

Mã sinh viên : 1971020025

Họ và Tên : Nguyễn Tuấn Anh

PHIẾU HỌC TẬP CSS [14.1] - GRID: DỤNG KHUNG (CONTAINER)

Thuộc tính (của Cha)	Mục đích sử dụng (Dùng để làm gì?)	Ví dụ giá trị phổ biến
display: grid;	Kích hoạt môi trường layout Grid	grid
grid-template-columns	Định nghĩa số lượng VÀ độ rộng của các CỘT	100px 100px, 1fr 2fr, repeat(3, 1fr)
grid-template-rows	Định nghĩa số lượng VÀ độ cao của các DÒNG	200px 400px, auto
gap (hoặc grid-gap)	Tạo khoảng trống	10px, 10px 20px

1. Đơn vị fr (Fraction - Phần số) có nghĩa là gì?

- **fr** là đơn vị dùng để phân chia phần còn lại của không gian trong Grid. 1fr tương ứng với một phần chia được trong tổng số các phần sử dụng. Nếu có 3 cột: 1fr 2fr 1fr, tổng cộng là 4 phần; cột đầu chiếm 1 phần, cột giữa chiếm 2 phần, cột cuối chiếm 1 phần.
- **fr** rất tiện để tạo bố cục tự động chia đều hoặc chia tỷ lệ linh hoạt.

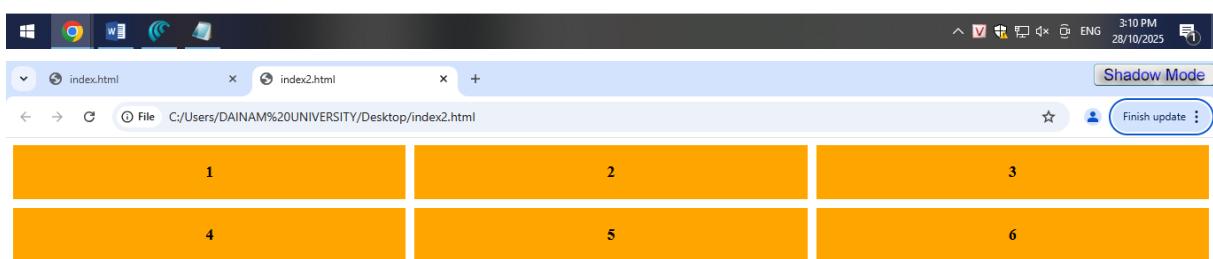
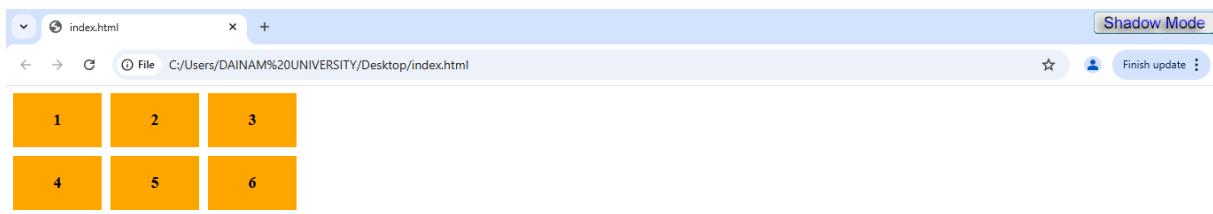
2. Nếu bạn viết grid-template-columns: 1fr 2fr; bố cục sẽ được chia như thế nào?

