Shape, square

Description automatically generated**TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

-----🙞🙜🕮🙞🙜-----Logo

Description automatically generated

**BÁO CÁO HỌC PHẦN**

**WEB QUẢN LÝ SINH VIÊN**

**Nhóm 13 – Lớp 20CN5**

***Giảng viên hướng dẫn: Nguyễn Thị Hạnh***

***Sinh viên thực hiện:***

1. **Phạm Đăng Phúc – 2055010203**
2. **Trần Mạnh Hùng – 2055010113**
3. **Nguyễn Tiến Dũng – 2055010041**

Mục lục

[1. HTML 5](#_Toc149122994)

[1.1. Định nghĩa 5](#_Toc149122995)

[1.2. Lịch sử 5](#_Toc149122996)

[1.3. Cấu trúc HTML 5](#_Toc149122997)

[1.3.1. Thẻ mở và đóng 5](#_Toc149122998)

[1.3.2. Phần tử 5](#_Toc149122999)

[1.4. Cú pháp của HTML 5](#_Toc149123000)

[1.4.1. Thẻ HTML 5](#_Toc149123001)

[1.4.2. Thẻ Head 5](#_Toc149123002)

[1.4.3. Thẻ Body 5](#_Toc149123003)

[1.4.4. Thẻ Heading 6](#_Toc149123004)

[1.4.5. Thẻ Paragraph 6](#_Toc149123005)

[1.4.6. Thẻ Hyperlink 6](#_Toc149123006)

[1.4.7. Thẻ Image 6](#_Toc149123007)

[1.4.8. Thẻ List 6](#_Toc149123008)

[1.4.9. Thẻ Table 6](#_Toc149123009)

[1.5. Ứng dụng 6](#_Toc149123010)

[1.6. Ưu điểm và nhược điểm 6](#_Toc149123011)

[1.6.1. Ưu điểm 6](#_Toc149123012)

[1.6.2. Nhược điểm 7](#_Toc149123013)

[1.7. Hướng dẫn cơ bản 7](#_Toc149123014)

[1.7.2. Tạo trang web mẫu với 8](#_Toc149123015)

[1.7.3. Một số thẻ hay dùng 8](#_Toc149123016)

[2. CSS 8](#_Toc149123017)

[2.1. Định nghĩa 8](#_Toc149123018)

[2.2. Lịch sử 9](#_Toc149123019)

[2.3. Cấu trúc của CSS 9](#_Toc149123020)

[2.4. Cú pháp của CSS 9](#_Toc149123021)

[2.5. Các đặc điểm của CSS 9](#_Toc149123022)

[2.6. Ứng dụng của CSS 10](#_Toc149123023)

[2.7. Áp dụng CSS 10](#_Toc149123024)

[2.7.1. Tạo file CSS 10](#_Toc149123025)

[2.7.2. Liên kết file CSS với HTML 10](#_Toc149123026)

[2.7.3. Một số các viết CSS 10](#_Toc149123027)

[2.8. Ví dụ 11](#_Toc149123028)

[3. JavaScript 12](#_Toc149123029)

[3.1. Định nghĩa 12](#_Toc149123030)

[3.2. Lịch sử 12](#_Toc149123031)

[3.3. Đặc điểm 12](#_Toc149123032)

[3.4. Ứng dụng 12](#_Toc149123033)

[3.5. Tạo biến và hàm sử dụng sự kiện 13](#_Toc149123034)

[3.5.1. Sử dụng biến 13](#_Toc149123035)

[3.5.2. Sử dụng hàm 13](#_Toc149123036)

[3.5.3. Sự kiện 13](#_Toc149123037)

[4. HTML Froms và Input Elements. Tạo biểu mẫu HTML 14](#_Toc149123038)

[4.1. Định nghĩa 14](#_Toc149123039)

[4.2. Phần tử trong HTML Forms 14](#_Toc149123040)

[4.3. Phần tử biểu mẫu 14](#_Toc149123041)

[4.3.1. Input Elements (Phần tử đầu vào) 14](#_Toc149123042)

[4.3.2. Text Fields (trường văn bản) 15](#_Toc149123043)

[4.3.3. Phần tử <label> 15](#_Toc149123044)

[4.3.4. Radio Buttons (Nút Radio) 16](#_Toc149123045)

[4.3.5. Check box 16](#_Toc149123046)

[4.4. CSS Forms (biểu mẫu CSS) 16](#_Toc149123047)

[4.5. Sử dụng JS 17](#_Toc149123048)

[4.5.1. Xác thực biểu mẫu JS 17](#_Toc149123049)

[4.5.2. Xác thực biểu mẫu HTML tự động 18](#_Toc149123050)

[4.5.3. Xác thực dữ liệu 18](#_Toc149123051)

[4.5.4. Xác thực ràng buộc HTML 18](#_Toc149123052)

[4.5.5. Xác thực ràng buộc thuộc tính đầu vào HTML 19](#_Toc149123053)

[4.5.6. Xác thực ràng buộc bộ chọn giả CSS 19](#_Toc149123054)

[5. Quy trình phát triển Web 19](#_Toc149123055)

[5.1. Thu thập thông tin 19](#_Toc149123056)

[5.2. Lập kế hoạch 20](#_Toc149123057)

[5.3. Thiết kế giao diện 20](#_Toc149123058)

[5.4. Viết tài liệu đặc tả 21](#_Toc149123059)

[5.5. Viết mã nguồn 21](#_Toc149123060)

[5.6. Kiểm thử, đánh giá và ra mắt sản phẩm 21](#_Toc149123061)

[5.7. Bảo trì 21](#_Toc149123062)

[6. Xây dựng Web hoàn chỉnh 22](#_Toc149123063)

[6.1. Nodejs 22](#_Toc149123064)

[6.2. Postgresql 22](#_Toc149123065)

[6.3. Giao diện web 23](#_Toc149123066)

[6.3.1. Giao diện đăng nhập: 23](#_Toc149123067)

[6.3.2. Giao diện trang chủ: 24](#_Toc149123068)

[6.3.3. Giao diện điểm thi: 24](#_Toc149123069)

1. HTML
   1. Định nghĩa

HTML (Hyper Text Markup Language) là một ngôn ngữ đánh dấu dùng để tạo và cấu trúc nội dung trên các trang web. Nó sử dụng cú pháp đơn giản và dựa trên các thẻ để xác định cấu trúc và nội dung của trang web. Các thẻ này bao gồm các phần tử và thuộc tính để mô tả và định dạng các phần tử khác nhau trên trang.

