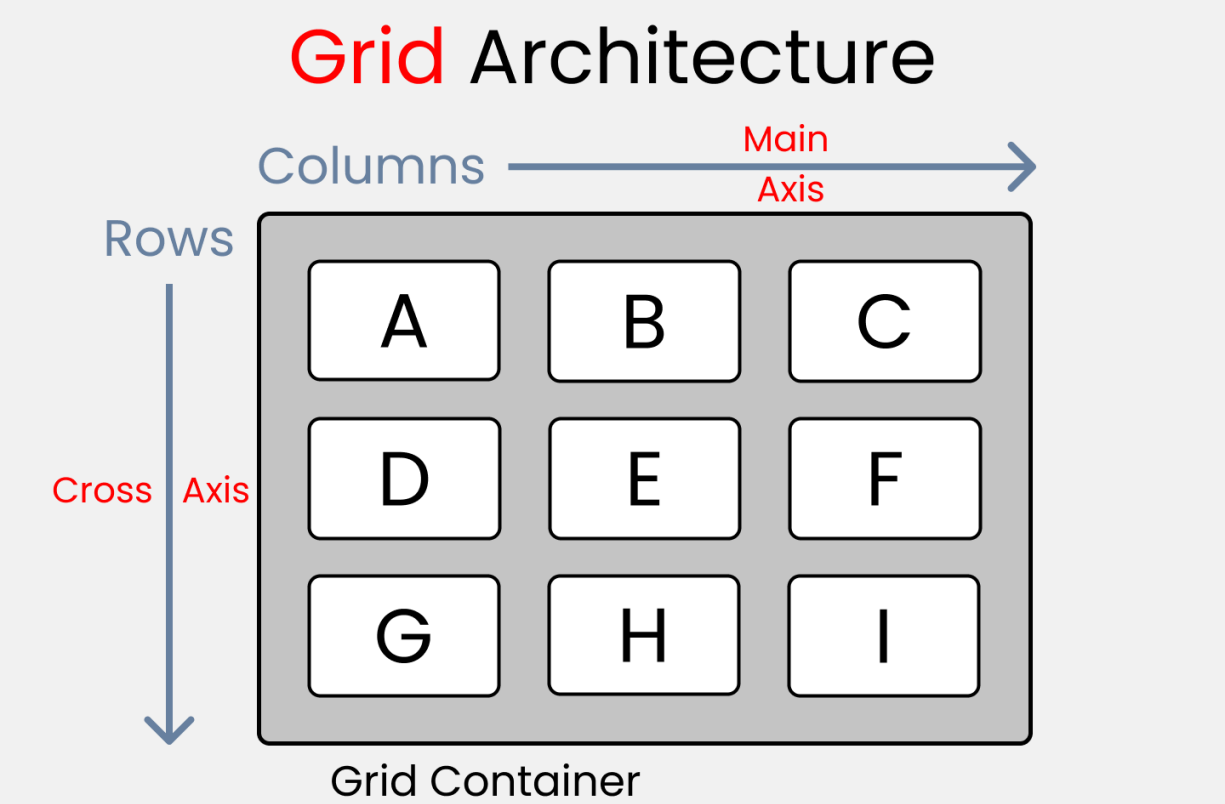
CSS Grid Properties

Grid Architecture

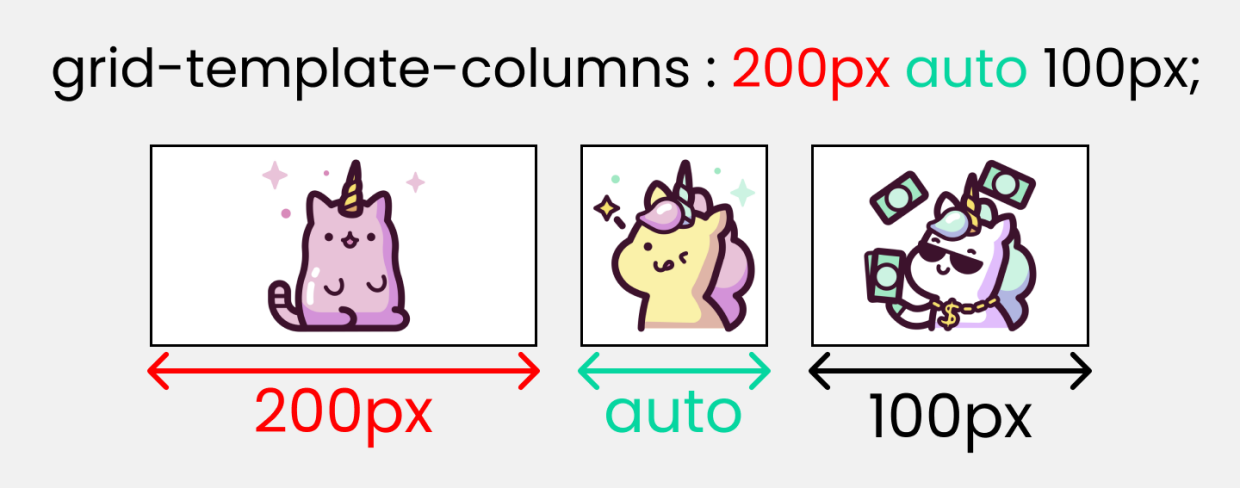
- So how does Grid architecture work? The Grid items [Contents] are distributed along the main axis and cross axis. Using various Grid properties, you can manipulate the items to create your website layouts.

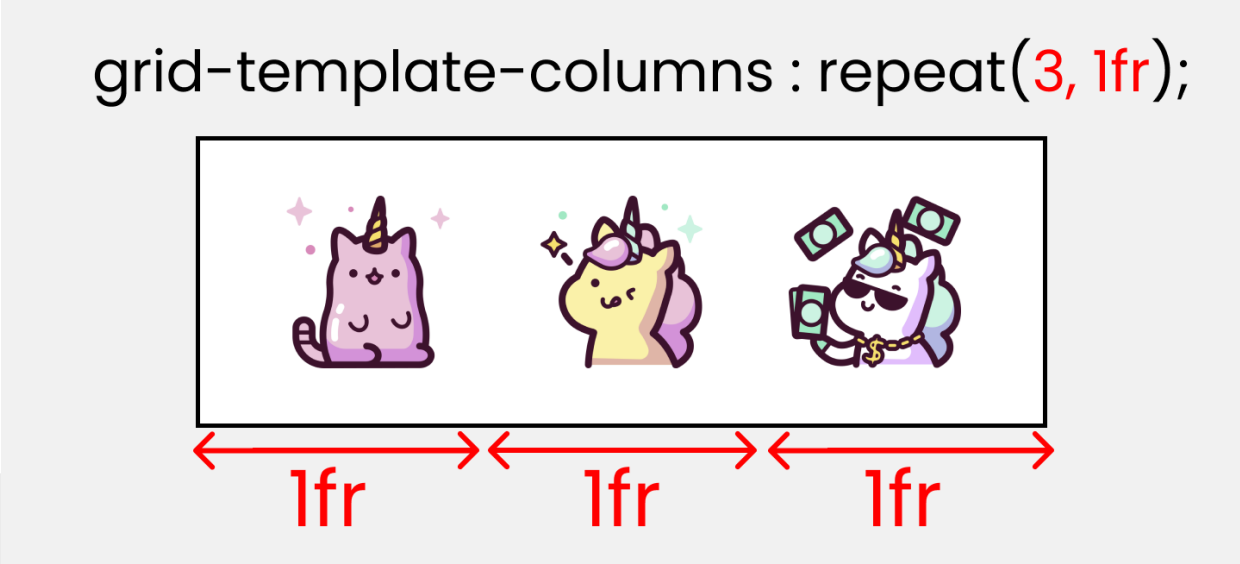


- By the way, you can join multiple rows and columns, just like in Excel software, which gives you more flexibility and options than Flexbox.

