



CSS



GRID

Grid-template- areas

The **grid-template-areas** property specifies areas within the grid layout.

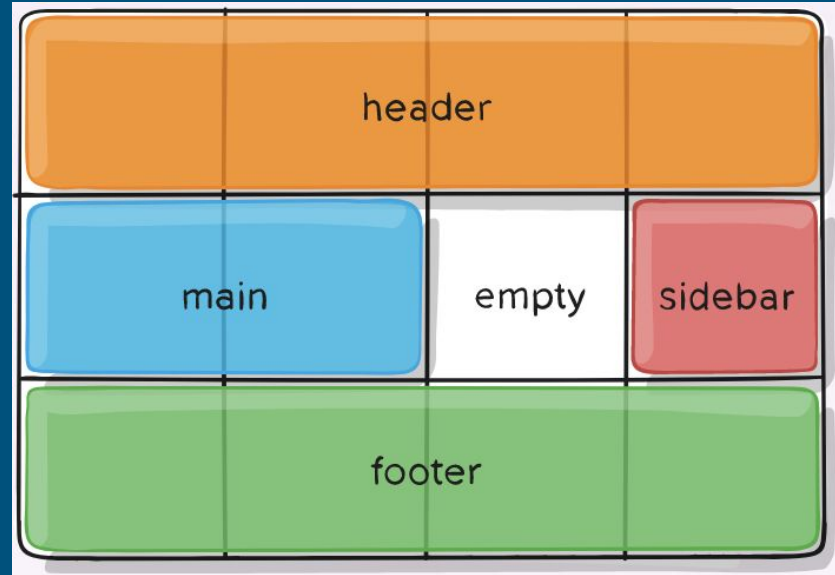
Repeating the name of a grid area causes the content to span those cells.

A period signifies an empty cell.

We use the **grid-template-areas** in the parent container and then assign them to each child with grid-area.

Grid-template-areas

```
#grid-container {  
  display: grid;  
  grid-template-columns: 1fr 1fr 1fr 1fr;  
  grid-template-rows: auto;  
  grid-template-areas:  
    "header header header header"  
    "main main . sidebar"  
    "footer footer footer footer";  
}
```



* You can use any number of adjacent periods to declare a single empty cell. As long as the periods have no spaces between them they represent a single cell.

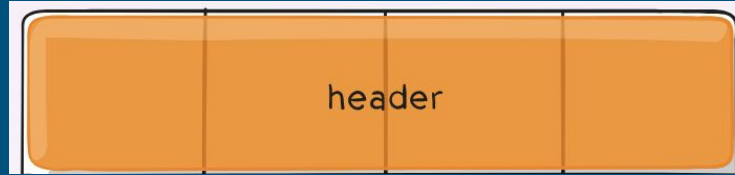
Grid-area

One way to use grid-area is by assigning a name grid items by using the **grid-area** property, and then reference to the name in the grid-template-areas property.

Grid areas must be rectangular/square as the css box model; it is not possible to create, for example, a T- or L-shaped **grid area**.

Grid-template-areas

```
header {  
  grid-area: header;  
}
```



```
main {  
  grid-area: main;  
}
```



```
.sidebar {  
  grid-area: sidebar;  
}
```

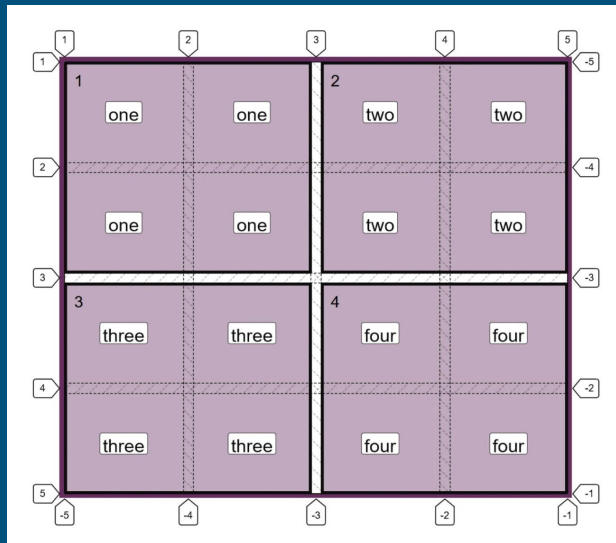


```
footer {  
  grid-area: footer;  
}
```



