

CSS Grid Layout

The `display: grid` CSS property is used to turn an element into a grid container. This allows you to layout child elements (known as grid items) in a two-dimensional grid. You can define rows and columns, and place items within this grid in a precise manner.

Basic Concept

When you apply `display: grid` to an element, its children become grid items. You can then use various grid properties to define how these items should be placed and sized within the grid.

Example Usage

HTML:

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS Grid Example</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <div class="grid-container">
    <div class="grid-item">1</div>
    <div class="grid-item">2</div>
    <div class="grid-item">3</div>
    <div class="grid-item">4</div>
    <div class="grid-item">5</div>
    <div class="grid-item">6</div>
  </div>
</body>
</html>
```

CSS:

```
.grid-container {
display: grid;
grid-template-columns: repeat(3, 1fr); /* Creates 3 equal columns */
grid-template-rows: repeat(2, 100px); /* Creates 2 rows, each 100px tall */
gap: 10px; /* Space between grid items */
}
.grid-item {
background-color: lightblue;
border: 1px solid #ccc;
display: flex;
justify-content: center;
align-items: center;
font-size: 20px;
}
```

Explanation

1. `display: grid`: This makes the container a grid container.
2. `grid-template-columns`: Defines the number and size of columns in the grid. In this case, `repeat(3, 1fr)` creates three columns, each taking up an equal fraction of the available space.
3. `grid-template-rows`: Defines the number and size of rows in the grid. Here, `repeat(2, 100px)` creates two rows, each with a fixed height of 100 pixels.
4. `gap`: Sets the space between the grid items. The value `10px` means there will be a 10-pixel gap between both rows and columns.
5. `.grid-item`: Styles for the grid items. The `display: flex` property is used here to center the content within each grid item.

More Advanced Grid Properties

- `grid-column` and `grid-row`: Define how many columns or rows an item should span.
- `justify-items` and `align-items`: Control the alignment of grid items along the row and column axes.
- `grid-auto-rows` and `grid-auto-columns`: Define the size of implicitly created rows and columns.

Example with Advanced Properties

CSS:

```
.grid-container {  
  display: grid;  
  grid-template-columns: 1fr 2fr 1fr;  
  grid-template-rows: auto;  
  gap: 20px;  
  justify-items: center;  
  align-items: start;  
}  
.grid-item:nth-child(1) {  
  grid-column: span 2; /* This item will span 2 columns */  
  grid-row: span 1; /* This item will span 1 row */  
}
```

In this example:

- The first item (.grid-item:nth-child(1)) spans across two columns.
- The grid has three columns with different widths (1fr 2fr 1fr).
- Items are centered along the row axis (justify-items: center) and aligned to the start of the column axis (align-items: start).

CSS Grid is a powerful layout system that provides great flexibility for designing responsive and complex web layouts.