The grid-template-columns property:

- You use this property to define the number and width of columns. You can either individually set the width of each column, or set a uniform width for all columns using the repeat() function.

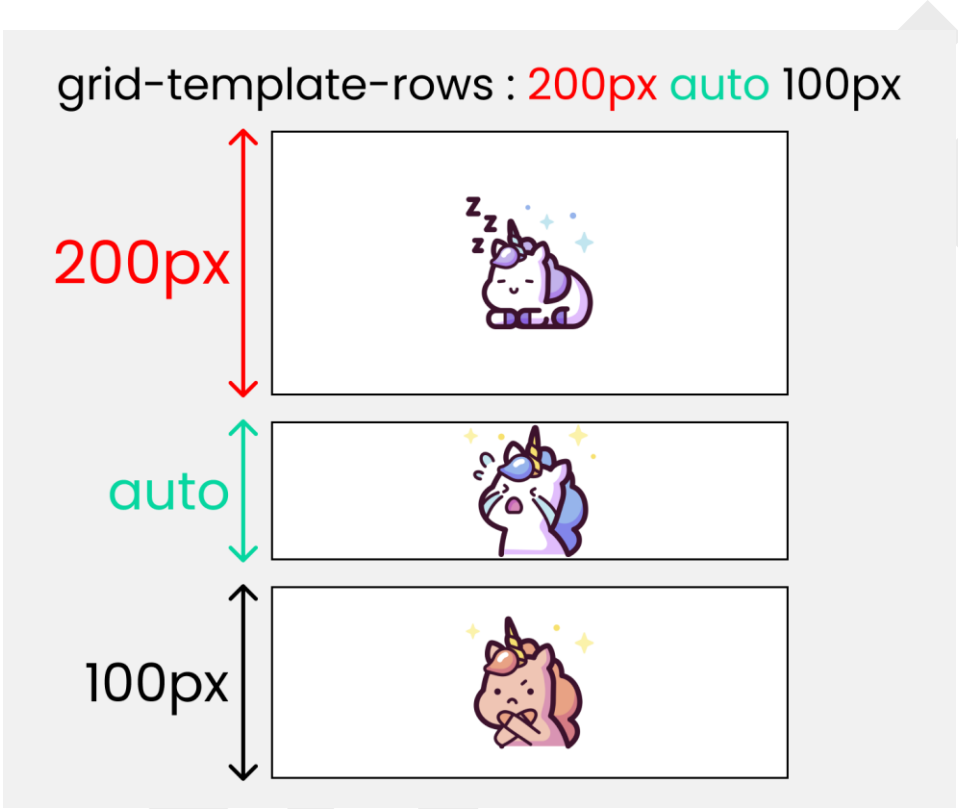


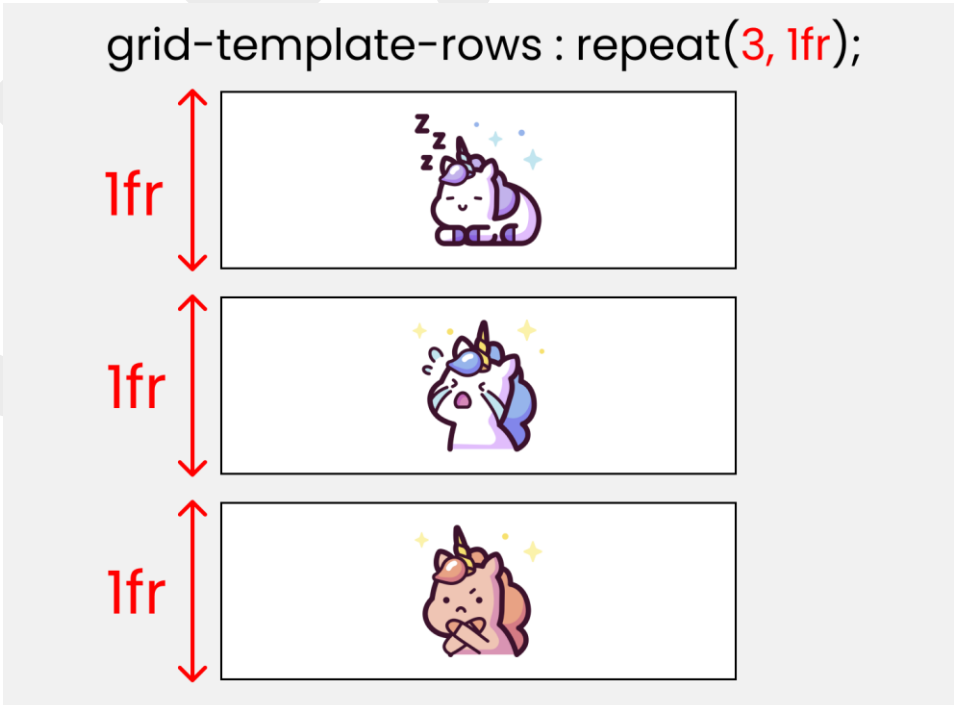


- **Note:**
  - The pixel values will be an exact measurement. The "auto" keyword will cover the available space.
  - If you use fr (fraction unit) as a unit of measurement, all the boxes will be equal in size.

The grid-template-rows property:

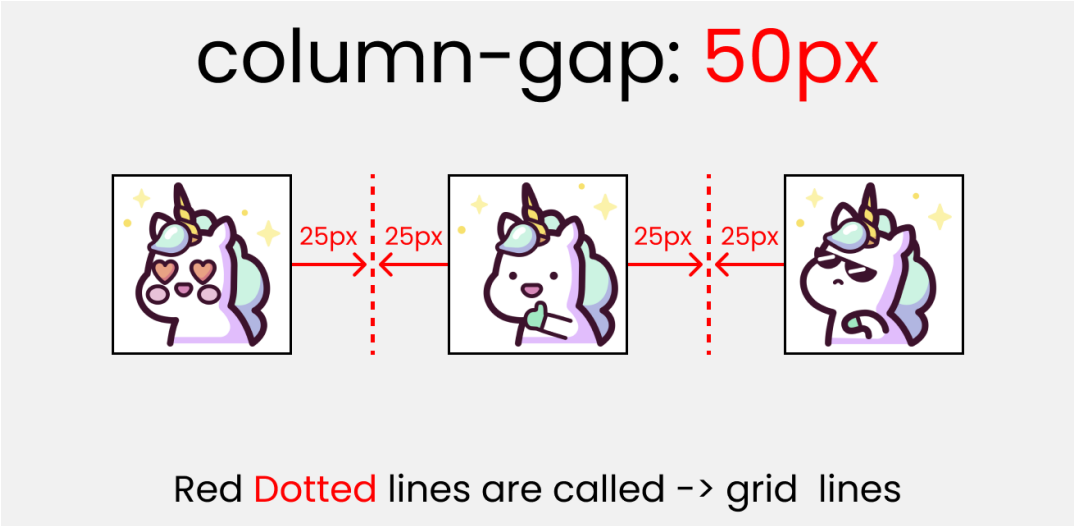
- You use this property to define the number and height of rows. You can either individually set the height of each row, or set a uniform height for all rows using the repeat() function.