* 1. Lịch sử

HTML được tạo ra lần đầu bởi Tim Berners – Lee vào năm 1991 tại CERN (European Organization for Nuclear Research).

Phiên bản đầu tiên, HTML 1.0, được giới thiệu vào năm 1995 và định nghĩ cấu trúc cơ bản của ngôn ngữ HTML.

Các phiên bản sau như HTML 2.0, HTML 3.2, HTML 4.01, XHTML 1.0 và HTML5 đã tiến hóa với thêm tính năng và cải tiến.

* 1. Cấu trúc HTML
     1. Thẻ mở và đóng

Cấu trúc cơ bản của HTML được đánh dấu bằng cặp thẻ mở <tag> và thẻ đóng </tag>.

* + 1. Phần tử

Gồm thẻ mở, nội dung và thẻ đóng, cùng với các thuộc tính để định dạng và quy định hành vi của phần tử đó.

* 1. Cú pháp của HTML
     1. Thẻ HTML

<html> xác định khối gốc của trang web.

* + 1. Thẻ Head

<head> chứa thông tin về trang web như tiêu đề, tạo tin CSS và meta – data.

* + 1. Thẻ Body

<body> chứa nội dung hiển thị trên trang web.

* + 1. Thẻ Heading

<h1>, <h2>, <h3>,...<h6> đánh đấu tiêu đề mới mức độ khác nhau.

* + 1. Thẻ Paragraph

<p> đánh dấu đoạn văn bản.

* + 1. Thẻ Hyperlink

<a> tạo liên kết đến trang web khác.

* + 1. Thẻ Image

<img> hiển thị hình ảnh trên trang.

* + 1. Thẻ List

<ul>, <ol>, <li> tạo danh sách không thứ tự (unordered), danh sách có thứ tự (ordered) và các mục danh sách.

* + 1. Thẻ Table

<table>, <tr> , <td> tạo bảng và các phần tử liên quan.

* 1. Ứng dụng

Xây dựng trang web: HTML là lõi của mọi trang web, cung cấp cấu trúc và nội dung cơ bản cho trang.

Tích hợp với CSS và JavaScript: kết hợp với CSS để tùy chỉnh giao diện người dùng và với JavaScript để thêm tương tác động và các chức năng phức tạp.

Phát triển ứng dụng web: HTML cùng với các công nghệ khác đã sử dụng để phát triển ứng dụng web đa dạng và phức tạp.

* 1. Ưu điểm và nhược điểm
     1. Ưu điểm

Dễ học và sử dụng: HTML có cú pháp đơn giản và dễ hiểu, cho phép người mới học nhanh chóng làm quen và bắt đầu xây dựng trang web cơ bản.

Tương thích đa nên tảng: HTML tương thích với hầu hết các trình duyệt web phổ biến, giúp đảm bảo rằng trang web sẽ hoạt động đúng cách trên nhiều thiết bị và hệ điều hành.

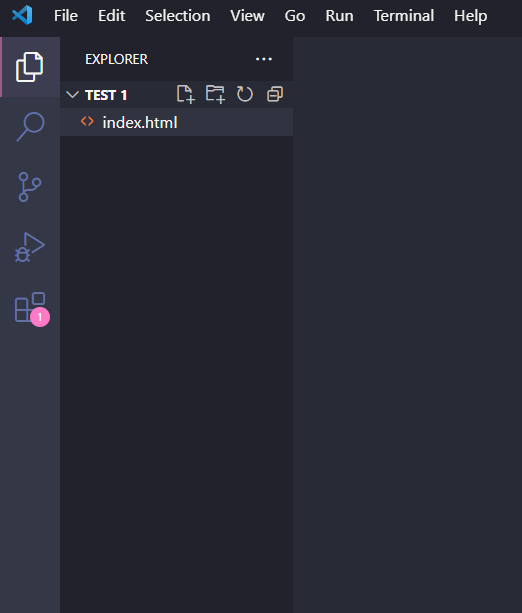
Tiêu chuẩn hóa: HTML tuân thủ các tiêu chuẩn quốc tế, giúp đảm bảo tính nhất quán và khả năng tương thích của các trang web.

* + 1. Nhược điểm

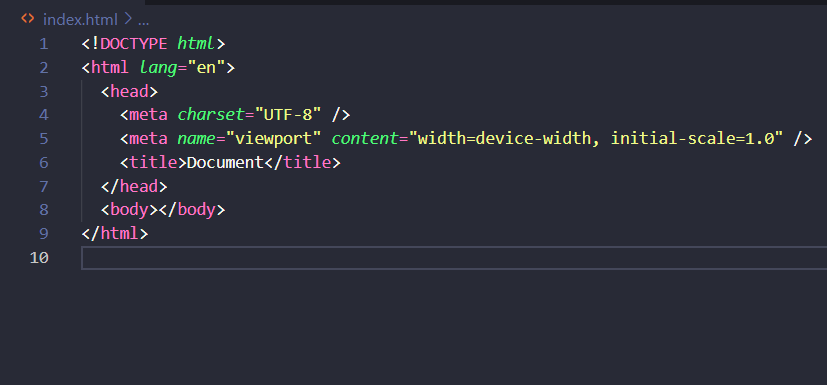
Hạn chế về giao diện người dùng động: HTML ban đầu thiết kế để tạo cấu trúc nội dung tĩnh. Điều này làm cho việc tạo giao diện người dùng động và tương tác phức tạp hơn.

