

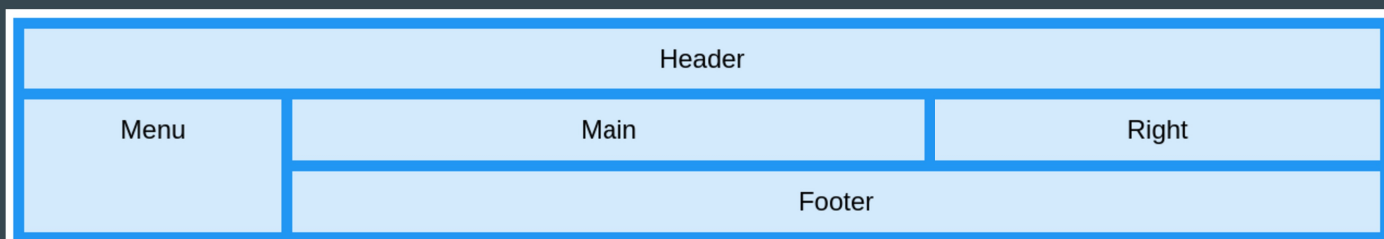


# CSS3 Grid

## Breve introducción

### ¿Qué es CSS3 Grid?

- Módulo que ofrece un sistema de layout basado en Grid.
  - Facilita el diseño de páginas web sin tener que usar **float** y **position**.
- Podríamos construir un layout como el siguiente, por ejemplo:



# CSS3 Grid

CSS3 Grid consta de un contenedor y una serie de elementos

```
<div class="contenedor">
  <div class="grid-item">1</div>
  <div class="grid-item">2</div>
  <div class="grid-item">3</div>
  <div class="grid-item">4</div>
  <div class="grid-item">5</div>
  <div class="grid-item">6</div>
  <div class="grid-item">7</div>
</div>
```

## CSS3 Grid: crear contenedor

```
<div class="contenedor">
  <div class="grid-item">1</div>
  <div class="grid-item">2</div>
  <div class="grid-item">3</div>
  <div class="grid-item">4</div>
  <div class="grid-item">5</div>
  <div class="grid-item">6</div>
  <div class="grid-item">7</div>
</div>
```

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



Convertir un elemento en  
contenedor Grid

# CSS3 Grid: añadir columnas

```
<div class="contenedor">
  <div class="grid-item">1</div>
  <div class="grid-item">2</div>
  <div class="grid-item">3</div>
  <div class="grid-item">4</div>
  <div class="grid-item">5</div>
  <div class="grid-item">6</div>
  <div class="grid-item">7</div>
</div>
```

Dos columnas de 100px

```
.contenedor {
  display: grid;
  grid-template-columns: 100px 100px;
}
```




# CSS3 Grid: especificar filas para altura sobre todo

```
<div class="contenedor">
  <div class="grid-item">1</div>
  <div class="grid-item">2</div>
  <div class="grid-item">3</div>
  <div class="grid-item">4</div>
  <div class="grid-item">5</div>
  <div class="grid-item">6</div>
  <div class="grid-item">7</div>
</div>
```

Cuatro filas de 50px

```
.contenedor {
  display: grid;
  grid-template-columns: 100px 100px;
  grid-template-rows: 50px 50px 50px 50px;
}
```



# Práctica Guiada

Veamos cómo funciona el ejemplo anterior

## CSS3 Grid: unidades para tamaños

**fr** → una fracción del tamaño disponible

**auto** → se adapta al tamaño del contenido

**%** → se ajusta en función del tamaño del contenedor

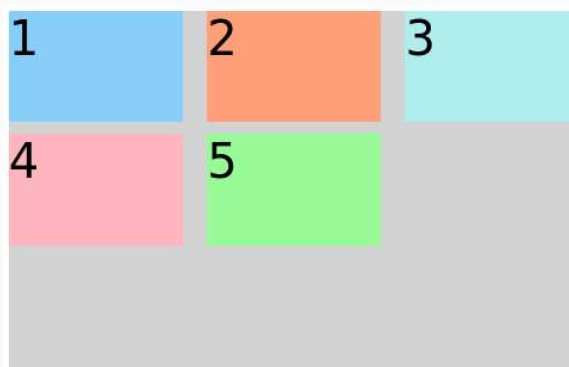
```
grid-template-columns: auto 50px 10% 2fr 1fr;
```

