# CHƯƠNG I: TỔNG QUAN VỀ TAILWIND CSS

## GIỚI THIỆU VỀ TAILWIND CSS

Tailwind CSS là một utility-first CSS framework, nó cũng giống như Bootstrap, nó có những class built-in (những class được định nghĩa sẵn) mà chúng ta có thể dùng. Tailwind CSS có nhiều các class bao gồm các thuộc tính CSS khác nhau mà chúng ta có thể sử dụng ngay mà không cần phải rời khỏi HTML hoặc viết bất kỳ dòng CSS nào và quan trọng, chúng ta có thể dễ dàng mở rộng tạo mới ra những class bằng chính những class của nó.

Ví dụ: giả sử bạn muốn tạo một nút có chiều cao cố định, phần đệm ngang, màu nền đen, bo cạnh tròn và phông chữ đậm màu trắng. Đây là HTML bạn sẽ sử dụng:

<button class="h-10 px-6 font-semibold rounded-md bg-black text-white" type="submit">

Buy now

</button>

Ta có kết quả như sau:



## ĐIỂM MỚI KHI SỬ DỤNG TAILWIND CSS

Tailwind CSS tươn tự như Bootstrap khi nó có những class build-in (những class được đinh nghĩa sẵn) tuy nhiên không như Bootstrap những class build-in là những phần tử của một trang web được định nghĩa sẵn như headers, heroes, dropdowns, sidebars,.. thì ở Taildwind CSS chúng ta có những class build-in là những thuộc tính CSS cụ thể như: bố cục, màu sắc, khoảng cách, font chữ,… việc có thêm nhiều class nhưng quy tắc đặt tên cực kỳ thân thiện với người dùng, người dùng có thể nhìn vào class cũng có thể biết ngay được nó đang style cái gì. Chúng ta cũng phải nói đến khả năng tuỳ biến mà mở rộng cao, đem đến sự linh động điều mà Bootstrap hay những framework khác khó thực hiện.

Ngoài ra, khi chúng ta sử dụng framework này ngoài việc sử dụng các class build-in của Tailwind có sẵn, ta có thể sử dụng chính những class build-in đó xây dựng một class mới. Khi đó ta có thể sử dụng từ khoá `@apply`.

.demo {

@apply flex justify-between items-center py-4 bg-blue-900;

}

Có thể nói đây là điểm riêng biệt mà framework này nổi bật so với framework CSS khác khi nó sử dụng class built-in của framework trong một class mới, làm cho code gọn gàng hơn rất nhiều. Nếu dùng Bootstrap thì sẽ phải viết hết tất cả các class built-in của framework mà ta cần vào thẻ đó, nó làm sauce code nặng, và dài dòng khó đọc.

Một điểm được ưa thích ở framework CSS này là Pseudo-Class Variants (biến lớp giả) ta có thể viết class trực tiếp cho các sự kiện như `hover, focus, active, disabled, visited, first-child, last-child, odd-child, even-child, focus-within`:

<form>

<input class="bg-gray-200 hover:bg-white hover:border-gray-300 focus:outline-none focus:bg-white focus:shadow-outline focus:border-gray-300 ...">

<button class="bg-teal-500 hover:bg-teal-600 focus:outline-none focus:shadow-outline ...">

Sign Up

</button>

</form>

Đoạn code trên tạo ra 1 ô input với nút Sign up, thay vì phải viết css dài thì ta chỉ cần thêm tiền tố hover, focus vào trước class thôi là chúng ta đã có thể thực hiện thay đổi style khi hover hoặc focus vào thẻ html.

# CHƯƠNG II: CÀI ĐẶT VÀ BẮT ĐẦU SỬ DỤNG

## CÁCH CÀI ĐẶT TAILWIND CSS VÀO DỰ ÁN

Cách đơn giản và nhanh nhất để thiết lập và chạy Tailwind CSS từ đầu là sử dụng công cụ CLI của Tailwind CSS.

