

<https://webkit.org/blog/7434/css-grid-layout-a-new-layout-module-for-the-web/>

To create a CSS grid, you need to first set the display property of the parent element to grid. This will tell the browser that the element should be treated as a grid container. Once the element is a grid container, you can then use the grid-template-columns and grid-template-rows properties to define the number and size of the columns and rows in the grid.

Here is an example of how to create a CSS grid with two columns and three rows:

**.grid-container {**

**display: grid;**

**grid-template-columns: 1fr 1fr;**

**grid-template-rows: repeat(3, 1fr);**

**}**

The grid-template-columns property defines the number and size of the columns in the grid. In this example, there are two columns, each of which will be 1fr wide. The fr unit is a flexible length unit, which means that it will automatically resize to fit the content inside the column.

The grid-template-rows property defines the number and size of the rows in the grid. In this example, there are three rows, each of which will be 1fr tall.

Once you have defined the columns and rows in the grid, you can then place your child elements inside the grid by using the grid-column and grid-row properties. The grid-column and grid-row properties take a value that specifies the column or row that the element should be placed in.

Here is an example of how to place two child elements inside the grid, one in the first column and one in the third row:

**.grid-container > div {**

**grid-column: 1;**

**grid-row: 3;**

**}**

This will place the first child element in the first column and the second child element in the third row.

You can also use the grid-area property to place child elements inside the grid. The grid-area property takes a value that specifies the area of the grid that the element should be placed in.

Here is an example of how to place two child elements inside the grid, one in the top-left corner and one in the bottom-right corner:

**.grid-container > div {**

**grid-area: 1/1;**

**grid-area: 3/3;**

**}**

This will place the first child element in the top-left corner of the grid and the second child element in the bottom-right corner of the grid.

CSS grid is a powerful tool that can be used to create complex and responsive layouts. By understanding the basics of CSS grid, you can create beautiful and functional web pages.



**Grid Sizing**

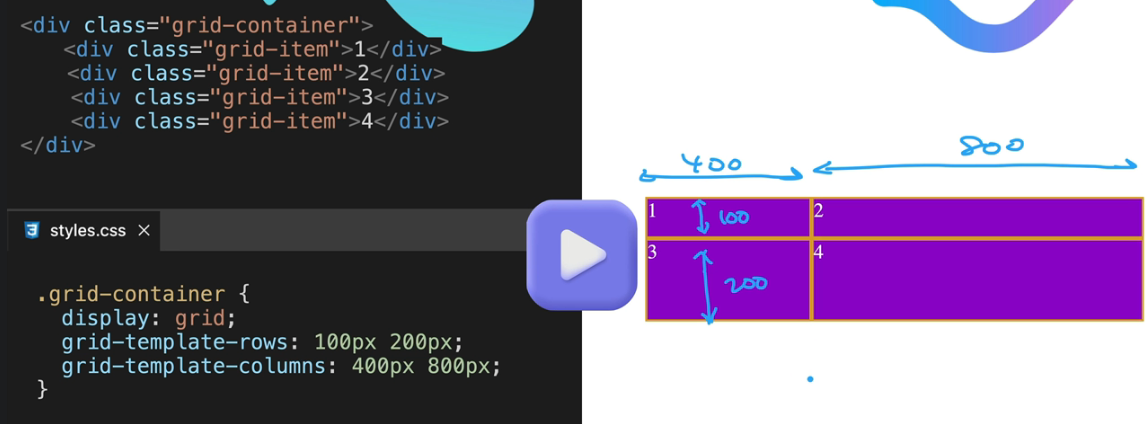
CSS Grid Sizing is a way to control the size of the columns and rows in a CSS grid layout. There are a few different ways to size grid tracks, and the best approach will depend on the specific needs of your layout.

One way to size grid tracks is to use fixed values. This means that the columns and rows will always be the same size, regardless of the content that is inside them. This can be useful for creating layouts that are very precise, but it can also be limiting if you need the layout to be flexible.

Another way to size grid tracks is to use flexible lengths. This means that the columns and rows will resize to fit the content inside them. This can be useful for creating layouts that are responsive and can adapt to different screen sizes.

Finally, you can also use a combination of fixed and flexible lengths to size grid tracks. This can be useful for creating layouts that have a certain amount of flexibility, but that also have some fixed elements.

