**TẬP ĐOÀN CÔNG NGHIỆP VIỄN THÔNG QUÂN ĐỘI VIETTEL**

**CÔNG TY ĐẦU TƯ CÔNG NGHỆ VIETTEL**



**BÁO CÁO THỰC TẬP FRONTEND**

**LÝ THUYẾT TUẦN 1**

|  |  |
| --- | --- |
| **NGƯỜI HƯỚNG DẪN:** | **MAI DUY HIẾU** |
| **THỰC TẬP SINH:** | **HOÀNG TIẾN ĐỨC** |
| **PHÒNG BAN:** | **KĨ THUẬT CÔNG NGHỆ** |
|  |  |

**HÀ NỘI - 2025**

**MỤC LỤC**

**[PHẦN 1: TỔNG QUAN VỀ HTML 2](#_Toc21520)**

[1.1. Cấu trúc của HTML 2](#_Toc4727)

[1.2. Các thẻ HTML cơ bản 3](#_Toc30729)

[1.3. Các thuộc tính HTML cơ bản 6](#_Toc23984)

**[PHẦN 2: TỔNG QUAN VỀ CSS 8](#_Toc16278)**

[2.1. CSS là gì? 8](#_Toc8069)

[2.2. Cấu trúc của CSS 8](#_Toc29760)

[2.3. CSS Selector 9](#_Toc18472)

[2.4. CSS Properties (Các thuộc tính css cơ bản) 15](#_Toc1816)

[2.5. Box model là gì? 19](#_Toc18300)

**[PHẦN 3: LAYOUT CƠ BẢN 21](#_Toc7200)**

[3.1. Flexbox 21](#_Toc27342)

[3.2. Grid 24](#_Toc28723)

# 

# PHẦN 1: TỔNG QUAN VỀ HTML

**[HTML](https://topdev.vn/viec-lam-it/html-kt75)** là viết tắt của cụm từ ****Hypertext Markup Language****(Ngôn ngữ đánh dấu siêu văn bản). ****HTML**** được sử dụng để tạo và cấu trúc các thành phần trong trang web hoặc ứng dụng. Nó xác định cách hiển thị văn bản, hình ảnh, và nội dung đa phương tiện trong trình duyệt web.

- HTML sử dụng các thẻ và thành phần để sắp xếp nội dung trang web.

- Được tạo ra vào năm 1990s để chia sẻ tài liệu qua Internet.

- HTML là nền tảng phát triển web, giúp các trang web dễ truy cập và có tổ chức.

## Cấu trúc của HTML

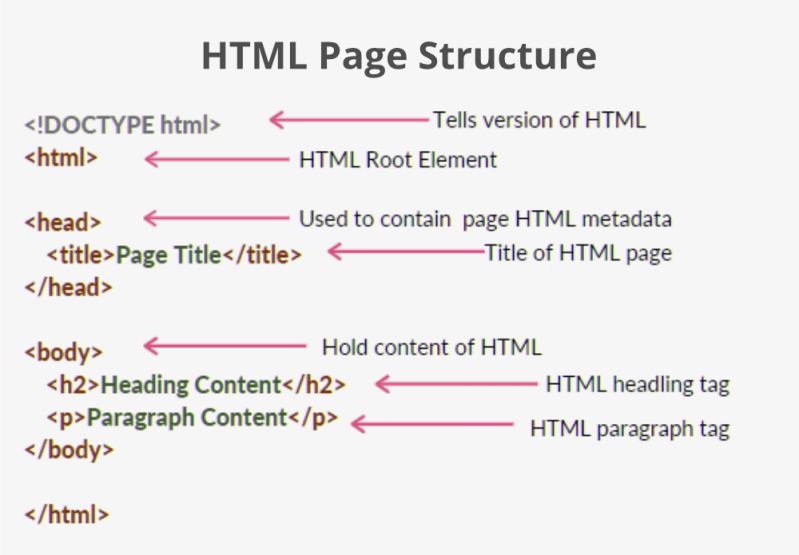
Cấu trúc cơ bản của trang HTML thường gồm các phần:

***<!DOCTYPE html>***: Khai báo loại tài liệu và phiên bản HTML, đảm bảo trình duyệt hiển thị chính xác. (Nó chỉ phải xuất hiện một lần, ở đầu trang, trước bất kỳ thẻ HTML nào)

***<html>***: Phần tử gốc bao gồm tất cả các phần tử html khác, biểu thị phần bắt đầu và kết thúc của tài liệu.

***<head></head>***: Chứa thông tin siêu dữ liệu và trang HTML

***<body></body>***: Bao gồm toàn bộ nội dung hiển thị của trang web, bao gồm văn bản, hình ảnh, liên kết, và các phương tiện truyền thông khác, …



## Các thẻ HTML cơ bản

Cấu trúc cơ bản của một thẻ HTML bao gồm:

**- Thẻ mở:** Xác định điểm bắt đầu của một phần tử. Ví dụ: <p>

**- Nội dung:** Nội dung của phần tử nằm giữa thẻ mở và thẻ đóng.

**- Thẻ đóng:** Xác định điểm kết thúc của một phần tử HTML: Ví dụ </h1>



***Note:*** *Có một số thẻ có thể tự đóng. Ví dụ như: <br>, <hr>*

**1.2.1. Thẻ cấu trúc**

|  |  |
| --- | --- |
| Tên | Mục đích |
| **<html>** | Phần tử gốc bao gồm tất cả các phần tử html khác, biểu thị phần bắt đầu và kết thúc của tài liệu. |
| **<head>** | hứa thông tin siêu dữ liệu và trang HTML. |
| **<body>** | Bao gồm toàn bộ nội dung hiển thị của trang web, bao gồm văn bản, hình ảnh, liên kết, và các phương tiện truyền thông khác, … |
| **<style>** | Dùng để định dạng các thành phần của website như màu sắc, font chữ, … |
| **<title>** | Được sử dụng trong thẻ <head> dùng để xác định tiêu đề trang web. |
| **<meta>** | Thường được đặt trong phần tử <head> và dùng để cung cấp metadata cho trình duyệt và công cụ tìm kiếm. |
| **<link>** | Đặt trong thẻ <head> dùng để định nghĩa một liên kết trên trang web với một liên kết ở bên ngoài. |

**1.2.2. Thẻ định dạng văn bản**

|  |  |
| --- | --- |
| **<h1> - <h6>** | Xác định tiêu đề chính trong một đoạn văn bản (mặc định chữ to và in đậm). |
| **<p>** | Xác định một đoạn văn bản trong <html>. |
| **<em>** | Nhấn mạnh văn bản quan trọng. |
| **<b>** | Định dạng chữ in đậm trong đoạn văn bản. |
| **<strong>** | Chức năng giống <b> nhưng dùng đánh dấu các văn bản ở mức độ quan trọng. |
| **<i>** | Định dạng chữ in nghiêng. |
| **<u>** | Định dạng chữ gạch chân trong nội dung văn bản. |
| **<small>** | Định dạng kích thước nhỏ trong văn bản. |
| **<pre>** | Hiển thị nội dung nguyên hình dạng như trong trình soạn thảo. |
| **<br>** | Được sử dụng để ngắt xuống dòng bới HTML. |
| **<!-- -->** | Chèn một đoạn chú thích. |