### Cài đặt Tailwind CSS bằng npm

Lưu ý: trước đó cần cài đặt Node.js để có thể sử dụng npm.

npm install -D tailwindcss

npx tailwindcss init

### Cấu hình đường dẫn mẫu

Ở bước này ta cấu hình đường dẫn ở content để Tailwind đọc html của bạn và tạo ra file CSS tương ứng.

module.exports = {

content: ["./src/\*\*/\*.{html,js}"],

theme: {

extend: {},

},

plugins: [],

}

### Thêm các lệnh Tailwind vào CSS

Thêm lệnh @tailwind cho từng lớp của Tailwind vào tệp CSS chính `input.css` (những CSS chung cho toàn bộ trang web).

@tailwind base;

@tailwind components;

@tailwind utilities;

### Chạy công cụ CLI

Chạy công cụ CLI để nó quét HTML và tìm ra những class và biên dịch nó sang CSS tương ứng `output.css`.

npx tailwindcss -i ./src/input.css -o ./dist/output.css –watch

### Bắt đầu sử dụng Tailwind CSS trong HTML

Liên kết CSS đã được biên dịch vào HTML

<!doctype html>

<html>

<head>

<meta charset="UTF-8">

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

<link href="/dist/output.css" rel="stylesheet">

</head>

<body>

<h1 class="text-3xl font-bold underline">

Hello world!

</h1>

</body>

</html>

## CẤU HÌNH CƠ BẢN VÀ CÁC TÙY CHỌN TÙY CHỈNH

Vì Tailwind CSS là một framework để xây dựng giao diện người dùng, nên nó được thiết kế ngay từ đầu với mục đích có thể tuỳ chỉnh linh hoạt.

Theo mặc định, Tailwind sẽ file tuỳ chỉnh `tailwind.congif.js` ở thư mục gốc của dự án. Mỗi một phần của file cấu hình đều là tùy chọn, vì vậy bạn chỉ phải chỉ định cụ thể những gì bạn muốn thay đổi. Mọi phần bị thiếu sẽ quay về cấu hình mặc định của Tailwind.

### Tạo một file cấu hình

Tạo file cấu hình Tailwind cho dự án bằng công cụ Tailwind CLI đi kèm khi cài đặt pack npm tailwindcss:

npx tailwindcss init

Lệnh này sẽ tạo file tailwind.congif.js ở thư mục gốc của dự án.

/\*\* @type {import('tailwindcss').Config} \*/

module.exports = {

content: [],

theme: {

extend: {},

},

plugins: [],

}

### Content

Phần này là nơi cấu hình đường dẫn đến các file HTML hay các components Javascript nơi chứa các class của Tailwind. `content`

/\*\* @type {import('tailwindcss').Config} \*/

module.exports = {

content: [

'./pages/\*\*/\*.{html,js}',

'./components/\*\*/\*.{html,js}',

],

// ...

}

### Theme

Phần theme là nơi định nghĩa bảng màu, phông chữ, tỷ lệ, kích thước, điểm ngắt (break point) - bất kỳ thư gì liên quan dến thiết kế của trang web.

/\*\* @type {import('tailwindcss').Config} \*/

module.exports = {

// ...

theme: {

colors: {

'blue': '#1fb6ff',

'purple': '#7e5bef',

'pink': '#ff49db',

'orange': '#ff7849',

'green': '#13ce66',

'yellow': '#ffc82c',

'gray-dark': '#273444',

'gray': '#8492a6',

'gray-light': '#d3dce6',

},

fontFamily: {

sans: ['Graphik', 'sans-serif'],

serif: ['Merriweather', 'serif'],

},

extend: {

spacing: {

'8xl': '96rem',

'9xl': '128rem',

},

borderRadius: {

'4xl': '2rem',

}

}

}

}

# CHƯƠNG III: CÚ PHÁP VÀ CẤU TRÚC CƠ BẢN

## GIẢI THÍCH CÚ PHÁP CỦA TAILWIND CSS

### Cú pháp chung

Mỗi class của Tailwind CSS bắt đầu bằng tiền tố ví dụ như: ` tw-`.

Mỗi class có thể chứa nhiều thuộc tính được kết hợp với nhau để tạo ra một style cụ thể.

