

Tutorial Completo de CSS GRID - De Cero a Experto

ASISTENCIA SOBRE APLICACIONES ESPECIFICAS - PROFE:
AGUSTIN GATICA

ÍNDICE

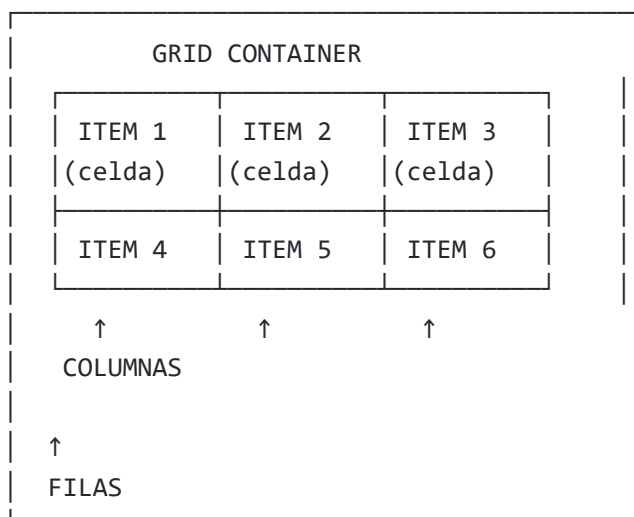
1. Conceptos Básicos
2. Propiedades del Contenedor
3. Propiedades de los Items
4. Ejercicios Prácticos Resueltos
5. Proyectos Avanzados

CONCEPTOS BÁSICOS DE GRID

¿Qué es CSS Grid?

CSS Grid es un sistema de layout bidimensional (filas y columnas) que permite crear diseños complejos de forma sencilla.

Terminología Principal



2 PROPIEDADES DEL CONTENEDOR GRID

Activar Grid

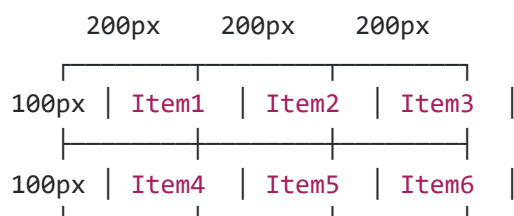
```
.container {  
  display: grid;  
}
```

Definir Filas y Columnas

A. grid-template-columns y grid-template-rows

```
.container {  
  display: grid;  
  grid-template-columns: 200px 200px 200px;  
  grid-template-rows: 100px 100px;  
}
```

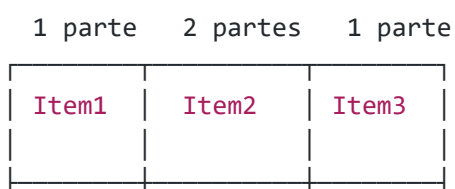
Esquema Visual:



B. Usando Fracciones (fr)

```
.container {  
  display: grid;  
  grid-template-columns: 1fr 2fr 1fr;  
  grid-template-rows: 100px 100px;  
}
```

Esquema Visual:





C. Función `repeat()`

```
.container {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  grid-template-rows: repeat(2, 100px);  
}
```

Es equivalente a: 1fr 1fr 1fr y 100px 100px

D. Función `minmax()`

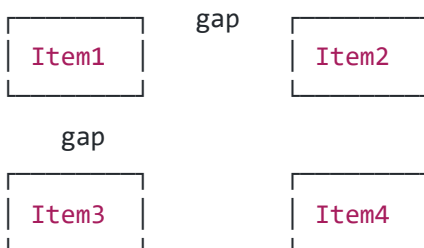
```
.container {  
  display: grid;  
  grid-template-columns: repeat(3, minmax(150px, 1fr));  
}
```

Mínimo 150px, máximo el espacio disponible.

Espaciado entre Celdas

```
.container {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  gap: 20px;           /* Espacio vertical y horizontal */  
  column-gap: 15px;    /* Solo columnas */  
  row-gap: 10px;       /* Solo filas */  
}
```

Esquema Visual:



Alineación del Contenido

```
.container {
  display: grid;
  justify-items: center; /* Alinea items horizontalmente */
  align-items: center; /* Alinea items verticalmente */
  justify-content: center; /* Alinea grid horizontalmente */
  align-content: center; /* Alinea grid verticalmente */
}
```

Valores posibles: start , center , end , stretch , space-between , space-around , space-evenly

3 PROPIEDADES DE LOS ITEMS

Posicionamiento de Items

```
.item1 {
  grid-column: 1 / 3; /* Ocupa de columna 1 a 3 */
  grid-row: 1 / 2; /* Ocupa fila 1 a 2 */
}
```

Esquema Visual:

	Col1	Col2	Col3
Fila1	Item1		Item3
Fila2	Item2	Item4	Item5

Usando span

```
.item1 {
  grid-column: span 2; /* Ocupa 2 columnas */
  grid-row: span 2; /* Ocupa 2 filas */
}
```