**1.2.3. Thẻ định dạng bảng**

|  |  |
| --- | --- |
| **<table>** | Thẻ để xác định dữ liệu bảng, thông tin bên trong thẻ được trình bày trong bảng gồm các hàng và cột dữ liệu. |
| **<caption>** | Xác định chú thích hoặc tiêu đề của bảng. |
| **<colgroup>** | Xác định một nhóm các cột trong một bảng. |
| **<col>** | Phần tử con của <colgroup> và được dùng để xác định thuộc tính cho cột. |
| **<thead>** | Xác định phần nội dung là tiêu đề của bảng. |
| **<tbody>** | Xác định phần nội dung chính của bảng. |
| **<tfoot>** | Xác định phần nội dung kết. |
| **<tr>** | Xác định một hàng trong bảng. |
| **<td>** | Phần tử con của <tr> dùng để xác định một ô trong bảng dữ liệu. |
| **<th>** | Phần tử con của <tr> dùng để xác định tiêu đề của một nhóm ô trong bảng. |

**1.2.4. Thẻ danh sách**

|  |  |
| --- | --- |
| **<ul>** | Xác định danh sách không có thứ tự. |
| **<ol>** | Xác định danh sách có thứ tự rõ ràng. |
| **<li>** | Là thẻ con của <ul>, <ol> dùng để định dạng danh sách. |

**1.2.5.Thẻ phương tiện**

|  |  |
| --- | --- |
| **<area>** | Dùng để xác định một bản đồ ảnh cho phép các hình ảnh bên trong có thể chèn liên kết. |
| **<audio>** | Tạo một đoạn âm thanh hoặc trình phát nhạc vào trang web. |
| **<img>** | Chèn hình ảnh vào trang web. |
| **<source>** | Xác định nguồn tài nguyên phù hợp cho các đa phương tiện (<audio> hoặc <video>). |
| **<video>** | Chèn các nội dung video cho trang . |

**1.2.6. Thẻ liên kết**

|  |  |
| --- | --- |
| **<a>** | Dùng để chèn liên kết vào trang web, thường dùng với thuộc tính href. |
| **<nav>** | Xác định tập hợp của các liên kết. |

**1.2.7. Thẻ biểu mẫu**

|  |  |
| --- | --- |
| **<form>** | Đại diện cho tệp tài liệu chứa các dạng thông tin biểu mẫu, sử dụng để thu thập dữ liệu từ người dùng và là thùng chứa cho các phần tử đầu vào khác nhau (<input>, <button>, …) |
| **<input>** | Dùng để nhập dữ liệu bao gồm các trường thông tin như họ tên, mật khẩu, email, số điện thoại, … |
| **<textarea>** | Dùng để chỉnh sửa văn bản, cho phép người dùng nhập nhiều dòng văn bản tự do như đánh giá, phản hồi, … |
| **<label>** | Thể hiện chú thích cho <input>, tăng khả năng truy cập bằng liên kết đến form tương ứng. |
| **<datalist>** | Chứa một tập hợp các phần tử trong thông tin biểu mẫu. |
| **<select>** | Dùng để tạo một danh sách lựa chọn (danh sách thả xuống). |
| **<option>** | Dùng để xác định các giá trị có trong phần tử <select> hoặc <datalist>. |
| **<button>** | Dùng để thực hiện một hành động (gửi biểu mẫu, hiển thị hộp thoại, …). |

**1.2.8. Các thẻ khác**

|  |  |
| --- | --- |
| **<div>** | Dùng để phân nội dung thành các đoạn bao gồm các phần tử html khác. |
| **<span>** | Dùng để chứa dữ liệu (văn bản) hoặc chứa phần tử ***inline*** khác. |
| **<hr>** | Tạo một đường kẻ nằm ngang. |

## Các thuộc tính HTML cơ bản

* Tất cả các phần tử HTML có thể có các thuộc tính.
* Thuộc tính cung cấp thông tin bổ sung về các yếu tố.
* Thuộc tính luôn được chỉ định trong thẻ bắt đầu.
* Các thuộc tính thường có trong các cặp tên/giá trị như: ***name = "value". Ví dụ: id=”header-container”***

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Thẻ sử dụng** | **Mục đích** |
| **id** | Tất cả | Dùng để đặt tên cho 1 phần tử (Không có nhiều hơn 1 phần tử có chung 1 id). |
| **Class** | Tất cả | Dùng để đặt tên cho 1 hoặc nhiều phần tử (Nhiều phần tử có thể có chung 1 class). |
| **href** | <a>, <link>, <area> | chỉ định URL của trang mà liên kết đến |
| **src** | <img>, <video>, <audio>, <input>, <source>, <script>, <iframe> | URL của nội dung có thể nhúng. |
| **alt** | <area>, <img>, <input> | Chỉ định văn bản thay thế cho hình ảnh, nếu hình ảnh vì lý do nào đó không thể được hiển thị. |
| **placeholder** | <input>, <textarea> | Cung cấp một gợi ý cho người dùng về những gì có thể được nhập. |
| **value** | <button>, <input>, <li>, <data>, <option> | Xác định giá trị mặc định sẽ được hiển thị trong phần tử. |
| **[title](https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes/title)** | Tất cả | Định nghĩa một số thông tin bổ sung cho một phần tử. |
| **type** | <button>, <input>, <script>, <ol>, <style>, <link>, <menu> | Xác định loại ủa phần tử. |
| **width** | <canvas>, <embed>, <iframe>, <img>, <input>,  <object>, <video> | Chỉ định chiều rộng của phần tử. |
| **height** | Chỉ định chiều cao của phần tử. |
| **[hidden](https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes/hidden)** | Tất cả | Sử dụng để ẩn một phần tử. |
| **style** | Tất cả | Được sử dụng để thêm kiểu vào một phần tử, chẳng hạn như màu sắc, phông chữ, kích thước |

# Phần 2: TỔNG QUAN VỀ CSS

## CSS LÀ GÌ?

CSS (Cascading Style Sheets) là ngôn ngữ dùng để tạo kiểu dáng cho các tài liệu HTML. Giúp định dạng giao diện của trang web bằng cách điều chỉnh màu sắc, font chữ, bố cục, kích thước, khoảng cách và nhiều thuộc tính khác.

## CẤU TRÚC CỦA CSS

Cấu trúc của CSS gồm ba phần chính: ***Bộ chọn (selector), thuộc tính (properties), và giá trị (value).***

**- Selector (Bộ chọn)**: Xác định phần tử HTML cần áp dụng CSS.

**- Property (Thuộc tính)**: Xác định kiểu dáng cần áp dụng (màu sắc, kích thước, font chữ,…).

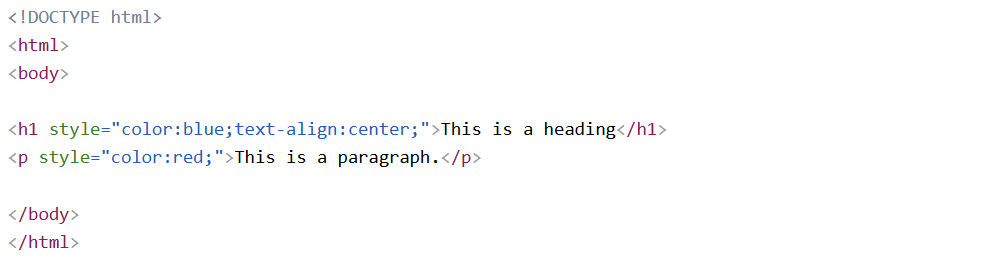
**- Value (Giá trị)**: Chỉ định giá trị cụ thể cho thuộc tính.

