# **CÁC LOẠI BỘ CHỌN CSS**

**✅ 1. Bộ chọn cơ bản (Basic Selectors)**

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Cách viết | Ý nghĩa | Ví dụ |
| Bộ chọn thẻ (Element) | p | Chọn tất cả thẻ HTML cùng loại | p { color: red; } – tất cả thẻ <p> |
| Bộ chọn lớp (Class) | .ten-lop | Chọn tất cả phần tử có class tương ứng | .menu { font-size: 16px; } |
| Bộ chọn ID | #ten-id | Chọn phần tử có id duy nhất | #header { background: gray; } |
| Bộ chọn toàn cục | \* | Chọn tất cả phần tử trên trang | \* { margin: 0; padding: 0; } |
| element1, element2 | h1, p | Chọn nhiều phần tử khác nhau | h1, p {color: red;} |

**✅ 2. Bộ chọn kết hợp (Combinator Selectors)**

|  |  |  |
| --- | --- | --- |
| Cách viết | Ý nghĩa | Ví dụ |
| A B (descendant) | Chọn tất cả phần tử B nằm **bên trong** A (con cháu) | div p – chọn tất cả thẻ p trong div |
| A > B (child) | Chọn phần tử B là **con trực tiếp** của A | ul > li – chọn li là con trực tiếp của ul |
| A + B (adjacent sibling) | Chọn phần tử B ngay **sau A** (anh em liền kề) | h1 + p – chọn p đứng ngay sau h1 |
| A ~ B (general sibling) | Chọn tất cả B là **anh em sau A** (cùng cấp) | h1 ~ p – tất cả p cùng cấp sau h1 |

**✅ Các bộ chọn kết hợp CSS và bản dịch chuẩn:**

|  |  |  |  |
| --- | --- | --- | --- |
| Tên tiếng Anh | Ký hiệu | Dịch chuẩn tiếng Việt | Giải thích ngắn gọn |
| Descendant selector | A B | **Bộ chọn hậu duệ** | Chọn mọi phần tử B nằm **bên trong** (mọi cấp) phần tử A |
| Child selector | A > B | **Bộ chọn con trực tiếp** | Chọn phần tử B là **con trực tiếp** của A (không qua trung gian) |
| Adjacent sibling selector | A + B | **Bộ chọn anh em liền kề** | Chọn phần tử B nằm **ngay sau** A và cùng cấp cha |
| General sibling selector | A ~ B | **Bộ chọn anh em chung cấp** | Chọn **mọi phần tử B** nằm **sau A** và **có chung cha**, không cần liền kề |

Đúng rồi! 👏 Bạn nhớ rất chính xác — và đó **không phải là combinator**, nhưng lại là **một kiểu bộ chọn rất phổ biến** trong CSS:

**✅ AB – Selector ghép (compound selector)**

Khi bạn viết **hai selector sát nhau, không có khoảng trắng**, như:  
👉 div.box, a.button, h1.title, input[type="text"]

**➤ Nghĩa: chọn phần tử thỏa mãn tất cả điều kiện liền kề.**

**🎯 Một số ví dụ cụ thể:**

| **Cú pháp** | **Ý nghĩa** |
| --- | --- |
| div.box | Chọn thẻ div có class box |
| .card.highlight | Chọn phần tử có **cả hai class** card và highlight |
| input[type="text"] | Chọn thẻ input có thuộc tính type="text" |
| a.button:focus | Chọn thẻ a có class button **khi được focus** |
| h2.warning.red | Chọn thẻ h2 có **cả hai class** warning và red |

**📌 Tên kỹ thuật:**

* CSS gọi kiểu này là **compound selector** – **bộ chọn phức hợp**.
* **Không phải combinator**, vì **không diễn tả quan hệ giữa các phần tử**, mà là **tập hợp điều kiện trên cùng một phần tử**.

**📎 Phân biệt:**

| **Kiểu** | **Ví dụ** | **Ý nghĩa** |
| --- | --- | --- |
| **Combinator** | div .box | .box nằm bên trong div |
| **Compound** | div.box | Là div và có class box |
| **Multiple class** | .a.b | Có cả class a và b |
| **Element + pseudo-class** | a:hover | Là thẻ a và đang được hover |

**✅ Tóm lại:**

* ✔️ **Bạn hoàn toàn đúng**: AB (liền nhau, không khoảng trắng) là một dạng **compound selector**.
* ❌ Nó **không phải combinator**, vì combinator phải có **khoảng trắng hoặc ký hiệu** (>, +, ~).
* ✅ Nhưng nó rất quan trọng và thường dùng để **chọn phần tử theo nhiều tiêu chí cùng lúc**.

Nếu bạn muốn mình tổng hợp lại **bảng phân biệt combinator vs. compound vs. group selector**, mình có thể làm gọn cho bạn trong 1 bảng.

**✅ 3. Bộ chọn nhóm (Group Selector)**

|  |  |  |
| --- | --- | --- |
| Cách viết | Ý nghĩa | Ví dụ |
| A, B, C | Áp dụng cùng style cho nhiều selector | h1, h2, h3 { color: blue; } |

**✅ 4. Bộ chọn thuộc tính (Attribute Selectors)**

|  |  |  |
| --- | --- | --- |
| Cách viết | Ý nghĩa | Ví dụ |
| [attr] | Chọn phần tử có thuộc tính đó | input[type] – tất cả input có thuộc tính type |
| [attr="value"] | Có giá trị chính xác | input[type="text"] |
| [attr~="value"] | Chứa giá trị (từ) trong chuỗi cách nhau bằng khoảng trắng | [title~="hoa"] |
| `[attr | ="value"]` | Giá trị bắt đầu bằng từ và có thể theo sau bởi - |
| [attr^="value"] | Bắt đầu bằng giá trị | a[href^="https"] |
| [attr$="value"] | Kết thúc bằng giá trị | img[src$=".jpg"] |
| [attr\*="value"] | Chứa chuỗi con bất kỳ | a[href\*="google"] |

|  |  |
| --- | --- |
| Selector | Mô tả |
| [attr] | Phần tử có thuộc tính attr |
| [attr=value] | Phần tử có attr với giá trị bằng value |
| [attr~=value] | attr chứa value như một từ riêng biệt |
| `[attr | =value]` |
| [attr^=value] | attr bắt đầu bằng value |
| [attr$=value] | attr kết thúc bằng value |
| [attr\*=value] | attr chứa chuỗi value |

**✅ 5. Bộ chọn giả lớp (Pseudo-class Selectors)**

|  |  |  |
| --- | --- | --- |
| Selector | Ý nghĩa | Ví dụ |
| :hover | Khi di chuột vào | a:hover { color: red; } |
| :active | Khi đang click | button:active { background: yellow; } |
| :focus | Khi phần tử được focus (ví dụ khi nhập vào ô input) | input:focus |
| :first-child | Là phần tử con đầu tiên | p:first-child |
| :last-child | Là phần tử con cuối cùng | p:last-child |
| :nth-child(n) | Là phần tử thứ n | li:nth-child(2) |
| :nth-of-type(n) | Phần tử thứ n cùng loại | p:nth-of-type(2) |
| :not(selector) | Loại trừ selector | div:not(.active) |
| :checked | Phần tử được chọn (checkbox/radio) | input:checked |
| :disabled / :enabled | Phần tử bị vô hiệu / hoạt động | button:disabled |

|  |  |
| --- | --- |
| Selector | Mô tả |
| :hover, :focus, :active | Trạng thái tương tác |
| :first-child, :last-child, :nth-child(n) | Vị trí con trong cây DOM |
| :nth-last-child(n) | Như nth-child nhưng từ cuối |
| :only-child | Là con duy nhất |
| :first-of-type, :last-of-type, :nth-of-type(n) | Theo loại phần tử |
| :only-of-type | Là phần tử duy nhất của kiểu đó |
| :empty | Không có con |
| :not(selector) | Không phải là selector |
| :is(selector) | (CSS4) Chọn nếu **là một trong các selector** |
| :where(selector) | Như :is, nhưng độ ưu tiên là 0 |
| :has(selector) | (CSS4) Chọn phần tử chứa selector con |
| :root | Phần tử gốc (thường là html) |
| :lang(lang) | Theo ngôn ngữ |
| `:dir(ltr | rtl)` |

**✅ 6. Bộ chọn giả phần tử (Pseudo-element Selectors)**

|  |  |  |
| --- | --- | --- |
| Selector | Ý nghĩa | Ví dụ |
| ::before | Thêm nội dung trước phần tử | p::before { content: "→ "; } |
| ::after | Thêm nội dung sau phần tử | p::after { content: " ✔"; } |
| ::first-letter | Chọn chữ cái đầu | p::first-letter { font-size: 200%; } |
| ::first-line | Chọn dòng đầu tiên | p::first-line { color: green; } |
| ::selection | Khi bôi đen văn bản | ::selection { background: yellow; } |

|  |  |
| --- | --- |
| Selector | Mô tả |
| ::before | Trước nội dung phần tử |
| ::after | Sau nội dung phần tử |
| ::first-letter | Ký tự đầu tiên |
| ::first-line | Dòng đầu tiên |
| ::selection | Phần văn bản được chọn |
| ::placeholder | Placeholder trong input |
| ::marker | Bullet/number trong list |
| ::file-selector-button | Nút chọn file trong <input type="file"> |

**✅ 7. Bộ chọn đặc biệt khác (Advanced/Modern Selectors)**

|  |  |  |
| --- | --- | --- |
| Selector | Ý nghĩa | Ví dụ |
| :is() | Rút gọn nhiều selector | :is(h1, h2, h3) { color: red; } |
| :where() | Giống :is() nhưng không có ưu tiên (specificity = 0) | :where(article, aside) { margin: 0; } |
| :has() | Chọn phần tử cha **nếu có con phù hợp** *(chỉ hỗ trợ trong trình duyệt hiện đại)* | div:has(img) |
| :nth-last-child() | Giống :nth-child() nhưng từ dưới lên | li:nth-last-child(1) |
| :empty | Không có nội dung con | p:empty |
| :root | Chọn phần tử gốc (<html>) | :root { --main-color: blue; } |

**✅ 6. Bộ chọn nâng cao (CSS4 & đề xuất)**

| **Selector** | **Mô tả** |
| --- | --- |
| :matches() → giờ là :is() | Tập hợp lựa chọn |
| :nth-child(2n of .class) | (CSS4) Chọn phần tử theo nhóm con có điều kiện |
| :has() | Chọn phần tử chứa phần tử con phù hợp |
| :not(:has(...)) | Loại trừ phần tử có phần tử con cụ thể |
| :host | (Web Components) chọn phần tử chứa shadow DOM |
| :host-context() | (Web Components) chọn theo ngữ cảnh |
| ::part(name) | Truy cập phần được đánh dấu part trong shadow DOM |
| ::slotted(selector) | Truy cập phần tử trong slot |
| :defined | Phần tử custom đã được xác định |

**✅ 7. Bộ chọn đặc biệt (Custom & Scoped CSS)**

| **Selector** | **Mô tả** |
| --- | --- |
| @scope (CSS Scoping) | Phạm vi áp dụng CSS |
| :scope | Selector cho phần tử gốc trong context |
| :nth-child(odd/even of selector) | (CSS4) nâng cao cho nhóm con cụ thể |

**🔍 Tổng hợp nhanh các bộ selector CSS hiện đại:**

* **Bộ chọn cơ bản**: \*, .class, #id, element
* **Tổ hợp**: >, +, ~, khoảng trắng
* **Thuộc tính**: [attr], [attr^=], v.v.
* **Giả lớp**: :hover, :not(), :has(), :is(), :where(), :nth-child(), ...
* **Giả phần tử**: ::before, ::after, ...
* **Shadow DOM/Web Component**: :host, ::part, ::slotted
* **Scoped CSS**: @scope, :scope

Nếu bạn cần **minh họa cụ thể** cho từng loại hoặc **ví dụ thực tế** sử dụng, mình có thể cung cấp chi tiết hơn nữa.

**✅ Tổng kết dạng sơ đồ (nhớ nhanh):**

less

Sao chépChỉnh sửa

1. Basic: \*, div, .class, #id

2. Group: div, p, h1

3. Combinators: A B, A > B, A + B, A ~ B

4. Attributes: [attr], [attr=value], [attr^=val]

5. Pseudo-class: :hover, :nth-child()

6. Pseudo-elem: ::before, ::after

7. Advanced: :is(), :has(), :where(), :root

Nếu bạn muốn ví dụ trực quan hơn (có HTML + CSS mẫu), mình có thể bổ sung ngay nhé!

ôi từng thấy ở đâu đó kiểu bộ chọn này: ":link"

**ChatGPT đã nói:**

Đúng vậy! Bộ chọn :link **hoàn toàn hợp lệ** và là một **giả lớp (pseudo-class)** trong CSS – tuy không phổ biến như :hover hay :visited, nhưng nó vẫn là một phần của chuẩn CSS từ **rất sớm (CSS1)**.

**✅ :link là gì?**

* :link chọn **các liên kết (thẻ <a>) chưa được người dùng truy cập** (chưa click).
* Nó **chỉ áp dụng cho các phần tử có thuộc tính href**, như <a>, <area>, hoặc bất kỳ phần tử nào có thể là liên kết.

**🔸 Cách hoạt động:**

css

Sao chépChỉnh sửa

a:link {

color: blue;

}

a:visited {

color: purple;

}

* a:link → liên kết chưa được truy cập
* a:visited → liên kết đã truy cập

**📌 Lưu ý quan trọng:**

1. :link **chỉ chọn các liên kết chưa truy cập**, không áp dụng cho a không có href.
2. Nếu bạn muốn xử lý toàn bộ liên kết, nên dùng theo thứ tự **LVHA**:  
   :link, :visited, :hover, :active
3. Trình duyệt thường **giới hạn khả năng tùy biến :visited vì lý do bảo mật**, nhưng :link thì không bị giới hạn như vậy.

**✅ Tóm tắt:**

* :link là một **pseudo-class chính thức của CSS**
* Là một trong những bộ chọn **cũ nhất**, xuất hiện từ **CSS1**
* Dùng để chọn liên kết **chưa được truy cập**
* Kết hợp tốt với :visited, :hover, :active để tạo hiệu ứng liên kết đầy đủ

Nếu bạn cần ví dụ hoặc cách sử dụng cụ thể, mình có thể viết demo HTML/CSS để bạn xem hiệu quả của :link.

**Bạn đã nói:**

còn loại bộ chọn nào bạn chưa liệt kê không?

**ChatGPT đã nói:**

Câu hỏi rất sắc sảo – cảm ơn bạn! Mình đã liệt kê **gần như đầy đủ** các loại bộ chọn CSS hiện đại, nhưng để **đảm bảo 100% không bỏ sót**, dưới đây là **một lượt kiểm tra bổ sung** để tìm ra bất kỳ loại nào **chưa được đề cập** trong danh sách trước.

**✅ Kiểm tra lại và bổ sung các bộ chọn CSS có thể bị bỏ sót:**

**🔹 1. Bộ chọn :target (❗ chưa liệt kê)**

* ✅ **Đã bỏ sót**.
* Chọn phần tử được liên kết qua anchor (#id) trong URL.

css

Sao chépChỉnh sửa

:target {

background-color: yellow;

}

Ví dụ: example.com/page#section1 sẽ làm <div id="section1"> bị chọn.

**🔹 2. Bộ chọn :checked, :enabled, :disabled, :required, :optional, :valid, :invalid, :in-range, :out-of-range, :read-only, :read-write, :default, :indeterminate (❗ chưa liệt kê chi tiết)**

* ✅ **Bị thiếu nhóm pseudo-class cho biểu mẫu (form)**.

css

Sao chépChỉnh sửa

input:checked {}

input:enabled {}

input:disabled {}

input:valid {}

input:invalid {}

Các bộ chọn này rất quan trọng trong thao tác với biểu mẫu (<input>, <textarea>, v.v.)

**🔹 3. Bộ chọn :nth-last-of-type() (✅ đã liệt kê dạng tương đương)**

* Đã có trong nhóm pseudo-classes. Không thiếu.

**🔹 4. Bộ chọn :user-invalid (ít phổ biến, experimental)**

* Chọn phần tử form mà **người dùng làm sai (client-side)**, nhưng không nằm trong :invalid ngay từ đầu.

css

Sao chépChỉnh sửa

input:user-invalid {

border-color: red;

}

⚠ Experimental – không phải trình duyệt nào cũng hỗ trợ.

**🔹 5. CSS Container Queries selectors (mới, ví dụ @container, :container)**

* ✅ Thuộc **CSS Container Queries**, không phải bộ chọn trực tiếp, nhưng có liên quan.

css

Sao chépChỉnh sửa

@container (min-width: 400px) {

.box {

background: red;

}

}

Đây là **at-rule**, không phải selector trực tiếp, nhưng đáng biết.

**🔹 6. Bộ chọn :modal (CSS Selectors Level 4 – mới)**

* Chọn phần tử đang là **modal dialog đang được hiển thị** (<dialog>)

css

Sao chépChỉnh sửa

dialog:modal {

border: 2px solid red;

}

Còn mới, chưa phổ biến, nhưng là một pseudo-class hợp lệ.

**🔹 7. Bộ chọn :nth-child(An+B of selector) (CSS4)**

* Đã được đề cập ngắn gọn ở mục nâng cao, nhưng chưa mở rộng ví dụ chi tiết.

css

Sao chépChỉnh sửa

li:nth-child(2n of .active) {

color: green;

}

**✅ Tổng kết các bộ chọn CSS bị thiếu hoặc ít nhắc đến:**

| **Loại bộ chọn bị thiếu hoặc chưa chi tiết** | **Trạng thái** |
| --- | --- |
| :target | ❌ Thiếu |
| :checked, :enabled, :disabled, v.v. | ❌ Thiếu nhóm |
| :user-invalid | ⚠ Experimental |
| :modal | ⚠ Mới, ít dùng |
| @container, :container | 🔸 Không phải selector trực tiếp |
| :default, :in-range, :out-of-range | ❌ Thiếu nhóm form |
| :read-only, :read-write | ❌ Thiếu |

Nếu bạn muốn mình cung cấp bản **tổng hợp đầy đủ (kèm ví dụ) ở dạng bảng HTML hoặc PDF**, mình có thể làm luôn.

Bạn có muốn mình cập nhật lại **toàn bộ danh sách hoàn chỉnh**, kể cả các selector vừa bổ sung này không?

**Bạn đã nói:**

(descendant) (child) (adjacent sibling) (general sibling) dịch chuẩn sang tiếng Việt nghĩa là gì?

**ChatGPT đã nói:**

Đây là bản dịch **chuẩn và chính xác** các bộ chọn kết hợp (combinators) trong CSS sang tiếng Việt, cả về nghĩa đen lẫn nghĩa chuyên ngành:

**✅ Các bộ chọn kết hợp CSS và bản dịch chuẩn:**

| **Tên tiếng Anh** | **Ký hiệu** | **Dịch chuẩn tiếng Việt** | **Giải thích ngắn gọn** |
| --- | --- | --- | --- |
| **Descendant selector** | A B | **Bộ chọn hậu duệ** | Chọn mọi phần tử B nằm **bên trong** (mọi cấp) phần tử A |
| **Child selector** | A > B | **Bộ chọn con trực tiếp** | Chọn phần tử B là **con trực tiếp** của A (không qua trung gian) |
| **Adjacent sibling selector** | A + B | **Bộ chọn anh em liền kề** | Chọn phần tử B nằm **ngay sau** A và cùng cấp cha |
| **General sibling selector** | A ~ B | **Bộ chọn anh em chung cấp** | Chọn **mọi phần tử B** nằm **sau A** và **có chung cha**, không cần liền kề |

**🔍 Minh họa nhanh:**

Giả sử có HTML sau:

html

Sao chépChỉnh sửa

<div class="A">

<p>Đây là P1</p>

<span>B1</span>

</div>

<span>B2</span>

<p>B3</p>

* div span → chọn mọi span **bên trong div** → B1
* div > span → chỉ chọn span **là con trực tiếp** của div → B1
* div + span → chọn span **ngay sau div** → B2
* div ~ p → chọn mọi p **sau div**, cùng cấp → B3

**📌 Tóm tắt:**

| **Ký hiệu** | **Tên thường gọi** | **Dịch đúng chuyên ngành** |
| --- | --- | --- |
| A B | Descendant selector | Bộ chọn hậu duệ |
| A > B | Child selector | Bộ chọn con trực tiếp |
| A + B | Adjacent sibling | Bộ chọn anh em liền kề |
| A ~ B | General sibling | Bộ chọn anh em chung cấp |

Nếu bạn cần hình ảnh minh họa hoặc bài tập nhỏ để phân biệt các loại selector này, mình có thể tạo ngay.

# **A. BASIC SELECTORS**

## **1. ELEMENT SELECTOR**

Tuyệt vời! Dưới đây là **phân tích toàn diện, dễ hiểu, chi tiết nhất về "Bộ chọn thẻ (Tag Selector)"** trong CSS dành cho **người không chuyên về kỹ thuật (non-tech)**. Bạn không cần biết lập trình vẫn có thể nắm được kiến thức này một cách rõ ràng.

**🌟 1. "Bộ chọn thẻ (Tag Selector)" là gì?**

**Bộ chọn thẻ** (còn gọi là **Element Selector** hoặc **Type Selector**) trong CSS là cách để bạn **áp dụng kiểu dáng (style)** cho **tất cả các thẻ HTML** có cùng tên trên trang web.

**🔤 Cú pháp:**

**tên\_thẻ {**

**thuộc\_tính: giá\_trị;**

**}**

**📌 Ví dụ đơn giản:**

**p {**

**color: red;**

**font-size: 16px;**

**}**

🔍 Ý nghĩa:

Tất cả các đoạn văn (thẻ <p>) trên website sẽ có chữ **màu đỏ** và **cỡ chữ 16px**.

**🧠 2. Tại sao cần dùng bộ chọn thẻ?**

* Để **định dạng hàng loạt** phần tử cùng loại mà không cần gắn thêm class hay ID.
* Giúp bạn kiểm soát giao diện của toàn bộ các phần tử HTML như: p, h1, ul, li, a, img, div, table, v.v.

**📚 3. Ví dụ thực tế dễ hiểu**

**✅ HTML:**

**<h1>Tiêu đề</h1>**

**<p>Đây là đoạn văn đầu tiên.</p>**

**<p>Đây là đoạn văn thứ hai.</p>**

**✅ CSS:**

**p {**

**color: blue;**

**line-height: 1.6;**

**}**

📌 Kết quả:

* Cả 2 thẻ <p> sẽ **có màu chữ xanh** và **giãn dòng 1.6**.

**🧩 4. Tính chất của bộ chọn thẻ**

|  |  |
| --- | --- |
| Thuộc tính | Mô tả |
| Chọn tất cả | Áp dụng cho tất cả thẻ cùng loại (không phân biệt ở đâu trong trang) |
| Không cần class hay id | Không cần thêm gì vào HTML – rất tiện cho những kiểu chung |
| Độ ưu tiên thấp | Có thể bị ghi đè bởi các selector khác như .class, #id, hoặc inline style |
| Rất phổ biến | Dùng để định dạng mặc định cho website, ví dụ: mọi đoạn văn, tiêu đề, danh sách... |

**⚠️ 5. Lưu ý quan trọng khi dùng bộ chọn thẻ**

**❗ 1. Đừng lạm dụng cho tất cả style**

* Nếu bạn muốn style **một vài phần tử riêng biệt**, dùng **.class hoặc #id** là tốt hơn.
* Dùng tag selector cho **định dạng chung** như:

body { font-family: Arial; }

h1, h2 { margin-bottom: 10px; }

**❗ 2. Sẽ áp dụng lên tất cả thẻ cùng loại**

* Nếu bạn viết:

/

p { color: red; }

→ TẤT CẢ đoạn văn sẽ có chữ màu đỏ, kể cả bạn không mong muốn.

✅ Cách khắc phục: thêm class riêng để kiểm soát:

css

Sao chépChỉnh sửa

.no-red { color: black; }

**❗ 3. Độ ưu tiên thấp (Specificity thấp)**

* Nếu có CSS như sau:

css

Sao chépChỉnh sửa

p { color: blue; }

.highlight { color: orange; }

<p class="highlight">...</p>

→ Màu **orange** được ưu tiên vì .class mạnh hơn tag.

**❗ 4. Có thể bị reset bởi CSS khác**

* Một số framework như Bootstrap có thể định dạng lại sẵn (reset CSS), nên tag selector của bạn có thể không có tác dụng nếu không ghi đè đúng.

**🔧 6. Kết hợp với các selector khác (nâng cao dần)**

|  |  |  |
| --- | --- | --- |
| Cú pháp | Ý nghĩa | Ví dụ |
| div p | Chọn thẻ p nằm trong thẻ div | Chỉ áp dụng cho đoạn văn bên trong vùng div |
| ul li | Chọn thẻ li nằm trong ul | Định dạng mục danh sách |
| section h2 | Chọn h2 trong section | Thường dùng trong bố cục trang |

**✅ 7. Khi nào nên dùng bộ chọn thẻ?**

|  |  |
| --- | --- |
| Tình huống | Nên dùng? |
| Muốn định dạng toàn bộ thẻ giống nhau | ✅ Có |
| Muốn thay đổi giao diện phần tử toàn cục | ✅ Có |
| Muốn style một phần tử cụ thể | ❌ Không – dùng .class hoặc #id |
| Trong dự án lớn, cần tách biệt từng vùng | ❌ Không – nên dùng selector rõ ràng hơn |

**🧾 8. Tổng kết dễ nhớ:**

|  |  |
| --- | --- |
| Ưu điểm | Nhược điểm |
| Ngắn gọn, dễ viết | Áp dụng hết, không phân biệt vùng |
| Tốt cho style mặc định | Khó kiểm soát nếu trang phức tạp |
| Dễ hiểu với người mới | Độ ưu tiên thấp, dễ bị ghi đè |

**📌 Tóm gọn trong 1 câu:**

**Bộ chọn thẻ (Tag Selector)** là cách đơn giản để áp dụng kiểu cho tất cả phần tử HTML cùng loại, tốt cho style mặc định, nhưng cần dùng cẩn thận để tránh mất kiểm soát giao diện.

**🧾 Ví dụ: Định dạng các đoạn văn <p> và tiêu đề <h1>**

**✅ HTML (nội dung trang web)**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html>

<head>

<title>Ví dụ về bộ chọn thẻ</title>

<style>

/\* Đây là CSS với bộ chọn thẻ \*/

/\* Áp dụng cho tất cả thẻ <p> \*/

p {

color: blue;

font-size: 18px;

line-height: 1.5;

}

/\* Áp dụng cho tất cả thẻ <h1> \*/

h1 {

color: darkgreen;

text-align: center;

}

</style>

</head>

<body>

<h1>Chào mừng bạn!</h1>

<p>Đây là đoạn văn đầu tiên. Nó sẽ được hiển thị với màu xanh và cỡ chữ 18px.</p>

<p>Đây là đoạn văn thứ hai. Nó cũng sẽ có cùng kiểu định dạng như đoạn trên.</p>

</body>

</html>

**🔍 Giải thích từng phần:**

* p { ... } → Tất cả các thẻ <p> trong tài liệu HTML này sẽ có:
  + Màu chữ **xanh dương**
  + Cỡ chữ **18px**
  + Dòng cách nhau 1.5 lần
* h1 { ... } → Tất cả các tiêu đề h1 (ở đây chỉ có 1 cái) sẽ có:
  + Màu chữ **xanh lá đậm**
  + Căn giữa

**🖼️ Kết quả hiển thị khi mở trên trình duyệt:**

css

Sao chépChỉnh sửa

Chào mừng bạn! ← (màu xanh lá, căn giữa, h1)

Đây là đoạn văn đầu tiên. ← (màu xanh, chữ to hơn bình thường, dòng cách thoáng)

Đây là đoạn văn thứ hai. ← (giống dòng trên)

Nếu bạn muốn mình tạo bản demo online trên [CodePen](https://codepen.io/) hoặc gửi file .html để bạn mở trên máy, mình cũng có thể làm giúp nhé. Bạn cần mình làm tiếp không? 😄

## **2. CLASS SELECTOR**

**🎯 1. "Bộ chọn lớp (Class Selector)" là gì?**

**Bộ chọn lớp** trong CSS là cách để **gán một kiểu (style)** cho **một nhóm phần tử** HTML mà bạn muốn chúng **trông giống nhau**.

