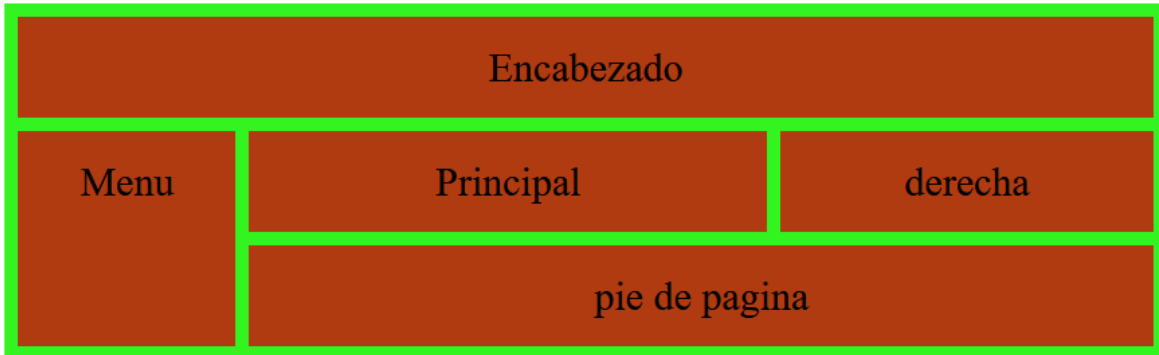


Módulo de diseño de CSS Grid



Diseño de cuadrícula

El módulo de CSS Grid ofrece un sistema de diseño basado en cuadrículas, con filas y columnas, lo que facilita el diseño de páginas web sin tener que usar flotadores y posicionamiento.

Elementos de CSS Grid

Un diseño de cuadrícula consta de un elemento principal, con uno o más elementos secundarios.

A	B	C
D	E	F
G	H	I

```
<!DOCTYPE html>
<html>
<head>
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto;
  background-color: #33f321;
  padding: 10px;
}

.grid-item{
  background-color: rgba(207, 12, 12, 0.8);
  border: 1px solid rgba(0, 0, 0, 0.8);
}
```

```

    text-align: center;
    padding: 20px 0;
    font-size: 30px;
}
</style>
</head>
<body>
<div class="grid-container">
  <div class="grid-item">A</div>
  <div class="grid-item">B</div>
  <div class="grid-item">C</div>
  <div class="grid-item">D</div>
  <div class="grid-item">E</div>
  <div class="grid-item">F</div>
  <div class="grid-item">G</div>
  <div class="grid-item">H</div>
  <div class="grid-item">I</div>
</div>

```

Propiedad de Display

Un elemento HTML se convierte en un contenedor de cuadrícula cuando su **display** propiedad se establece en **grid** o **inline-grid**.

```

.grid-container {
  display: grid;
  grid-template-columns: auto auto auto;
}

```

A	B	C
D	E	F
G	H	I

```

.grid-container {
  display: inline grid;
  grid-template-columns: auto auto auto;
}

```

A	B	C
D	E	F
G	H	I

Todos los hijos directos del contenedor de cuadrícula se convierten automáticamente en elementos de cuadrícula.

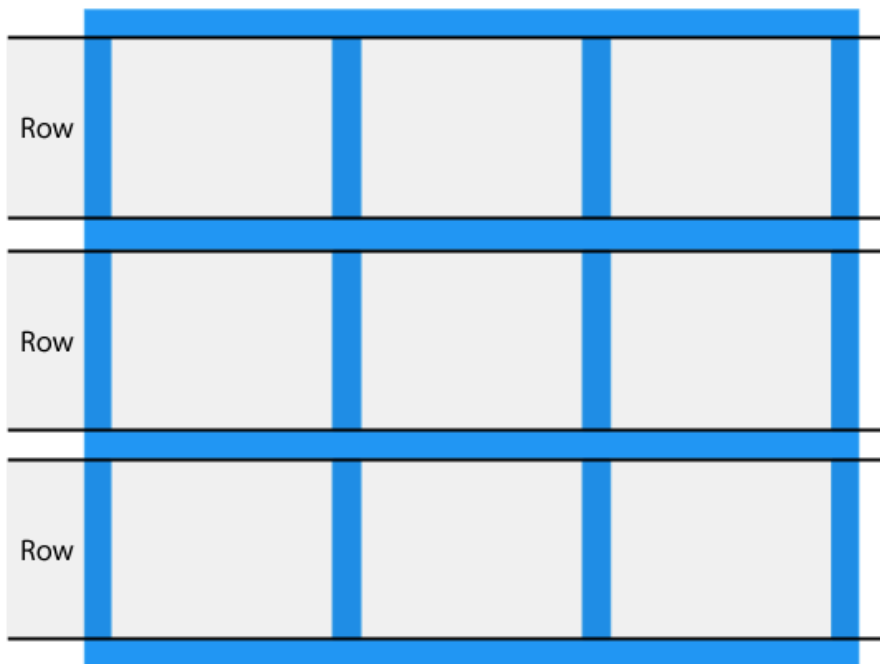
Columnas de cuadrícula (Grid Columns)

Las líneas verticales de los elementos de la cuadrícula se denominan columnas.

Column	Column	Column

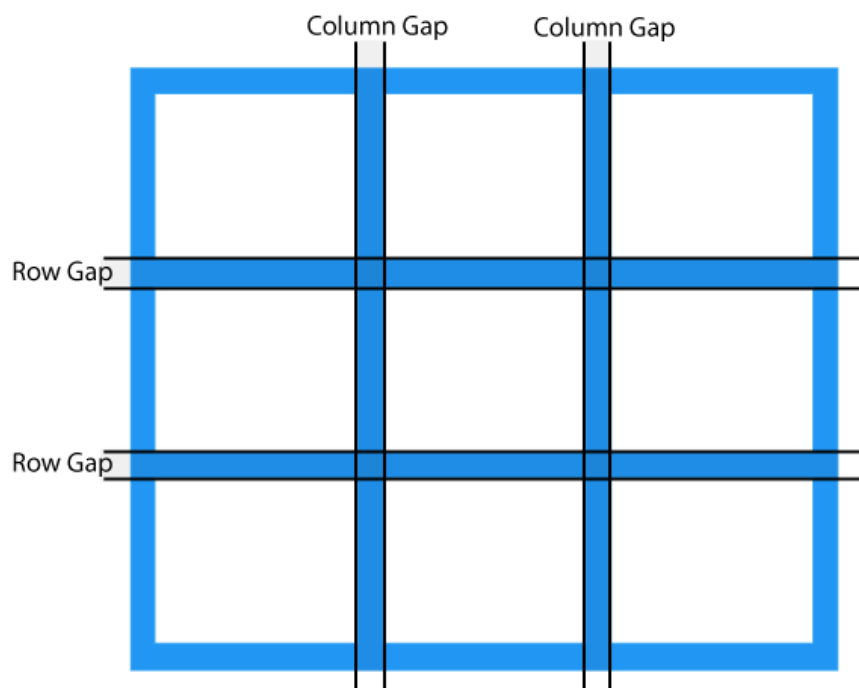
Filas de cuadrícula (Grid Rows)

Las líneas horizontales de los elementos de la cuadrícula se denominan filas.