Khả năng mở rộng hạn chế: HTML cơ bản không thể xử lý logic phức tạp và không có khả năng mở rộng lớn mà cần phải phụ thuộc vào các ngôn ngữ lập trình khác như JavaScript để làm điều này.

* 1. Hướng dẫn cơ bản
     1. Tạo 1 file html

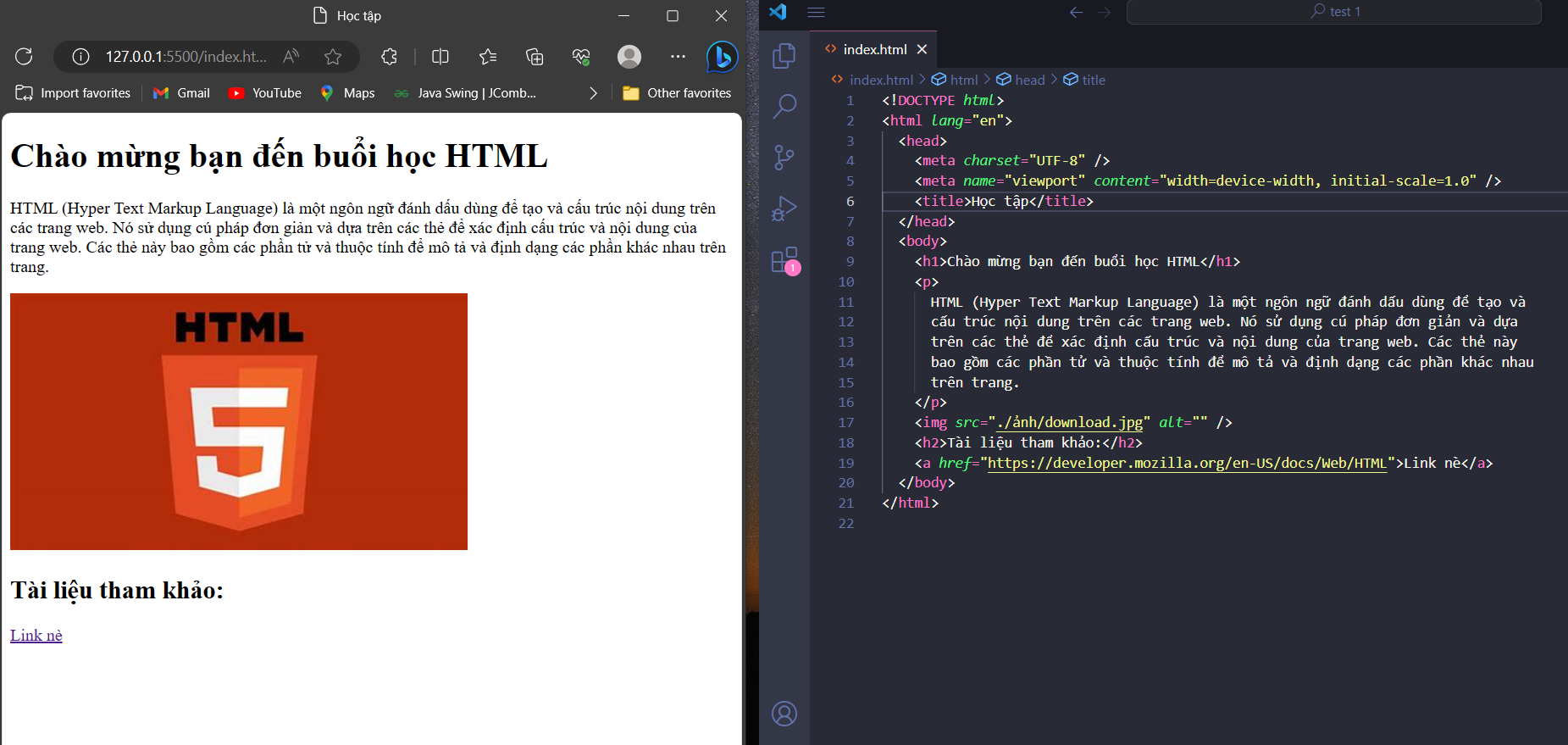
Mở VS code >> mở 1 folder >> tạo 1 file index.html

Mở file index.html >> gỡ “!” >> nhấn Tab để có cấu trúc cơ bản, ấn Go Live ở góc bên phải màn hình (Entension: Live Server để thuận tiện trong quá trình chỉnh sửa).