The Column-gap Property :

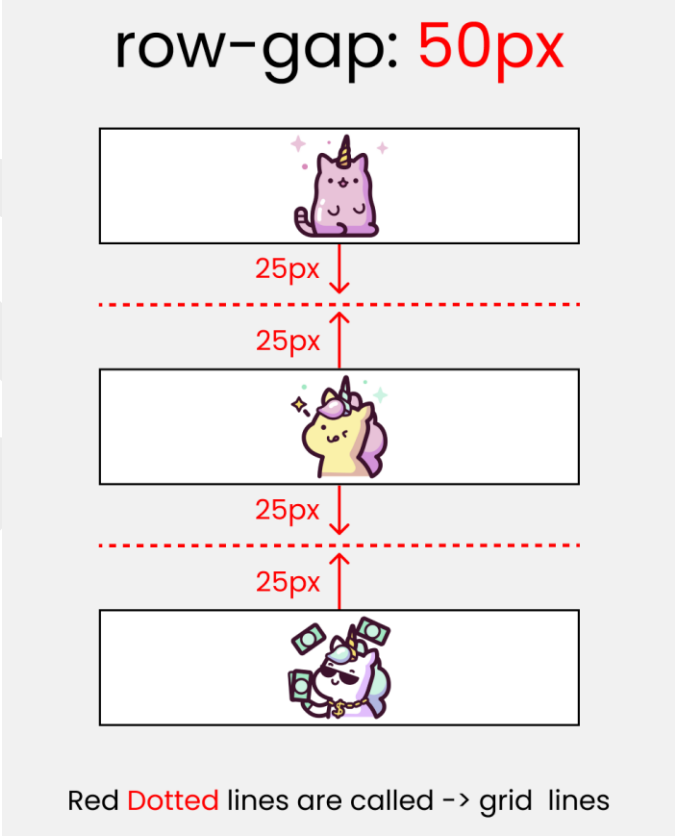
- You use this property to place a gap between **Columns** inside the grid



Note: column-gap works with grid-template-columns.

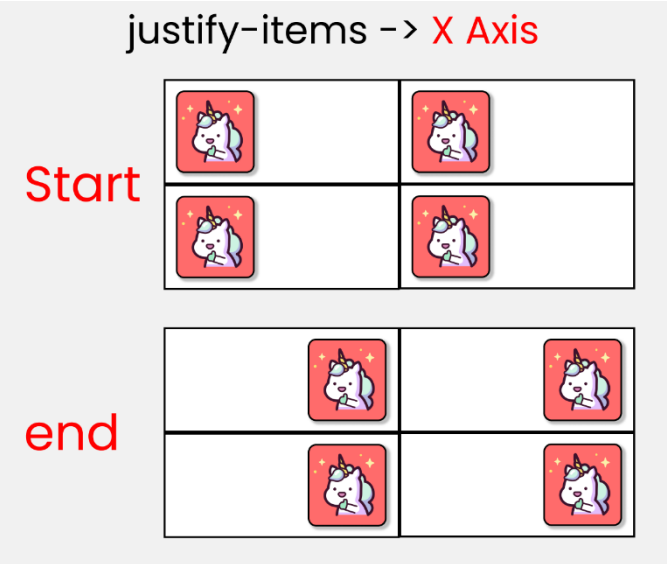
The row-gap Property :

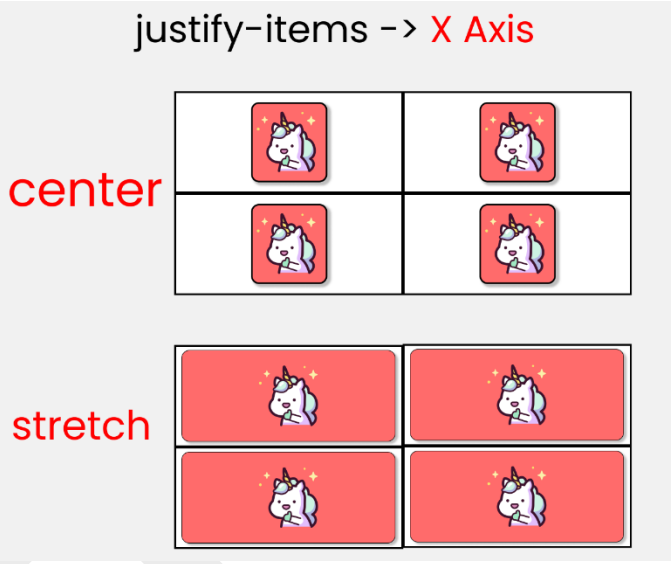
- You use this property to place a gap between **Rows** inside the grid



The justify-items property:

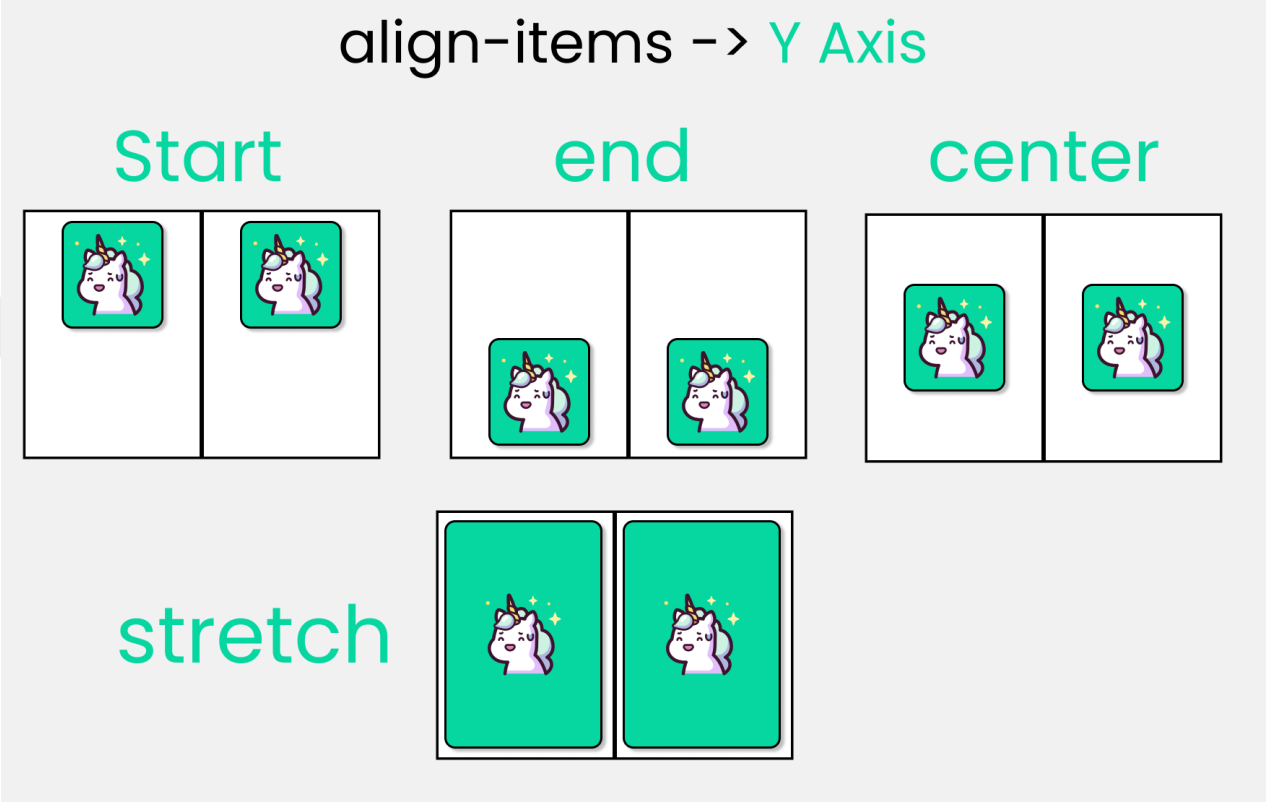
- You use this property to position grid-items (children) inside grid containers along the X-Axis [Main Axis]. The 4 values are 👇





