**Bạc Anh Tuấn**

|  |  |  |
| --- | --- | --- |
| Ngày sinh: 06/02/2004  Trường: ĐH Công Nghệ GTVT  Ngành: Công Nghệ Thông Tin  SĐT: 0965152243  Profile: facebook.com/anhtuanmc.24/ | Địa chỉ: Kiến Hưng, Hà Đông  Email: [anhtuan60204@gmail.com](mailto:anhtuan60204@gmail.com)  Link github báo cáo: https://github.com/AnhTuan2310/Intern-Report  Ngày vào làm: 27/09/2025 |  |

**Thu hoạch học tập ngày 29/10/2025**

# Front End (28/10/2025):

## HTML/CSS

### Các thẻ cơ bản

|  |  |
| --- | --- |
| <article> | Đại diện cho một nội dung độc lập (bài báo, bài đăng blog,...) |
| <aside> | Dành cho nội dung liên quan bên lề (sidebar, quảng cáo) |
| <details> | Tạo ra một tiện ích có thể thu gọn và mở rộng |
| <figcaption> | Dùng để nhóm 1 ảnh, sơ đồ <figure> với chú thích của nó <figcaption> |
| <footer> | Định nghĩa phần chân trang, thư mục (Thường là thông tin liên hệ, bản quyền,...) |
| <header> | Định nghĩa đầu trang hoặc của một mục (Thanh điều hướng, Logo, ...) |
| <main> | Đại diện cho nội dung chính, duy nhất của tài liệu |
| <mark> | Làm nổi bật văn bản trong một đoạn văn |
| <nav> | Dành riêng cho các liên kết điều hướng chính (Menu) |
| <section> | Nhóm các nội dung có liên quan đến nhau (1 Chương, 1 tab,...) |
| <summary> | Tương tự <details> |
| <time> | Định nghĩa ngày tháng, thời gian |

### Các layout cơ bản

#### CSS Flexbox (Flexible box):

Đây là hệ thống layout 1 chiều (hoặc hàng, hoặc cột).

Hoàn hảo cho: Căn chỉnh các mục trong menu (header), sắp xếp các thẻ (card) trong một hàng, căn giữa nội dung theo chiều dọc và chiều ngang.

#### CSS Grid (Lưới):

Đây là hệ thống layout 2 chiều (cả hàng và cột).

Hoàn hảo cho: Bố cục tổng thể của toàn bộ trang web (chia cột sidebar, main content), tạo các thư viện ảnh (gallery) phức tạp.

#### Mô hình hộp (The Box Model):

Một khái niệm cơ bản: Mọi phần tử HTML đều là một cái "hộp".

Bao gồm 4 phần: Content (nội dung), Padding (vùng đệm), Border (viền), và Margin (lề). Hiểu rõ điều này là bắt buộc để căn chỉnh layout.

#### Positioning (Định vị):

Sử dụng thuộc tính position (với các giá trị static, relative, absolute, fixed, sticky).

Giúp bạn đặt một phần tử ở một vị trí chính xác trên trang, hoặc "ghim" nó lại khi cuộn.

#### Float:

Cách cũ để cho văn bản "chảy" xung quanh một hình ảnh.

Ngày nay, không nên dùng float để xây dựng layout chính của trang (dùng Flexbox hoặc Grid).

### Responsive

#### Media Queries (@media):

Đây là công cụ quan trọng nhất. Nó cho phép bạn áp dụng các quy tắc CSS khác nhau dựa trên điều kiện của thiết bị

*(Chẳng hạn: "nếu chiều rộng màn hình nhỏ hơn 768px, hãy cho menu hiển thị thành dạng nút bấm").*

Ví dụ: **@media (max-width: 600px) { ... }**

#### Viewport Meta Tag:

Một dòng bắt buộc phải có trong thẻ <head> của HTML.

Nó bảo trình duyệt di động: "Hãy hiển thị trang web với chiều rộng bằng chiều rộng của thiết bị và không phóng to".

**<meta name="viewport" content="width=device-width, initial- scale=1.0">**

#### Fluid Grids (Lưới linh hoạt):

Sử dụng Flexbox/Grid kết hợp với các đơn vị tương đối (như %) thay vì các đơn vị cố định (như px) để chiều rộng các cột tự co giãn theo màn hình.

#### Flexible Units (Đơn vị linh hoạt):

Ưu tiên dùng các đơn vị như %, vw (chiều rộng viewport), vh (chiều cao viewport), rem, em (cho kích thước chữ) thay vì px.

#### Flexible Media (Hình ảnh/Video linh hoạt):

Một kỹ thuật CSS đơn giản để đảm bảo hình ảnh không bao giờ bị "tràn" ra ngoài màn hình:

**img, video { max-width: 100%; height: auto; }**

#### Mobile-First (Ưu tiên di động):

Đây là một chiến lược thiết kế: Xây dựng layout cho di động trước, sau đó dùng Media Queries để thêm các yếu tố phức tạp hơn cho màn hình lớn hơn.

