

# HTML

---

## ▼ DOCTYPE là gì ?

- là dòng mã đầu tiên được yêu cầu trong HTML
- hướng dẫn trình duyệt về phiên bản HTML
- đảm bảo trang web được phân tích cú pháp giống nhau bởi trình duyệt khác nhau.

## ▼ Thuộc tính DATA\_? là gì ?

- cho phép lưu trữ thêm thông tin dữ liệu trong DOM
- get data → getAttribute trong JS
- set data → setAttribute trong JS

## ▼ Cách set nhiều ngôn ngữ trên trang ?

- dùng thuộc tính **lang** trong thẻ html

## ▼ HTML SEMANTIC là gì?

- sử dụng thẻ thích hợp nhất cho các nhiệm vụ hiện có
- sử dụng những phần tử có ý nghĩa như form, article, table, thay vì sử dụng div hay span

## ▼ WEB ACCESSIBILITY?

- đảm bảo web có thể sử dụng cho người khuyết tật

## ▼ Element ≠ attribute ?

- **element** trong HTML là các thẻ như: div, span, a, img, ...
- **attribute** dùng để mô tả đặc điểm của element như: src, class, id

## ▼ display: none ≠ visibility: hidden

- cả 2 đều dùng để ẩn element
- none sẽ ẩn đi hoàn toàn element và không chiếm không gian trong giao diện

- hidden chỉ ẩn đi nhưng vẫn chiếm không gian trên giao diện

### ▼ <Script/>, <Script async/>, <Script defer/>

Khi tải trang HTML sẽ có 2 điều chỉnh được thực hiện:

- Chuyển đổi HTML
- Tải các đoạn script
- <Script/>: thực hiện tuần tự, chặn phân tích HTML → nạp script → thực thi → phân tích lại HTML sau khi thực thi xong
- <Script async/>: thực hiện không đồng bộ → không chặn phân tích cú pháp HTML
- <Script defer/>: phân tích cú pháp HTML xong rồi mới thực thi script

### ▼ DOM là gì?

- Là 1 api cho HTML, XML
- Đại diện các HTML, XML dưới dạng các node và object, có thể sử dụng thông qua JS
- Có cấu trúc dạng tree và mỗi element trên dom là một node

### ▼ Bố cục HTML

- HTML5 giới thiệu 1 số thẻ giúp chúng ta có thể định nghĩa bố cục trang web một cách rõ ràng hơn như:
  - **<header>**: Lưu trữ thông tin bắt đầu về trang web.
  - **<footer>**: Biểu diễn phần cuối cùng của trang.
  - **<nav>**: Menu điều hướng của trang HTML.
  - **<article>**: Nó là một tập hợp thông tin.
  - **<section>**: Nó được sử dụng bên trong **article** để xác định cấu trúc cơ bản của một trang.
  - **<aside>**: Nội dung menu dọc của trang.

### ▼ Cách tối ưu hiệu suất tải trang web

- Lưu trữ CDN → giảm độ trễ

- Nén tập tin → giảm kích thước nội dung → tăng tốc độ truyền dữ liệu
- Nối tập tin → giảm số lượng request được gọi
- Giảm thiểu (minify) tập lệnh: giảm kích thước tệp JS, CSS
- Lazy loading: tải nội dung cần thiết

#### ▼ Web worker là gì ?

- JS đơn luồng, do đó những tác vụ lớn đòi hỏi tính toán phức tạp hoặc thời gian phản hồi lâu thì khả năng làm cho giao diện đơ cứng → Web worker ra đời và giải quyết vấn đề nói trên
- Web worker không phải của JS mà là tính năng của browser
- Ta chỉ cần khởi tạo và truyền tham số đầu vào là file JS trong đó chứa các code cần xử lý
- WW giúp thực thi song song với nhưng chạy dưới dạng background
- WW là đa luồng
- Gồm 3 loại:
  - Service worker
  - Shared worker
  - Dedicated worker

#### ▼ Thẻ <a/> và <link/> khác gì nhau ?

- Thẻ <a> được dùng để mở liên kết đến trang web khác hay một phần nào đó ở web hiện tại nên nó có thể click vào.
- Thẻ <link> xác định một liên kết đến một nguồn tài nguyên bên ngoài, nó không thể click.

#### ▼ Khi nào sử dụng script ở header hoặc body

- Nếu các script chứa các hàm event-trigger hoặc thư viện jquery thì nên đặt ở head.
- Nếu script viết nội dung hoặc không nằm trong hàm thì nên đặt ở cuối body.
- Nói tóm lại có 3 điểm cần nhớ sau:

- Đặt thư viện hoặc sự kiện script ở head.
- Đặt script thông thường ở head cho đến khi có vấn đề gì đó về hiệu suất.
- Đặt script hiển thị nội dung ở cuối body.

### ▼ image ≠ figure

- Thẻ `<figure>` chỉ định nội dung như ảnh, sơ đồ, code snippets,... được dùng để tổ chức các nội dung như ảnh, tiêu đề ảnh
- Thẻ `<img>` dùng để nhúng một ảnh vào HTML5.

### ▼ Manifest file là gì ?

- File manifest được sử dụng để liệt kê các tài nguyên có thể được lưu vào bộ nhớ đệm.
- Trình duyệt sử dụng thông tin này để làm cho trang web tải nhanh hơn lần đầu tiên.
- Có 3 phần trong manifest:
  - CACHE Manifest - File cần lưu vào bộ đệm
  - Network - File không bao giờ lưu vào bộ đệm, cần kết nối mạng.
  - Fallback - File dự phòng trong trường hợp trang không tiếp cận được.

### ▼ srcset trong thẻ img là gì ?

- srcset cho phép bạn khai báo một tập hợp các hình ảnh sẽ được hiển thị trên các kích thước khung nhìn khác nhau.
- Bạn chỉ cần lưu và hình ảnh ở các độ phân giải khác nhau
- VD: `img_small.png 200w`, `img_medium.png 500w`, `img_large.png 1000w`
- và chúng được ngăn cách bởi dấu phẩy

### ▼ lazy loading là gì ? Các cách triển khai ?

- là kỹ thuật ngăn trình duyệt tải tất cả các resource cùng 1 lúc, thay vào đó chỉ tải những resource cần thiết
- Giúp tăng tốc độ tải trang, trải nghiệm người dùng cũng như tiết kiệm băng thông

- Một số cách triển khai:
  - Bắt sự kiện scroll → tạo data\_src → scroll tới target thì copy data\_src sang src
  - Dùng Intersection Observer API của browser
  - Dùng thuộc tính loading trong thẻ image
  - Sử dụng một số thư viện có sẵn