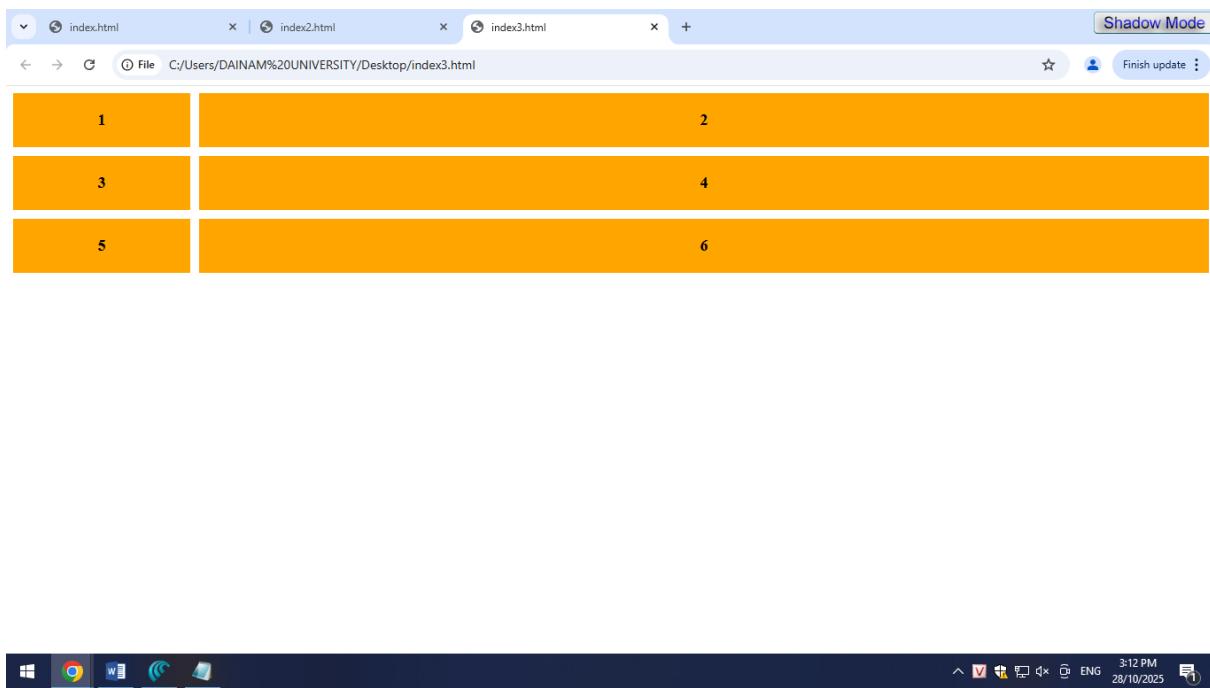
- Tổng cộng là **3 phần**, cột đầu chiếm 1 phần, cột sau chiếm 2 phần. Diện tích cột thứ hai sẽ gấp đôi cột thứ nhất.

3. Hàm repeat(3, 1fr) tương đương với việc viết 1fr 1fr 1fr đúng không?

- Đúng, repeat(3, 1fr) có nghĩa là tạo 3 cột đều nhau, mỗi cột chiếm 1fr. Viết 1fr 1fr 1fr hoặc repeat(3, 1fr) kết quả giống nhau.

2. Ví dụ thực hành





3. Câu hỏi/Thắc mắc của tôi

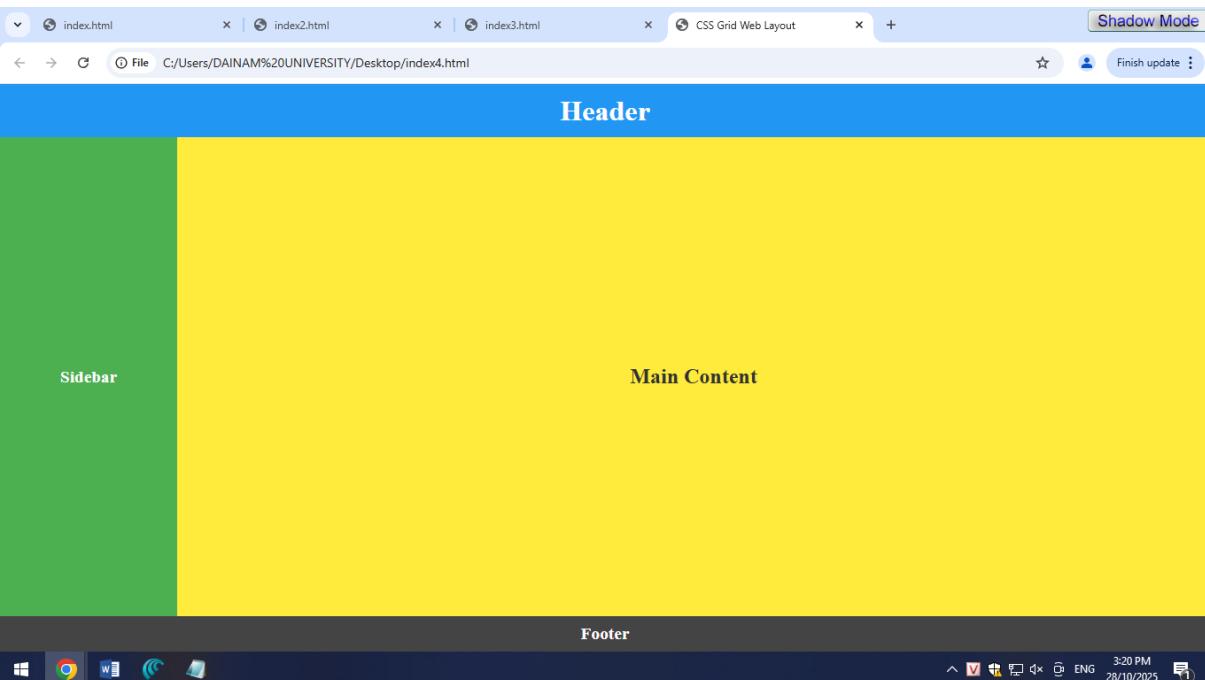
PHIẾU HỌC TẬP CSS [14.2] - GRID: SẮP XÉP ITEMS (CON)