## JS ES6:

#### 1. Biến (Variables):

Trong phiên bản cũ, biến hay dùng **var** nhưng var hay bị lỗi là tràn ra ngoài block của chính nó, ví dụ khai báo trong **for** nhưng ngoài **for** lại gọi lên được

Phiên bản mới hiện nay bổ sung **let** để giải quyết vấn đề đó, ví dụ **let** trong **for** thì ngoài for kh gọi được

**const** thì phải gán giá trị và kh thể thay đổi giá trị trong quá trình làm trừ đối với Obj và Array thì có thể thay đổi data bên trong được nhưng kh ghi đè cho cấu trúc cũ được

Ví dụ:

**const LIST\_DATA = {id = "5"}** thì có thể set **LIST\_DATA.id = "6"** chứ không thể **LIST\_DATA = {id = "7"}**

#### 2. Hàm (Method)

Tham số Mặc định (Default Parameters): Cho phép gán giá trị mặc định cho tham số nếu nó không được truyền vào.

Ví dụ:

**function chaoES6(ten = 'Bạn') {**

**console.log(`Chào ${ten}`);**

**}**

**chaoES6(); // "Chào Bạn"**

**chaoES6("TEN"); // "Chào TEN"**

Tham số Còn lại (Rest Parameters): Dùng ... để gom tất cả các tham số còn lại vào một mảng.

Ví dụ:

**function tinhTong(...so) {**

**// 'so' là một mảng, chẳng hạn [1, 2, 3, 4]**

**return so.reduce((total, num) => total + num, 0);**

**}**

**console.log(tinhTong(1, 2, 3, 4)); // 10**

#### 3. Arrow Func (Hàm mũi tên):

Đây là cú pháp viết hàm ngắn gọn và giải quyết từ khóa this

Ví dụ:

Cũ:

**var binhPhuong = function(x){**

**return x\*x;**

**}**

Mới:

**const binhPhuong = (x) =>{**

**return x\*x;**

**}**

Đặc biệt nếu chỉ có 1 tham số và cần 1 return thì có thể viết:

**const binhPhuong = x => x\*x;**

Lexical:

Ví dụ:

ES5 (Phải dùng 'self' vì 'this' trong setTimeout là 'window')

function DongHo() {

this.giay = 0;

var self = this;

setInterval(function() {

self.giay++;

console.log(self.giay);

}, 1000);

}

ES6 (Arrow function sử dụng luôn 'this' của DongHoES6)

function DongHoES6() {

this.giay = 0;

setInterval(() => {

this.giay++;

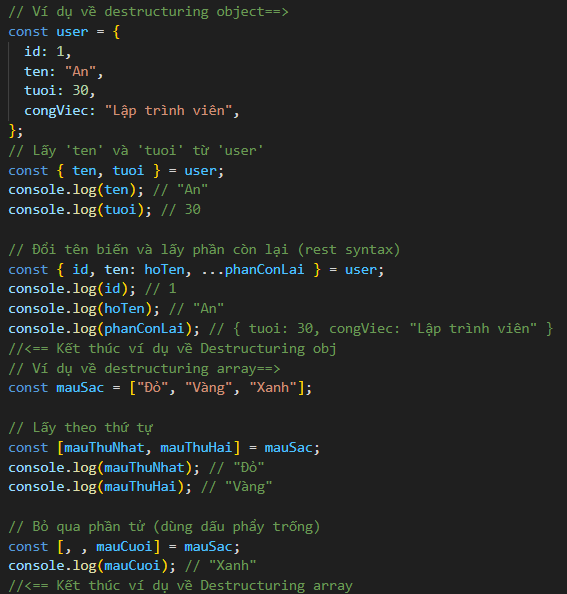
console.log(this.giay);

}, 1000);

}

#### 4. Destructuring: Phá vỡ cấu trúc

Đây là cú pháp giúp bóc tách phần tử từ Objs/ Arrays cho phép bạn "bóc tách" các giá trị từ Objects hoặc Arrays vào các biến riêng lẻ. Giúp code ngắn gọn và dễ đọc hơn rất nhiều.



#### 5. Module

Module hiểu đơn giản là một file JS độc lập, có thể xuất (export) các biến/hàm/lớp… ra ngoài để file khác nhập (import) vào dùng.

Tác dụng: chia nhỏ code, tái sử dụng, tránh trùng tên, và dễ bảo trì.

#### 6. Export và Import:

Export:

Named Export: Dùng khi muốn export nhiều thứ trong cùng file.

Default Export: Dùng khi file chỉ có một giá trị chính cần export

Import

Named Import: Import những Named Export phải sử dụng {}

Default Import: Import từ những Default Export không cần sử dụng {}



# Back End (29/10/2025):