p { color: blue; }

- Có 3 cách để chèn CSS:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Inline** | **Internal** | **External** |
| **Khái niệm** | Viết trực tiếp trong ***thuộc tính style*** của thẻ HTML. | Viết trong thẻ ***<style>*** bên trong phần <head> của tài liệu HTML. | Viết trong một tệp ***“.css”*** riêng biệt. |
| **Phạm vi** | Chỉ áp dụng cho một phần tử cụ thể. | Ảnh hưởng đến toàn bộ trang nhưng chỉ trong tệp HTML đó. | Sử dụng thẻ <link> trong phần <head> của HTML để liên kết. |
| **Tính linh hoạt** | Không tái sử dụng được, làm code dài và khó quản lý. | Dễ quản lý hơn inline CSS. | Dễ tái sử dụng, quản lý tốt và tối ưu cho website lớn. |

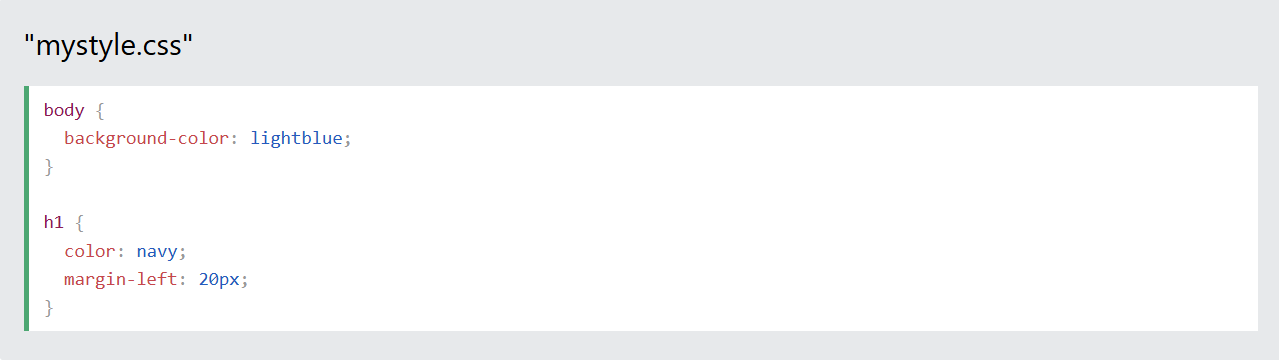
**- Inline CSS**



- **Internal** CSS



- **External CSS**



## Css selector

1. **CSS Simple Selector**

***- Element selector:*** Chọn bất kì phần tử nào đó.

p { color: blue; }

div { color: magenta; }

***- ID selector:*** Chọn tất cả các phần tử có Id đã cho.

#app {

color: red;}

***- Class selector:*** Chọn tất cả các phần tử có class đã cho.

.hello {

color: red;}

***- Universal selector:*** Chọn tất cả các phần tử.

\* {

color: yellow;}

1. **Attributes CSS Selector**

Bộ chọn thuộc tính chọn các phần tử HTML với một tập thuộc tính đã cho.

***- Bộ chọn thuộc tính cơ bản: A[B].*** Chọn tất cả phần tử A có thuộc tính B.

div[for] {

color: red;}

***- Bộ chọn thuộc tính cụ thể A[B=”C”].*** Chọn tất cả các phần tử A có thuộc tính B với giá trị C.

div[for="chocolate"]{

color: blue;}

***- Bộ chọn thuộc tính cụ thể A[B^=”C”].*** Chọn tất cả các phần tử A có thuộc tính B với giá trị bắt đầu C.

div[for^="cho"] {

color: magenta;}

***- Bộ chọn thuộc tính cụ thể A[B$=”C”].*** Chọn tất cả các phần tử A có thuộc tính B với giá trị kết thúc C.

div[for$="er"] {

color: yellow;}

***- Bộ chọn thuộc tính cụ thể A[B\*=”C”].*** Chọn tất cả các phần tử A có thuộc tính B chưa giá trị C.

div[for\*="ut"] {

color: cyan;}

1. **Descendant CSS Selector**

Bộ chọn hậu duệ của bất kì phần tử nào.

***- Any descendant selector: A B.*** Chọn bất kì phần tử B nào là hậu duệ của A.

.container .hello-class {

color: red;}

***- Child selector: A > B.*** Chọn phần tử B là hậu duệ trực tiếp của A.

.paragraph-container > .hello-class {

color: blue;}

1. **Combination CSS Selector**

Conbination CSS Selector cho phép bạn chọn thành phần rất cụ thể bằng nhiều tham chiếu.

p#hola-id {

color: blue;}

1. **Multiple CSS Selector**

Multiple CSS Selector cho phép chúng ta chọn nhiều phần tử không liên quan với nhau.

.outside-class, .again-class, .hola-class {

color: purple;}

1. **Sibling CSS Selector**

Sibling CSS Selector nhắm chọn các phần tử anh chị em.

***- Bộ chọn Anh / Chị / Em liền kề (Nghiêm ngặt): A + B.*** Nhắm mục tiêu một phần tử anh chị em được đặt ngay sau phần tử đó.

#hola-id + .hello-class {

color: blue;}

***- Bộ chọn Anh / Chị / Em liền kề (Không nghiêm ngặt): A ~ B.*** Giống như bộ chọn bên trên nhưng không giới hạn 1 phần tử đầu tiên.

#hola-id ~ .hello-class {

color: purple;}

1. **Pesudo CSS Selector**

***- Chọn phần tử đầu tiên: A:first-child.*** Nó chọn đến phần tử con đầu tiên. (A phải có cha mẹ).

li:first-child {

color: blue;}

***- Chọn phần tử con cuối cùng: A:last-child.*** Hoạt động như A:first-child, ngoại trừ thay vì chọn phần tử con đầu tiên, nó chọn phần tử con cuối cùng.

li:last-child {

color: purple;}

***- Chọn phần tử con thứ N: A:nth-child(n).*** Nó chọn từng mục tiêu là con thứ n của cha mẹ nó.

li:nth-child(2) {

color: red;}

***- Hover Selector: A:hover.*** Chọn phần tử được hover. Thường được sử dụng để làm nổi bật các liên kết.

a:hover {

color: red;}

***- Focus Selector: A:focus.*** Chọn phần tử bạn đang tập trung vào nó. Thường được sử dụng với input.

input:focus {

background: red;}

***- Active Selector: A:active.*** Chọn phần tử đang có trạng thái active.

.cheesyLink:active {

background: red;}

## CSS Properties (Các thuộc tính CSS cơ bản)

***- Thuộc tính display:***

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Giá trị** | **Ví dụ** | **Mục đích** |
| display | block | *display: block;* | Thành phần hiển thị như một khối, khi sử dụng giá trị block thành phần sẽ đứng một hàng độc lập so với thành phần trước và sau nó. |
| inline | *display: inline;* | Thành phần sẽ hiển thị như một nội tuyến (inline, không ngắt dòng), đây là dạng mặc định. |
| inline-block | *display: inline-block;* | Thành phần sẽ hiển thị như một khối, nhưng là một khối nội tuyến |
| flex | *display: flex;* | Sắp xếp, bố cục các thành phần khối (block) một cách dễ dàng, linh hoạt. |
| inline-flex | *display: inline-flex;* | Sắp xếp, bố cục các thành phần khối (block) một cách dễ dàng, linh hoạt (dưới dạng inline). |
| grid | *display: grid;* | Sắp xếp các thành phần trên một trang theo các vùng. |