Brechas de cuadrícula (Grid Gaps)

Los espacios entre cada columna / fila se denominan espacios.

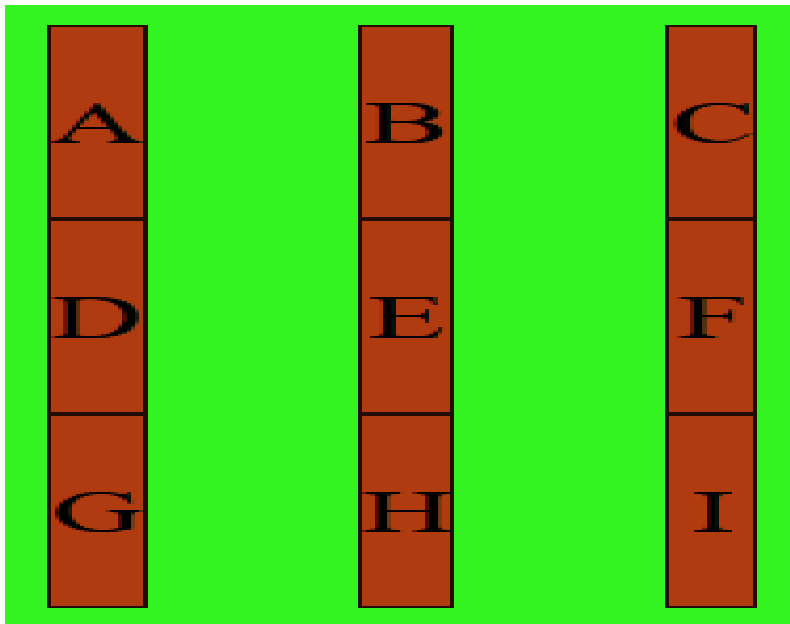


Se puede ajustar el tamaño del espacio mediante una de las siguientes propiedades:

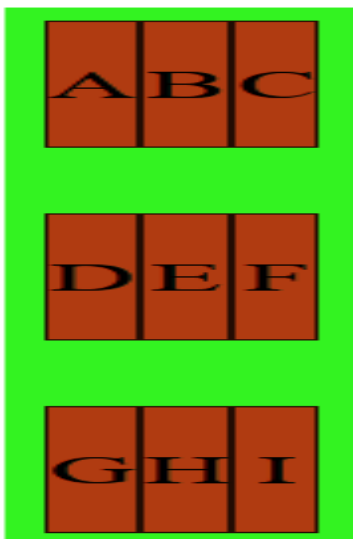
- `grid-column-gap`
- `grid-row-gap`
- `grid-gap`

grid-column-gap establece el espacio entre las columnas:

```
.grid-container {  
  display: inline grid;  
  grid-column-gap: 50px;  
  grid-template-columns: auto auto auto;  
}
```

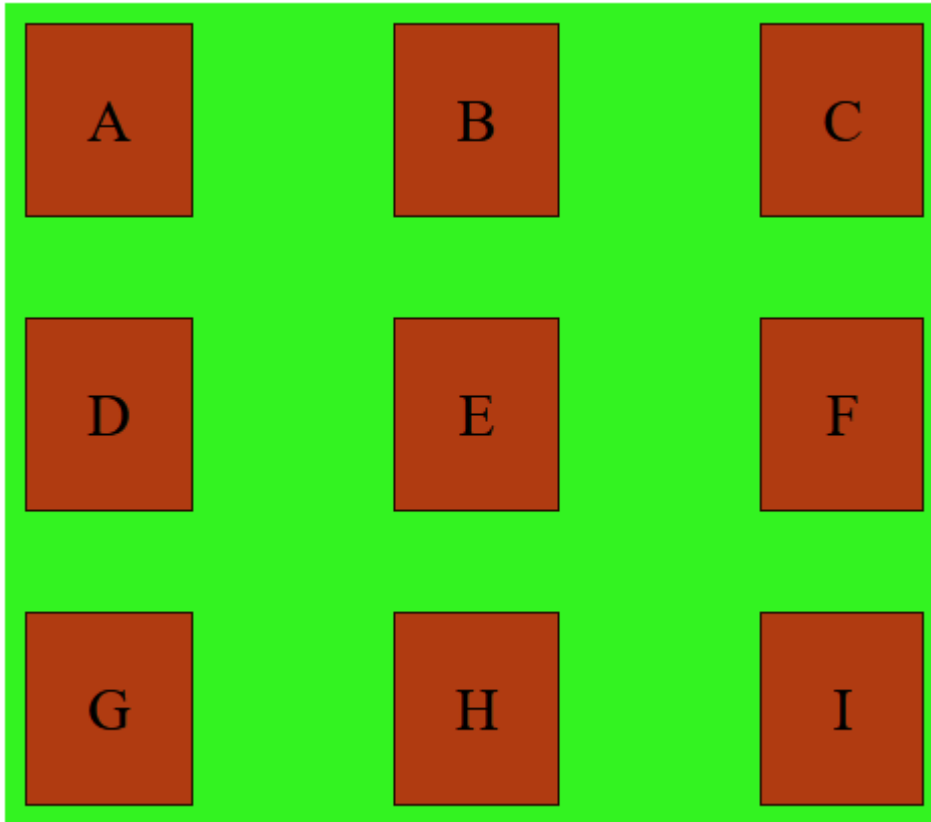


grid-row-gap establece el espacio entre las filas:



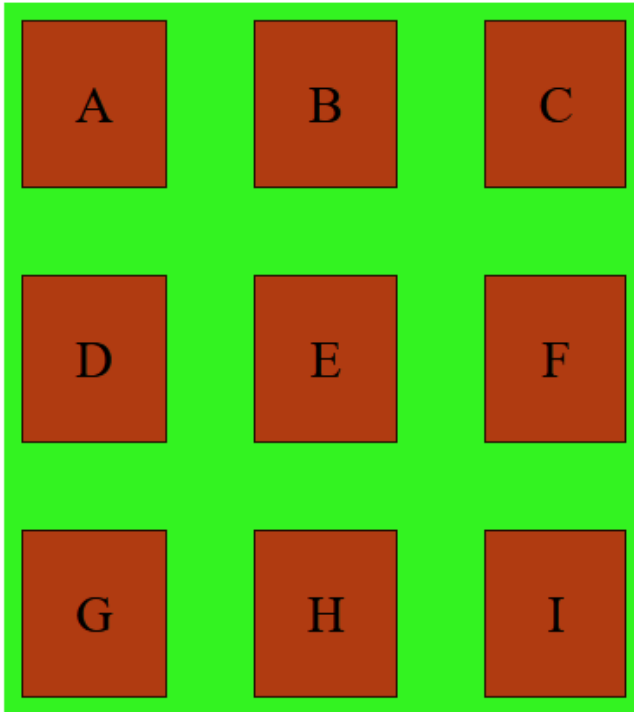
grid-gap es una propiedad abreviada de **grid-row-gap** y las **grid-column-gap**:

```
.grid-container {  
  display: inline grid;  
  grid-gap: 50px 100px;  
  grid-template-columns: auto auto auto;  
}
```



grid-gap también se puede usar para establecer tanto el espacio entre filas como el espacio entre columnas en un valor:

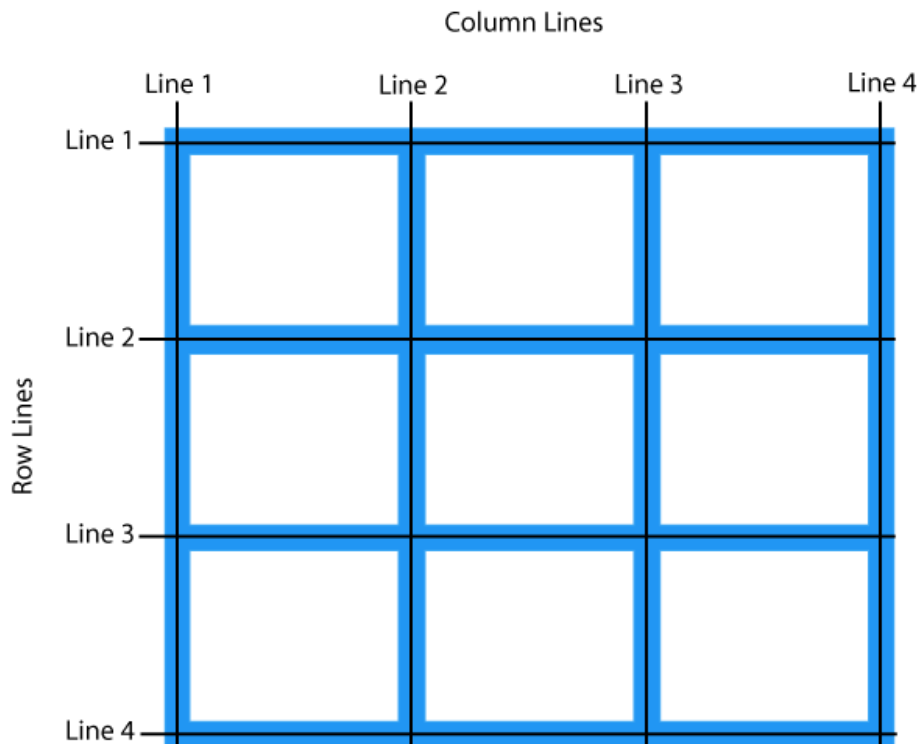
```
.grid-container {  
  display: inline grid;  
  grid-gap: 50px;  
  grid-template-columns: auto auto auto;  
}
```



Líneas de cuadrícula (Grid Lines)

Las líneas entre columnas se denominan líneas de columna.

Las líneas entre filas se denominan líneas de fila.



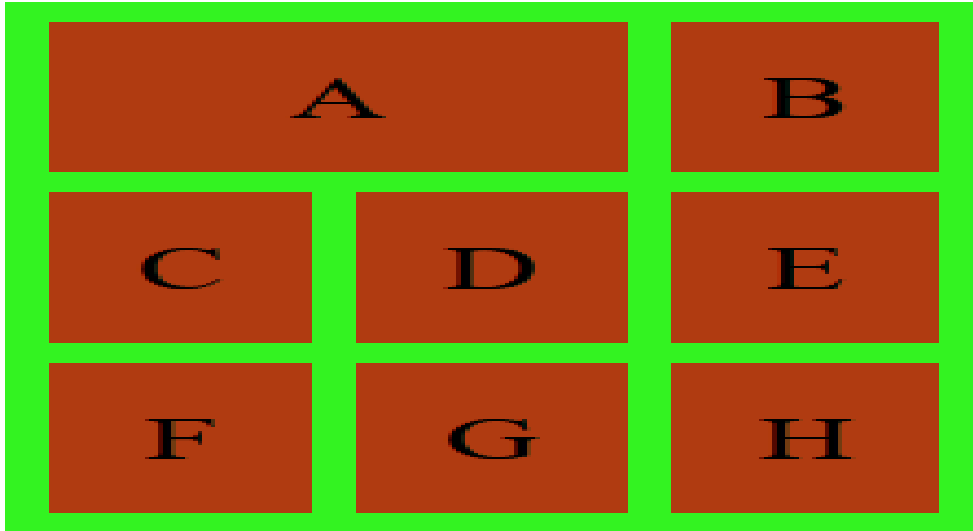
Consulte los números de línea cuando coloque un elemento de cuadrícula en un contenedor de cuadrícula:

```
<!DOCTYPE html>
<html>
<head>
<style>
.grid-container {
  display: inline grid;
  grid-gap: 10px;
  grid-template-columns: auto auto auto;
  background-color: #33f321;
  padding: 10px;
}

.grid-container> div{
  background-color: rgba(207, 12, 12, 0.8);
  text-align: center;
  padding: 20px ;
  font-size: 30px;
}

.item1{
  grid-column-start: 1;
  grid-column-end: 3;
}
</style>
</head>
<body>
<div class="grid-container">
  <div class="item1">A</div>
  <div class="item2">B</div>
  <div class="item3">C</div>
  <div class="item4">D</div>
  <div class="item5">E</div>
  <div class="item6">F</div>
  <div class="item7">G</div>
  <div class="item8">H</div>
</div>

</body>
</html>
```

Contenedor de cuadrícula CSS (CSS Grid Container)

A	B	C
D	E	F
G	H	I

Grid Container

Para que un elemento HTML se comporte como un contenedor de cuadrícula, debe establecer la propiedad **display** en **grid** o **inline-grid**.

Los contenedores Grid consisten en elementos de rejilla, colocados dentro de columnas y filas.

La propiedad **grid-template-columns**

grid-template-columns define el número de columnas en su diseño de cuadrícula y puede definir el ancho de cada columna.

El valor es una lista separada por espacios, donde cada valor define el ancho de la columna respectiva.