📌 Nói đơn giản:  
Bạn đặt “nhãn” (gọi là class) cho các phần tử HTML, rồi dùng CSS để định nghĩa kiểu dáng cho những phần tử có nhãn đó.

**🧪 2. Cú pháp**

**✅ Trong HTML:**

html

Sao chépChỉnh sửa

<p class="highlight">Đoạn văn nổi bật</p>

Thêm thuộc tính class="highlight" vào thẻ HTML. Đây là “nhãn” tên là highlight.

**✅ Trong CSS:**

css

Sao chépChỉnh sửa

.highlight {

color: red;

font-weight: bold;

}

Bắt đầu bằng dấu . (dấu chấm) rồi tên lớp. Phần tử nào có class="highlight" sẽ có chữ **đỏ và đậm**.

**🧠 3. Lợi ích của class (vì sao class rất quan trọng)**

| **Tính năng** | **Giải thích** |
| --- | --- |
| 🔁 **Tái sử dụng** | Một class có thể dùng cho **nhiều phần tử** – tiết kiệm công sức. |
| 🎯 **Nhắm đúng đối tượng** | Chỉ áp dụng style cho những phần tử bạn muốn, không ảnh hưởng các phần tử khác. |
| 🎨 **Tùy biến giao diện** | Dễ dàng tạo nhiều kiểu dáng khác nhau cho các vùng khác nhau của website. |
| 🧩 **Kết hợp với JavaScript** | Các class rất hữu ích khi cần thay đổi style động bằng JS. |

**📚 4. Ví dụ minh họa trực quan**

**✅ HTML:**

html

Sao chépChỉnh sửa

<p class="note">Đây là một ghi chú quan trọng.</p>

<p class="note">Thêm một ghi chú khác nữa.</p>

**✅ CSS:**

css

Sao chépChỉnh sửa

.note {

background-color: yellow;

padding: 10px;

border: 1px solid orange;

}

📝 Kết quả:

* Hai đoạn văn có **nền vàng**, **viền cam**, **khoảng đệm 10px** vì cả hai có cùng class note.

**🔢 5. Một phần tử có thể có nhiều class?**

👉 **Có!**

**Ví dụ:**

html

Sao chépChỉnh sửa

<p class="note important">Nội dung quan trọng</p>

→ Phần tử này sẽ **kế thừa style từ cả 2 class**: .note và .important.

**🔗 6. Cách kết hợp bộ chọn lớp với các selector khác**

| **CSS** | **Ý nghĩa** |
| --- | --- |
| .menu | Tất cả phần tử có class menu |
| p.menu | Chỉ áp dụng cho thẻ <p> có class menu |
| div.menu-item | Thẻ <div> có class menu-item |
| .menu .item | Chọn phần tử có class item **nằm trong** phần tử có class menu |

**🧩 7. So sánh Class với Tag và ID**

| **Tiêu chí** | **Tag selector (p)** | **Class selector (.note)** | **ID selector (#header)** |
| --- | --- | --- | --- |
| Áp dụng cho | Tất cả thẻ cùng loại | Phần tử có cùng class | Một phần tử duy nhất |
| Có thể dùng lại? | ✅ Có | ✅ Có (nhiều lần) | ❌ Không nên |
| Ký hiệu bắt đầu | không có | . | # |
| Mức độ ưu tiên | Thấp | Trung bình | Cao |

**⚠️ 8. Những lưu ý khi dùng class**

**✅ Đặt tên class rõ ràng, dễ hiểu**

* Không nên dùng tên chung chung như style1, box2
* Nên dùng tên ý nghĩa như warning-message, card-title, menu-item

**✅ Không bắt đầu bằng số**

* Tên class không được bắt đầu bằng số, ví dụ: .123abc là **không hợp lệ**.

**✅ Không trùng với từ khóa HTML**

* Tránh đặt class tên là div, body, html để tránh gây nhầm lẫn.

**✅ Tránh trùng tên class trong các mục khác nhau mà khác style**

* Ví dụ, .title dùng ở phần tin tức và sản phẩm nhưng style lại khác nhau → nên đặt là .news-title và .product-title

**✅ Cẩn thận khi ghi đè**

* Nếu có 2 class cùng style, class **viết sau trong CSS** hoặc có **specificity cao hơn** sẽ được áp dụng.

**📌 9. Ví dụ đầy đủ tích hợp HTML + CSS**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html>

<head>

<style>

.highlight {

color: red;

font-weight: bold;

}

.underline {

text-decoration: underline;

}

p.note {

background-color: #ffffcc;

padding: 10px;

}

</style>

</head>

<body>

<p class="highlight">Đây là đoạn văn nổi bật (chữ đỏ và đậm).</p>

<p class="highlight underline">Đoạn này vừa đỏ, đậm và có gạch chân.</p>

<p class="note">Đây là một ghi chú màu vàng nhạt.</p>

</body>

</html>

**🧾 10. Tóm tắt dễ nhớ về class selector**

| **Đặc điểm chính** | **Mô tả** |
| --- | --- |
| Dùng . để khai báo | .class-name { ... } |
| Gán bằng class="..." trong HTML |  |
| Có thể áp dụng cho nhiều phần tử |  |
| Một phần tử có thể có nhiều class |  |
| Dễ quản lý, dễ tái sử dụng |  |

**🎁 Gợi ý thêm:**

* Muốn luyện kỹ hơn? Hãy thử viết đoạn HTML + CSS nhỏ thay đổi màu nền, font chữ, canh lề bằng cách dùng class.
* Hoặc bạn muốn mình tạo demo tương tác để bạn thực hành? Mình có thể giúp luôn!

Bạn muốn tiếp tục học phần selector nào tiếp theo? 😊

Cho ví dụ minh họa về "Bộ chọn lớp (Class Selector)".

**ChatGPT đã nói:**

Tuyệt vời! Dưới đây là một ví dụ **minh họa chi tiết về "Bộ chọn lớp (Class Selector)"** trong CSS để bạn dễ dàng hiểu và thực hành.

**🎯 Ví dụ về Bộ chọn lớp (Class Selector)**

**✅ HTML:**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ Bộ chọn lớp</title>

<style>

/\* Bộ chọn lớp .highlight \*/

.highlight {

color: red;

font-weight: bold;

}

/\* Bộ chọn lớp .blue-background \*/

.blue-background {

background-color: lightblue;

padding: 10px;

}

/\* Bộ chọn lớp .center-text \*/

.center-text {

text-align: center;

}

</style>

</head>

<body>

<h1 class="highlight center-text">Chào mừng bạn đến với ví dụ CSS!</h1>

<p class="highlight">Đây là đoạn văn đầu tiên. Nó sẽ có chữ màu đỏ và đậm.</p>

<p class="blue-background">Đây là đoạn văn có nền màu xanh nhạt và khoảng đệm xung quanh.</p>

<div class="blue-background">

<p class="highlight">Đoạn văn trong một <code>div</code> có nền xanh nhạt và chữ đỏ, đậm.</p>

</div>

</body>

</html>

**✅ Giải thích chi tiết:**

* **HTML**:
  + Thẻ <h1> có hai lớp: **highlight** và **center-text**. Điều này có nghĩa là đoạn văn này sẽ có **chữ đỏ**, **đậm** và được **canh giữa**.
  + Thẻ <p> đầu tiên có lớp **highlight**, vì vậy nó sẽ có **chữ đỏ** và **đậm**.
  + Thẻ <p> thứ hai có lớp **blue-background**, sẽ có **nền xanh nhạt** và **khoảng đệm** 10px.
  + Thẻ <div> có lớp **blue-background**, làm cho nó có nền xanh nhạt và các phần tử bên trong (đoạn văn <p>) sẽ có lớp **highlight** (chữ đỏ và đậm).
* **CSS**:
  + .highlight: Áp dụng **màu đỏ** cho chữ và **font-weight: bold** (chữ đậm) cho bất kỳ phần tử nào có lớp highlight.
  + .blue-background: Thêm **nền xanh nhạt** và **padding 10px** cho phần tử có lớp blue-background.
  + .center-text: Căn giữa **văn bản** của phần tử có lớp center-text.

**✅ Kết quả hiển thị trên trình duyệt:**

* Tiêu đề **Chào mừng bạn đến với ví dụ CSS!** sẽ có **màu đỏ**, **chữ đậm**, và **canh giữa**.
* Đoạn văn đầu tiên sẽ có **chữ màu đỏ và đậm**.
* Đoạn văn thứ hai sẽ có **nền xanh nhạt** và **khoảng đệm xung quanh**.
* Đoạn văn trong thẻ <div> sẽ có **nền xanh nhạt** và **chữ đỏ đậm**.

**🎯 Tóm tắt về bộ chọn lớp:**

* Bộ chọn lớp trong CSS là cách để bạn **định dạng** các phần tử có cùng **class** mà không cần phải chỉ định từng phần tử riêng biệt.
* Bạn có thể sử dụng một hoặc nhiều lớp cho mỗi phần tử HTML. Việc sử dụng lớp giúp bạn **tái sử dụng** style ở nhiều nơi mà không cần phải lặp lại mã CSS.

Hy vọng bạn đã hiểu rõ hơn về **Bộ chọn lớp (Class Selector)**. Nếu bạn muốn mình giải thích thêm hoặc có bất kỳ câu hỏi nào, đừng ngần ngại nhé! 😄

## **3. ID SELECTOR**

Rất tuyệt, bây giờ chúng ta sẽ đi sâu vào **“Bộ chọn ID (ID Selector)” trong CSS** — một khái niệm quan trọng và nền tảng. Mình sẽ trình bày **toàn diện, chi tiết từ cơ bản đến nâng cao**, đặc biệt **dễ hiểu cho người không chuyên kỹ thuật (non-tech)**. Cùng bắt đầu nhé!

**🎯 1. Bộ chọn ID là gì?**

**Bộ chọn ID (ID Selector)** là cách để áp dụng một kiểu định dạng (CSS style) cho **một phần tử HTML duy nhất** thông qua một **mã định danh duy nhất gọi là ID**.

📌 Hình dung: ID giống như **chứng minh thư** – mỗi phần tử HTML chỉ có **một ID duy nhất**, không trùng nhau trong toàn bộ trang web.

**🧪 2. Cú pháp**

**✅ Trong HTML:**

html

Sao chépChỉnh sửa

<p id="special">Đây là đoạn văn đặc biệt.</p>

Thêm thuộc tính id="special" vào phần tử bạn muốn định dạng.

**✅ Trong CSS:**

css

Sao chépChỉnh sửa

#special {

color: blue;

font-weight: bold;

}

Trong CSS, bạn chọn ID bằng dấu # rồi đến tên ID (không có dấu ngoặc kép).

**🧠 3. Cách hoạt động của ID**

* ID sẽ **chỉ định style cho một phần tử duy nhất**.
* Không nên sử dụng cùng một ID cho nhiều phần tử HTML trên cùng một trang (sẽ gây lỗi hoặc xung đột).
* ID có **mức độ ưu tiên cao** hơn class và tag.

**📚 4. Ví dụ minh họa trực quan**

**✅ HTML:**

html

Sao chépChỉnh sửa

<h1 id="title">Trang Chủ</h1>

<p>Đây là một đoạn văn bình thường.</p>

<p id="highlighted">Đây là đoạn văn đặc biệt.</p>

**✅ CSS:**

css

Sao chépChỉnh sửa

#title {

color: darkgreen;

text-align: center;

}

#highlighted {

color: red;

font-size: 20px;

}

📝 Kết quả:

* Thẻ h1 có ID là title sẽ có chữ **xanh lá đậm**, **căn giữa**.
* Thẻ p có ID highlighted sẽ có chữ **đỏ**, **cỡ chữ 20px**.

**🧩 5. So sánh: ID vs Class vs Tag**

| **Đặc điểm** | **Tag (p)** | **Class (.note)** | **ID (#highlight)** |
| --- | --- | --- | --- |
| Ký hiệu | Không có | . (dấu chấm) | # (dấu thăng) |
| Dùng cho | Tất cả thẻ cùng loại | Nhóm phần tử | Một phần tử duy nhất |
| Có thể tái sử dụng? | ✅ Có | ✅ Có | ❌ Không (nên tránh lặp ID) |
| Độ ưu tiên | Thấp | Trung bình | Cao nhất |

**💡 6. Khi nào nên dùng ID?**

| **Trường hợp** | **Dùng ID?** |
| --- | --- |
| Cần định dạng một phần tử duy nhất | ✅ Có |
| Cần nhắm chính xác phần tử (với JS hoặc liên kết nội bộ) | ✅ Có |
| Muốn áp dụng style cho nhiều phần tử cùng loại | ❌ Không – nên dùng class |
| Trong dự án lớn, cần tái sử dụng kiểu nhiều lần | ❌ Không – ID không nên trùng |

**⚠️ 7. Lưu ý quan trọng khi dùng ID**

**❗ 1. Mỗi ID phải là duy nhất**

Không được dùng cùng một ID cho nhiều phần tử.

Ví dụ **sai**:

html

Sao chépChỉnh sửa

<p id="main">Văn bản 1</p>

<p id="main">Văn bản 2</p> <!-- Không hợp lệ -->

**❗ 2. Không được bắt đầu bằng số**

ID không thể bắt đầu bằng số hoặc chứa khoảng trắng.

Ví dụ **sai**:

css

Sao chépChỉnh sửa

#123abc { } /\* Lỗi \*/

Ví dụ **đúng**:

css

Sao chépChỉnh sửa

#main123 { }

#abc-456 { }

**❗ 3. Độ ưu tiên cao – cẩn thận khi ghi đè**

Style từ ID sẽ ghi đè style từ class hoặc tag nếu áp dụng cùng lúc.

Ví dụ:

css

Sao chépChỉnh sửa

p {

color: black;

}

.note {

color: blue;

}

#note {

color: red;

}

HTML:

html

Sao chépChỉnh sửa

<p class="note" id="note">Xin chào</p>

✅ Màu chữ sẽ là **đỏ** vì #note có độ ưu tiên cao hơn cả .note và p.

**🧾 8. Ứng dụng đặc biệt của ID**

**🔗 1. Dẫn liên kết nội bộ**

Bạn có thể tạo nút hoặc menu để nhảy đến một phần cụ thể trong trang:

html

Sao chépChỉnh sửa

<a href="#about">Giới thiệu</a>

...

<h2 id="about">Phần giới thiệu</h2>

→ Khi nhấn vào liên kết, trình duyệt cuộn xuống phần có ID là about.

**🧾 9. Ví dụ đầy đủ tích hợp HTML + CSS**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html>

<head>

<style>

#title {

color: teal;

text-align: center;

font-size: 30px;

}

#special-note {

background-color: lightyellow;

padding: 15px;

border-left: 5px solid orange;

}

</style>

</head>

<body>

<h1 id="title">Bài viết hôm nay</h1>

<p>Đây là đoạn văn thông thường.</p>

<p id="special-note">Đây là ghi chú đặc biệt dành cho bạn!</p>

</body>

</html>

**✅ 10. Tóm tắt nhanh – Bộ chọn ID là gì?**

| **Tên** | **Ký hiệu** | **Mục đích chính** |
| --- | --- | --- |
| **ID Selector** | # | Chọn **một phần tử duy nhất** để áp dụng style đặc biệt |

Dùng ID để tạo **điểm nhấn riêng biệt**, **liên kết nội bộ**, hoặc **xác định phần tử trong JavaScript**.

Nếu bạn muốn mình tạo bản mẫu .html sẵn để bạn mở trực tiếp, hoặc demo trên CodePen, mình có thể hỗ trợ ngay.  
Bạn muốn học tiếp về phần nào: Combinator, Attribute selector, hay Pseudo-class? 😊

Cho ví dụ minh họa về "Bộ chọn ID (ID Selector".

**ChatGPT đã nói:**

Dĩ nhiên! Hãy cùng mình khám phá **ví dụ minh họa chi tiết về "Bộ chọn ID (ID Selector)"** trong CSS, để bạn dễ dàng hiểu và thực hành.

**🎯 Ví dụ về Bộ chọn ID (ID Selector)**

**✅ HTML:**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ Bộ chọn ID</title>

<style>

/\* Bộ chọn ID #header \*/

#header {

background-color: lightcoral;

color: white;

text-align: center;

padding: 20px;

}

/\* Bộ chọn ID #main-content \*/

#main-content {

background-color: lightyellow;

padding: 30px;

}

/\* Bộ chọn ID #footer \*/

#footer {

background-color: lightgray;

text-align: center;

padding: 10px;

position: fixed;

bottom: 0;

width: 100%;

}

</style>

</head>

<body>

<div id="header">

<h1>Chào mừng đến với website của chúng tôi!</h1>

</div>

<div id="main-content">

<p>Đây là phần nội dung chính của trang web.</p>

<p>Đoạn văn này có nền màu vàng nhạt và khoảng đệm xung quanh.</p>

</div>

<div id="footer">

<p>&copy; 2025 Website của chúng tôi</p>

</div>

</body>

</html>

**✅ Giải thích chi tiết:**

* **HTML**:
  + **<div id="header">**: Phần tử này có ID là header, sẽ được định dạng theo kiểu CSS liên quan đến ID #header.
  + **<div id="main-content">**: Phần tử này có ID là main-content, sẽ có kiểu nền màu vàng nhạt và padding xung quanh.
  + **<div id="footer">**: Phần tử này có ID là footer, sẽ có nền màu xám nhạt, căn giữa văn bản và được **gắn cố định** ở dưới cùng của trang.
* **CSS**:
  + **#header**: ID header có các thuộc tính CSS như **nền màu đỏ nhạt**, **chữ trắng**, **căn giữa văn bản** và **padding** xung quanh.
  + **#main-content**: ID main-content có nền **vàng nhạt** và **padding 30px**.
  + **#footer**: ID footer có nền **xám nhạt**, **căn giữa** văn bản và sẽ luôn **được cố định ở dưới cùng của trang** nhờ thuộc tính position: fixed.

**✅ Kết quả hiển thị trên trình duyệt:**

* **Phần tử có ID header**: Sẽ có **nền màu đỏ nhạt**, **chữ trắng**, **căn giữa** và có khoảng đệm xung quanh.
* **Phần tử có ID main-content**: Sẽ có nền **vàng nhạt** và **padding** để tạo không gian xung quanh nội dung.
* **Phần tử có ID footer**: Sẽ có nền **xám nhạt**, **căn giữa** văn bản, và luôn **hiển thị ở dưới cùng của trang**.

**🎯 Tóm tắt về Bộ chọn ID:**

1. **ID Selector** được viết với dấu # theo sau là tên của ID. Mỗi phần tử chỉ nên có **một ID duy nhất** trong trang HTML.
2. **Ưu điểm**:
   * **Định dạng duy nhất cho phần tử cụ thể**: ID giúp bạn nhắm vào đúng phần tử bạn muốn, mà không làm ảnh hưởng đến các phần tử khác.
   * **Độ ưu tiên cao**: Kiểu từ ID có **độ ưu tiên cao** hơn so với class và tag.
3. **Lưu ý**:
   * **Không dùng trùng ID**: Mỗi ID phải là **duy nhất** trong một trang HTML. Không nên sử dụng cùng một ID cho nhiều phần tử.
   * **Tránh lạm dụng ID**: Để CSS dễ bảo trì, bạn chỉ nên sử dụng ID khi cần áp dụng **đặc biệt** cho một phần tử duy nhất.

Hy vọng bạn đã hiểu rõ hơn về **Bộ chọn ID (ID Selector)** trong CSS! Nếu bạn có thêm bất kỳ câu hỏi nào, hoặc muốn mình giải thích thêm ví dụ nào khác, đừng ngần ngại chia sẻ nhé! 😄

## **4. UNIVERSAL SELECTOR**

Hãy giải thích chi tiết tất tần tật, sâu rộng toàn bộ và toàn diện kiến thức về "Bộ chọn toàn cục" và những lưu ý (nếu có) về nó để cho người non-tech hiểu sâu và rõ.

**ChatGPT đã nói:**

Tuyệt vời! Bây giờ chúng ta sẽ cùng khám phá **"Bộ chọn toàn cục (Universal Selector)"** trong CSS, một khái niệm rất quan trọng và cực kỳ hữu ích khi làm việc với trang web. Mình sẽ **giải thích chi tiết, dễ hiểu, và toàn diện** để bạn nắm rõ từ cơ bản đến nâng cao, dù bạn không phải là người chuyên về kỹ thuật. Hãy cùng bắt đầu!

**🎯 1. Bộ chọn toàn cục là gì?**

**Bộ chọn toàn cục** (còn gọi là **Universal Selector**) trong CSS là một cách để **áp dụng kiểu (style)** cho **tất cả các phần tử** trong trang web mà không cần phải chỉ định cụ thể từng loại thẻ HTML.

**🔤 Cú pháp:**

css

Sao chépChỉnh sửa

\* {

thuộc\_tính: giá\_trị;

}

Dấu sao \* là đại diện cho **tất cả các phần tử HTML** trong trang. Khi bạn dùng \*, mọi phần tử trong trang web đều nhận kiểu mà bạn chỉ định.

**🧪 2. Ví dụ cơ bản**

**✅ HTML:**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html>

<head>

<title>Ví dụ Bộ Chọn Toàn Cục</title>

<style>

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

</style>

</head>

<body>

<h1>Chào mừng bạn!</h1>

<p>Đây là đoạn văn trong trang.</p>

<div>

<p>Đoạn văn khác trong một div.</p>

</div>

</body>

</html>

**✅ CSS:**

css

Sao chépChỉnh sửa

\* {

margin: 0; /\* Xóa bỏ khoảng cách ngoài các phần tử \*/

padding: 0; /\* Xóa bỏ khoảng cách trong các phần tử \*/

box-sizing: border-box; /\* Đảm bảo padding và border không làm tăng kích thước phần tử \*/

}

📝 Kết quả:

* **Tất cả phần tử trong trang** sẽ không có khoảng cách (margin, padding) mặc định.
* Đảm bảo kích thước của các phần tử sẽ bao gồm cả border và padding mà không bị phồng lên (do box-sizing).

**🧠 3. Khi nào sử dụng bộ chọn toàn cục?**

**✅ 1. Thiết lập kiểu mặc định cho toàn bộ trang**

Khi bạn muốn tất cả phần tử trên trang web có cùng một kiểu cơ bản, bạn có thể dùng bộ chọn toàn cục để **tạo kiểu mặc định** cho tất cả phần tử.

**Ví dụ**:

css

Sao chépChỉnh sửa

\* {

font-family: Arial, sans-serif;

line-height: 1.5;

}

→ Tất cả các phần tử trên trang web sẽ sử dụng font chữ **Arial** và **dãn cách dòng 1.5**.

**✅ 2. Khôi phục hoặc reset CSS**

Các trình duyệt có kiểu mặc định cho các phần tử (như margin, padding, font-size). Bộ chọn toàn cục thường được dùng để **reset các kiểu mặc định** này trước khi áp dụng style riêng cho dự án.

**Ví dụ**:

css

Sao chépChỉnh sửa

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

→ Đoạn CSS này sẽ giúp bạn loại bỏ **mặc định margin và padding** mà các trình duyệt áp dụng, từ đó tạo nền tảng sạch sẽ cho việc thiết kế.

**🔧 4. Lợi ích của Bộ chọn Toàn cục**

| **Lợi ích** | **Giải thích** |
| --- | --- |
| **Dễ dàng reset CSS** | Giúp loại bỏ tất cả các khoảng cách (margin, padding) mặc định của trình duyệt. |
| **Tiết kiệm thời gian** | Thay vì phải áp dụng kiểu cho từng phần tử cụ thể, bạn chỉ cần dùng \* để áp dụng cho tất cả. |
| **Tạo nền tảng nhất quán** | Đảm bảo mọi phần tử trên trang sẽ có cùng kiểu cơ bản, dễ dàng kiểm soát layout. |
| **Giúp căn chỉnh dễ dàng hơn** | Với box-sizing: border-box, bạn dễ dàng kiểm soát kích thước các phần tử mà không lo padding hay border làm thay đổi kích thước. |

**⚠️ 5. Những lưu ý khi dùng bộ chọn toàn cục**

**❗ 1. Hiệu suất**

* Sử dụng bộ chọn toàn cục sẽ áp dụng style cho **tất cả các phần tử** trong trang. Nếu trang web có nhiều phần tử (ví dụ, hơn 1000 phần tử), điều này có thể **tăng tải** cho trình duyệt và làm giảm **hiệu suất** của trang.
* **Lời khuyên**: Dùng \* với mục đích **reset** hoặc **tạo kiểu chung** cho một số kiểu cơ bản, không nên lạm dụng cho tất cả.

**❗ 2. Khó kiểm soát các phần tử riêng biệt**

* Vì bộ chọn toàn cục áp dụng cho **tất cả các phần tử** trong trang, nên đôi khi bạn sẽ khó kiểm soát được kiểu dáng của những phần tử đặc biệt nếu không sử dụng thêm **class** hoặc **ID**.

**Ví dụ**:

css

Sao chépChỉnh sửa

\* {

color: blue;

}

→ **Mọi phần tử trong trang** sẽ có màu chữ **xanh**, và bạn sẽ phải **điều chỉnh lại** từng phần tử nếu cần màu sắc khác.

**❗ 3. Không nên dùng cho mọi trường hợp**

* Chỉ nên sử dụng **khi cần thiết** như khi bạn muốn tạo kiểu cơ bản chung hoặc **reset** trang. Nếu dùng quá nhiều có thể gây khó khăn cho việc kiểm soát style chi tiết cho các phần tử riêng biệt.

**❗ 4. Không phải tất cả các thuộc tính đều thích hợp**

* Một số thuộc tính như background, border có thể không cần thiết phải áp dụng cho **tất cả các phần tử**. Hãy chắc chắn rằng bạn chỉ áp dụng những thuộc tính có lợi cho **toàn bộ trang**.

**🧩 6. Ví dụ nâng cao về Bộ chọn Toàn cục**

**✅ HTML:**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html>

<head>

<title>Trang Web Demo</title>

<style>

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

font-family: 'Arial', sans-serif;

background-color: #f4f4f4;

}

h1 {

color: #333;

}

p {

margin-bottom: 20px;

}

</style>

</head>

<body>

<h1>Trang Demo</h1>

<p>Đoạn văn này sẽ có margin dưới 20px do thuộc tính riêng của <b>p</b>.</p>

<p>Đoạn văn khác.</p>

</body>

</html>

**✅ Giải thích:**

* Bộ chọn toàn cục \* được dùng để **reset margin và padding** cho tất cả phần tử trong trang.
* Body và h1 có style riêng biệt để thiết lập giao diện trang cho phù hợp.
* Đoạn văn <p> có khoảng cách dưới 20px do CSS riêng của nó.

**🧾 7. Tóm tắt về Bộ chọn Toàn cục**

| **Tính năng** | **Mô tả** |
| --- | --- |
| **Ký hiệu** | \* |
| **Áp dụng cho** | Tất cả phần tử HTML trên trang |
| **Ứng dụng** | Để reset các kiểu mặc định của trình duyệt, tạo kiểu nền tảng chung cho trang |
| **Lưu ý** | Cẩn thận khi dùng với hiệu suất và tránh làm thay đổi kiểu không mong muốn cho các phần tử riêng biệt |

**🎁 Gợi ý thêm:**

* Bạn có muốn mình tạo một file HTML thực hành để bạn xem trực quan hoặc demo trên CodePen không?
* Hoặc bạn muốn tiếp tục khám phá thêm các bộ chọn CSS khác?

Hãy cho mình biết nếu bạn cần thêm bất kỳ trợ giúp nào nhé! 😊

Cho ví dụ minh họa về "Bộ chọn toàn cục (Universal Selector)".

**ChatGPT đã nói:**

Chắc chắn rồi! Hãy cùng mình đi qua một ví dụ **minh họa chi tiết về "Bộ chọn toàn cục (Universal Selector)"** trong CSS để bạn dễ dàng nắm bắt cách sử dụng.