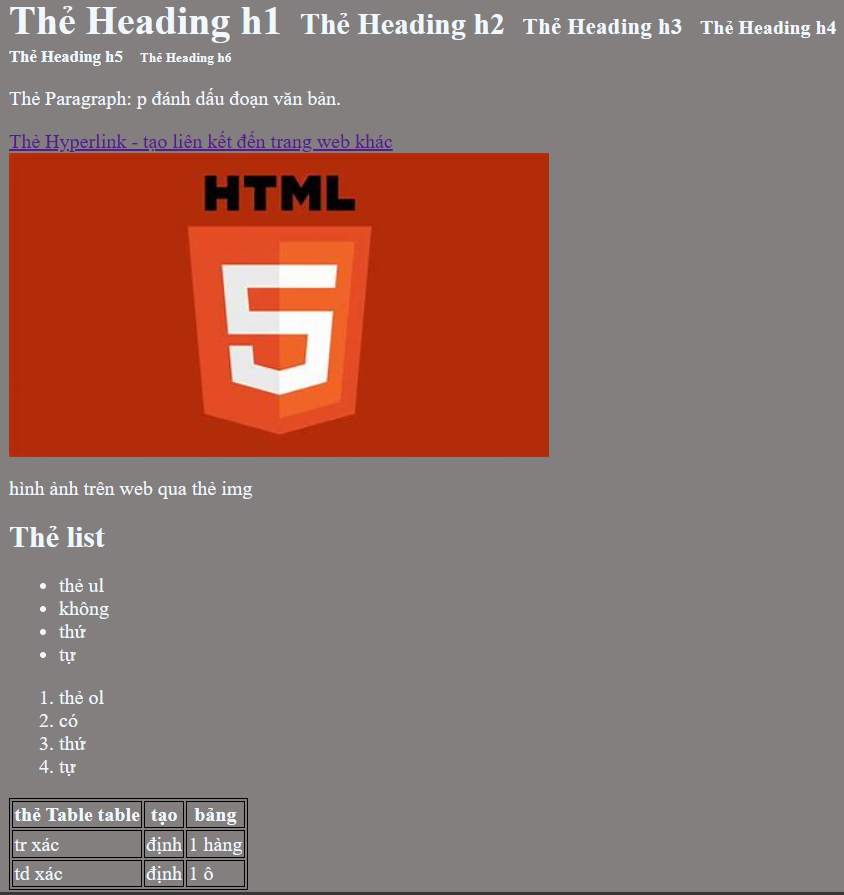


* + 1. Tạo trang web mẫu với

Tiêu đề, đoạn văn bản, hình ăn và siêu liên kết.



* + 1. Một số thẻ hay dùng



1. CSS
   1. Định nghĩa

CSS (Cascading Style Sheets) là một ngôn ngữ đánh dấu được sử dụng để định dạng và trình bày giao diện người dùng (UI) của trang web được viết bằng HTML. CSS cho phép quy định các quy tắc kiểu dáng, bố cục và thiết kế cho các phần tử HTML, giúp tách biệt cấu trúc và giao diện, cũng như tăng tính bảo trì và linh hoạt của trang web.

* 1. Lịch sử

CSS được giới thiệu lần đàu vào năm 1996 bởi W3c (World Wide Web Consortium) để giải quyết vấn đề tách biệt giữa cấu trúc và giao diện của trang web.

CSS1 ra đời năm 1996, CSS2 vào năm 1998 và CSS3 bắt đầu phát triển từ năm 1999.

CSS3 là phiền bản hiện tại và dược phát triển liên tục với nhiều tính năng mới và cải tiến.

* 1. Cấu trúc của CSS

CSS bao gồm các phần chính sau:

Selector (bộ chọn): xác định các phần tử HTML mà muốn áp dụng kiểu.

Properties (thuộc tính): định nghĩa các thuộc tính mà muốn áp dụng cho các phần tử.

Values (giá trị): xác định giá trị của các thuộc tính.

* 1. Cú pháp của CSS

Một quy tắc CSS bao gồm một bộ chọn, một dấu hai chấm và một khối chứa các thuộc tính và giá trị.

Selector {

Property: value;

}

Các quy tắc CSS có thể được nhóm lại thành một tập hợp gọi là tệp CSS để quản lý và duy trì dễ dàng hơn.

* 1. Các đặc điểm của CSS

Tách biệt cấu trúc và giao diện: CSS giúp tách biệt cấu trúc (HTML) và giao diện (CSS) làm cho mã nguồn dễ đọc và dễ bảo trì hơn.

Tính đồng nhất và quy chuẩn hóa: CSS cho phép xác định các kiểu dáng chung và tái sử dụng chúng trên nhiều trang, giúp đạt tính đồng nhất trong giao diện của trang web.

Kiểm soát kiểu dáng linh hoạt: CSS cung cấp nhiều thuộc tính để kiểm soát kiểu dáng, từ màu sắc và font chữ đến kích thước và khoảng cách, mang lại linh hoạt trong thiết kế giao diện.

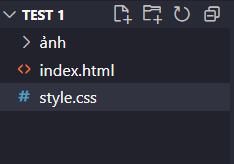
* 1. Ứng dụng của CSS

Thiết kế giao diện web: CSS được sử dụng chủ yếu để thiết kế giao diện trang web, bao gồm màu sắc, font chữ, kích thước, vị trí, khoảng cách và hiệu ứng.

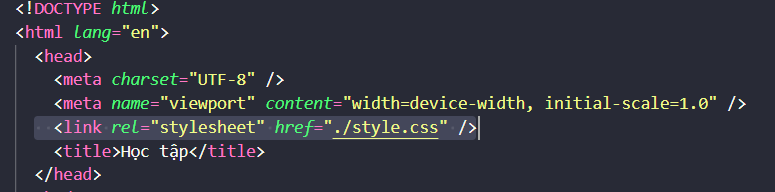
Responsive web design: CSS đóng vai trò quan trọng trong việc tạo ra trang web có thể thích ứng với các kích thước màn hình khác nhau, đảm bảo trải nghiệm tốt trên cả điện thoại di động, máy tính bảng và máy tính.

Tối ưu hóa trải nghiệm người dùng: CSS giúp cải thiện trải nghiệm người dùng bằng cách làm cho trang web trực quan hơn, dễ đọc hơn và tương tác tốt hơn.

