



Grid Layout



Grid Layout

- Grid Layout es un nuevo modelo de diseño para CSS que tiene capacidades poderosas para controlar el tamaño y la posición de los cuadros y sus contenidos.
- A diferencia del diseño flexbox, que está orientado a un solo eje, el diseño de grilla está optimizado para diseños de 2 dimensiones: aquellos en los que se desea la alineación del contenido en ambas dimensiones.
- Una cuadrícula es un conjunto de líneas horizontales y verticales que se intersectan - un grupo define columnas y el otro filas.
- Los elementos se pueden colocar en la cuadrícula respetando estas columnas y filas.



Características

- Posicionamiento preciso de elementos.
- Control de alineación de elementos.
- Grilla de tamaño fijo y flexible.
- Elementos superpuestos.

Display

- Define el elemento como un contenedor de cuadrícula y establece un nuevo contexto de formato de cuadrícula para su contenido.
- Las propiedades que se pueden usar son grid y inline-grid.

```
.container {
```

```
  display: grid | inline-grid;
```

```
}
```



Grid Template

- Define las columnas y filas de la cuadrícula con una lista de valores separados por espacios. Los valores representan el tamaño del ítem y el espacio entre ellos representa la línea de la cuadrícula.
- Posibles valores: **grid-template-columns** y **grid-template-rows**.
- Grid también introduce una unidad de longitud adicional para ayudarnos a crear vías de cuadrícula flexibles, **fr**.
- La nueva unidad **fr** representa una fracción del espacio disponible en el contenedor de la cuadrícula.

Grid Template - Repeat

- Las cuadrículas grandes con muchas vías o celdas pueden utilizar la notación `repeat()` con el fin de repetir todas o una sección de la lista de vías.
- La notación de repetición se puede utilizar para una parte del listado.

```
.wrapper {  
  display: grid;  
  grid-template-columns: 20px repeat(6, 1fr) 20px;  
}
```



Grid Implícita y Explícita

- Cuando definimos `grid-template-columns`, grid crea las filas necesarias automáticamente para el contenido.
- Estas filas se crean en la cuadrícula implícita.
- La cuadrícula explícita consiste en las filas y columnas que se definen con las propiedades `grid-template-columns` y `grid-template-rows`.
- Si coloca algo fuera de la cuadrícula ya definida, o si debido a la cantidad de contenido, se necesitarán más vías o celdas, entonces grid crea filas y columnas en la cuadrícula implícita.
- También puede definir un tamaño para el conjunto de vías creadas en la cuadrícula implícita con las propiedades `grid-auto-rows` y `grid-auto-columns`.

Tamaño minmax

- Se pueden definir los tamaños mínimos y máximos de una cuadrícula para asegurarse que se adapten a cualquier contenido.
- La función minmax define un rango de tamaño mayor o igual que min y menor o igual que max.
- La función minmax puede ser usada con las siguientes propiedades: `grid-template-columns`, `grid-template-rows`, `grid-auto-columns` y `grid-auto-rows`

```
.wrapper {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  grid-auto-rows: minmax(100px, auto);  
}
```

