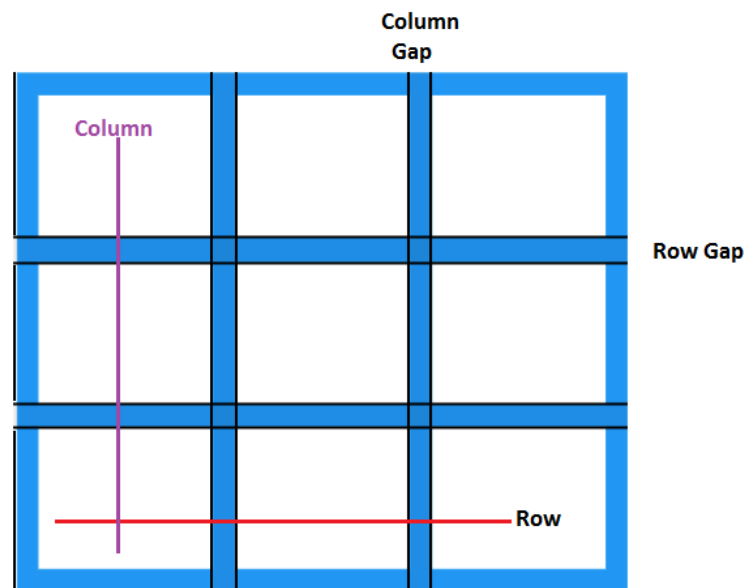


## **W3.CSS Grid Layout**

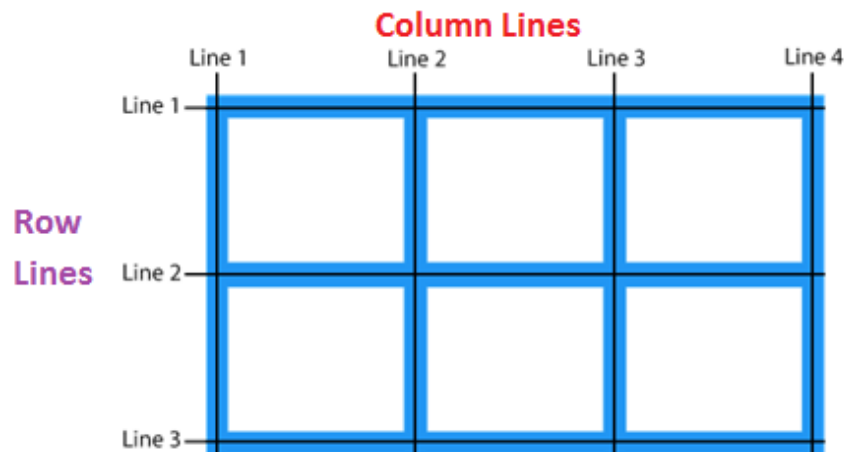
W3.CSS provides Grid Layout based system, just like matrix, of rows and columns. Using the grid, web pages may be designed more effectively and easily.

Lets have a look on some grid terminologies:

- Grid **Columns** are the vertical lines of grid items.
- Grid **Rows** are the horizontal lines of grid items.
- Grid **Gaps** is the space between each column/row. It can be adjusted by using **grid-column-gap**, **grid-row-gap**, **grid-gap** property.



- Grid **Column Lines** are the lines between columns are called *column lines*.
- Grid **Row Lines** are the lines between rows.



## Grid Container:

- Whenever display property of a HTML Element is set to **grid** or **inline-grid**, it becomes a grid container. Grid containers consist of grid items, placed inside columns and rows. For example:

```
.grid-container {  
    display: grid;    --- display property  
    .....  
}
```

- All direct children of the grid container automatically become **grid items**.
- **Grid Elements (items)** : A Grid Layout must have a parent element with, with one or more child elements in it.

## Properties of Grid Container

### 1. grid-template-columns property

It defines the number of columns in grid layout with the width of each Column separated by space. We specify width as “auto” for each column to make all the columns of the same width, or we may specify width of each column. For example:

**grid-template-columns: auto auto auto auto;**  
will have 4 columns with equal width.

**grid-template-columns: 80px 200px auto 40px;**

will have 4 columns with specified width of 3 columns and 4<sup>th</sup> one will be automatically adjusted.

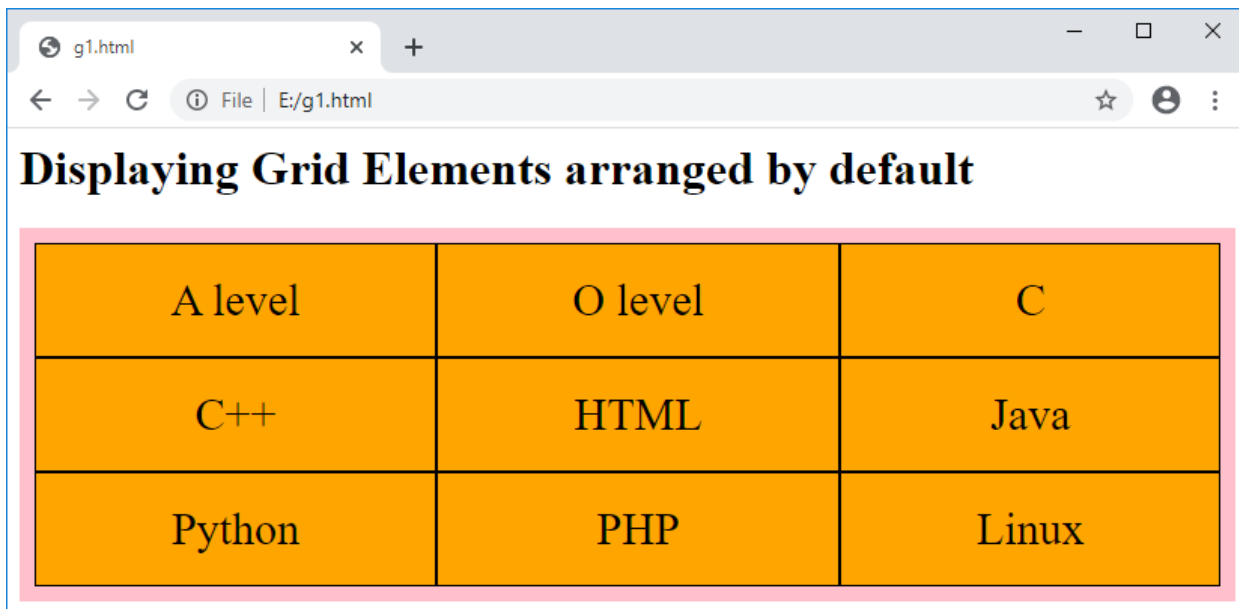
**Example:** Displaying Grid Elements arranged by default column width

```
<html>  
<head>  
<style>  
    .grid-container {  
        display: grid;  
        grid-template-columns: auto auto auto;  
        background-color: pink;  
        padding: 10px;  
    }  
    .grid-item {  
        background-color: orange;  
        border: 1px solid black;
```

```

padding: 20px;
font-size: 30px;
text-align: center;
}
</style>
</head>
<body>
<h1>Displaying Grid Elements arranged by default</h1>
<div class="grid-container">
  <div class="grid-item">A level</div>
  <div class="grid-item">O level</div>
  <div class="grid-item">C</div>
  <div class="grid-item">C++</div>
  <div class="grid-item">HTML</div>
  <div class="grid-item">Java</div>
  <div class="grid-item">Python</div>
  <div class="grid-item">PHP</div>
  <div class="grid-item">Linux</div>
</div>
</body>
</html>

```



## 2. grid-template-rows property

It defines the number of rows in grid layout with the height of each row separated by space. For example:

**grid-template-rows: 80px 200px 160px;**  
will have 3 rows of specified height.

## 3. justify-content Property

This property is used to align the whole grid inside the container as per the value of the property.

**justify-content: values;**

where values can be

**space-evenly**- will give the columns equal amount of space between, and around them  
**space-around** - will give the columns equal amount of space around them  
**space-between** - will give the columns equal amount of space between them  
**center**- will align the grid in the middle of the container  
**start**- will align the grid at the beginning of the container  
**end** - will align the grid at the end of the container

#### 4. align-content Property

This property is used to *vertically* align the whole grid inside the container.

