

Day 8

Today's session introduced CSS Grid, a two-dimensional layout system that allows for more precise control over both rows and columns. While Flexbox is great for one-dimensional layouts (either row or column), Grid is ideal for designing full page sections like galleries, grids of features, and structured content layouts.

1. What is CSS Grid?

- A layout method for arranging items into rows and columns.
- Offers explicit control over placement and spacing.
- Great for cards, image galleries, page sections, and dashboards.

2. Creating a Responsive Grid Section:

Sample HTML:

```
<div class="features-grid">  
  <div class="feature">Feature A</div>  
  <div class="feature">Feature B</div>  
  <div class="feature">Feature C</div>  
  <div class="feature">Feature D</div>  
</div>
```

CSS:

```
.features-grid {  
  display: grid;  
  grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));  
  gap: 20px;  
  padding: 30px;  
}  
  
.feature {  
  background-color: #ffffff;  
  border-radius: 8px;
```

```
padding: 20px;  
box-shadow: 0 0 8px rgba(0,0,0,0.1);  
}
```

Summary:

CSS Grid gave me more control and flexibility than I expected. It complements Flexbox beautifully. Where Flexbox helped align items, Grid helped me structure entire sections. I now understand how professional websites create clean, symmetrical layouts that adjust perfectly to different screen sizes.