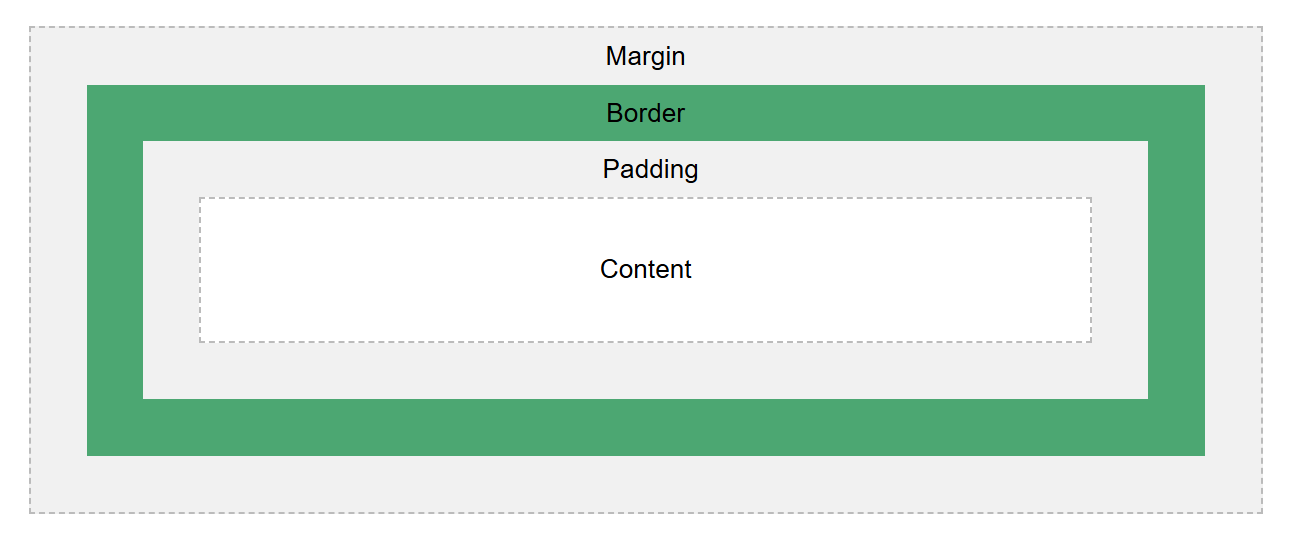
|  |  |
| --- | --- |
| display: block; | {281FDD65-AA8B-49C3-8DBD-0929D6F67739} |
| display: inline; |  |
| display: inline-block; |  |
| display: flex; |  |
| display: inline-flex; |  |
| display: grid; |  |

***- Các thuộc tính khác:***

|  |  |  |
| --- | --- | --- |
| **Box Properties** | | |
| **width** | Thiết lập chiều rộng của phần tử. | *width: 200px;* |
| **height** | Thiết lập chiều cap của phần tử. | *height: 100px;* |
| **margin** | Thiết lập khoảng cách bên ngoài xung quanh phần tử. | *margin-top: 5px;* |
| *margin-right: 10px;* |
| *margin-bottom: 10px;* |
| *margin-left: 5px;* |
| ***margin: top right bottom left*** | | |
| **padding** | Thiết lập khoảng cách bên trong giữa nội dung và viền của phần tử. | *padding-top: 5px;* |
| *padding-right: 10px;* |
| *padding-bottom: 10px;* |
| *padding-left: 5px;* |
| ***padding: top right bottom left*** | | |
| **border** | Thiết lập viền của phần tử. | *border-color: #000;* |
| *border-style: solid;* |
| *border-width: 2px;* |
| *border-top: 2px solid #000;* |
| *border-right: 2px solid #000;* |
| *border-bottom: 2px solid #000;* |
| *border-left: 2px solid #000;* |
| ***border: 2px solid #000;*** | | |
| **border-radius** | Định dạng các dạng bo góc của border. | *border-radius: 12px* |
| **Text Properties** | | |
| **color** | Thiết lập màu chữ. | *color: #f1f1f1* |
| **font-family** | Thiết lập phông chữ cho văn bản. | *font-family: “Quicksand”, sans-serif;* |
| **font-size** | Thiết lập kích thước chữ. | *font-size: 16px;* |
| **text-align** | Căn lề văn bản (trái, phải, trên, dưới). | *text-align: center;*  *text-align: left;* |
| **text-decoration** | Trang trí văn bản (gạch chân, gạch ngang, nhấp nháy, không hiệu ứng). | *text-decoration: underline;*  *text-decoration: none;* |
| **line-height** | Thiết lập chiều cao dòng. | *line-height: 1.5;* |
| **letter-spacing** | Thiết lập khoảng cách giữa các ký tự. | *letter-spacing: 4px;* |
| **text-transform** | Thiết lập các ký tự viết hoa cho văn bản. | *text-transform: uppercase;*  *text-transform: lowercase;* |
| **Background Properties** | | |
| **background-color** | Thiết lập màu nền. | *background-color: red* |
| **background-image** | Thiết lập hình nền. | *background-image: url(…)* |
| **background-repeat** | Thiết lập cách lặp lại của hình nền. | *background-repeat: no-repeat;*  *background-repeat: repeat;* |
| **background-position** | Thiết lập vị trí của hình nền. | *background-position: top;*  *background-position: left;*  *background-position: center;*  *background-position: 100px 200px;* |
| **background-size** | Thiết lập kích thước của hình nền. | *background-size: contain;*  *background-size: cover;*  *background-size: 300px 100px;* |
| ***background: url position / size repeat*** | | |
| **Positioning Properties** | | |
| **position** | Thiết lập kiểu định vị của phần tử (static, relative, absolute, fixed, sticky). | *position: relative;* |
| **top** | Thiết lập vị trí trên của phần tử. | *top: 10px;* |
| **right** | Thiết lập vị trí phải của phần tử. | *right: 10px;* |
| **bottom** | Thiết lập vị trí dưới của phần tử. | *bottom: 10px;* |
| **left** | Thiết lập vị trí trái của phần tử. | *left: 10px;* |
| **z-index** | Thiết lập thứ tự xếp chồng của phần tử. | *z-index: 99;* |
| **Shadow Properties** | | |
| **box-shadow** | Thiết lập bóng cho phần tử. | box-shadow: 2px 2px 5px 10px #ccc; |
| ***box-shadow: h-offset v-offset blur spread color optional;*** | | |
| **text-shadow** | Thiết lập bóng cho văn bản. |  |
| ***text-shadow: h-offset v-offset blur color;*** | | |

## Box model là gì?

Bất kỳ phần tử HTML nào của trang web đều được trình duyệt thể hiện dưới dạng một hình hộp chữ nhật.



