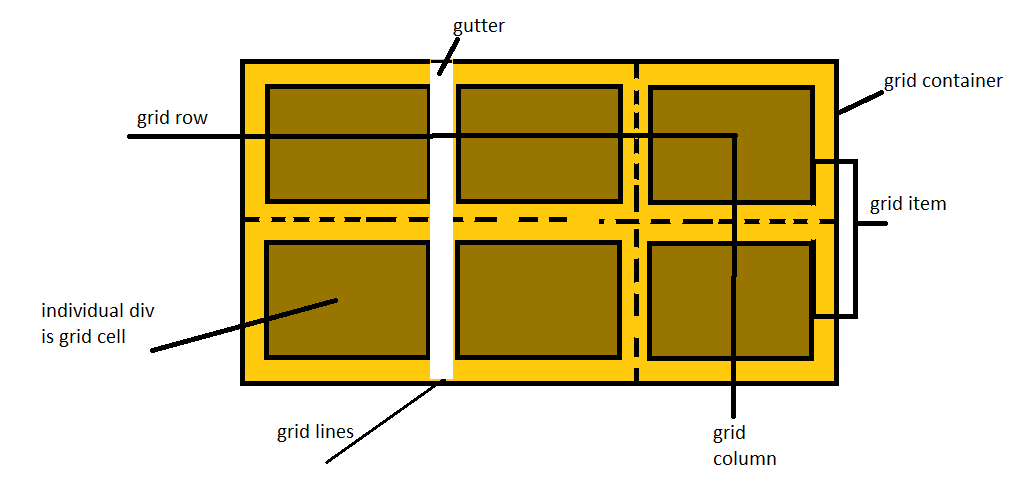
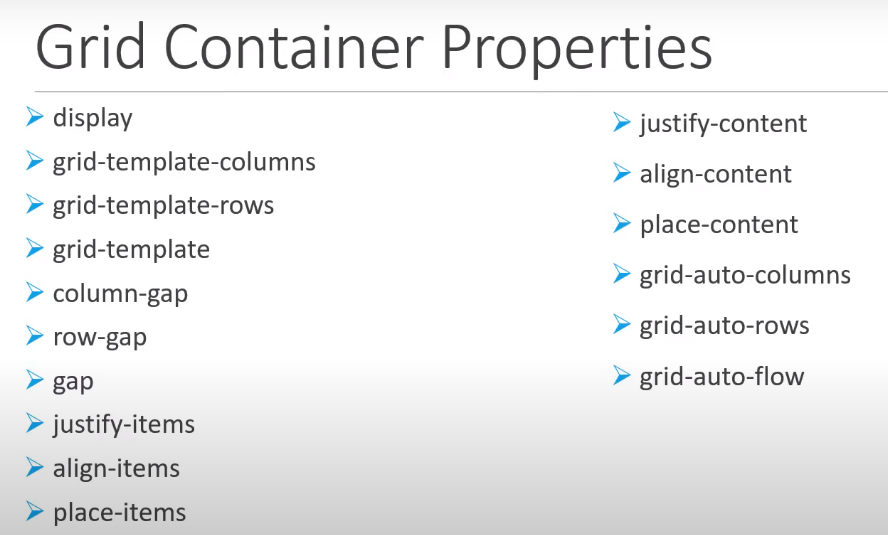
Css grid:

* It is two dimensional grid system
* We can handle row and column combinedly
* Here we skip the work of float left/right with grid
* We don’t have the confusion of padding and margin between the div
* We can create a layout with Fast and clean coding
* Not need to use frame work like angular

Grid system :







|  |  |
| --- | --- |
| <div class="container">      <div class="item item1">one</div>      <div class="item item2">two</div>      <div class="item item3">three</div>      <div class="item item4">four</div>      <div class="item item5">five</div>      <div class="item item6">six</div>      <div class="item item7">seven</div>       </div> | .container{      display: grid;      height: 600px;      width: 1000px;      background-color: gray;      grid-template-columns: 200px 300px  auto;      grid-template-rows: 200px auto  300px    } |

