* We can turn a container into grid by assigning **display: grid.**
* All direct children of the grid container automatically become grid items.
* The vertical lines of grid items are called **columns**.
* The horizontal lines of grid items are called **rows**.
* The spaces between each column/row are called **gaps**. We can define using **grid-gap**.
* **Gap:** A shorthand property for the **row-gap** and the **column-gap** properties. Used on *Grid container***.**

1. **column-gap** Specifies the gap between the columns
2. **row-gap** Specifies the gap between the grid rows

* **Grid**: A shorthand property for the **grid-template-rows, grid-template-columns, grid-template-areas, grid-auto-rows, grid-auto-columns,** andthe **grid-auto-flow** properties**.** Used on *Grid container*.
  1. **grid-template** A shorthand property for the **grid-template-rows, grid-template-columns** and **grid-areas** properties.
     + **grid-template-rows:** Specifies the size of the rows in a grid layout, and how many rows in a grid layout
     + **grid-template-columns:** Specifies the size of the columns, and how many columns in a grid layout
     + **grid-template-areas:** Specifies how to display columns and rows, using named grid items. Easier to understand visually.
  2. **grid-auto-rows**: Specifies a default row size.
  3. **grid-auto-columns**: Specifies a default column size.
  4. **grid-auto-flow** : Specifies how auto-placed items are inserted in the grid
* **grid-column**: A shorthand property for **the grid-column-start** andthe **grid-column-end** properties. We can specify how many columns to span by defining span here. Used on *grid items*.
  1. **grid-column-start**: Specifies where to start the grid item
  2. **grid-column-end:** Specifies where to end the grid item
* **grid-row**: A shorthand property for the **grid-row-start** and **the grid-row-end** properties. Used on *grid items*.
  1. **grid-row-start:** Specifies where to start the grid item
  2. **grid-row-end:** Specifies where to end the grid item
* **~~grid-gap~~**~~: A shorthand property for the~~ **~~grid-row-gap~~** ~~and~~ **~~grid-column-gap~~** ~~properties.~~ (**This is deprecated, use gap instead)**
  1. **~~grid-row-gap~~**~~: Specifies the size of the gap between rows~~
  2. **~~grid-column-gap~~**~~: Specifies the size of the gap between columns~~

W3Schools for more details on the properties.

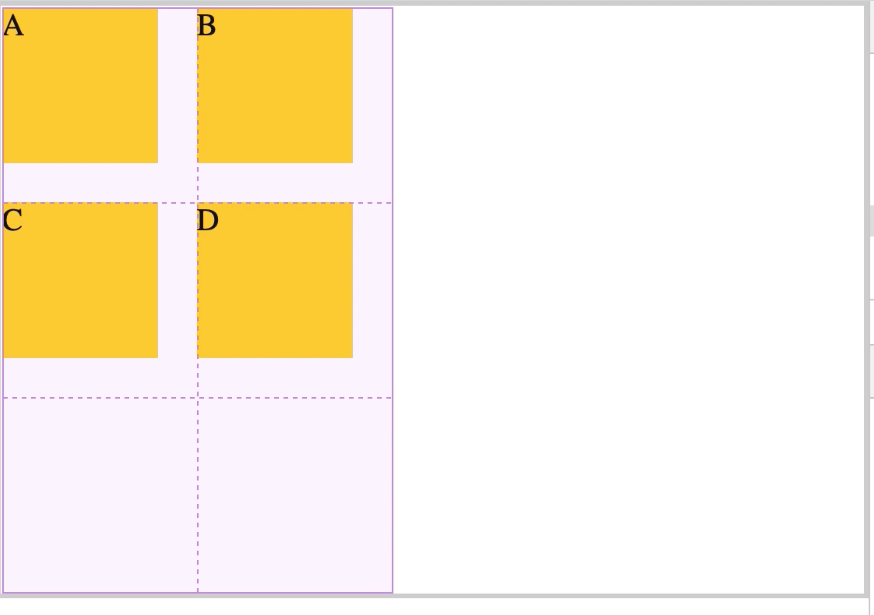
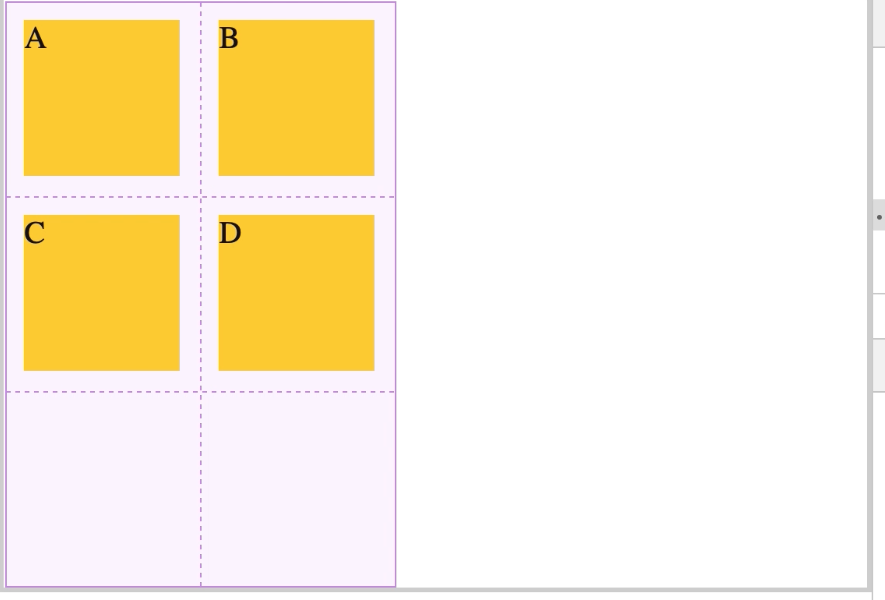
Chrome Dev tools -> Layout has some extremely useful debugging options.

* **Justify/align-items** VS **justify/align-content:** content is used to control the entire grid/container. But items are used to control individual child of the container.

Here is thread discussing grid-area vs grid-row/grid-column issues

<https://stackoverflow.com/questions/55213061/css-grid-grid-row-column-start-end-vs-grid-area-grid-template-area>

Grid without and with Justify-items and align-items. (note by default each element are stretched, so unless we specify the height and width of each element we wont get this resultant box)

Now if we use justify-content, this is the result.