### Kiểu chữ và kích thước.

` text-{size}`: Thiết lập kích thước của văn bản.

` font-{weight}`: Thiết lập độ đậm của văn bản.

### Màu sắc.

` text-{color}`: Thiết lập màu chữ.

`bg-{color}`: Thiết lập màu nền của phần tử.

### Bố cục và Kích thước:

` p-{size}`: Padding.

` m-{size}`: Margin.

` w-{size}`: Chiều rộng.

` h-{size}`: Chiều cao.

### Kiểu đường viền và Góc bo tròn.

` border `: Định rõ đường viền.

` rounded `: Góc bo tròn.

### Hiệu ứng hover và Trạng thái.

` hover:{property}`: Áp dụng khi hover.

` focus:{property}`: Áp dụng khi phần tử đang được chú ý (đối với input, textarea, v.v.).

### Flexbox và Grid.

` flex `: Sử dụng Flexbox.

` grid `: Sử dụng CSS Grid.

### Ẩn/Hiện và Độ mờ.

` hidden `: Ẩn phần tử.

` opacity-{value}`: Độ mờ của phần tử.

### Vị trí và Căn chỉnh.

` absolute, relative `: Định vị tuyệt đối và tương đối.

` left, right, top, bottom `: Điều chỉnh vị trí.

## LÀM THẾ NÀO TAILDWIND CSS THAY ĐỔI CÁCH BẠN VIẾT HTML VÀ CSS SO VỚI CÁC PHƯƠNG PHÁP TRUYỀN THỐNG

### HTML và CSS sử dụng các utility classes

* Truyền thống: Trong phương pháp truyền thống, bạn thường viết CSS trong các tệp riêng biệt và sau đó áp dụng các lớp hoặc ID vào HTML.
* Tailwind CSS: Bạn sử dụng các utility classes trực tiếp trong HTML để định dạng style và bố cục. Ví dụ, thay vì viết `.my-class { color: red; }`, bạn sẽ viết `class="text-red-500"`.

### Tuỳ chỉnh tại chỗ

* Truyền thống: Việc tùy chỉnh kiểu dáng thường đòi hỏi viết CSS tùy chỉnh mới hoặc thay đổi tệp CSS hiện tại.
* Tailwind CSS: Bạn có thể tùy chỉnh kiểu dáng ngay tại chỗ trong HTML bằng cách sử dụng các utility classes, giúp tiết kiệm thời gian và giảm cần thiết phải mở rộng tệp CSS.

### Số Lượng Classes Lớn:

* Truyền thống: Thường cần một số lượng lớn các classes CSS để bao gồm tất cả các trường hợp sử dụng.
* Tailwind CSS: Cung cấp một số lượng lớn các utility classes có sẵn có thể kết hợp để tạo ra các style khác nhau.

# CHƯƠNG IV: RESPONSIVE DESIGN VỚI TAILDWIND CSS

## SỬ DỤNG CÁC CLASS REPONSIVE ĐỂ TỐI ƯU HÓA TRẢI NGHIỆM NGƯỜI DÙNG TRÊN CÁC THIẾT BỊ KHÁC NHAU.

### Sử Dụng Class Responsive Theo Kích Thước Màn Hình

Sử dụng các class responsive để điều chỉnh kiểu dáng dựa trên kích thước màn hình.

Ví dụ:

<div class="text-2xl lg:text-4xl xl:text-6xl">

Responsive Text

</div>

Trong ví dụ này, kích thước chữ sẽ là 2xl trên mọi thiết bị, tăng lên thành 4xl trên màn hình lớn (lg), và 6xl trên màn hình rất lớn (xl).

### Ẩn/Hiện Phần Tử Theo Kích Thước Màn Hình

Sử dụng class hidden và block responsive để ẩn/hiện phần tử dựa trên kích thước màn hình.

Ví dụ:

<div class="hidden lg:block">

Visible on large screens

</div>

Trong trường hợp này, phần tử sẽ ẩn trên mọi thiết bị, nhưng sẽ hiện trên màn hình lớn (lg).

### Thiết lập kích thước cho Margin và Padding:

Sử dụng class responsive để thiết lập kích thước cho margin và padding.

