**LEARN ABOUT HTML**

***Overview***

* Hyper Text Markup Language – ngôn ngữ đánh dấu siêu văn bản, được sử dụng để tạo và định dạng nội dung trên trang web.

***Các tính năng mới trong HTML5***

* HTML5 đã mang đến nhiều tính năng mới và cải tiến so với các phiên bản HTML trước đó. Dưới đây là một số tính năng đáng chú ý trong HTML5:

**1. Hỗ trợ đa phương tiện:** HTML5 cung cấp các phần tử như `<video>` và `<audio>` cho phép nhúng và phát đa phương tiện trực tiếp trên trang web mà không cần sử dụng các plugin bên ngoài như Flash. Điều này giúp tăng tính tương thích và hiệu suất của các trang web đa phương tiện.

**2. Đồ họa và đồ thị:** HTML5 bổ sung phần tử `<canvas>` cho phép vẽ đồ họa và tạo ra đồ thị phức tạp bằng JavaScript. Ngoài ra, HTML5 cũng hỗ trợ đồ họa vector thông qua các phần tử như `<svg>`.

**3. Web Storage:** HTML5 cung cấp các API cho phép lưu trữ dữ liệu trên máy khách (client-side) như localStorage và sessionStorage. Điều này cho phép ứng dụng web lưu trữ và truy xuất dữ liệu trên máy tính của người dùng mà không cần phải gửi dữ liệu lên máy chủ.

**4. Web Workers:** HTML5 giới thiệu khái niệm Web Workers, cho phép thực hiện các tác vụ phức tạp trên nền tảng web mà không làm treo giao diện người dùng. Điều này cải thiện tính năng đáp ứng và trải nghiệm người dùng của các ứng dụng web.

**5. Đa luồng và tiến trình:** HTML5 hỗ trợ các tiến trình và luồng làm việc song song, cho phép thực hiện các tác vụ phức tạp và tương tác đa nhiệm trên trang web.

**6. Hỗ trợ định vị địa lý:** HTML5 cung cấp các API như Geolocation cho phép trang web truy cập vị trí địa lý của người dùng, điều này hữu ích trong các ứng dụng dựa trên vị trí như bản đồ, hướng dẫn đường, và quyền truy cập dựa trên vị trí.

**7. Các phần tử và thuộc tính mới:** HTML5 đưa ra nhiều phần tử mới như `<header>`, `<nav>`, `<section>`, `<article>`, `<footer>`,... giúp cải thiện cấu trúc và semantica của trang web. Ngoài ra, nó cũng cung cấp các thuộc tính mới cho các phần tử như `placeholder`, `required`, `pattern`,... để kiểm soát dữ liệu đầu vào và tương tác với người dùng.

***Hỗ trợ đa phương tiện***

* Trước khi <audio> và <video> xuất hiện, để nhúng video và âm thanh, người ta sử dụng các phần tử như <object>, <embed> và các plugin như Flash Player.
* Việc sử dụng plugin bên ngoài như Flash, phải yêu cầu người dùng cài đặt và cập nhật plugin đó trên trình duyệt của họ. Có thể không tương thích với 1 số thiết bị di động.

***Đồ họa và đồ thị***

* Thẻ <canvas> cho phép vẽ đồ họa và tạo ra đồ thị phức tạp trên trang web bằng cách sử dụng JS. <canvas> cung cấp một không gian làm việc trống mà ta có thể vẽ và tương tác thông qua mã JS.

***Web storage***

* Là 1 tính năng cho phép lưu trữ dữ liệu phía client – server (lưu trực tiếp trên trình duyệt của người dùng). Có 2 cơ chế lưu trữ chính **‘localStorage’,** và **‘sessionStorage’**:
* **‘localStorage’:** cho phép lưu trữ dữ liệu mà không có ngày hết hạn. Dữ liệu vẫn tồn tại sau khi người dùng đóng trình duyệt và mở lại trang web. Dữ liệu trong **‘localStorage’** không được gửi đến máy chủ khi gửi yêu cầu HTTP.
* **‘sessionStorage’:** cho phép lưu trữ dữ liệu trên trình duyệt, nhưng dữ liệu chỉ tồn tại trong phiên làm việc hiện tại. Khi đóng trình duyệt và mở lại thì dữ liệu trong **‘sessionStorage’** sẽ bị xóa đi.
* Ưu điểm: lưu trữ dữ liệu trực tiếp trên trình duyệt, giam phụ thuộc vào máy chủ; đối với **‘localStorage’** giúp lưu trữ dữ liệu của phiên làm việc; **‘sessionStorage’** giúp lưu trữ dữ liệu tạm thời của phiên làm việc đó; cải thiện hiệu suất, giảm lưu lượng mạng, giảm thời gian phản hồi.

