**MỤC LỤC**

[CHƯƠNG 1. CÚ PHÁP 3](#_Toc200832667)

[1.1. GIỚI THIỆU 4](#_Toc200832668)

[1.1.1. Quy tắc đặt tên BEM (Block Element Modifier) 4](#_Toc200832669)

[1.2. Selectors 6](#_Toc200832670)

[1.2.1. Selector cơ bản 6](#_Toc200832671)

[1.2.2. Selector kết hợp 6](#_Toc200832672)

[1.2.3. Pseudo-class (lớp giả) 7](#_Toc200832673)

[1.2.4. Pseudo-element (phần tử giả) 7](#_Toc200832674)

[1.2.5. Attribute selector (theo thuộc tính) 8](#_Toc200832675)

[1.3. Sepecificity 9](#_Toc200832676)

[1.4. Box Model 10](#_Toc200832677)

[CHƯƠNG 2. Layout 11](#_Toc200832678)

[2.1. Flex box 11](#_Toc200832679)

[2.2. GRID 11](#_Toc200832680)

[CHƯƠNG 3. Responsive Design 12](#_Toc200832681)

[3.1. Media queries 12](#_Toc200832682)

[3.2. Mobile first 12](#_Toc200832683)

[CHƯƠNG 4. Animation 13](#_Toc200832684)

[4.1. Keyframs 13](#_Toc200832685)

[4.2. Trasitions 13](#_Toc200832686)

[CHƯƠNG 5. Tìm hiểu về css framework 14](#_Toc200832687)

[5.1. Cơ bản về tailwindCSS. 14](#_Toc200832688)

**MỤC LỤC HÌNH ẢNH**

[Hình 1.1 Quy tắc đặt biến định kiểu CSS. 4](#_Toc200832723)

[Hình 1.2 sơ đồ minh họa box model 10](#_Toc200832724)

**MỤC LỤC BẢNG BIỂU**

[Bảng 1.1 Quy tắc đặt tên BEM 6](#_Toc200832783)

[Bảng 1.2 ví dụ cho bem 6](#_Toc200832784)

[Bảng 1.3 Các slector cơ bản 7](#_Toc200832785)

[Bảng 1.4 Các slector kết hợp 7](#_Toc200832786)

[Bảng 1.5 Các secletor cho các lớp giả 8](#_Toc200832787)

[Bảng 1.6 Các selector cho phần tử giả 9](#_Toc200832788)

[Bảng 1.7 Các selector theo thuộc tính 9](#_Toc200832789)

[Bảng 1.8 Thứ tự ưu tiên của các thành phần css 10](#_Toc200832790)

# CÚ PHÁP

Lưu ý: Các kiến thức cần nắm

**BEM**: block element và modifier (khối thành phần và đặc điểm của khối hoặc thành phần)

**SELECTORS:** Cần học đầy đủ cả các loại:

* Basic (element, class, id)
* Combinators (descendant div p, child div > p)
* Attribute selectors [type="text"]
* Pseudo-classes (:hover, :nth-child(), v.v.)
* Pseudo-elements (::before, ::after)

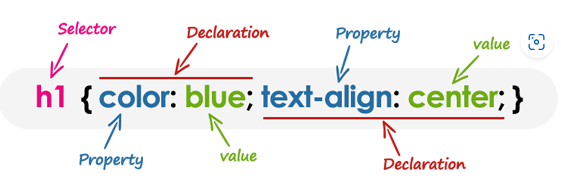
**SPECIFICITY:** Quan trọng, cần làm bài tập để hiểu rõ mức độ ưu tiên.

**BOX MODEL:** Rất cần thiết, nên bổ sung thêm các kiến thức như:

* margin collapse
* box-sizing: border-box

## GIỚI THIỆU

Cú pháp của biến định kiểu css là một tập hợp các quy tắc được web diễn giải và áp dụng cho các thẻ html.



Hình 1.1 Quy tắc đặt biến định kiểu CSS.

* CSS đều phân biệt chữ hoa chữ thường.
* Có thể lồng nhau giữa các biến định kiểu.

### Quy tắc đặt tên BEM (Block Element Modifier)

Tại sao phải dùng BEM:

Khi sử dụng css thường xuyên gặp các vấn đề như là:  
⇨ Cấu trúc html lồng nhau khi thay đổi một cấu trúc html thì css sẽ không còn hoạt động đúng như yêu cầu dẫn đến code khó bào trì và mở rộng.

.header ul li a span {

color: red;

}.