# CSS3 Grid: gap entre elementos

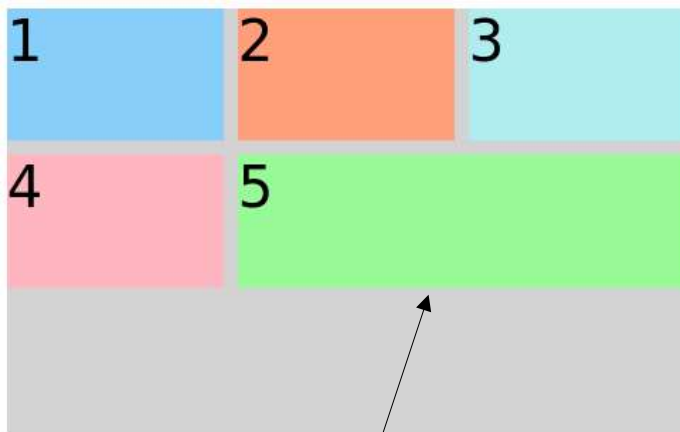
**grid-column-gap** → hueco entre columnas

**grid-row-gap** → hueco entre filas

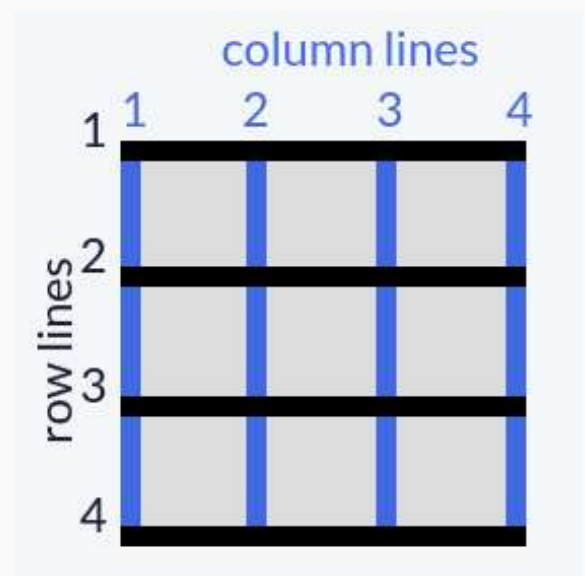
**grid-gap** → shorthand property (atajo)



## CSS3 Grid: control del espacio Ejemplo 1



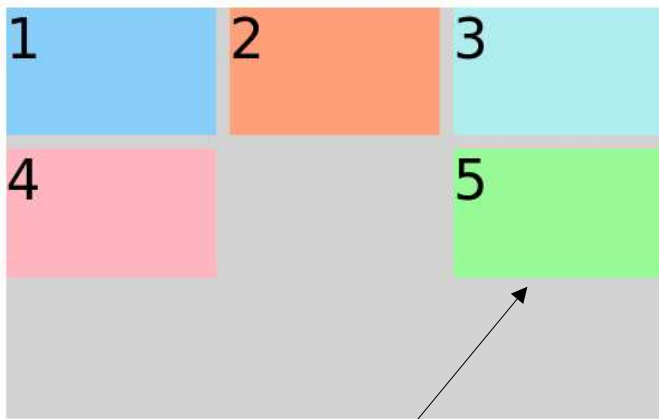
**grid-column: 2 / 4;**



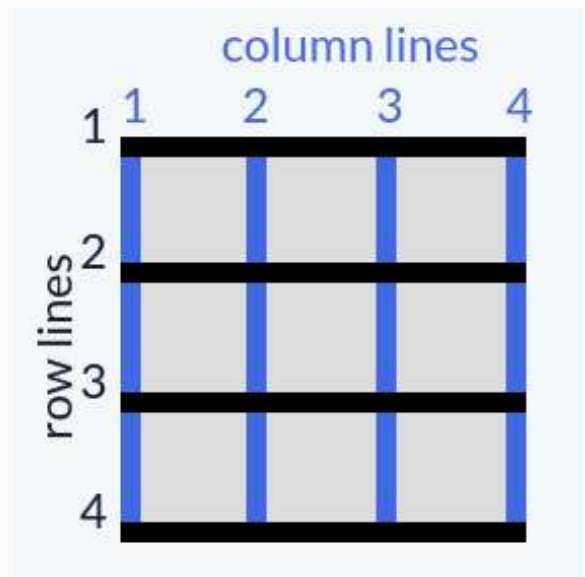
\* La propiedad grid-column se añade a los elementos hijo