Si desea que su diseño de cuadrícula contenga 4 columnas, especifique el ancho de las 4 columnas, o "auto" si todas las columnas deben tener el mismo ancho.

```
.grid-container {
```

```
display: grid;
grid-gap: 10px;
grid-template-columns: 80px 300px auto 10px;
```

A	B	C	I
E	F	G	I

Nota: Si tiene más de 4 elementos en una cuadrícula de 4 columnas, la cuadrícula agregará automáticamente una nueva fila para colocar los elementos.

grid-template-columns también se puede utilizar para especificar el tamaño (width) de las columnas.

La propiedad grid-template-rows

grid-template-rows define la altura de cada fila.

```
.grid-container {
  display: grid;
  grid-gap: 10px;
  grid-template-columns: auto auto auto auto;
  grid-template-rows: 100px 200px;
```

A	B	C	D
E	F	G	H

El valor es una lista separada por espacios, donde cada valor define la altura de la fila respectiva:

propiedad justify-content

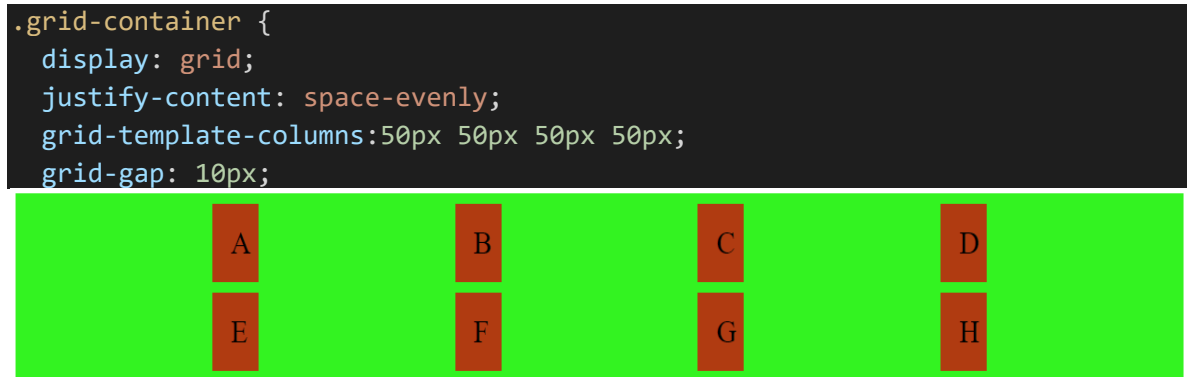
justify-content se utiliza para alinear toda la cuadrícula dentro del contenedor.

Nota: El ancho total de la cuadrícula debe ser menor que el ancho del contenedor para que la **justify-content** tenga algún efecto.

Espacio uniformemente **justify-content**: space-evenly

Utilice la propiedad *justify-content* para alinear la cuadrícula dentro del contenedor.

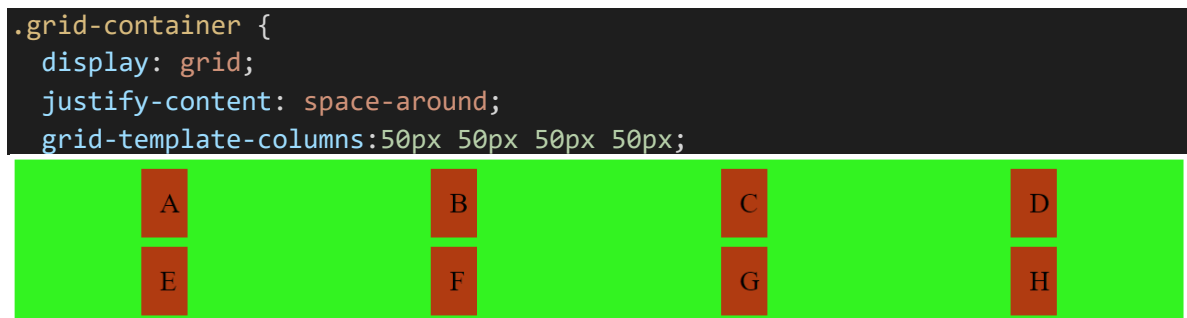
El valor " **space-evenly** " dará a las columnas la misma cantidad de espacio entre ellas y alrededor de ellas:



Espacio alrededor **justify-content: space-around**

Utilice la propiedad ***justify-content*** para alinear la cuadrícula dentro del contenedor.

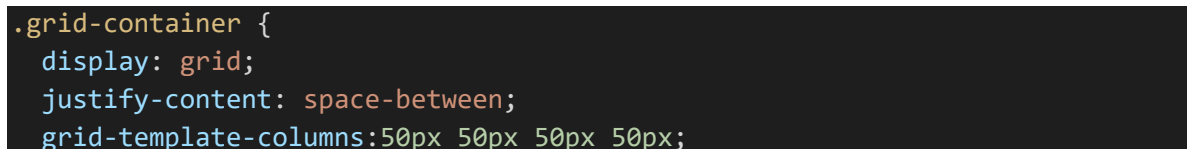
El valor " **space-around** " dará a las columnas la misma cantidad de espacio alrededor de ellas:



Espacio entre **justify-content: space-between**

Utilice la propiedad ***justify-content*** para alinear la cuadrícula dentro del contenedor.

El valor " **space-between** " dará a las columnas la misma cantidad de espacio entre ellas:



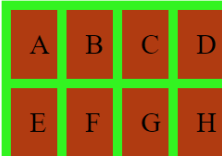


Inicio `justify-content: start`

Utilice la propiedad `justify-content` para alinear la cuadrícula dentro del contenedor.

El valor " `start` " alinearé la cuadrícula al comienzo del contenedor:

```
.grid-container {  
  display: grid;  
  justify-content: start;  
  grid-template-columns: 50px 50px 50px 50px;  
}
```



Final `justify-content: end`

Utilice la propiedad `justify-content` para alinear la cuadrícula dentro del contenedor.

El valor " `end` " alinearé la cuadrícula al final del contenedor:

```
.grid-container {  
  display: grid;  
  justify-content: end;  
  grid-template-columns: 50px 50px 50px 50px;  
}
```



Propiedad `align-content`

La `align-content` se utiliza para alinear *verticalmente* toda la cuadrícula dentro del contenedor.