**🎯 Ví dụ về Bộ chọn toàn cục (Universal Selector)**

**✅ HTML:**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ Bộ chọn toàn cục</title>

<style>

/\* Bộ chọn toàn cục (\*) \*/

\* {

margin: 0; /\* Xóa khoảng cách ngoài phần tử \*/

padding: 0; /\* Xóa khoảng cách trong phần tử \*/

box-sizing: border-box; /\* Đảm bảo padding và border không làm thay đổi kích thước phần tử \*/

font-family: Arial, sans-serif; /\* Áp dụng font chữ Arial cho tất cả phần tử \*/

}

/\* Các kiểu riêng biệt cho từng phần tử \*/

h1 {

color: darkblue;

text-align: center;

}

p {

font-size: 16px;

line-height: 1.6;

}

.highlight {

background-color: yellow;

padding: 10px;

}

</style>

</head>

<body>

<h1>Chào mừng bạn đến với website!</h1>

<p>Đây là đoạn văn đầu tiên. Nó sẽ có khoảng cách dòng lớn và cỡ chữ 16px.</p>

<p class="highlight">Đoạn văn này có nền vàng và padding xung quanh.</p>

</body>

</html>

**✅ Giải thích chi tiết:**

* **HTML**:
  + **<h1>**: Tiêu đề chính của trang. Sử dụng bộ chọn toàn cục để loại bỏ margin và padding mặc định.
  + **<p>**: Các đoạn văn trong trang. Bộ chọn toàn cục cũng áp dụng cho các thẻ <p> để xóa padding và margin, sau đó áp dụng kiểu riêng cho từng đoạn văn.
  + **.highlight**: Một lớp (class) để làm nổi bật một đoạn văn với **nền vàng** và **padding** xung quanh.
* **CSS**:
  + **Bộ chọn toàn cục (\*)**:
    - **margin: 0; padding: 0;**: Loại bỏ tất cả margin và padding mặc định của các trình duyệt, giúp giao diện sạch sẽ hơn.
    - **box-sizing: border-box;**: Đảm bảo rằng **border** và **padding** sẽ không làm thay đổi kích thước phần tử, giúp bạn dễ dàng kiểm soát kích thước của các phần tử.
    - **font-family: Arial, sans-serif;**: Áp dụng font chữ **Arial** cho tất cả các phần tử trong trang.
  + **Kiểu riêng cho các phần tử**:
    - **h1**: Màu chữ là **đậm xanh đậm**, và văn bản được **canh giữa**.
    - **p**: Cỡ chữ là **16px**, và **khoảng cách dòng** là 1.6, giúp văn bản dễ đọc hơn.
    - **.highlight**: Đoạn văn có nền **vàng**, và **padding** xung quanh để làm nổi bật nội dung.

**✅ Kết quả hiển thị trên trình duyệt:**

1. Tiêu đề **Chào mừng bạn đến với website!** sẽ có **màu xanh đậm**, **canh giữa** trên trang.
2. Đoạn văn đầu tiên sẽ có **cỡ chữ 16px** và **khoảng cách dòng 1.6**.
3. Đoạn văn thứ hai có **nền vàng** và **padding xung quanh**, làm nổi bật nội dung với lớp highlight.

**🎯 Tóm tắt về Bộ chọn toàn cục (Universal Selector)**

* Bộ chọn toàn cục (\*) được sử dụng để **áp dụng kiểu cho tất cả các phần tử** trong trang web.
* **Ưu điểm**:
  + Tiết kiệm thời gian khi muốn áp dụng kiểu chung cho toàn bộ trang.
  + Thường dùng để **reset CSS** (loại bỏ margin, padding mặc định).
* **Nhược điểm**:
  + Nếu áp dụng quá nhiều, có thể làm giảm **hiệu suất** của trang web, vì nó sẽ ảnh hưởng đến **tất cả các phần tử**.
  + Không thể **chọn riêng lẻ** các phần tử mà bạn muốn có kiểu khác biệt. Bạn nên kết hợp với các bộ chọn khác như class hoặc id.

**🎯 Lưu ý khi sử dụng Bộ chọn toàn cục**

1. **Không lạm dụng**: Bộ chọn toàn cục áp dụng cho tất cả các phần tử, điều này có thể ảnh hưởng đến hiệu suất của trang web, đặc biệt nếu trang có nhiều phần tử.
2. **Sử dụng cho mục đích reset**: Bộ chọn toàn cục là công cụ tuyệt vời khi bạn muốn **reset** hoặc **tạo nền tảng sạch sẽ** trước khi áp dụng kiểu riêng biệt cho các phần tử.

Hy vọng bạn đã hiểu rõ hơn về **Bộ chọn toàn cục (Universal Selector)** trong CSS! Nếu bạn có câu hỏi nào hoặc cần thêm ví dụ khác, đừng ngần ngại chia sẻ nhé! 😄

# **B. BỘ CHỌN KẾT HỢP (COMBINATOR SELECTORS)**

## **1. BỘ CHỌN KẾT HỢP: A B**

Hãy giải thích chi tiết tất tần tật, sâu rộng toàn bộ và toàn diện kiến thức về "Bộ chọn kết hợp: A B" và những lưu ý (nếu có) về nó để cho người non-tech hiểu sâu và rõ.

**ChatGPT đã nói:**

Tuyệt vời! Chúng ta sẽ cùng khám phá **"Bộ chọn kết hợp: A B"** trong CSS, một khái niệm rất hữu ích và mạnh mẽ trong việc thiết kế trang web. Mình sẽ giải thích chi tiết, dễ hiểu từ cơ bản đến nâng cao để bạn có thể nắm rõ ngay cả khi không có nền tảng kỹ thuật.

**🎯 Bộ chọn kết hợp: A B là gì?**

**Bộ chọn kết hợp A B** trong CSS là một cách để **chọn các phần tử B** nằm **bên trong** các phần tử A trong HTML. Nói cách khác, bộ chọn này cho phép bạn **áp dụng kiểu (style)** cho một phần tử (B) mà chỉ khi nó nằm **bên trong** một phần tử khác (A), không cần phải có mối quan hệ cha-con chính thức.

**📜 Cú pháp:**

css

Sao chépChỉnh sửa

A B {

/\* Các thuộc tính CSS \*/

}

Trong đó:

* **A** là phần tử cha (hoặc phần tử chứa)
* **B** là phần tử con mà bạn muốn áp dụng kiểu

Khi bạn viết kiểu như vậy, nó sẽ áp dụng **cho mọi phần tử B nằm bên trong A**, không cần quan tâm đến sự phân cấp hay cấu trúc HTML phức tạp.

**🧪 Ví dụ minh họa về Bộ chọn kết hợp: A B**

**✅ HTML:**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ Bộ chọn kết hợp A B</title>

<style>

/\* Bộ chọn kết hợp div p \*/

div p {

color: green;

font-size: 18px;

}

/\* Bộ chọn kết hợp ul li \*/

ul li {

list-style-type: square; /\* Chấm vuông cho danh sách \*/

color: blue;

}

</style>

</head>

<body>

<div>

<h1>Chào mừng bạn đến với website của chúng tôi!</h1>

<p>Đoạn văn này sẽ có màu xanh lá cây và cỡ chữ 18px.</p>

<p>Đoạn văn này cũng sẽ có màu xanh lá cây và cỡ chữ 18px.</p>

</div>

<ul>

<li>Đây là mục 1 trong danh sách.</li>

<li>Đây là mục 2 trong danh sách.</li>

<li>Đây là mục 3 trong danh sách.</li>

</ul>

</body>

</html>

**✅ Giải thích:**

1. **div p { ... }**:
   * Bộ chọn này áp dụng cho **tất cả các thẻ <p> nằm bên trong thẻ <div>**.
   * Cụ thể, mọi đoạn văn <p> trong thẻ <div> sẽ có **màu chữ xanh lá cây** và **cỡ chữ 18px**.
2. **ul li { ... }**:
   * Bộ chọn này áp dụng cho **tất cả các phần tử <li> bên trong thẻ <ul>**.
   * Các mục trong danh sách (thẻ <li>) sẽ có **dấu chấm vuông** thay vì dấu chấm tròn mặc định của trình duyệt và **màu chữ xanh dương**.

**✅ Kết quả hiển thị trên trình duyệt:**

* **Tất cả các thẻ <p>** trong thẻ <div> sẽ có **màu chữ xanh lá cây** và **cỡ chữ 18px**.
* **Các mục trong danh sách <ul>** sẽ có dấu **chấm vuông** và **màu chữ xanh dương**.

**🧠 Cơ chế hoạt động của Bộ chọn kết hợp: A B**

* Bộ chọn kết hợp A B có thể được hiểu như **"chọn tất cả phần tử B ở bất kỳ đâu trong phần tử A"**.
* Điều này có nghĩa là:
  + **A** có thể là một thẻ HTML (như <div>, <section>, <header>, v.v.), hoặc một class, hoặc một ID.
  + **B** có thể là bất kỳ phần tử nào (như <p>, <h1>, <li>, v.v.).
  + Phần tử **B** sẽ bị chọn nếu nó **nằm trong A**, dù nó có là phần tử con trực tiếp hay không.

**Ví dụ về phân cấp**:

html

Sao chépChỉnh sửa

<div>

<p>Đoạn văn này sẽ có màu xanh lá cây và cỡ chữ 18px.</p>

<section>

<p>Đoạn văn này cũng sẽ có màu xanh lá cây và cỡ chữ 18px.</p>

</section>

</div>

* Cả hai thẻ <p> trong <div> và trong <section> sẽ **được chọn** nhờ bộ chọn div p, vì chúng đều **nằm trong thẻ <div>**.

**🎯 Khi nào sử dụng Bộ chọn kết hợp: A B?**

**✅ 1. Kiểm soát kiểu của phần tử con trong một phần tử cha cụ thể**

Bộ chọn kết hợp rất hữu ích khi bạn muốn **áp dụng kiểu cho các phần tử con** chỉ khi chúng nằm **trong một phần tử cha cụ thể**. Điều này giúp bạn có thể kiểm soát kiểu dáng của các phần tử mà không làm ảnh hưởng đến các phần tử cùng loại ở nơi khác trong trang.

**✅ 2. Tăng tính linh hoạt trong thiết kế**

Bạn có thể dễ dàng áp dụng kiểu cho một phần tử ở một **vị trí nhất định** mà không cần thêm class hoặc ID. Điều này giúp **tiết kiệm thời gian** và giảm bớt sự phức tạp trong việc quản lý CSS.

**Ví dụ**:

css

Sao chépChỉnh sửa

nav a {

color: blue;

}

→ Áp dụng màu chữ xanh cho tất cả các thẻ <a> nằm trong thẻ <nav>, nhưng không ảnh hưởng đến các thẻ <a> ở nơi khác trên trang.

**⚠️ Những lưu ý khi sử dụng Bộ chọn kết hợp: A B**

**1. Đảm bảo cấu trúc HTML hợp lý**

* Bộ chọn kết hợp A B chỉ hoạt động khi **B** thực sự nằm trong **A**. Nếu cấu trúc HTML thay đổi (ví dụ: nếu bạn thay đổi vị trí của phần tử), kiểu CSS có thể không còn áp dụng như mong muốn.

**2. Kiểm soát độ cụ thể (Specificity)**

* Bộ chọn kết hợp có độ cụ thể thấp hơn so với các bộ chọn ID và class. Điều này có nghĩa là nếu bạn muốn áp dụng một kiểu cụ thể, bạn cần chú ý đến độ ưu tiên của các bộ chọn khác.

**3. Không nên dùng quá nhiều bộ chọn kết hợp**

* Dùng quá nhiều bộ chọn kết hợp có thể dẫn đến **CSS trở nên phức tạp** và khó bảo trì. Nếu bạn áp dụng quá nhiều bộ chọn kết hợp, điều này có thể khiến cho việc điều chỉnh layout sau này trở nên khó khăn.

**4. Chú ý đến hiệu suất:**

* Bộ chọn kết hợp hoạt động tốt với các phần tử không có quá nhiều lớp hoặc các phần tử phức tạp. Tuy nhiên, nếu bạn sử dụng quá nhiều bộ chọn kết hợp trên một trang có nhiều phần tử, nó có thể làm **giảm hiệu suất**.

**🎯 Ví dụ nâng cao về Bộ chọn kết hợp: A B**

**✅ HTML:**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ Bộ chọn kết hợp A B</title>

<style>

/\* Chỉ chọn <span> bên trong <section> \*/

section span {

font-weight: bold;

color: green;

}

/\* Chỉ chọn <p> bên trong <article> \*/

article p {

font-size: 16px;

line-height: 1.5;

}

</style>

</head>

<body>

<section>

<p>Đoạn văn này không được chọn.</p>

<span>Đoạn văn này sẽ được in đậm và màu xanh lá cây.</span>

</section>

<article>

<p>Đoạn văn này sẽ có cỡ chữ 16px và khoảng cách dòng là 1.5.</p>

</article>

</body>

</html>

**✅ Giải thích:**

* **section span**: Chỉ **span** bên trong **section** sẽ có **màu xanh lá cây** và **in đậm**.
* **article p**: Chỉ **p** bên trong **article** sẽ có **cỡ chữ 16px** và **khoảng cách dòng 1.5**.

**🎯 Tóm tắt về Bộ chọn kết hợp: A B**

| **Tính năng** | **Mô tả** |
| --- | --- |
| **Cú pháp** | A B (A là phần tử cha, B là phần tử con) |
| **Áp dụng kiểu** | Chọn phần tử **B** nằm trong phần tử **A** và áp dụng kiểu cho chúng. |
| **Ưu điểm** | Tiết kiệm thời gian, dễ quản lý, tăng tính linh hoạt. |
| **Lưu ý** | Cần chú ý đến cấu trúc HTML, không lạm dụng bộ chọn kết hợp để tránh phức tạp và giảm hiệu suất. |

Hy vọng rằng bạn đã hiểu rõ về **Bộ chọn kết hợp A B** trong CSS! Nếu bạn có thêm câu hỏi hoặc cần thêm ví dụ, đừng ngần ngại chia sẻ với mình nhé! 😄

**Bạn đã nói:**

Bạn hãy cho mình ví dụ minh họa bao quát về " Bộ chọn kết hợp: A B".

**ChatGPT đã nói:**

Chắc chắn rồi! Dưới đây là một **ví dụ bao quát** về **Bộ chọn kết hợp: A B** trong CSS, giúp bạn hiểu rõ hơn về cách áp dụng kiểu cho các phần tử con nằm trong các phần tử cha. Mình sẽ cung cấp ví dụ có đầy đủ các loại phần tử để bạn dễ dàng thấy được cách hoạt động và cách sử dụng hiệu quả.

**🎯 Ví dụ bao quát về Bộ chọn kết hợp: A B**

**✅ HTML:**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ Bộ chọn kết hợp: A B</title>

<style>

/\* Bộ chọn kết hợp div p \*/

div p {

color: green;

font-size: 18px;

}

/\* Bộ chọn kết hợp section p \*/

section p {

color: orange;

font-size: 16px;

line-height: 1.6;

}

/\* Bộ chọn kết hợp article h2 \*/

article h2 {

color: blue;

text-align: center;

}

/\* Bộ chọn kết hợp ul li \*/

ul li {

list-style-type: square; /\* Chấm vuông cho danh sách \*/

color: purple;

}

/\* Bộ chọn kết hợp header nav a \*/

header nav a {

text-decoration: none;

color: white;

background-color: blue;

padding: 10px 15px;

margin: 5px;

border-radius: 5px;

}

/\* Bộ chọn kết hợp footer p \*/

footer p {

font-size: 14px;

text-align: center;

color: gray;

}

</style>

</head>

<body>

<header>

<nav>

<a href="#">Trang chủ</a>

<a href="#">Giới thiệu</a>

<a href="#">Liên hệ</a>

</nav>

</header>

<section>

<p>Đoạn văn trong section này có màu cam và cỡ chữ 16px.</p>

</section>

<div>

<p>Đoạn văn trong div này có màu xanh lá cây và cỡ chữ 18px.</p>

</div>

<article>

<h2>Tiêu đề trong article</h2>

<p>Đoạn văn này trong article có màu xám và khoảng cách dòng là 1.6.</p>

</article>

<ul>

<li>Mục 1</li>

<li>Mục 2</li>

<li>Mục 3</li>

</ul>

<footer>

<p>&copy; 2025 Website của chúng tôi</p>

</footer>

</body>

</html>

**✅ Giải thích chi tiết từng bộ chọn kết hợp:**

1. **div p { ... }**:
   * Bộ chọn này áp dụng **cho tất cả các thẻ <p> nằm trong thẻ <div>**.
   * Các thẻ <p> trong <div> sẽ có **màu xanh lá cây** và **cỡ chữ 18px**.
2. **section p { ... }**:
   * Bộ chọn này áp dụng **cho tất cả các thẻ <p> nằm trong thẻ <section>**.
   * Các thẻ <p> trong <section> sẽ có **màu cam**, **cỡ chữ 16px** và **khoảng cách dòng 1.6**.
3. **article h2 { ... }**:
   * Bộ chọn này áp dụng **cho tất cả các thẻ <h2> nằm trong thẻ <article>**.
   * Các thẻ <h2> trong <article> sẽ có **màu xanh dương**, và **canh giữa** (text-align: center).
4. **ul li { ... }**:
   * Bộ chọn này áp dụng **cho tất cả các thẻ <li> trong thẻ <ul>**.
   * Các mục trong danh sách (<li>) sẽ có **dấu chấm vuông** thay vì dấu chấm tròn mặc định và **màu tím**.
5. **header nav a { ... }**:
   * Bộ chọn này áp dụng **cho tất cả các thẻ <a> trong thẻ <nav> nằm trong thẻ <header>**.
   * Các liên kết (<a>) trong <nav> sẽ có **nền màu xanh dương**, **chữ màu trắng**, **không có gạch chân** và **bo góc**.
6. **footer p { ... }**:
   * Bộ chọn này áp dụng **cho tất cả các thẻ <p> trong thẻ <footer>**.
   * Các thẻ <p> trong <footer> sẽ có **cỡ chữ 14px**, **căn giữa** và **màu xám**.

**✅ Kết quả hiển thị trên trình duyệt:**

1. **div p**: Đoạn văn trong thẻ <div> sẽ có **màu xanh lá cây** và **cỡ chữ 18px**.
2. **section p**: Đoạn văn trong thẻ <section> sẽ có **màu cam**, **cỡ chữ 16px** và **khoảng cách dòng 1.6**.
3. **article h2**: Tiêu đề trong thẻ <article> sẽ có **màu xanh dương** và **canh giữa**.
4. **ul li**: Các mục trong danh sách <ul> sẽ có **dấu chấm vuông** và **màu tím**.
5. **header nav a**: Các liên kết trong thẻ <nav> trong <header> sẽ có **nền xanh dương**, **chữ trắng**, **không gạch chân** và **bo góc**.
6. **footer p**: Đoạn văn trong thẻ <footer> sẽ có **cỡ chữ 14px**, **căn giữa** và **màu xám**.

**🎯 Tóm tắt về Bộ chọn kết hợp: A B**

**Công dụng:**

* **Bộ chọn kết hợp: A B** giúp bạn chọn các phần tử con **B** nằm trong phần tử cha **A**, mà không quan tâm đến việc chúng là phần tử con trực tiếp hay không. Điều này rất hữu ích khi bạn muốn áp dụng một kiểu chung cho tất cả các phần tử con trong một phần tử cha nhất định.

**Ưu điểm:**

* **Tính linh hoạt**: Bạn có thể áp dụng kiểu cho nhiều phần tử mà không cần phải sử dụng thêm class hay id.
* **Giảm bớt việc lặp lại CSS**: Giúp bạn dễ dàng áp dụng kiểu cho các phần tử con mà không cần phải viết quá nhiều mã.

**Lưu ý:**

* Hãy chắc chắn rằng cấu trúc HTML của bạn ổn định và phần tử con thực sự nằm trong phần tử cha mà bạn muốn áp dụng kiểu.
* Đôi khi sử dụng quá nhiều bộ chọn kết hợp có thể khiến CSS của bạn trở nên phức tạp, vì vậy hãy cố gắng duy trì sự rõ ràng và dễ bảo trì trong mã của bạn.

Hy vọng ví dụ trên giúp bạn hiểu rõ hơn về cách **Bộ chọn kết hợp: A B** hoạt động trong CSS. Nếu bạn có thêm câu hỏi hay cần thêm ví dụ khác, đừng ngần ngại chia sẻ với mình nhé! 😄

## **2. BỘ CHỌN KẾT HỢP: A > B**

Bạn hãy giải thích chi tiết tất tần tật, sâu rộng toàn bộ và toàn diện kiến thức về "Bộ chọn kết hợp: A > B" và những lưu ý (nếu có) về nó để cho người non-tech hiểu sâu và rõ.

**ChatGPT đã nói:**

Chắc chắn rồi! Hãy cùng khám phá chi tiết về **"Bộ chọn kết hợp: A > B"** trong CSS. Đây là một trong những bộ chọn khá mạnh mẽ và cụ thể trong việc áp dụng kiểu cho các phần tử con trực tiếp của một phần tử cha. Mình sẽ giải thích chi tiết, từ cơ bản đến nâng cao, để bạn có thể dễ dàng hiểu dù không có nền tảng kỹ thuật.

**🎯 Bộ chọn kết hợp: A > B là gì?**

**Bộ chọn kết hợp A > B** trong CSS được gọi là **Bộ chọn con trực tiếp (Direct Child Selector)**. Nó giúp bạn **chọn các phần tử B** chỉ khi chúng là **con trực tiếp của phần tử A**. Điều này có nghĩa là phần tử B sẽ chỉ được chọn nếu nó **là con trực tiếp của A**, không phải là một phần tử cháu hay xa hơn nữa trong cấu trúc HTML.

**📜 Cú pháp:**

css

Sao chépChỉnh sửa

A > B {

/\* Các thuộc tính CSS \*/

}

Trong đó:

* **A** là phần tử cha.
* **B** là phần tử con trực tiếp của A mà bạn muốn áp dụng kiểu.

**🧪 Ví dụ minh họa về Bộ chọn kết hợp: A > B**

**✅ HTML:**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ Bộ chọn kết hợp A > B</title>

<style>

/\* Bộ chọn kết hợp div > p \*/

div > p {

color: red;

font-size: 20px;

}

/\* Bộ chọn kết hợp section > p \*/

section > p {

color: blue;

font-size: 18px;

}

/\* Bộ chọn kết hợp article > h2 \*/

article > h2 {

color: green;

font-size: 24px;

}

</style>

</head>

<body>

<div>

<p>Đoạn văn trong div này sẽ có màu đỏ và cỡ chữ 20px.</p>

<section>

<p>Đoạn văn trong section này sẽ có màu xanh và cỡ chữ 18px.</p>

<article>

<h2>Tiêu đề trong article sẽ có màu xanh lá và cỡ chữ 24px.</h2>

<p>Đoạn văn này sẽ không bị ảnh hưởng vì nó không phải là con trực tiếp của article.</p>

</article>

</section>

</div>

</body>

</html>

**✅ Giải thích chi tiết các bộ chọn:**

1. **div > p { ... }**:
   * Bộ chọn này chỉ áp dụng cho các phần tử <p> **là con trực tiếp của thẻ <div>**.
   * Do đó, **đoạn văn trong thẻ <div>** sẽ có **màu đỏ** và **cỡ chữ 20px**.
2. **section > p { ... }**:
   * Bộ chọn này chỉ áp dụng cho các phần tử <p> **là con trực tiếp của thẻ <section>**.
   * **Đoạn văn trong thẻ <section>** sẽ có **màu xanh** và **cỡ chữ 18px**.
3. **article > h2 { ... }**:
   * Bộ chọn này chỉ áp dụng cho các phần tử <h2> **là con trực tiếp của thẻ <article>**.
   * **Tiêu đề <h2> trong thẻ <article>** sẽ có **màu xanh lá cây** và **cỡ chữ 24px**.

**✅ Kết quả hiển thị trên trình duyệt:**

* **div > p**: Đoạn văn trong thẻ <div> có **màu đỏ** và **cỡ chữ 20px**.
* **section > p**: Đoạn văn trong thẻ <section> có **màu xanh** và **cỡ chữ 18px**.
* **article > h2**: Tiêu đề trong thẻ <article> có **màu xanh lá cây** và **cỡ chữ 24px**.
* **Lưu ý**: Đoạn văn trong thẻ <article> nằm bên dưới <section> sẽ **không bị ảnh hưởng** bởi bộ chọn article > h2, vì nó không phải là phần tử con trực tiếp của thẻ <article>.

**🧠 Cơ chế hoạt động của Bộ chọn kết hợp: A > B**

Bộ chọn A > B chỉ áp dụng **cho các phần tử con trực tiếp** của một phần tử cha. Điều này có nghĩa là nếu phần tử B không phải là con trực tiếp của phần tử A mà là cháu (nằm ở nhiều cấp dưới của A), bộ chọn sẽ không áp dụng.

**Ví dụ minh họa phân cấp:**

html

Sao chépChỉnh sửa

<div>

<p>Đoạn văn này sẽ được chọn bởi div > p.</p> <!-- Con trực tiếp -->

<section>

<p>Đoạn văn này không bị chọn bởi section > p vì nó không phải con trực tiếp của section.</p> <!-- Cháu -->

</section>

</div>

* **div > p** chọn đoạn văn đầu tiên vì nó là con trực tiếp của <div>.
* **section > p** không chọn đoạn văn thứ hai vì thẻ <p> này là cháu của <div> và không phải con trực tiếp của <section>.

**🎯 Khi nào sử dụng Bộ chọn kết hợp: A > B?**

**✅ 1. Kiểm soát kiểu cho phần tử con trực tiếp**

Bộ chọn này rất hữu ích khi bạn muốn **áp dụng kiểu cho các phần tử con trực tiếp** của một phần tử cha mà không ảnh hưởng đến các phần tử cháu hoặc các phần tử nằm sâu hơn trong cấu trúc HTML.

**✅ 2. Giới hạn phạm vi kiểu**

Bạn chỉ muốn kiểu của một phần tử ảnh hưởng đến **phần tử con trực tiếp** của nó, không muốn kiểu đó áp dụng cho các phần tử con ở các cấp độ sâu hơn trong cấu trúc HTML.

**Ví dụ**:

css

Sao chépChỉnh sửa

div > p {

font-weight: bold;

}

→ **Chỉ đoạn văn <p> trực tiếp trong <div>** sẽ được in đậm, các đoạn văn <p> trong các thẻ con như <section> hay <article> sẽ **không bị ảnh hưởng**.

**⚠️ Những lưu ý khi sử dụng Bộ chọn kết hợp: A > B**

**1. Chỉ áp dụng cho phần tử con trực tiếp**

* Bộ chọn này chỉ có tác dụng với **các phần tử con trực tiếp** của phần tử cha, không ảnh hưởng đến các phần tử cháu nằm sâu hơn trong cây DOM.

**Ví dụ không chọn được**:

html

Sao chépChỉnh sửa

<div>

<section>

<p>Đoạn văn này không được chọn vì nó là cháu của div.</p>

</section>

</div>

* **div > p** không chọn đoạn văn này vì nó không phải là con trực tiếp của <div>, mà là cháu của <div>.

**2. Chú ý đến cấu trúc HTML**

* Khi sử dụng bộ chọn A > B, bạn cần đảm bảo rằng cấu trúc HTML của bạn đúng như bạn mong muốn, với các phần tử được xếp theo đúng cấp bậc con-cha.

**3. Độ cụ thể của bộ chọn**

* Bộ chọn A > B có độ cụ thể tương đương với bộ chọn thông thường, nhưng bạn cần lưu ý nếu có các bộ chọn khác có độ cụ thể cao hơn như id, thì chúng có thể ghi đè kiểu của bạn.

**4. Không nên lạm dụng**

* Sử dụng quá nhiều bộ chọn con trực tiếp có thể làm cho CSS của bạn **trở nên khó bảo trì**, đặc biệt khi cấu trúc HTML thay đổi.

**🎯 Tóm tắt về Bộ chọn kết hợp: A > B**

| **Tính năng** | **Mô tả** |
| --- | --- |
| **Cú pháp** | A > B (A là phần tử cha, B là phần tử con trực tiếp) |
| **Áp dụng kiểu** | Chỉ chọn các phần tử B là con trực tiếp của A |
| **Ưu điểm** | Giới hạn phạm vi kiểu cho các phần tử con trực tiếp, giúp dễ dàng kiểm soát layout |
| **Lưu ý** | Chỉ áp dụng cho phần tử con trực tiếp, tránh sử dụng quá nhiều để không làm phức tạp CSS |

Hy vọng rằng bạn đã hiểu rõ hơn về **Bộ chọn kết hợp: A > B** trong CSS! Nếu bạn có thêm câu hỏi hoặc muốn thảo luận chi tiết hơn về ví dụ, đừng ngần ngại chia sẻ với mình nhé! 😄

**Bạn đã nói:**

Bạn hãy cho mình ví dụ minh họa bao quát về "Bộ chọn kết hợp: A > B"

**ChatGPT đã nói:**

Chắc chắn rồi! Dưới đây là một ví dụ **bao quát** về **Bộ chọn kết hợp: A > B** trong CSS, giúp bạn hiểu rõ hơn về cách sử dụng và ứng dụng nó trong HTML.