# CSS3 Grid: control del espacio

## Ejemplo 2

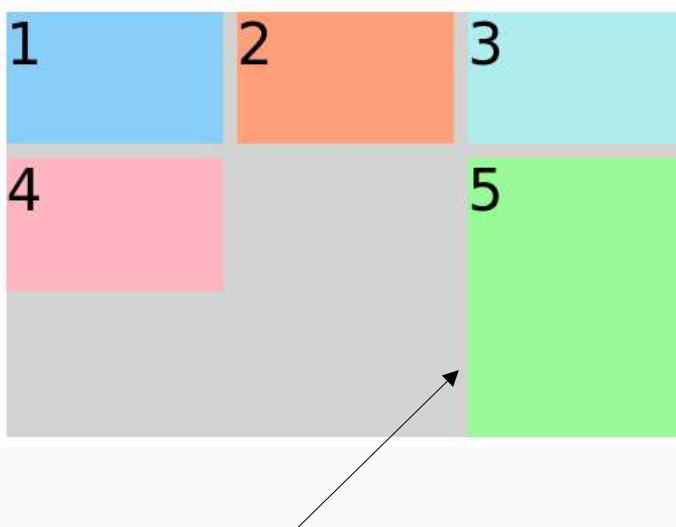


`grid-column: 3 / 4;`



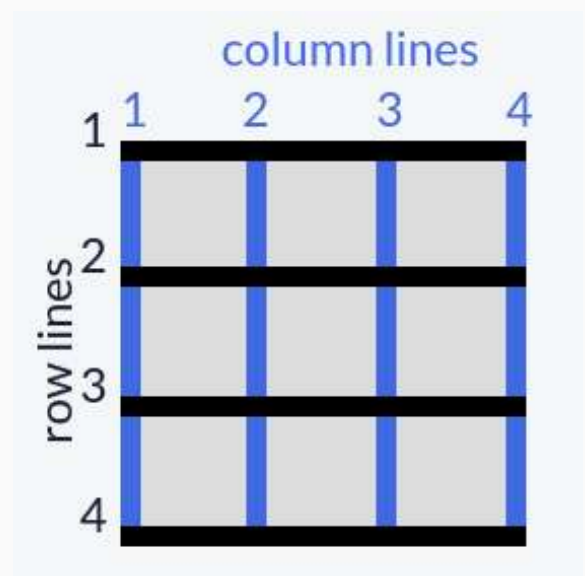
# CSS3 Grid: control del espacio

## Ejemplo 3



`grid-column: 3 / 4;`

`grid-row: 2 / 4;`



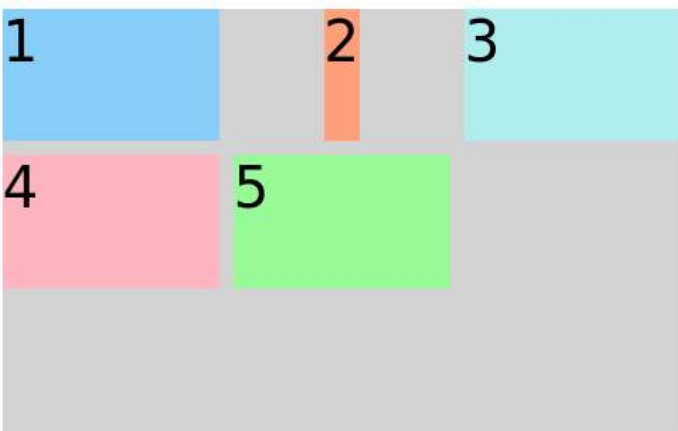
# Alineación Horizontal

`justify-self`  
`justify-items`

## CSS3 Grid: alineación horizontal

Para un elemento del grid:

```
justify-self: center;
```

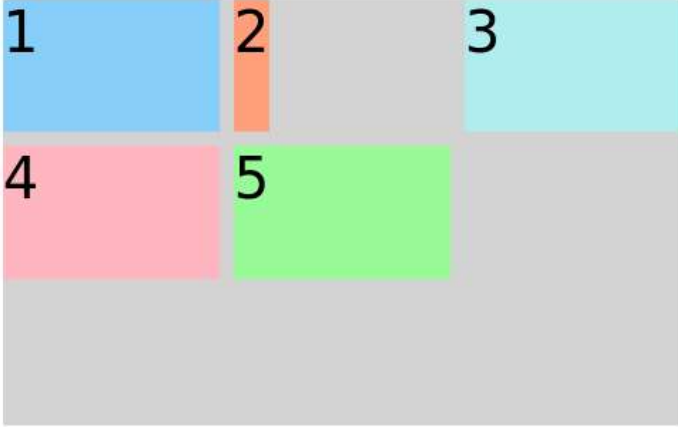


# CSS3 Grid: alineación horizontal

Para un elemento del grid:

```
justify-self: center;
```

```
justify-self: start;
```