**Fixed Grid Size**



/\* Fixed sizing \*/

.grid-container {

display: grid;

grid-template-columns: 100px 200px;

grid-template-rows: 50px 100px;

}

/\* Flexible sizing \*/

.grid-container {

display: grid;

grid-template-columns: repeat(2, 1fr);

grid-template-rows: repeat(2, 1fr);

}

/\* Combined sizing \*/

.grid-container {

display: grid;

grid-template-columns: 100px 1fr;

grid-template-rows: 50px 1fr;

}

In the first example, the columns and rows are sized to 100px and 50px respectively. This will create a grid with four columns and two rows.

In the second example, the columns and rows are sized to 1fr, which means that they will resize to fit the content inside them. This will create a grid with columns and rows that are the same width and height as the content inside them.

In the third example, the first column is sized to 100px and the second column is sized to 1fr. This will create a grid with three columns, the first of which is 100px wide and the other two of which are the same width as the content inside them.

The best way to choose between fixed and flexible sizing will depend on the specific needs of your layout. If you need a layout that is very precise, then fixed sizing may be the best option. If you need a layout that is responsive and can adapt to different screen sizes, then flexible sizing may be the best option. If you need a combination of both precision and flexibility, then you may want to consider using a combination of fixed and flexible lengths.

CSS Grid Placement

CSS Grid Placement is a way to control where grid items are placed in a grid layout. There are a few different ways to place grid items, and the best approach will depend on the specific needs of your layout.

One way to place grid items is to use line-based placement. This means that you can specify the row and column that the grid item should be placed on.

Another way to place grid items is to use area-based placement. This means that you can specify the area of the grid that the grid item should be placed in.

Finally, you can also use a combination of line-based and area-based placement to place grid items.

Here are some examples of how to use CSS Grid Placement:

/\* Line-based placement \*/

.grid-item {

grid-column: 1;

grid-row: 1;

}

/\* Area-based placement \*/

.grid-item {

grid-area: 1/1;

}

/\* Combined placement \*/

.grid-item {

grid-column: 1;

grid-row: 1;

grid-area: 1/1;

}

In the first example, the grid item will be placed on the first row and the first column of the grid.

In the second example, the grid item will be placed in the area of the grid that is defined by the first row and the first column.

In the third example, the grid item will be placed on the first row and the first column of the grid, and it will also be placed in the area of the grid that is defined by the first row and the first column.

The best way to choose between line-based, area-based, and combined placement will depend on the specific needs of your layout. If you need to place a grid item in a specific location, then line-based placement may be the best option. If you need to place a grid item in a specific area of the grid, then area-based placement may be the best option. If you need to place a grid item in a specific location and in a specific area of the grid, then combined placement may be the best option.

In addition to line-based, area-based, and combined placement, you can also use the grid-template-areas property to place grid items in a grid layout. The grid-template-areas property takes a value that specifies the areas of the grid that should be filled with grid items.

Here is an example of how to use the grid-template-areas property:

.grid-container {

display: grid;

grid-template-areas:

"header header"

"content content"

"footer footer";

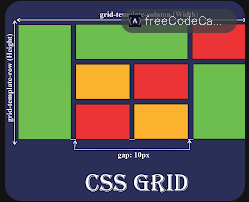
}

In this example, the grid will have three areas: header, content, and footer. The header area will be filled with the first child element of the grid container, the content area will be filled with the second child element of the grid container, and the footer area will be filled with the third child element of the grid container.

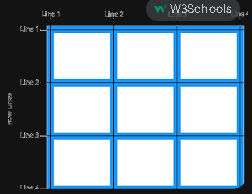
The grid-template-areas property can be used to create complex and responsive layouts. By understanding the basics of CSS Grid Placement, you can create beautiful and functional web pages.

**Terms in CSS Grid layout**

Grid container: A grid container is an element that has the display: grid property set on it. The grid container is the parent of all the grid items in the grid layout.



Grid item: A grid item is an element that is a child of a grid container. Grid items are placed in the grid layout according to the rules of the grid container.



Grid track: A grid track is a column or row in the grid layout. Grid tracks are created by the grid-template-columns and grid-template-rows properties.