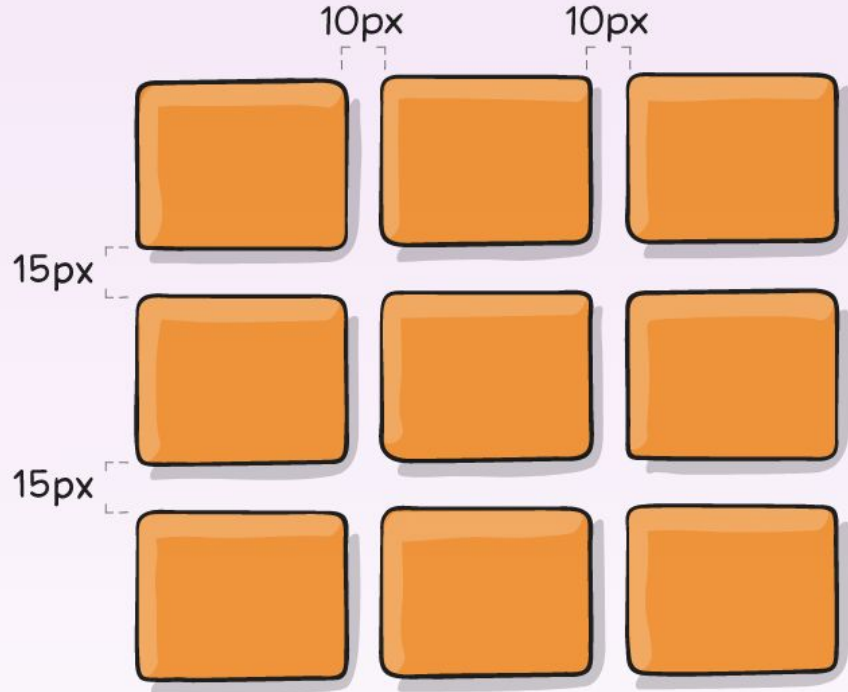

Grid Gap I

- Especifica el tamaño de las líneas de la cuadrícula.
- Puede pensar en ello como establecer el ancho del espacio entre las columnas y las filas.
- Las propiedades son: grid-row-gap y grid-column-gap.
- Una alternativa a estos es grid-gap.

```
.container {  
  grid-column-gap: <line-size>;  
  grid-row-gap: <line-size>;  
}
```

Grid Gap II

```
.container {  
  grid-template-columns: 100px 50px  
  100px;  
  grid-template-rows: 80px auto  
  80px;  
  grid-column-gap: 10px;  
  grid-row-gap: 15px;  
}
```



Justify y Align Items

- **justify-items:** Alinea los elementos de la cuadrícula a lo largo del eje en línea (fila)
- **align-items:** Alinea los elementos de la cuadrícula a lo largo del eje del bloque (columna).
- Los valores que pueden tomar son: start, end, center y stretch.
- Puede utilizarse place-items para configurar las dos.

```
.container {  
  align-items: start  
  justify-items: stretch;  
}
```

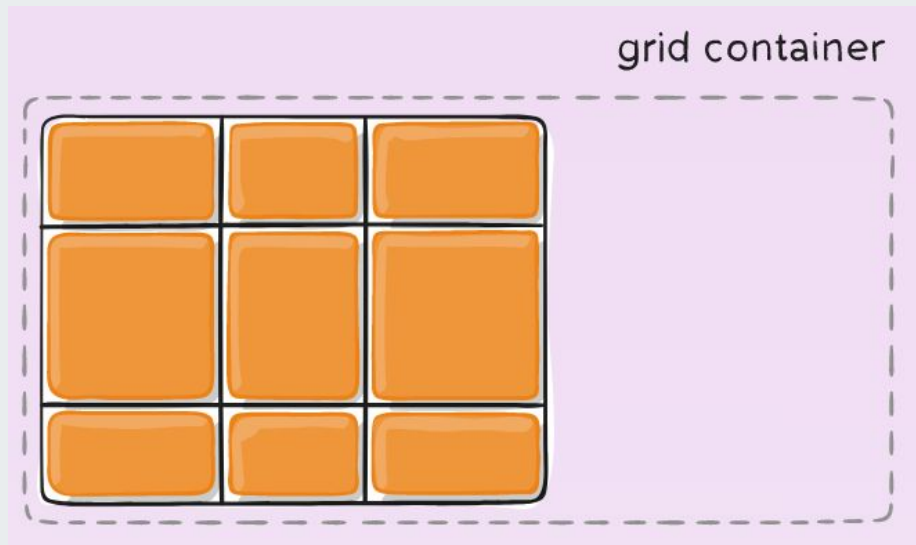


Justify Content I

- A veces, el tamaño total de su cuadrícula puede ser menor que el tamaño de su contenedor de cuadrícula.
- Esto podría suceder si todos los elementos de la cuadrícula se dimensionan con unidades no flexibles como px.
- En este caso, puede establecer la alineación de la cuadrícula dentro del contenedor de la cuadrícula.
- Esta propiedad alinea la cuadrícula a lo largo del eje en línea (fila).
- Los posibles valores son: start, end, center, stretch, space-around, space-between, space-evenly.