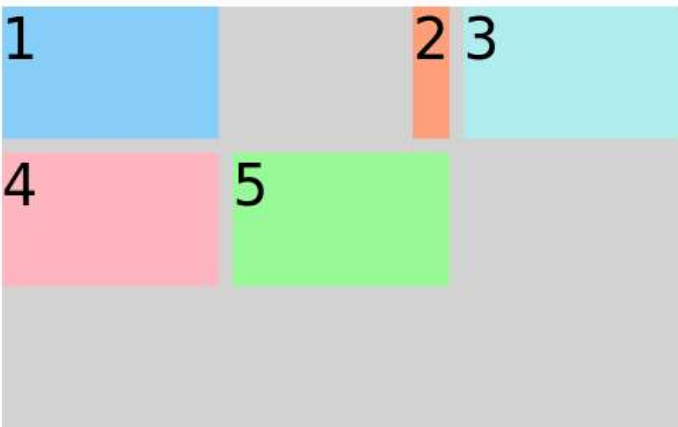
# CSS3 Grid: alineación horizontal

Para un elemento del grid:

```
justify-self: center;
```

```
justify-self: start;
```

```
justify-self: end;
```





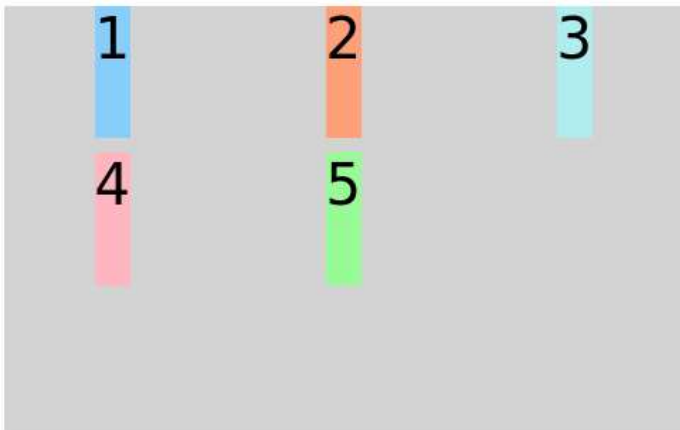
# CSS3 Grid: alineación horizontal

Para un elemento del grid:

```
justify-self: center;
```

```
justify-self: start;
```

```
justify-self: end;
```



Para todo el grid:

```
justify-items: center;
```

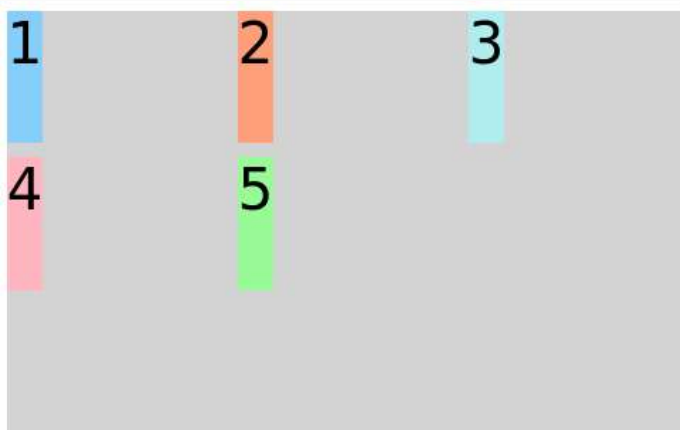
# CSS3 Grid: alineación horizontal

Para un elemento del grid:

```
justify-self: center;
```

```
justify-self: start;
```

```
justify-self: end;
```



