GRID in CSS

In CSS, a grid is a layout system that allows you to create complex and flexible layouts for your web pages. It works by dividing a web page into a grid of rows and columns, and then placing content within those rows and columns.

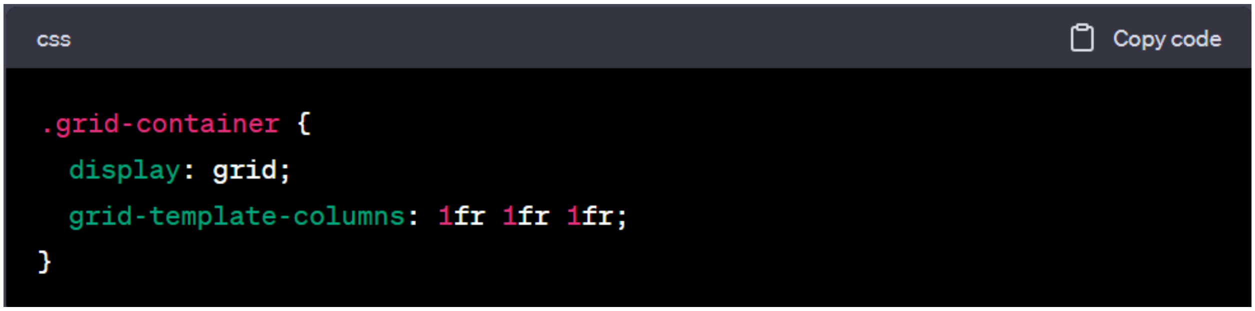
The CSS Grid layout provides two-dimensional layout control, where you can place items in rows and columns. It allows you to create a grid container, which is the parent element that holds the grid items (child elements).

To Create Grid Container use the Following Property

display : grid

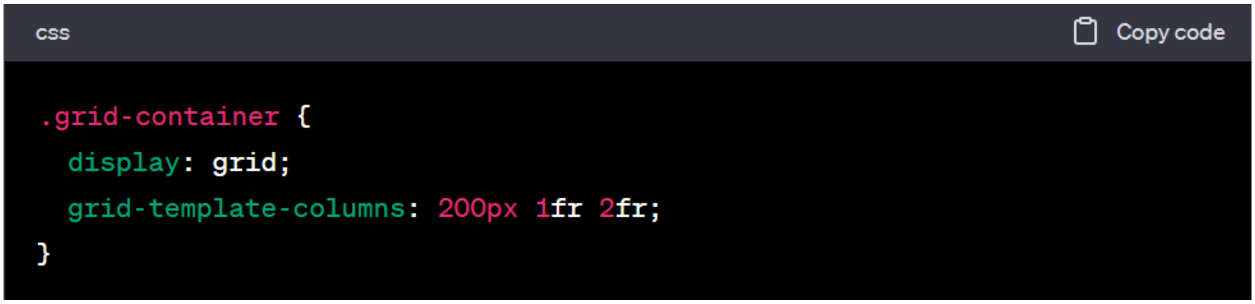
The **grid-template-columns** property in CSS Grid layout is used to define the columns of a grid. It specifies the size, shape, and number of columns in a grid container.

The grid-template-columns property can take one or more values, which specify the size of each column. You can use various units like pixels, percentages, em, rem, auto, and fractional units (like fr) to define the size of the columns. Here's an example that creates a grid with three columns of equal width:



```css
.grid-container {
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
}
```

In this example, the grid-template-columns property has three values, each with a value of 1fr. The fr unit is a fractional unit that divides the available space among the columns. In this case, the available space is divided into three equal parts.



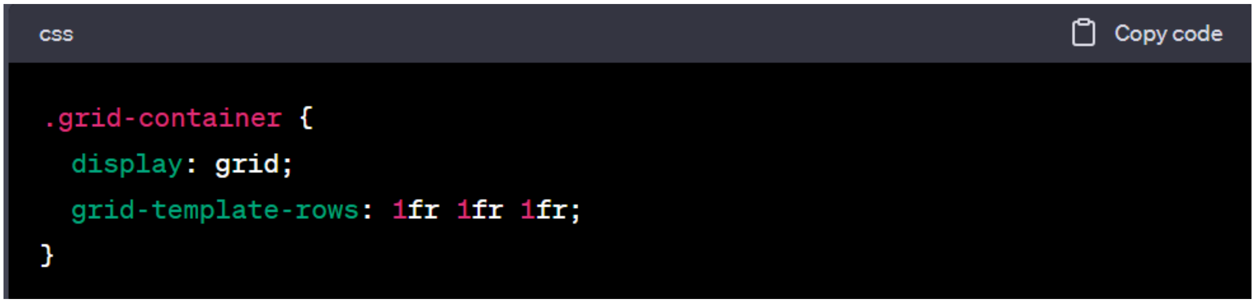
```css
.grid-container {
  display: grid;
  grid-template-columns: 200px 1fr 2fr;
}
```

In this example, the first column is set to a fixed width of 200px, while the second and third columns are set to take up one-third and two-thirds of the remaining space, respectively.