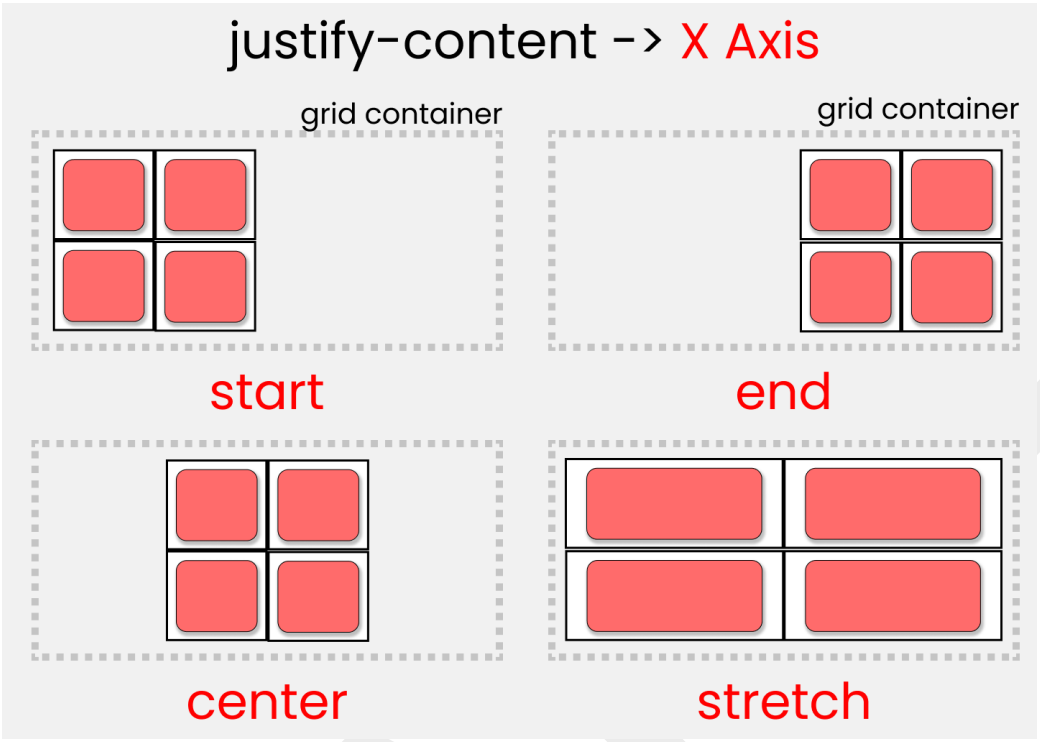
The align-items property:

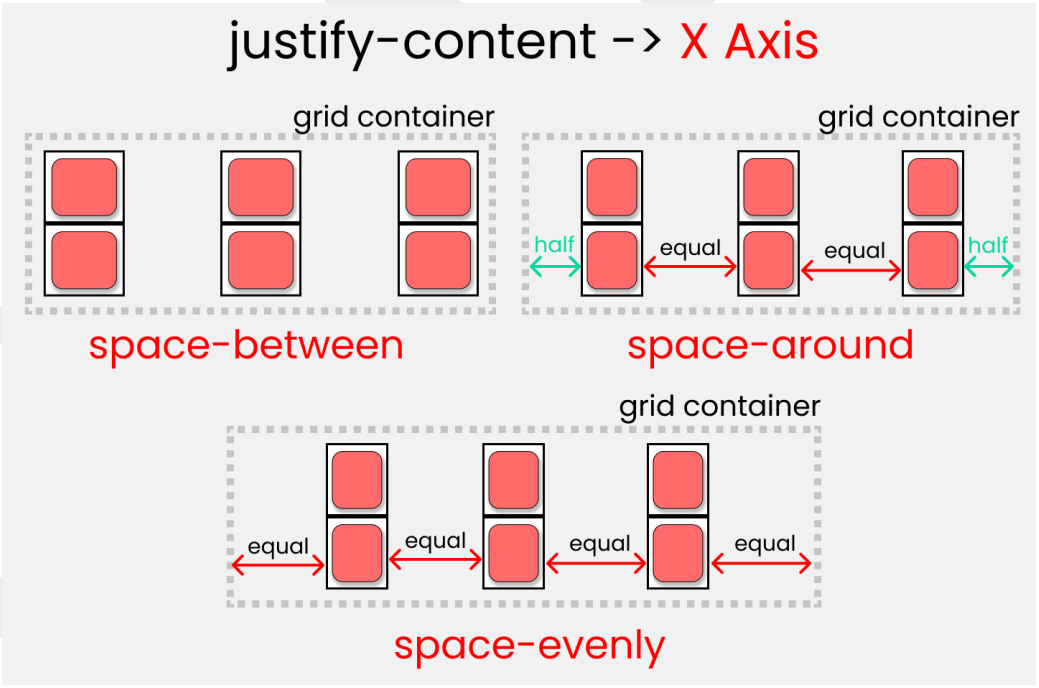
- You use this property to position grid-items (children) inside the grid container along the Y-Axis [Cross Axis]. The 4 values are 👇



The justify-content property:

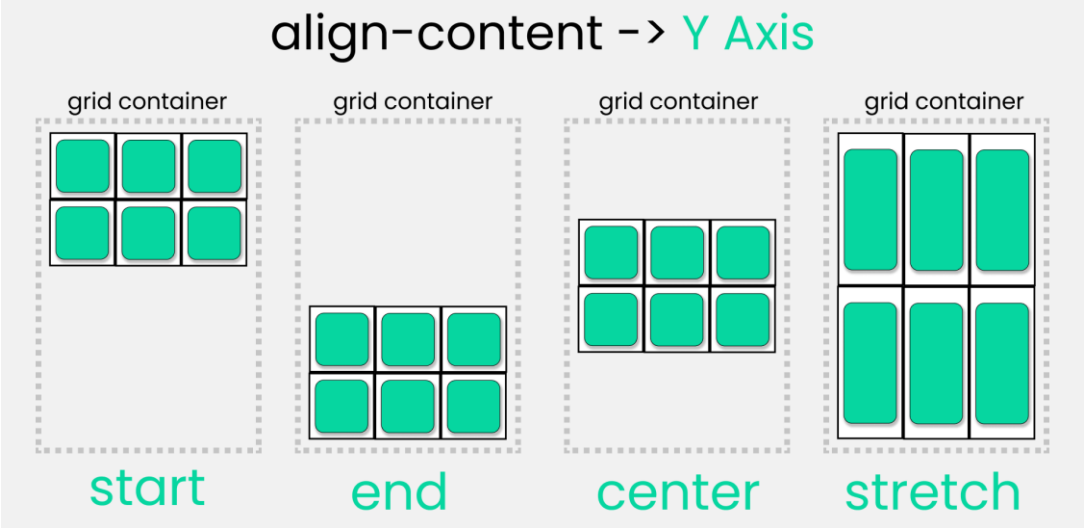
- You use this property to position your grid [Basically everything] inside the grid container along the X-Axis [Main Axis]. The 7 values are 👇

