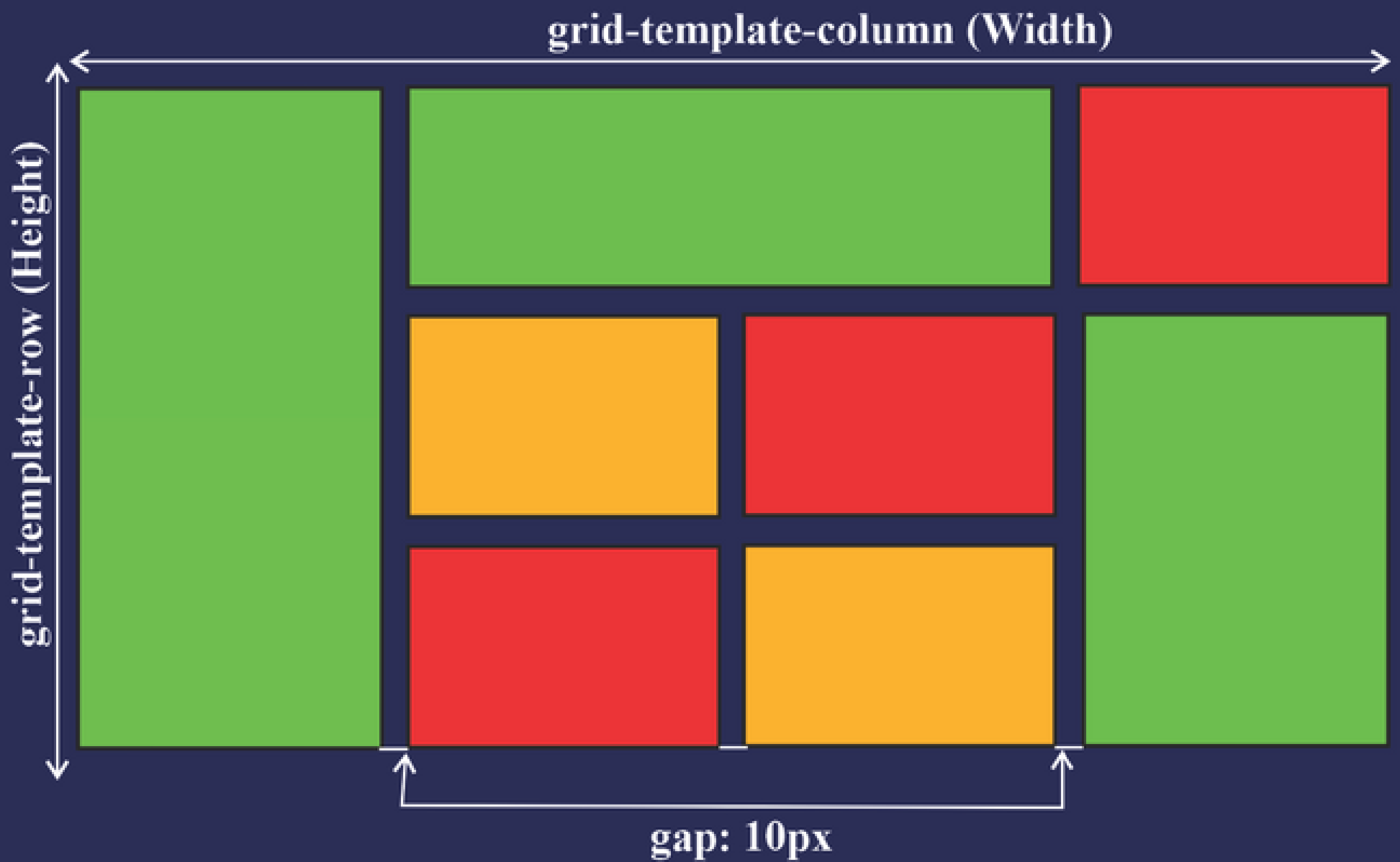




How to Use CSS Grid Layout – Grid Properties Explained with Examples

1/17



CSS GRID


Have you ever had a problem positioning items on your web browser? Perhaps every time you try to think of a solution, you become tired and give up. If so, stay tuned as I reveal a new method for resolving these kinds of problems with minimal or no stress.

