Grid Property

The display: grid property in CSS is used to create a grid container for layout. It allows you to define rows and columns, and place items within the grid using specific properties. Below are some of the main properties and how they work:

Container Properties

These properties are applied to the parent element with display: grid.

1. **grid-template-columns**
   Defines the columns of the grid.

Example:

```
grid-template-columns: 100px 200px auto;
grid-template-columns: 1fr 2fr 1fr;
grid-template-columns: repeat(3, 1fr);
```

2. **grid-template-rows**
   Defines the rows of the grid.

Example:

```
grid-template-rows: 50px auto 100px;
grid-template-rows: repeat(2, 1fr);
```

3. **gap** (or grid-gap)
   Specifies the space between rows and columns.

Example:

```
gap: 10px; /* Applies to both rows and columns */
row-gap: 10px;
column-gap: 15px;
```

4. **justify-items**
   Aligns items horizontally within their grid area.
   - Values: start | end | center | stretch

Example:
justify-items: center;

5. **align-items**
   Aligns items vertically within their grid area.
   - Values: start | end | center | stretch

Example:
align-items: stretch;

6. **justify-content**
   Aligns the entire grid horizontally within the container.
   - Values: start | end | center | stretch | space-between | space-around | space-evenly

Example:
justify-content: space-between;

7. **align-content**
   Aligns the entire grid vertically within the container.
   - Values: start | end | center | stretch | space-between | space-around | space-evenly

Example:
align-content: center;

8. **grid-auto-rows / grid-auto-columns**
   Specifies the size of implicitly created rows or columns.

Example:
grid-auto-rows: 100px;
grid-auto-columns: 50px;

9. **grid-auto-flow**
   Specifies how auto-placed items are added to the grid.
   - Values: row | column | dense | row dense | column dense

Example:
grid-auto-flow: row dense;

Item Properties

These properties are applied to the grid items.

1. **grid-column**
   Specifies which columns an item should occupy.

Example:

```
grid-column: 1 / 3; /* From column 1 to column 3 */
grid-column: span 2; /* Spans 2 columns */
```

2. **grid-row**
   Specifies which rows an item should occupy.

Example:

```
grid-row: 2 / 4; /* From row 2 to row 4 */
grid-row: span 3; /* Spans 3 rows */
```

3. **justify-self**
   Aligns an item horizontally within its grid area.
   - Values: start | end | center | stretch

Example:

```
justify-self: end;
```

4. **align-self**
   Aligns an item vertically within its grid area.
   - Values: start | end | center | stretch
     align-self: start;

Display Values for Table Layout

1. display: table

   The element behaves like a <table>

2. display: table-row

   The element behaves like a <tr> (table row).

3. display: table-cell

   The element behaves like a <td> (table cell).

4. display: table-column

   The element behaves like a <col> (table column, used for column styling).

5. display: table-column-group

   The element behaves like a <colgroup>.

6. display: table-row-group

   The element behaves like a <tbody>.

7. display: table-header-group

   The element behaves like a <thead>.

8. display: table-footer-group

   The element behaves like a <tfoot>.

9. display: table-caption

   The element behaves like a <caption>.