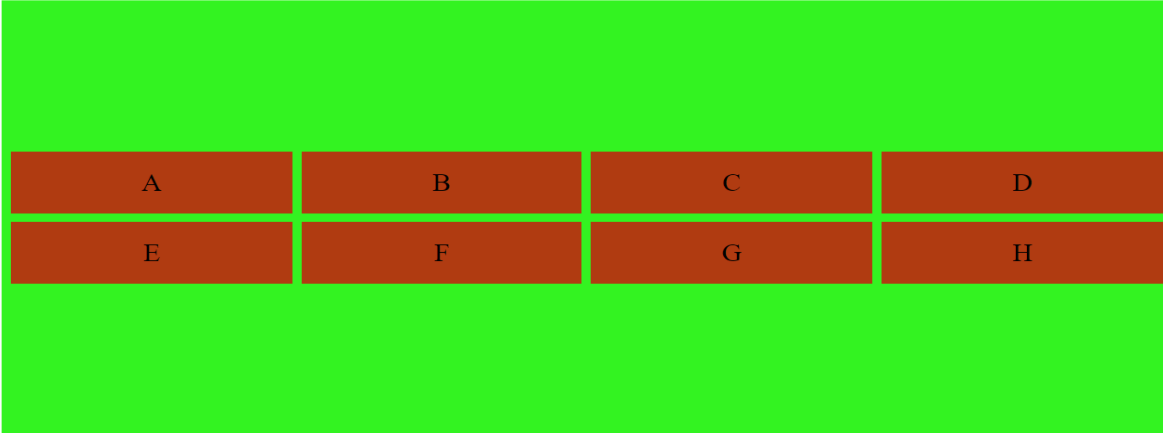
Nota: La altura total de la cuadrícula debe ser menor que la altura del contenedor para que la `align-content` tenga algún efecto.

Centrado `align-content: center`

Utilice la propiedad ***align-content*** para alinear verticalmente la cuadrícula dentro del contenedor.

El valor " **center**" alineará las filas en el medio del contenedor:

```
.grid-container {  
  display: grid;  
  height: 500px;  
  align-content: center;  
  grid-template-columns:auto auto auto auto;  
}
```

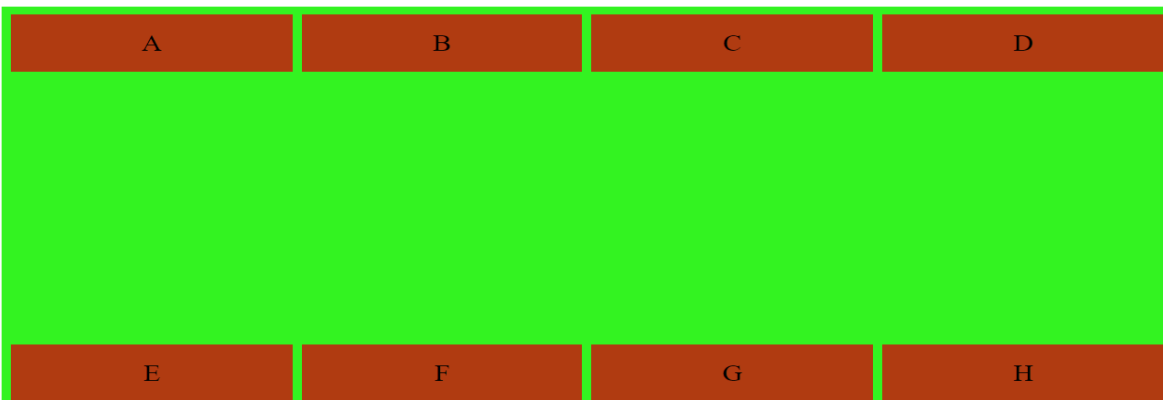


Espacio entre ***align-content***: **space-between**

Utilice la propiedad ***align-content*** para alinear verticalmente la cuadrícula dentro del contenedor.

El valor " **space-between** " dará a las filas la misma cantidad de espacio entre ellas:

```
.grid-container {  
  display: grid;  
  height: 500px;  
  align-content: space-between;  
  grid-template-columns:auto auto auto auto;  
}
```

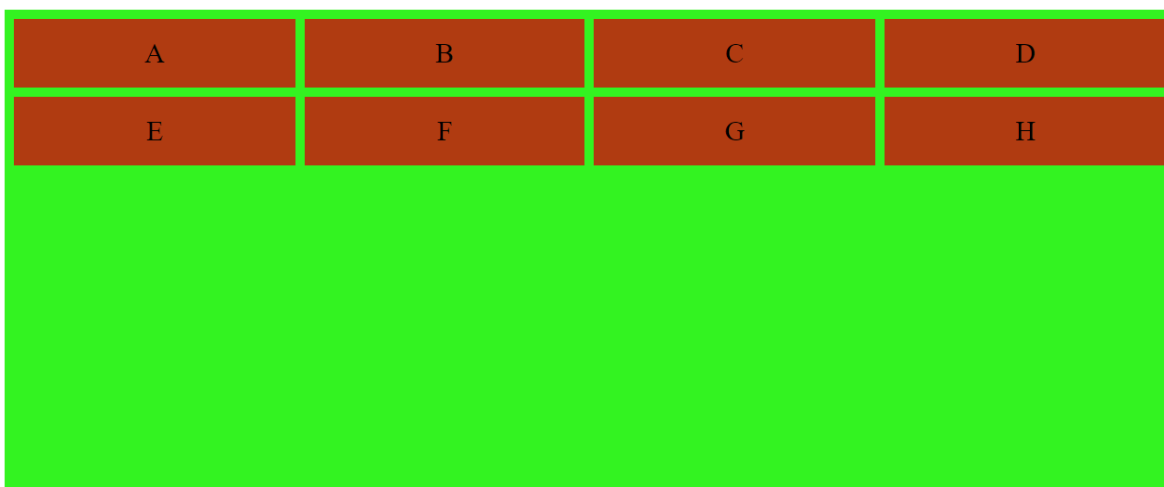


Inicio `align-content: start`

Utilice la propiedad `align-content` para alinear verticalmente la cuadrícula dentro del contenedor.

El valor "`start`" alineará las filas al comienzo del contenedor:

```
.grid-container {  
  display: grid;  
  height: 500px;  
  align-content: start;  
  grid-template-columns:auto auto auto auto;  
}
```