1. Kiến thức cốt lõi (Vượt ô - Spanning)

Thuộc tính (của Con)	Mục đích sử dụng (Dùng để làm gì?)	Ví dụ cú pháp
grid-column-start	Chỉ định đường kẻ cột bắt đầu	1
grid-column-end	Chỉ định đường kẻ cột kết thúc	3
grid-column	Cú pháp tắt cho 2 thuộc tính trên (bắt đầu/kết thúc)	1 / 3
grid-row-start	Chỉ định đường kẻ dòng bắt đầu	1
grid-row-end	Chỉ định đường kẻ dòng kết thúc	2

Thuộc tính (của Con)	Mục đích sử dụng (Dùng để làm gì?)	Ví dụ cú pháp
grid-row	Cú pháp tắt cho 2 thuộc tính trên (bắt đầu/kết thúc)	1 / 2

2. Ví dụ thực hành



3. Câu hỏi/Thắc mắc của tôi

PHIẾU HỌC TẬP CSS [15] - GIỚI THIỆU RESPONSIVE WEB DESIGN (RWD)

1. Kiến thức cốt lõi (Trang web "co giãn")

1. Responsive Web Design là gì?

Là làm web sao cho lên máy tính, điện thoại, máy tính bảng gì cũng đẹp, không bị méo, không cần kéo qua kéo lại, nhìn đâu cũng vừa mắt.

2. Viewport là gì?

Đại loại viewport là cái “cửa sổ nhìn” web trên điện thoại hoặc máy tính, kiểu như mình nhìn web được bao nhiêu là do viewport quyết định.

3. Thẻ meta bắt buộc:

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

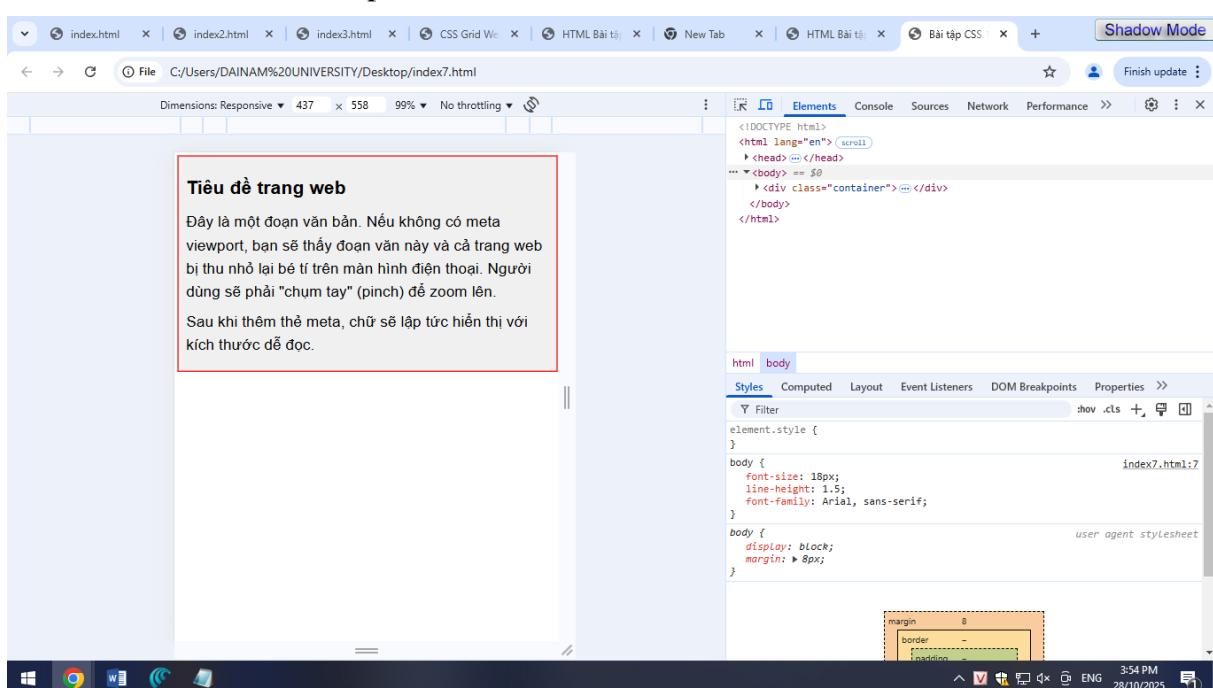
- **width=device-width:** Nói với trình duyệt “Ê, độ rộng web phải bằng đúng cái màn hình máy luôn nha!”
- **initial-scale=1.0:** Lúc vào trang thì hiển thị đúng tỷ lệ gốc, không phóng to hay thu nhỏ gì hết.