***Web workers***

* Có 2 loại chính:
* **Delicated Workers:** chạy trong 1 luồng riêng biệt, không chia sẻ bộ nhớ với luồng giao diện người dùng. Đảm bảo việc thực hiện các tác vụ phức tạp không làm tăc nghẽn giao diện người dùng. Dedicated Workers có thể tương tác với giao diện người dùng thông qua gửi và nhận message.
* **Shared Workers:** có thể được sử dụng bởi nhiều trang hoặc cửa sổ chia sẻ cùng 1 nguồn worker. Cho phép truyền dữ liệu giữa các tab hoặc trong cùng 1 trình duyệt
* Là một cách mạnh mẽ để thực hiện các tác vụ phức tạp trên nền tảng web mà không làm tắc nghẽn giao diện người dùng. Cho phép chạy các luồng công việc độc lập và tương tác thông qua gửi và nhận thông điệp.

***XHTML và HTML***

* **XHTML (Extensible HyperText Markup Language)** và **HTML (HyperText Markup Language)** đều là ngôn ngữ đánh dấu để xây dựng và hiển thị các trang web trên Internet. Dưới đây là chi tiết về cả hai:

**1.** **HTML (HyperText Markup Language):**

* HTML là một ngôn ngữ đánh dấu dùng để xây dựng cấu trúc và định dạng các trang web. Nó sử dụng các thẻ (tags) để xác định các phần tử và cấu trúc của trang web.
* HTML không yêu cầu việc tuân thủ nghiêm ngặt các quy tắc cú pháp. Nó cho phép sử dụng các thẻ không được đóng và thuộc tính có thể không được trích dẫn.
* HTML5 là phiên bản mới nhất của HTML và đi kèm với nhiều tính năng mới và cải tiến, bao gồm hỗ trợ video, audio, canvas, geolocation, drag and drop, local storage, và nhiều tính năng khác.

**2. XHTML (Extensible HyperText Markup Language):**

XHTML là một biến thể của HTML, tuân thủ nghiêm ngặt các quy tắc XML (Extensible Markup Language). Nó kết hợp cú pháp XML với các quy tắc cú pháp của HTML.

* XHTML yêu cầu việc tuân thủ chặt chẽ các quy tắc cú pháp, bao gồm việc sử dụng các thẻ phải được đóng đúng cú pháp và thuộc tính phải được trích dẫn.
* XHTML đảm bảo tính nhất quán và tương thích với các công cụ xử lý XML, giúp tạo ra các trang web tương thích và dễ dàng mở rộng.
* **So sánh giữa XHTML và HTML:**
* ***Cú pháp:*** XHTML yêu cầu tuân thủ chặt chẽ cú pháp XML, trong khi HTML không yêu cầu sự nghiêm ngặt này.
* ***Tính nhất quán:*** XHTML đảm bảo tính nhất quán trong việc đóng và mở các thẻ, định dạng thuộc tính, v.v. Trong khi HTML cho phép sử dụng các thẻ không được đóng và thuộc tính không được trích dẫn.
* ***Tương thích công cụ xử lý XML:*** XHTML tương thích tốt với các công cụ xử lý XML, giúp việc phân tích và xử lý dữ liệu dễ dàng hơn.
* ***Phát triển tiếp:*** Vì tính chặt chẽ hơn trong việc tuân thủ cú pháp, việc phát triển tiếp trên XHTML có thể khó khăn hơn so với HTML.
* ***Sự lựa chọn:*** Hiện nay, HTML5 được sử dụng phổ biến hơn XHTML trong việc phát triển trang web, vì HTML5 kết hợp giữa tính linh hoạt của HTML và các tính năng mạnh mẽ của XHTML.
* Tóm lại, HTML và XHTML đều là ngôn ngữ đánh dấu để xây dựng trang web, nhưng XHTML tuân thủ chặt chẽ cú pháp XML và đảm bảo tính nhất quán hơn. Tuy nhiên, HTML5 hiện đang được sử dụng phổ biến hơn và được coi là tiêu chuẩn phát triển trang web hiện đại.