**✅ HTML:**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ Bộ chọn kết hợp A > B</title>

<style>

/\* Bộ chọn kết hợp div > p \*/

div > p {

color: red;

font-size: 20px;

}

/\* Bộ chọn kết hợp section > p \*/

section > p {

color: blue;

font-size: 18px;

}

/\* Bộ chọn kết hợp article > h2 \*/

article > h2 {

color: green;

font-size: 24px;

}

/\* Bộ chọn kết hợp div > ul > li \*/

div > ul > li {

color: purple;

font-size: 16px;

}

/\* Bộ chọn kết hợp section > div > p \*/

section > div > p {

color: orange;

font-size: 22px;

}

</style>

</head>

<body>

<div>

<p>Đoạn văn trong div này sẽ có màu đỏ và cỡ chữ 20px.</p>

<ul>

<li>Danh sách trong div này có màu tím và cỡ chữ 16px.</li>

<li>Danh sách trong div này có màu tím và cỡ chữ 16px.</li>

</ul>

</div>

<section>

<p>Đoạn văn trong section này sẽ có màu xanh và cỡ chữ 18px.</p>

<div>

<p>Đoạn văn này trong div nằm trong section sẽ có màu cam và cỡ chữ 22px.</p>

</div>

</section>

<article>

<h2>Tiêu đề trong article sẽ có màu xanh lá và cỡ chữ 24px.</h2>

<p>Đoạn văn này không bị ảnh hưởng vì nó không phải con trực tiếp của article.</p>

</article>

</body>

</html>

**✅ Giải thích chi tiết về các bộ chọn:**

1. **div > p { ... }**:
   * Bộ chọn này chọn **tất cả các thẻ <p> là con trực tiếp của thẻ <div>**.
   * **Đoạn văn trong <div>** sẽ có **màu đỏ** và **cỡ chữ 20px**.
2. **section > p { ... }**:
   * Bộ chọn này chọn **tất cả các thẻ <p> là con trực tiếp của thẻ <section>**.
   * **Đoạn văn trong <section>** sẽ có **màu xanh** và **cỡ chữ 18px**.
3. **article > h2 { ... }**:
   * Bộ chọn này chọn **tất cả các thẻ <h2> là con trực tiếp của thẻ <article>**.
   * **Tiêu đề trong <article>** sẽ có **màu xanh lá cây** và **cỡ chữ 24px**.
4. **div > ul > li { ... }**:
   * Bộ chọn này áp dụng **cho tất cả các thẻ <li> nằm trong <ul> nằm trong thẻ <div>**.
   * **Các mục trong danh sách <ul> trong <div>** sẽ có **màu tím** và **cỡ chữ 16px**.
5. **section > div > p { ... }**:
   * Bộ chọn này áp dụng **cho tất cả các thẻ <p> nằm trong <div> nằm trong thẻ <section>**.
   * **Đoạn văn trong <div> nằm trong <section>** sẽ có **màu cam** và **cỡ chữ 22px**.

**✅ Kết quả hiển thị trên trình duyệt:**

* **div > p**: Đoạn văn trong thẻ <div> có **màu đỏ** và **cỡ chữ 20px**.
* **section > p**: Đoạn văn trong thẻ <section> có **màu xanh** và **cỡ chữ 18px**.
* **article > h2**: Tiêu đề trong thẻ <article> có **màu xanh lá cây** và **cỡ chữ 24px**.
* **div > ul > li**: Các mục trong danh sách (<li>) trong <ul> trong <div> sẽ có **màu tím** và **cỡ chữ 16px**.
* **section > div > p**: Đoạn văn trong thẻ <div> nằm trong thẻ <section> có **màu cam** và **cỡ chữ 22px**.

**Lưu ý:**

* **article > p** không bị ảnh hưởng vì thẻ <p> không phải là con trực tiếp của <article>, mà là phần tử con của các phần tử khác như <div> hay <section>.
* Chỉ có những phần tử con **trực tiếp** (ngay dưới cha) mới bị ảnh hưởng bởi bộ chọn A > B.

**✅ Tóm tắt về Bộ chọn kết hợp: A > B**

* **Công dụng**: Chọn các phần tử **con trực tiếp** của một phần tử cha.
* **Ví dụ sử dụng**: Khi bạn muốn áp dụng kiểu cho các phần tử con ngay trong phần tử cha mà không ảnh hưởng đến các phần tử con ở các cấp độ sâu hơn trong cấu trúc HTML.
* **Lưu ý**: Nếu bạn muốn chọn phần tử con ở nhiều cấp, cần chú ý cấu trúc cây DOM và sử dụng bộ chọn kết hợp hợp lý.

Hy vọng ví dụ này giúp bạn hiểu rõ hơn về cách **Bộ chọn kết hợp: A > B** hoạt động trong CSS! Nếu bạn có thêm câu hỏi hay muốn thảo luận chi tiết hơn, đừng ngần ngại chia sẻ với mình nhé! 😄

## **3. BỘ CHỌN KẾT HỢP: A + B**

Bạn hãy giải thích chi tiết tất tần tật, sâu rộng toàn bộ và toàn diện kiến thức về "Bộ chọn kết hợp: A + B" và những lưu ý (nếu có) về nó để cho người non-tech hiểu sâu và rõ.

**ChatGPT đã nói:**

Chắc chắn rồi! Hôm nay mình sẽ giải thích chi tiết về **"Bộ chọn kết hợp: A + B"** trong CSS. Đây là một bộ chọn khá đặc biệt và hữu ích, nhưng nếu không hiểu rõ, nó có thể gây chút nhầm lẫn. Mình sẽ giải thích từ cơ bản đến nâng cao, giúp bạn hiểu thật sâu về cách sử dụng bộ chọn này một cách hiệu quả.

**🎯 Bộ chọn kết hợp: A + B là gì?**

**Bộ chọn kết hợp A + B** là **Bộ chọn anh em liền kề** (Adjacent Sibling Selector) trong CSS. Bộ chọn này giúp bạn chọn **phần tử B** chỉ khi nó **là anh em liền kề** với phần tử **A** trong cấu trúc HTML. Điều này có nghĩa là phần tử **B** phải xuất hiện **ngay sau phần tử A**, và chúng phải cùng cấp (cùng cha).

**📜 Cú pháp:**

css

Sao chépChỉnh sửa

A + B {

/\* Các thuộc tính CSS \*/

}

Trong đó:

* **A** là phần tử đứng trước.
* **B** là phần tử đứng ngay sau **A**, và là anh em liền kề.

**🧪 Ví dụ minh họa về Bộ chọn kết hợp: A + B**

**✅ HTML:**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ Bộ chọn kết hợp A + B</title>

<style>

/\* Bộ chọn kết hợp h1 + p \*/

h1 + p {

color: red;

font-size: 20px;

}

/\* Bộ chọn kết hợp h2 + p \*/

h2 + p {

color: blue;

font-size: 18px;

}

/\* Bộ chọn kết hợp p + ul \*/

p + ul {

background-color: lightyellow;

padding: 10px;

}

</style>

</head>

<body>

<h1>Tiêu đề 1</h1>

<p>Đoạn văn ngay sau h1 sẽ có màu đỏ và cỡ chữ 20px.</p>

<h2>Tiêu đề 2</h2>

<p>Đoạn văn ngay sau h2 sẽ có màu xanh và cỡ chữ 18px.</p>

<p>Đoạn văn này không bị ảnh hưởng vì không phải là con liền kề với h2.</p>

<p>Đoạn văn này sẽ được ảnh hưởng bởi p + ul.</p>

<ul>

<li>Mục 1</li>

<li>Mục 2</li>

</ul>

</body>

</html>

**✅ Giải thích chi tiết về bộ chọn A + B:**

1. **h1 + p { ... }**:
   * Bộ chọn này **chỉ áp dụng cho thẻ <p>** ngay **sau thẻ <h1>**.
   * Trong ví dụ trên, đoạn văn <p> ngay sau tiêu đề <h1> sẽ có **màu đỏ** và **cỡ chữ 20px**.
2. **h2 + p { ... }**:
   * Bộ chọn này **chỉ áp dụng cho thẻ <p>** ngay **sau thẻ <h2>**.
   * Đoạn văn <p> ngay sau tiêu đề <h2> sẽ có **màu xanh** và **cỡ chữ 18px**.
3. **p + ul { ... }**:
   * Bộ chọn này chọn **thẻ <ul>** ngay **sau thẻ <p>**.
   * Do đó, **danh sách <ul>** ngay sau đoạn văn <p> sẽ có **nền màu vàng nhạt** và **padding 10px**.

**✅ Kết quả hiển thị trên trình duyệt:**

1. **h1 + p**: Đoạn văn <p> ngay sau thẻ <h1> có **màu đỏ** và **cỡ chữ 20px**.
2. **h2 + p**: Đoạn văn <p> ngay sau thẻ <h2> có **màu xanh** và **cỡ chữ 18px**.
3. **p + ul**: Danh sách <ul> ngay sau đoạn văn <p> có **nền màu vàng nhạt** và **padding 10px**.
4. **Lưu ý**: Đoạn văn và danh sách không phải là "anh em liền kề" với các thẻ tiêu đề, vì chúng không xuất hiện ngay sau các thẻ đó.

**🧠 Cơ chế hoạt động của Bộ chọn A + B**

Bộ chọn **A + B** sẽ chỉ áp dụng cho phần tử **B** nếu:

* **B** là phần tử **anh em liền kề** với **A**.
* **B** phải **xuất hiện ngay sau A** trong cùng một cấp cha (cùng cấp trong cây DOM).

**Ví dụ phân cấp:**

html

Sao chépChỉnh sửa

<div>

<h1>Tiêu đề 1</h1>

<p>Đoạn văn ngay sau h1 sẽ được chọn.</p> <!-- Được chọn -->

<p>Đoạn văn không được chọn vì không phải là anh em liền kề với h1.</p> <!-- Không được chọn -->

<h2>Tiêu đề 2</h2>

<p>Đoạn văn ngay sau h2 sẽ được chọn.</p> <!-- Được chọn -->

</div>

* **h1 + p**: Đoạn văn <p> ngay sau <h1> sẽ được chọn và áp dụng kiểu.
* **h2 + p**: Đoạn văn <p> ngay sau <h2> sẽ được chọn và áp dụng kiểu.
* Đoạn văn thứ ba **không được chọn** vì nó không phải là **anh em liền kề** với <h2>.

**🎯 Khi nào sử dụng Bộ chọn kết hợp: A + B?**

**✅ 1. Chọn phần tử ngay sau phần tử khác**

Bộ chọn này rất hữu ích khi bạn chỉ muốn áp dụng kiểu cho phần tử **ngay sau một phần tử khác**, và chúng có cùng cấp. Ví dụ, bạn có thể muốn phần tử <p> ngay sau <h1> có màu sắc khác so với các đoạn văn khác.

**✅ 2. Giới hạn phạm vi áp dụng kiểu**

Bạn không muốn áp dụng kiểu cho **tất cả các phần tử** cùng loại, mà chỉ cho các phần tử **liền kề** với một phần tử khác. Điều này giúp bạn tiết kiệm mã CSS và làm cho mã của bạn dễ quản lý hơn.

**Ví dụ sử dụng:**

css

Sao chépChỉnh sửa

h1 + p {

font-style: italic;

}

→ **Chỉ đoạn văn ngay sau tiêu đề <h1>** sẽ được in nghiêng, không ảnh hưởng đến các phần tử <p> khác.

**⚠️ Những lưu ý khi sử dụng Bộ chọn kết hợp: A + B**

**1. Chỉ áp dụng cho anh em liền kề**

Bộ chọn này chỉ hoạt động nếu phần tử B **ngay lập tức** đứng sau phần tử A, không có bất kỳ phần tử nào khác nằm giữa chúng. Nếu có bất kỳ phần tử nào giữa A và B, bộ chọn sẽ không áp dụng.

**Ví dụ không chọn được**:

html

Sao chépChỉnh sửa

<h1>Tiêu đề 1</h1>

<p>Đoạn văn này không được chọn.</p>

<div>Phần tử này nằm giữa h1 và p.</div>

<p>Đoạn văn này cũng không được chọn vì nó không phải là anh em liền kề với h1.</p>

* **h1 + p** không chọn đoạn văn thứ hai vì có thẻ <div> ở giữa.

**2. Không chọn được phần tử sau phần tử khác ngoài A**

Chỉ có các phần tử **B** ngay sau **A** mới được chọn. Nếu **B** không ngay sau **A** (có thể là phần tử khác ở giữa), bộ chọn sẽ không áp dụng.

**3. Bộ chọn có độ cụ thể thấp**

Mặc dù bộ chọn này rất hữu ích, nhưng độ cụ thể của nó thấp hơn so với các bộ chọn như **ID** hoặc **class**. Điều này có thể dẫn đến việc bị ghi đè bởi các bộ chọn có độ cụ thể cao hơn.

**🎯 Tóm tắt về Bộ chọn kết hợp: A + B**

| **Tính năng** | **Mô tả** |
| --- | --- |
| **Cú pháp** | A + B (A là phần tử đứng trước, B là phần tử đứng ngay sau A) |
| **Áp dụng kiểu** | Chỉ áp dụng kiểu cho phần tử **B** khi **B là anh em liền kề với A** trong cây DOM |
| **Ưu điểm** | Giúp áp dụng kiểu cho các phần tử ngay sau phần tử khác mà không ảnh hưởng đến các phần tử khác. |
| **Lưu ý** | Phải có phần tử B ngay sau A, nếu không có thì bộ chọn sẽ không áp dụng. |

Hy vọng bạn đã hiểu rõ hơn về **Bộ chọn kết hợp: A + B** và có thể áp dụng vào công việc của mình một cách hiệu quả! Nếu có câu hỏi nào thêm, đừng ngần ngại chia sẻ với mình nhé! 😊

Bạn hãy cho mình ví dụ minh họa bao quát về "Bộ chọn kết hợp: A + B".

**ChatGPT đã nói:**

Chắc chắn rồi! Dưới đây là một ví dụ minh họa bao quát về **Bộ chọn kết hợp: A + B** trong CSS. Cùng xem cách hoạt động của bộ chọn này trong HTML nhé!

**✅ HTML:**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ Bộ chọn kết hợp A + B</title>

<style>

/\* Bộ chọn kết hợp h1 + p \*/

h1 + p {

color: red;

font-size: 20px;

}

/\* Bộ chọn kết hợp h2 + p \*/

h2 + p {

color: blue;

font-size: 18px;

}

/\* Bộ chọn kết hợp p + ul \*/

p + ul {

background-color: lightyellow;

padding: 10px;

}

/\* Bộ chọn kết hợp h1 + div \*/

h1 + div {

background-color: lightblue;

padding: 20px;

}

</style>

</head>

<body>

<h1>Tiêu đề 1</h1>

<p>Đoạn văn này ngay sau h1 sẽ có màu đỏ và cỡ chữ 20px.</p>

<h2>Tiêu đề 2</h2>

<p>Đoạn văn này ngay sau h2 sẽ có màu xanh và cỡ chữ 18px.</p>

<p>Đoạn văn này không bị ảnh hưởng vì không phải là anh em liền kề với h2.</p>

<p>Đoạn văn này có ảnh hưởng đến phần tử ul ngay sau nó.</p>

<ul>

<li>Mục 1</li>

<li>Mục 2</li>

</ul>

<h1>Tiêu đề 1 tiếp theo</h1>

<div>Div này ngay sau h1 sẽ có nền màu xanh lam và padding 20px.</div>

<h1>Tiêu đề 1 khác</h1>

<p>Đoạn văn này không được chọn vì không phải là anh em liền kề với h1.</p>

</body>

</html>

**✅ Giải thích chi tiết về bộ chọn A + B trong ví dụ:**

1. **h1 + p { ... }**:
   * Bộ chọn này sẽ **áp dụng kiểu cho tất cả các thẻ <p> ngay sau thẻ <h1>**.
   * Trong ví dụ này, đoạn văn <p> ngay sau tiêu đề <h1> đầu tiên sẽ có **màu đỏ** và **cỡ chữ 20px**.
2. **h2 + p { ... }**:
   * Bộ chọn này sẽ **áp dụng kiểu cho tất cả các thẻ <p> ngay sau thẻ <h2>**.
   * Đoạn văn <p> ngay sau tiêu đề <h2> sẽ có **màu xanh** và **cỡ chữ 18px**.
3. **p + ul { ... }**:
   * Bộ chọn này sẽ **áp dụng kiểu cho thẻ <ul> ngay sau thẻ <p>**.
   * Danh sách <ul> ngay sau đoạn văn <p> thứ hai sẽ có **nền màu vàng nhạt** và **padding 10px**.
4. **h1 + div { ... }**:
   * Bộ chọn này sẽ **áp dụng kiểu cho thẻ <div> ngay sau thẻ <h1>**.
   * Thẻ <div> ngay sau tiêu đề <h1> thứ hai sẽ có **nền màu xanh lam** và **padding 20px**.

**✅ Kết quả hiển thị trên trình duyệt:**

1. **h1 + p**: Đoạn văn ngay sau <h1> đầu tiên sẽ có **màu đỏ** và **cỡ chữ 20px**.
2. **h2 + p**: Đoạn văn ngay sau <h2> sẽ có **màu xanh** và **cỡ chữ 18px**.
3. **p + ul**: Danh sách <ul> ngay sau đoạn văn <p> thứ ba sẽ có **nền màu vàng nhạt** và **padding 10px**.
4. **h1 + div**: Thẻ <div> ngay sau <h1> thứ hai sẽ có **nền màu xanh lam** và **padding 20px**.

**✅ Phân tích về sự kết hợp các phần tử:**

* **h1 + p** chọn đoạn văn ngay sau **tiêu đề <h1>**.
* **h2 + p** chọn đoạn văn ngay sau **tiêu đề <h2>**.
* **p + ul** chọn danh sách <ul> ngay sau **đoạn văn <p>**.
* **h1 + div** chọn thẻ <div> ngay sau **tiêu đề <h1>**.

**Lưu ý quan trọng:**

* **Bộ chọn A + B** chỉ áp dụng khi **phần tử B là anh em liền kề** và **ngay sau phần tử A**. Nếu có một phần tử khác nằm giữa A và B, thì bộ chọn này sẽ không áp dụng.

Hy vọng ví dụ này giúp bạn hiểu rõ hơn về cách sử dụng **Bộ chọn kết hợp: A + B** trong CSS! Nếu có thêm bất kỳ câu hỏi nào, đừng ngần ngại chia sẻ với mình nhé! 😊

## **4. BỘ CHỌN KẾT HỢP: A ~ B**

Chắc chắn rồi! Hôm nay, mình sẽ giải thích chi tiết về **"Bộ chọn kết hợp: A ~ B"** trong CSS. Đây là một bộ chọn khá mạnh mẽ, nhưng nếu không hiểu rõ, nó có thể khiến bạn bối rối. Mình sẽ giải thích từ cơ bản đến nâng cao để bạn có thể dễ dàng hiểu và áp dụng.

**🎯 Bộ chọn kết hợp: A ~ B là gì?**

**Bộ chọn kết hợp A ~ B** là **Bộ chọn anh em tổng quát** (General Sibling Selector) trong CSS. Nó cho phép bạn chọn **tất cả các phần tử B** là **anh em của A**, nhưng không nhất thiết phải là phần tử ngay sau A. Điều này có nghĩa là **B** có thể xuất hiện **sau A**, nhưng **không nhất thiết phải ngay liền kề** với A, miễn là chúng **cùng cấp cha** (cùng cấp trong cấu trúc DOM).

**📜 Cú pháp:**

css

Sao chépChỉnh sửa

A ~ B {

/\* Các thuộc tính CSS \*/

}

Trong đó:

* **A** là phần tử đứng trước.
* **B** là phần tử bạn muốn chọn, và **B** phải là phần tử **anh em cùng cấp** với **A** (cùng nằm trong cùng một phần tử cha).

**🧪 Cơ chế hoạt động của Bộ chọn A ~ B**

* **Bộ chọn A ~ B** sẽ áp dụng cho **tất cả các phần tử B** sau **phần tử A**, miễn là **B** là phần tử anh em cùng cấp với **A** trong cấu trúc HTML.
* Bộ chọn này không yêu cầu **B** phải ngay sau **A**, mà chỉ cần **B** xuất hiện sau **A**, và cả hai phải thuộc cùng một cấp trong cây DOM.

**✅ Ví dụ minh họa về Bộ chọn A ~ B**

**HTML:**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ Bộ chọn kết hợp A ~ B</title>

<style>

/\* Bộ chọn kết hợp h1 ~ p \*/

h1 ~ p {

color: red;

font-size: 18px;

}

/\* Bộ chọn kết hợp h2 ~ p \*/

h2 ~ p {

color: blue;

font-size: 20px;

}

/\* Bộ chọn kết hợp div ~ p \*/

div ~ p {

color: green;

font-size: 16px;

}

</style>

</head>

<body>

<h1>Tiêu đề 1</h1>

<p>Đoạn văn này ngay sau h1 sẽ có màu đỏ và cỡ chữ 18px.</p>

<div>Phần tử div này sẽ không ảnh hưởng gì đến đoạn văn p</div>

<p>Đoạn văn này có màu đỏ và cỡ chữ 18px vì là anh em của h1.</p>

<h2>Tiêu đề 2</h2>

<p>Đoạn văn này ngay sau h2 sẽ có màu xanh và cỡ chữ 20px.</p>

<p>Đoạn văn này là anh em của h2 và sẽ có màu xanh và cỡ chữ 20px.</p>

<div>Phần tử div này cũng ảnh hưởng đến đoạn văn p sau đó</div>

<p>Đoạn văn này có màu xanh vì nó là anh em của h2.</p>

</body>

</html>

**✅ Giải thích chi tiết về bộ chọn A ~ B trong ví dụ:**

1. **h1 ~ p { ... }**:
   * Bộ chọn này sẽ **áp dụng kiểu cho tất cả các thẻ <p> là anh em của thẻ <h1>** trong cùng một cấp (cùng cha).
   * Trong ví dụ trên, **đoạn văn <p>** ngay sau **h1** và **đoạn văn <p> thứ hai** sẽ được chọn và áp dụng **màu đỏ** và **cỡ chữ 18px**.
2. **h2 ~ p { ... }**:
   * Bộ chọn này sẽ **áp dụng kiểu cho tất cả các thẻ <p> là anh em của thẻ <h2>**.
   * Các đoạn văn <p> sau thẻ <h2> sẽ được chọn và có **màu xanh** và **cỡ chữ 20px**.
3. **div ~ p { ... }**:
   * Bộ chọn này sẽ **áp dụng kiểu cho tất cả các thẻ <p> là anh em của thẻ <div>**.
   * Các đoạn văn <p> sau thẻ <div> trong cùng một cấp cha sẽ có **màu xanh** và **cỡ chữ 20px**.

**✅ Kết quả hiển thị trên trình duyệt:**

1. **h1 ~ p**:
   * Đoạn văn <p> đầu tiên ngay sau <h1> sẽ có **màu đỏ** và **cỡ chữ 18px**.
   * Đoạn văn <p> thứ hai cũng sẽ có **màu đỏ** và **cỡ chữ 18px** vì nó là anh em cùng cấp với <h1>.
2. **h2 ~ p**:
   * Đoạn văn <p> ngay sau <h2> sẽ có **màu xanh** và **cỡ chữ 20px**.
   * Đoạn văn <p> thứ hai sau thẻ <h2> cũng sẽ có **màu xanh** và **cỡ chữ 20px** vì nó là anh em của <h2>.
3. **div ~ p**:
   * Đoạn văn <p> sau thẻ <div> cũng sẽ có **màu xanh** và **cỡ chữ 20px**, vì nó là anh em cùng cấp với thẻ <div>.

**✅ Phân tích về sự kết hợp các phần tử:**

* **h1 ~ p** chọn tất cả các phần tử <p> sau thẻ <h1>, không quan trọng là chúng có xuất hiện ngay sau <h1> hay không, miễn là chúng là anh em cùng cấp.
* **h2 ~ p** chọn tất cả các phần tử <p> sau thẻ <h2>.
* **div ~ p** chọn tất cả các phần tử <p> sau thẻ <div>.

**🎯 Khi nào sử dụng Bộ chọn kết hợp: A ~ B?**

**✅ 1. Chọn tất cả các phần tử anh em cùng cấp với A**

Bộ chọn này rất hữu ích khi bạn muốn áp dụng kiểu cho tất cả các phần tử **B** là **anh em cùng cấp** với phần tử **A**, mà không cần quan tâm đến thứ tự xuất hiện của chúng. Ví dụ, bạn muốn tất cả các đoạn văn **p** sau một tiêu đề **h1** hoặc một phần tử **div** có cùng kiểu dáng.

**✅ 2. Chọn nhiều phần tử sau A, không nhất thiết ngay sau**

Điều này giúp bạn có thể chọn các phần tử **B** bất kể chúng xuất hiện ngay sau **A** hay không, miễn là chúng là anh em cùng cấp. Đây là điểm khác biệt so với bộ chọn **A + B**, nơi **B** phải ngay sau **A**.

**Ví dụ sử dụng:**

css

Sao chépChỉnh sửa

h1 ~ p {

font-style: italic;

}

→ **Tất cả các phần tử <p>** sau bất kỳ thẻ <h1> nào, miễn là chúng là anh em cùng cấp, sẽ được in nghiêng.

**⚠️ Những lưu ý khi sử dụng Bộ chọn kết hợp: A ~ B**

**1. Chỉ áp dụng cho anh em cùng cấp**

* Bộ chọn này chỉ hoạt động khi **A** và **B** là anh em cùng cấp trong cây DOM. Nếu chúng không có chung một phần tử cha, bộ chọn này sẽ không hoạt động.

**2. Bộ chọn có độ cụ thể thấp**

* Độ cụ thể của bộ chọn này thấp hơn so với các bộ chọn như **ID** hoặc **class**. Điều này có thể khiến bộ chọn bị ghi đè bởi các bộ chọn có độ cụ thể cao hơn.

**3. Không cần phải ngay sau**

* Điều đặc biệt về **A ~ B** là **B** không cần phải ngay lập tức đứng sau **A** như bộ chọn **A + B**. Chỉ cần **B** xuất hiện sau **A** trong cây DOM và chúng có cùng cấp là được.

**🎯 Tóm tắt về Bộ chọn kết hợp: A ~ B**

| **Tính năng** | **Mô tả** |
| --- | --- |
| **Cú pháp** | A ~ B (Chọn tất cả phần tử B là anh em của A, xuất hiện sau A) |
| **Áp dụng kiểu** | Chọn tất cả các phần tử **B** là anh em của **A**, không cần **B** phải ngay sau **A**, miễn là cùng cấp cha. |
| **Ưu điểm** | Chọn được nhiều phần tử cùng loại trong một lần, không yêu cầu chúng phải ngay sau A. |
| **Lưu ý** | Bộ chọn này chỉ áp dụng với phần tử **B** là anh em của **A** trong cùng một cấp. |

Hy vọng rằng bạn đã hiểu rõ hơn về **Bộ chọn kết hợp: A ~ B** và cách sử dụng nó trong CSS. Nếu còn bất kỳ câu hỏi nào, đừng ngần ngại chia sẻ với mình nhé! 😊

Bạn hãy cho mình ví dụ minh họa bao quát về "Bộ chọn kết hợp: A ~ B".

**ChatGPT đã nói:**

Chắc chắn rồi! Dưới đây là một ví dụ minh họa bao quát về **Bộ chọn kết hợp: A ~ B** trong CSS. Bộ chọn này sẽ giúp bạn chọn tất cả các phần tử **B** là anh em của **A** và xuất hiện sau **A**, nhưng không nhất thiết phải ngay sau.