4. Mobile First là gì?

Là làm giao diện web cho điện thoại trước, sau đó mới canh chỉnh rộng ra cho máy tính sau. Vì giờ ai cũng xài điện thoại nên ưu tiên nó trước, làm cho nhẹ, dễ nhìn, mượt trên di động.

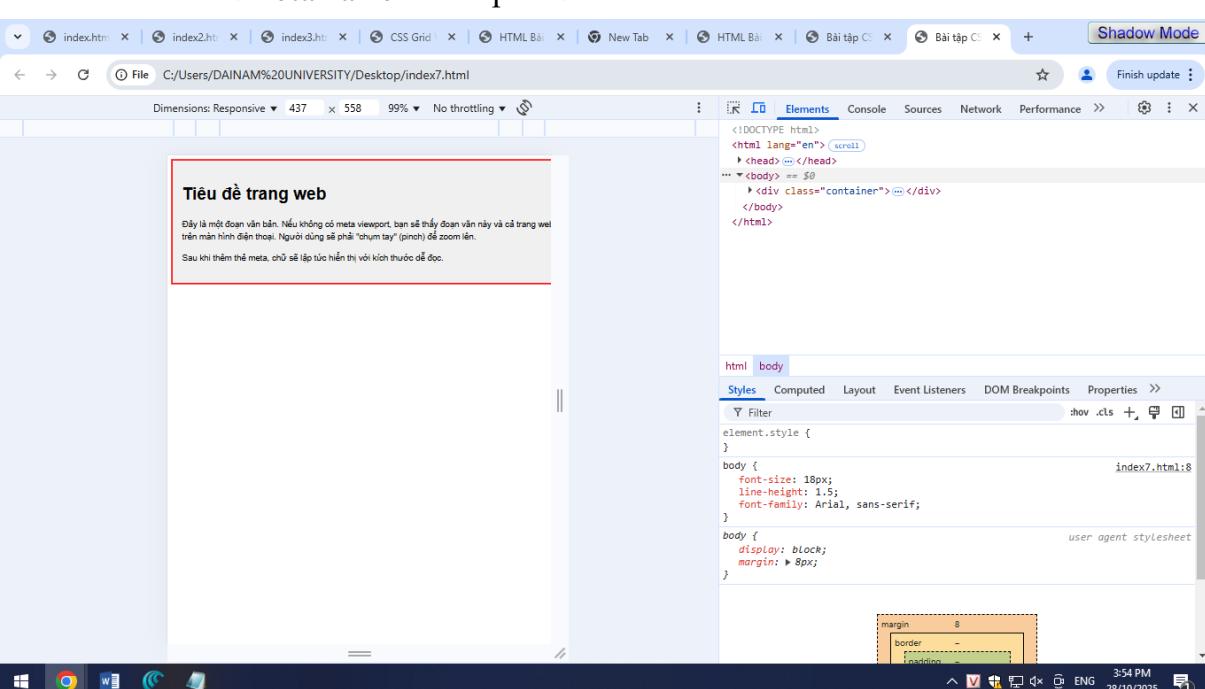
2. Ví dụ thực hành

Ảnh 1: KHÔNG CÓ < meta name="viewport" >



- Nhận xét: Trang web bị thu nhỏ chữ khó đọc

Ảnh 2: ĐÃ CÓ < meta name="viewport" >



• Nhận xét: chữ đã dễ nhìn hơn những viền đỏ đã bị chèn ra ngoài màn hình điện thoại