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **PROPERTY** | **VALUES** | **USABILITY** |
| 1 | display: | Grid | To use other grid properties like display flex |
|  |  | | |
| 2 | grid-template-columns: | 200px 300px auto  Repeat( 3, 200px)  Repeat(3, minmax(200px(minimum),  1fr(grow if have more space)) | to give the width to the columns |

|  |  |  |  |
| --- | --- | --- | --- |
| 3 |  | | |
|  | grid-template-row: | 200px 300px 400px  Repeat( 3, 200px)  Repeat(3, minmax(200px(minimum),  1fr(grow if have more space)) | to give the width to the columns |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | |  | | |
|  | grid-template | | grid-template:200px 200px 200px/200px 200px 200px ;  --------------------------------------------------  grid-template:repeat(3,200px)/repeat(3,minmax(200px,1fr )) | Shorthand for row and column both values |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **For the gap between the component / GRID GAP** | | |
|  |  | | |
|  | Grid-gap: | 20px | It use to give the gutter/ gap between the grid items in row and column also |
|  | Grid-row-gap: | 20px | To gutter/gap between the grid items which are in one Row. |
|  | Grid-column-gap: | 50px | To gutter/gap between the grid items which are in one column. |
|  | grid-gap: | 50px(for column) 100px(for row) | It is the short hand property for grid gap in column and row |

|  |  |  |
| --- | --- | --- |
| EX | grid-gap:20px 10px; | grid-row-gap: 20px;  grid-column-gap: 10px; |
|  |  | |

Justify-items:

it use to align or positioning the item content horizontally left to right in the cell/ grid item.

Values:

stretch (default) : will stretch the item content till the cell/ height/item horizontally.

Start : all the item content align to the left of the cell/ height/item horizontally.

Center: it put all the item content at center of cell /height/item horizontally.

End: it put all the item content at right of the cell/height/item horizontally.

=========================================================================

align-items: it use to align the item content vertically-top to bottom

Values:

stretch (default) : will stretch the items content till the cell/ height/item vertically.

Start : all the items content align to the top of the cell/ height/item vertically.

Center: it put all the items content at center of cell /height/item vertically.

End: it put all the items content at bottom of the cell/height/item vertically.

==============================================================================

Place-items: it is the shorthad property for justfy-items and align-items both property.

That first value will for align-items

Second value will for the justify-items

Separate both value with space

Ex. Place-items:start center; -> will push item vertically top and horizontally center.

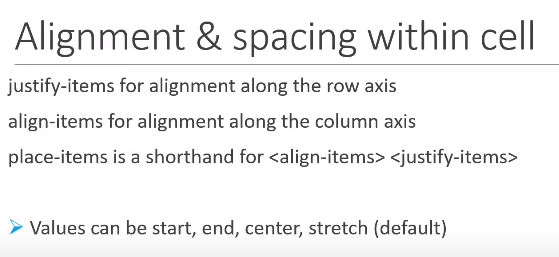
*\*we also can give a single value of center to set the items content vertically and horizontally center*

Ex. Place-item:center; -> it will push item center both vertically and horizontally.

Place-item:start; -> it will push item top-left

Place-item:end; -> it will push item bottom-right

==========================================================================-



jutify-content and align-content:

*To show the practice for align-content the containers height and width should be greater than items height and width*

|  |  |
| --- | --- |
| Html | Css |
| <div class="container">          <div class="grid\_item item1">one</div>          <div class="grid\_item item2">two</div>          <div class="grid\_item item3">three</div>          <div class="grid\_item item4">four</div>          <div class="grid\_item item5">five</div>          <div class="grid\_item item6">six</div>          <div class="grid\_item item7">seven</div>          <div class="grid\_item item8">eight</div>          <div class="grid\_item item9">nine</div>        </div> | .container{      border: 5px solid black;      text-align: center;      font-size: 1.5rem;      display:grid;      height: 800px;      grid-template:repeat(3,200px)/repeat(3,200px);      grid-gap: 10px;  } |

Justify-content: gives the facility to positioning and pacing the item horizontally as flex property.