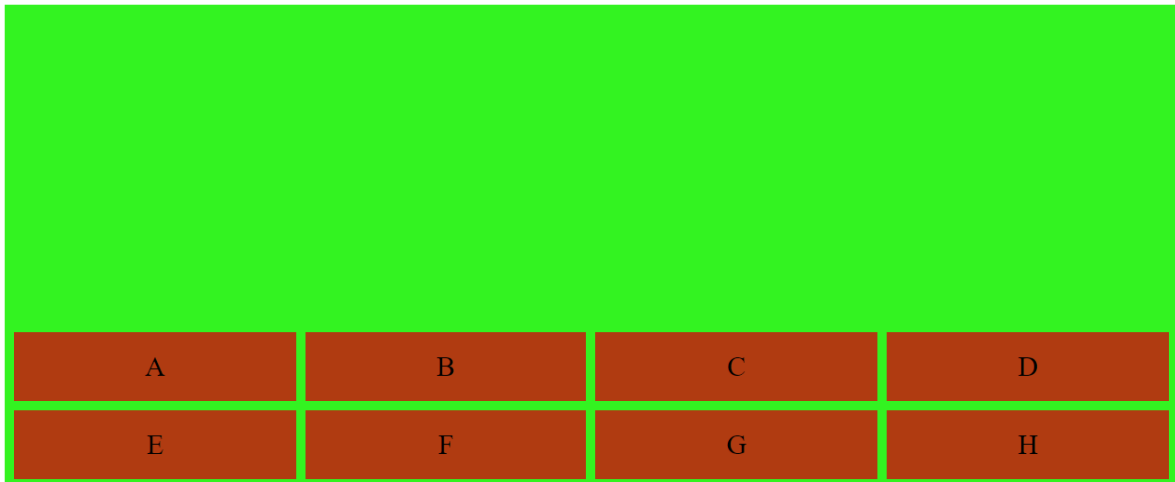


Fin `align-content: end`

Utilice la propiedad `align-content` para alinear verticalmente la cuadrícula dentro del contenedor.

El valor "`end`" alineará las filas al final del contenedor:

```
.grid-container {  
  display: grid;  
  height: 500px;  
  align-content: end;  
  grid-template-columns:auto auto auto auto;  
}
```



Elemento de CSS Grid

```
<!DOCTYPE html>
<html>
<head>
<style>
.grid-container {
  display: grid;
  grid-gap: 10px;
  background-color: #33f321;
  padding: 10px;
}

.grid-container> div{
  background-color: rgba(207, 12, 12, 0.8);
  text-align: center;
  padding: 20px ;
  font-size: 30px;
}

.elemento1{
  grid-column: 1 / span 2;
  grid-row: 1;
}
.elemento2 {
  grid-column: 3;
  grid-row: 1 / span 2;
}

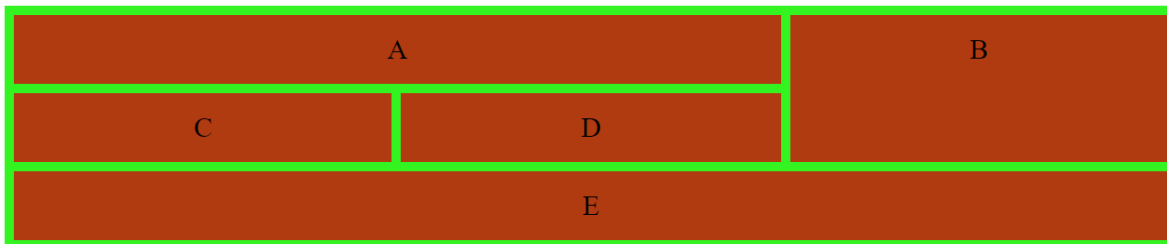
.elemento5 {
  grid-column: 1 / span 3;
```

```

    grid-row: 3;
}
</style>
</head>
<body>
<div class="grid-container">
  <div class="grid-item elemento1">A</div>
  <div class="grid-item elemento2">B</div>
  <div class="grid-item elemento3">C</div>
  <div class="grid-item elemento4">D</div>
  <div class="grid-item elemento5">E</div>
</div>

</body>
</html>

```



Elementos secundarios (Items)

Un *contenedor CSS Grid* contiene *elementos de cuadrícula*.

De forma predeterminada, un contenedor tiene un elemento de cuadrícula para cada columna, en cada fila, pero puede diseñar los elementos de la cuadrícula para que abarquen varias columnas y / o filas.

La propiedad de la columna de cuadrícula (**grid-column**)

grid-column define en qué columna (s) colocar un artículo.

Usted define dónde comenzará el artículo y dónde terminará el artículo.

Nota: La **grid-column** es una propiedad abreviada de **grid-column-start** y las **grid-column-end**.

Para colocar un elemento, puede hacer referencia a los *números de línea* o utilizar la palabra clave "span" para definir cuántas columnas abarcará el elemento.

Ejemplo

El elemento1 comienza en la columna 1 y termina antes de la columna 5:

```
<!DOCTYPE html>
<html>
<head>
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto auto auto auto;
  grid-gap: 10px;
  background-color: #33f321;
  padding: 10px;
}

.grid-container> div{
  background-color: rgba(207, 12, 12, 0.8);
  text-align: center;
  padding: 20px ;
  font-size: 30px;
}

.elemento1{
  grid-column: 1 / 5;
}

</style>
</head>
<body>
<div class="grid-container">
  <div class="grid-item elemento1">A</div>
  <div class="grid-item elemento2">B</div>
  <div class="grid-item elemento3">C</div>
  <div class="grid-item elemento4">D</div>
  <div class="grid-item elemento5">E</div>
  <div class="grid-item elemento6">A</div>
  <div class="grid-item elemento7">B</div>
  <div class="grid-item elemento8">C</div>
  <div class="grid-item elemento9">D</div>
  <div class="grid-item elemento10">E</div>
  <div class="grid-item elemento11">A</div>
  <div class="grid-item elemento12">B</div>
  <div class="grid-item elemento12">C</div>
  <div class="grid-item elemento14">D</div>
  <div class="grid-item elemento15">E</div>
</div>
```

```
</body>
</html>
```

A				B	C
D	E	A	B	C	D
E	A	B	C	D	E

Ejemplo

El elemento1 comienza en la columna 1 y abarqua 3 columnas:

```
<!DOCTYPE html>
<html>
<head>
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto auto auto auto;
  grid-gap: 10px;
  background-color: #33f321;
  padding: 10px;
}

.grid-container> div{
  background-color: rgba(207, 12, 12, 0.8);
  text-align: center;
  padding: 20px ;
  font-size: 30px;
}

.elemento1{
  grid-column: 1 / span 3;
}

</style>
</head>
<body>
<div class="grid-container">
  <div class="grid-item elemento1">A</div>
  <div class="grid-item elemento2">B</div>
  <div class="grid-item elemento3">C</div>
```

```

<div class="grid-item elemento4">D</div>
<div class="grid-item elemento5">E</div>
<div class="grid-item elemento6">A</div>
<div class="grid-item elemento7">B</div>
<div class="grid-item elemento8">C</div>
<div class="grid-item elemento9">D</div>
<div class="grid-item elemento10">E</div>
<div class="grid-item elemento11">A</div>
<div class="grid-item elemento12">B</div>
<div class="grid-item elemento12">C</div>
<div class="grid-item elemento14">D</div>
<div class="grid-item elemento15">E</div>
<div class="grid-item elemento16">F</div>
</div>

</body>
</html>

```

A			B	C	D
E	A	B	C	D	E
A	B	C	D	E	F

Ejemplo

El elemento2 comienza en la columna 2 y abarca 3 columnas:

```

.elemento2{
  grid-column: 2 / span 3;
}

```

A	B			C	D
E	A	B	C	D	E
A	B	C	D	E	F

La propiedad de la fila de la cuadrícula (**grid-row**)

grid-row define en qué fila colocar un artículo.