**✅ HTML:**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ Bộ chọn kết hợp A ~ B</title>

<style>

/\* Bộ chọn kết hợp h1 ~ p \*/

h1 ~ p {

color: red;

font-size: 18px;

}

/\* Bộ chọn kết hợp h2 ~ p \*/

h2 ~ p {

color: blue;

font-size: 20px;

}

/\* Bộ chọn kết hợp div ~ p \*/

div ~ p {

color: green;

font-size: 16px;

}

</style>

</head>

<body>

<h1>Tiêu đề 1</h1>

<p>Đoạn văn này ngay sau h1 sẽ có màu đỏ và cỡ chữ 18px.</p>

<div>Phần tử div này sẽ không ảnh hưởng gì đến đoạn văn p</div>

<p>Đoạn văn này có màu đỏ và cỡ chữ 18px vì là anh em của h1.</p>

<h2>Tiêu đề 2</h2>

<p>Đoạn văn này ngay sau h2 sẽ có màu xanh và cỡ chữ 20px.</p>

<p>Đoạn văn này là anh em của h2 và sẽ có màu xanh và cỡ chữ 20px.</p>

<div>Phần tử div này cũng ảnh hưởng đến đoạn văn p sau đó</div>

<p>Đoạn văn này có màu xanh vì nó là anh em của h2.</p>

</body>

</html>

**✅ Giải thích chi tiết về bộ chọn A ~ B trong ví dụ:**

1. **h1 ~ p { ... }**:
   * Bộ chọn này sẽ **áp dụng kiểu cho tất cả các thẻ <p> là anh em của thẻ <h1>** trong cùng một cấp (cùng cha).
   * Trong ví dụ trên, **đoạn văn <p>** ngay sau **h1** và **đoạn văn <p> thứ hai** sẽ được chọn và áp dụng **màu đỏ** và **cỡ chữ 18px**.
2. **h2 ~ p { ... }**:
   * Bộ chọn này sẽ **áp dụng kiểu cho tất cả các thẻ <p> là anh em của thẻ <h2>**.
   * Các đoạn văn <p> sau thẻ <h2> sẽ được chọn và có **màu xanh** và **cỡ chữ 20px**.
3. **div ~ p { ... }**:
   * Bộ chọn này sẽ **áp dụng kiểu cho tất cả các thẻ <p> là anh em của thẻ <div>**.
   * Các đoạn văn <p> sau thẻ <div> trong cùng một cấp cha sẽ có **màu xanh** và **cỡ chữ 20px**.

**✅ Kết quả hiển thị trên trình duyệt:**

1. **h1 ~ p**:
   * Đoạn văn <p> đầu tiên ngay sau <h1> sẽ có **màu đỏ** và **cỡ chữ 18px**.
   * Đoạn văn <p> thứ hai cũng sẽ có **màu đỏ** và **cỡ chữ 18px** vì nó là anh em cùng cấp với <h1>.
2. **h2 ~ p**:
   * Đoạn văn <p> ngay sau <h2> sẽ có **màu xanh** và **cỡ chữ 20px**.
   * Đoạn văn <p> thứ hai sau thẻ <h2> cũng sẽ có **màu xanh** và **cỡ chữ 20px** vì nó là anh em của <h2>.
3. **div ~ p**:
   * Đoạn văn <p> sau thẻ <div> cũng sẽ có **màu xanh** và **cỡ chữ 20px**, vì nó là anh em cùng cấp với thẻ <div>.

**✅ Phân tích về sự kết hợp các phần tử:**

* **h1 ~ p** chọn tất cả các phần tử <p> sau thẻ <h1>, không quan trọng là chúng có xuất hiện ngay sau <h1> hay không, miễn là chúng là anh em cùng cấp.
* **h2 ~ p** chọn tất cả các phần tử <p> sau thẻ <h2>.
* **div ~ p** chọn tất cả các phần tử <p> sau thẻ <div>.

**✅ Tóm tắt về Bộ chọn kết hợp A ~ B:**

| **Tính năng** | **Mô tả** |
| --- | --- |
| **Cú pháp** | A ~ B (Chọn tất cả phần tử B là anh em của A, xuất hiện sau A) |
| **Áp dụng kiểu** | Chọn tất cả các phần tử **B** là anh em của **A**, không cần **B** phải ngay sau **A**, miễn là cùng cấp cha. |
| **Ưu điểm** | Chọn được nhiều phần tử cùng loại trong một lần, không yêu cầu chúng phải ngay sau A. |
| **Lưu ý** | Bộ chọn này chỉ áp dụng với phần tử **B** là anh em của **A** trong cùng một cấp. |

Hy vọng ví dụ này giúp bạn hiểu rõ hơn về cách sử dụng **Bộ chọn kết hợp: A ~ B** trong CSS! Nếu có thêm câu hỏi nào, đừng ngần ngại chia sẻ với mình nhé! 😊

# **C. BỘ CHỌN NHÓM (GROUP SELECTOR)**

ạn hãy giải thích chi tiết tất tần tật, sâu rộng toàn bộ và toàn diện kiến thức về "Bộ chọn nhóm (Group Selector)" và những lưu ý (nếu có) về nó để cho người non-tech hiểu sâu và rõ.

**ChatGPT đã nói:**

Chắc chắn rồi! Hôm nay, mình sẽ giải thích chi tiết về **"Bộ chọn nhóm (Group Selector)"** trong CSS. Đây là một khái niệm khá hữu ích, giúp bạn tiết kiệm thời gian và viết mã CSS ngắn gọn hơn. Hãy cùng khám phá nhé!

**🎯 Bộ chọn nhóm (Group Selector) là gì?**

**Bộ chọn nhóm** trong CSS là một kỹ thuật giúp bạn nhóm lại các phần tử có cùng kiểu dáng để áp dụng **một tập hợp các thuộc tính CSS** cho tất cả chúng. Thay vì phải viết các quy tắc CSS riêng biệt cho từng phần tử, bạn có thể **nhóm chúng lại** bằng cách sử dụng dấu **phẩy (,)** để áp dụng cùng một kiểu cho nhiều phần tử khác nhau.

**📜 Cú pháp của Bộ chọn nhóm:**

css

Sao chépChỉnh sửa

A, B, C {

/\* Các thuộc tính CSS \*/

}

Trong đó:

* **A, B, C** là các bộ chọn (selectors), có thể là **thẻ (element)**, **lớp (class)**, **ID** hoặc bất kỳ bộ chọn nào.
* Các phần tử này được **ngăn cách bằng dấu phẩy (,)** và sẽ **chung một tập hợp các thuộc tính CSS**.

**🎯 Cách hoạt động của Bộ chọn nhóm**

Bộ chọn nhóm giúp bạn nhóm các phần tử lại với nhau và áp dụng **các thuộc tính CSS giống nhau** cho tất cả những phần tử đó. Việc này không chỉ giúp **tiết kiệm mã nguồn**, mà còn giúp **giảm thiểu sự trùng lặp** khi bạn muốn tất cả các phần tử trong nhóm đó có cùng kiểu dáng.

Ví dụ, nếu bạn muốn thẻ <h1>, <h2>, và <p> có cùng màu chữ và kích thước chữ, thay vì viết các quy tắc CSS cho từng thẻ, bạn có thể nhóm chúng lại với nhau và áp dụng các thuộc tính đó.

**✅ Ví dụ minh họa về Bộ chọn nhóm**

**HTML:**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ Bộ chọn nhóm (Group Selector)</title>

<style>

/\* Bộ chọn nhóm cho các thẻ h1, h2 và p \*/

h1, h2, p {

color: #333;

font-family: Arial, sans-serif;

font-size: 16px;

}

/\* Bộ chọn nhóm cho các lớp \*/

.button, .card, .container {

padding: 10px;

border-radius: 5px;

background-color: #f2f2f2;

}

</style>

</head>

<body>

<h1>Tiêu đề lớn</h1>

<h2>Tiêu đề nhỏ</h2>

<p>Đoạn văn này sẽ có màu chữ #333, font là Arial và cỡ chữ 16px.</p>

<div class="button">Nút</div>

<div class="card">Thẻ thông tin</div>

<div class="container">Chứa nội dung</div>

</body>

</html>

**✅ Giải thích chi tiết về Bộ chọn nhóm trong ví dụ:**

1. **h1, h2, p { ... }**:
   * Bộ chọn nhóm này áp dụng các kiểu cho cả ba phần tử <h1>, <h2>, và <p>.
   * Các thuộc tính được áp dụng:
     + **Màu chữ (color)**: Màu #333 cho cả ba thẻ.
     + **Font chữ (font-family)**: Chữ sử dụng font **Arial**, và nếu font này không có thì sẽ dùng **sans-serif**.
     + **Kích thước chữ (font-size)**: Cỡ chữ 16px cho cả ba phần tử.

Như vậy, dù bạn có ba phần tử <h1>, <h2>, và <p>, bạn chỉ cần viết một lần bộ chọn nhóm này thay vì viết riêng cho từng phần tử.

1. **.button, .card, .container { ... }**:
   * Bộ chọn nhóm này áp dụng các kiểu cho ba lớp CSS .button, .card, và .container.
   * Các thuộc tính được áp dụng:
     + **Padding**: Cung cấp khoảng cách 10px từ bên trong các phần tử.
     + **Border-radius**: Làm cho các góc của các phần tử trở nên tròn với bán kính 5px.
     + **Background-color**: Màu nền của các phần tử là **#f2f2f2** (màu xám nhạt).

Nhờ bộ chọn nhóm, bạn có thể tái sử dụng các thuộc tính CSS chung cho các phần tử, giúp mã nguồn trở nên **ngắn gọn và dễ bảo trì** hơn.

**✅ Tại sao sử dụng Bộ chọn nhóm?**

1. **Tiết kiệm mã nguồn**:
   * Nếu bạn muốn áp dụng cùng một kiểu cho nhiều phần tử, thay vì phải viết lại các quy tắc CSS cho từng phần tử, bạn có thể nhóm chúng lại và viết một lần.
2. **Giảm sự trùng lặp**:
   * Việc nhóm các phần tử lại giúp giảm sự trùng lặp trong mã CSS, làm cho mã của bạn trở nên **dễ đọc** và **dễ bảo trì**.
3. **Cải thiện hiệu suất**:
   * Trình duyệt sẽ phải duyệt qua mã CSS ít hơn khi bạn sử dụng bộ chọn nhóm, giúp **tăng hiệu suất** khi tải trang web.

**✅ Lưu ý khi sử dụng Bộ chọn nhóm**

1. **Cẩn thận khi nhóm nhiều phần tử không liên quan**:
   * Khi sử dụng bộ chọn nhóm, bạn cần chắc chắn rằng tất cả các phần tử trong nhóm đó **thực sự cần cùng một kiểu dáng**. Nếu không, việc nhóm chúng lại có thể làm cho CSS trở nên không hợp lý hoặc không đáp ứng đúng yêu cầu thiết kế.
2. **Không thể nhóm các thuộc tính khác nhau**:
   * Bộ chọn nhóm chỉ có thể áp dụng các thuộc tính **giống nhau** cho các phần tử. Ví dụ, bạn không thể nhóm một thẻ <h1> và một lớp .button và hy vọng chúng sẽ có các kiểu dáng khác nhau như màu nền, vì điều này sẽ gây ra sự không nhất quán trong thiết kế.
3. **Sử dụng bộ chọn nhóm cho các phần tử cùng loại**:
   * Thường thì bộ chọn nhóm sẽ hiệu quả nhất khi bạn nhóm các phần tử cùng loại (ví dụ, các thẻ tiêu đề như <h1>, <h2>, hoặc các đoạn văn <p>). Sự nhóm này sẽ giúp bạn dễ dàng duy trì kiểu dáng cho tất cả các phần tử mà không cần phải viết mã lặp lại.

**✅ Ví dụ mở rộng về Bộ chọn nhóm**

**HTML:**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ Bộ chọn nhóm mở rộng</title>

<style>

/\* Bộ chọn nhóm các thẻ tiêu đề \*/

h1, h2, h3 {

color: #555;

text-align: center;

}

/\* Bộ chọn nhóm các thẻ đoạn văn \*/

p, span, li {

font-size: 14px;

line-height: 1.6;

}

/\* Bộ chọn nhóm các nút \*/

.btn-primary, .btn-secondary {

padding: 12px 24px;

background-color: #007BFF;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

}

</style>

</head>

<body>

<h1>Đây là tiêu đề cấp 1</h1>

<h2>Đây là tiêu đề cấp 2</h2>

<h3>Đây là tiêu đề cấp 3</h3>

<p>Đây là đoạn văn bản đầu tiên.</p>

<span>Đây là một span</span>

<ul>

<li>Danh sách 1</li>

<li>Danh sách 2</li>

</ul>

<button class="btn-primary">Nút chính</button>

<button class="btn-secondary">Nút phụ</button>

</body>

</html>

**Giải thích:**

* **h1, h2, h3 { ... }** sẽ làm tất cả các tiêu đề có **màu #555** và **canh giữa**.
* **p, span, li { ... }** sẽ áp dụng **font size 14px** và **line-height 1.6** cho tất cả các đoạn văn, span, và mục trong danh sách.
* **.btn-primary, .btn-secondary { ... }** sẽ tạo ra kiểu dáng giống nhau cho các nút với **màu nền xanh**, **chữ màu trắng**, và **các góc bo tròn**.

**✅ Tóm tắt về Bộ chọn nhóm (Group Selector):**

| **Tính năng** | **Mô tả** |
| --- | --- |
| **Cú pháp** | A, B, C { ... } |
| **Công dụng** | Nhóm các phần tử lại với nhau để áp dụng cùng một bộ thuộc tính CSS. |
| **Ưu điểm** | Tiết kiệm mã nguồn, giảm sự trùng lặp, dễ bảo trì. |
| **Lưu ý** | Chỉ nên nhóm các phần tử có kiểu dáng tương tự, tránh nhóm các phần tử có kiểu dáng khác nhau. |

Hy vọng bạn đã hiểu rõ về **Bộ chọn nhóm (Group Selector)** trong CSS! Nếu có bất kỳ câu hỏi nào, đừng ngần ngại chia sẻ với mình nhé! 😊

**Bạn đã nói:**

Bạn hãy cho mình ví dụ minh họa bao quát về "Bộ chọn nhóm (Group Selector)"

**ChatGPT đã nói:**

Chắc chắn rồi! Dưới đây là ví dụ minh họa chi tiết và bao quát về **"Bộ chọn nhóm (Group Selector)"** trong CSS. Mình sẽ sử dụng nhiều loại phần tử khác nhau như thẻ tiêu đề, đoạn văn, và nút để bạn thấy rõ cách sử dụng bộ chọn nhóm trong thực tế.

**✅ HTML:**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ Bộ chọn nhóm (Group Selector)</title>

<style>

/\* Bộ chọn nhóm cho các thẻ tiêu đề \*/

h1, h2, h3 {

color: #333;

text-align: center;

}

/\* Bộ chọn nhóm cho các đoạn văn và span \*/

p, span, li {

font-size: 14px;

line-height: 1.6;

color: #555;

}

/\* Bộ chọn nhóm cho các nút \*/

.btn-primary, .btn-secondary {

padding: 12px 24px;

background-color: #007BFF;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

}

/\* Đặc biệt cho các nút phụ \*/

.btn-secondary {

background-color: #6c757d;

}

</style>

</head>

<body>

<h1>Tiêu đề cấp 1</h1>

<h2>Tiêu đề cấp 2</h2>

<h3>Tiêu đề cấp 3</h3>

<p>Đây là một đoạn văn ngắn mô tả nội dung trên trang web.</p>

<span>Đây là một span element, cũng có kiểu dáng giống như p.</span>

<ul>

<li>Mục đầu tiên trong danh sách</li>

<li>Mục thứ hai trong danh sách</li>

</ul>

<button class="btn-primary">Nút Chính</button>

<button class="btn-secondary">Nút Phụ</button>

</body>

</html>

**✅ Giải thích chi tiết về Bộ chọn nhóm trong ví dụ:**

1. **h1, h2, h3 { ... }**:
   * Bộ chọn này sẽ áp dụng **một kiểu dáng chung** cho tất cả các thẻ tiêu đề <h1>, <h2>, và <h3>.
   * Các thuộc tính CSS được áp dụng:
     + **Màu chữ (color)**: Các thẻ tiêu đề sẽ có màu **#333** (màu xám đậm).
     + **Canh giữa (text-align: center)**: Các thẻ tiêu đề sẽ được canh giữa trên trang.
2. **p, span, li { ... }**:
   * Bộ chọn này sẽ áp dụng kiểu cho các thẻ <p>, <span>, và các phần tử danh sách <li>.
   * Các thuộc tính CSS được áp dụng:
     + **Kích thước chữ (font-size: 14px)**: Các phần tử này sẽ có cỡ chữ 14px.
     + **Khoảng cách dòng (line-height: 1.6)**: Dòng chữ sẽ có khoảng cách 1.6 giữa các dòng.
     + **Màu chữ (color)**: Màu chữ sẽ là **#555** (màu xám nhẹ).
3. **.btn-primary, .btn-secondary { ... }**:
   * Bộ chọn này áp dụng kiểu cho hai lớp **.btn-primary** và **.btn-secondary** (các nút trên trang).
   * Các thuộc tính CSS được áp dụng:
     + **Padding**: Cung cấp khoảng cách giữa nội dung và viền của nút (12px ở trên và dưới, 24px ở trái và phải).
     + **Màu nền (background-color)**: Màu nền cho các nút chính là **#007BFF** (màu xanh da trời), còn nút phụ có màu nền **#6c757d** (màu xám).
     + **Màu chữ (color)**: Chữ màu **trắng**.
     + **Bo góc (border-radius)**: Bo tròn các góc của nút với bán kính **5px**.
     + **Con trỏ chuột (cursor: pointer)**: Khi di chuột qua nút, con trỏ sẽ thay đổi thành **hình tay**.

**✅ Kết quả khi hiển thị trên trình duyệt:**

1. Các **thẻ tiêu đề <h1>, <h2>, <h3>** sẽ có **màu chữ #333**, **căn giữa** trên trang.
2. Các phần tử **đoạn văn <p>**, **<span>**, và **mục trong danh sách <li>** sẽ có **màu chữ #555**, **cỡ chữ 14px**, và **khoảng cách dòng 1.6**.
3. Các nút **có lớp .btn-primary và .btn-secondary** sẽ có **padding**, **màu nền** và **bo góc** đẹp mắt. Nút chính sẽ có màu nền xanh da trời, trong khi nút phụ có màu nền xám.

**✅ Tại sao sử dụng Bộ chọn nhóm?**

* **Tiết kiệm mã nguồn**: Bạn không phải lặp lại cùng một kiểu cho các phần tử giống nhau. Bạn có thể nhóm chúng lại và chỉ viết một bộ chọn CSS duy nhất cho tất cả.
* **Giảm sự trùng lặp**: Thay vì viết riêng các quy tắc cho từng phần tử, bạn có thể nhóm các phần tử lại để giữ mã nguồn **ngắn gọn và dễ duy trì**.
* **Tăng hiệu quả**: Bộ chọn nhóm giúp giảm số lượng dòng mã CSS, từ đó có thể **tăng hiệu suất** tải trang web.

**✅ Tóm tắt về Bộ chọn nhóm (Group Selector):**

| **Tính năng** | **Mô tả** |
| --- | --- |
| **Cú pháp** | A, B, C { ... } (Chọn các phần tử A, B, C và áp dụng cùng một kiểu dáng) |
| **Công dụng** | Nhóm các phần tử lại với nhau và áp dụng **các thuộc tính giống nhau** cho tất cả chúng. |
| **Ưu điểm** | Tiết kiệm mã nguồn, giảm sự trùng lặp, dễ bảo trì và tăng hiệu suất tải trang. |
| **Lưu ý** | Chỉ nên nhóm các phần tử có kiểu dáng tương tự, tránh nhóm các phần tử có kiểu dáng khác nhau. |

Hy vọng ví dụ này giúp bạn hiểu rõ hơn về **Bộ chọn nhóm (Group Selector)** trong CSS! Nếu có bất kỳ câu hỏi nào, đừng ngần ngại chia sẻ với mình nhé! 😊

# **D. BỘ CHỌN THUỘC TÍNH**

## **1. BỘ CHỌN THUỘC TÍNH [ATTR]**

Bạn hãy giải thích chi tiết tất tần tật, sâu rộng toàn bộ và toàn diện kiến thức về "Bộ chọn thuộc tính: [attr]" và những lưu ý (nếu có) về nó để cho người non-tech hiểu sâu và rõ.

**ChatGPT đã nói:**

**Bộ chọn thuộc tính CSS: [attr]**

Trong CSS, **Bộ chọn thuộc tính** (Attribute Selector) là một công cụ mạnh mẽ giúp bạn chọn các phần tử HTML dựa trên các thuộc tính của chúng. Cụ thể hơn, **[attr]** là cú pháp để áp dụng các kiểu dáng cho các phần tử có một thuộc tính nhất định.

Hãy cùng tìm hiểu chi tiết và toàn diện về bộ chọn thuộc tính này để bạn có thể hiểu rõ cách hoạt động của nó!

**🎯 Bộ chọn thuộc tính là gì?**

**Bộ chọn thuộc tính** cho phép bạn chọn các phần tử HTML dựa trên sự có mặt hoặc giá trị của một thuộc tính cụ thể. Thay vì chỉ chọn các phần tử dựa trên thẻ, lớp, hoặc ID, bạn có thể chọn các phần tử có thuộc tính nào đó, ví dụ như **href**, **src**, **alt**, **class**, và nhiều thuộc tính khác.

Cú pháp cơ bản của bộ chọn thuộc tính là:

css

Sao chépChỉnh sửa

[attribute]

Trong đó:

* **attribute** là tên thuộc tính mà bạn muốn áp dụng bộ chọn.

Bằng cách này, bạn có thể chọn tất cả các phần tử có thuộc tính đó, bất kể giá trị của thuộc tính là gì.

**🎯 Các loại Bộ chọn thuộc tính**

Có nhiều cách sử dụng bộ chọn thuộc tính, và bạn có thể chọn các phần tử dựa trên:

* **Sự có mặt của thuộc tính**.
* **Giá trị của thuộc tính**.
* **Giá trị của thuộc tính bắt đầu, kết thúc, hoặc chứa một chuỗi nào đó**.

**1. Bộ chọn dựa trên sự có mặt của thuộc tính**

Cú pháp: [attr]

* **Sự có mặt của thuộc tính**: Chọn tất cả các phần tử có thuộc tính attr bất kể giá trị của nó.

Ví dụ:

css

Sao chépChỉnh sửa

/\* Chọn tất cả các thẻ <a> có thuộc tính href \*/

a[href] {

color: blue;

}

Trong ví dụ này, tất cả các thẻ <a> có thuộc tính href sẽ có màu chữ xanh.

**2. Bộ chọn dựa trên giá trị thuộc tính**

Cú pháp: [attr="value"]

* **Giá trị chính xác**: Chọn các phần tử có thuộc tính attr với một giá trị cụ thể.

Ví dụ:

css

Sao chépChỉnh sửa

/\* Chọn tất cả các thẻ <a> có thuộc tính href bằng "https://example.com" \*/

a[href="https://example.com"] {

color: red;

}

Trong ví dụ này, chỉ các thẻ <a> có thuộc tính href bằng **"**[**https://example.com**](https://example.com)**"** mới được chọn và có màu chữ đỏ.

**3. Bộ chọn bắt đầu với giá trị thuộc tính**

Cú pháp: [attr^="value"]

* **Bắt đầu với giá trị**: Chọn tất cả các phần tử có thuộc tính attr bắt đầu với một giá trị cụ thể.

Ví dụ:

css

Sao chépChỉnh sửa

/\* Chọn tất cả các thẻ <a> có thuộc tính href bắt đầu với "https://" \*/

a[href^="https://"] {

color: green;

}

Trong ví dụ này, tất cả các thẻ <a> có href bắt đầu với **"https://"** sẽ có màu chữ xanh lá.

**4. Bộ chọn kết thúc với giá trị thuộc tính**

Cú pháp: [attr$="value"]

* **Kết thúc với giá trị**: Chọn tất cả các phần tử có thuộc tính attr kết thúc với một giá trị cụ thể.

Ví dụ:

css

Sao chépChỉnh sửa

/\* Chọn tất cả các thẻ <a> có thuộc tính href kết thúc với ".com" \*/

a[href$=".com"] {

color: purple;

}

Trong ví dụ này, tất cả các thẻ <a> có href kết thúc bằng **".com"** sẽ có màu chữ tím.

**5. Bộ chọn chứa giá trị trong thuộc tính**

Cú pháp: [attr\*="value"]

* **Chứa giá trị**: Chọn tất cả các phần tử có thuộc tính attr chứa một chuỗi giá trị cụ thể ở bất kỳ đâu trong thuộc tính đó.

Ví dụ:

css

Sao chépChỉnh sửa

/\* Chọn tất cả các thẻ <a> có thuộc tính href chứa "example" \*/

a[href\*="example"] {

color: orange;

}

Trong ví dụ này, tất cả các thẻ <a> có href chứa từ **"example"** sẽ có màu chữ cam.

**6. Kết hợp nhiều bộ chọn thuộc tính**

Bạn có thể kết hợp nhiều bộ chọn thuộc tính để áp dụng kiểu dáng cho các phần tử có các thuộc tính và giá trị cụ thể.

Ví dụ:

css

Sao chépChỉnh sửa

/\* Chọn tất cả các thẻ <a> có thuộc tính href bắt đầu với "https://" và chứa "example" \*/

a[href^="https://"][href\*="example"] {

font-weight: bold;

}

Trong ví dụ này, tất cả các thẻ <a> có thuộc tính href bắt đầu với **"https://"** và chứa từ **"example"** sẽ có **chữ đậm**.

**🎯 Ví dụ minh họa chi tiết về Bộ chọn thuộc tính**

**HTML:**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ Bộ chọn thuộc tính</title>

<style>

/\* Chọn tất cả các thẻ <a> có thuộc tính href \*/

a[href] {

color: blue;

}

/\* Chọn tất cả các thẻ <a> có thuộc tính href bằng "https://example.com" \*/

a[href="https://example.com"] {

color: red;

}

/\* Chọn tất cả các thẻ <a> có thuộc tính href bắt đầu với "https://" \*/

a[href^="https://"] {

font-weight: bold;

}

/\* Chọn tất cả các thẻ <a> có thuộc tính href kết thúc với ".org" \*/

a[href$=".org"] {

text-decoration: underline;

}

/\* Chọn tất cả các thẻ <a> có thuộc tính href chứa "example" \*/

a[href\*="example"] {

color: green;

}

</style>

</head>

<body>

<a href="https://example.com">Example 1</a><br>

<a href="https://example.org">Example 2</a><br>

<a href="https://example.net">Example 3</a><br>

<a href="http://test.com">Test 1</a><br>

</body>

</html>

**Giải thích:**

1. **Tất cả các thẻ <a> có thuộc tính href** sẽ có màu **xanh dương** (color: blue).
2. **Chỉ các thẻ <a> có href="https://example.com"** sẽ có màu **đỏ** (color: red).
3. **Các thẻ <a> có href bắt đầu bằng "https://"** sẽ có **chữ đậm** (font-weight: bold).
4. **Các thẻ <a> có href kết thúc bằng ".org"** sẽ có **gạch chân** (text-decoration: underline).
5. **Các thẻ <a> có href chứa "example"** sẽ có màu **xanh lá cây** (color: green).

**🎯 Lưu ý khi sử dụng Bộ chọn thuộc tính**

1. **Hiệu suất**:
   * Các bộ chọn thuộc tính có thể ảnh hưởng đến hiệu suất nếu bạn áp dụng chúng trên nhiều phần tử. Hãy sử dụng chúng hợp lý để tránh làm giảm tốc độ tải trang.
2. **Độ cụ thể của bộ chọn**:
   * Bộ chọn thuộc tính có độ cụ thể cao hơn so với bộ chọn thẻ, nhưng thấp hơn so với bộ chọn ID. Do đó, khi sử dụng bộ chọn thuộc tính, bạn cần chú ý đến thứ tự của các quy tắc CSS để tránh bị ghi đè.
3. **Thực tế sử dụng**:
   * Bộ chọn thuộc tính rất hữu ích khi bạn muốn chọn các phần tử dựa trên giá trị của thuộc tính, chẳng hạn như các liên kết có URL chứa từ khóa nhất định hoặc các hình ảnh có thuộc tính alt.
4. **Chỉ áp dụng cho các thuộc tính HTML**:
   * Bộ chọn thuộc tính chỉ hoạt động với các thuộc tính HTML. Bạn không thể sử dụng bộ chọn thuộc tính cho các kiểu lớp CSS, vì chúng không phải là thuộc tính của các phần tử.

