

## Kiến thức cần nhớ

1. CSS Grid là một kỹ thuật nâng cao **vô cùng mạnh mẽ giúp xây dựng layout 2 chiều** và phức tạp một cách nhanh chóng.
2. CSS Grid có tư duy phân bổ layout thành một mạng lưới bao gồm các đường (**line**) ngang và dọc đan vào nhau tạo thành các ô (**cell**). Khoảng cách giữa các ô là **gap**. Các ô thẳng hàng tạo thành cột hoặc hàng được gọi là rãnh (**track**).
3. Nắm vững tư duy và các thuộc tính của CSS Grid giúp bạn hoàn thiện kỹ năng xây dựng layout khó, là **tiền đề cho xử lý giao diện thích ứng mọi thiết bị**.
4. Các thuộc tính thường gặp ở grid container:
  - **display: grid;**: Khai báo container là grid.
  - **grid-template-columns, grid-template-rows**: Xác định số lượng và kích thước của cột và hàng.
  - **grid-template-areas**: Xác định layout bằng cách đặt tên cho các vùng lưới.
  - **grid-template**: Kết hợp **grid-template-rows, grid-template-columns** và **grid-template-areas**.
  - **gap, grid-gap**: Khoảng cách giữa các phần tử trong lưới.
  - **justify-content, align-content, justify-items, align-items, place-items**: Căn chỉnh các phần tử trong lưới.
5. Các thuộc tính thường gặp ở grid items:
  - **grid-column-start, grid-column-end, grid-row-start, grid-row-end**: Xác định vị trí bắt đầu và kết thúc của phần tử con theo cột và hàng.
  - **grid-column, grid-row**: Shorthand syntax của **grid-column-start, grid-column-end, grid-row-start, grid-row-end**.
  - **grid-area**: Xác định vùng lưới mà phần tử con sẽ chiếm.
  - **justify-self, align-self, place-self**: Căn chỉnh phần tử con trong ô lưới.
  - **order**: Xác định thứ tự hiển thị của phần tử con.
6. Các hàm và đơn vị hỗ trợ:
  - **minmax(), repeat()**: Hàm hỗ trợ xác định kích thước của phần tử con.
  - **fr**: Đơn vị đo tương đối, phân chia không gian còn lại của lưới.
  - **auto-fill, auto-fit**: Tự động tạo thêm cột trên cùng hàng.
  - **max-content(), min-content()**: Xác định kích thước của phần tử dựa trên nội dung bên trong.