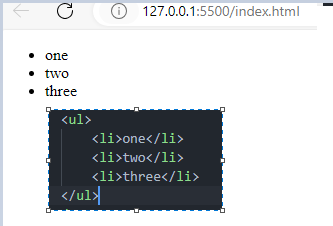


Thẻ <img>: Để hiển thị hình ảnh.

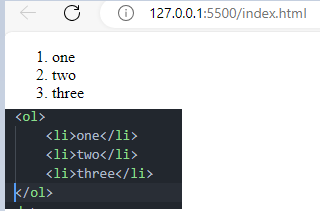


Thẻ <li>: Để tạo một mục trong danh sách.(nên kết hợp với <ol>|<ul>)

Thẻ <ul>: Để tạo danh sách không sắp xếp (unordered list).



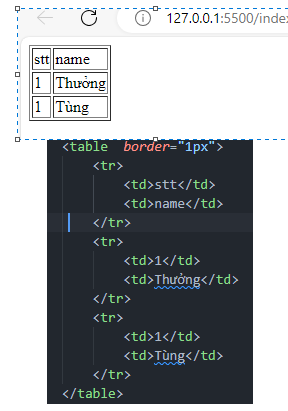
Thẻ <ol>: Để tạo danh sách có sắp xếp (ordered list).



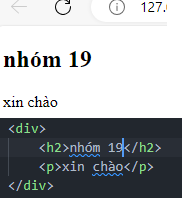
Thẻ <table>: Để tạo bảng.

Thẻ <tr>: Để tạo hàng trong bảng

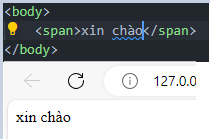
Thẻ <td>: Để tạo ô dữ liệu trong bảng.



Thẻ <div>: Được sử dụng để nhóm và định dạng phần của trang web.



Thẻ <span>: Được sử dụng để định dạng một phần nhỏ của văn bản hoặc inline element.

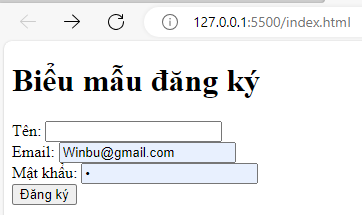


Thẻ <form>: Được sử dụng để tạo biểu mẫu để người dùng nhập thông tin(thường gồm các thẻ lấy dữ liệu từ người dùng).

Thẻ <input>: Được sử dụng trong biểu mẫu để tạo các ô nhập dữ liệu.

Thẻ <button>: Được sử dụng để tạo nút.





Trong ví dụ này:

Chúng ta đã tạo một biểu mẫu sử dụng thẻ <form>. Thuộc tính action xác định URL mà biểu mẫu sẽ gửi dữ liệu đến khi người dùng nhấn nút "Đăng ký". Thuộc tính method định nghĩa phương thức gửi dữ liệu, ở đây là "post".

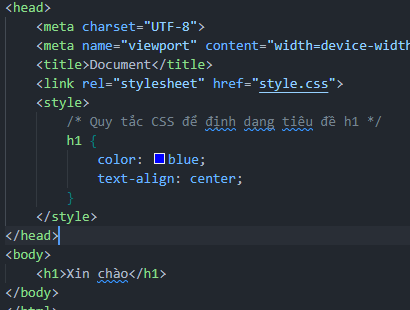
Để cho phép người dùng nhập thông tin, chúng ta đã sử dụng các thẻ <input> với các loại khác nhau như text, email, và password. Thuộc tính name xác định tên của trường dữ liệu, và thuộc tính id được sử dụng để liên kết nhãn (<label>) với ô nhập dữ liệu.

Để làm cho các trường bắt buộc, chúng ta đã sử dụng thuộc tính required.

Cuối cùng, chúng ta có một nút "Đăng ký" để gửi biểu mẫu

Khi người dùng điền thông tin và nhấn nút "Đăng ký", dữ liệu biểu mẫu sẽ được gửi đến URL được xác định trong thuộc tính action. Bạn cần có một máy chủ hoặc script phía máy chủ để xử lý dữ liệu biểu mẫu sau khi nó được gửi đi.

Thẻ <style>: Được sử dụng để đặt các quy tắc CSS trực tiếp trong tài liệu HTML.



