

Bố cục với Flexbox & CSS Grid



Mục tiêu

Chủ đề chính:

- Làm chủ bố cục trang web với **Flexbox** và **CSS Grid**

Mục tiêu buổi học:

- ✓ Hiểu và sử dụng được **Flexbox** để sắp xếp phần tử theo hàng/dòng
- ✓ Áp dụng **CSS Grid** để chia bố cục thành hàng - cột linh hoạt
- ✓ Biết cách **kết hợp cả hai** để tạo giao diện đẹp, dễ responsive



Flexbox là gì?

Flexbox (Flexible Box):

- Là mô hình bố cục giúp **sắp xếp phần tử theo 1 chiều** (ngang / dọc)
- Giúp căn chỉnh – phân bố – co giãn các phần tử trong container

Dùng khi:

- ✓ Cần xếp các item theo hàng (menu, navbar)
- ✓ Cần xếp các item theo cột (sidebar, danh sách)
- ✓ Muốn các item **co giãn hợp lý** theo kích thước màn hình

Flexbox = Giải pháp cho layout nhỏ – gọn – dễ kiểm soát!



Các thuộc tính của Flex Container

Để sử dụng Flexbox: thêm `display: flex` vào phần tử cha (container)

Các thuộc tính quan trọng:

<code>flex-direction:</code>	Chiều sắp xếp: row (ngang), column (dọc)
<code>justify-content:</code>	Căn **theo trục chính**: start, center, space-between
<code>align-items:</code>	Căn **theo trục phụ**: center, stretch, flex-start...

Mẹo:

- ▶ Muốn menu nằm ngang → `flex-direction: row`
- ▶ Muốn căn giữa toàn bộ → kết hợp `justify-content` + `align-items`



Thuộc tính của Flex Item

Các thuộc tính quan trọng:

flex-grow gian)	Mở rộng thêm không? (tỉ lệ giãn khi còn dư không
flex-shrink	Có co lại không? (thu nhỏ khi không đủ chỗ)
flex-basis	Kích thước ban đầu (trước khi co giãn)



Thực hành: Flexbox Gallery

Mục tiêu:

- Tạo 1 gallery ảnh đơn giản nhưng responsive bằng Flexbox

Hướng dẫn:

✓ Container:

- `display: flex;`
- `flex-wrap: wrap;`
- `justify-content: center;`

✓ Ảnh:

```
img {  
  width: 100%;  
  max-width: 200px;  
  margin: 10px;  
}
```

Kết quả mong muốn:

- Ảnh tự co giãn, xuống dòng nếu không đủ chỗ, luôn được căn giữa



CSS Grid là gì?

CSS Grid = Hệ thống bố cục ****2 chiều**** (hàng + cột)

- ✓ Dễ dàng chia layout thành các khu rõ ràng
- ✓ Linh hoạt cho layout phức tạp: dashboard, báo, app admin
- ✓ Tự tính toán kích thước và sắp xếp phần tử

Dùng khi:

- Cần chia vùng trang (header, sidebar, content, footer)
- Thiết kế lưới ảnh, bài viết, danh sách sản phẩm
- Muốn layout ổn định theo cả hàng + cột cùng lúc



Grid Container: Thuộc tính chính

Bắt đầu bằng: `display: grid;`

Thuộc tính quan trọng:

`grid-template-columns:` Tạo số và kích thước các ****cột****
`grid-template-rows:` Tạo số và kích thước các ****hàng****
`gap` hoặc `grid-gap:` Khoảng cách giữa các ô

Ví dụ:

```
container {  
  display: grid;  
  grid-template-columns: 1fr 1fr 1fr; /* 3 cột bằng nhau */  
  grid-template-rows: 100px 200px;  
  gap: 20px;  
}
```




Grid Item: Điều khiển vị trí từng phần tử

Các thuộc tính hay dùng:

`grid-column: start / end;` → chiếm bao nhiêu cột

`grid-row: start / end;` → chiếm bao nhiêu hàng

Ví dụ:

```
.item1 {  
  grid-column: 1 / 3; /* chiếm 2 cột */  
  grid-row: 1 / 2; /* chiếm 1 hàng */  
}
```

Gợi ý:

- ▶ Giống như “chỉ định tọa độ” cho từng khối trong lưới
- ▶ Dễ làm layout dạng bảng, card grid, hoặc dashboard