Para todo el grid:

```
justify-items: center;
```

```
justify-items: start;
```

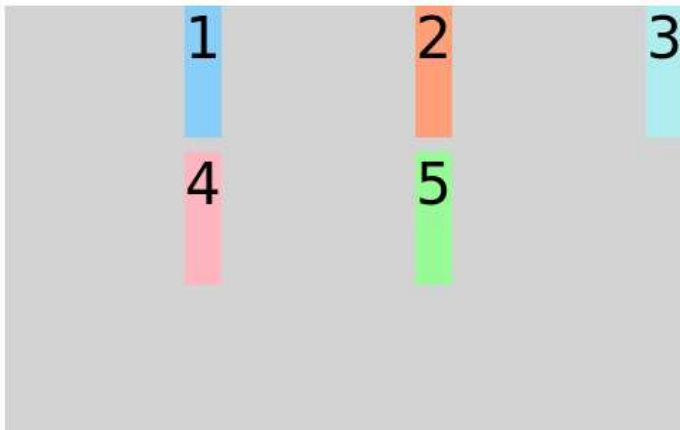
# CSS3 Grid: alineación horizontal

Para un elemento del grid:

```
justify-self: center;
```

```
justify-self: start;
```

```
justify-self: end;
```



Para todo el grid:

```
justify-items: center;
```

```
justify-items: start;
```

```
justify-items: end;
```

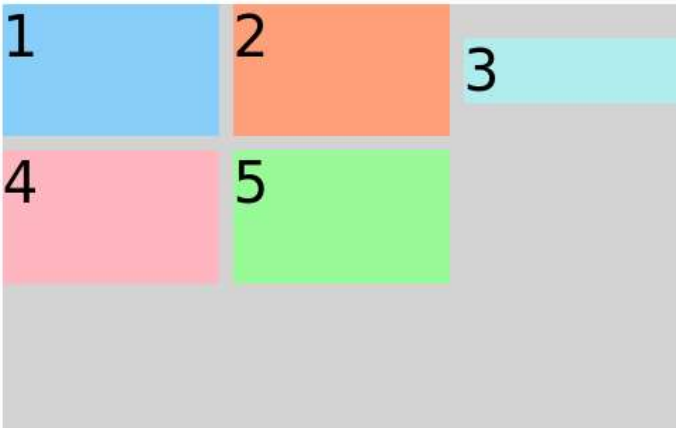
## Alineación Vertical

`align-self`  
`align-items`

# CSS3 Grid: alineación vertical

Para un elemento del grid:

```
align-self: center;
```

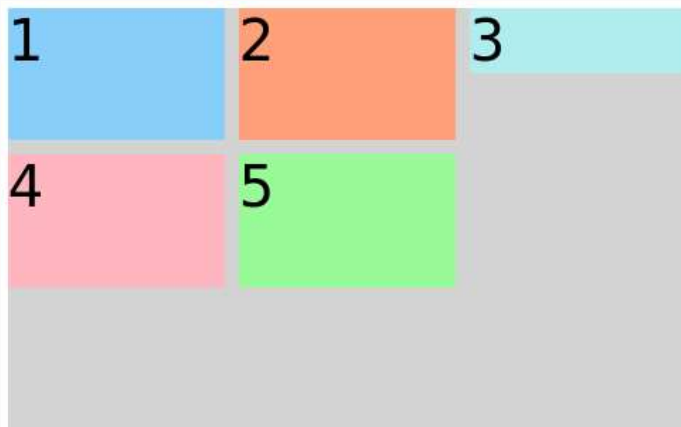


# CSS3 Grid: alineación vertical

Para un elemento del grid:

```
align-self: center;
```

```
align-self: start;
```



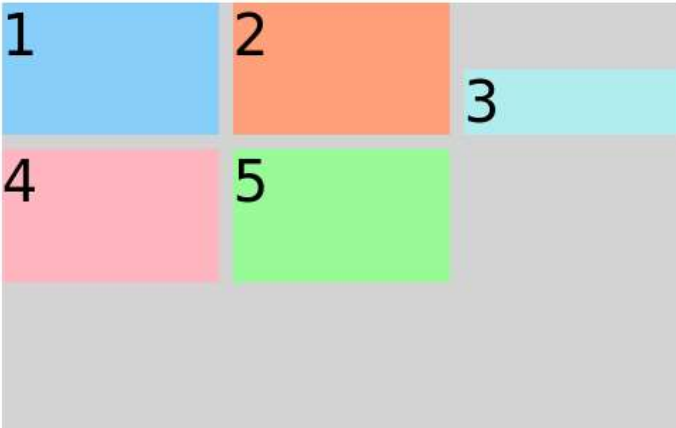
# CSS3 Grid: alineación vertical

Para un elemento del grid:

```
align-self: center;
```

```
align-self: start;
```

```
align-self: end;
```