What is CSS Grid?

Here is a concise definition of the CSS Grid layout: CSS Grid is a powerful tool that allows for two-dimensional layouts for columns and rows to be created on the web.



When Should You Use CSS Grid

Although you can use CSS Grid in practically any aspect of web development, there are certain situations when it's ideal.



For example, when we have a complex design layout to implement, CSS Grid is better than the CSS float property. This is because Grid is a two-dimensional layout (with columns and rows), whereas the CSS float property is a one-dimensional layout (with columns or rows).

Grid is also a good choice when we need a space or gap between elements. By using the CSS grid gap property, the spacing of two elements is much easier than using the CSS margin and padding properties which might end up complicating things.



CSS Grid Properties

The CSS grid layout consists of many grid properties. Now we'll take a look at some of them so we can learn how to use them.

Grid container property

This is a CSS grid property that houses the grid items/elements. We implement the CSS grid container property by setting the container to a display property of grid or in-line grid.

For Example:

`display: grid;` or `display: in-line grid;`

Grid-template-column property

This is a property used to set each column's width. It can also define how many columns you want to set to your project.

You can implement the CSS grid column property using grid-template-column.

For Example:

```
grid-template-column: 100px auto 100px;
```

Grid-template-row property

You use the CSS row property to set the height of each column. You can also use it to define how many rows you want to set in your project.

You can implement the CSS grid row property using grid-template-row, like this:

```
grid-template-row: 50px 50px;
```

Column-gap property

As the name states, it is a grid property that assigns a space between two or more columns in a container. You can do this by using the column-gap property and giving it a value. For example:

`column-gap: 20px;`

From the code above, you can see that a gap of 20px was assigned to the column.

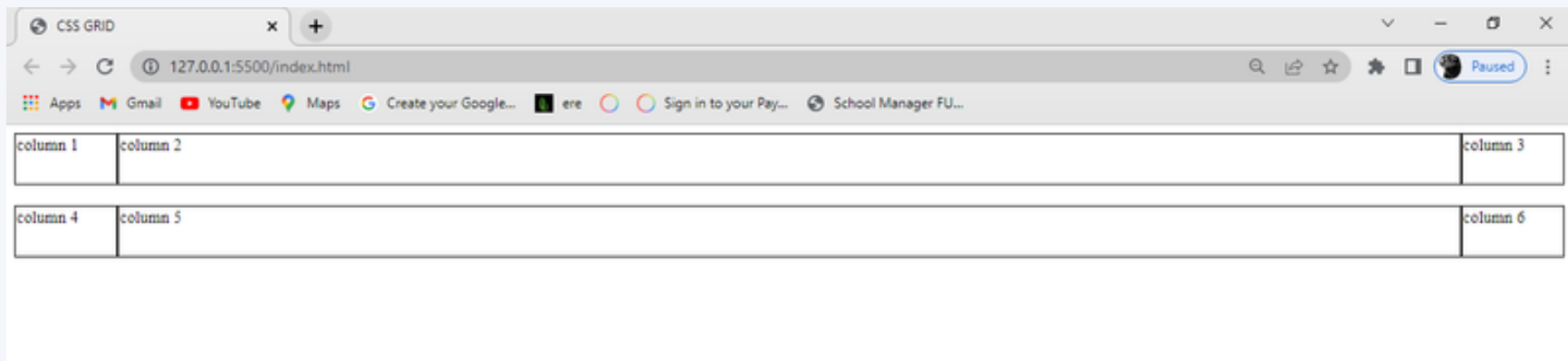
column 1	column 2	column 3
column 4	column 5	column 6

20px column-gap

Row-gap property

Just like column-gap, row-gap is a CSS property that assigns a space between two or more rows in a container. For example:

row-gap: 50px;



row-gap: 50px;

Note that we can also assign a gap to both the columns and rows of a container by using the gap property. For this to work, we only assign one value to both the columns and the rows of the container, just like we did in the code above.

Here's an example:

gap: 20px;

Justify-content property

This is a grid property that you use in positioning items (columns and rows) horizontally in a container. It displays how the web browser positions the spaces around the items (columns and rows).

