

CSS Grid Crash Course Refresher

Basics:

(1) Define grid as display

(2) Set column and row size with grid-template-columns and grid-template-rows

(3). Place children with grid-column and grid-row

↳ The source order of items does not matter and can be re-arranged with CSS
* J.S. Anchors, grid allows for easier positioning in two dimensions

Terms:

- Grid Container: element w/ display applied
- Grid Item: the direct descendants of the grid container
- Grid line: the dividing lines that make up the grid. Can be horizontal or vertical.
- Grid cell: the space between two adjacent row and two adjacent column lines
- Grid Track: space between two adjacent grid lines i.e rows and columns
- Grid Area: total space surrounded by four grid lines

Properties for Children

grid-column-start
grid-column-end
grid-row-start
grid-row-end } -Item location in grid as designated by grid lines
{ as values:
-line, or span <number>, or span <name>, or auto
* Auto indicates auto placement, an automatic span, or default span of 1

grid-column is shorthand i.e. grid-column: 2 / 4;
grid-row is also a shorthand. span keyword can also be used.
grid-area gives an item a name or coordinates 1 / 1 / 3 / 4
grid-area shorthand
row-start
row-end
column-start
column-end

Properties for parent

grid-template-columns
grid-template-rows } track size can be length or a percentage
{ specifying over and over is tedious. Use the repeat (#, x) function

auto can be used to size row based on content size of other cells in row.

CSS Grid Crash Course (continued)

→ we can put things @ the end w/ auto-fill and grid-column-end -1

- The **repeat function** can get fancy with keywords
 - ↳ **auto-fit**: fit as many columns as possible on row even if they are empty
 - ↳ **auto-fit**: fit whatever columns there are in the empty space. Prefer expanding columns over empty
- Grid has an **implicit** vs **explicit** concept. Browser is smart enough to calculate things!
 - ↳ you only provide two rows but have extra items
 - ↳ you only define columns and row # is implicit.
 - ↳ `grid-auto-columns` and `grid-auto-rows` can be used to specify size of generated tracks
- **fr**: fractional unit. Think of this as free space, left after all elements laid out
- **spanning** in CSS grid can be used to dictate size of child items w/out affecting column sizes.
 - ↳ using `grid-column` and `grid-row` w/ 1/1 will span the entire explicit grid
- Using `min-max` in conjunctions with **auto-fit** and **auto-fill** can often replace media queries!