Cấu trúc của box model gồm 4 phần: ***Content, Padding, Border, Margin.***

**- Content area:** là vùng chứa nội dung của một element, với chiều rộng/cao được xác định qua thuộc tính width và height. Vùng này thường chứa text, hình ảnh, video…

**- Padding area:** Các phần đệm nằm xung quanh vùng nội dung. (Sử dụng thuộc tính padding để xác định kích thước vùng).

**- Border area**: Một biên giới bao xung quanh phần đệm và nội dung.(Sử dụng thuộc tính border để xác định kích thước vùng).

**- Margin area:** Khoảng trắng ngoài cùng bao bọc các phần bên trong (border, padding, content). (Sử dụng thuộc tính margin để xác định kích thước vùng).

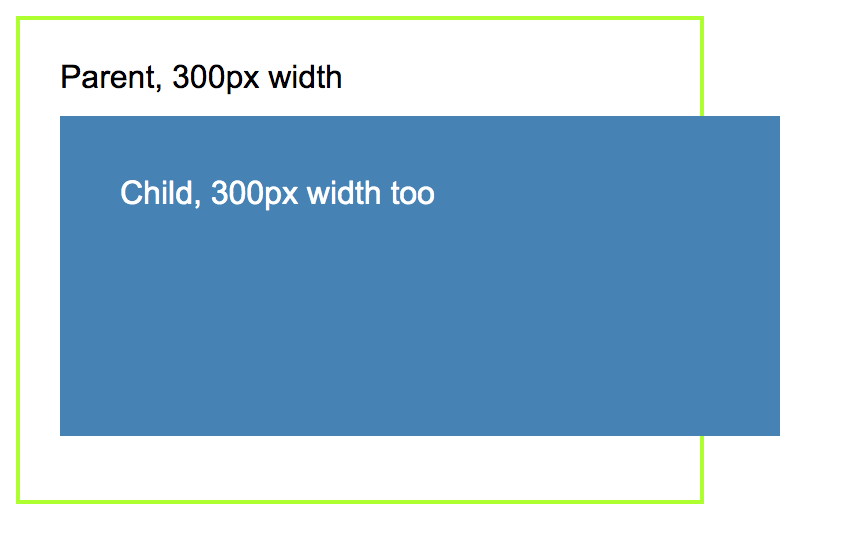
#### Công thức tính đơn giản mặc định:

#### **- Chiều rộng** của một element = chiều rộng content **width**+ **padding trái** +**padding phải** + **border trái** + **border phải**

#### **- Chiều cao**của một element = chiều cao content**height + padding trên + padding dưới + border trên + border dưới**

***- Các vấn đề thường gặp ở box-model***

* **Vấn đề 1:** Khi layout một trang web, chắc chắn sẽ có những lúc bạn tăng giảm padding của một element, hay thêm border cho nó đẹp hơn (theo thiết kế mới của designer chẳng hạn). Vấn đề phát sinh là khi bạn thực hiện các thay đổi trên, tổng *chiều rộng của element sẽ thay đổi và làm cả layout bị xê dịch* vì những thay đổi đó.
* **Vấn đề 2:** chiều rộng của element con bị tràn ra khỏi cha



***- Giải pháp***

div {

Box-sizing: border-box;}

#### Công thức tính theo **box-sizing: border-box**

#### **- Chiều rộng** của một element **= width = chiều rộng content (auto) + padding trái + padding phải + border trái + border phải**

#### **- Chiều cao** của một element **= height = chiều cao content (auto) + padding trên + padding dưới + border trên + border dưới**

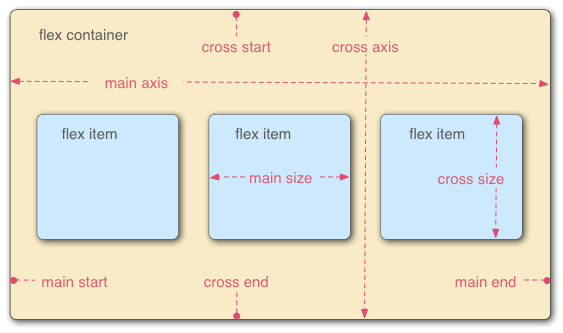
# Phần 3: Layout cơ bản

Layout hay còn gọi là bố cục trang, là sự sắp xếp hợp lý các yếu tố thiết kế (văn bản, hình ảnh, đồ họa, nội dung ) để tạo sự đồng nhất, thẩm mỹ cho một ấn phẩm.

## Flexbox

CSS flexbox là một one-dimensional(hay còn gọi là 1D) layout pattern. Với flexbox, chúng ta có thể dễ dàng sắp xếp các items từ trái sang phải, từ trên xuống dưới, đồng thời kiểm soát khoảng cách và thứ tự của các items trong container.

**- Cấu trúc của Flexbox:**



Hai thành phần chính của flexbox là: thùng chứa cha **(flex container)** và các phần tử con nằm bên trong **(flex items)**.

* **main axis:** Trục này chính là hướng của các item hiển thị, mặc định thì sẽ chạy từ trái qua phải.
* **cross axis:** Trục này vuông góc với main axis, chạy từ trên xuống dưới.
* **main size:**Là kích thước của mỗi item dựa theo trục main axis.
* **cross size:** Là kích thước của mỗi item dựa theo trục cross axis.
* **main start, main end, cross start, cross end:** Điểm bắt đầu của container theo main axis được gọi là main start, điểm kết thúc của container theo main axis gọi là main end, với cross start và cross cũng tương tự nhưng dựa theo cross axis.

**- Các thuộc tính của flex container**

***display: flex - Khởi tạo một flex container***

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Mục đích** | **Giá trị** | **Ví dụ** |
| **flex-direction** | Chỉ định hướng hiển thị của các item. | row | IMG_256 |
| row-reverse | IMG_256 |
| column | IMG_256 |
| column-reverse | IMG_256 |
| **flex-wrap** | Cho phép item tự động xuống dòng khi kích thước container thay đổi. | nowrap (default) | IMG_256 |
| wrap | IMG_256 |
| wrap-reverse | IMG_256 |
| **justify-content** | Dùng để căn chỉnh vị trí của các items so với trục chính(main axis). | flex-start | IMG_256 |
| flex-end | IMG_256 |
| center | IMG_256 |
| space-between | IMG_256 |
| space-around | IMG_256 |
| space-evenly | IMG_256 |
| **align-items** | Dùng để xác định cách mà các flex item được đặt trong container dọc theo chiều cross axis. | stretch | IMG_256 |
| flex-start | IMG_256 |
| flex-end | IMG_256 |
| center | IMG_256 |
| baseline | IMG_256 |
| **align-content** | Tương tự như justify-content chỉ khác một chỗ là thay vì căn theo trục main axis thì căn theo trục cros axis. | stretch | IMG_256 |
| flex-start | IMG_256 |
| flex-end | IMG_256 |
| center | IMG_256 |
| space-between | IMG_256 |
| space-around | IMG_256 |

- **Các thuộc tính của flex item**

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Mục đích** | **Ví dụ** | |
| **flex-grow** | Cho phép các phần tử giãn theo độ rộng của container. | *flex-grow: 2;* |  |
| **flex-shrink** | Ngược lại với thuộc tính flex-grow, nó không giãn ra mà lại co lại khi chúng ta thay đổi độ rộng của container. | *flex-shrink: 2;* | **{B6C776E2-9B43-4865-9FF0-6629499A2012}** |
| **flex-basis** | Sử dụng để xác định độ dài ban đầu của một item. | *flex-basis: 200px;* |  |