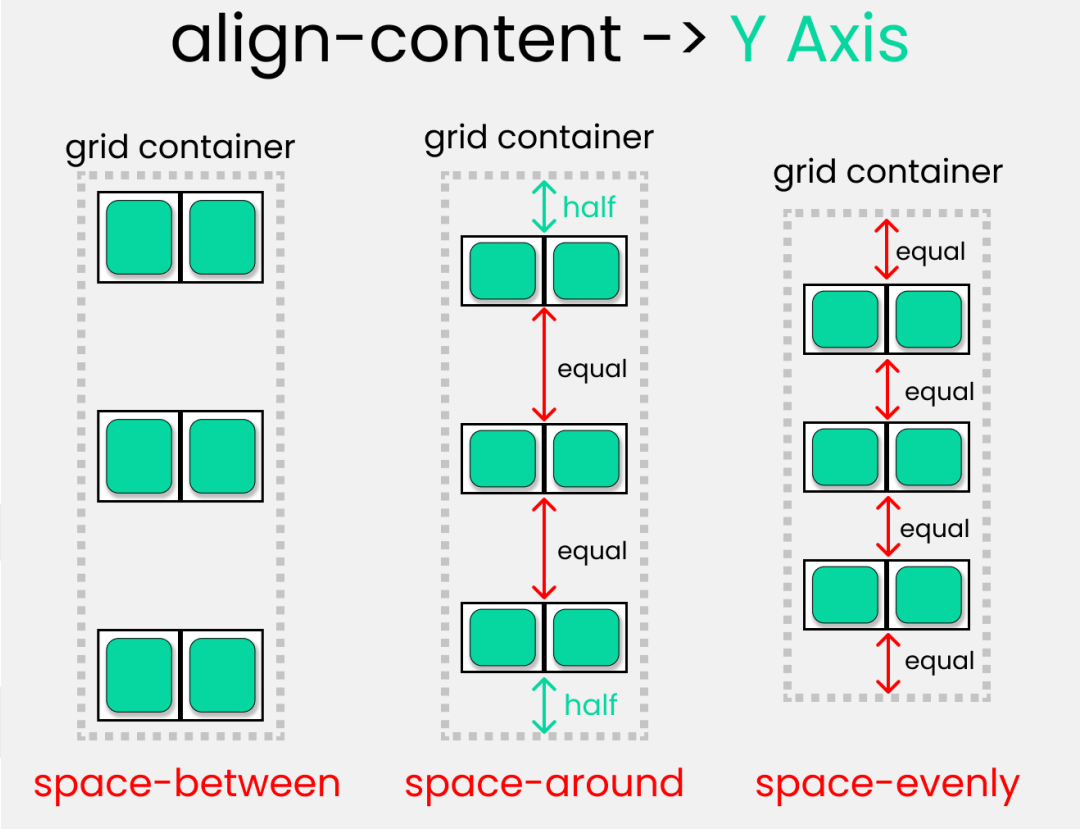




The align-content property:

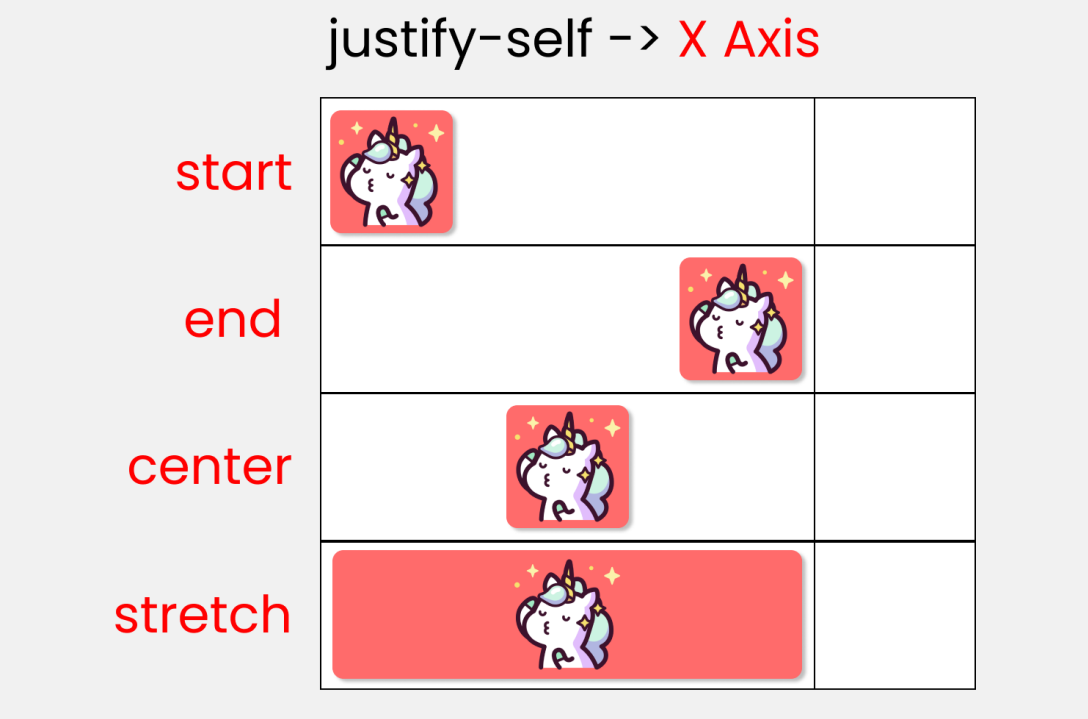
- You use this property to position our grid [Basically everything] inside the grid container along the Y-Axis [Cross Axis]. The 7 values are 👇





