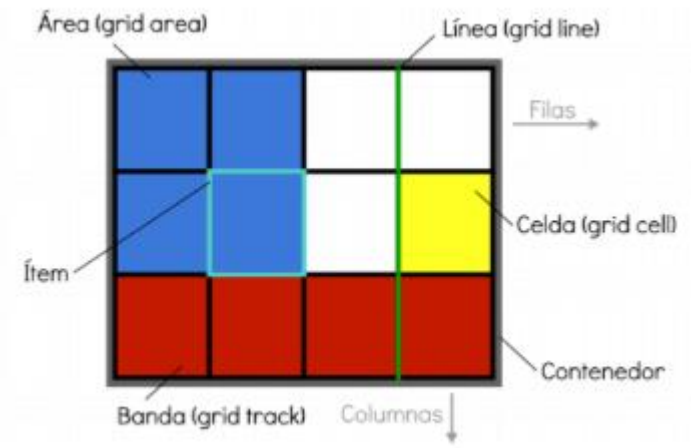
Esquema GRID



Para trabajar con GRID:

display: grid; display: inline-grid;

Definir la estructura de nuestro grid:

Para filas y columnas

- Grid-template-columns
- Grid-template-rows
- Unidades: px, rem, em, vh, vw, %, auto

Repeticion de filas y columnas:

- repeat([numero de veces], [tamaño])

Unidades especiales: Min-content, max-content, fit-content(maxtama):

- Min-content: Toma como ancho la parte más ancha del contenido y establece automáticamente ese valor como ancho mínimo.
- Max-content: Toma como ancho todo el contenido y establece ese valor como minimo.
- Fit-content: Toma como ancho mínimo va a ser el min-content y el ancho máximo puede crecer hasta que llegue al valor del argumento maxtama o hasta que llegue al max-content, el primero de los valores que se dé cuando estemos aumentando el tamaño.

Responsive grid: Minmax(), auto-fill, auto-fit

La función minmax() no permite trabajar de forma responsiva con grid, para ello, tenemos que pasarle dos valores que nos permiten establecer el tamaños mínimo y el máximo que pueden tener los ítems, cuando redimensionemos nuestro grid.

Auto-fill: crea grid-items vacíos como quepan en el viewport respetando las medidas.

Auto-fit: los elementos ocupan todo el espacio disponible en el grid.

RESUMEN DE UNIDADES

Estos son los valores que podemos utilizar para grid-template-columns y grid-template-rows:

- Longitud: px, rem, em, vh, vw, % ...
- Fracciones: fr
- Automático: auto
- repeat()
- min-content, max-content, fit-content() Tamaño de filas o columnas
- minmax()
- auto-fit y auto-fill

Espaciado entre elementos

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

Shrothands:

grid-gap: 2px 10px; grid-gap: 1em;

Agnadir elementos a nuestro grid: grid-column y grid-row

- Grid-column
- Grid-row
- Grid-column-start
- Grid-column-end
- Grid-row-start
- Grid-row-end
- El valor que van a tomar son las lineas donde empiezan y acaban los elementos que se desean agnadir.
- Admiten valor positive, valor negative, y span (numero de columnas o filas que ha de ocupar).
- Existe un shorthand que engloba las dos propiedades
- grid-column: start / end grid-row: start / end

Elementos implícitos

Grid-auto-columns y grid-auto-rows

Establecen qué hacer con las columnas y filas no definidas, para ello le damos el tamaño que queremos que tengan las columnas y filas que no hemos definido con grid-template-columns y grid-template-rows, en el explicit grid.

Grid-auto-flow

Establece la dirección en la que van a aparecer los elementos

Valores que admite: columna, row, dense.

Alineación

Mirar la cheatsheet

GRID TEMPLATE AREAS

Es una forma de distribuir los elementos dentro de nuestro grid utilizando identificadores, para ello vamos a dar un nombre a cada uno de nuestros elementos y con ellos vamos a realizar la distribución de los elementos.

Vamos a utilizar dos propiedades, la propiedad grid-template-areas, que vamos a utilizar en nuestro grid-container, en la que vamos a definir la estructura de nuestro grid, utilizando los identificadores y la segunda propiedad que vamos a utilizar es grid-area, que la vamos a definir

en cada uno de nuestros elementos y le daremos el nombre que queremos utilizar como identificador y que luego utilizamos en grid-template-areas.

Shorthand:

Grid-template engloba: grid-template-areas, grid-template-rows y grid-template-columns.