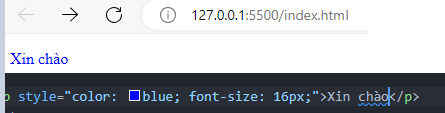
Thẻ <script>: Được sử dụng để chèn mã JavaScript vào trang web.

### 1.2. CSS

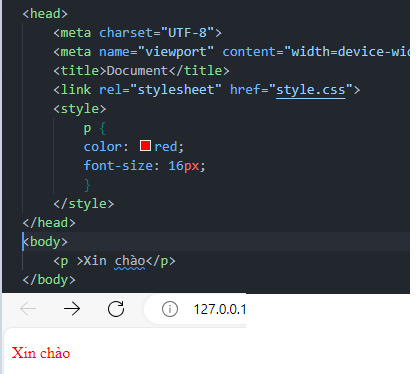
CSS là viết tắt của "Cascading Style Sheets" (Bảng điều khiển kiểu dáng độc lập). Nó là một ngôn ngữ lập trình được sử dụng trong phát triển web để định dạng và kiểu dáng trang web. CSS cho phép bạn xác định cách mà các phần tử HTML trên trang web của bạn sẽ được hiển thị trên trình duyệt, bao gồm màu sắc, kích thước, khoảng cách, font chữ, và nhiều thuộc tính khác.

#### 1.2.1. Áp dụng css vào html

##### a) Cách 1 : viết inline

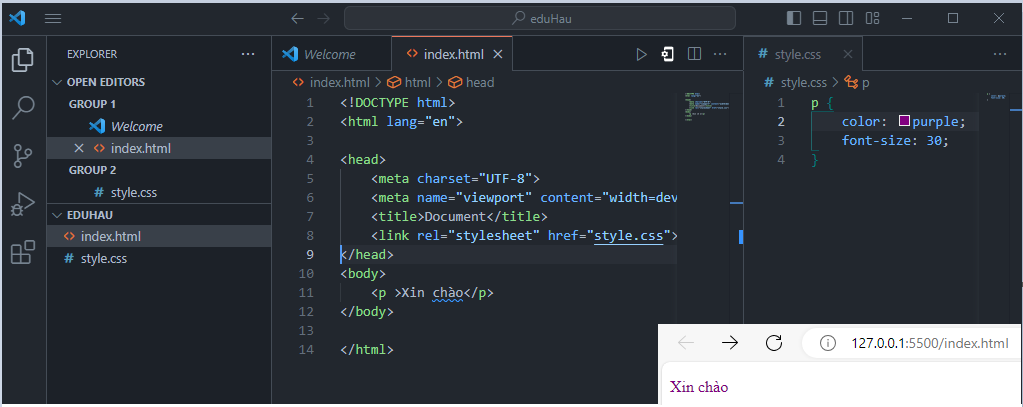


##### b) Cách 2 : Internal



##### c) Cách 3 : External

Tạo file .css (viết css bên trong ) rồi dùng thẻ <link> để liên kết giữa file html và css



#### 1.2.2. Một số thuộc tính cơ bản css

##### a) Color

:+ color: Màu sắc nội dung trong phần tử

+ background-color: Màu sắc nền phần tử

##### b) CSS Backgrounds

+ background-color: Màu sắc nền

+ background-image: Hình ảnh nền.

'+ background-repeat: Lặp lại trong nền

'+ background-attachment: Loại hiển thị nền

+ background-position: Vị trí hiển thị trong nền

+ background (shorthand property): Nền

##### c) CSS Borders

'+ borderwidth: Độ rộng viền

+ border-color: Màu sắc viền

+ borderstyle: : Kiểu viền

+ bordertop-style: Kiểu viền trên

+ border-right-style: Kiểu viền phải

+ border-bottom-style: Kiểu viền dưới

+ Border-left-style: Kiểu viền trái

+ border-radius: Kiểu viền bo tròn

+ border (style required): Kiểu viền

##### d) CSS Margins

+ margin-bottom; Lề dưới

+ margin-left: Lề trái

+margin-right: lề phải

+margin-top: lề trên

+ margin: Canh lề

##### e) CSS Padding

+ padding-top: Đệm/lót trên

+ padding-right: Đệm/lót phải

+ padding-bottom: Đệm/lót dưới

+ padding-left: Đệm/lót trái

+padding: đệm

##### f) CSS Height, Width

+height : chiều cao phần tử

+ width: Chiều rộng phần tử.