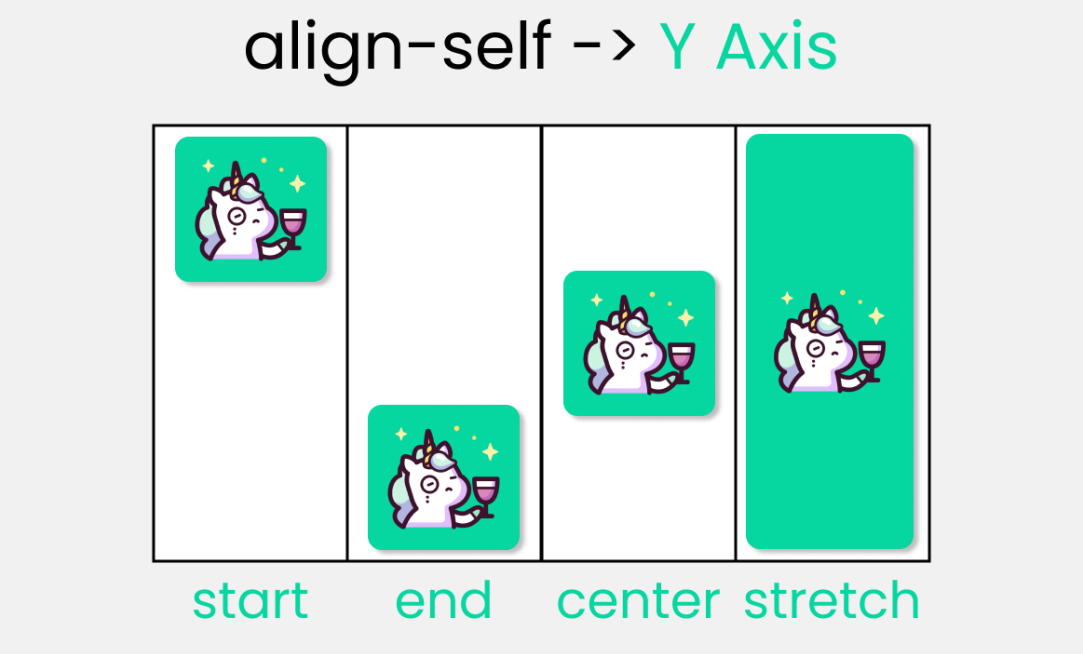
The justify-self property:

- You use this property to position 1 individual grid-item (child) inside a grid container along the X-Axis [Main Axis]. The 4 values are 👇



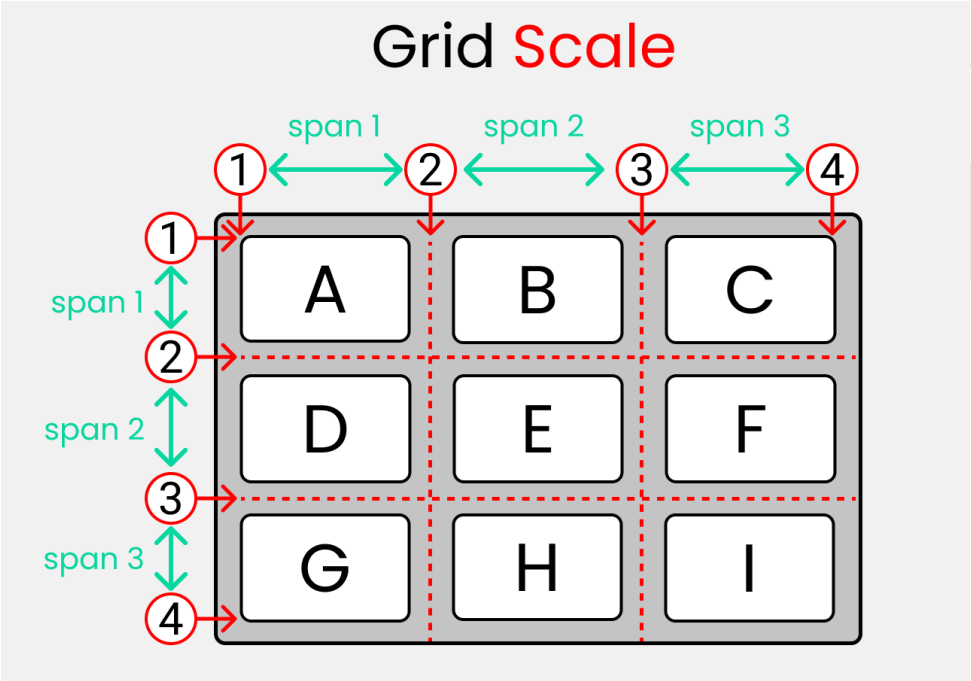
The align-self property:

- You use this property to position 1 individual grid-item (child) inside a grid container along the Y-Axis [Cross Axis]. The 4 values are 👇

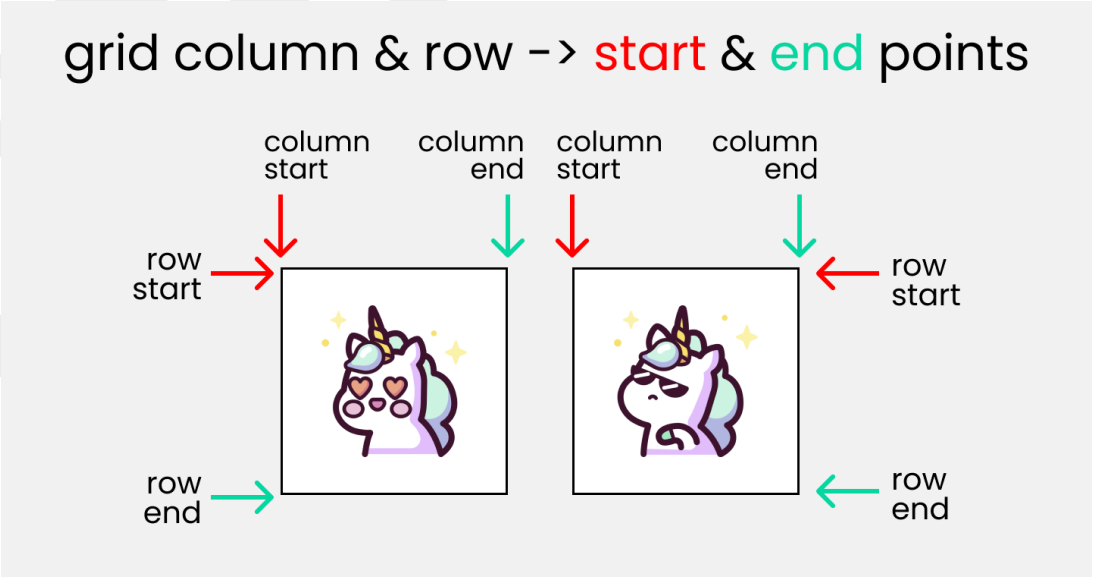


The CSS Grid Scale:

- I made this grid scale to demonstrate the calculations of how rows and columns are joined together. We can use any 1 of the 2 types of measurement:
  - **The digit [1,2,3,4, etc...]**
  - **The span keyword**



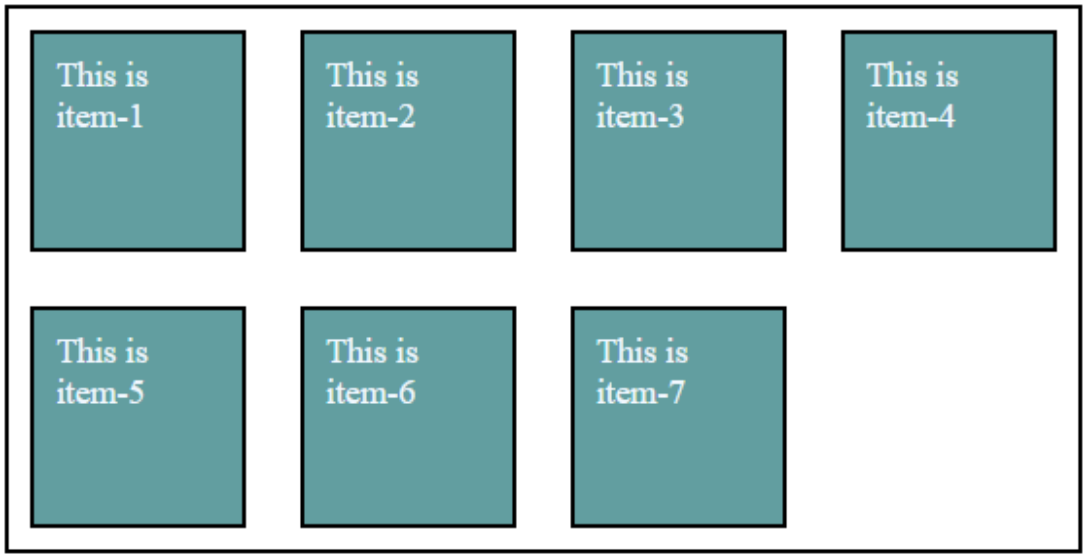
- The illustration below 👇 shows the start and end points of rows and columns of a single cell.