* 1. Áp dụng CSS
     1. Tạo file CSS

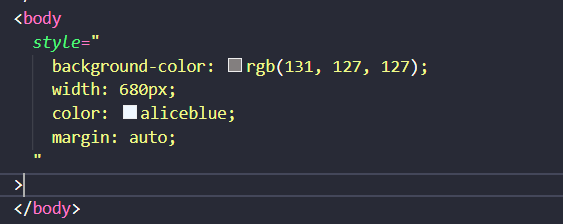


* + 1. Liên kết file CSS với HTML

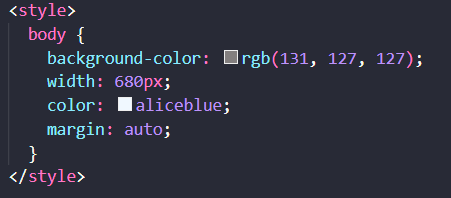


* + 1. Một số các viết CSS

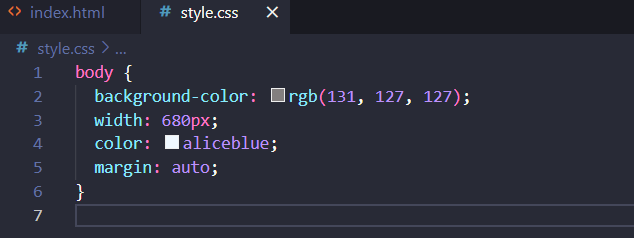
Cách 1: Inline CSS



Cách 2: Internal CSS



Cách 3: External CSS



* 1. Ví dụ



1. JavaScript
   1. Định nghĩa

JavaScript là một ngôn ngữ lập trình thông dịch (interpreted), đa mô hình (multi-paradigm) và dựa trên sự kiện (event-driven). Nó được sử dụng để tạo ra các trang web tương tác, ứng dụng web động, quản lý dữ liệu dựa trên máy khách và nhiều ứng dụng phức tạp khác.

* 1. Lịch sử

JS được tạo ra bởi Brendan Eich tại NetsCape Communications Corporation vào năm 1995.

Ban đầu, nó được thiết kế để tương tác với người dùng trên trình duyệt Netscape Navigator.

ECMAScript là một chuẩn hóa quốc tế định rõ cú pháp, cấu trúc và quy tắc hoạt động của JavaScript. Phiên bản đầu tiên của chuẩn ECMAScript được công bố vào năm 1997.

* 1. Đặc điểm

Được thực thi ở máy khách: JS thường được thực thi trên trình duyệt web của người dùng, giúp tăng tốc độ tải trang và mang lại trải nghiệm tương tác người tốt hơn.

Ngôn ngữ linh hoạt và đa mô hình: JS hỗ trợ nhiều mô hình lập trình bao gồm lập trình hướng đối tượng (OOP), lập trình cấu trúc và lập trình hàm. Điều này cho phép phát triển ứng dụng đa dạng.

Tương tác với DOM: JS có khả năng thao tác và thay đổi cấu trúc và nội dung của trang web thông qua DOM, cho phép tương tác động với người dùng.

Bộ thư viện và framework phong phú: JS có một cộng đồng lớn và đa dạng với nhiều thư viện và framework mạnh mẽ như React, Angular, Vue.js để giúp phát triển ứng dụng phức tạp một cách hiệu quả.

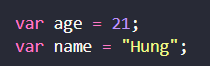
* 1. Ứng dụng

Phát triển ứng dụng Web: JS là một phần thiết yếu trong phát triển của ứng dụng web động, trong web tương tác, ứng dụng di động và ứng dụng đa nền tảng.

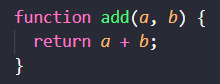
Tích cực với trình duyệt: JS là lựa chọn phổ biến để tạo các tiện ích mở rộng, bổ sung chức năng và tác dộng trong trình duyệt web.

Phát triển game và ứng dụng desktop: JS cũng có thể được sử dụng để phát triển game và ứng dụng desktop thông qua các framework như Electron.

* 1. Tạo biến và hàm sử dụng sự kiện
     1. Sử dụng biến

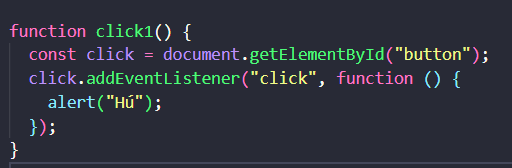


* + 1. Sử dụng hàm

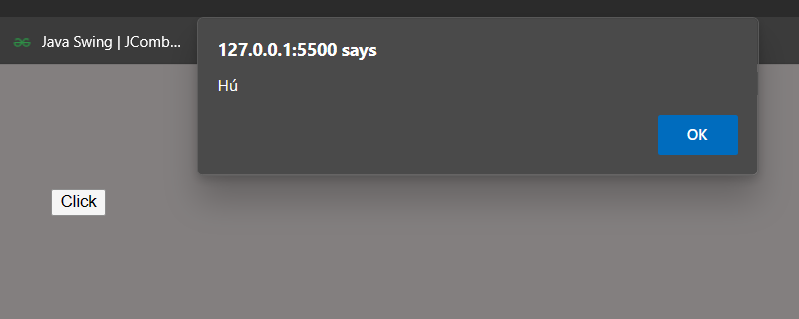


* + 1. Sự kiện





* 1. Ví dụ



1. HTML Froms và Input Elements. Tạo biểu mẫu HTML
   1. Định nghĩa

Biểu mãu HTML (HTML Forms) được sử dụng để thu thập thông tin đầu vào của người dùng. Đầu vào của người dùng thường được gửi đến một máy chủ để xử lý.

* 1. Phần tử trong HTML Forms

Phần tử <form> dùng để tạo biểu mẫu HTML cho người dùng nhập

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Phàn tử là một thùng chứa cho các loại phần tử đầu vào khác nhau, ví dụ như: trường văn bản, hộp kiếm, radio nút, nút gửi,...

* 1. Phần tử biểu mẫu
     1. Input Elements (Phần tử đầu vào)

Phần tử đầu vào được sử dụng nhiều nhất là phần tử <input>. Một phần tử có thể được hiển thị bằng nhiều cách, tùy thuộc vào thuộc tính, dưới đây là một số ví dụ.

Ảnh có chứa văn bản, Phông chữ, số, hàng

Mô tả được tạo tự động

* + 1. Text Fields (trường văn bản)

Một trường đầu vào một dòng cho văn bản <input type= “text”>

Ví dụ: biểu mẫu có các trường nhập liệu cho văn bản

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, hàng, Phông chữ

Mô tả được tạo tự động

* + 1. Phần tử <label>

Thẻ xác định nhãn cho nhiều yều tố hình thức.