3. Câu hỏi/Thắc mắc của tôi

Một điều tôi chưa hiểu rõ hoặc muốn hỏi thêm về chủ đề này:

• thêm thẻ meta đã dễ đọc hơn nhưng cái hộp nó lại bị chèn ra bị ngoài màn hình liệu nó có bị mất chữ không và nếu mất chữ thì liệu thẻ meta có thực sự hữu dụng hay còn vào tùy vào px của thẻ

PHIẾU HỌC TẬP CSS [16] - MEDIA QUERIES (Bí thuật Responsive)

1. Kiến thức cốt lõi

1. @media:

- Dùng để viết các quy tắc CSS chỉ áp dụng cho từng loại màn hình khác nhau (responsive – ví dụ: điện thoại, tablet, PC...).

2. (min-width: 768px) (Triết lý Mobile First):

- (min-width: 768px) có nghĩa là: "Áp dụng style này cho các màn hình có chiều rộng **lớn hơn hoặc bằng** 768px."
- (lớn hơn hay nhỏ hơn?): **lớn hơn**

3. (max-width: 600px) (Triết lý Desktop First):

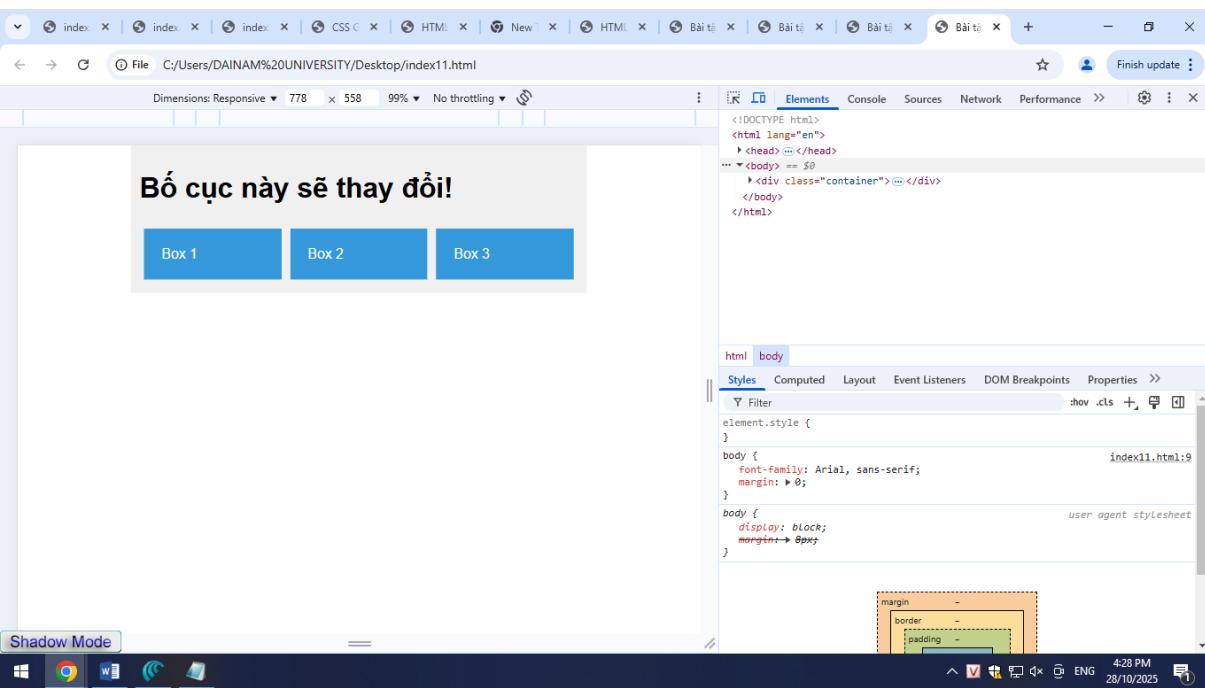
- (max-width: 600px) có nghĩa là: "Áp dụng style này cho các màn hình có chiều rộng **nhỏ hơn hoặc bằng** 600px."
- (lớn hơn hay nhỏ hơn?): **nhỏ hơn**