The grid-column: start/end properties :

- You use these two properties to join multiple COLUMNS together. It is a shorthand of 2 properties:
  - **grid-column-start**
  - **grid-column-end**

grid-template-columns: repeat(4, auto);

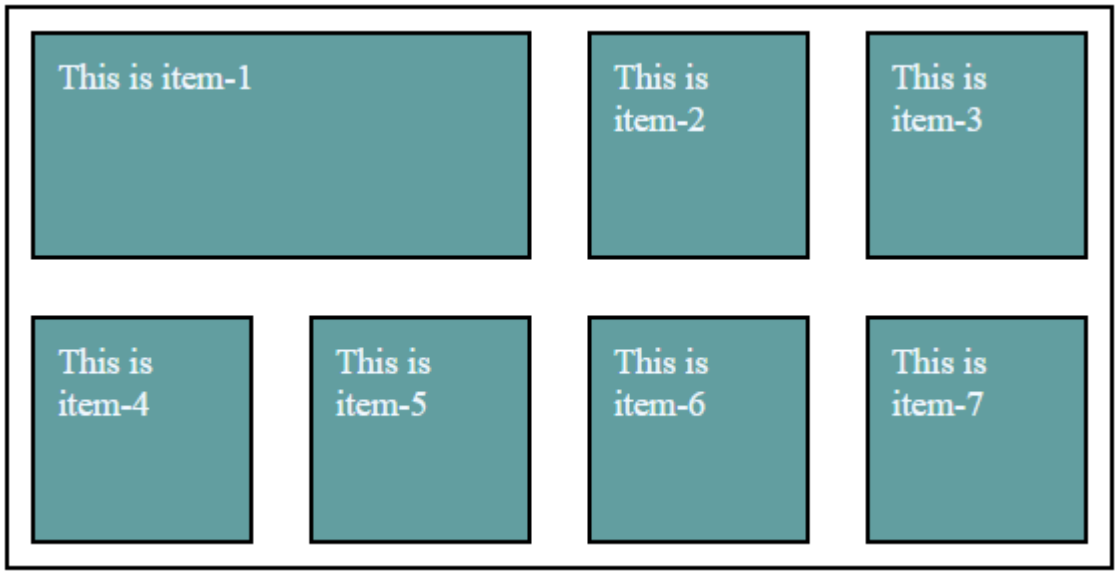


grid-columns-start: 1;

grid-columns-end: 2;

OR

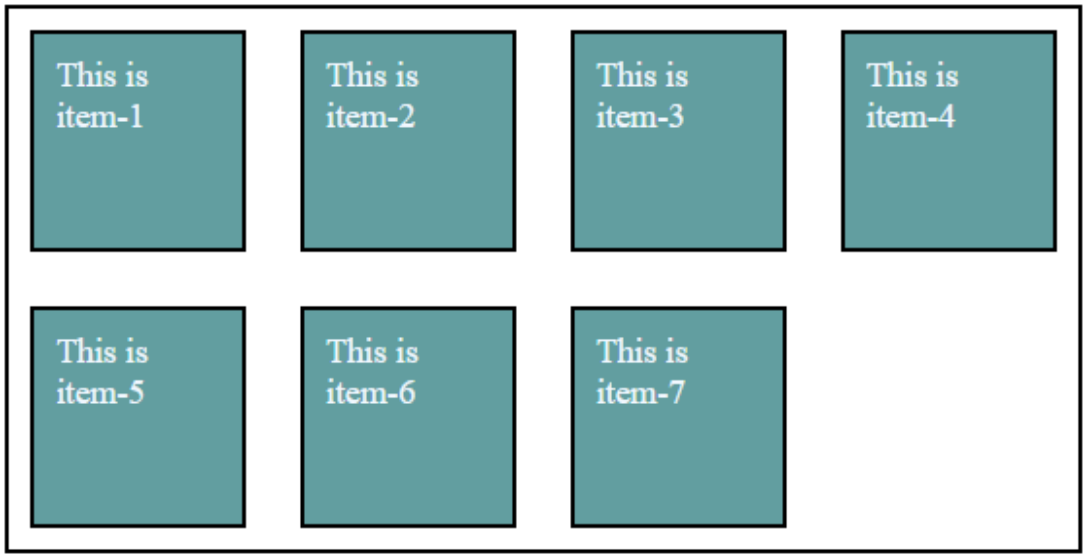
grid-column: 1 / span 2;



The grid-row: start/end properties :

- You use these two properties to join multiple ROWS together. It is shorthand of 2 properties:
  - **grid-row-start**
  - **grid-row-end**

grid-template-columns: repeat(4, auto);

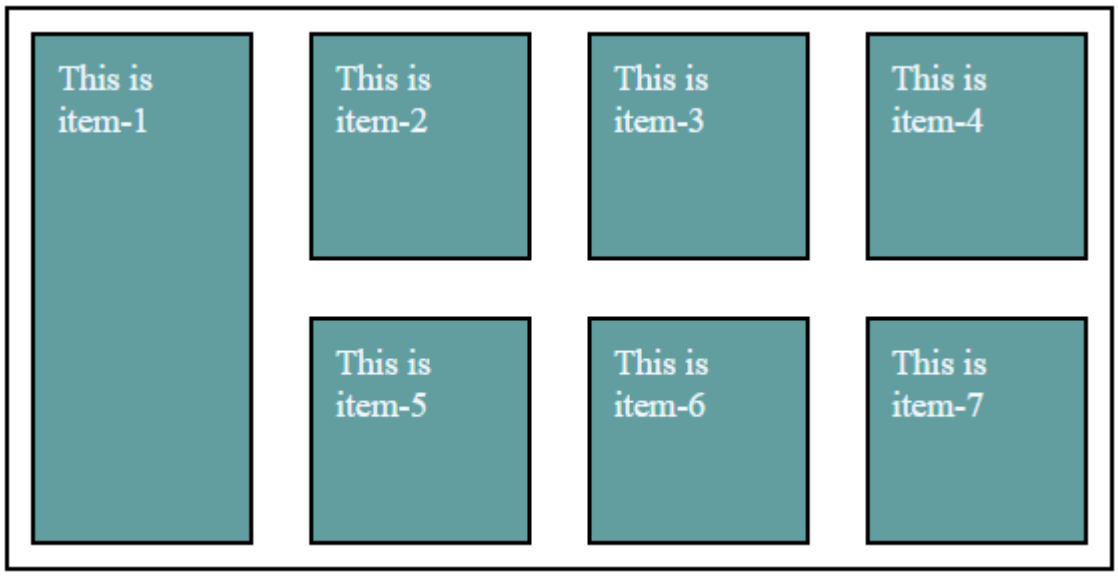


grid-row-start: 1;

grid-row-end: 2;