***Cú pháp XML***

* XML (Xtensible Markup Language) không phải là ngôn ngữ mà chỉ là một tiêu chuẩn định dạng dữ liệu được sử dụng để mô tả và truyền thông tin.
* XML cung cấp một cú pháp để định nghĩa cấu trúc dữ liệu và quy tắc cú pháp cho các tài liệu. Nó không chứa các quy tắc xử lý hoặc cú pháp cho việc thực thi logic hay xử lý sự kiện.
* XML được sử dụng rộng rãi như một công cụ để truyền thông tin giữa các hệ thống khác nhau, bởi vì nó là một ngôn ngữ độc lập, tức là có thể được sử dụng trên nhiều nền tảng và cho phép dữ liệu được truyền đi một cách cấu trúc.

***Các tags cơ bản trong HTML***

1. **<html>**: là phần tử cấp cao nhất trong cấu trúc tài liệu HTML. Xác định phần tử gốc cho toàn bộ tài liệu HTML và bao gồm các phần tử khác nhau của trang web.
2. **<head>**: xác định thông tin đầu trang của tài liệu HTML. Không hiển thị trực tiếp trên trình duyệt, chứa các thông tin quan trọng như tiêu đề liên kết CSS, thẻ meta, …
3. **<title>**: nằm bên trong thẻ <head>, sử dụng để đặt tiêu đề cho tài liệu HTML.
4. **<link>**: dùng để liên kết tài nguyên bên ngoài với tài liệu HTML hiện tại. Thường được sử dụng để liên kết các tệp CSS, biểu đồ, thư viện JS. Các thuộc tính:

* **‘rel’**: xác định mối quan hệ giữa tài liệu hiện tại và tài nguyên được liên kết. Bao gồm các giá trị phổ biến **‘stylesheet’** – liên kết CSS với tài liệu HTML, **‘icon’** – liên kết biểu tượng với trang web,
* **‘href’**: xác định địa chỉ URL của tài nguyên được liên kết. Có thể là 1 đường dẫn tương đối hoặc đưuòng dẫn tuyệt đối đến tệp nguồn.
* **‘type’**: xác định kiểu dữ liệu của tài nguyen được liên kết. Được sử dụng chủ yêu khi liên kết các tệp không phải là CSS hoặc hình ảnh. Ví dụ **‘type=text/javascript’** cho loại tệp JS.
* **<link>** cũng có thể được sử dụng để liên kết các font chữ chưa có sẵn vào tài liệu.

1. **<meta>**: dùng để cung cấp các thông tin bổ sung về tài liệu HTML. Một số thuộc tính phổ biến trong HTML:

* **‘charset’**: xác định mã kí tự được sử dụng trong HTML. Thường là **‘UTF-8’**, một phần của tiêu chuẩn Unicode, một chuẩn mã hóa ký tự được sử dụng trong việc biểu diễn các ký tự và biểu tượng từ nhiều ngôn ngữ khác nhau.
* **‘name’ và ‘content’**: được sử dụng cùng nhau để cung cấp thêm cac thông tin về trang web. Ví dụ **‘name = description’** dùng để cung cấp các thông tin về trang web. **‘name = keywords’**  dùng để xác định các từ khóa liên quan.
* **‘viewport’**: xác định cách hiển thị trên các thiết bị di động, cho phép kiểm soát tỉ lệ, thu phóng và hiển thị trên màn hình.
* **‘http-equiv’**: dùng để cung cấp các chỉ thị cho trình duyệt và máy chủ web.

1. **<body>**: là con trực tiếp của <html>. Chứa toàn bộ nội dung hiển thị trực tiếp trên trình duyệt bao gồm: văn bản, hình ảnh, video, âm thanh, các nút, …
2. **<script>**: nhúng mã JS vào tài liệu HTML. Có thể được đặt trong phần <head> hoặc phần <body> của tài liệu HTML. Nhưng thương được đặt ở cuối phần <body> trước khi đóng thẻ. Để đảm bảo răng mã JS được thực thi sau khi các phần tử HTML được tải. Cũng có thể đặt thẻ <script> ở phần <head>, nhưng cần sử dụng các kỹ thuật khác như sự kiện **‘DOMContentLoaded’** để đảm bảo JS được thực thi sau khi tài liệu HTML đã được tải hoàn toàn.