+ max-height: Chiều cao tối đa

+ max-width: Chiều rộng tối đa

+min-height: Chiều cao tối thiểu

+min-width: Chiều rộng tối thiểu

##### g) CSS Box Model

+ Content - Nội dung

+ Padding - Phần đệm quan nội dung

+ Border - Viền quanh vùng đệm

+ Margin - Vùng ngoài cùng

##### h) CSS Text

+ color: Màu nội dung

+ Text Alignment

+ text-align: Hướng canh lề

+ text-align-last: Hướng canh lề

+ direction: Hướng hiển thị

'+ unicode-bidi: Hướng hiển thị

+ vertical-align: Canh chỉnh chiều dọc

+ Text Decoration

+ text-decoration-line: Trang trí dòng

+text-decoration-color : trang trí màu

+ text-decoration-style: Trang trí kiểu

+ text-decoration-thickness: Trang trí độ dày mỏng

+ text-decoration: Trang trí

+ Text Transformation

+ text-transform: Thay đổi text

+ Text Spacing

+ text-indent: Thụt lề

+ letter-spacing: Khoảng cách các kí tự

+ line-height: Khoảng cách các dòng

+ word-spacing: Khoảng cách các từ.

+ white-space: Khoảng trắng

+ Text Shadow

+ text-shadow: Bộ bóng của text

##### i) CSS Fonts

+ font-family: Chỉ định font chữ cho text

##### j) CSS cons

+ Sử dụng thẻ nội tuyến (ví dụ <ï> hoặc <span>)

##### k) CSS Table

+ Table Border

##### l) CSS Height, Width

+ width: Chiều rộng phần tử.

+ maxcheight: Chiều cao tối đa

+ max-width: Chiều rộng tối đa

##### k) CSS Box Model

+ Content - Nội dung

+ Padding - Phần đệm quan nội dung

+ Border - Viền quanh vùng đệm

+ Margin - Vùng ngoài cùng

##### Thực nghiệm:



### 1.3. JavaScript

JavaScript (JS) là một ngôn ngữ lập trình thông dịch (interpreted) và đa năng, thường được sử dụng cho phía máy khách (client-side) để tạo ra các ứng dụng web tương tác và động.

- Dưới đây là một số sự kiện phổ biến của các phần tử HTML để kết hợp với js :

+ click: Sự kiện này xảy ra khi người dùng nhấp chuột lên một phần tử, chẳng hạn như nút hoặc liên kết.

+ submit: Xảy ra khi người dùng gửi một biểu mẫu (form) bằng cách nhấn nút "Gửi" hoặc tương tự.

+ change: Sự kiện này xảy ra khi giá trị của một phần tử đầu vào (input) thay đổi, chẳng hạn như khi chọn một tùy chọn từ danh sách thả xuống (dropdown).

+ keydown, keyup: Xảy ra khi người dùng nhấn phím xuống và nhả phím lên, tương ứng.

+ mouseover, mouseout: Xảy ra khi chuột đi vào và rời khỏi một phần tử.

+ mouseenter, mouseleave: Tương tự như mouseover và mouseout, nhưng không kích hoạt cho các phần tử con bên trong.

+ focus, blur: Xảy ra khi một phần tử nhận hoặc mất focus, thường được sử dụng cho các phần tử đầu vào.

+ load: Xảy ra khi một tài nguyên (như hình ảnh hoặc video) đã tải xong.

+ error: Xảy ra khi xảy ra lỗi trong quá trình tải tài nguyên.

+ scroll: Xảy ra khi người dùng cuộn trang web.

+ resize: Xảy ra khi cửa sổ trình duyệt thay đổi kích thước.

+ contextmenu: Xảy ra khi người dùng nhấp chuột phải, hiển thị menu ngữ cảnh (context menu).

+ dblclick: Xảy ra khi người dùng nhấp đúp chuột lên một phần tử.