In addition to setting the size of columns, you can also use the grid-template-columns property to define the number and position of grid lines, which can be used to position grid items within the grid.

The **grid-template-rows** property in CSS Grid layout is used to define the rows of a grid. It specifies the size, shape, and number of rows in a grid container.

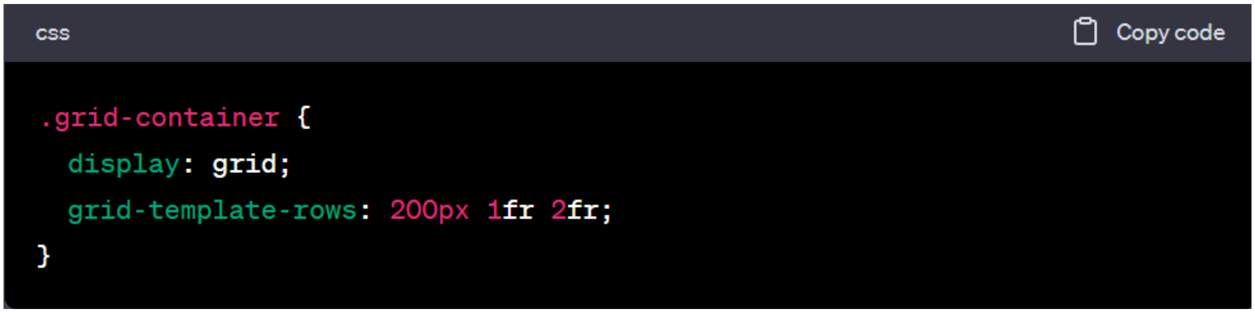
The **grid-template-rows** property can take one or more values, which specify the size of each row. You can use various units like pixels, percentages, em, rem, auto, and fractional units (like fr) to define the size of the rows. Here's an example that creates a grid with three rows of equal height:



```css
.grid-container {
  display: grid;
  grid-template-rows: 1fr 1fr 1fr;
}
```

In this example, the grid-template-rows property has three values, each with a value of 1fr. Like with grid-template-columns, the fr unit is a fractional unit that divides the available space among the rows. In this case, the available space is divided into three equal parts.

You can also use other units and values in combination with the fr unit to create more complex layouts. For example:



```css
.grid-container {
  display: grid;
  grid-template-rows: 200px 1fr 2fr;
}
```

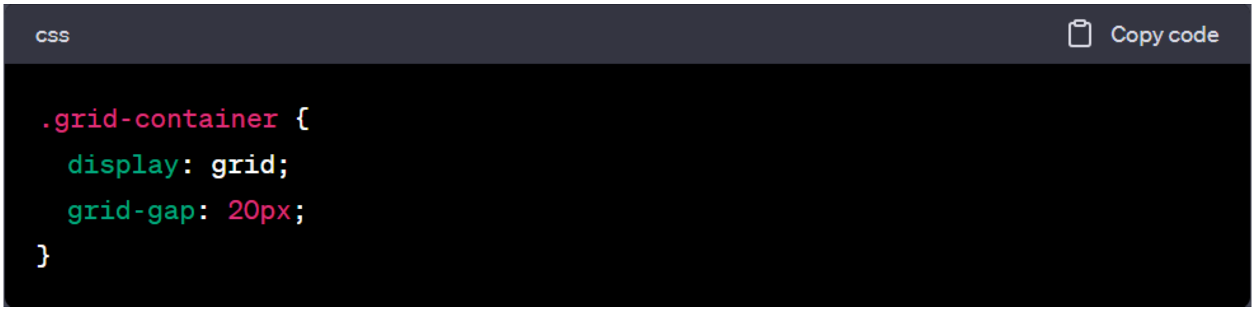
In this example, the first row is set to a fixed height of 200px, while the second and third rows are set to take up one-third and two-thirds of the remaining space, respectively.

In addition to setting the size of rows, you can also use the grid-template-rows property to define the number and position of grid lines, which can be used to position grid items within the grid.

Grid gap in CSS refers to the space between the rows and columns of a CSS grid layout. It is the amount of space that separates the grid tracks, which are the rows and columns that make up the grid.

To set the grid gap, you can use the grid-gap property in CSS. The grid-gap property accepts one or two values, which determine the size of the gap between the grid tracks.