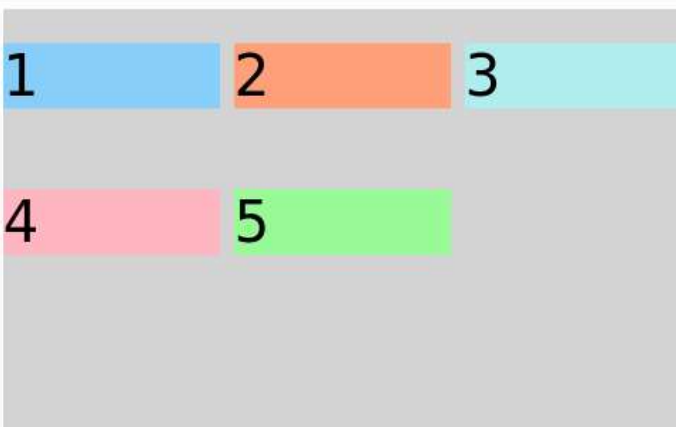
# CSS3 Grid: alineación vertical

Para un elemento del grid:

```
align-self: center;
```

```
align-self: start;
```

```
align-self: end;
```



Para todo el grid:

```
align-items: center;
```

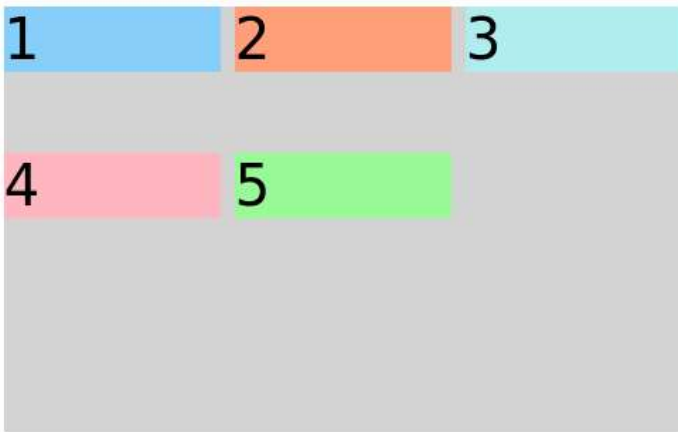
# CSS3 Grid: alineación horizontal

Para un elemento del grid:

```
align-self: center;
```

```
align-self: start;
```

```
align-self: end;
```



Para todo el grid:

```
align-items: center;
```

```
align-items: start;
```

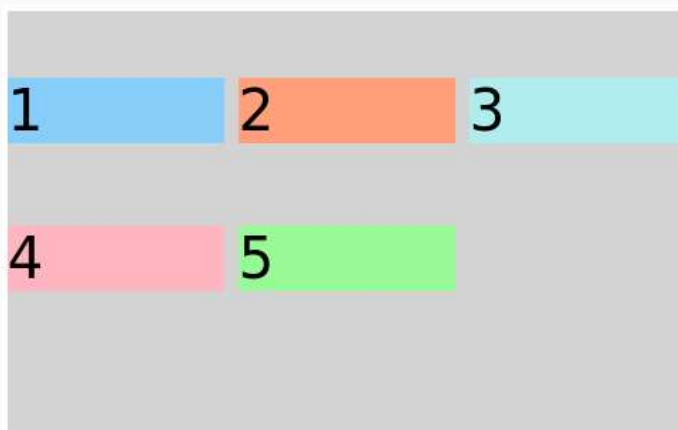
# CSS3 Grid: alineación horizontal

Para un elemento del grid:

```
align-self: center;
```

```
align-self: start;
```

```
align-self: end;
```



Para todo el grid:

```
align-items: center;
```

```
align-items: start;
```

```
align-items: end;
```