


Conceptos Básicos de CSS Grid:

1. **display: grid;**
 - Define un contenedor como un grid.
2. **grid-template-columns y grid-template-rows:**
 - Define el número y el tamaño de las columnas y filas del grid.


CSS

 Copy code

```
grid-template-columns: 100px 200px 100px;  
grid-template-rows: 50px 100px;
```

3. **grid-gap:**
 - Establece el espacio entre las filas y columnas del grid.

CSS


 Copy code

```
grid-gap: 10px;
```

Colocación de Elementos en el Grid:

1. **grid-column y grid-row:**
 - Define en qué columnas y filas debe ubicarse un elemento.


CSS

 Copy code

```
grid-column: 2 / 4;  
grid-row: 1 / 3;
```

2. **grid-area:**
 - Define en qué área del grid debería estar un elemento.

CSS


 Copy code

```
grid-area: 1 / 2 / 3 / 4;
```

Alineación en CSS Grid:

1. **justify-items y align-items:**
 - Alinea los elementos dentro de las celdas del grid a lo largo del eje principal y transversal, respectivamente.


CSS

 Copy code

```
justify-items: center;  
align-items: end;
```

2. **justify-self y align-self:**
 - Alinea un elemento específico dentro de su celda.

CSS

 Copy code


```
justify-self: start;  
align-self: center;
```

Repetición y Fracciones:

1. `repeat()`:

- Permite repetir la definición de columnas o filas.

CSS


 Copy code

```
grid-template-columns: repeat(3, 100px);
```

2. `fr`:

- Una unidad flexible que asigna espacio proporcionalmente en función de la disponibilidad.

CSS

 Copy code


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

CSS Grid en Responsividad:

1. `minmax()`:

- Define un rango para el tamaño de columna o fila.

CSS


 Copy code

```
grid-template-columns: minmax(100px, 1fr) 2fr 1fr;
```

2. `auto-fill` y `auto-fit`:

- Permiten que el número de columnas o filas sea automático según el tamaño del contenedor.


CSS

 Copy code

```
grid-template-columns: repeat(auto-fill, minmax(100px, 1fr));
```

Sintaxis de `repeat()`:

CSS

 Copy code

```
/* Sintaxis básica */
grid-template-columns: repeat(number_of_repeats, track_size);

/* Ejemplo: Crear 3 columnas de 100px cada una */
grid-template-columns: repeat(3, 100px);

/* También puede aceptar múltiples valores */
grid-template-columns: repeat(2, 100px 50px);

/* Puede combinar con otras unidades o fracciones */
grid-template-columns: repeat(4, 1fr 2fr);
```

Ejemplos de Uso:

Crear Columnas de Pares e Impares:

CSS

Copy code

```
/* Crear 5 columnas, alternando entre 100px y 50px */  
grid-template-columns: repeat(5, 100px 50px);
```

Crear Columnas con Secuencias:

CSS

Copy code

```
/* Crear 4 columnas con secuencias de 100px, 150px, 200px y 250px */  
grid-template-columns: repeat(4, 100px 150px 200px 250px);
```

Combinar `repeat()` con Otras Unidades:

CSS

Copy code

```
/* Crear 3 columnas, la primera con 100px, las siguientes dos con 1fr */  
grid-template-columns: 100px repeat(2, 1fr);
```

Notas Adicionales:

- Puedes usar la función `auto-fill` o `auto-fit` con `repeat()` para crear un número automático de columnas o filas que se ajusten al tamaño del contenedor.

CSS

Copy code

```
/* Crear columnas automáticas con un mínimo de 100px y máximo de 1fr */  
grid-template-columns: repeat(auto-fill, minmax(100px, 1fr));
```

1. `auto-fill`:

- `auto-fill` intenta llenar la cuadrícula con tantas columnas o filas como sea posible, incluso si no hay suficientes elementos para llenar todas las celdas. Si hay menos elementos que el número especificado en `repeat()`, algunas celdas estarán vacías.

CSS

Copy code

```
grid-template-columns: repeat(auto-fill, minmax(100px, 1fr));
```

2. `auto-fit`:

- `auto-fit` también intenta llenar la cuadrícula con tantas columnas o filas como sea posible. Sin embargo, a diferencia de `auto-fill`, si hay menos elementos que el número especificado en `repeat()`, `auto-fit` ajustará automáticamente el tamaño de las columnas o filas para ocupar todo el espacio disponible, evitando celdas vacías.

CSS

Copy code

```
grid-template-columns: repeat(auto-fit, minmax(100px, 1fr));
```