For example, to set a gap of 20 pixels between the rows and columns of a grid, you can use the following CSS:



```css
.grid-container {
  display: grid;
  grid-gap: 20px;
}
```

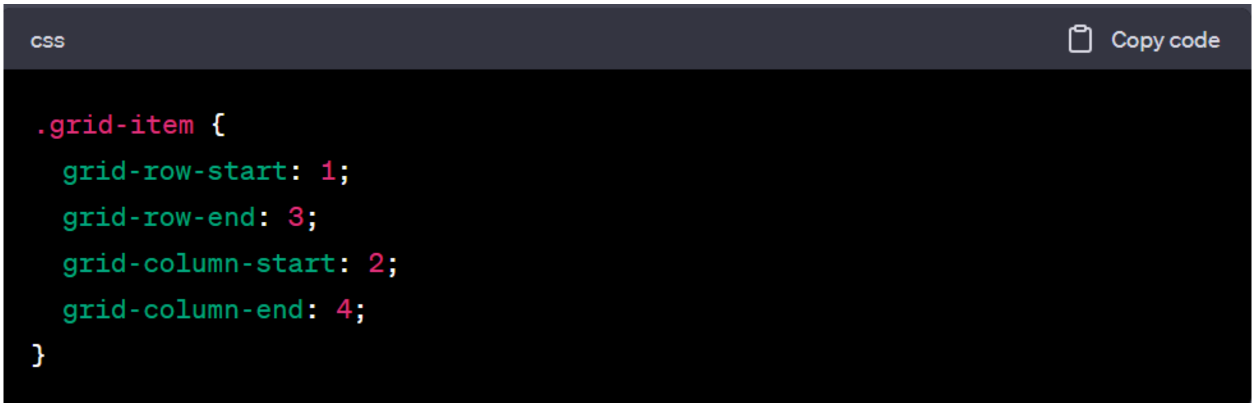
In this example, the grid-gap property sets a gap of 20 pixels between the rows and columns of the grid. You can also specify different values for the row and column gaps using the syntax grid-row-gap and grid-column-gap.

Note that the grid-gap property has been replaced by the gap property in newer versions of CSS. The gap property works the same way as grid-gap but can be used for other layout techniques such as flexbox.

In CSS grid, grid lines refer to the lines that separate the rows and columns of a grid. Each grid line can be referenced by its line number, starting from 1 for the first line.

You can use grid lines to position grid items within the grid container by specifying where the grid item should start and end. This can be done using the grid-row-start, grid-row-end, grid-column-start, and grid-column-end properties.

For example, to position a grid item to span across the second and third columns and the first two rows of a grid, you can use the following CSS:



```css
.grid-item {
  grid-row-start: 1;
  grid-row-end: 3;
  grid-column-start: 2;
  grid-column-end: 4;
}
```

The `grid-column` property specifies a grid item's size and location in a grid layout, and is a shorthand property for the following properties:

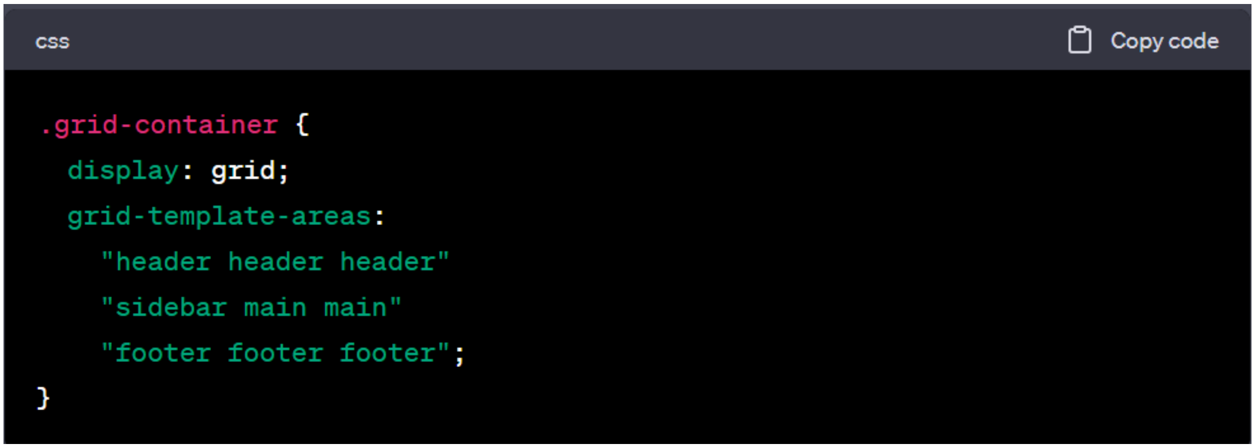
- grid-column-start
- grid-column-end

The `grid-row` property specifies a grid item's size and location in a grid layout, and is a shorthand property for the following properties:

- grid-row-start
- grid-row-end

grid-template-areas is a CSS property that defines a grid by giving names to areas of the grid instead of explicitly specifying the size and location of each grid item.