## Grid

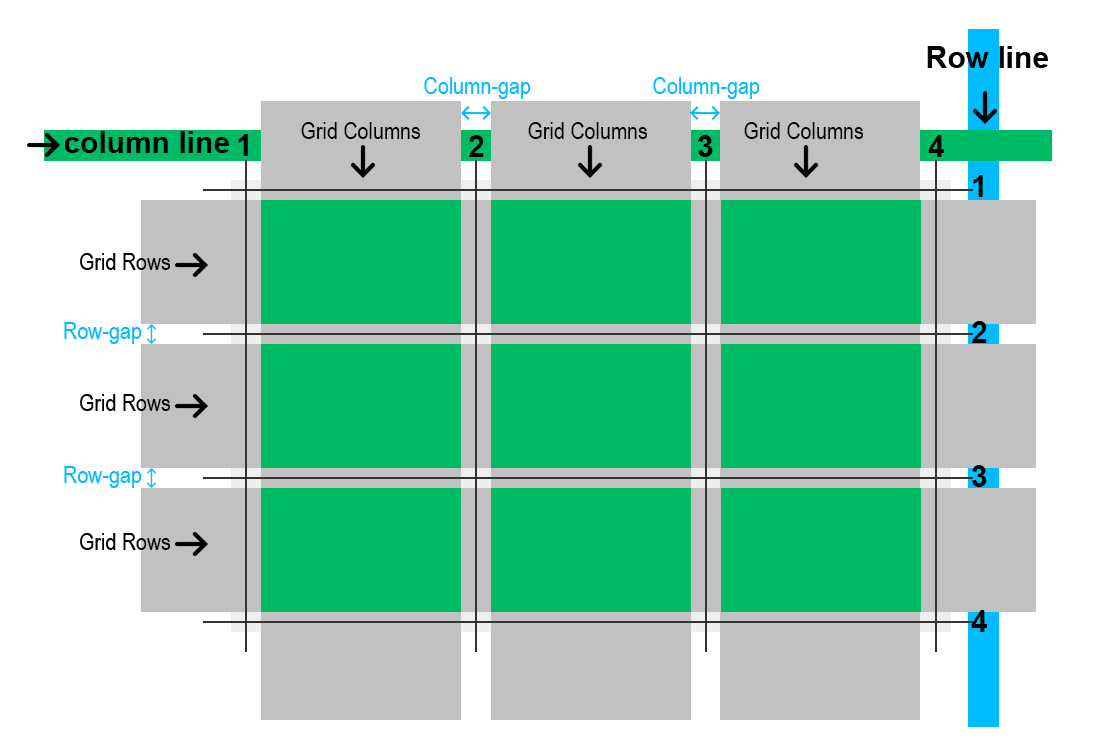
* Mô-đun bố cục lưới cung cấp một hệ thống bố cục dựa trên lưới, với các hàng và cột.
* Bố cục lưới cho phép các nhà phát triển dễ dàng tạo bố cục web phức tạp.
* Bố cục lưới giúp thiết kế cấu trúc bố cục đáp ứng dễ dàng hơn mà không cần sử dụng float hoặc possition.
* ***So sánh Flexbox và Grid***

|  |  |
| --- | --- |
| **Flexbox** | **Grid** |
| Một chiều | Hai chiều |
| Gồm hàng hoặc cột | Gồm hàng và cột |

* ***Thành phần của Grid CSS bao gồm:***
* ***Grid Container*** (thùng chứa các grid items) và Grid Items.

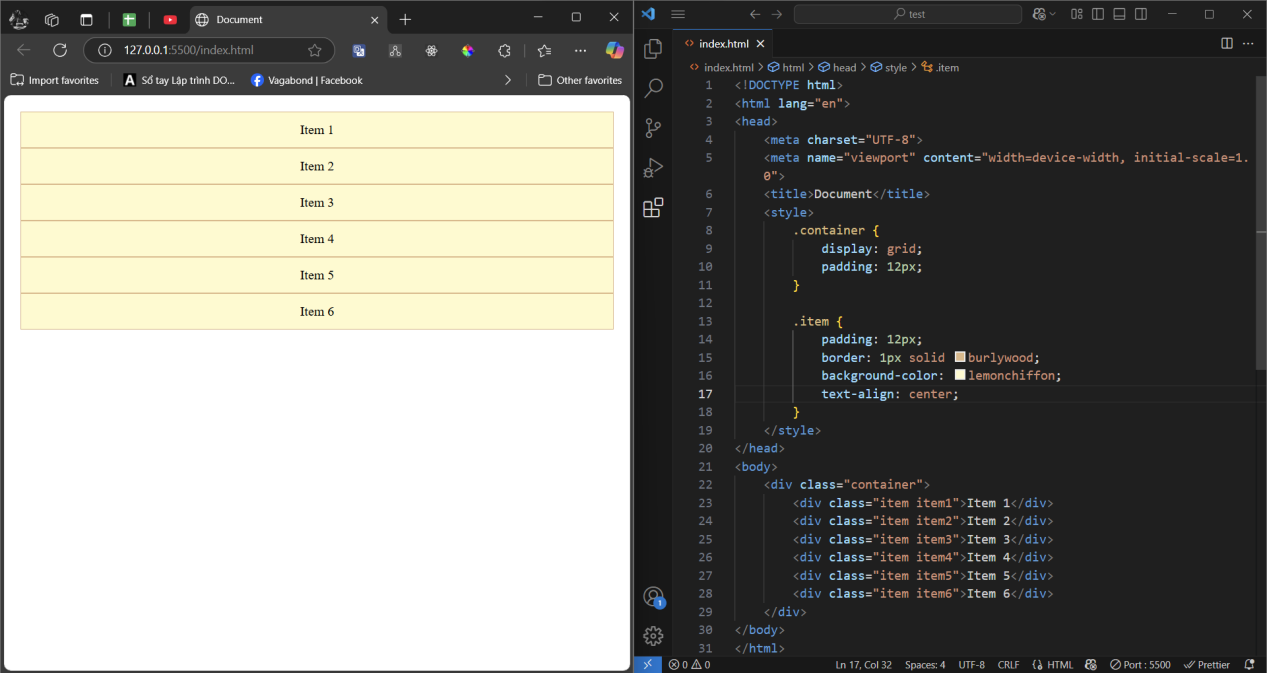
- Sử dụng display: grid; để tạo một Grid Container và các phần tử con trực tiếp của nó sẽ là các grid items.

* Trên Grid có các hàng và các cột được gọi là ***Grid track*** (grid columns - cột và grid rows - hàng).
* Mỗi hàng hoặc cột được tạo bởi 2 lề được gọi là ***Grid line*** .
* Khi các Grid line giao nhau sẽ tạo ra một ô được gọi là ***Grid cells*** (đơn vị nhỏ nhất trong Grid CSS).
* ***Grid Area*** một vùng mà có thể quy định được vùng đó có bao nhiêu cột và bao nhiêu hàng.

