

# CSS Grid Layout

two-dimensional grid system



## for the parent (Grid Container)

### 1 Define the element as a grid container

**display:** grid or inline-grid;

### 2 Define the rows & columns (when it's clear how many rows/cols you need : explicit grid)

**grid-template-rows:** 50px 50px 50px 50px;

**grid-template-columns:** 100px 100px 100px;

**grid-template-areas:** "header header header"  
" main main . "  
" main main . "  
" . . . " ;

All the direct children are set in rows automatically, except for children have {position: fixed; or absolute;};

available units

px em rem % auto fr  
min-content max-content  
minmax (50px, 100px)  
repeat (4, 50px)  
repeat (auto-fill, 50px)  
repeat (auto-fit, 50px)

[row1-start] 50px [row1-end row2-start] 50px [row2-end row3-start] 50px [row3-end] 50px [line5];

**sh** **grid-template:** 50px 50px 50px 50px / 100px 100px 100px;

grid-template-rows / grid-template-columns;

**sh** **grid-template:** "header header header" 50px  
" main main . " 50px  
" main main . " 50px  
" . . . " 50px  
/ 100px 100px 100px;

grid-template-areas \_ grid-template-rows / grid-template-columns;

### Define the rows & columns (when you don't know how many rows/cols you need : implicit grid)

**grid-auto-rows:** auto or 50px or minmax(50px, auto);

**grid-auto-columns:** auto or 100px or minmax(100px, auto);

**grid-auto-flow:** row or row dense or column or column dense;

grid: is a shorthand for grid-template-rows, grid-template-columns, grid-template-areas, grid-auto-rows, grid-auto-columns, and grid-auto-flow.

**sh** **grid:** auto-flow 50px / 100px 100px 100px;

grid-auto-flow \_ grid-auto-rows / grid-template-columns;



**grid-row-gap:** 10px;

**grid-column-gap:** 20px;

**sh** **grid-gap:** 10px 20px;

grid-row-gap \_ grid-column-gap;



**align-content:** start | center | end | stretch | space-around | space-between | space-evenly;

**justify-content:** start | center | end | stretch | space-around | space-between | space-evenly;

**sh** **place-content:** center start;

align-content \_ justify-content;

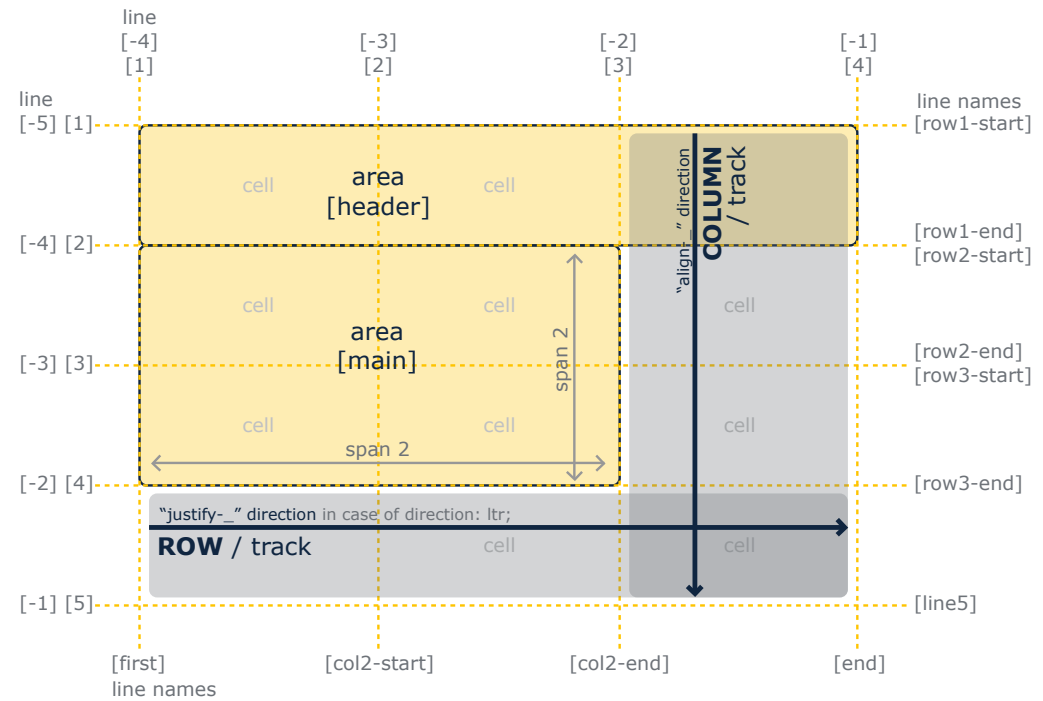


**align-items:** stretch or start or center or end;

**justify-items:** stretch or start or center or end;

**sh** **place-items:** end center;

align-items \_ justify-items;



## for the children (Grid Items)

### 3 Determine a grid item's location

**.main**  
**grid-row-start:** 2;

**grid-row-end:** 4;

**sh** **grid-row:** 2 / 4;

**grid-column-start:** 1;

**grid-column-end:** 3;

**sh** **grid-column:** 1 / 3;

**sh** **grid-area:** 2 / 1 / 4 / 3;

**grid-area:** main;

available units

2 / span 2;  
row2-start / row3-end;

When grid-area is set, the DOM order is not respected.

grid-row-start / grid-column-start / grid-row-end / grid-column-end;



**align-self:** stretch or start or center or end;

**justify-self:** stretch or start or center or end;

**sh** **place-self:** end center; align-self \_ justify-self;

**z-index:** 0;

**order:** 0;

Grid items can occupy the same cells, if you explicitly set their location. Z-index can change which is on top.

[references] [https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Grid\\_Layout](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout)  
<https://css-tricks.com/snippets/css/complete-guide-grid/>  
<https://www.udemy.com/course/css-the-complete-guide-incl-flexbox-grid-sass/>