**Values for positioning:** start = left (default) , center = center , end = right. (horizontally)

**Values for spaceing :** space-between, space-around, space-evenly.

Align-content : gives the facility to positioning and pacing the item vertically

**Values for positioning**: start = top(default) , center = center , end = bottom. (vertically)

**Values for spacing** : space-between, space-around, space-evenly.

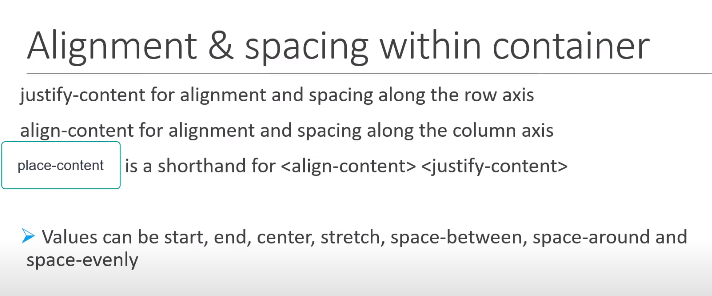
**Place-content:** it is the short hand property for justify-content and align-content for positioning and spacing between items vertically and horizontally at a time.

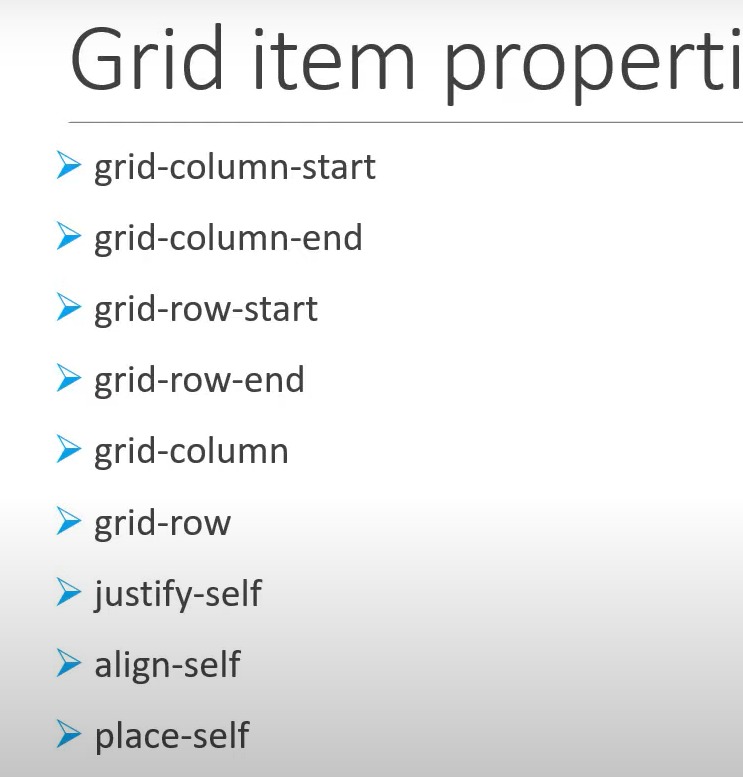
Ex.  place-content: space-between center;

*\*Specify the values with space*

*\*we also can use single value for both value.*

Ex.  place-content: space-between;





