Tema 7: CSS Grid

Grid es una cuadrícula:



Grid container

display: grid;

Grid elements: Elementos que se muestran en pantalla,es reponsive de manera nativa.

Con grid en tres columnas, cuando llegue a cierto tamaño pues se reorganiza como por ejemplo a dos columnas.

Grid lines: líneas de la cuadrícula, cuando se interconectan unas con otras se convierten en la específica.



Grid cell: celda de la cuadrícula.

grid-row-start: 3;

grid-row-end: 4;

grid-colum-start:3;

grid-column-end:4;

grid-area:3/3/4/4; →

Grid Track: Se refiere a una fila o una columna.



→ Grid track

Grid-template-areas: pulsar enter significa algo.

“qwaffswfsfsfsfsfsfsf/n

sfsfsfsfsfsfsfsfsfsw/n

sdfgagwagawgfawg”

La cuadrícula por defecto tiene el 100% de su contenedor.

Unidad **Fr** : Fracción, proporción de la cuadrícula.

grid-template-rows:→Número de filas.

grid-template-columns: →Número de columnas → 100px 100px 200px Creame tres columnas de ese tamaño por ejemplo, o 1fr 1fr 2fr 0.5fr

grid-template-columns:repeat (5, 1fr)

repeat (5,1fr) 2fr

row-gap / column-gap , espacio entre celdas o columnas.

contenedor{

display: grid;

grid-template-rows: [L1]1fr [L2]2fr [L3]1fr;

grid-template-columns:[a]2fr [b]1fr[c];

}

Así se les asignará un nombre.



.e1{

grid-rows-start:L1;

grid-row-end: L3;

grid-column-start:a;

grid-column-end:b;

}

Empieza y termina el cuadrado.

grid-row-gap → gap

grid-column-gap→ gap

gap: espacio entre filas y columnas.

gap: 10px 20px;

Hacer una especie de esquema de las diferentes partes en las que vamos a dividir una cuadrícula

.

grid-template-areas

.contenedor{

display:grid;

grid-rows: repeat(4,1fr);

grid-column: repeat(4,1fr);

grid-template-areas:

“area1 area1 . .”

“area1 area1 area2 area2”

“. area3 area3 area3”

“. area3 area3 area3”}

.a{

grid-area:area1;

}

filas y columnas.

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

grid-template:

“a1 a1 .” 2fr

“a1 a1 .” 1fr  
“. a2 a2” 1fr / 1fr 1fr 1fr;



start

end →align-items → Vertical

→ justify-items →Horizontal →place-items: vertical horizon;

center

stretch

align-content →vert

→place-content

justify-content →hor

grid-column-start:

grid-column-end:

grid-row-start:

grid-row-end:

grid-column-start:1;

grid-column-end:span2;

grid-column:1/span2;

grid-row-start, grid-column-start, grid-row-end, seguido de grid-column-end.

grid-area 3/3/4/5;

Si los elementos de la cuadrícula no se sitúan explícitamente con grid-area, grid-column, grid-row, etc., se sitúan automáticamente de acuerdo al orden en el código fuente. Puedes sobrescribir esto usando la propiedad order, que es una de las ventajas de la cuadrícula frente al diseño basado en tablas.

order:1; (Funciona como el z-index);

Función minmax(valor1,valor2);

minimo izquierda y máximo a la derecha.

grid-template-columns: repeat(5,minmax(250px, 1fr));

grid-temaplte-columns: repeat(autofit,minmax(150px,1fr));

o autofill

/\*Indica el foramto de las columnas que se quedan fuera de lo explicito\*/

/\*grid-auto-rows: ;\*/

grid-auto-columns: minmax(10px,2fr);

Hacer rotar las imagenes:

backface-visibility: hidden;

transform-style: preserve-3d

.girada{

transform: rotateY(180deg);