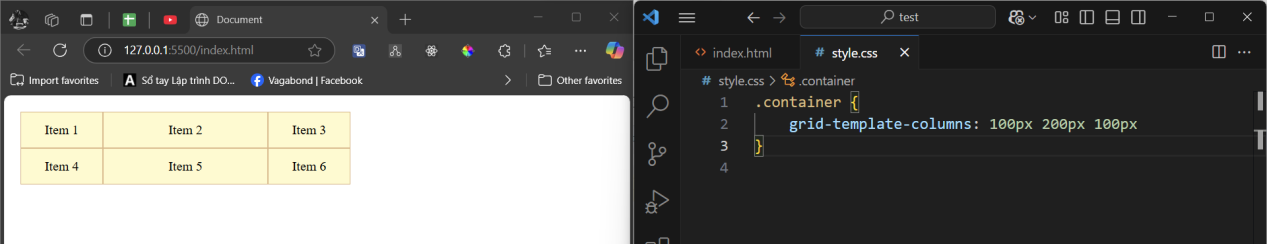


* ***Các thuộc tính Grid CSS***

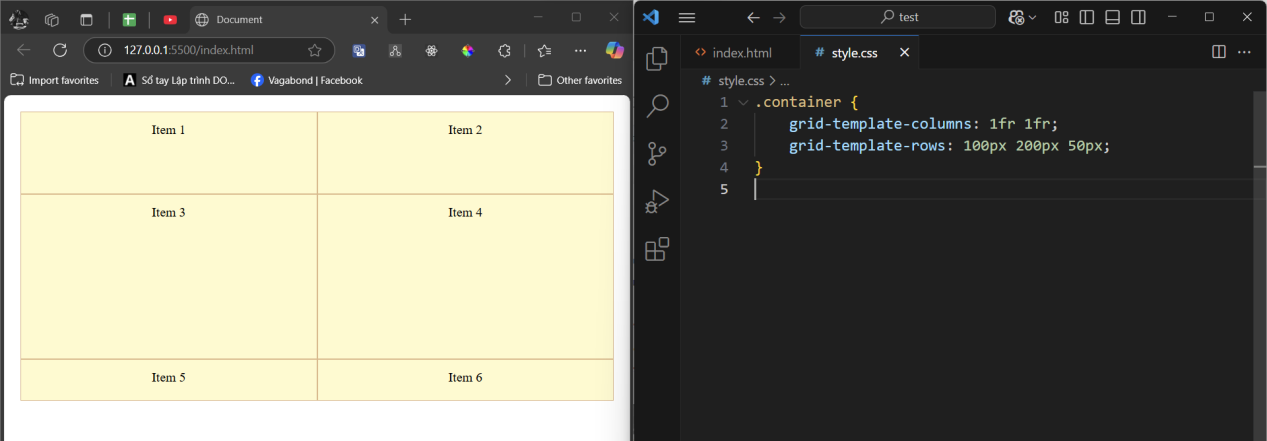
***- display: grid - Tạo một Grid container.***

******

- **grid-template-columns:** Khai bảo grid có bao nhiêu cột với kích thước bao nhiêu (số value của thuộc tính là số lượng cột).

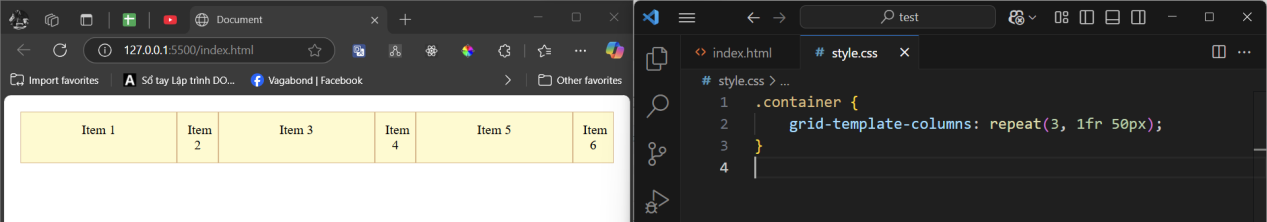
******

**- grid-template-rows:** Khai bảo grid có bao nhiêu hàng với kích thước bao nhiêu (số value của thuộc tính là số lượng hàng ).



***Note***: Trong trường hợp muốn chia cột hoặc hàng với số lượng lớn:

**- grid-template-columns: repeat(<số lần lặp>, số lượng cột);**

****

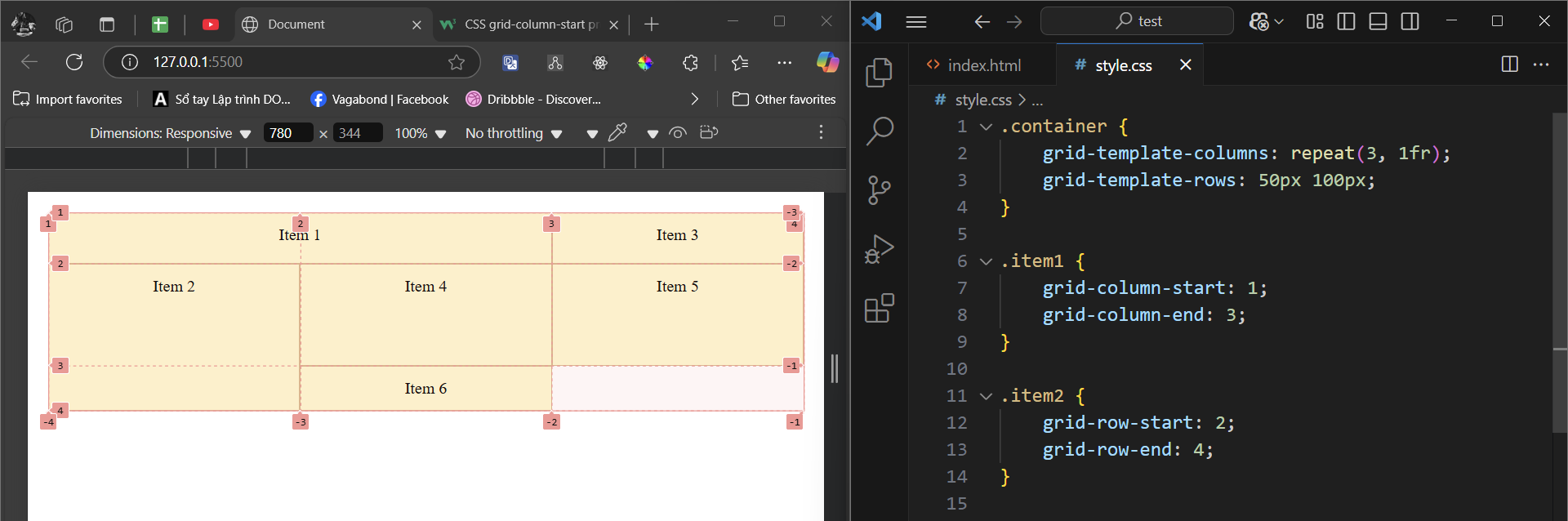
**- grid-template-columns: minmax(<giá trị min>, <giá trị max>):** Xác định phạm vi kích thước lớn hơn hoặc bằng giá trị tối thiểu và nhỏ hơn hoặc bằng giá trị tối đa

* **Thuộc tính phân bố grid item theo grid line (Các thuộc tính được sử dụng trên grid item)**