Phần tử này hữu ích cho người dùng trình đọc màn hình vì trình đọc màn hình sẽ đọc to nhãn khi người dùng tập trung vào phần tử đầu vào.

Yếu tố này sẽ giúp người dùng khó nhấp vào các vùng rất nhỏ (chẳng hạn như nút Radio hoặc check box) bởi vì khi người dùng nhập vào văn bản trong phần tử, nó sẽ chuyển nút radio/ check box.

Thuộc tính của thẻ nên bằng thuộc tính của phần tử để liên kết chúng lại với nhau.

* + 1. Radio Buttons (Nút Radio)

Nút Radio cho phép người dùng chọn một trong số các lựa chọn giới hạn

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Đây là cách mã HTML trên sẽ được hiển thị trong trình duyệt

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, màu trắng

Mô tả được tạo tự động

* + 1. Check box

Cũng giống như Nút Radio, Check box cho phép người dùng chọn tùy chọn Zero hoặc More của một số lựa chọn giới hạn.

* 1. CSS Forms (biểu mẫu CSS)

Giao diện của một biểu mẫu HTML có thể được cải thiện rất nhiều với CSS.

Các trường nhập liệu kiểu mảng (Styling Input Fields): sử dụng thuộc tính để xác định chiều rộng của trường nhập: width.

Nếu bạn chỉ muốn tạo kiểu cho một loại đầu vào cụ thể, bạn có thể sử dụng bộ chọn thuộc tính:

* Input[type = text]: sẽ chỉ chọn các trường văn bản.
* Input[type = password]: sẽ chỉ chọn các trường mật khẩu.
* Input[type = number]: sẽ chỉ chọn các trường số...
  1. Sử dụng JS
     1. Xác thực biểu mẫu JS

Xác thực biểu mẫu HTML có thể được thực hiện tự động bởi trình duyệt.

Nếu trường biểu mẫu (fname) trống, hàm này sẽ cảnh báo thư và trả về false, để ngăn cản không cho biểu mẫu được gửi.

Ví dụ

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phông chữ

Mô tả được tạo tự động

Hàm có thể được gọi khi biểu mẫu được gửi

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phông chữ

Mô tả được tạo tự động

* + 1. Xác thực biểu mẫu HTML tự động

Xác thực biểu mẫu HTML có thể được thực hiện tự động bởi trình duyệt. Nếu thường biểu mẫu (fname) trống, thuộc tính sẽ ngắn biểu mẫu này gửi: required.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

* + 1. Xác thực dữ liệu

Xác thực dữ liệu là quá trình đảm bảo rằng đầu vào của người dùng sạch sẽ, chính xác và hữu ích.

Các nhiệm vụ xác thực điển hình là:

* Người dùng đã điền vào tất cả các trường bắt buộc chưa?
* Người dùng đã nhập ngày hợp lệ chưa?
* Người dùng đã nhập văn bản vào trường số chưa?

Thông thường, mục đích của xác thực dữ liệu là để đảm bảo đầu vào chính xác của người dùng.

Xác nhận có thể được xác định bằng nhiều phương pháp khác nhau và được triển khai trong nhiều phương pháp những cách khác nhau.

Xác thực phía máy chủ được thực hiện bởi một máy chủ web, sau khi đầu vào được gửi đến máy chủ.

Xác thực phía máy khách được thực hiện bởi trình duyệt web, trước khi đầu vào được gửi đến máy chủ web.

* + 1. Xác thực ràng buộc HTML

HTML5 giới thiệu một khái niệm xác thực HTML mới được gọi là constraint xác nhận.

Xác thực ràng buộc HTML dựa trên:

* Xác thực ràng buộc thuộc tính đầu vào HTML.
* Xác thực ràng buộc thuộc bộ chọn giả CSS.
* Xác thực ràng buộc thuộc tính và phương thức DOM.
  + 1. Xác thực ràng buộc thuộc tính đầu vào HTML

**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động**

* + 1. Xác thực ràng buộc bộ chọn giả CSS

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

1. Quy trình phát triển Web
   1. Thu thập thông tin

Mục đích, mục tiêu chính và đối tượng khách hàng. Giai đoạn này tập trung tìm hiểu và nghiên cứu về yếu cầu khách hàng để quyết định các bước tiếp theo sẽ thực hiện như thế nào. Nhiệm vụ quan trọng nhất trong bước này là hiểu rõ về mục đích của website, các chức năng website sẽ cung cấp, nhóm người dùng mục tiêu của website và thu thập đầy đủ các yêu cầu từ phía khách hàng, người dùng. Trả lời những câu hỏi trên sẽ giúp chúng ta lựa chọn được phương án tốt nhất để triển khai dự án sau này.

* 1. Lập kế hoạch

Lập trình viên sẽ đưa cho khách hàng cái nhìn tổng quan về website bằng các sitemap, wireframe hoặc bảng biểu dựa trên các thông tin đã thu thập được ở bước 1.

Sitemap sẽ mô tả mối liên kết các thành phần chính của website. Việc trình bày như vậy sẽ giúp khách hàng hiểu website được sử dụng như thế nào, nó cho thấy quan hệ giữa các trang khác nhau, vì vậy khách hàng có thể đánh gias được người dùng cuối sẽ dễ dàng tìm thấy thông tin hoặc dịch vụ từ trang chính Homepage. Lý do của việc tạo site map là để xây dựng website thân thiện nhất với những người dùng và dễ dàng điều hướng.