Dưới đây là một số khái niệm và ví dụ cơ bản về cú pháp JavaScript:

**1.3.1 Biến (Variables):**

+JavaScript sử dụng từ khóa var, let, hoặc const để khai báo biến. Ví dụ:

var x = 10; // Sử dụng var để khai báo biến let

y = 5; // Sử dụng let để khai báo biến (có thể thay đổi giá trị)

const z = 20; // Sử dụng const để khai báo biến (không thể thay đổi giá trị sau khi gán)

**1.3.2 Hàm (Functions):**

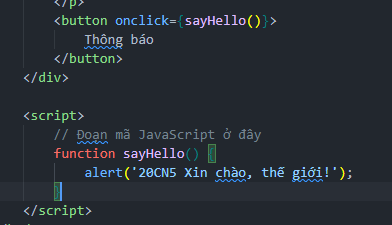
Hàm trong JavaScript được định nghĩa bằng từ khóa function. Ví dụ:



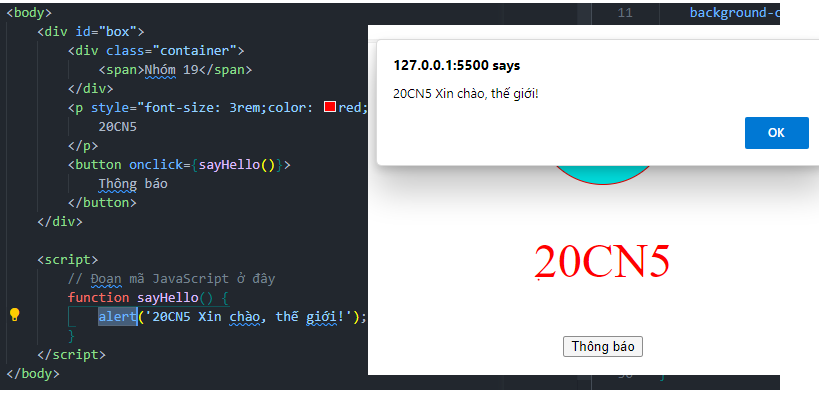
**1.3.3 Sự kiện (Events):**

JavaScript có thể sử dụng để xử lý sự kiện trên trang web.

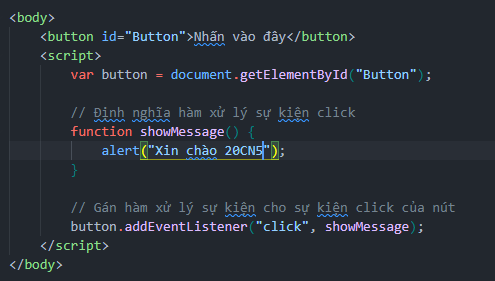
Ví dụ dưới đây là một ví dụ về cách sử dụng sự kiện click để hiển thị thông báo:



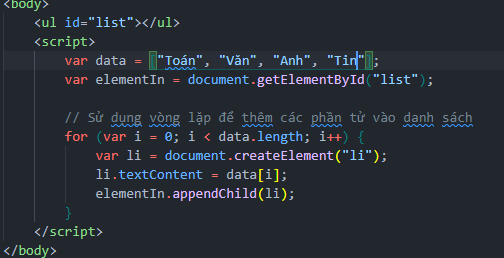
Kết quả



Ngoài ra còn :



Kết hợp với vòng lặp for để tạo nhiều elemet



**1.4. Quy trình phát triển web**

Quy trình phát triển web (web development process) là chuỗi các bước và hoạt động mà một nhóm phát triển web hoặc một cá nhân thực hiện để xây dựng một trang web hoặc ứng dụng web. Quy trình này có thể thay đổi tùy theo dự án cụ thể và phương pháp phát triển được sử dụng, nhưng dưới đây là một quy trình phát triển web cơ bản:

**1.4.1. Lập kế hoạch và phân tích yêu cầu:**

Lập kế hoạch dự án (Project Planning): Đầu tiên, quản lý dự án (Project Manager) hoặc quản lý sản phẩm (Product Manager) phải xác định mục tiêu, phạm vi và nguồn lực cần thiết cho dự án. Lên kế hoạch về lịch trình, ngân sách và tài nguyên.