Usted define dónde comenzará el artículo y dónde terminará el artículo.

Nota: La **grid-row** es una propiedad abreviada de **grid-row-start** y **grid-row-end**.

Para colocar un elemento, puede hacer referencia a los *números de línea* o utilizar la palabra clave "span" para definir cuántas filas ocupará el elemento:

Ejemplo

El elemento1 comienza en la fila 1 y termina en la fila 4:

```
.elemento1 {  
    grid-row: 1 / 4;  
}
```

A	B	C	D	E	A
	B	C	D	E	A
	B	C	D	E	F

Ejemplo

El elemento1 comienza en la fila 1 y abarca 2 filas:

```
.elemento1 {  
    grid-row: 1 / span 2;  
}
```

A	B	C	D	E	A
	B	C	D	E	A
B	C	D	E	F	

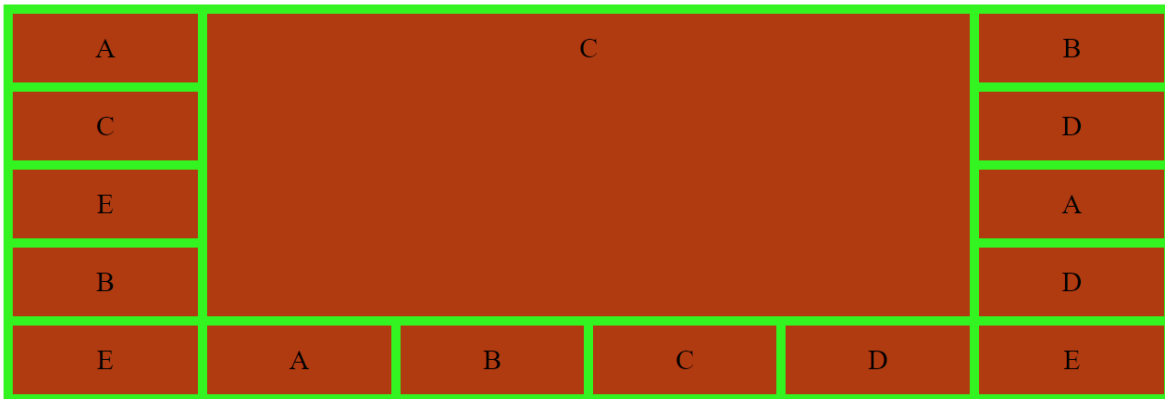
La propiedad del área de la cuadrícula (**grid-area**)

grid-area puede ser utilizada como una propiedad abreviada para el **grid-row-start**, **grid-column-start**, **grid-row-end** y las **grid-column-end**.

Ejemplo

El elemento8 comienza en la fila 1 y la columna 2, y termina en la fila 5 y la columna 6:

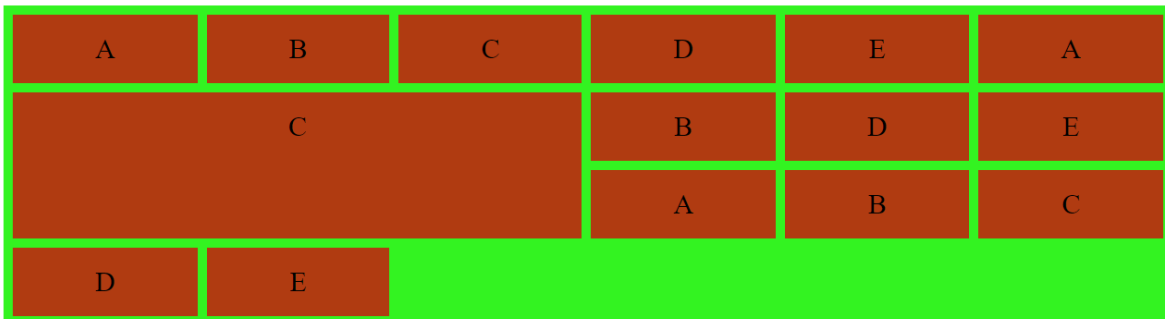
```
.elemento8 {
  grid-area: 1 /2 /5 /6;
}
```



Ejemplo

El elemento8 comienza en la línea de la fila 2 y la línea de la columna 1, y abarca 2 filas y 3 columnas:

```
.elemento8 {
  grid-area: 2 / 1 / span 2 / span 3;
}
```



Nombrar elementos de CSS Grid

La **grid-area** también se puede utilizar para asignar nombres a elementos de la cuadrícula.

Los elementos de CSS Grid con nombre se pueden denominar mediante la **grid-template-areas** del contenedor de cuadrícula.

Ejemplo

Item1 obtiene el nombre "myArea" y abarca las cinco columnas en un diseño de cuadrícula de cinco columnas:

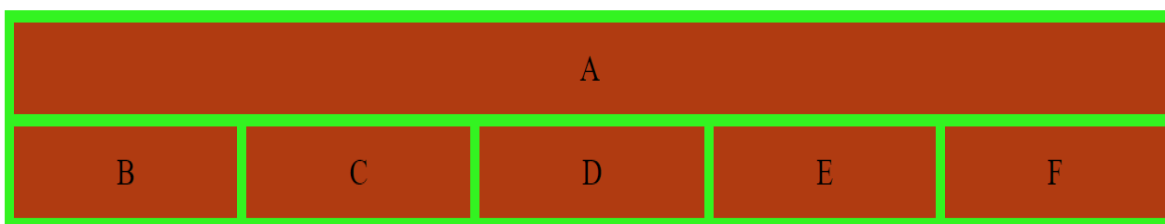
```
<!DOCTYPE html>
<html>
```

```
<head>
  <style>
    .item1{
      grid-area: myArea;
    }

    .grid-container {
      display: grid;
      grid-template-areas: 'myArea myArea myArea myArea myArea';
      grid-gap: 10px;
      background-color: #33f321;
      padding: 10px;
    }

    .grid-container>div {
      background-color: rgba(207, 12, 12, 0.8);
      text-align: center;
      padding: 20px 0;
      font-size: 30px;
    }
  </style>
</head>
<body>
  <div class="grid-container">
    <div class= "item1">A</div>
    <div class= "item2">B</div>
    <div class= "item3">C</div>
    <div class= "item4">D</div>
    <div class= "item5">E</div>
    <div class= "item6">F</div>

  </div>
</body>
</html>
```



Cada fila está definida por apóstrofes (")

Las columnas de cada fila se definen dentro de los apóstrofes, separadas por un espacio.