Sitemap cho phép bạn hiểu được cấu trúc bên trong của website nhưng không mô ta giao diện người dùng. Bởi vậy, khách hàng đôi khi chưa hình dung rõ ràng được cách mà người dùng của họ điều hướng trên trang như nào. Trong trường hợp này, một wireframe được tạp ra, là một bản trình bày trực quan cho giao diện người dùng. Nhưng nó không chứa bất kỳ yếu tố nào như màu sắc, biểu tương,... Nó chỉ mô tả các yếu tố sẽ được thêm vào trang và vị trí của chúng.

* 1. Thiết kế giao diện

Trang web sẽ trực quan hơn với các nút, hình ảnh, video,... sẽ được tạo ra. Một lần nữa, tất cả thông tin được thu thập trong giai đoạn đầu tiên là rất quan trọng. Các yêu cầu của khách hàng và trải nghiệm người dùng phải luôn được ghi nhỏ trong khi thực hiện thiết kế.

Bố cực trang web là một bản phác họa đồ họa hoặc một thiết kế đồ họa thực tế. Chức năng chính của bố cục là thể hiện cấu trúc thông tin, trực quan hóa nội dung và các chức năng cơ bản.

Sau đó, khách hàng có thể xem bố cực trang web và gửi lại phải hồi của họ. Nếu khách hàng không đồng ý về thiết kế của bạn, bạn nên thay đổi và gửi cho khách hàng đánh giá lại. Chu kỳ này nên được lặp lại đến khi khách hàng hài lòng.

* 1. Viết tài liệu đặc tả

SRS là tài liệu được sử dụng để mô ta chi tiết các yêu cầu chức năng và phi chức năng của hệ thống. Tài liệu này sẽ hỗ trợ đưa ra các tính năng của hệ thống hoặc dùng chp việc đọc hiểu hệ thống của phía khách hàng.

Đây là một tài liệu quan trọng cho đội phát triển và tester. Ngoài ra, dựa vào các yêu cầu trong SRS, ta có thể đánh giá được số lượng scope, thời gian hoàn thành và các chi phí trong quá trình phát triển sản phẩm một cách nhanh chóng và dễ dàng hơn.

* 1. Viết mã nguồn

Ở bước này, cuối cùng bạn có thể bắt tay vào tạo trang web. Trucows tiên khi bắt tay vào viết mà nguồn, chúng ta cần lựa chọn công nghệ, ngôn ngữ lập trình phù hợp cho dự án.

Các lập trình viên sẽ dựa trên các thiết kế trang web từ giai đoạn trước và sử dụng mã nguồn để làm chon trang web hoạt động và chạy trơn tru. Theo sơ đồ trang web, trang chủ sẽ được thiết kế đầu tiên, sau đó là các trang chức năng khác.

* 1. Kiểm thử, đánh giá và ra mắt sản phẩm

Quá trình kiểm thử là quá trình thường xuyên nhất của một quy trình. Mọi liên kết nên được kiểm tra để dảm bảo rằng không có liên kết nào bị lỗi hay gián đoạn. Bạn nên kiểm tra mọi biểu mẫu, mọi câu lệnh và kiểm tra lỗi chính tả của toàn bộ trang web. Sử dụng các tiêu chuẩn web chung để kiểm tra xem mã nguồn có đảm bảo sự tương thích giữa các trình duyệt hay không.

Sau khi đã kiểm thử xong, chúng ta sẽ triển khai trang web lên máy chủ (server). Sau khi triển khai xong mã nguồn lên máy chủ, chúng ta nên chạy thêm một bài kiểm tra cuối cùng để đảm bảo rằng các mã nguồn đã được cài đặt chính xác.

* 1. Bảo trì

Hãy coi trang web là một dịch vụ hơn là một sản phẩm. Bạn nên đảm bảo mọi thứ đều hoạt động tốt, mọi người đều hài lòng và luôn chuẩn bị sẵn sàng để thực hiện các thay đổi trong trường hợp khác.

Hãy bổ sung thêm hệ thống phản hồi vào chức năng của trang web để cho phép bạn phát hiện các vấn đề có thể xảy ra mà người dùng cuối gặp phải. Trong trường hợp này, ưu tiên cao nhất là phải khắc phục sự cố càng nhanh càng tốt. Nếu không, người dùng có thể sẽ không sử dụng trang web của bạn mà sử dụng một trang web khác tốt hơn.

Điều quan trọng khác là giữ cho trang web luôn được cặp nhật. Các bản cập nhật thường xuyên sẽ giúp tránh được lỗi và giảm rủi ro bảo mật.

1. Xây dựng Web hoàn chỉnh
   1. Node.js

Node.js là một môi trường thực thi mã JavaScript phía máy chủ được xây dựng trên JavaScript Engine V8 của Google Chrome. Nó cho phép bạn chạy mã JavaScript phía máy chủ, mà trước đây thường chỉ chạy trên trình duyệt web. Node.js có thư viện và module mạnh mẽ, cho phép bạn phát triển các ứng dụng máy chủ hiệu quả và mạnh mẽ.

Node.js thường được sử dụng để xây dựng ứng dụng web phía máy chủ, ứng dụng mạng, ứng dụng dựa trên dữ liệu thời gian thực, và nhiều loại ứng dụng máy chủ khác. Nó giúp xây dựng các ứng dụng có hiệu suất cao bằng cách sử dụng các sự kiện không chặn và I/O không chặn.

Một trong những đặc điểm quan trọng của Node.js là khả năng xử lý nhiều kết nối cùng một lúc một cách hiệu quả, đây là điểm mạnh khi xây dựng các ứng dụng có tính năng thời gian thực như ứng dụng trò chơi trực tuyến, ứng dụng chat trực tuyến, hoặc ứng dụng phát sóng trực tuyến.

Node.js đã trở thành một phần quan trọng trong việc phát triển các ứng dụng web và máy chủ hiện đại và được sử dụng rộng rãi trong cộng đồng phát triển phần mềm.