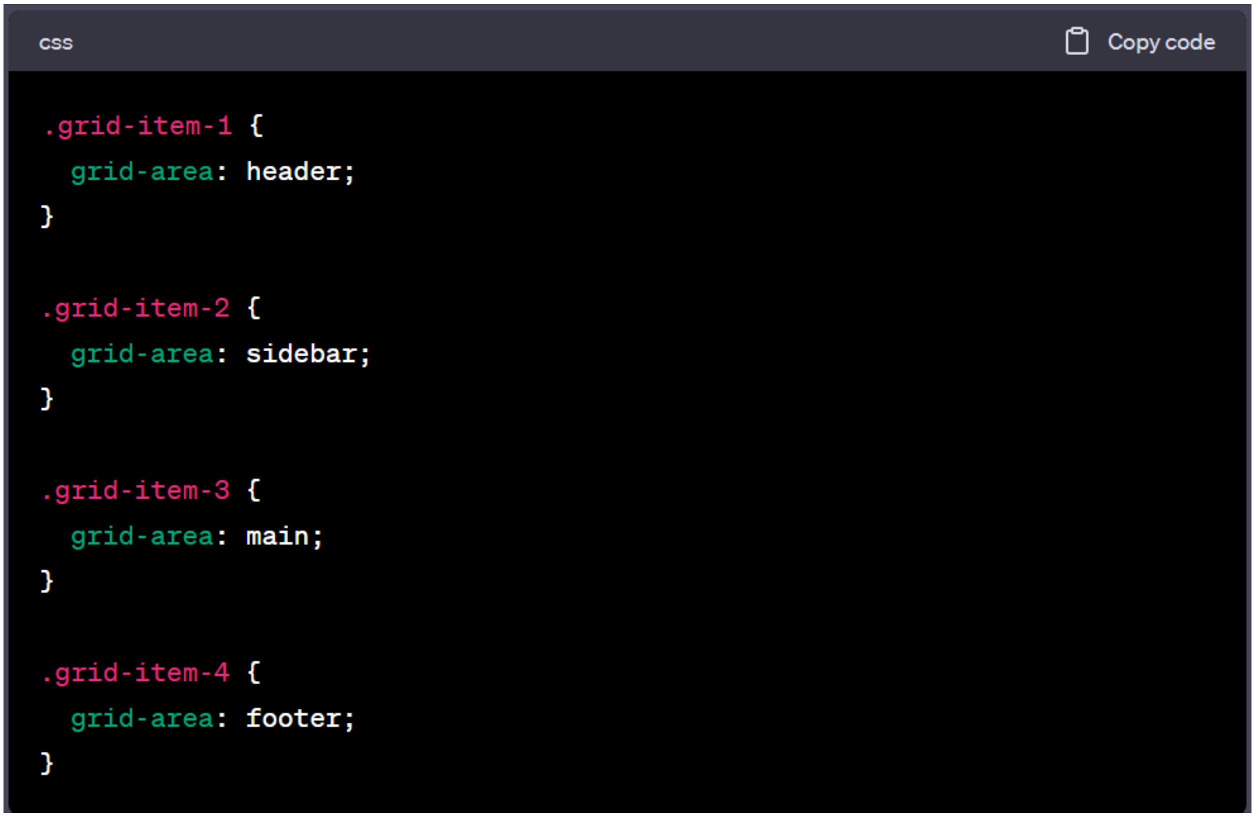
With grid-template-areas, you can create a visual map of your grid layout using a series of strings that represent the names of each grid area. Each string represents a row of the grid, with individual areas separated by whitespace. For example:



```css
.grid-container {
  display: grid;
  grid-template-areas:
    "header header header"
    "sidebar main main"
    "footer footer footer";
}
```

In this example, we have three rows and three columns. The first row has three areas named "header". The second row has one area named "sidebar" and two areas named "main". The third row has three areas named "footer".

To assign a grid item to a particular area, you can use the grid-area property and specify the name of the area. For example:



```css
.grid-item-1 {
  grid-area: header;
}

.grid-item-2 {
  grid-area: sidebar;
}

.grid-item-3 {
  grid-area: main;
}

.grid-item-4 {
  grid-area: footer;
}
```

In this example, we have four grid items that are assigned to their corresponding grid areas.

It is important to note that grid-template-areas can only be used with grid layouts that have a fixed number of rows and columns. If you need a more flexible grid layout, you should use grid-template-rows and grid-template-columns to define the size and location of each grid item.