**🎯 Tóm tắt về Bộ chọn thuộc tính**

| **Tính năng** | **Mô tả** |
| --- | --- |
| **Cú pháp** | [attribute], [attribute="value"], [attribute^="value"], [attribute$="value"], [attribute\*="value"] |
| **Công dụng** | Chọn các phần tử dựa trên sự có mặt hoặc giá trị của thuộc tính |
| **Ưu điểm** | Dễ dàng chọn các phần tử có thuộc tính cụ thể mà không cần phải thêm lớp hoặc ID |
| **Lưu ý** | Cần sử dụng hợp lý để tránh ảnh hưởng đến hiệu suất và tránh sự xung đột với các quy tắc CSS khác |

Hy vọng rằng với những giải thích và ví dụ minh họa chi tiết này, bạn đã hiểu rõ về **Bộ chọn thuộc tính (Attribute Selector)** trong CSS! Nếu có bất kỳ câu hỏi nào, đừng ngần ngại để lại câu hỏi nhé! 😊

**Bạn đã nói:**

Bạn hãy cho mình ví dụ minh họa bao quát về "Bộ chọn thuộc tính (Attribute Selector) "

**ChatGPT đã nói:**

Dưới đây là một ví dụ minh họa bao quát về **Bộ chọn thuộc tính (Attribute Selector)** trong CSS. Ví dụ này sẽ giúp bạn hiểu rõ cách sử dụng các bộ chọn thuộc tính khác nhau để áp dụng các kiểu dáng cho các phần tử HTML dựa trên thuộc tính của chúng.

**HTML:**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ Bộ chọn thuộc tính</title>

<style>

/\* Chọn tất cả các thẻ <a> có thuộc tính href \*/

a[href] {

color: blue;

}

/\* Chọn tất cả các thẻ <a> có thuộc tính href bằng "https://example.com" \*/

a[href="https://example.com"] {

color: red;

}

/\* Chọn tất cả các thẻ <a> có thuộc tính href bắt đầu với "https://" \*/

a[href^="https://"] {

font-weight: bold;

}

/\* Chọn tất cả các thẻ <a> có thuộc tính href kết thúc với ".org" \*/

a[href$=".org"] {

text-decoration: underline;

}

/\* Chọn tất cả các thẻ <a> có thuộc tính href chứa "example" \*/

a[href\*="example"] {

color: green;

}

/\* Chọn tất cả các thẻ <img> có thuộc tính src chứa "image" \*/

img[src\*="image"] {

border: 5px solid #000;

}

</style>

</head>

<body>

<h1>Ví dụ về Bộ chọn thuộc tính trong CSS</h1>

<a href="https://example.com">Link 1 (https://example.com)</a><br>

<a href="https://example.org">Link 2 (https://example.org)</a><br>

<a href="https://test.com">Link 3 (https://test.com)</a><br>

<a href="http://anotherexample.com">Link 4 (http://anotherexample.com)</a><br>

<img src="images/sample-image.jpg" alt="Sample Image 1">

<img src="images/another-image.jpg" alt="Sample Image 2">

<img src="image/misc.jpg" alt="Sample Image 3">

</body>

</html>

**Giải thích chi tiết về các bộ chọn thuộc tính trong ví dụ trên:**

1. **Chọn tất cả các thẻ <a> có thuộc tính href**:

css

Sao chépChỉnh sửa

a[href] {

color: blue;

}

* + **Ý nghĩa**: Bộ chọn này sẽ áp dụng kiểu dáng cho tất cả các thẻ <a> có thuộc tính href (dù giá trị của nó là gì).
  + **Kết quả**: Tất cả các thẻ <a> trong trang có thuộc tính href sẽ có **màu chữ xanh dương**.

1. **Chọn tất cả các thẻ <a> có thuộc tính href bằng "**[**https://example.com**](https://example.com)**"**:

css

Sao chépChỉnh sửa

a[href="https://example.com"] {

color: red;

}

* + **Ý nghĩa**: Bộ chọn này chỉ áp dụng kiểu dáng cho thẻ <a> có thuộc tính href chính xác bằng **"**[**https://example.com**](https://example.com)**"**.
  + **Kết quả**: Thẻ <a> có href="https://example.com" sẽ có **màu chữ đỏ**.

1. **Chọn tất cả các thẻ <a> có thuộc tính href bắt đầu với "https://"**:

css

Sao chépChỉnh sửa

a[href^="https://"] {

font-weight: bold;

}

* + **Ý nghĩa**: Bộ chọn này sẽ áp dụng kiểu dáng cho tất cả các thẻ <a> có thuộc tính href bắt đầu với chuỗi **"https://"**.
  + **Kết quả**: Các thẻ <a> có href bắt đầu với "https://" sẽ có **chữ đậm**.

1. **Chọn tất cả các thẻ <a> có thuộc tính href kết thúc với ".org"**:

css

Sao chépChỉnh sửa

a[href$=".org"] {

text-decoration: underline;

}

* + **Ý nghĩa**: Bộ chọn này sẽ áp dụng kiểu dáng cho các thẻ <a> có thuộc tính href kết thúc với **".org"**.
  + **Kết quả**: Các thẻ <a> có href kết thúc bằng ".org" sẽ có **gạch chân**.

1. **Chọn tất cả các thẻ <a> có thuộc tính href chứa từ "example"**:

css

Sao chépChỉnh sửa

a[href\*="example"] {

color: green;

}

* + **Ý nghĩa**: Bộ chọn này sẽ áp dụng kiểu dáng cho tất cả các thẻ <a> có thuộc tính href chứa chuỗi **"example"** ở bất kỳ đâu trong giá trị của thuộc tính href.
  + **Kết quả**: Các thẻ <a> có href chứa "example" sẽ có **màu chữ xanh lá cây**.

1. **Chọn tất cả các thẻ <img> có thuộc tính src chứa từ "image"**:

css

Sao chépChỉnh sửa

img[src\*="image"] {

border: 5px solid #000;

}

* + **Ý nghĩa**: Bộ chọn này sẽ áp dụng kiểu dáng cho tất cả các thẻ <img> có thuộc tính src chứa từ **"image"**.
  + **Kết quả**: Tất cả các hình ảnh có src chứa từ "image" sẽ có **viền đen** dày 5px.

**Kết quả khi hiển thị trên trình duyệt:**

1. **Các liên kết <a>**:
   * **Link 1** (<https://example.com>) có màu chữ **đỏ**.
   * **Link 2** (<https://example.org>) có màu chữ **xanh dương**.
   * **Link 3** (<https://test.com>) có màu chữ **xanh dương** và **chữ đậm** vì có href bắt đầu bằng "https://".
   * **Link 4** (<http://anotherexample.com>) có màu chữ **xanh dương** và **chữ đậm**.
2. **Các hình ảnh <img>**:
   * **Image 1 và Image 2** có thuộc tính src chứa từ **"image"**, vì vậy chúng sẽ có **viền đen**.
   * **Image 3** cũng có thuộc tính src chứa từ **"image"** và được áp dụng **viền đen**.

**Tóm tắt về Bộ chọn thuộc tính:**

* **[attr]**: Chọn các phần tử có thuộc tính attr bất kể giá trị của nó.
* **[attr="value"]**: Chọn các phần tử có thuộc tính attr có giá trị chính xác là "value".
* **[attr^="value"]**: Chọn các phần tử có thuộc tính attr bắt đầu với "value".
* **[attr$="value"]**: Chọn các phần tử có thuộc tính attr kết thúc với "value".
* \**[attr*="value"]\*\*: Chọn các phần tử có thuộc tính attr chứa "value" ở bất kỳ đâu trong giá trị của thuộc tính.

Hy vọng ví dụ trên giúp bạn hiểu rõ hơn về **Bộ chọn thuộc tính (Attribute Selector)** trong CSS! Nếu bạn có bất kỳ câu hỏi nào, đừng ngần ngại chia sẻ với mình nhé! 😊

## **2. BỘ CHỌN THUỘC TÍNH: [ATTR="VALUE"]**

**Bộ chọn thuộc tính CSS: [attr="value"]**

**Bộ chọn thuộc tính CSS** cho phép bạn chọn các phần tử HTML dựa trên các thuộc tính của chúng, thay vì chỉ chọn bằng cách sử dụng thẻ, lớp (class) hay ID. Một trong những loại bộ chọn thuộc tính mạnh mẽ là bộ chọn theo giá trị của thuộc tính, với cú pháp [attr="value"], giúp bạn chọn những phần tử có thuộc tính attr có giá trị chính xác bằng value.

**🎯 Cú pháp và cấu trúc của bộ chọn thuộc tính [attr="value"]**

Cú pháp của bộ chọn thuộc tính này rất đơn giản:

css

Sao chépChỉnh sửa

[attr="value"]

Trong đó:

* **attr** là tên của thuộc tính mà bạn muốn chọn.
* **value** là giá trị của thuộc tính mà bạn muốn tìm.

**Ví dụ cụ thể:**

Giả sử bạn muốn chọn tất cả các liên kết có thuộc tính href có giá trị chính xác là "<https://example.com>". Bạn sẽ viết mã CSS như sau:

css

Sao chépChỉnh sửa

a[href="https://example.com"] {

color: red;

}

**Giải thích:**

* **a**: Đây là phần tử mà bạn muốn áp dụng kiểu dáng. Trong ví dụ trên, là tất cả các thẻ <a> (thẻ liên kết).
* **[href="https://example.com"]**: Bộ chọn này chỉ áp dụng cho các thẻ <a> có thuộc tính href có giá trị chính xác là **"**[**https://example.com**](https://example.com)**"**.
* **color: red;**: Chỉ định rằng màu chữ của các thẻ <a> này sẽ là **đỏ**.

**🎯 Khi nào sử dụng [attr="value"]?**

Bộ chọn này rất hữu ích khi bạn muốn áp dụng kiểu dáng cho các phần tử mà có một thuộc tính với một giá trị chính xác. Ví dụ:

* Bạn có thể muốn thay đổi màu sắc của tất cả các liên kết dẫn tới một trang web cụ thể.
* Bạn có thể muốn thay đổi kiểu dáng của tất cả các hình ảnh có thuộc tính src trỏ đến một thư mục hoặc tệp cụ thể.

**🎯 Ví dụ minh họa chi tiết**

**1. Thay đổi màu sắc của các liên kết có href bằng "**[**https://example.com**](https://example.com)**"**

HTML:

html

Sao chépChỉnh sửa

<a href="https://example.com">Liên kết 1</a>

<a href="https://example.org">Liên kết 2</a>

<a href="https://example.com">Liên kết 3</a>

CSS:

css

Sao chépChỉnh sửa

a[href="https://example.com"] {

color: red;

}

**Giải thích**:

* Chỉ các thẻ <a> có thuộc tính href bằng **"**[**https://example.com**](https://example.com)**"** sẽ có màu chữ **đỏ**.
* Kết quả là **Liên kết 1** và **Liên kết 3** sẽ có màu đỏ, trong khi **Liên kết 2** sẽ không thay đổi.

**2. Chọn các hình ảnh có thuộc tính src bằng một giá trị nhất định**

HTML:

html

Sao chépChỉnh sửa

<img src="images/cat.jpg" alt="Cat">

<img src="images/dog.jpg" alt="Dog">

<img src="images/cat.jpg" alt="Cat">

CSS:

css

Sao chépChỉnh sửa

img[src="images/cat.jpg"] {

border: 3px solid blue;

}

**Giải thích**:

* Bộ chọn [src="images/cat.jpg"] chọn tất cả các thẻ <img> có thuộc tính src có giá trị chính xác là **"images/cat.jpg"**.
* Tất cả các thẻ <img> có src="images/cat.jpg" sẽ có **viền xanh dương**. Vì vậy, hình ảnh có thuộc tính src="images/cat.jpg" sẽ được áp dụng kiểu dáng này.

**3. Thay đổi màu nền cho các liên kết có title cụ thể**

HTML:

html

Sao chépChỉnh sửa

<a href="#" title="Home">Trang chủ</a>

<a href="#" title="About Us">Giới thiệu</a>

<a href="#" title="Contact">Liên hệ</a>

CSS:

css

Sao chépChỉnh sửa

a[title="Home"] {

background-color: yellow;

}

**Giải thích**:

* Bộ chọn [title="Home"] chọn tất cả các thẻ <a> có thuộc tính title có giá trị chính xác là **"Home"**.
* Thẻ <a> có title="Home" sẽ có nền **màu vàng**.

**🎯 Lưu ý khi sử dụng [attr="value"]**

1. **So khớp chính xác**:
   * Bộ chọn [attr="value"] chỉ áp dụng cho các phần tử có thuộc tính attr có giá trị **chính xác** bằng value. Điều này có nghĩa là chỉ các giá trị khớp hoàn toàn mới được chọn. Ví dụ, nếu bạn muốn chọn các thẻ <a> có href="https://example.com", thì chỉ các thẻ có giá trị chính xác này mới được áp dụng kiểu dáng, chứ không phải các thẻ có giá trị tương tự như "https://example.com/page".
2. **Không phân biệt chữ hoa chữ thường**:
   * Các bộ chọn thuộc tính CSS là **không phân biệt chữ hoa chữ thường**. Điều này có nghĩa là href="https://example.com" và href="https://EXAMPLE.COM" sẽ được coi là giống nhau. Tuy nhiên, điều này có thể thay đổi tùy vào cách sử dụng thuộc tính trong HTML và quy tắc của từng trình duyệt.
3. **Sử dụng cho các thuộc tính có giá trị cố định**:
   * Bộ chọn [attr="value"] rất hiệu quả khi bạn có các thuộc tính có giá trị cố định mà bạn muốn áp dụng kiểu dáng, như href, src, alt, title, và các thuộc tính khác có giá trị định rõ.
4. **Độ ưu tiên**:
   * Bộ chọn [attr="value"] có độ ưu tiên thấp hơn so với các bộ chọn ID (ví dụ #id) và bộ chọn lớp (ví dụ .class). Do đó, bạn cần đảm bảo rằng không có quy tắc CSS khác với độ ưu tiên cao hơn sẽ ghi đè lên các quy tắc của bạn.

**🎯 Tóm tắt về [attr="value"]**

| **Tính năng** | **Mô tả** |
| --- | --- |
| **Cú pháp** | [attr="value"] |
| **Chọn phần tử** | Chọn phần tử có thuộc tính attr có giá trị chính xác bằng value. |
| **Ứng dụng phổ biến** | Chọn các phần tử với giá trị thuộc tính cụ thể (ví dụ: liên kết đến một URL nhất định, hình ảnh có tên tệp nhất định). |
| **Ưu điểm** | Dễ dàng chọn các phần tử có thuộc tính và giá trị xác định mà không cần thêm lớp hoặc ID. |
| **Lưu ý** | Bộ chọn này chỉ hoạt động với các giá trị chính xác và không phân biệt chữ hoa chữ thường. |

Hy vọng giải thích này giúp bạn hiểu rõ hơn về bộ chọn thuộc tính [attr="value"] trong CSS! Nếu bạn có thêm câu hỏi hoặc cần giải thích thêm, đừng ngần ngại chia sẻ với tôi nhé! 😊

Bạn hãy cho mình ví dụ minh họa bao quát về "Bộ chọn thuộc tính: [attr="value"]"

**ChatGPT đã nói:**

Dưới đây là một ví dụ minh họa bao quát về **Bộ chọn thuộc tính: [attr="value"]** trong CSS. Ví dụ này sẽ giúp bạn hiểu cách sử dụng bộ chọn thuộc tính để áp dụng kiểu dáng cho các phần tử dựa trên giá trị của thuộc tính cụ thể.

**HTML:**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ về Bộ chọn thuộc tính: [attr="value"]</title>

<style>

/\* Chọn các liên kết có thuộc tính href có giá trị chính xác là "https://example.com" \*/

a[href="https://example.com"] {

color: red;

font-weight: bold;

}

/\* Chọn các hình ảnh có thuộc tính src có giá trị chính xác là "images/cat.jpg" \*/

img[src="images/cat.jpg"] {

border: 5px solid blue;

}

/\* Chọn các liên kết có thuộc tính title có giá trị chính xác là "Home" \*/

a[title="Home"] {

background-color: yellow;

}

</style>

</head>

<body>

<h1>Ví dụ về Bộ chọn thuộc tính: [attr="value"]</h1>

<a href="https://example.com" title="Home">Liên kết 1 (https://example.com)</a><br>

<a href="https://example.org" title="About Us">Liên kết 2 (https://example.org)</a><br>

<a href="https://example.com" title="Contact">Liên kết 3 (https://example.com)</a><br>

<img src="images/cat.jpg" alt="Cat Image 1">

<img src="images/dog.jpg" alt="Dog Image">

<img src="images/cat.jpg" alt="Cat Image 2">

</body>

</html>

**Giải thích chi tiết về ví dụ trên:**

1. **Bộ chọn liên kết có thuộc tính href="https://example.com"**:

css

Sao chépChỉnh sửa

a[href="https://example.com"] {

color: red;

font-weight: bold;

}

* + **Ý nghĩa**: Bộ chọn này sẽ áp dụng kiểu dáng **màu đỏ** và **chữ đậm** cho tất cả các thẻ <a> có thuộc tính href có giá trị chính xác là **"**[**https://example.com**](https://example.com)**"**.
  + **Kết quả**:
    - **Liên kết 1** và **Liên kết 3** sẽ có **màu chữ đỏ** và **chữ đậm**.
    - **Liên kết 2** sẽ không thay đổi vì href="https://example.org" không khớp với bộ chọn.

1. **Bộ chọn hình ảnh có thuộc tính src="images/cat.jpg"**:

css

Sao chépChỉnh sửa

img[src="images/cat.jpg"] {

border: 5px solid blue;

}

* + **Ý nghĩa**: Bộ chọn này sẽ áp dụng kiểu dáng **viền xanh dương dày 5px** cho tất cả các thẻ <img> có thuộc tính src có giá trị chính xác là **"images/cat.jpg"**.
  + **Kết quả**:
    - Các thẻ <img> có src="images/cat.jpg" (hình ảnh con mèo) sẽ có **viền xanh dương**.
    - Thẻ <img> có src="images/dog.jpg" sẽ không thay đổi vì giá trị src không khớp.

1. **Bộ chọn liên kết có thuộc tính title="Home"**:

css

Sao chépChỉnh sửa

a[title="Home"] {

background-color: yellow;

}

* + **Ý nghĩa**: Bộ chọn này sẽ áp dụng kiểu dáng **màu nền vàng** cho tất cả các thẻ <a> có thuộc tính title có giá trị chính xác là **"Home"**.
  + **Kết quả**:
    - **Liên kết 1** sẽ có **màu nền vàng** vì title="Home".
    - **Liên kết 2** và **Liên kết 3** sẽ không thay đổi vì giá trị title của chúng không phải là "Home".

**Kết quả khi hiển thị trên trình duyệt:**

1. **Liên kết 1 (**[**https://example.com**](https://example.com)**)**:
   * Màu chữ **đỏ** và **chữ đậm**.
   * Màu nền **vàng** vì thuộc tính title="Home".
2. **Liên kết 2 (**[**https://example.org**](https://example.org)**)**:
   * Không thay đổi màu sắc vì không có thuộc tính href="https://example.com" và title="Home".
3. **Liên kết 3 (**[**https://example.com**](https://example.com)**)**:
   * Màu chữ **đỏ** và **chữ đậm**.
   * Không thay đổi màu nền vì title="Contact".
4. **Hình ảnh 1 (src="images/cat.jpg")**:
   * **Viền xanh dương** vì src="images/cat.jpg".
5. **Hình ảnh 2 (src="images/dog.jpg")**:
   * Không thay đổi vì src="images/dog.jpg" không khớp.
6. **Hình ảnh 3 (src="images/cat.jpg")**:
   * **Viền xanh dương** vì src="images/cat.jpg".

**Tóm tắt về Bộ chọn thuộc tính [attr="value"]:**

* Bộ chọn này dùng để áp dụng kiểu dáng cho phần tử dựa trên giá trị chính xác của một thuộc tính cụ thể.
* Các thuộc tính có thể là href, src, title, alt, và nhiều thuộc tính khác.
* Bộ chọn này giúp bạn dễ dàng áp dụng các kiểu dáng cụ thể cho các phần tử có giá trị thuộc tính nhất định mà không cần phải sử dụng lớp (class) hay ID.

Hy vọng ví dụ trên giúp bạn hiểu rõ hơn về **Bộ chọn thuộc tính [attr="value"]** trong CSS! Nếu bạn có câu hỏi nào thêm hoặc muốn thử thêm ví dụ khác, đừng ngần ngại yêu cầu nhé! 😊

## **3. BỘ CHỌN THUỘC TÍNH [ATTR~="VALUE"]**

Hãy giải thích chi tiết tất tần tật, sâu rộng toàn bộ và toàn diện kiến thức về "BỘ CHỌN THUỘC TÍNH [ATTR~="VALUE"]" và những lưu ý (nếu có) về nó để cho người non-tech hiểu sâu và rõ.

**ChatGPT đã nói:**

**Bộ chọn thuộc tính CSS [attr~="value"]**

Bộ chọn thuộc tính [attr~="value"] trong CSS là một công cụ mạnh mẽ giúp bạn chọn các phần tử dựa trên việc chúng có chứa giá trị cụ thể trong thuộc tính của chúng hay không, nhưng với một điều kiện là giá trị này phải là một phần trong chuỗi các giá trị được phân tách bằng dấu cách.

**🎯 Cú pháp và cấu trúc của [attr~="value"]**

Cú pháp của bộ chọn này như sau:

css

Sao chépChỉnh sửa

[attr~="value"]

Trong đó:

* **attr**: là tên của thuộc tính mà bạn muốn chọn.
* **value**: là giá trị bạn muốn tìm kiếm trong thuộc tính của phần tử, nhưng chú ý rằng nó sẽ tìm kiếm nếu giá trị này tồn tại như một phần trong một danh sách các giá trị được phân tách bằng dấu cách.

**🎯 Cách thức hoạt động của [attr~="value"]**

Bộ chọn [attr~="value"] chỉ áp dụng khi thuộc tính có giá trị là một chuỗi các từ được phân tách bằng dấu cách, và bộ chọn sẽ tìm **một trong những từ** trong chuỗi đó khớp với **value** mà bạn chỉ định.

**Lưu ý quan trọng**: Bộ chọn này chỉ hoạt động với các thuộc tính có giá trị là một chuỗi các từ được phân cách bởi dấu cách, như class, rel, data-\*, v.v.

**🎯 Ví dụ về [attr~="value"]**

**Ví dụ 1: Chọn các phần tử có thuộc tính class chứa một lớp cụ thể**

Giả sử bạn muốn chọn tất cả các phần tử có lớp (class) chứa lớp "highlight", nhưng lớp này có thể nằm trong danh sách các lớp của phần tử đó.

**HTML**:

html

Sao chépChỉnh sửa

<div class="box highlight">Box 1</div>

<div class="box">Box 2</div>

<div class="box highlight special">Box 3</div>

**CSS**:

css

Sao chépChỉnh sửa

div[class~="highlight"] {

background-color: yellow;

}

**Giải thích**:

* Bộ chọn [class~="highlight"] sẽ chọn tất cả các phần tử <div> có thuộc tính class chứa từ "highlight" dưới dạng một phần của danh sách các lớp.
* Trong ví dụ này, **Box 1** và **Box 3** sẽ có nền màu **vàng** vì chúng có chứa lớp **"highlight"** trong thuộc tính class. Còn **Box 2** sẽ không thay đổi vì không có lớp "highlight".

**Ví dụ 2: Chọn các liên kết có thuộc tính rel chứa từ "nofollow"**

Giả sử bạn có các liên kết và bạn muốn chọn các liên kết có thuộc tính rel chứa từ khóa "nofollow".

**HTML**:

html

Sao chépChỉnh sửa

<a href="https://example.com" rel="nofollow">Liên kết 1</a>

<a href="https://example.com" rel="noopener noreferrer">Liên kết 2</a>

<a href="https://example.com" rel="nofollow noreferrer">Liên kết 3</a>

**CSS**:

css

Sao chépChỉnh sửa

a[rel~="nofollow"] {

color: red;

}

**Giải thích**:

* Bộ chọn [rel~="nofollow"] sẽ chọn các phần tử <a> có thuộc tính rel chứa từ "nofollow" trong danh sách các giá trị.
* **Liên kết 1** và **Liên kết 3** sẽ có màu chữ **đỏ** vì thuộc tính rel của chúng có chứa từ "nofollow".
* **Liên kết 2** sẽ không thay đổi vì giá trị của rel không chứa từ "nofollow" (chỉ có "noopener" và "noreferrer").

**Ví dụ 3: Chọn các thẻ hình ảnh có thuộc tính alt chứa từ "cat"**

**HTML**:

html

Sao chépChỉnh sửa

<img src="cat.jpg" alt="cute cat">

<img src="dog.jpg" alt="cute dog">

<img src="cat2.jpg" alt="funny cat">

**CSS**:

css

Sao chépChỉnh sửa

img[alt~="cat"] {

border: 3px solid green;

}

**Giải thích**:

* Bộ chọn [alt~="cat"] sẽ chọn các thẻ <img> có thuộc tính alt chứa từ "cat".
* **Hình ảnh đầu tiên** và **hình ảnh thứ ba** sẽ có **viền xanh lá cây** vì thuộc tính alt của chúng chứa từ "cat".
* **Hình ảnh thứ hai** sẽ không thay đổi vì giá trị alt không chứa từ "cat".

**🎯 Những lưu ý khi sử dụng [attr~="value"]**

1. **Giá trị phải là một chuỗi các từ phân cách bằng dấu cách**:
   * Bộ chọn [attr~="value"] chỉ áp dụng cho các thuộc tính có giá trị là một chuỗi các từ được phân cách bằng dấu cách, ví dụ như class, rel, data-\*, v.v.
   * Ví dụ, bạn không thể sử dụng bộ chọn này với thuộc tính src vì src chỉ có một giá trị duy nhất (ví dụ, src="image.jpg").
2. **Tìm kiếm theo từ chính xác**:
   * Bộ chọn [attr~="value"] tìm kiếm chính xác **từng từ** trong chuỗi phân tách bởi dấu cách. Điều này có nghĩa là nó không tìm kiếm một phần của từ. Ví dụ, với class="box highlight", bộ chọn [class~="highlight"] sẽ chọn đúng phần tử này, nhưng nếu bạn dùng [class~="high"] thì không tìm thấy gì, vì "high" không phải là một từ riêng biệt trong chuỗi.
3. **Không phân biệt chữ hoa chữ thường**:
   * Các bộ chọn thuộc tính trong CSS, bao gồm cả [attr~="value"], là **không phân biệt chữ hoa chữ thường**. Điều này có nghĩa là class="highlight" và class="HIGHLIGHT" đều sẽ bị coi là giống nhau.
4. **Không dùng cho các thuộc tính không có giá trị chuỗi phân tách**:
   * Bộ chọn này không hoạt động với các thuộc tính có giá trị không phải chuỗi phân tách bởi dấu cách, chẳng hạn như id, src, v.v.

**🎯 Tóm tắt về [attr~="value"]**

| **Tính năng** | **Mô tả** |
| --- | --- |
| **Cú pháp** | [attr~="value"] |
| **Chọn phần tử** | Chọn phần tử có thuộc tính attr có giá trị là một chuỗi các từ phân cách bằng dấu cách, và trong đó có chứa từ value. |
| **Ứng dụng phổ biến** | Chọn các phần tử dựa trên một từ cụ thể trong thuộc tính như class, rel, data-\*. |
| **Ưu điểm** | Dễ dàng chọn các phần tử có chứa giá trị cụ thể trong danh sách thuộc tính phân tách bằng dấu cách. |
| **Lưu ý** | Bộ chọn này không hoạt động với các thuộc tính không phải là chuỗi các từ phân tách bằng dấu cách, như id, src, v.v. |

**Ví dụ tổng hợp:**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ về Bộ chọn thuộc tính [attr~="value"]</title>

<style>

/\* Chọn tất cả các phần tử có class chứa từ "highlight" \*/

div[class~="highlight"] {

background-color: yellow;

}

/\* Chọn tất cả các liên kết có rel chứa từ "nofollow" \*/

a[rel~="nofollow"] {

color: red;

}

/\* Chọn tất cả các hình ảnh có alt chứa từ "cat" \*/

img[alt~="cat"] {

border: 3px solid green;

}

</style>

</head>

<body>

<div class="box highlight">Box 1</div>

<div class="box">Box 2</div>

<div class="box highlight special">Box 3</div>

<a href="https://example.com" rel="nofollow">Liên kết 1</a>

<a href="https://example.org" rel="noopener noreferrer">Liên kết 2</a>

<a href="https://example.com" rel="nofollow noreferrer">Liên kết 3</a>

<img src="cat.jpg" alt="cute cat">

<img src="dog.jpg" alt="cute dog">

<img src="cat2.jpg" alt="funny cat">

</body>

</html>

Hy vọng bài viết giúp bạn hiểu rõ về **bộ chọn thuộc tính [attr~="value"]** trong CSS. Nếu bạn có thêm câu hỏi nào hoặc muốn tìm hiểu thêm về CSS, đừng ngần ngại yêu cầu nhé! 😊

**Bạn đã nói:**

Bạn hãy cho mình ví dụ minh họa bao quát về bộ chọn thuộc tính [attr~="value"]

**ChatGPT đã nói:**

Dưới đây là ví dụ minh họa bao quát về bộ chọn thuộc tính [attr~="value"] trong CSS. Ví dụ này sẽ giúp bạn hiểu cách sử dụng bộ chọn thuộc tính này để áp dụng các kiểu dáng cho các phần tử dựa trên một giá trị cụ thể trong thuộc tính của chúng.