* 1. Postgresql

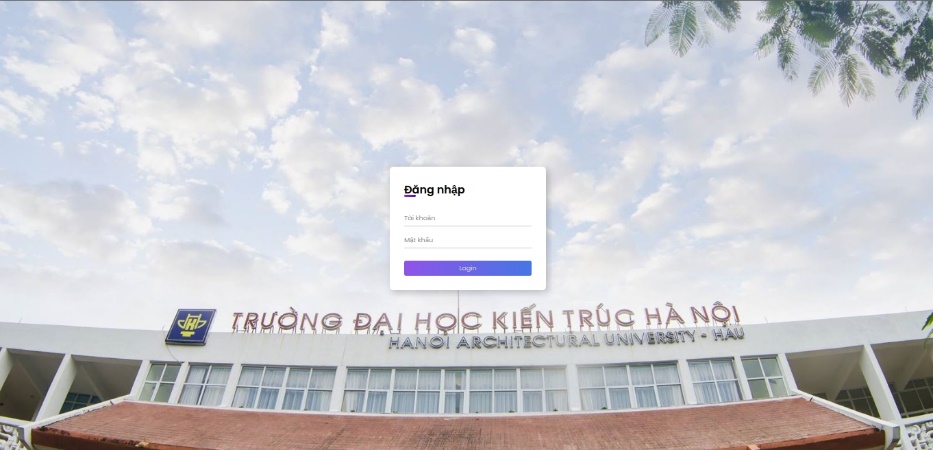
PostgreSQL là một hệ quản trị cơ sở dữ liệu (DBMS) mã nguồn mở và mạnh mẽ. Nó là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) được thiết kế để lưu trữ và quản lý dữ liệu trong các bảng và cung cấp một loạt các tính năng mạnh mẽ để truy vấn, cập nhật, và quản lý dữ liệu.

Dưới đây là một số điểm nổi bật về PostgreSQL:

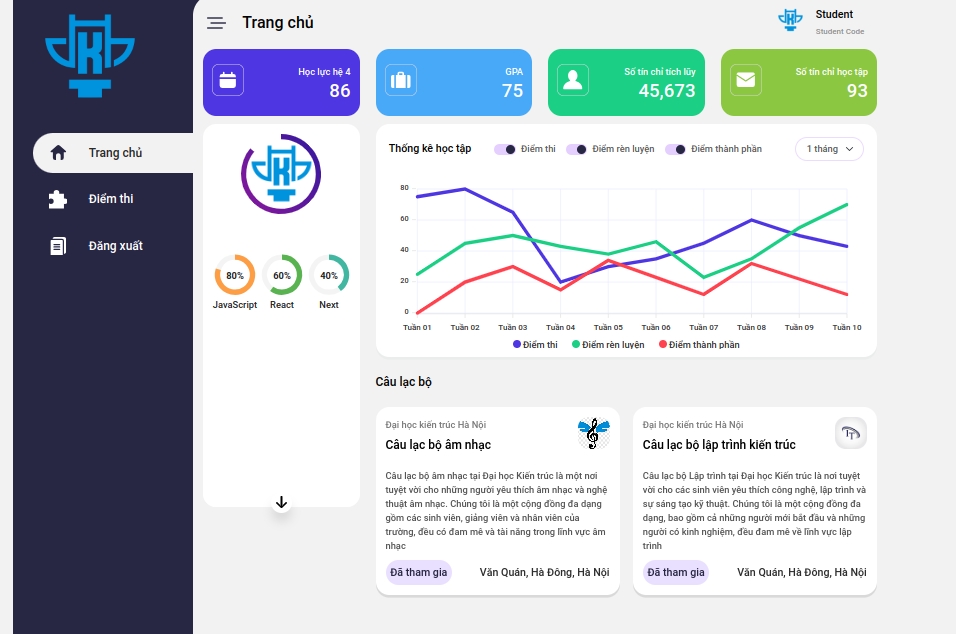
* Mã nguồn mở: PostgreSQL là một phần mềm mã nguồn mở và miễn phí. Điều này có nghĩa bạn có thể sử dụng, tùy chỉnh và phân phối nó mà không cần trả tiền cho giấy phép sử dụng.
* Hỗ trợ RDBMS đầy đủ: PostgreSQL là một hệ quản trị cơ sở dữ liệu quan hệ hoàn chỉnh, với khả năng hỗ trợ các tính năng như ACID (Atomicity, Consistency, Isolation, Durability) để đảm bảo tính nhất quán và an toàn cho dữ liệu.
* Đa nền tảng: PostgreSQL hỗ trợ trên nhiều hệ điều hành như Linux, Windows, macOS và nhiều nền tảng khác.
* Hỗ trợ tiêu chuẩn SQL: PostgreSQL tuân thủ tiêu chuẩn SQL ANSI, giúp dễ dàng chuyển đổi từ các hệ quản trị cơ sở dữ liệu quan hệ khác hoặc phát triển ứng dụng sử dụng SQL.
* Mở rộng và mạnh mẽ: PostgreSQL hỗ trợ cơ chế mở rộng như lập chỉ mục, giao dịch, và cơ chế nâng cao khác. Nó cũng cho phép bạn tạo các kiểu dữ liệu và hàm người dùng tùy chỉnh.
* Tích hợp tiện ích: PostgreSQL có nhiều tiện ích mở rộng và module mở rộng để cung cấp các tính năng bổ sung, ví dụ: PostGIS cho địa lý, hỗ trợ JSON và JSONB, và nhiều module khác.
* Cộng đồng phát triển lớn: PostgreSQL có một cộng đồng phát triển đông đảo và chuyên nghiệp, vì vậy nó liên tục được cập nhật và cải tiến.

PostgreSQL thường được sử dụng trong các dự án phát triển phần mềm, ứng dụng web, và hệ thống doanh nghiệp để lưu trữ và quản lý dữ liệu.

* 1. Giao diện web
     1. Giao diện đăng nhập:



* + 1. Giao diện trang chủ:



* + 1. Giao diện điểm thi:

