CSS Grid Display Properties

1Enabling Grid

```
.container {
    display: grid; /* Block-level grid */
    display: inline-grid; /* Inline-level grid */
}
```

2Defining Rows & Columns

```
.container {
    grid-template-columns: 100px 200px auto;
    grid-template-rows: 150px 100px;
}
```

Use `repeat()` to simplify:

```
.container {
    grid-template-columns: repeat(3, 100px);
    grid-template-rows: repeat(2, 150px);
}
```

3Grid Item Placement

```
.item {
    grid-column: 1 / 3; /* Spans column 1 to 3 */
    grid-row: 1 / 2; /* Spans row 1 */
}
```

Using `span`:

```
.item {
    grid-column: span 2;
    grid-row: span 3;
}
```

4Grid Gaps (Spacing)

```
.container {
    gap: 10px; /* Space between rows & columns */
}
```

Separate gaps:

```
.container {
    row-gap: 20px;
    column-gap: 10px;
}
```

5Auto Placement

```
.container {
    grid-auto-flow: row; /* Default */
    grid-auto-flow: column; /* Places items column-wise */
}
```

6Justifying & Aligning Grid Items

✓ Aligning Grid as a Whole:

```
.container {
    justify-content: center; /* Horizontal */
    align-content: space-between; /* Vertical */
}
```

✓ Aligning Items Inside Cells:

```
.container {
    justify-items: center; /* Horizontally */
    align-items: stretch; /* Vertically */
}
```

7Grid Areas (Named Layouts)

```
.container {
    grid-template-areas:
        "header header"
        "sidebar main"
        "footer footer";
    grid-template-columns: 150px auto;
    grid-template-rows: 50px auto 50px;
}

.header { grid-area: header; }
.sidebar { grid-area: sidebar; }
.main { grid-area: main; }
.footer { grid-area: footer; }
```

8.Responsive Grid with `auto-fit` & `auto-fill`

Auto-fit: Shrinks items to fit available space.

```
.container {
    grid-template-columns: repeat(auto-fit, minmax(100px, 1fr));
}
```

Auto-fill: Fills available space with invisible columns.

```
.container {
    grid-template-columns: repeat(auto-fill, minmax(150px, 1fr));
}
```