**HTML:**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ Bộ chọn thuộc tính [attr~="value"]</title>

<style>

/\* Chọn tất cả các phần tử có class chứa từ "highlight" \*/

div[class~="highlight"] {

background-color: yellow;

}

/\* Chọn tất cả các liên kết có rel chứa từ "nofollow" \*/

a[rel~="nofollow"] {

color: red;

}

/\* Chọn tất cả các hình ảnh có alt chứa từ "cat" \*/

img[alt~="cat"] {

border: 3px solid green;

}

</style>

</head>

<body>

<h2>Ví dụ về Bộ chọn thuộc tính [attr~="value"]</h2>

<div class="box highlight">Box 1 - Có class highlight</div>

<div class="box">Box 2 - Không có class highlight</div>

<div class="box highlight special">Box 3 - Có class highlight và special</div>

<a href="https://example.com" rel="nofollow">Liên kết 1 - Có rel nofollow</a><br>

<a href="https://example.org" rel="noopener noreferrer">Liên kết 2 - Không có rel nofollow</a><br>

<a href="https://example.com" rel="nofollow noreferrer">Liên kết 3 - Có rel nofollow</a><br>

<img src="cat.jpg" alt="cute cat">

<img src="dog.jpg" alt="cute dog">

<img src="cat2.jpg" alt="funny cat">

</body>

</html>

**Giải thích chi tiết về bộ chọn thuộc tính [attr~="value"] trong ví dụ trên:**

1. **Bộ chọn cho class chứa từ "highlight"**:

css

Sao chépChỉnh sửa

div[class~="highlight"] {

background-color: yellow;

}

* + **Giải thích**: Bộ chọn này sẽ áp dụng kiểu dáng **màu nền vàng** cho tất cả các thẻ <div> có thuộc tính class chứa từ "highlight". Nó tìm kiếm từ "highlight" trong chuỗi các lớp được phân tách bởi dấu cách trong thuộc tính class.
  + **Kết quả**:
    - **Box 1** và **Box 3** sẽ có **màu nền vàng** vì chúng có lớp "highlight".
    - **Box 2** sẽ không thay đổi vì không có lớp "highlight".

1. **Bộ chọn cho rel chứa từ "nofollow"**:

css

Sao chépChỉnh sửa

a[rel~="nofollow"] {

color: red;

}

* + **Giải thích**: Bộ chọn này sẽ áp dụng **màu chữ đỏ** cho tất cả các thẻ <a> có thuộc tính rel chứa từ "nofollow".
  + **Kết quả**:
    - **Liên kết 1** và **Liên kết 3** sẽ có **màu chữ đỏ** vì thuộc tính rel của chúng có chứa từ "nofollow".
    - **Liên kết 2** sẽ không thay đổi vì rel của nó không chứa từ "nofollow".

1. **Bộ chọn cho alt chứa từ "cat"**:

css

Sao chépChỉnh sửa

img[alt~="cat"] {

border: 3px solid green;

}

* + **Giải thích**: Bộ chọn này sẽ áp dụng **viền xanh lá cây dày 3px** cho tất cả các thẻ <img> có thuộc tính alt chứa từ "cat" trong giá trị của nó.
  + **Kết quả**:
    - **Hình ảnh đầu tiên** và **hình ảnh thứ ba** sẽ có **viền xanh lá cây** vì thuộc tính alt của chúng chứa từ "cat".
    - **Hình ảnh thứ hai** sẽ không thay đổi vì thuộc tính alt của nó không chứa từ "cat".

**Kết quả khi xem trên trình duyệt:**

1. **Box 1 và Box 3**: Có nền **vàng** vì thuộc tính class của chúng chứa từ "highlight".
2. **Box 2**: Không có nền vàng vì thuộc tính class của nó không chứa "highlight".
3. **Liên kết 1 và Liên kết 3**: Có màu chữ **đỏ** vì thuộc tính rel của chúng chứa "nofollow".
4. **Liên kết 2**: Không thay đổi vì thuộc tính rel của nó không chứa "nofollow".
5. **Hình ảnh đầu tiên và hình ảnh thứ ba**: Có **viền xanh lá cây** vì thuộc tính alt chứa từ "cat".
6. **Hình ảnh thứ hai**: Không thay đổi vì thuộc tính alt không chứa từ "cat".

**Tóm tắt về bộ chọn thuộc tính [attr~="value"]:**

* **Công dụng**: Bộ chọn này giúp bạn chọn các phần tử có thuộc tính là một chuỗi các từ phân tách bằng dấu cách, và nó sẽ tìm kiếm **một trong những từ** trong chuỗi đó khớp với giá trị value mà bạn chỉ định.
* **Thường sử dụng cho**: Thuộc tính như class, rel, data-\* và các thuộc tính tương tự có thể chứa nhiều từ được phân tách bằng dấu cách.

Hy vọng ví dụ trên giúp bạn hiểu rõ về cách sử dụng **bộ chọn thuộc tính [attr~="value"]** trong CSS! Nếu bạn có thêm câu hỏi nào hoặc muốn thử thêm ví dụ khác, đừng ngần ngại yêu cầu nhé! 😊

## **4. BỘ CHỌN THUỘC TÍNH [ATTRIBUTE|="VALUE"]**

Hãy giải thích chi tiết tất tần tật, sâu rộng toàn bộ và toàn diện kiến thức về " Bộ chọn thuộc tính: [attribute|="value"]" và những lưu ý (nếu có) về nó để cho người non-tech hiểu sâu và rõ.

**ChatGPT đã nói:**

**Bộ chọn thuộc tính [attribute|="value"] trong CSS**

Bộ chọn thuộc tính [attribute|="value"] là một trong những bộ chọn đặc biệt trong CSS, giúp bạn chọn các phần tử có thuộc tính cụ thể với giá trị bắt đầu bằng một từ xác định, và sau đó có thể có hoặc không có dấu gạch nối (-) sau từ đó.

Cấu trúc của bộ chọn này sẽ tìm các phần tử có thuộc tính có giá trị bắt đầu với một từ khóa cụ thể và nếu có dấu gạch nối thì sẽ là một phần mở rộng của từ khóa đó.

**🎯 Cú pháp và cấu trúc của [attribute|="value"]**

Cú pháp của bộ chọn này là:

css

Sao chépChỉnh sửa

[attribute|="value"]

Trong đó:

* **attribute**: là tên của thuộc tính mà bạn muốn chọn.
* **value**: là giá trị mà bạn muốn tìm kiếm. Bộ chọn này sẽ chọn phần tử nếu giá trị của thuộc tính bắt đầu bằng value và theo sau là một dấu gạch nối - hoặc không có dấu gạch nối.

**🎯 Cách thức hoạt động của [attribute|="value"]**

Bộ chọn [attribute|="value"] tìm kiếm các phần tử có thuộc tính có giá trị bắt đầu với từ value và có thể có thêm một hoặc nhiều phần mở rộng, được phân tách bằng dấu gạch nối. Cụ thể:

* Bộ chọn này tìm phần tử có thuộc tính attribute có giá trị là:
  + **Giá trị chính xác bằng value**.
  + **Giá trị bắt đầu bằng value và theo sau là một dấu gạch nối -** (ví dụ: value-xyz).

Bộ chọn này thường được sử dụng với các thuộc tính như lang, hreflang, rel, v.v.

**🎯 Ví dụ về [attribute|="value"]**

**Ví dụ 1: Chọn các phần tử có thuộc tính lang bắt đầu với "en-"**

Giả sử bạn muốn chọn tất cả các phần tử có thuộc tính lang bắt đầu với "en" (chẳng hạn như "en-US" hoặc "en-GB").

**HTML**:

html

Sao chépChỉnh sửa

<html lang="en-US">

<body>

<h1>Chào bạn!</h1>

</body>

</html>

<html lang="en-GB">

<body>

<h1>Welcome!</h1>

</body>

</html>

<html lang="fr-FR">

<body>

<h1>Bienvenue!</h1>

</body>

</html>

**CSS**:

css

Sao chépChỉnh sửa

html[lang|="en"] {

background-color: yellow;

}

**Giải thích**:

* Bộ chọn [lang|="en"] sẽ chọn tất cả các phần tử <html> có thuộc tính lang bắt đầu bằng en, và có thể có thêm phần mở rộng sau dấu gạch nối (en-US, en-GB, v.v.).
* **Kết quả**:
  + Các trang có lang="en-US" và lang="en-GB" sẽ có nền **vàng** vì giá trị của thuộc tính lang bắt đầu với "en".
  + Trang có lang="fr-FR" sẽ không thay đổi vì giá trị của lang không bắt đầu với "en".

**Ví dụ 2: Chọn các liên kết với thuộc tính rel có giá trị bắt đầu bằng "noopener"**

**HTML**:

html

Sao chépChỉnh sửa

<a href="https://example.com" rel="noopener noreferrer">Liên kết 1</a>

<a href="https://example.com" rel="noopener">Liên kết 2</a>

<a href="https://example.com" rel="nofollow">Liên kết 3</a>

**CSS**:

css

Sao chépChỉnh sửa

a[rel|="noopener"] {

color: red;

}

**Giải thích**:

* Bộ chọn [rel|="noopener"] sẽ chọn tất cả các liên kết (<a>) có thuộc tính rel bắt đầu bằng noopener.
* **Kết quả**:
  + **Liên kết 1** và **Liên kết 2** sẽ có **màu chữ đỏ** vì thuộc tính rel của chúng bắt đầu với "noopener".
  + **Liên kết 3** sẽ không thay đổi vì rel="nofollow" không bắt đầu với "noopener".

**Ví dụ 3: Chọn các phần tử có thuộc tính data-type bắt đầu với "user-"**

**HTML**:

html

Sao chépChỉnh sửa

<div data-type="user-admin">Admin</div>

<div data-type="user-member">Member</div>

<div data-type="guest">Guest</div>

**CSS**:

css

Sao chépChỉnh sửa

div[data-type|="user"] {

font-weight: bold;

}

**Giải thích**:

* Bộ chọn [data-type|="user"] sẽ chọn tất cả các phần tử <div> có thuộc tính data-type bắt đầu với user-, như user-admin và user-member.
* **Kết quả**:
  + **Phần tử đầu tiên và thứ hai** sẽ có **chữ in đậm** vì thuộc tính data-type của chúng bắt đầu với "user".
  + **Phần tử thứ ba** sẽ không thay đổi vì thuộc tính data-type không bắt đầu với "user".

**🎯 Lưu ý khi sử dụng [attribute|="value"]**

1. **Sử dụng với các thuộc tính có giá trị phân tách bằng dấu gạch nối**:
   * Bộ chọn này đặc biệt hữu ích cho các thuộc tính như lang, hreflang, rel, type có giá trị phân tách bằng dấu gạch nối. Nó giúp bạn chọn các phần tử có thuộc tính bắt đầu với một từ khóa xác định và có thể có các phần mở rộng.
2. **Không phân biệt chữ hoa chữ thường**:
   * Bộ chọn thuộc tính trong CSS không phân biệt chữ hoa chữ thường. Điều này có nghĩa là lang="en-us" và lang="EN-US" đều sẽ được coi là giống nhau.
3. **Sự khác biệt giữa [attr|="value"] và [attr="value"]**:
   * Bộ chọn [attr|="value"] không chỉ chọn các phần tử có giá trị đúng bằng value, mà còn chọn các phần tử có giá trị bắt đầu bằng value và theo sau có một dấu gạch nối (value-xyz).
   * Bộ chọn [attr="value"] chỉ chọn các phần tử có giá trị chính xác là value.
4. **Ứng dụng chủ yếu với các thuộc tính ngữ nghĩa (semantic attributes)**:
   * Bộ chọn này thường được sử dụng với các thuộc tính như lang, hreflang, rel để xác định các nhóm hoặc thể loại của các phần tử. Ví dụ, bạn có thể chọn tất cả các phần tử có lang bắt đầu bằng "en-" để áp dụng kiểu dáng cho tất cả các ngôn ngữ tiếng Anh.

**🎯 Tóm tắt về bộ chọn [attribute|="value"]**

| **Tính năng** | **Mô tả** |
| --- | --- |
| **Cú pháp** | `[attribute |
| **Chọn phần tử** | Chọn phần tử có thuộc tính attribute có giá trị bắt đầu bằng value và có thể có thêm một phần mở rộng phân tách bằng dấu gạch nối. |
| **Ứng dụng phổ biến** | Các thuộc tính như lang, hreflang, rel, v.v. |
| **Lưu ý** | Bộ chọn này không phân biệt chữ hoa chữ thường và chỉ hoạt động với các thuộc tính có giá trị phân tách bằng dấu gạch nối. |

**Ví dụ tổng hợp:**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ về Bộ chọn thuộc tính [attribute|="value"]</title>

<style>

/\* Chọn tất cả các phần tử có lang bắt đầu bằng "en-" \*/

html[lang|="en"] {

background-color: yellow;

}

/\* Chọn tất cả các liên kết có rel bắt đầu bằng "noopener" \*/

a[rel|="noopener"] {

color: red;

}

/\* Chọn tất cả các phần tử có data-type bắt đầu bằng "user-" \*/

div[data-type|="user"] {

font-weight: bold;

}

</style>

</head>

<body>

<html lang="en-US">

<h1>Chào bạn!</h1>

</html>

<html lang="en-GB">

<h1>Welcome!</h1>

</html>

<html lang="fr-FR">

<h1>Bienvenue!</h1>

</html>

<a href="https://example.com" rel="noopener noreferrer">Liên kết 1</a><br>

<a href="https://example.com" rel="noopener">Liên kết 2</a><br>

<a href="https://example.com" rel="nofollow">Liên kết 3</a><br>

<div data-type="user-admin">Admin</div>

<div data-type="user-member">Member</div>

<div data-type="guest">Guest</div>

</body>

</html>

Hy vọng bài viết giúp bạn hiểu rõ về **bộ chọn thuộc tính [attribute|="value"]** trong CSS. Nếu bạn có thêm câu hỏi hoặc muốn tìm hiểu thêm về bất kỳ phần nào, đừng ngần ngại yêu cầu nhé! 😊

Bạn hãy cho mình ví dụ minh họa bao quát về Bộ chọn thuộc tính: [attribute|="value"]

**ChatGPT đã nói:**

Dưới đây là ví dụ minh họa bao quát về **Bộ chọn thuộc tính: [attribute|="value"]** trong CSS. Ví dụ này sẽ giúp bạn hiểu cách sử dụng bộ chọn thuộc tính này để áp dụng các kiểu dáng cho các phần tử dựa trên một giá trị bắt đầu với một từ khóa xác định và có thể có dấu gạch nối (hyphen) sau từ khóa đó.

**HTML:**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ Bộ chọn thuộc tính [attribute|="value"]</title>

<style>

/\* Chọn tất cả các phần tử có lang bắt đầu bằng "en-" \*/

html[lang|="en"] {

background-color: yellow;

}

/\* Chọn tất cả các liên kết có rel bắt đầu bằng "noopener" \*/

a[rel|="noopener"] {

color: red;

}

/\* Chọn tất cả các phần tử có data-type bắt đầu bằng "user-" \*/

div[data-type|="user"] {

font-weight: bold;

}

</style>

</head>

<body>

<!-- Các phần tử <html> với lang bắt đầu bằng "en-" -->

<html lang="en-US">

<h1>Chào bạn!</h1>

</html>

<html lang="en-GB">

<h1>Welcome!</h1>

</html>

<html lang="fr-FR">

<h1>Bienvenue!</h1>

</html>

<!-- Các liên kết với rel bắt đầu bằng "noopener" -->

<a href="https://example.com" rel="noopener noreferrer">Liên kết 1 - rel="noopener noreferrer"</a><br>

<a href="https://example.com" rel="noopener">Liên kết 2 - rel="noopener"</a><br>

<a href="https://example.com" rel="nofollow">Liên kết 3 - rel="nofollow"</a><br>

<!-- Các phần tử <div> với data-type bắt đầu bằng "user-" -->

<div data-type="user-admin">Admin</div>

<div data-type="user-member">Member</div>

<div data-type="guest">Guest</div>

</body>

</html>

**Giải thích chi tiết:**

1. **Bộ chọn html[lang|="en"]**:

css

Sao chépChỉnh sửa

html[lang|="en"] {

background-color: yellow;

}

* + **Mục đích**: Bộ chọn này sẽ chọn tất cả các phần tử <html> có thuộc tính lang bắt đầu với "en", ví dụ như en-US, en-GB, v.v.
  + **Kết quả**:
    - Các phần tử <html> có lang="en-US" và lang="en-GB" sẽ có nền **vàng**.
    - Phần tử <html> có lang="fr-FR" sẽ không thay đổi vì giá trị không bắt đầu với "en".

1. **Bộ chọn a[rel|="noopener"]**:

css

Sao chépChỉnh sửa

a[rel|="noopener"] {

color: red;

}

* + **Mục đích**: Bộ chọn này sẽ chọn tất cả các liên kết <a> có thuộc tính rel bắt đầu với "noopener", chẳng hạn như noopener hoặc noopener noreferrer.
  + **Kết quả**:
    - **Liên kết 1** (rel="noopener noreferrer") và **Liên kết 2** (rel="noopener") sẽ có **màu chữ đỏ**.
    - **Liên kết 3** (rel="nofollow") sẽ không thay đổi vì giá trị rel của nó không bắt đầu với "noopener".

1. **Bộ chọn div[data-type|="user"]**:

css

Sao chépChỉnh sửa

div[data-type|="user"] {

font-weight: bold;

}

* + **Mục đích**: Bộ chọn này sẽ chọn tất cả các phần tử <div> có thuộc tính data-type bắt đầu với "user-", ví dụ như user-admin, user-member.
  + **Kết quả**:
    - **Div đầu tiên** (data-type="user-admin") và **Div thứ hai** (data-type="user-member") sẽ có **chữ in đậm**.
    - **Div thứ ba** (data-type="guest") sẽ không thay đổi vì giá trị data-type không bắt đầu với "user-".

**Kết quả trên trình duyệt:**

* **Phần tử <html> có lang="en-US" và lang="en-GB"** sẽ có nền **vàng**.
* **Liên kết 1 và 2** sẽ có **màu chữ đỏ**.
* **Div 1 và Div 2** sẽ có **chữ in đậm**.
* **Các phần tử không khớp với điều kiện** (như phần tử có lang="fr-FR", rel="nofollow", và data-type="guest") sẽ không bị thay đổi gì.

**Tóm tắt về Bộ chọn thuộc tính [attribute|="value"]:**

* Bộ chọn [attribute|="value"] tìm các phần tử có thuộc tính có giá trị bắt đầu bằng value và có thể có dấu gạch nối - tiếp theo, ví dụ như value-abc.
* Bộ chọn này thường được dùng cho các thuộc tính như lang, hreflang, rel, type để chọn các phần tử có thuộc tính với giá trị xác định, bao gồm các từ khóa có dấu gạch nối phân tách.

Hy vọng ví dụ này giúp bạn dễ dàng hiểu và sử dụng bộ chọn thuộc tính [attribute|="value"] trong CSS! Nếu bạn có thêm câu hỏi hoặc muốn thử thêm ví dụ khác, đừng ngần ngại yêu cầu nhé! 😊

## **5. BỘ CHỌN THUỘC TÍNH [ATTR^="VALUE"]**

Hãy giải thích chi tiết tất tần tật, sâu rộng toàn bộ và toàn diện kiến thức về "Bộ chọn thuộc tính: [attr^="value"]" và những lưu ý (nếu có) về nó để cho người non-tech hiểu sâu và rõ.

**ChatGPT đã nói:**

**Giới thiệu về Bộ chọn thuộc tính [attr^="value"] trong CSS**

Bộ chọn thuộc tính [attr^="value"] là một trong những bộ chọn nâng cao trong CSS, cho phép bạn chọn các phần tử dựa trên giá trị của thuộc tính bắt đầu bằng một chuỗi cụ thể. Cụ thể, bộ chọn này chỉ áp dụng cho các phần tử mà thuộc tính của chúng có giá trị bắt đầu bằng một chuỗi ký tự nhất định.

**Cú pháp và Cách thức hoạt động của [attr^="value"]**

* **Cú pháp**: [attr^="value"]
  + **attr** là tên của thuộc tính mà bạn muốn kiểm tra.
  + **value** là giá trị chuỗi mà bạn muốn phần tử có thuộc tính đó bắt đầu với.

**Cách thức hoạt động:**

* Bộ chọn này tìm các phần tử có thuộc tính attr có giá trị **bắt đầu với** chuỗi value mà bạn chỉ định. Chú ý rằng bộ chọn này chỉ tìm giá trị bắt đầu đúng với value mà không quan tâm đến những ký tự khác sau value.

**Ví dụ về cách sử dụng:**

Giả sử bạn có một trang web với các liên kết và bạn muốn thay đổi kiểu dáng của các liên kết bắt đầu bằng "https" trong thuộc tính href.

**Cú pháp CSS:**

css

Sao chépChỉnh sửa

a[href^="https"] {

color: green;

}

* Trong ví dụ này:
  + **Bộ chọn** a[href^="https"] sẽ tìm tất cả các phần tử <a> có thuộc tính href bắt đầu bằng https (chẳng hạn như "<https://example.com>").
  + Các liên kết này sẽ có **màu chữ xanh**.

**Ví dụ minh họa chi tiết về [attr^="value"]**

**HTML:**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ Bộ chọn thuộc tính [attr^="value"]</title>

<style>

/\* Bộ chọn này áp dụng cho tất cả các liên kết có href bắt đầu bằng "https" \*/

a[href^="https"] {

color: green;

}

</style>

</head>

<body>

<a href="https://www.example.com">Liên kết 1 (HTTPS)</a><br>

<a href="http://www.example.com">Liên kết 2 (HTTP)</a><br>

<a href="https://secure-site.com">Liên kết 3 (HTTPS)</a><br>

<a href="ftp://example.com">Liên kết 4 (FTP)</a><br>

</body>

</html>

**Giải thích:**

* **Bộ chọn CSS** a[href^="https"] sẽ tìm tất cả các phần tử <a> mà thuộc tính href của chúng bắt đầu với "https".
* **Kết quả**:
  + **Liên kết 1** (href="https://www.example.com") và **Liên kết 3** (href="https://secure-site.com") sẽ có **màu chữ xanh** vì chúng bắt đầu với "https".
  + **Liên kết 2** (href="http://www.example.com") và **Liên kết 4** (href="ftp://example.com") sẽ không thay đổi vì thuộc tính href của chúng không bắt đầu với "https".

**Các Lưu ý khi sử dụng [attr^="value"]**

1. **Chú ý về độ chính xác**:
   * Bộ chọn [attr^="value"] sẽ chỉ tìm kiếm các phần tử có thuộc tính attr có giá trị bắt đầu **hoàn toàn** với value. Nếu giá trị thuộc tính bắt đầu với một chuỗi khác hoặc không bắt đầu chính xác với value, phần tử đó sẽ không được chọn.
2. **Không phân biệt chữ hoa chữ thường**:
   * Bộ chọn thuộc tính trong CSS **không phân biệt chữ hoa chữ thường**. Điều này có nghĩa là href^="https" sẽ chọn cả https:// và HTTPS://.
3. **Sử dụng với các thuộc tính phổ biến**:
   * Bộ chọn [attr^="value"] thường được sử dụng với các thuộc tính như href, src, class, id, v.v., khi bạn cần áp dụng kiểu dáng cho các phần tử có thuộc tính bắt đầu bằng một chuỗi cụ thể.
4. **Chỉ áp dụng với phần tử có thuộc tính**:
   * Bộ chọn này chỉ áp dụng cho các phần tử có thuộc tính được chỉ định. Nếu phần tử không có thuộc tính đó, CSS sẽ không áp dụng kiểu dáng cho phần tử đó.

**Tóm tắt về Bộ chọn thuộc tính [attr^="value"]**

| **Tính năng** | **Mô tả** |
| --- | --- |
| **Cú pháp** | [attr^="value"] |
| **Chọn phần tử** | Chọn phần tử có thuộc tính attr với giá trị bắt đầu bằng chuỗi value. |
| **Ứng dụng phổ biến** | Các thuộc tính như href, src, class, v.v. |
| **Lưu ý** | Bộ chọn này chỉ chọn các phần tử có giá trị thuộc tính bắt đầu với value. Không phân biệt chữ hoa chữ thường. |

**Ví dụ tổng hợp về [attr^="value"]**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ Bộ chọn thuộc tính [attr^="value"]</title>

<style>

/\* Chọn các liên kết có href bắt đầu bằng "https" \*/

a[href^="https"] {

color: green;

}

/\* Chọn các hình ảnh có src bắt đầu bằng "img/" \*/

img[src^="img/"] {

border: 3px solid red;

}

</style>

</head>

<body>

<a href="https://example.com">Liên kết HTTPS 1</a><br>

<a href="http://example.com">Liên kết HTTP</a><br>

<a href="https://secure.com">Liên kết HTTPS 2</a><br>

<img src="img/photo1.jpg" alt="Ảnh 1">

<img src="img/photo2.jpg" alt="Ảnh 2">

<img src="video/video1.mp4" alt="Ảnh 3">

</body>

</html>

**Kết quả**:

* **Liên kết 1** và **Liên kết 3** có màu chữ **xanh** vì thuộc tính href bắt đầu bằng "https".
* **Các hình ảnh** với src bắt đầu bằng "img/" (hình ảnh 1 và 2) sẽ có **viền đỏ**.
* **Hình ảnh thứ ba** không có viền vì src="video/video1.mp4" không bắt đầu bằng "img/".

Hy vọng qua ví dụ và giải thích trên, bạn đã hiểu rõ cách sử dụng **Bộ chọn thuộc tính [attr^="value"]** trong CSS. Nếu bạn có thêm câu hỏi hoặc cần làm rõ bất kỳ phần nào, đừng ngần ngại yêu cầu nhé! 😊

**Bạn đã nói:**

Bạn hãy cho mình ví dụ minh họa bao quát về Bộ chọn thuộc tính [attr^="value"]

**ChatGPT đã nói:**

Dưới đây là một ví dụ minh họa bao quát về **Bộ chọn thuộc tính [attr^="value"]** trong CSS, giúp bạn hiểu rõ cách sử dụng bộ chọn này trong thực tế.

**HTML:**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ Bộ chọn thuộc tính [attr^="value"]</title>

<style>

/\* Bộ chọn này áp dụng cho tất cả các liên kết có href bắt đầu bằng "https" \*/

a[href^="https"] {

color: green;

}

/\* Bộ chọn này áp dụng cho tất cả các hình ảnh có src bắt đầu bằng "img/" \*/

img[src^="img/"] {

border: 3px solid red;

}

/\* Bộ chọn này áp dụng cho tất cả các liên kết có class bắt đầu bằng "btn-" \*/

a[class^="btn-"] {

background-color: yellow;

padding: 10px;

text-decoration: none;

}

</style>

</head>

<body>

<h1>Ví dụ về Bộ chọn thuộc tính [attr^="value"]</h1>

<!-- Các liên kết -->

<a href="https://example.com">Liên kết 1 (https)</a><br>

<a href="http://example.com">Liên kết 2 (http)</a><br>

<a href="https://secure-site.com">Liên kết 3 (https)</a><br>

<!-- Các hình ảnh -->

<img src="img/photo1.jpg" alt="Ảnh 1">

<img src="img/photo2.jpg" alt="Ảnh 2">

<img src="video/video1.mp4" alt="Ảnh 3">

<!-- Các liên kết có class bắt đầu bằng "btn-" -->

<a class="btn-primary" href="#">Button Primary</a><br>

<a class="btn-secondary" href="#">Button Secondary</a><br>

<a class="other-class" href="#">Other Button</a>

</body>

</html>

**Giải thích:**

**1. Bộ chọn a[href^="https"]:**

css

Sao chépChỉnh sửa

a[href^="https"] {

color: green;

}

* **Mục đích**: Bộ chọn này sẽ chọn tất cả các phần tử <a> có thuộc tính href bắt đầu với "https".
* **Kết quả**:
  + **Liên kết 1** (href="https://example.com") và **Liên kết 3** (href="https://secure-site.com") sẽ có **màu chữ xanh**.
  + **Liên kết 2** (href="http://example.com") sẽ không thay đổi vì href của nó không bắt đầu với "https".