4 EJERCICIOS PRÁCTICOS RESUELTOS

EJERCICIO 1: Galería Simple 3x2

Objetivo: Crear una galería de 3 columnas y 2 filas

```

<!DOCTYPE html>
<html>
<head>
<style>
.galeria {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  gap: 20px;
  padding: 20px;
}

.foto {
  background: linear-gradient(45deg, #667eea 0%, #764ba2 100%);
  height: 200px;
  display: flex;
  align-items: center;
  justify-content: center;
  color: white;
  font-size: 18px;
  border-radius: 8px;
}
</style>
</head>
<body>

<div class="galeria">
  <div class="foto">Foto 1</div>
  <div class="foto">Foto 2</div>
  <div class="foto">Foto 3</div>
  <div class="foto">Foto 4</div>
  <div class="foto">Foto 5</div>
  <div class="foto">Foto 6</div>
</div>

</body>
</html>

```

Esquema:

Foto 1	Foto 2	Foto 3
Foto 4	Foto 5	Foto 6

EJERCICIO 2: Layout de Blog

Objetivo: Header, Sidebar, Contenido Principal, Footer

```
<!DOCTYPE html>
<html>
<head>
<style>
  body {
    margin: 0;
  }

  .blog-layout {
    display: grid;
    grid-template-columns: 250px 1fr;
    grid-template-rows: 80px 1fr 60px;
    gap: 20px;
    padding: 20px;
    min-height: 100vh;
  }

  header {
    grid-column: 1 / -1; /* Ocupa todas las columnas */
    background: #333;
    color: white;
    padding: 20px;
    border-radius: 8px;
  }

  aside {
    background: #f0f0f0;
    padding: 20px;
    border-radius: 8px;
  }

  main {
    background: white;
    padding: 20px;
    border: 1px solid #ddd;
    border-radius: 8px;
  }

  footer {
    grid-column: 1 / -1;
    background: #333;
    color: white;
    padding: 20px;
    border-radius: 8px;
  }
</style>
</head>
<body>

<div class="blog-layout">
  <header>
    <h1>Mi Blog</h1>
```

```

</header>

<aside>
  <h3>Categorías</h3>
  <ul>
    <li>CSS</li>
    <li>JavaScript</li>
    <li>HTML</li>
  </ul>
</aside>

<main>
  <h2>Artículo Principal</h2>
  <p>Contenido del artículo...</p>
</main>

<footer>
  <p>&copy; 2026 Mi Blog. Todos los derechos reservados.</p>
</footer>
</div>

</body>
</html>

```

Esquema:

HEADER (span 2)	
SIDEBAR (250px)	MAIN CONTENT (1fr)
FOOTER (span 2)	

EJERCICIO 3: Dashboard con Items Variables

Objetivo: Un dashboard donde algunos items ocupan más espacio

```

<!DOCTYPE html>
<html>
<head>
<style>
  .dashboard {
    display: grid;
    grid-template-columns: repeat(4, 1fr);

```

```

    grid-template-rows: repeat(3, 150px);
    gap: 15px;
    padding: 20px;
}

.widget {
    background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
    border-radius: 8px;
    padding: 20px;
    color: white;
    display: flex;
    align-items: center;
    justify-content: center;
    font-size: 18px;
    font-weight: bold;
}

.widget-large {
    grid-column: span 2;
    grid-row: span 2;
}

.widget-medium {
    grid-column: span 2;
}
</style>
</head>
<body>

<div class="dashboard">
  <div class="widget widget-large">Widget Grande</div>
  <div class="widget">Widget 1</div>
  <div class="widget">Widget 2</div>

  <div class="widget">Widget 3</div>
  <div class="widget">Widget 4</div>

  <div class="widget widget-medium">Widget Mediano</div>
  <div class="widget">Widget 5</div>
</div>

</body>
</html>

```

Esquema:

	W1	W2
WIDGET GRANDE	W3	W4
(2x2)		

	WIDGET MED
WIDGET MED	W5

EJERCICIO 4: Responsive Grid

Objetivo: Grid que se adapta a diferentes tamaños de pantalla

```
<!DOCTYPE html>
<html>
<head>
<style>
  * {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
  }

  .responsive-grid {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
    gap: 20px;
    padding: 20px;
  }

  .card {
    background: white;
    border-radius: 8px;
    overflow: hidden;
    box-shadow: 0 2px 10px rgba(0,0,0,0.1);
    transition: transform 0.3s;
  }

  .card:hover {
    transform: translateY(-5px);
  }

  .card-image {
    background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
    height: 200px;
  }

  .card-content {
    padding: 20px;
  }

  /* Responsive */
  @media (max-width: 768px) {
```

```

    .responsive-grid {
      grid-template-columns: repeat(auto-fit, minmax(150px, 1fr));
    }
  }

  @media (max-width: 480px) {
    .responsive-grid {
      grid-template-columns: 1fr;
    }
  }
</style>
</head>
<body>

<div class="responsive-grid">
  <div class="card">
    <div class="card-image"></div>
    <div class="card-content">
      <h3>Producto 1</h3>
      <p>Descripción del producto</p>
    </div>
  </div>

  <div class="card">
    <div class="card-image"></div>
    <div class="card-content">
      <h3>Producto 2</h3>
      <p>Descripción del producto</p>
    </div>
  </div>

  <div class="card">
    <div class="card-image"></div>
    <div class="card-content">
      <h3>Producto 3</h3>
      <p>Descripción del producto</p>
    </div>
  </div>

  <div class="card">
    <div class="card-image"></div>
    <div class="card-content">
      <h3>Producto 4</h3>
      <p>Descripción del producto</p>
    </div>
  </div>
</div>

</body>
</html>