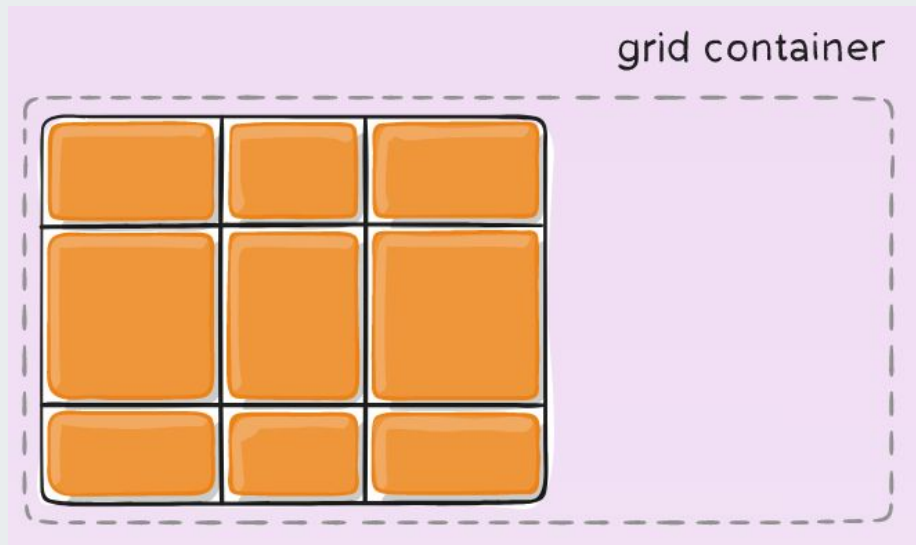
Justify Content II

```
.container {  
  justify-content: start;  
}
```



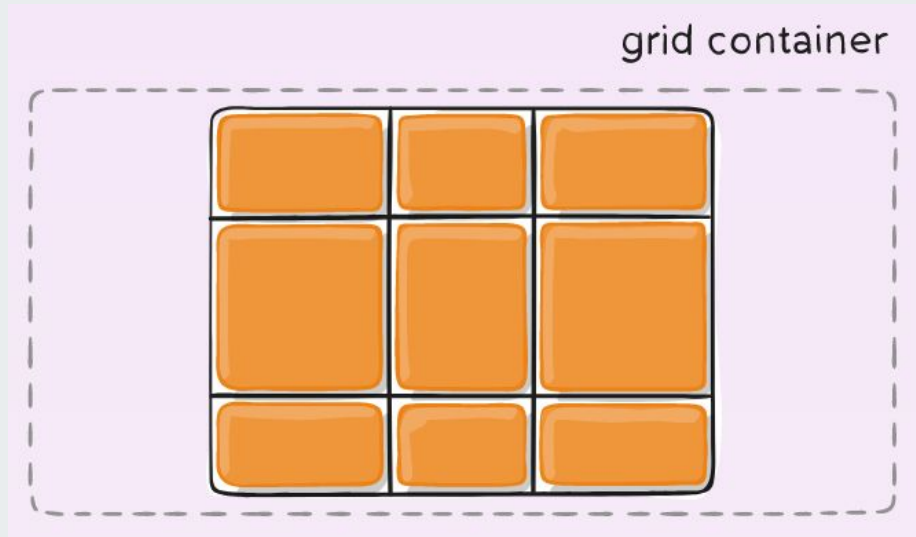
Justify Content II

```
.container {  
  justify-content: start;  
}
```



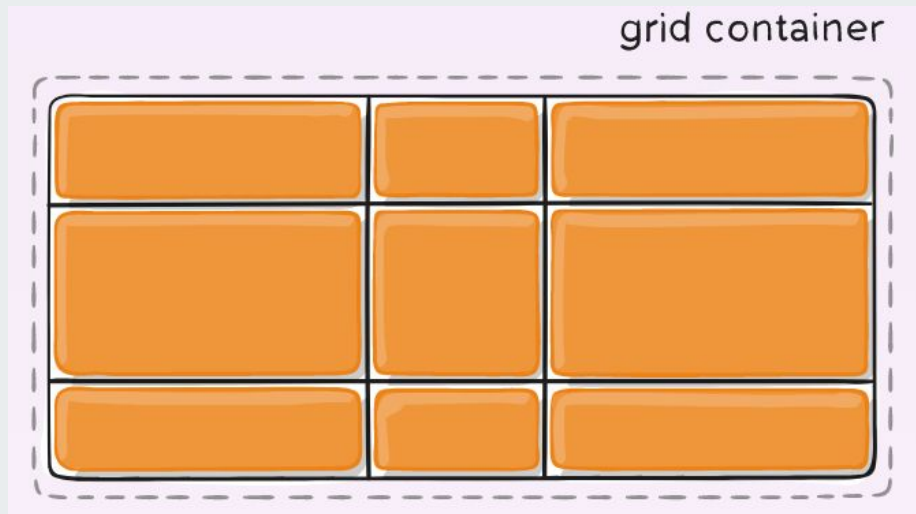
Justify Content III

```
.container {  
  justify-content: center;  
}
```