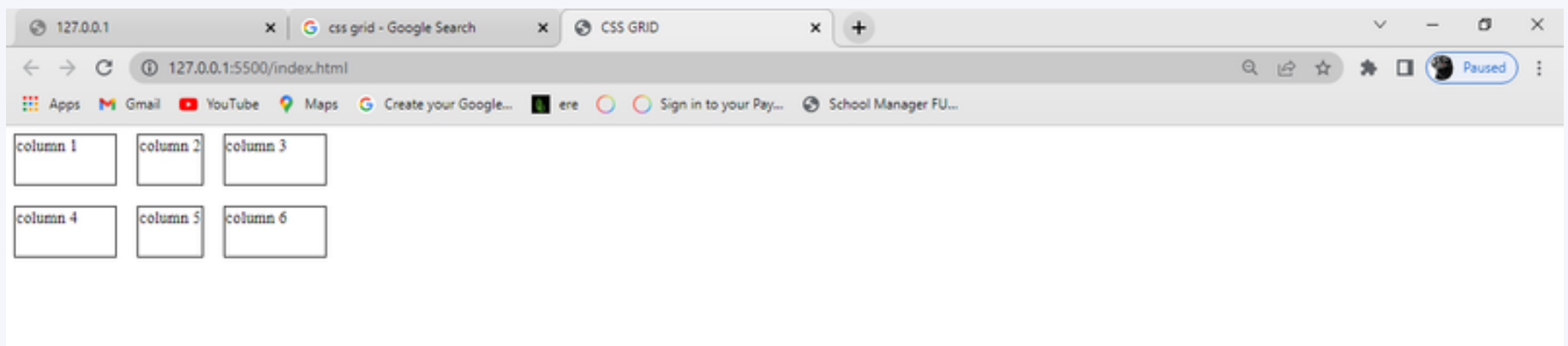
The justify-content property has six possible values:

- **Start**
- **end**
- **center**
- **space-around**
- **space-between**
- **space-evenly**

1) Start

This positions the items at the left side of the browser and can be executed with the following code:

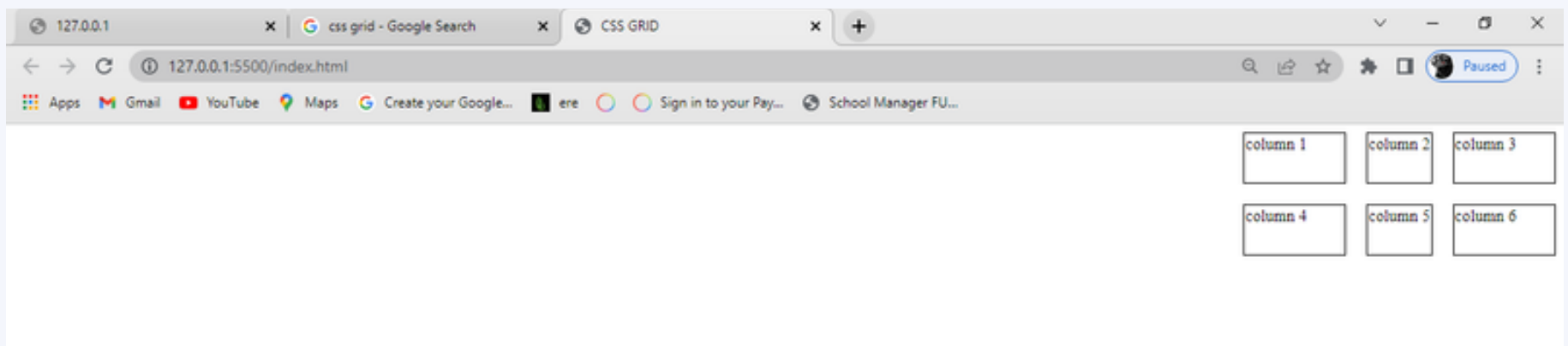
justify-content: start;



2) End

This positions the items at the right side of the browser and can be executed with the following code:

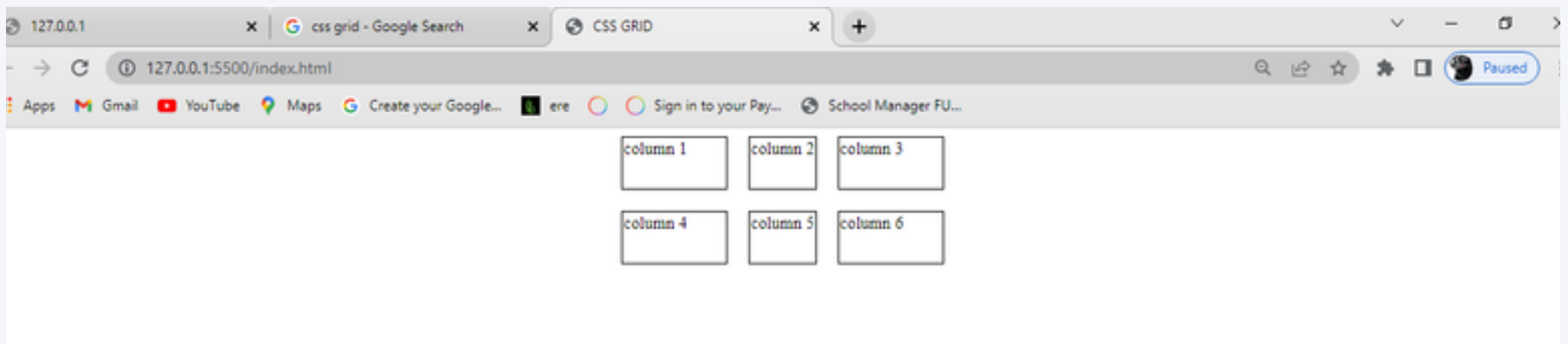
justify-content: end;



3) Center

This positions the items at the center of the browser and can be executed with the following code:

justify-content: center;

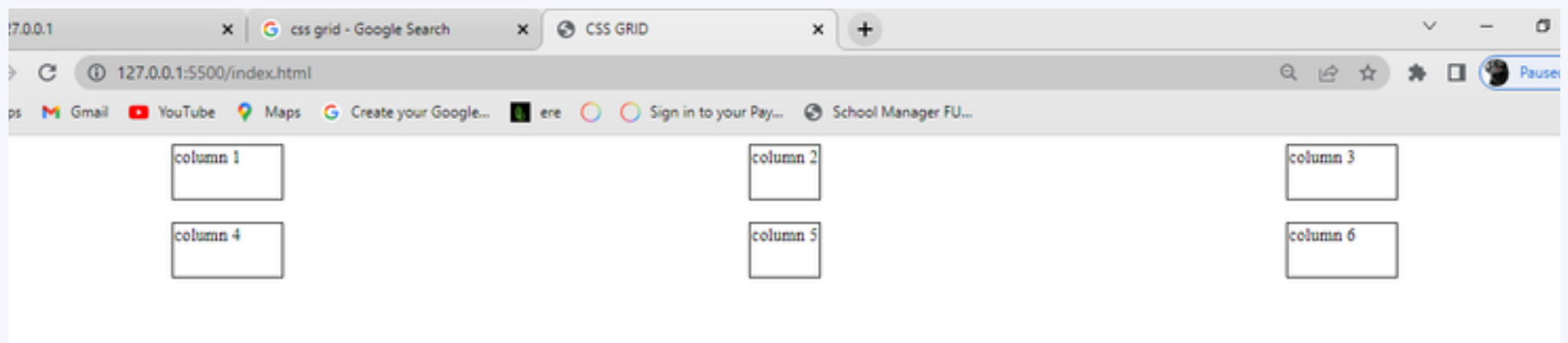


4) Space-around

This property distributes the items in the container evenly, where each item in the container has an equal amount of space from the next container.