* Style bị ghi đè ngoài ý muốn. Code không có tổ chức. Phụ thuộc quá nhiều vào HTML DOM.
* Từ đó Quy tắc BEM ra đời. ngoài quy tắc BEM vẫn còn nhiều quy tắc đặt tên cho class.

Cú pháp chung của BEM là: .block\_\_element—modifier

Trong đó:

|  |  |  |
| --- | --- | --- |
| **Thành phần** | **Ý nghĩa** | **Ví dụ** |
| Block | Thành phần độc lập, có thể tái sử dụng | menu, button, form |
| Element | Thành phần con phụ thuộc vào block | menu\_\_item, form\_\_input |
| Modifier | Trạng thái, phiên bản của block hoặc element | button--active, menu\_\_item--highlighted |

Bảng 1.1 Quy tắc đặt tên BEM

Ví dụ:



Bảng 1.2 Ví dụ cho bem

* Dấu cách trong thuộc tính class là biểu thị cho nhiều class như ví dụ trên thì li đầu tiên có 2 class menu\_item và menu\_item—active.

#### Lưu ý khi sử dụng BEM

👉 Đặt tên theo **ngữ nghĩa**, **chức năng**, **vai trò**, không theo màu sắc hay kích thước.

👉 Không dùng **BEM** cho những thứ nhỏ nhặt ví dụ các thẻ hr br.

👉 Chỉ sử dụng **class selectors** trong CSS.

## Selectors

**Selector** (bộ chọn) là phần dùng để **chọn ra các phần tử HTML mà bạn muốn áp dụng kiểu dáng (style)**.

### Selector cơ bản

|  |  |  |  |
| --- | --- | --- | --- |
| **Loại Selector** | **Cú pháp** | **Ý nghĩa** | **Ví dụ** |
| Phần tử | element | Chọn tất cả thẻ có tên tương ứng | p, h1, div |
| Class | .class | Chọn phần tử có class cụ thể | .menu, .title |
| ID | #id | Chọn phần tử có id cụ thể | #header, #main |
| Tất cả phần tử | \* | Chọn mọi phần tử trong tài liệu | \* { margin: 0; } |

Bảng 1.3 Các slector cơ bản

#### Lưu ý khi sử dụng selector

* Không nên viết selector lồng nhiều cấp dễ bị phụ thuộc cấu trúc DOM thay vì vậy ta nên sử dụng selector class với quy tắc BEM để style cho tags.

### Selector kết hợp

|  |  |  |
| --- | --- | --- |
| **Loại kết hợp** | **Cú pháp** | **Ý nghĩa** |
| **Chọn trong nhau** | div p | Chọn tất cả p nằm trong div |
| **Chọn trực tiếp** | div > p | Chọn p là con **trực tiếp** của div |
| **Chọn anh em liền kề** | h2 + p | Chọn p đứng **ngay sau** h2 , và cả hai đều có cùng một phần tử cha. |
| **Chọn tất cả anh em sau** | h2 ~ p | Chọn tất cả p đứng **sau** h2 cùng cấp, và cả hai đều có cùng một phần tử cha. |

Bảng 1.4 Các slector kết hợp

### Pseudo-class (lớp giả)

Lớp giả giúp bạn định kiểu cho một phần tử khi nó ở một trạng thái đặc biệt hoặc mối quan hệ nhất định với các phần tử khác.

|  |  |
| --- | --- |
| **Selector** | **Ý nghĩa** |
| :active | Chọn một phần tử khi nó đang được kích hoạt (ví dụ: khi nhấp chuột và giữ trên một nút). |
| :hover | Khi trỏ chuột vào phần tử |
| :focus | Chọn một phần tử khi nó nhận được tiêu điểm (ví dụ: khi nhấp vào một trường nhập liệu hoặc điều hướng đến nó bằng phím Tab). |
| :visited | Chọn một liên kết đã được truy cập. |
| :first-child | Phần tử đầu tiên trong cha |
| :last-child | Phần tử cuối cùng trong cha |
| :nth-child(n) | Phần tử thứ n trong cha |
| :not(selector) | Chọn tất cả trừ những cái khớp với selector |

Bảng 1.5 Các secletor cho các lớp giả

### Pseudo-element (phần tử giả)

Phần tử giả giúp bạn định kiểu cho **các phần cụ thể bên trong một phần tử** hoặc để **thêm nội dung trang trí** mà không cần sửa đổi HTML.

|  |  |
| --- | --- |
| **Selector** | **Ý nghĩa** |
| ::before | Thêm nội dung **trước** phần tử |
| ::after | Thêm nội dung **sau** phần tử |
| ::first-line | Áp dụng cho dòng đầu tiên của đoạn văn |
| ::first-letter | Áp dụng cho chữ cái đầu của đoạn văn |

Bảng 1.6 Các selector cho phần tử giả

#### Lưu ý sử dụng Phần tử giả

* Bắt buộc phải có thuộc tính content khi sử dụng phần tử giả.