|  |  |  |  |
| --- | --- | --- | --- |
|  | **GRID ITEM POSITIONING/ and SPANING (SIZING) PROPERTIES:**  **These properties should use on “grid item”**  **For Row** | | |
|  | grid-row-start: | 1/2/3 etc | (from which row should it start) |
|  | Grid-row-end: | 1/2/3/4 etc | (at which row should it end) |
|  | grid-row: | 1(starting row) /2(ending row) | \*it is short hand for row start and row end combinedly |
|  | **For Column** | | |
|  | grid-column-start: | 1/2/3 etc | (from which column should it start) |
|  | Grid-column-end: | 1/2/3/4 etc | (at which column should it end) |
|  | grid-column: | 1(starting column) /2(ending column) | \*it is short hand for row start and row end combinedly |
|  | grid-column: | 1/span 2 | (it is span the column till the two column) |
|  | grid-column: | 1/-1 | (it stretch the column till the end of viewport ) |
|  | grid-area: | 1(r.starting)/  2(c.starting)/  3(r.ending)/  4(c.ending) | (it is also the shorthand for grid-column and grid-row combinly) |
| Ex | .item2{      background-color: rgb(195, 77, 96);      grid-column-start: 2;      grid-column-end: 5;  }  .item3{      background-color: rgb(65, 6, 16);      grid-column-start: 1;      grid-column-end: 2;  }  .item4{      background-color: rgb(78, 103, 8);      grid-column-start: 2;      grid-column-end: 4; | | |
|  |  | | |
|  | GRID LINES NAMING:  We can give the name to the grid line using [] quare bracket before the size given in column/row hight/width ex. | | |
|  | grid-template-columns: | [header-start]200px [header-end menu-start]300px [menu-end section-start]400px | Now we can use the names on the place of grid line numbers given in the templete to span the  and columns hight and width. |
|  | GRID Area NAMING: | grid-area: header  grid-area: menu  grid-area:section |  |
|  | grid-templete-area: | header header header | \*It get the width of three column |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **MIN-MAX FUNCTION IN GRID:** | | |
|  | **grid-template-columns** | **max-content** | \* it get the width maximum as which content in your column |
|  | **grid-template-columns** | **min-content** | \* it get the width mnimum as which content in your column |
|  | **grid-template-row:** | **min-content** | \* it get the Hight as per the content in your column |
|  |  |  |  |
|  |  |  |  |

**\*if we want to set a particular height and width but want to make changes if the content of the cell is enlarging itself then we have to use min max function.**

**grid-template-row:repeat(2 minmax(150px, min-content)**

**\* it will give the minimum height of row as 150px and the maximum height of row as per the content given in site. If the content will be large the height will manage itself as per the content.**

**IMPLICIT AND EXPLICIT GRID:**

**EXPLICITE GRID:**

**When we give the manual height and width to the cell it called explicit grid.**

**Implicit GRID:**

**The default sizing or setting of cells called implicit grid..**

**We can manage the height and width of implicit grid using some new properties**

**Like:**

**grid-auto-row: 100px**

**\*auto means implicit grid.**

**grid-auto-flow: column**

**\*if I want to add new grid in column not in row**

**grid-auto-column: .5fr**

**\*to give width of inplicit grid/content**

**-----------------------------------------------------------**

**NAMING GRID AREA:**

**The property to give the name to any of the grid item / div**

**grid-area: header**

**grid-area: menu**

**In the container div**

**grid-template-area: header header header**

**grid-template-area: menu menu menu**

**It will grow the width of header till the third colume**

**-----------------------------------------------------------**

**ALIGNING GRID ITEM:**

**We have 6 properties to alignment of grid items.**

**Align-item: start / end/ strech/ center**

**These both use for vertical alignment**

**We use these properties to the container**

**Align-self: start / end/ strech/ center**

**We use these properties to the content**

**We use these properties to align the iterms horizontaly.**

**justify-item: start / end/ strech/ center**

**We use these properties to the container**

**justify-self: start / end/ strech/ center**

**It is use choose the one content for horozontally alignment**

**Palce-item: <align-items><justify-items>**

**It is the short hand property for the align item and justify item**

**align-place: <align-self><justify-srlf>**

**It is the short hand property for the align self and justify self**

**grid track alignment :**

**we use these properties to align grid items horizontally**

**Align-content: start / end/ strech/ center/space-between/space-evenly/space-around**

* **It move all the content in as value in the container varticaly oposite of justify-content.**

**justify-content: start / end/ strech/ center/space-between/space-evenly/space-around**

**-it not move the value of content it move total content to the center of container as horizontally.**

**Palce-content:**

**-it is short hand of justify-content and align-content.**