Phân tích yêu cầu (Requirements Analysis): Người phân tích yêu cầu (Business Analyst) hoặc nhóm phân tích yêu cầu phải tương tác với khách hàng và người dùng cuối để hiểu rõ yêu cầu của họ. Họ tạo tài liệu yêu cầu chi tiết, định nghĩa tính năng và chức năng của trang web

**1.4.2. Thiết kế giao diện và trải nghiệm người dùng:**

Thiết kế giao diện (UI/UX Design): Nhóm thiết kế giao diện người dùng (UI Designer) và chuyên gia trải nghiệm người dùng (UX Designer) làm việc để tạo wireframes, mockups và thiết kế giao diện dựa trên yêu cầu và nhu cầu của người dùng. Họ đảm bảo trải nghiệm người dùng mượt mà và thú vị.

**1.4.3. . Phát triển:**

Lập trình (Development): Nhóm lập trình viên (Developers) xây dựng trang web bằng sử dụng các ngôn ngữ lập trình như HTML, CSS, JavaScript, và các framework hoặc thư viện. Họ triển khai các tính năng và chức năng theo yêu cầu.

Quản lý cơ sở dữ liệu (Database Management): Nếu có cơ sở dữ liệu, người quản lý cơ sở dữ liệu (Database Administrator) xây dựng và quản lý cơ sở dữ liệu, bao gồm việc thiết kế cơ sở dữ liệu, quản lý dữ liệu, và tối ưu hóa truy vấn.

**1.4.4.Kiểm thử:**Kiểm thử chức năng (Functional Testing): Nhóm kiểm thử (Testers) kiểm tra từng tính năng để đảm bảo rằng chúng hoạt động đúng cách và không có lỗi.

Kiểm thử tương tác người dùng (User Interaction Testing): Kiểm tra giao diện và trải nghiệm người dùng để đảm bảo tính tương tác và thú vị.

Kiểm thử tích hợp (Integration Testing): Kiểm tra tích hợp giữa các phần của trang web để đảm bảo chúng hoạt động cùng nhau.

**1.4.5. Tối ưu hóa và hiệu chỉnh:**

Tối ưu hóa hiệu suất (Performance Optimization): Nhóm phát triển và quản lý cơ sở dữ liệu cần tối ưu hóa hiệu suất của trang web, bao gồm tải nhanh, xử lý tải lớn và tối ưu hóa cơ sở dữ liệu.

Hiệu chỉnh và điều chỉnh (Refinement and Adjustments): Dựa trên phản hồi từ quá trình kiểm thử, các thay đổi và điều chỉnh cuối cùng được thực hiện để cải thiện trang web và sửa các lỗi.

**1.4.6.. Triển khai và vận hành:**

Quản lý dự án: Triển khai trực tiếp trang web lên máy chủ hoặc nền tảng hosting.

Kiểm thử viên: Kiểm tra cuối cùng để đảm bảo trang web đã triển khai hoạt động đúng trên môi trường sản phẩm.

Vai trò quản lý và lãnh đạo:

Quản lý dự án (Project Manager): Quản lý và điều hành toàn bộ quy trình phát triển, đảm bảo rằng dự án được hoàn thành đúng tiến độ và ngân sách.

Vai trò thiết kế:

Thiết kế giao diện (UI Designer): Tạo giao diện đẹp mắt và hấp dẫn cho trang web.

Chuyên gia trải nghiệm người dùng (UX Designer): Đảm bảo trải nghiệm người dùng mượt mà và thú vị.

Vai trò phát triển:

Lập trình viên (Front-end và Back-end Developers): Xây dựng và triển khai trang web theo yêu cầu.

Quản lý cơ sở dữ liệu (Database Administrator): Xây dựng và quản lý cơ sở dữ liệu liên quan đến trang web.

Vai trò kiểm thử:

Kiểm thử viên (Testers): Kiểm tra và đảm bảo chất lượng của trang web bằng cách thực hiện ki

#### 1.4.7. Bảo trì và cải tiến (Maintenance and Enhancement)

+ Theo dõi và duy trì trang web để đảm bảo tính ổn định và hiệu suất.

+ Thực hiện các cải tiến, bảo mật và nâng cấp theo thời gian khi cần thiết.