**2. Bộ chọn img[src^="img/"]:**

css

Sao chépChỉnh sửa

img[src^="img/"] {

border: 3px solid red;

}

* **Mục đích**: Bộ chọn này sẽ chọn tất cả các phần tử <img> có thuộc tính src bắt đầu với "img/".
* **Kết quả**:
  + **Ảnh 1** (src="img/photo1.jpg") và **Ảnh 2** (src="img/photo2.jpg") sẽ có **viền đỏ**.
  + **Ảnh 3** (src="video/video1.mp4") sẽ không có viền đỏ vì src của nó không bắt đầu với "img/".

**3. Bộ chọn a[class^="btn-"]:**

css

Sao chépChỉnh sửa

a[class^="btn-"] {

background-color: yellow;

padding: 10px;

text-decoration: none;

}

* **Mục đích**: Bộ chọn này sẽ chọn tất cả các phần tử <a> có thuộc tính class bắt đầu với "btn-".
* **Kết quả**:
  + **Button Primary** (class="btn-primary") và **Button Secondary** (class="btn-secondary") sẽ có **nền màu vàng**, **padding 10px** và không có gạch chân (do text-decoration: none).
  + **Button Other** (class="other-class") sẽ không thay đổi vì class của nó không bắt đầu với "btn-".

**Kết quả hiển thị trên trình duyệt:**

1. **Liên kết 1** và **Liên kết 3** có **màu chữ xanh**.
2. **Ảnh 1** và **Ảnh 2** có **viền đỏ**.
3. **Button Primary** và **Button Secondary** có **nền màu vàng** và không có gạch chân.

**Tóm tắt:**

* **Bộ chọn thuộc tính [attr^="value"]** là một bộ chọn mạnh mẽ trong CSS, cho phép bạn áp dụng kiểu dáng cho các phần tử có thuộc tính bắt đầu với một giá trị xác định.
* Các ứng dụng phổ biến của nó bao gồm việc chọn các liên kết (link) với URL bắt đầu bằng "http" hoặc "https", chọn các hình ảnh với đường dẫn bắt đầu bằng "img/", hoặc chọn các phần tử có class bắt đầu bằng một chuỗi cụ thể.

Hy vọng qua ví dụ trên, bạn đã hiểu rõ hơn về cách sử dụng bộ chọn thuộc tính [attr^="value"] trong CSS! Nếu bạn có thêm câu hỏi, đừng ngần ngại yêu cầu nhé! 😊

## **6. BỘ CHỌN THUỘC TÍNH [ATTR$="VALUE"]**

Hãy giải thích chi tiết tất tần tật, sâu rộng toàn bộ và toàn diện kiến thức về "Bộ chọn thuộc tính: [attr$="value"]" và những lưu ý (nếu có) về nó để cho người non-tech hiểu sâu và rõ.

**ChatGPT đã nói:**

**Giới thiệu về Bộ chọn thuộc tính [attr$="value"] trong CSS**

Bộ chọn thuộc tính [attr$="value"] là một công cụ mạnh mẽ trong CSS cho phép bạn chọn các phần tử dựa trên giá trị của thuộc tính kết thúc bằng một chuỗi cụ thể. Với bộ chọn này, bạn có thể áp dụng kiểu dáng cho các phần tử mà thuộc tính của chúng có giá trị kết thúc bằng một chuỗi bạn chỉ định.

**Cú pháp và cách thức hoạt động của [attr$="value"]**

* **Cú pháp**: [attr$="value"]
  + **attr**: Là tên thuộc tính mà bạn muốn kiểm tra.
  + **value**: Là giá trị chuỗi mà bạn muốn phần tử có thuộc tính attr của nó kết thúc bằng.

**Cách thức hoạt động:**

Bộ chọn [attr$="value"] chọn các phần tử có thuộc tính attr mà giá trị của thuộc tính đó **kết thúc bằng** chuỗi value. Điều này có nghĩa là nếu thuộc tính có giá trị là một chuỗi dài, bạn chỉ cần kiểm tra xem chuỗi đó có kết thúc bằng value hay không.

**Ví dụ minh họa về cách sử dụng [attr$="value"]**

Giả sử bạn có một trang web với các liên kết và bạn muốn thay đổi kiểu dáng cho các liên kết có đuôi URL là .pdf.

**HTML:**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ Bộ chọn thuộc tính [attr$="value"]</title>

<style>

/\* Bộ chọn này áp dụng cho tất cả các liên kết có href kết thúc bằng ".pdf" \*/

a[href$=".pdf"] {

color: red;

font-weight: bold;

}

</style>

</head>

<body>

<a href="document1.pdf">Tài liệu 1 (PDF)</a><br>

<a href="image1.jpg">Hình ảnh 1 (JPG)</a><br>

<a href="document2.pdf">Tài liệu 2 (PDF)</a><br>

<a href="audio1.mp3">Âm thanh 1 (MP3)</a><br>

</body>

</html>

**Giải thích về ví dụ:**

* Bộ chọn a[href$=".pdf"] sẽ chọn tất cả các phần tử <a> có thuộc tính href **kết thúc với** chuỗi .pdf.
* **Kết quả**:
  + Các liên kết **Tài liệu 1** và **Tài liệu 2** sẽ có **màu chữ đỏ** và **chữ đậm** vì href của chúng kết thúc bằng .pdf.
  + Các liên kết khác như **Hình ảnh 1** và **Âm thanh 1** không bị ảnh hưởng vì href của chúng không kết thúc bằng .pdf.

**Ứng dụng của [attr$="value"] trong thực tế**

1. **Chọn các liên kết tải file**:
   * Nếu bạn muốn thay đổi kiểu dáng của các liên kết tải file, bạn có thể sử dụng bộ chọn này để chọn các liên kết có đuôi .pdf, .docx, .zip, v.v.

Ví dụ:

css

Sao chépChỉnh sửa

a[href$=".zip"] {

background-color: yellow;

}

1. **Chọn các hình ảnh với định dạng cụ thể**:
   * Bạn có thể chọn các hình ảnh có đuôi .jpg, .png, .gif để áp dụng kiểu dáng riêng biệt.

Ví dụ:

css

Sao chépChỉnh sửa

img[src$=".jpg"] {

border: 2px solid blue;

}

1. **Chọn các tài liệu hoặc bài viết**:
   * Giả sử bạn muốn áp dụng kiểu dáng cho tất cả các liên kết đến các tài liệu PDF hoặc các bài viết có định dạng .html:

css

Sao chépChỉnh sửa

a[href$=".html"] {

text-decoration: underline;

}

**Các lưu ý khi sử dụng [attr$="value"]**

1. **Phân biệt chữ hoa chữ thường**:
   * Bộ chọn thuộc tính CSS **không phân biệt chữ hoa chữ thường**. Điều này có nghĩa là a[href$=".pdf"] sẽ chọn cả .pdf và .PDF.
2. **Chỉ áp dụng cho thuộc tính có giá trị chuỗi**:
   * Bộ chọn này chỉ có thể áp dụng cho các thuộc tính có giá trị chuỗi (string), chẳng hạn như href, src, alt, title, v.v. Nó không thể áp dụng cho các thuộc tính như checked (có giá trị boolean).
3. **Giới hạn trong việc chọn các phần tử**:
   * Bộ chọn này chỉ áp dụng cho các phần tử có thuộc tính với giá trị phù hợp. Nếu phần tử không có thuộc tính hoặc giá trị không phù hợp, CSS sẽ không được áp dụng.
4. **Không hỗ trợ các phần tử không có thuộc tính**:
   * Chỉ các phần tử có thuộc tính được chọn. Ví dụ, <p> không có thuộc tính href, nên không thể áp dụng bộ chọn [href$="value"] cho các phần tử <p>.

**Tóm tắt về Bộ chọn thuộc tính [attr$="value"]**

| **Tính năng** | **Mô tả** |
| --- | --- |
| **Cú pháp** | [attr$="value"] |
| **Mô tả** | Chọn các phần tử có thuộc tính attr mà giá trị của nó kết thúc với chuỗi value. |
| **Ứng dụng phổ biến** | Dùng cho các thuộc tính như href, src, class, v.v., khi bạn cần chọn các phần tử có giá trị thuộc tính kết thúc bằng chuỗi cụ thể. |
| **Lưu ý** | Bộ chọn không phân biệt chữ hoa chữ thường và chỉ áp dụng cho các thuộc tính có giá trị chuỗi. |

**Ví dụ tổng hợp về [attr$="value"]**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ Bộ chọn thuộc tính [attr$="value"]</title>

<style>

/\* Chọn tất cả các liên kết có href kết thúc bằng ".pdf" \*/

a[href$=".pdf"] {

color: red;

font-weight: bold;

}

/\* Chọn tất cả các hình ảnh có src kết thúc bằng ".jpg" \*/

img[src$=".jpg"] {

border: 2px solid blue;

}

/\* Chọn tất cả các liên kết có href kết thúc bằng ".html" \*/

a[href$=".html"] {

text-decoration: underline;

}

</style>

</head>

<body>

<a href="document1.pdf">Tài liệu 1 (PDF)</a><br>

<a href="image1.jpg">Hình ảnh 1 (JPG)</a><br>

<a href="about.html">Về chúng tôi (HTML)</a><br>

<a href="document2.pdf">Tài liệu 2 (PDF)</a><br>

<img src="img/photo1.jpg" alt="Ảnh 1">

<img src="img/photo2.jpg" alt="Ảnh 2">

<img src="video/video1.mp4" alt="Ảnh 3">

</body>

</html>

**Kết quả:**

* **Tài liệu 1 (PDF)** và **Tài liệu 2 (PDF)** sẽ có **màu chữ đỏ** và **chữ đậm**.
* **Hình ảnh 1** và **Hình ảnh 2** sẽ có **viền xanh** vì src của chúng kết thúc với .jpg.
* **Liên kết "Về chúng tôi (HTML)"** sẽ có **gạch chân** vì href của nó kết thúc với .html.

**Kết luận:**

Bộ chọn thuộc tính [attr$="value"] là một công cụ rất hữu ích trong CSS để áp dụng kiểu dáng cho các phần tử dựa trên giá trị kết thúc của thuộc tính. Khi bạn cần chọn các phần tử có thuộc tính kết thúc bằng một chuỗi cụ thể, bộ chọn này giúp bạn thực hiện điều đó một cách dễ dàng.

**Bạn đã nói:**

Bạn hãy cho mình ví dụ minh họa bao quát về Bộ chọn thuộc tính [attr$="value"]

**ChatGPT đã nói:**

Dưới đây là một ví dụ minh họa bao quát về **Bộ chọn thuộc tính [attr$="value"]** trong CSS, giúp bạn hiểu rõ cách sử dụng bộ chọn này trong thực tế.

**HTML:**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ Bộ chọn thuộc tính [attr$="value"]</title>

<style>

/\* Bộ chọn này sẽ áp dụng cho tất cả các liên kết có href kết thúc bằng ".pdf" \*/

a[href$=".pdf"] {

color: red;

font-weight: bold;

}

/\* Bộ chọn này sẽ áp dụng cho tất cả các hình ảnh có src kết thúc bằng ".jpg" \*/

img[src$=".jpg"] {

border: 2px solid blue;

}

/\* Bộ chọn này sẽ áp dụng cho tất cả các liên kết có href kết thúc bằng ".html" \*/

a[href$=".html"] {

text-decoration: underline;

}

</style>

</head>

<body>

<h1>Ví dụ về Bộ chọn thuộc tính [attr$="value"]</h1>

<!-- Các liên kết -->

<a href="document1.pdf">Tài liệu 1 (PDF)</a><br>

<a href="image1.jpg">Hình ảnh 1 (JPG)</a><br>

<a href="about.html">Về chúng tôi (HTML)</a><br>

<a href="document2.pdf">Tài liệu 2 (PDF)</a><br>

<!-- Các hình ảnh -->

<img src="img/photo1.jpg" alt="Ảnh 1">

<img src="img/photo2.jpg" alt="Ảnh 2">

<img src="video/video1.mp4" alt="Ảnh 3">

</body>

</html>

**Giải thích về ví dụ:**

* Bộ chọn a[href$=".pdf"] sẽ chọn tất cả các phần tử <a> có thuộc tính href **kết thúc với** chuỗi .pdf.
* **Kết quả**:
  + Các liên kết **Tài liệu 1 (PDF)** và **Tài liệu 2 (PDF)** sẽ có **màu chữ đỏ** và **chữ đậm** vì href của chúng kết thúc bằng .pdf.
  + Các liên kết khác như **Hình ảnh 1 (JPG)** và **Về chúng tôi (HTML)** sẽ không bị ảnh hưởng vì href của chúng không kết thúc bằng .pdf.
* Bộ chọn img[src$=".jpg"] sẽ chọn tất cả các phần tử <img> có thuộc tính src **kết thúc với** chuỗi .jpg.
* **Kết quả**:
  + Các hình ảnh **Ảnh 1** và **Ảnh 2** sẽ có **viền xanh** vì src của chúng kết thúc bằng .jpg.
  + **Ảnh 3** không có viền xanh vì src của nó kết thúc bằng .mp4 thay vì .jpg.
* Bộ chọn a[href$=".html"] sẽ chọn tất cả các phần tử <a> có thuộc tính href **kết thúc với** chuỗi .html.
* **Kết quả**:
  + Liên kết **Về chúng tôi (HTML)** sẽ có **gạch chân** vì href của nó kết thúc bằng .html.

**Kết quả hiển thị trên trình duyệt:**

* **Tài liệu 1 (PDF)** và **Tài liệu 2 (PDF)** sẽ có **màu chữ đỏ** và **chữ đậm**.
* **Hình ảnh 1** và **Hình ảnh 2** sẽ có **viền xanh**.
* **Liên kết "Về chúng tôi (HTML)"** sẽ có **gạch chân**.

**Tóm tắt:**

* **Bộ chọn thuộc tính [attr$="value"]** chọn tất cả các phần tử có thuộc tính attr mà giá trị của thuộc tính đó kết thúc với chuỗi value.
* Bạn có thể sử dụng bộ chọn này để áp dụng kiểu dáng cho các phần tử có giá trị thuộc tính phù hợp (ví dụ như các liên kết .pdf, các hình ảnh .jpg, v.v.).

Hy vọng qua ví dụ này, bạn đã có cái nhìn rõ ràng về cách sử dụng bộ chọn thuộc tính [attr$="value"].

## **7. BỘ CHỌN THUỘC TÍNH: [ATTR\*="VALUE"]**

Hãy giải thích chi tiết tất tần tật, sâu rộng toàn bộ và toàn diện kiến thức về "Bộ chọn thuộc tính: [attr\*="value"]" và những lưu ý (nếu có) về nó để cho người non-tech hiểu sâu và rõ.

**ChatGPT đã nói:**

**Bộ chọn thuộc tính: [attr\*="value"] trong CSS**

Bộ chọn thuộc tính [attr\*="value"] là một công cụ mạnh mẽ trong CSS, giúp bạn chọn các phần tử HTML dựa trên giá trị của một thuộc tính, trong đó giá trị của thuộc tính chứa một chuỗi cụ thể. Bộ chọn này rất hữu ích khi bạn muốn áp dụng kiểu dáng cho các phần tử có thuộc tính có chứa chuỗi bạn đã chỉ định ở bất kỳ vị trí nào trong giá trị của thuộc tính đó.

**Cú pháp và cách thức hoạt động của [attr\*="value"]**

* **Cú pháp**: [attr\*="value"]
  + **attr**: Là tên của thuộc tính mà bạn muốn kiểm tra (ví dụ: href, src, class, v.v.).
  + **value**: Là chuỗi mà bạn muốn tìm kiếm trong giá trị của thuộc tính.

**Cách thức hoạt động:**

Bộ chọn [attr\*="value"] chọn tất cả các phần tử có thuộc tính attr mà **giá trị của nó chứa** chuỗi value ở bất kỳ vị trí nào trong giá trị của thuộc tính đó. Điều này có nghĩa là nếu giá trị của thuộc tính chứa chuỗi bạn chỉ định, phần tử đó sẽ được chọn.

**Ví dụ minh họa về cách sử dụng [attr\*="value"]**

Giả sử bạn có một trang web và bạn muốn thay đổi kiểu dáng cho các liên kết chứa từ khóa .pdf trong thuộc tính href. Bộ chọn [attr\*="value"] sẽ giúp bạn làm điều này.

**HTML:**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ Bộ chọn thuộc tính [attr\*="value"]</title>

<style>

/\* Bộ chọn này sẽ áp dụng cho tất cả các liên kết có href chứa chuỗi "pdf" \*/

a[href\*="pdf"] {

color: red;

font-weight: bold;

}

/\* Bộ chọn này sẽ áp dụng cho tất cả các hình ảnh có src chứa chuỗi "photo" \*/

img[src\*="photo"] {

border: 2px solid blue;

}

</style>

</head>

<body>

<h1>Ví dụ về Bộ chọn thuộc tính [attr\*="value"]</h1>

<!-- Các liên kết -->

<a href="document1.pdf">Tài liệu 1 (PDF)</a><br>

<a href="image1.jpg">Hình ảnh 1 (JPG)</a><br>

<a href="about\_us.html">Về chúng tôi (HTML)</a><br>

<a href="document2\_pdf">Tài liệu 2 (PDF không chính thức)</a><br>

<!-- Các hình ảnh -->

<img src="img/photo1.jpg" alt="Ảnh 1">

<img src="img/photo2.png" alt="Ảnh 2">

<img src="video/video1.mp4" alt="Ảnh 3">

</body>

</html>

**Giải thích về ví dụ:**

* Bộ chọn a[href\*="pdf"] sẽ chọn tất cả các phần tử <a> có thuộc tính href chứa chuỗi **"pdf"** ở bất kỳ vị trí nào trong giá trị của thuộc tính href.
  + Kết quả: Liên kết **Tài liệu 1 (PDF)** và **Tài liệu 2 (PDF không chính thức)** sẽ có **màu chữ đỏ** và **chữ đậm** vì thuộc tính href của chúng chứa chuỗi "pdf".
  + Liên kết **Hình ảnh 1 (JPG)** và **Về chúng tôi (HTML)** không bị ảnh hưởng vì giá trị href của chúng không chứa "pdf".
* Bộ chọn img[src\*="photo"] sẽ chọn tất cả các phần tử <img> có thuộc tính src chứa chuỗi **"photo"** ở bất kỳ vị trí nào trong giá trị của thuộc tính src.
  + Kết quả: Các hình ảnh **Ảnh 1** và **Ảnh 2** sẽ có **viền xanh** vì giá trị src của chúng chứa chuỗi "photo".
  + **Ảnh 3** không có viền xanh vì src của nó chứa "video" thay vì "photo".

**Các lưu ý khi sử dụng [attr\*="value"]**

1. **Phân biệt chữ hoa và chữ thường**:
   * Bộ chọn thuộc tính CSS **không phân biệt chữ hoa chữ thường**. Điều này có nghĩa là a[href\*="pdf"] sẽ chọn cả các thuộc tính href chứa "PDF", "pdf", hay "Pdf". Tất cả sẽ được coi là tương đương.
2. **Áp dụng cho tất cả các thuộc tính có giá trị chuỗi**:
   * Bộ chọn này chỉ có thể áp dụng cho các thuộc tính có giá trị chuỗi (string), chẳng hạn như href, src, class, title, v.v. Nó không thể áp dụng cho các thuộc tính không phải chuỗi như checked, disabled (boolean).
3. **Không hỗ trợ các phần tử không có thuộc tính**:
   * Bộ chọn này chỉ áp dụng cho các phần tử có thuộc tính chứa chuỗi bạn tìm kiếm. Ví dụ, nếu một phần tử <p> không có thuộc tính href, bạn không thể áp dụng bộ chọn a[href\*="value"] cho nó.
4. **Chọn bất kỳ phần tử nào chứa chuỗi**:
   * Điều này có nghĩa là bộ chọn sẽ không chỉ tìm kiếm chuỗi "value" ở cuối hay đầu của thuộc tính, mà nó sẽ tìm kiếm ở bất kỳ vị trí nào trong giá trị của thuộc tính. Ví dụ, a[href\*="pdf"] sẽ chọn tất cả các liên kết có chứa "pdf" ở bất kỳ đâu trong URL, bao gồm các chuỗi như "document.pdf", "somefile\_pdf" hoặc "mypdfdocument".
5. **Khả năng chọn các giá trị không chính xác**:
   * Lưu ý rằng vì bộ chọn này chỉ tìm kiếm sự xuất hiện của chuỗi con, nó có thể chọn những phần tử không mong muốn nếu chúng chứa chuỗi con đó. Ví dụ, a[href\*="pdf"] sẽ chọn các liên kết có chứa "pdf" ở bất kỳ đâu, kể cả "document\_pdf", mà bạn có thể không muốn chọn.

**Ứng dụng của [attr\*="value"] trong thực tế**

1. **Chọn các liên kết tải tài liệu**:
   * Giả sử bạn muốn thay đổi kiểu dáng cho tất cả các liên kết đến tài liệu PDF, bạn có thể sử dụng bộ chọn này để chọn tất cả các liên kết chứa "pdf" trong URL.

css

Sao chépChỉnh sửa

a[href\*="pdf"] {

font-style: italic;

}

1. **Chọn các hình ảnh với tên đặc biệt**:
   * Nếu bạn muốn chọn tất cả các hình ảnh có tên chứa từ khóa "photo", bạn có thể sử dụng bộ chọn [src\*="photo"] để chọn các hình ảnh như photo1.jpg, photo2.png, v.v.

css

Sao chépChỉnh sửa

img[src\*="photo"] {

border: 3px solid green;

}

1. **Chọn các phần tử có chứa chuỗi cụ thể trong tên lớp (class)**:
   * Bạn có thể chọn các phần tử có tên lớp chứa một chuỗi cụ thể. Ví dụ, bạn có thể muốn chọn tất cả các phần tử có lớp chứa từ "highlight".

css

Sao chépChỉnh sửa

.highlight {

background-color: yellow;

}

**Tóm tắt về Bộ chọn thuộc tính [attr\*="value"]**

| **Tính năng** | **Mô tả** |
| --- | --- |
| **Cú pháp** | [attr\*="value"] |
| **Mô tả** | Chọn các phần tử có thuộc tính attr mà giá trị của nó **chứa** chuỗi value ở bất kỳ vị trí nào trong giá trị của thuộc tính đó. |
| **Ứng dụng phổ biến** | Dùng để chọn các phần tử có thuộc tính chứa chuỗi con cụ thể ở bất kỳ vị trí nào trong giá trị của thuộc tính đó. |
| **Lưu ý** | Bộ chọn không phân biệt chữ hoa chữ thường và chỉ áp dụng cho các thuộc tính có giá trị chuỗi. |

**Ví dụ tổng hợp về [attr\*="value"]**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ Bộ chọn thuộc tính [attr\*="value"]</title>

<style>

/\* Chọn tất cả các liên kết có href chứa "pdf" \*/

a[href\*="pdf"] {

color: red;

font-weight: bold;

}

/\* Chọn tất cả các hình ảnh có src chứa "photo" \*/

img[src\*="photo"] {

border: 2px solid blue;

}

</style>

</head>

<body>

<a href="document1.pdf">Tài liệu 1 (PDF)</a><br>

<a href="image1.jpg">Hình ảnh 1 (JPG)</a><br>

<a href="about\_us.html">Về chúng tôi (HTML)</a><br>

<a href="document2\_pdf">Tài liệu 2 (PDF không chính thức)</a><br>

<img src="img/photo1.jpg" alt="Ảnh 1">

<img src="img/photo2.png" alt="Ảnh 2">

<img src="video/video1.mp4" alt="Ảnh 3">

</body>

</html>

**Kết quả hiển thị trên trình duyệt:**

* **Tài liệu 1 (PDF)** và **Tài liệu 2 (PDF không chính thức)** sẽ có **màu chữ đỏ** và **chữ đậm** vì href của chúng chứa "pdf".
* Các hình ảnh **Ảnh 1** và **Ảnh 2** sẽ có **viền xanh** vì src của chúng chứa "photo".

**Bạn đã nói:**

Bạn hãy cho mình ví dụ minh họa bao quát về Bộ chọn thuộc tính [attr\*="value"]

**ChatGPT đã nói:**

Dưới đây là một ví dụ minh họa bao quát về **Bộ chọn thuộc tính [attr\*="value"]** trong CSS, giúp bạn hiểu rõ cách sử dụng bộ chọn này.

**HTML:**

html

Sao chépChỉnh sửa

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

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

<title>Ví dụ Bộ chọn thuộc tính [attr\*="value"]</title>

<style>

/\* Bộ chọn này sẽ áp dụng cho tất cả các liên kết có href chứa chuỗi "pdf" \*/

a[href\*="pdf"] {

color: red;

font-weight: bold;

}

/\* Bộ chọn này sẽ áp dụng cho tất cả các hình ảnh có src chứa chuỗi "photo" \*/

img[src\*="photo"] {

border: 2px solid blue;

}

/\* Bộ chọn này sẽ áp dụng cho tất cả các liên kết có href chứa chuỗi "html" \*/

a[href\*="html"] {

text-decoration: underline;

}

</style>

</head>

<body>

<h1>Ví dụ về Bộ chọn thuộc tính [attr\*="value"]</h1>

<!-- Các liên kết -->

<a href="document1.pdf">Tài liệu 1 (PDF)</a><br>

<a href="image1.jpg">Hình ảnh 1 (JPG)</a><br>

<a href="about\_us.html">Về chúng tôi (HTML)</a><br>

<a href="document2\_pdf">Tài liệu 2 (PDF không chính thức)</a><br>

<!-- Các hình ảnh -->

<img src="img/photo1.jpg" alt="Ảnh 1">

<img src="img/photo2.png" alt="Ảnh 2">

<img src="video/video1.mp4" alt="Ảnh 3">

</body>

</html>

**Giải thích về ví dụ:**

1. **Bộ chọn a[href\*="pdf"]**:
   * Bộ chọn này chọn tất cả các phần tử <a> có thuộc tính href chứa chuỗi "pdf".
   * **Kết quả**: Liên kết **Tài liệu 1 (PDF)** và **Tài liệu 2 (PDF không chính thức)** sẽ có **màu chữ đỏ** và **chữ đậm** vì href của chúng chứa chuỗi "pdf".
2. **Bộ chọn img[src\*="photo"]**:
   * Bộ chọn này chọn tất cả các phần tử <img> có thuộc tính src chứa chuỗi "photo".
   * **Kết quả**: Các hình ảnh **Ảnh 1** sẽ có **viền xanh** vì src của nó chứa chuỗi "photo".
   * **Ảnh 2** không bị ảnh hưởng vì src của nó chứa "photo" nhưng không có chuỗi con "photo".
3. **Bộ chọn a[href\*="html"]**:
   * Bộ chọn này chọn tất cả các phần tử <a> có thuộc tính href chứa chuỗi "html".
   * **Kết quả**: Liên kết **Về chúng tôi (HTML)** sẽ có **gạch chân** vì href của nó chứa chuỗi "html".

**Kết quả hiển thị trên trình duyệt:**

* Liên kết **Tài liệu 1 (PDF)** và **Tài liệu 2 (PDF không chính thức)** sẽ có **màu chữ đỏ** và **chữ đậm**.
* **Ảnh 1** sẽ có **viền xanh** vì src của nó chứa "photo".
* Liên kết **Về chúng tôi (HTML)** sẽ có **gạch chân** vì href của nó chứa "html".

**Tóm tắt:**

* **Bộ chọn thuộc tính [attr\*="value"]** chọn các phần tử có thuộc tính attr mà giá trị của thuộc tính đó **chứa** chuỗi value ở bất kỳ vị trí nào trong giá trị của thuộc tính.