```

Función auto-fit :

Desktop (ancho):

Card 1	Card 2	Card 3	Card 4
--------	--------	--------	--------

Tablet:

Card 1	Card 2
Card 3	Card 4

Mobile:

Card 1
Card 2
Card 3
Card 4

EJERCICIO 5: Portfolio Completo

Objetivo: Sitio de portfolio con múltiples secciones

```
<!DOCTYPE html>
<html>
<head>
<style>
  * {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
  }

  body {
    font-family: Arial, sans-serif;
    background: #f5f5f5;
  }

  .portfolio {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
    gap: 30px;
    padding: 40px;
    max-width: 1200px;
    margin: 0 auto;
  }
```

```

}

.project {
  display: grid;
  grid-template-rows: 250px auto auto;
  background: white;
  border-radius: 10px;
  overflow: hidden;
  box-shadow: 0 4px 15px rgba(0,0,0,0.1);
  transition: all 0.3s ease;
}

.project:hover {
  transform: translateY(-10px);
  box-shadow: 0 8px 25px rgba(0,0,0,0.15);
}

.project-image {
  background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
}

.project-title {
  padding: 20px;
  font-size: 20px;
  font-weight: bold;
  border-bottom: 2px solid #f0f0f0;
}

.project-description {
  padding: 20px;
  color: #666;
  line-height: 1.6;
}

/* Item especial que ocupa 2 columnas */
.project-featured {
  grid-column: span 2;
  display: grid;
  grid-template-columns: 1fr 1fr;
  grid-template-rows: auto auto;
}

.project-featured .project-image {
  grid-row: 1 / 3;
}

@media (max-width: 768px) {
  .project-featured {
    grid-column: span 1;
    grid-template-columns: 1fr;
  }

  .project-featured .project-image {

```

```
        grid-row: 1;
        height: 250px;
    }
}
</style>
</head>
<body>

<div class="portfolio">
    <div class="project project-featured">
        <div class="project-image"></div>
        <div class="project-title">Proyecto Destacado</div>
        <div class="project-description">
            Este es mi proyecto más importante. Un sitio web completo con todas las funcionalidades
        </div>
    </div>

    <div class="project">
        <div class="project-image"></div>
        <div class="project-title">Proyecto 1</div>
        <div class="project-description">Descripción corta del proyecto</div>
    </div>

    <div class="project">
        <div class="project-image"></div>
        <div class="project-title">Proyecto 2</div>
        <div class="project-description">Descripción corta del proyecto</div>
    </div>

    <div class="project">
        <div class="project-image"></div>
        <div class="project-title">Proyecto 3</div>
        <div class="project-description">Descripción corta del proyecto</div>
    </div>

    <div class="project">
        <div class="project-image"></div>
        <div class="project-title">Proyecto 4</div>
        <div class="project-description">Descripción corta del proyecto</div>
    </div>

    <div class="project">
        <div class="project-image"></div>
        <div class="project-title">Proyecto 5</div>
        <div class="project-description">Descripción corta del proyecto</div>
    </div>
</div>

</body>
</html>
```

Propiedades del Contenedor

Propiedad	Valor	Uso
display	grid	Activa Grid
grid-template-columns	200px 1fr 2fr	Ancho de columnas
grid-template-rows	100px 200px	Alto de filas
gap	20px	Espaciado entre celdas
justify-items	center	Alineación horizontal items
align-items	center	Alineación vertical items
justify-content	center	Alineación horizontal grid
align-content	center	Alineación vertical grid

Propiedades de los Items

Propiedad	Valor	Uso
grid-column	1 / 3	Posición y span columnas
grid-row	1 / 2	Posición y span filas
justify-self	center	Alineación horizontal item
align-self	center	Alineación vertical item

Funciones Útiles

repeat(3, 1fr)

/* Repite 3 veces 1fr */

minmax(150px, 1fr)

/* Mínimo 150px, máximo 1fr */

auto-fit

/* Ajusta items automáticamente */

auto-fill

/* Llena el espacio disponible */

TIPS AVANZADOS

1. Nombres de Líneas

```
.container {  
  grid-template-columns: [inicio] 1fr [centro] 1fr [fin] 1fr;  
}  
  
.item {  
  grid-column: inicio / centro;  
}
```

2. Grid implícito

```
.container {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  grid-auto-rows: 100px; /* Filas adicionales automáticas */  
}
```

3. Densidad de Grid

```
.container {  
  grid-auto-flow: dense; /* Rellena huecos automáticamente */  
}
```

CONCLUSIÓN

Cuándo usar Grid:

-  Layouts bidimensionales
-  Diseños complejos
-  Componentes de página completa
-  Dashboards y portales

Cuándo usar Flexbox:

-  Layouts unidimensionales
-  Componentes pequeños
-  Alineación de elementos