This code can be executed like this:

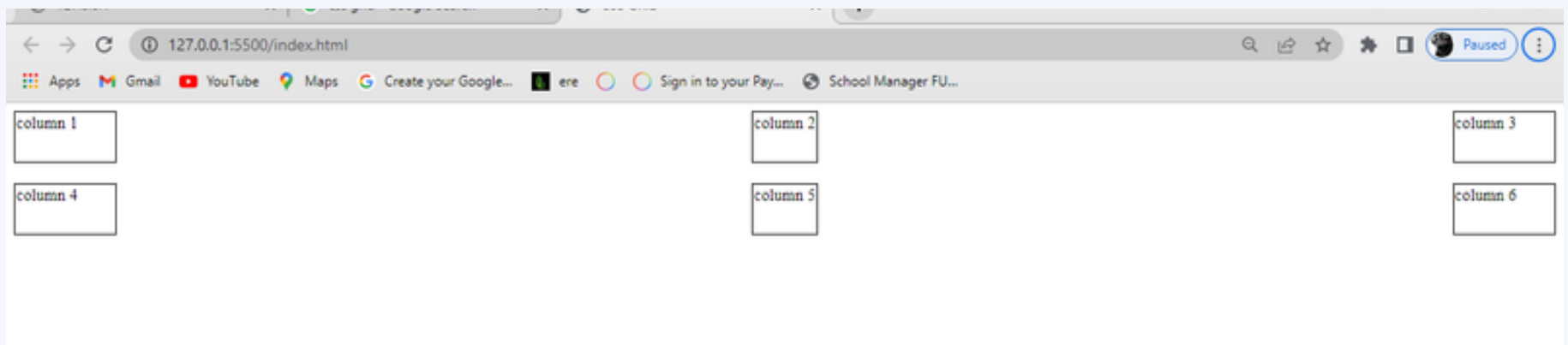
`justify-content: space-around;`



5) Space-between

Just like the space-around property, space-between distributes the items in the container evenly, where each item in the container has an equal amount of space from the next one in the container. It takes up the entire width of the container. This code can be executed like this:

justify-content: space-between;

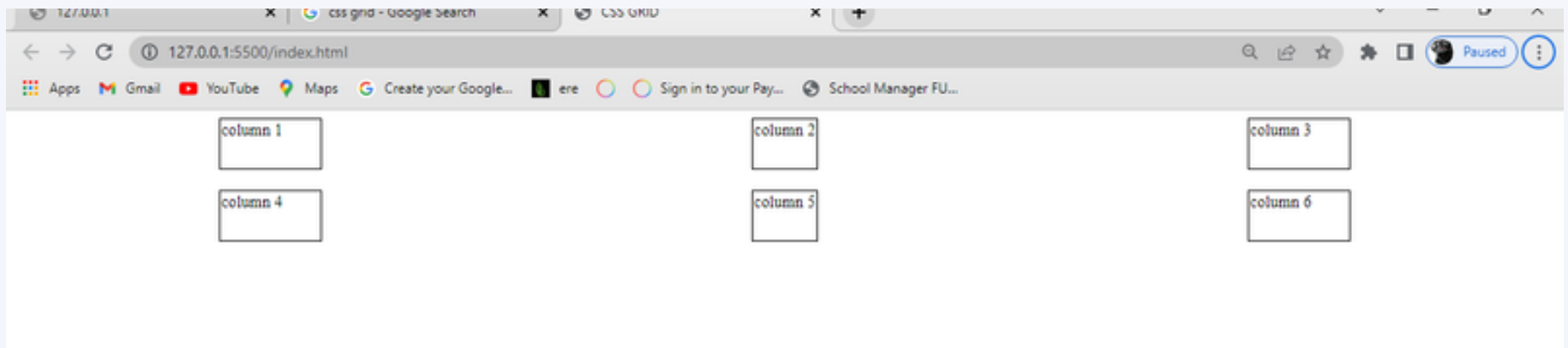


6) Space-evenly

Just as the name states, this property distributes the items in the container evenly, where each item in the container has an equal amount of space from the next one in the container.

This code can be executed like this:

`justify-content: space-evenly;`



Align-content property

Align-content is the opposite of justify-content. You use the align-content property in positioning items vertically in a container. Just like justify-content, the align-content property has six possible values:

- **Start**
- **end**
- **center**
- **space-around**
- **space-between**
- **space-evenly**



Love This Post ?

Don't forget to share and save this tips for later