What is CSS Grid?

CSS Grid is a layout system in CSS that allows developers to design web pages by organizing content into rows and columns.
It is a two-dimensional system, meaning it handles layouts in both rows and columns simultaneously.

How to Enable CSS Grid?

To use the grid layout, set the `display` property of a container to `grid` .

```
.container {
    display: grid;
}
```

Key Concepts in CSS Grid

1. **Grid Container**:
   The element with `display: grid` or `display: inline-grid`. It is the parent element of the grid items.

2. **Grid Items**:
   The direct child elements of the grid container. These items are automatically placed into the grid.

CSS Grid Container Properties

1. **grid-template-columns**
   Defines the number and size of columns in the grid.
   Example:

   ```
   grid-template-columns: 100px 200px 1fr;
   ```

2. **grid-template-rows**
   Defines the number and size of rows in the grid.
   Example:

   ```
   grid-template-rows: 50px auto 50px;
   ```

3. **gap** (or `row-gap` and `column-gap`)
   Specifies the spacing between grid rows and columns.
   Example:

   ```
   gap: 10px;
   row-gap: 20px;
   column-gap: 15px;
   ```

4. **grid-auto-rows**
   Defines the size of rows that are automatically generated.
   Example:

   ```
   grid-auto-rows: 100px;
   ```

5. **justify-content**
   Aligns the entire grid horizontally within the container.
   Example:

   ```
   justify-content: space-between;
   ```

6. **align-content**
   Aligns the entire grid vertically within the container.
   Example:

   ```
   align-content: space-around;
   ```

CSS Grid Item Properties

1. **grid-column**
   Specifies the starting and ending column for a grid item.
   Example:

   ```
   grid-column: span 2;
   ```

2. **grid-row**
   Specifies the starting and ending row for a grid item.
   Example:

   ```
   grid-row: span 3;
   ```

Difference Between CSS Grid and Flexbox

Aspect	CSS Grid	Flexbox
Layout Type	Two-dimensional layout system (rows and columns).	One-dimensional layout system (either rows or columns).
Primary Purpose	Best for creating complex layouts involving both rows and columns.	Best for aligning and distributing items in a single row or column.
Child Alignment	Allows placement and alignment of items both vertically and horizontally.	Aligns items along the main or cross axis in one dimension.
Container vs. Items	Focuses on the entire grid container structure.	Focuses on aligning individual items.
Content Reordering	Items can span multiple rows and columns (`grid-row`, `grid-column`).	Reordering works within a single row or column (`order`).
Syntax Complexity	More complex and verbose syntax for setting up grids.	Simpler syntax for arranging items in one dimension.
Use Cases	Ideal for page layouts, dashboards, and complex UI designs.	Ideal for navbars, toolbars, and linear alignments.