

# CSS Grid Made Simple (Visual Layout Guide)

Build complex layouts without confusion

by Vasy.dev

INTRO

CSS Grid is the most powerful layout system in modern CSS.

It allows you to create complex, responsive layouts with clean and readable code.

This guide simplifies Grid so you can use it instantly in real projects.

## 1. ACTIVATE GRID

```
display: grid;
```

Turns element into grid container

## 2. DEFINE COLUMNS

```
grid-template-columns: 1fr 1fr;
```

Creates 2 equal columns

## 3. RESPONSIVE COLUMNS (POWERFUL)

```
grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
```

Automatically adjusts columns based on screen size

THIS IS GOLD

## 4. GAP (SPACING)

```
gap: 1rem;
```

Clean spacing between grid items

## 5. ROWS

grid-template-rows: auto auto;

Define row heights

## 6. SPAN COLUMNS

grid-column: span 2;

Item takes 2 columns

## 7. SPAN ROWS

grid-row: span 2;

Item takes 2 rows

## 8. ALIGN ITEMS

justify-items: center;

align-items: center;

Controls alignment inside grid

## 9. PLACE ITEMS (SHORTCUT)

place-items: center;

Same as align + justify

## 10. REAL EXAMPLE (CARD GRID)

```
.container {  
  display: grid;  
  grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));  
  gap: 1rem;  
}
```

Perfect for:

- cards
- portfolios
- dashboards

Use Flexbox for 1D layouts (row OR column)

Use Grid for 2D layouts (rows AND columns)

# COMMON MISTAKE

Don't mix Grid and Flex randomly.

Use Grid for layout structure

Use Flexbox for internal alignment

Next guides:

→ UI Design Mistakes (very important)

→ Performance Optimization Guide

Want more frontend guides like this?

I share practical tools, cheat sheets and real-world techniques.

Visit: <https://vasy-dev.netlify.app>

Contact: [milanjakub9@gmail.com](mailto:milanjakub9@gmail.com)

Build faster. Work smarter.