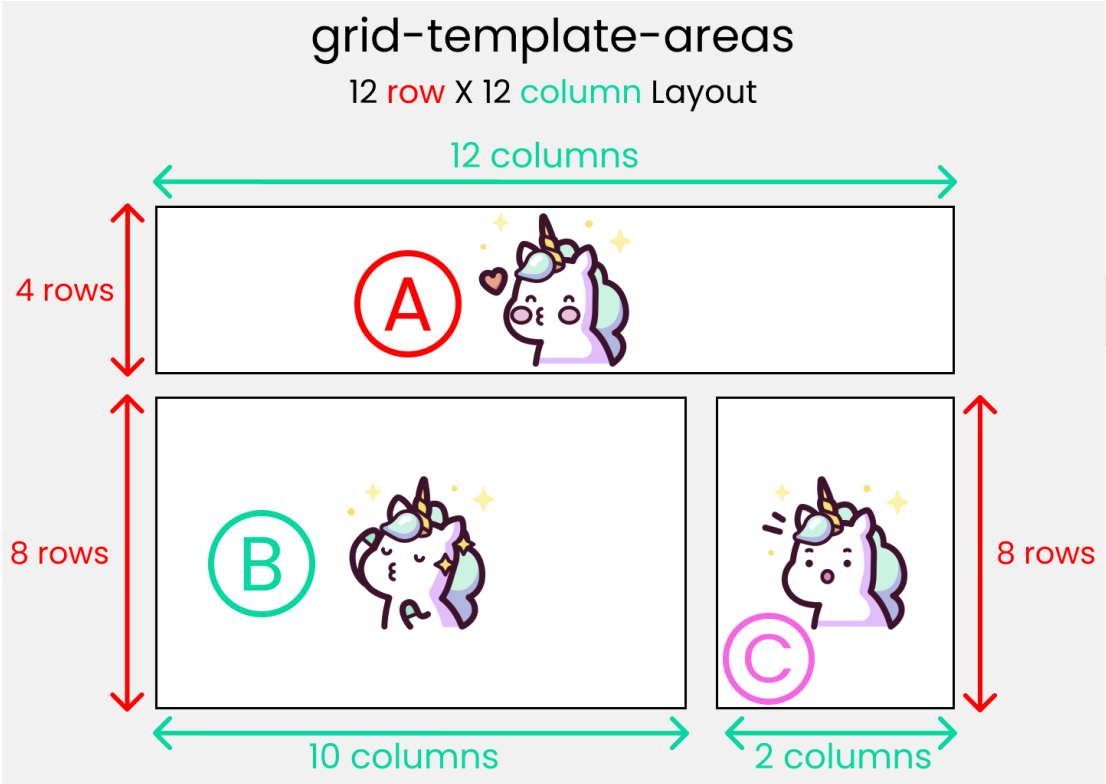
OR

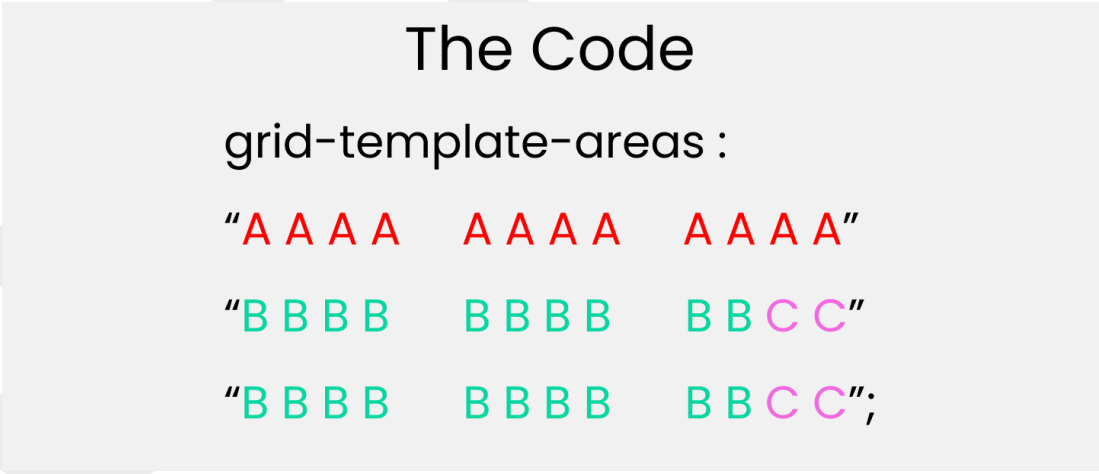
grid-row: 1 / span 2;



The grid-template-areas property :

- You use this property to specify the amount of space a grid cell should carry in terms of columns and rows across the parent container. Life's much easier with this property as it allows us to see visually what we're doing.





- To experiment with this, you need to understand both the parent and child properties:
  - **grid-template-areas**: The parent property that will create the blueprint
  - **grid-area**: the child property that will follow the blueprint.

Shorthand for CSS Grid Properties

place-content:

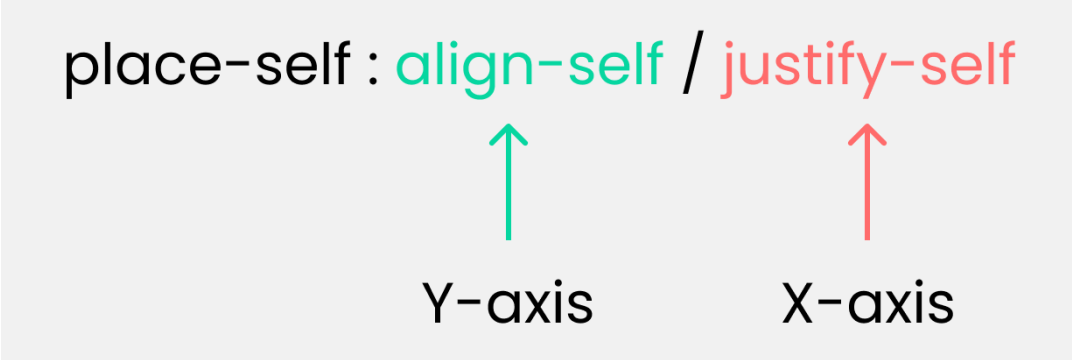
- This is the shorthand of 2 properties:
  - **align-content**
  - **justify-content**

place-items:

- This is the shorthand of 2 properties:
  - **align-items**
  - **justify-items**

place-self:

- This is the shorthand of 2 properties:
  - **align-self**
  - **justify-self**



grid-template:

- This is the shorthand of 2 properties:
  - **grid-template-rows**
  - **grid-template-columns**

```
grid-template : grid-template-rows / grid-template-columns
```

grid-template : 100px 100px / 200px 200px;

gap/grid-gap:

- This is the shorthand of 2 properties:
  - **row-gap**
  - **column-gap**

```
gap : row-gap  column-gap;
```

Reference Link :

- https://yoksel.github.io/grid-cheatsheet/
- https://grid.malven.co/
- https://learncssgrid.com/
- https://cssgridgarden.com/