Chapter 15
GRID LAYOUT

- In order to use the **Grid Layout**, the **display** property must be set to **grid**.

- The **grid-auto-flow** property controls the placement of auto-flowed items in the grid. We will focus on two of its possible settings: **row and column**. The default **row** setting places the items in rows. The **column** setting does so in a single column / multiple columns.

- The **grid-auto-columns** property determines the width of the columns. The size can be specified in terms of *<length>, <percentage>, <flex> [fr] or <track>.*

- When specified in terms of *<length>,* then if a single value is specified, that value determines the width of all the columns. If two values (**a** and **b**) are specified and there are more than 2 columns (say a total of 6 columns), then the length of the first column will be **a**, the length of the second column **b,** the length of the third column **a**, the length of the fourth column **b** and so on.

- The example immediately above can also be achieved by using the **repeat** function as **repeat(2, 1fr 2fr)**.

- When specified in terms of *fr (fraction units),* (Eg - 1fr 2fr 2fr), then the length of the second and the third columns will be twice that of the first column, the length of the fifth and the sixth column will be twice that of the third while the length of the first column will be equal to that of the third.

- The **grid-auto-rows** property is used to decide the height of rows in a **Grid Layout**. It can be specified in various ways. We will cover how to specify the height in **px** and also as functions.

- When the height is specified as a single value in **px**, then it applies to all the rows in the grid.

- When the height is specified as **minmax(200px, auto)**, then the minimum height of each row will be 200px. However, if more space is available to some of the rows, then they can grow larger than 200px.

- The **gap** property is used to put a gap between rows and columns. It takes 2 values, one for gaps between rows and the other for gaps between columns.

- The **grid-column-start** and **grid-column-end** properties define the line at which an item should start and the line at which it should end respectively, along the columns, likewise for **grid-row-start** and **grid-row-end** along the rows.

- The **grid-row** and **grid-column** properties can take the start and end values as **start_value / end_value** for the line at which the item should start and the line at which the item should end along the row and column respectively.

- A **grid item** can also be a **grid container** simultaneously. The **align-content** property determines the position of a grid item along the row while the **justify-content** property determines the position of a grid item along the column. The settings of these properties are the same as those used on the **Flexbox Model**.

- The **align-content** and **justify-content** settings are taken in that order by the **place-content** property.

- The gaps between the rows and the columns are called **gutters**.

- *The grid template example shown in the video is described below: There are 9 items in the grid. The grid-auto-rows is set to 75px auto 75px. The grid-template-columns is set to repeat(9, 1fr). The grid-template-areas is set to "hd hd hd hd hd hd hd hd hd" in the 1st row, "mn mn mn mn mn mn mn sb sb" in the 2nd row and "ft ft ft ft ft ft ft ft ft". Here "hd" stands for header, "mn" stands for main, "sb" stands for sidebar and "ft" stands for footer.*

- *The header, main, sidebar and footer sections are already defined in the HTML code.*

- *Next, we define the "hd", "mn", "sb" and "ft" in our CSS code.*

- *The ".el" class selector has background-color: rebeccapurple; color: white; display: grid; AND place-content: center;*

- *The ".header" class selector has grid-area: hd;*

- *The ".sb" class selector has grid-area: sb; AND background-color blue:*

- *The ".footer" class selector has grid-area; ft;*

- *The ".container" class selector has grid-area: mn; AND a lot of other settings in it.*

- The **cssgridgarden.com** is a website that can help you practise and understand grid.