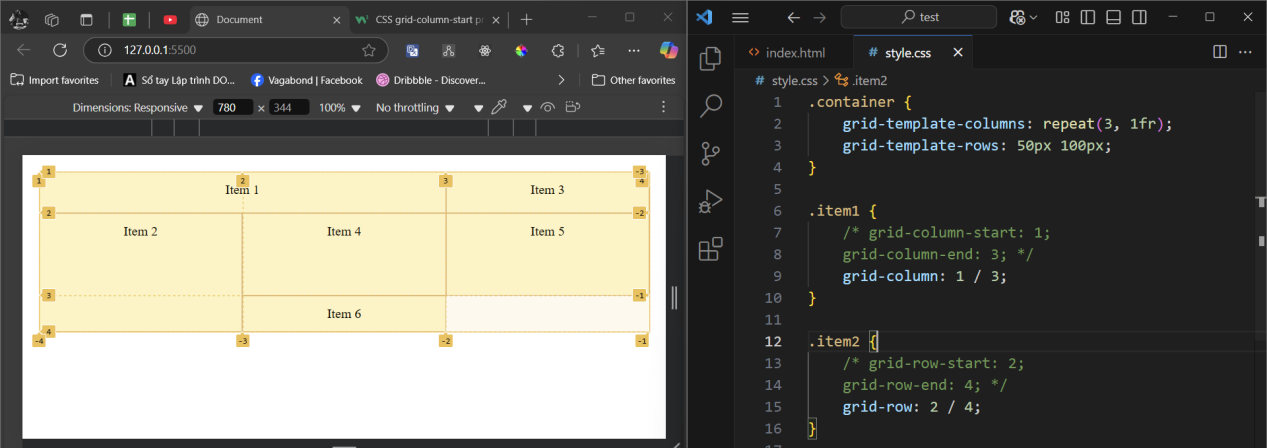
Grid-column-start: Thuộc tính xác định trên grid line nào, grid item sẽ bắt đầu.

Grid-column-end: Thuộc tính xác định trên grid line nào, grid item kết thúc.

Tương tự với grid-row-start và grid-row-end.



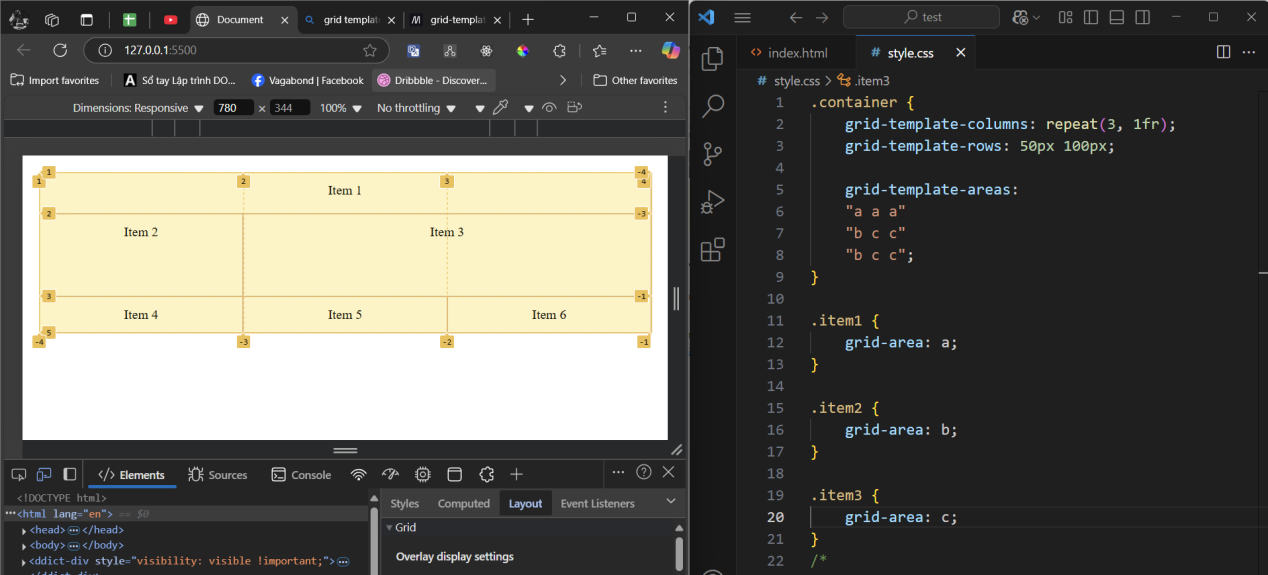
*Cách viết ngắn gọn:*



* **Thuộc tính phân bố grid item theo grid area**

- Khai báo một grid area: **grid-template-areas: <hình dáng>** (Thuộc tính sẽ viết tại phần tử Grid container).

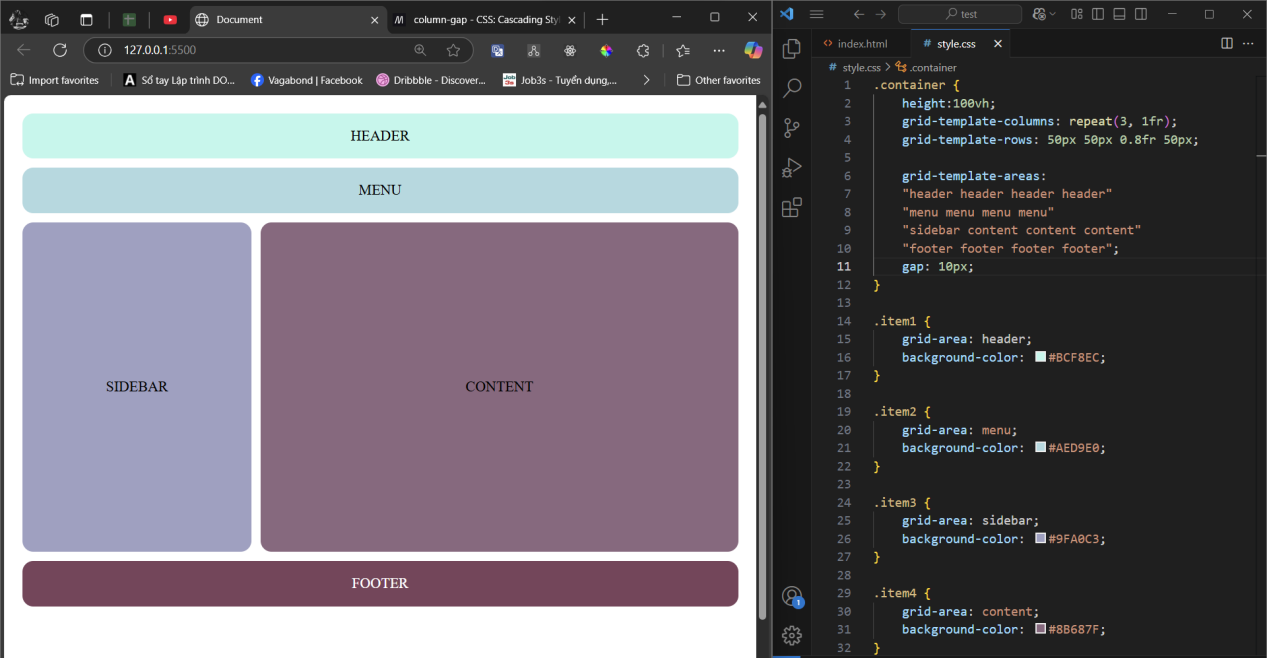
- **grid-area**: chỉ định Grid item vào một vị trí nào đó trong Grid area.

****

- **column-gap:** Tạo khoảng cách giữa các cột của một phần tử.

- **column-row:** Tạo khoảng cách giữa các dòng của một phần tử.

* **Ví dụ thêm:**

****