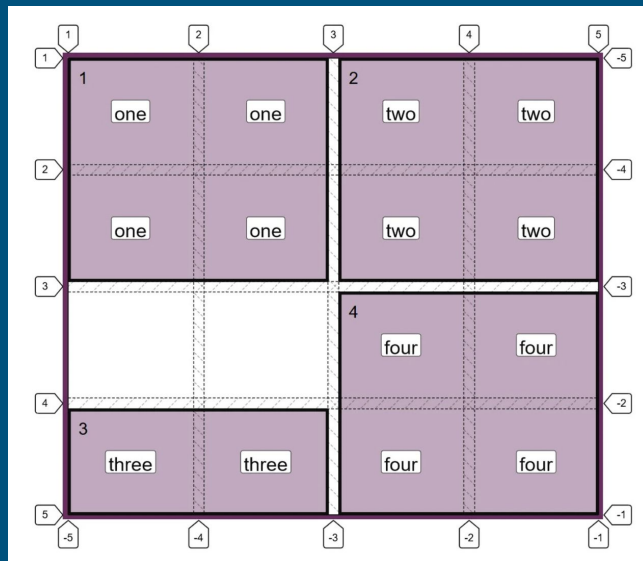
Example:

```
grid-template-areas: "one one two two"  
                    "one one two two"  
                    "three three four four"  
                    "three three four four";
```



Example:

```
grid-template-areas: "one one two two"  
                    "one one two two"  
                    ". . four four"  
                    "three three four four";
```

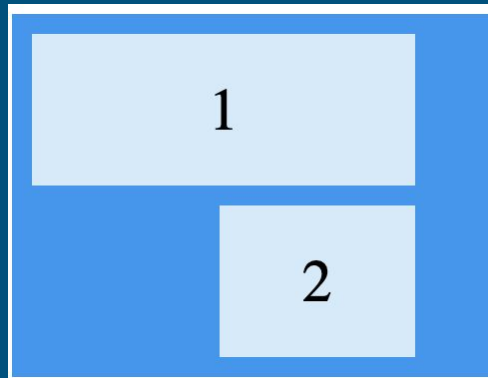


Grid-area

This is another way to use Grid-area

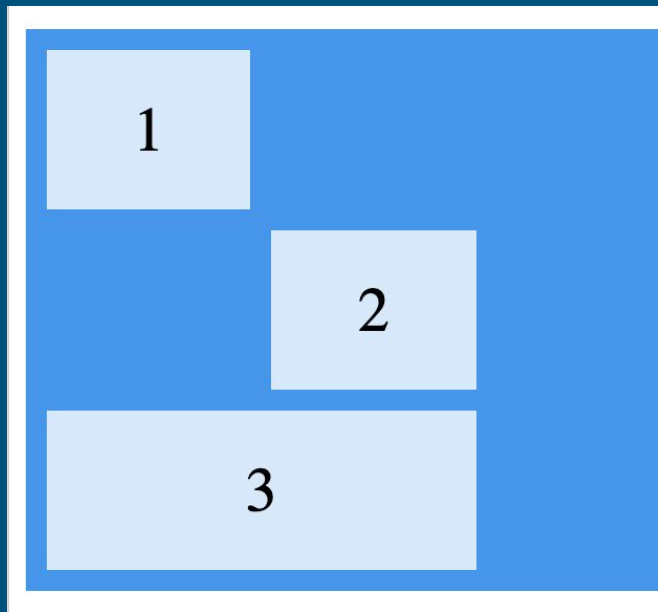
grid-area: grid-row-start / grid-column-start / grid-row-end / grid-column-end;

```
.item1 {  
  grid-area: 1 / 1 / 2 / 3;  
}  
  
.item2 {  
  grid-area: 2 / 2 / 2 / 3;  
}
```



Example:

```
.item1 {  
  grid-area: 1 / 1 / 1 / 1;  
}  
.item2 {  
  grid-area: 2 / 2 / 2 / 2;  
}  
.item3 {  
  grid-area: 3 / 1 / 3 / 3;  
}
```



ANY QUESTIONS?