4. Breakpoints (Điểm ngắt):

- Các giá trị như 768px (tablet), 992px (laptop) được gọi là gì?
- **Điểm ngắt (breakpoints)**

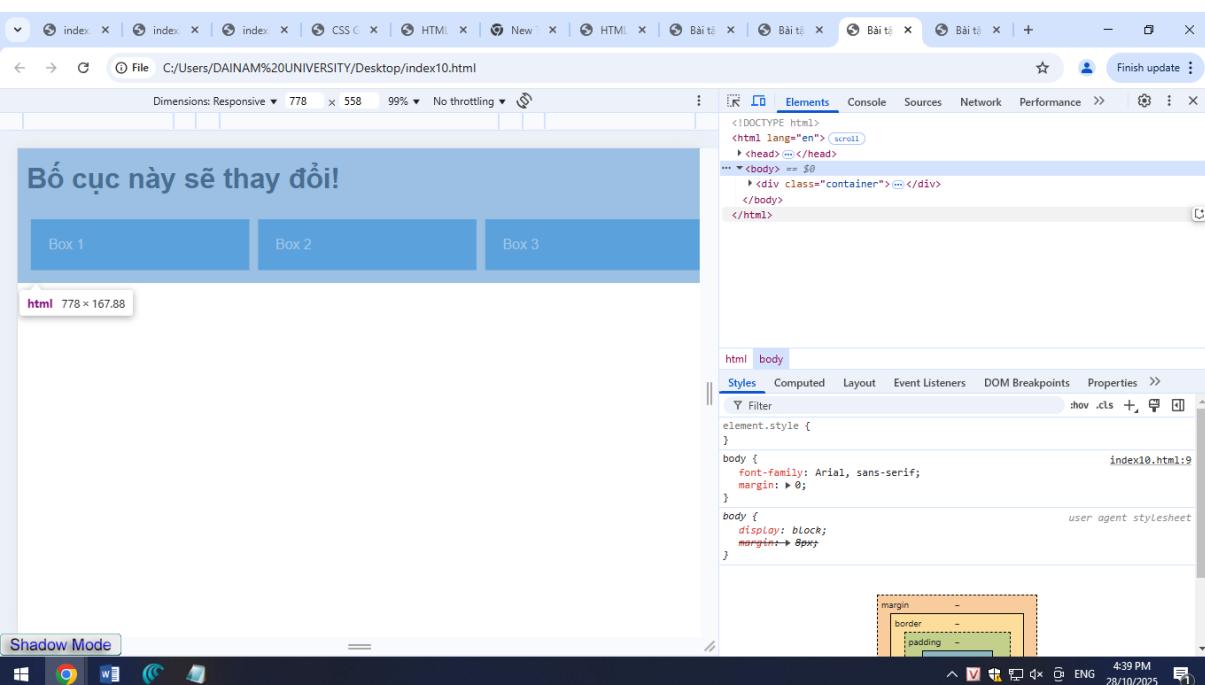
2. Ví dụ thực hành (Giải cứu layout)

Ảnh 1: Màn hình điện thoại (Dưới 768px)



- Nhận xét: 3 cái hộp màu xanh xếp **ngang**, mỗi hộp chiếm **1/3** chiều rộng.

Ảnh 2: Màn hình Desktop (Trên 768px)



- Nhận xét: 3 cái hộp màu xanh xếp **ngang**, lấp đầy container.

3. Câu hỏi/Thắc mắc của tôi

1. Vấn đề (The Problem)

2. Kiến thức cốt lõi (Giải pháp "Chất lỏng")

Thuộc tính	Mục đích sử dụng (Giải thích bằng lời của bạn)
max-width: 100%	Đảm bảo chiều rộng của ảnh/video/iframe tự động co giãn vừa với khung chứa, không vượt quá khung.
height: auto	Giữ đúng tỷ lệ gốc của ảnh/video/iframe khi chiều rộng thay đổi, tránh méo hình.

Object-fit:

- **object-fit: cover;**
 - Giữ tỷ lệ, lắp đầy hộp; chấp nhận bị xén (ảnh sẽ bị cắt méo, nhưng luôn phủ kín hộp).
- **object-fit: contain;**
 - Giữ tỷ lệ, vừa khít trong hộp; chấp nhận bị hở (ảnh nhỏ lại để vừa hộp, vẫn đúng tỷ lệ, nhiều khi có khoảng trống/hở quanh ảnh).

3. Ví dụ thực hành