Justify Content II

```
.container {  
  justify-content: stretch;  
}
```



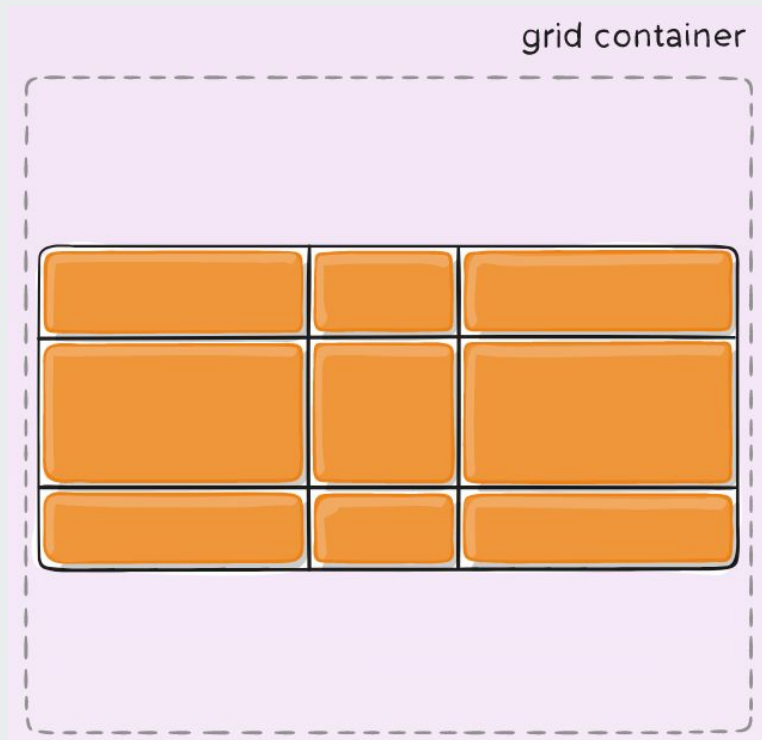


Align Content I

- A veces, el tamaño total de su cuadrícula puede ser menor que el tamaño de su contenedor de cuadrícula.
- Esto podría suceder si todos los elementos de la cuadrícula se dimensionan con unidades no flexibles como px.
- En este caso, puede establecer la alineación de la cuadrícula dentro del contenedor de la cuadrícula.
- Esta propiedad alinea la cuadrícula a lo largo del eje del bloque (columna) (en oposición al contenido de justificación que alinea la cuadrícula a lo largo del eje en línea (fila)).
- Los posibles valores son: start, end, center, stretch, space-around, space-between, space-evenly.

Align Content II

```
.container {  
  align-content: center;  
}
```



Template Areas I

- La propiedad CSS grid-template-areas especifica nombres para cada una de las grid areas.
- Es decir, podemos nombrar las áreas de nuestra grilla

```
<div class="grid-container">  
  <div class="item1">Header</div>  
  <div class="item2">Menu</div>  
  <div class="item3">Main</div>  
  <div class="item4">Right</div>  
  <div class="item5">Footer</div>  
</div>
```

Template Areas II

```
.grid-container {  
  display: grid;  
  grid-template-areas:  
    'header header header header header header'  
    'menu main main main right right'  
    'menu footer footer footer footer footer';  
  grid-gap: 10px;  
  background-color: #2196F3;  
  padding: 10px;  
}
```

Template Areas III