::before và ::after chỉ hoạt động trên các phần tử có thể chứa nội dung (ví dụ: div, p, span, a, button). Chúng không hoạt động trên các phần tử rỗng (<img>, <input>, <br>).

* Không thêm vào DOM thực. Vì chỉ tạo phần tử giả nên nó không thêm thẻ html thật nên ko tạo ra DOM.

### Attribute selector (theo thuộc tính)

|  |  |
| --- | --- |
| **Cú pháp** | **Ý nghĩa** |
| [type] | Chọn phần tử có thuộc tính type |
| [type="text"] | Chọn phần tử có type=text |
| [href^="https"] | Bắt đầu bằng |
| [href$=".pdf"] | Kết thúc bằng |
| [href\*="google"] | Chứa từ "google" trong giá trị thuộc tính |

Bảng 1.7 Các selector theo thuộc tính

## Sepecificity

**Specificity** (độ ưu tiên của bộ chọn) là cơ chế mà **trình duyệt dùng để quyết định style nào sẽ được áp dụng**, khi có **nhiều rule CSS cùng áp dụng cho một phần tử HTML**.

|  |  |  |
| --- | --- | --- |
| **Nhóm** | **Loại selector** | **Giá trị tăng thêm** |
| a | Inline style | +1000 |
| b | ID selector (#id) | +100 |
| c | Class, attribute, pseudo-class (.class, [type], :hover) | +10 |
| d | Phần tử, pseudo-element (div, p, ::before) | +1 |

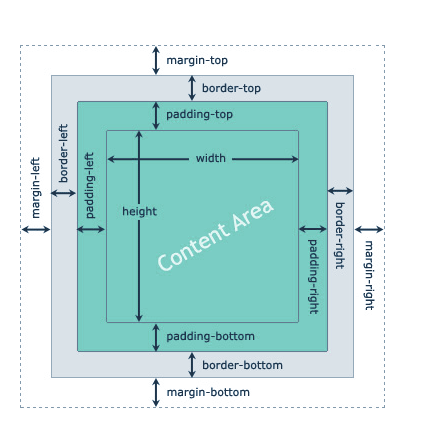
Bảng 1.8 Thứ tự ưu tiên của các thành phần css

### Lưu ý

**Tránh lạm dụng ID**: Bộ chọn ID có specificity cao, có thể gây khó khăn khi cần ghi đè (override) quy tắc.

**Sử dụng class thay vì ID**: Class có specificity thấp hơn, dễ bảo trì và tái sử dụng hơn.

## Box Model



Hình 1.2 Sơ đồ minh họa box model

# Layout

Lưu ý: Các kiến thức cần nắm

**FLEXBOX**: Rất quan trọng trong thiết kế hiện đại. Nên thực hành tạo layout đơn giản như navbar, card...

**GRID:** Khuyến khích học sau Flexbox, dùng cho layout phức tạp hơn như gallery, dashboard...

## Flex box

## GRID

# Responsive Design

Lưu ý: Các kiến thức cần nắm

**MEDIA QUERIES:** Cần học rõ cú pháp, breakpoints phổ biến.

**MOBILE FIRST:** Nên giải thích rõ triết lý "mobile-first" là gì (viết CSS cho màn hình nhỏ trước, sau đó mở rộng dần).

## Media queries

## Mobile first

# Animation

Lưu ý kiến thức cần nắm:

**KEYFRAMES:** Cần minh họa với ví dụ sinh động như loading spinner, fade in/out.

**TRANSITIONS:** Nên học rõ các thuộc tính có thể transition và cách kiểm soát timing (ease-in, ease-out, cubic-bezier, v.v.)

## Keyframs

## Trasitions

# Tìm hiểu về css framework

Lưu ý kiến thức cần nắm:

**TAILWINDCSS:** Rất phù hợp, nhưng cần đảm bảo bạn đã hiểu CSS thuần trước.

* Nên học: cấu trúc class tiện ích, cách cấu hình, responsive trong Tailwind
* Có thể mở rộng thêm Bootstrap để so sánh cách tiếp cận giữa "utility-first" và "component-based"

## Cơ bản về tailwindCSS.