**Grid**

Chart, treemap chart

Description automatically generated

if we make this with flexbox, it’s a bit hard since flexbox is only good in one direction. As well, we need flex containers within flex containers such as nav and aside being in a flex container which itself is in a flex container with main which is in a flex container with header and footer.

with grid, without any nested containers. as well, the order doesn’t matter so footer can be above header

Text

Description automatically generatedGraphical user interface, application

Description automatically generated

the grid treats the webpage as a grid, which makes the right two things ez to make

Chart

Description automatically generatedChart, treemap chart

Description automatically generatedTreemap chart

Description automatically generated

**Columns:**

- grids have two directions, rows and columns

- we can have as many columns as we want

A picture containing text

Description automatically generatedText

Description automatically generatedA picture containing text, building

Description automatically generated

content will be the grid container

to make a div a grid, do display: grid; this doesn’t actually change anything by itself, it simply makes this container a grid.

since we have 3 numbers below, we have 3 columns. The numbers represent the width of each column respectively.

grid-template-columns: 33.3% 33.3% 33.3%;

Text

Description automatically generatedGraphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generatedA picture containing shape

Description automatically generated

we can use fractions as well. The fraction of the total grid size a column takes up is

= the column’s fraction number/sum of the all the columns’ fraction number

in the below example, the second column takes up 2/(1+2+1) = ½ of the total width

grid-template-columns: 1fr 2fr 1fr;

Text

Description automatically generated Graphical user interface, application

Description automatically generated

grid-template-columns: repeat(numberOfColumns, 1fr);

the below is the same as saying grid-template-columns: 1fr 1fr 1fr 1fr 1fr 1fr 1fr 1fr 1fr;

Text

Description automatically generated

**Rows:**

- rows are automatically created when we have the content to do it

- they height of these rows are governed by the content within the items, as well as padding and stuff

Text

Description automatically generated

- if I add a lot of content to the third item,

A picture containing treemap chart

Description automatically generated

if we want each row to have a certain height, we can do grid-auto-rows: 200px; which makes every row have 200px in height. However, notice it will cut off the content to maintain the row heigh of 200px.

A picture containing background pattern

Description automatically generated

grid-auto-rows: minmax(200px, auto); will make the minimum height of a row 200px and the max height to be whatever it needs to be which is determined by content, padding, margin, etc

A picture containing treemap chart

Description automatically generated

grid-template-rows: 200px 300px 400px 200px;

this makes the first, second, third, and fourth rows have heights of 200, 300,400,200px respectively. Notice that there will be space created for the fourth row even through nothing is inside it.

Background pattern

Description automatically generated

grid-template-rows: repeat(3, 200px) creates 3 rows, each with a height of 200px but it still cuts off content

Shape

Description automatically generated with low confidence

grid-template-rows: repeat(3, minmax(200px, auto)) creates 3 rows, each with a height of 200px without cutting off content

A picture containing background pattern

Description automatically generated

don’t, use margins to create spaces around each grid item, since it creates margin to the left of item 1 and to the right of item 3 which we don’t want. rather we use grid-column-gap: 10px; which makes the gap between column 10px;

A picture containing background pattern

Description automatically generated

as well, we can use grid-column-gap: 10px; which makes the gap between rows 10px;

Text

Description automatically generatedA picture containing background pattern

Description automatically generated

we can also use grid-gap: 10px; to make the row and column gap 10px

**Grid Lines:**

- help us define positions gird items in a grid

imagine we have an 8 by 8 grid, there are 9 column lines and 9 row lines

A picture containing background pattern

Description automatically generated

to put an element like below, we say we want to go from column line 3 to 6 and row line 2 to 5

Chart

Description automatically generated

without any grid lines, yet the code might look as follows. we have defined 6 columns and 4 rows(even though we cant see the rows, they are still there). Thus we have 7 column lines and 5 row lines.

Text

Description automatically generated Text

Description automatically generated

Graphical user interface, chart

Description automatically generated

.one{

grid-column-start: 1; //this line says start at column line 1

grid-column-end: 3; //this line says end at column line 3

grid-column: 1 / 3; // this is a condensed way of saying go from column line 1 to 3

}

Chart, treemap chart

Description automatically generated

now we make item 2 go from column line 3 to 7

Text

Description automatically generatedA picture containing graphical user interface

Description automatically generated

grid-row: 2 / 4; makes item 3 go from row line 2 to 4

Text

Description automatically generatedChart, treemap chart

Description automatically generated

the items don’t have to take up all the space, notice item 6 is pushed to the next row

Text

Description automatically generatedChart, treemap chart

Description automatically generated

the items don’t have to be in order either

Text

Description automatically generatedChart, treemap chart

Description automatically generated

**Nest Grids:**

without nested grids, it might look as follows

Text

Description automatically generated with low confidenceA picture containing shape

Description automatically generated

with nested grids, it might look as follows. the div with class = ‘nested’ is a gird inside the div with id = ‘content’ which is also a grid, hence nested grid.

As well, we can say grid-column: span 3; means to take up 3 columns

Text

Description automatically generatedGraphical user interface

Description automatically generated with medium confidence

**Aligning and Justifying Items:**

by default,

Text

Description automatically generatedText

Description automatically generatedA picture containing square

Description automatically generated

within the grid container, we can use align-items to make items go up and down in their space. we can do align-items: start; to change all grid items to go to the top of their space.

Text

Description automatically generatedWebsite, timeline

Description automatically generated

as well, we can also do align-items: end; or align-items: stretch; to make items go to the bottom or take up the entire item space height respectively

end:

Website, timeline

Description automatically generated

stretch:

A picture containing website

Description automatically generated

within the grid container, we can use justify-items to make items go left and right in their space. we can do justify-items: start; to change all grid items to go to the left of their space.

Chart, waterfall chart

Description automatically generated

as well, we can also do justify-items: end to make items go to the right. We can also do justify-items: stretch to take up all the width. justify-items: end is shown below.

Graphical user interface

Description automatically generated with medium confidence

if we use both align-items: start; and justify-items: end; we can make items look like

Graphical user interface

Description automatically generated with medium confidence

instead of changing all the grid items, we can specify which grid items we want to change

the commands are the same, but instead of align-items and justify-items in the grid container, we say align-self and justify-self inside the div for the grid item

Text

Description automatically generated Square

Description automatically generated

**Grid Areas:**

while we can use grid-column or grid-row, we can also use grid areas. give each grid item a name with the syntax

grid-area: grid\_area\_name;

Text

Description automatically generated Text

Description automatically generated

then in the grid container,

we can say grid-template-areas: and input the grid\_area\_name for each grid item. Use a . to represent a space.

Text

Description automatically generatedChart

Description automatically generated