<div class="p-4 md:p-8 lg:p-12"

Responsive Padding

</div>

Kích thước padding sẽ là 4 trên mọi thiết bị, 8 trên màn hình trung bình (md), và 12 trên màn hình lớn (lg).

Ngoài ra cũng có thể sử dụng các responsive cho nhiều tính năng khác nhau như màu sắc, font size, và nhiều hơn nữa.

# MỤC LỤC

[CHƯƠNG I: TỔNG QUAN VỀ TAILWIND CSS 1](#_Toc150694846)

[I. GIỚI THIỆU VỀ TAILWIND CSS 1](#_Toc150694847)

[II. ĐIỂM MỚI KHI SỬ DỤNG TAILWIND CSS 1](#_Toc150694848)

[CHƯƠNG II: CÀI ĐẶT VÀ BẮT ĐẦU SỬ DỤNG 2](#_Toc150694849)

[I. CÁCH CÀI ĐẶT TAILWIND CSS VÀO DỰ ÁN 2](#_Toc150694850)

[1.1. Cài đặt Tailwind CSS bằng npm 2](#_Toc150694851)

[1.2. Cấu hình đường dẫn mẫu 2](#_Toc150694852)

[1.3. Thêm các lệnh Tailwind vào CSS 2](#_Toc150694853)

[1.4. Chạy công cụ CLI 3](#_Toc150694854)

[1.5. Bắt đầu sử dụng Tailwind CSS trong HTML 3](#_Toc150694855)

[II. CẤU HÌNH CƠ BẢN VÀ CÁC TÙY CHỌN TÙY CHỈNH 3](#_Toc150694856)

[1.1. Tạo một file cấu hình 3](#_Toc150694857)

[1.2. Content 4](#_Toc150694858)

[1.3. Theme 4](#_Toc150694859)

[CHƯƠNG III: CÚ PHÁP VÀ CẤU TRÚC CƠ BẢN 5](#_Toc150694860)

[I. GIẢI THÍCH CÚ PHÁP CỦA TAILWIND CSS 5](#_Toc150694861)

[1.1. Cú pháp chung 5](#_Toc150694862)

[1.2. Kiểu chữ và kích thước. 5](#_Toc150694863)

[1.3. Màu sắc. 5](#_Toc150694864)

[1.4. Bố cục và Kích thước: 5](#_Toc150694865)

[1.5. Kiểu đường viền và Góc bo tròn. 5](#_Toc150694866)

[1.6. Hiệu ứng hover và Trạng thái. 5](#_Toc150694867)

[1.7. Flexbox và Grid. 5](#_Toc150694868)

[1.8. Ẩn/Hiện và Độ mờ. 5](#_Toc150694869)

[1.9. Vị trí và Căn chỉnh. 5](#_Toc150694870)

[II. LÀM THẾ NÀO TAILDWIND CSS THAY ĐỔI CÁCH BẠN VIẾT HTML VÀ CSS SO VỚI CÁC PHƯƠNG PHÁP TRUYỀN THỐNG 6](#_Toc150694871)

[1.1. HTML và CSS sử dụng các utility classes 6](#_Toc150694872)

[1.2. Tuỳ chỉnh tại chỗ 6](#_Toc150694873)

[1.3. Số Lượng Classes Lớn: 6](#_Toc150694874)

[CHƯƠNG IV: RESPONSIVE DESIGN VỚI TAILDWIND CSS 6](#_Toc150694875)

[I. SỬ DỤNG CÁC CLASS REPONSIVE ĐỂ TỐI ƯU HÓA TRẢI NGHIỆM NGƯỜI DÙNG TRÊN CÁC THIẾT BỊ KHÁC NHAU. 6](#_Toc150694876)

[1.1. Sử Dụng Class Responsive Theo Kích Thước Màn Hình 6](#_Toc150694877)

[1.2. Ẩn/Hiện Phần Tử Theo Kích Thước Màn Hình 6](#_Toc150694878)

[1.3. Thiết lập kích thước cho Margin và Padding: 7](#_Toc150694879)