Nota: Un signo de punto representa un elemento de la cuadrícula sin nombre.

Ejemplo

"myArea" abarca dos columnas en un diseño de cuadrícula de cinco columnas (los signos de punto representan elementos sin nombre):

```
<head>
  <style>
    .item1 {
      grid-area: myArea;
    }

    .grid-container {
      display: grid;
      grid-template-areas: 'myArea myArea . . .';
      grid-gap: 10px;
      background-color: #33f321;
      padding: 10px;
    }

    .grid-container>div {
      background-color: rgba(207, 12, 12, 0.8);
      text-align: center;
      padding: 20px 0;
      font-size: 30px;
    }
  </style>
</head>

<body>
  <div class="grid-container">
    <div class="item1">A</div>
    <div class="item2">B</div>
    <div class="item3">C</div>
    <div class="item4">D</div>
    <div class="item5">E</div>
    <div class="item6">F</div>
    <div class="item6">G</div>
    <div class="item6">H</div>
```

```

        <div class="item6">I</div>
    </div>
</body>

</html>

```

A		B	C	D
E	F	G	H	I

Para definir dos filas, defina la columna de la segunda fila dentro de otro conjunto de apóstrofes:

Ejemplo

item1" abarca dos columnas y dos filas:

```

<!DOCTYPE html>
<html>

<head>
    <style>
        .item1 {
            grid-area: myArea;
        }

        .grid-container {
            display: grid;
            grid-template-
areas: 'myArea myArea . . .' 'myArea myArea . . .';
            grid-gap: 10px;
            background-color: #33f321;
            padding: 10px;
        }

        .grid-container>div {
            background-color: rgba(207, 12, 12, 0.8);
            text-align: center;
            padding: 20px 0;
            font-size: 30px;
        }
    </style>
</head>

```



```

<body>
  <div class="grid-container">
    <div class="item1">A</div>
    <div class="item2">B</div>
    <div class="item3">C</div>
    <div class="item4">D</div>
    <div class="item5">E</div>
    <div class="item6">F</div>
    <div class="item6">G</div>
  </div>
</body>

</html>

```

A	B	C	D
	E	F	G

Ejemplo

Nombra todos los elementos y crea una plantilla de página web lista para usar:

```

<!DOCTYPE html>
<html>

<head>
  <style>
    .item1 {
      grid-area: header;
    }

    .item2 {
      grid-area: menu;
    }

    .item3 {
      grid-area: main;
    }

    .item4 {
      grid-area: right;
    }

    .item5 {
      grid-area: footer;
    }
  </style>

```

```

    }

    .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: #33f321;
      padding: 10px;
    }

    .grid-container>div {
      background-color: rgba(207, 12, 12, 0.8);
      text-align: center;
      padding: 20px 0;
      font-size: 30px;
    }
  </style>
</head>

<body>
  <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>
</body>
</html>

```



El orden de los artículos

El diseño de cuadrícula nos permite colocar los elementos en cualquier lugar que queramos.

El primer elemento del código HTML no tiene que aparecer como el primer elemento de la cuadrícula

Ejemplo

```
<!DOCTYPE html>
<html>

<head>
  <style>
    .grid-container {
      display: grid;
      grid-template-areas: auto auto auto;
      grid-gap: 10px;
      background-color: #33f321;
      padding: 10px;
    }

    .grid-container>div {
      background-color: rgba(207, 12, 12, 0.8);
      text-align: center;
      padding: 20px 0;
      font-size: 30px;
    }

    .item1 {
      grid-area: 1 / 3 / 2 / 4;
    }

    .item2 {
      grid-area: 2 / 3 / 3 / 4;
    }

    .item3 {
      grid-area: 1 / 1 / 2 / 2;
    }

    .item4 {
      grid-area: 1 / 2 / 2 / 3;
    }
  
```

```

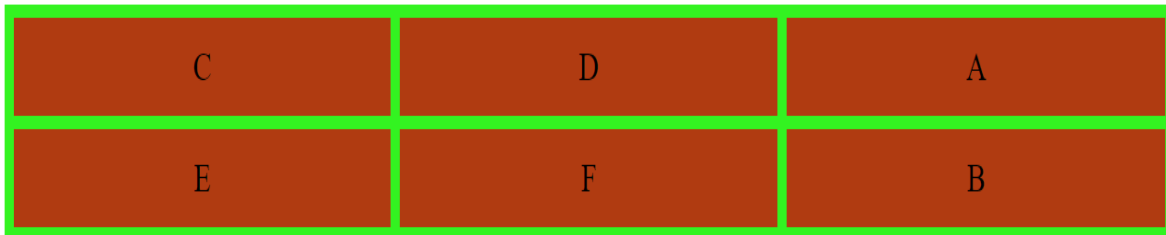
        .item5 {
            grid-area: 2 / 1 / 3 / 2;
        }

        .item6 {
            grid-area: 2 / 2 / 3 / 3;
        }
    </style>
</head>

<body>
    <div class="grid-container">
        <div class="item1">A</div>
        <div class="item2">B</div>
        <div class="item3">C</div>
        <div class="item4">D</div>
        <div class="item5">E</div>
        <div class="item6">F</div>
    </div>
</body>

</html>

```



Puede reorganizar el orden para ciertos tamaños de pantalla mediante consultas de medios:

Ejemplo

```

<!DOCTYPE html>
<html>

<head>
    <style>
        .grid-container {
            display: grid;
            grid-template-columns: auto auto auto;
            grid-gap: 10px;
            background-color: #33f321;

```

```

        padding: 10px;
    }

    .grid-container>div {
        background-color: rgba(207, 12, 12, 0.8);
        text-align: center;
        padding: 20px 0;
        font-size: 30px;
    }

    @media only screen and (max-width: 500px) {

        .item1 {
            grid-area: 1 / span 3 / 2 / 4;
        }

        .item2 {
            grid-area: 3 / 3 / 3 / 4;
        }

        .item3 {
            grid-area: 2 / 1 / 3 / 2;
        }

        .item4 {
            grid-area: 2 / 2 / span 2 / 3;
        }

        .item5 {
            grid-area: 3 / 1 / 4 / 2;
        }

        .item6 {
            grid-area: 2 / 3 / 3 / 4;
        }

    }

</style>
</head>

<body>
    <div class="grid-container">
        <div class="item1">A</div>
        <div class="item2">B</div>
        <div class="item3">C</div>
        <div class="item4">D</div>
        <div class="item5">E</div>

```

```

    <div class="item6">F</div>
  </div>
</body>

</html>

```

A	B	C
D	E	F