**align-content:** values;

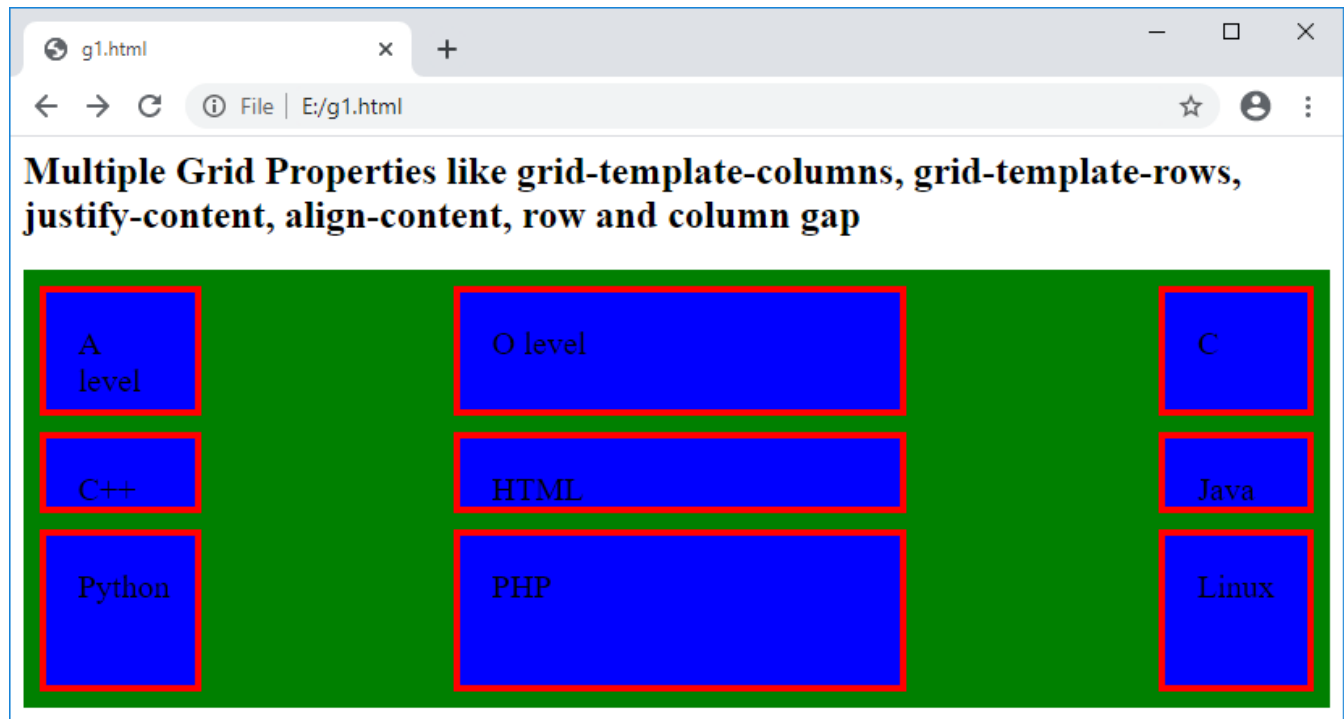
where values can be

**space-evenly** - will give the rows equal amount of space between, and around them  
**centre** - will align the rows in the middle of the container  
**space-around** - will give the rows equal amount of space around them  
**space-between** - will give the rows equal amount of space between them  
**start**- will align the row at the beginning of the container  
**end** - will align the row at the end of the container

#### Example:

```
<html>
<head>
<style>
.grid-container {
  display: grid;
  grid-template-columns: 100px 280px auto;
  grid-template-rows: 80px 50px 100px;
  justify-content: space-between ;
  align-content: start;
  grid-column-gap: 5px;
  grid-row-gap: 10px;
  background-color: green;
  padding: 10px;
}
.grid-item {
  background-color: blue;
  border: 4px solid red;
  padding: 20px;
  font-size: 20px ;
}
</style>
</head>
<body>
<h2>Multiple Grid Properties like grid-template-columns, grid-template-rows, justify-content, align-
content, row and column gap </h2>
<div class="grid-container">
  <div class="grid-item">A level</div>
```

```
<div class="grid-item">O level</div>
<div class="grid-item">C</div>
<div class="grid-item">C++</div>
<div class="grid-item">HTML</div>
<div class="grid-item">Java</div>
<div class="grid-item">Python</div>
<div class="grid-item">PHP</div>
<div class="grid-item">Linux</div>
</div>
</body>
</html>
```



## Assignment

1. What is grid layout? What is use of grid layout?
2. What is grid container? What are the properties of grid container?