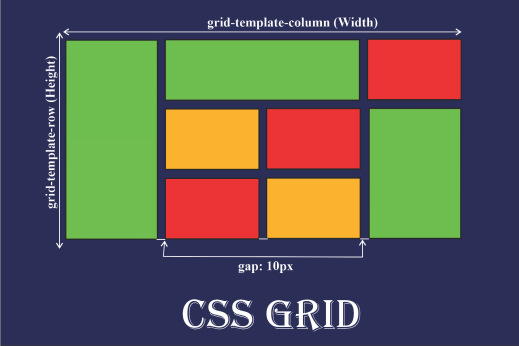
CSS GRID **THEORY**

1

**Video to watch before sessionhttps://www.youtube.com/watch?v=9zBsdzdE4sM**

**OBJECTIVES FOR TODAY **

✔ Grid **Terminology**

✔ Create the **Grid Temple** 

✔ Naming **Grid Areas**

✔ Properties of grid container:

▪ **grid-template-columns**

▪ **grid-template-areas**

▪ **grid-area**

▪ **gap**

EXPLAIN

05 MIN

How **CSS Grids Work**

✓Set an elements display gird to create **grid container**. Its children would become **grid items**.

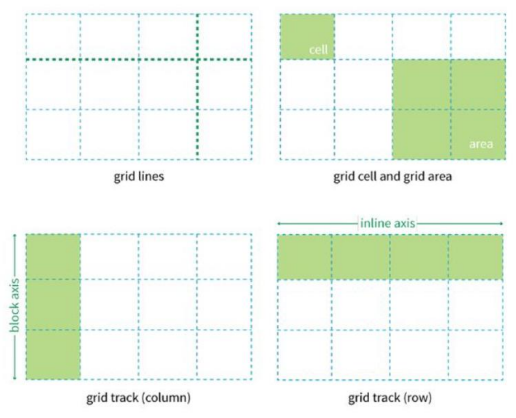
✓Set up the **columns** and **rows** for the grid.

✓Assign each **grid item** to an area on the **grid**.

EXPLAIN

05 MIN

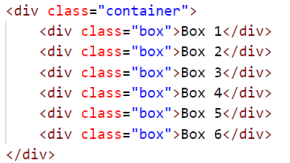
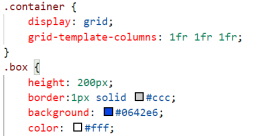
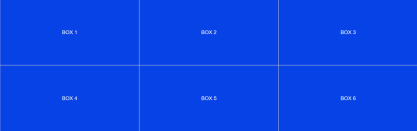
**Grid Terminology**

****

EXPLAIN

05 MIN

**Creating** a **grid container**

To make an element a **grid container**, set its **display** property to **grid** All of its children automatically become grid items

EXPLAIN

05 MIN

**Grid Gap**

To make the space in grid layout we can use **gap** or **margin**.

.grid-container {

10 px

gap: 10px;

}

*Row = 10px , column = 10px*

10 px

The **value** of the **gap** here is used in **both row**s and **columns**. It means we set the **row-gap = 10px** and the **column- gap = 10px.**

EXPLAIN

05 MIN

**Grid Gap**

To make the space in grid layout we can use **gap** or **margin**. 10 px

.grid-container {

gap:0 10px;

}

*Row = 0px , column = 10px*

EXPLAIN

05 MIN

**Grid Gap**

What is the **value** of **gap** here?

10 px

.grid-container {

gap:?? ??;

}

*Row = ?? , column = ??*

ACTIVITY

10 MIN

**Activity 1**

Define the layout like this picture with HTML that teacher provide

Using **gap** properties

✓ Make space only row

✓ Make space only column

✓ Make the space row & column



EXPLAIN



05 MIN

.grid-container{

display: grid;

**Grid Template Column**

grid-template-columns: 50% 50%;

}

.grid-container{

display: grid;

grid-template-columns: auto auto;

}

.grid-container{

display: grid;

grid-template-columns: 1fr 1fr;

} CSS Grid Layout introduces a new **unit** of **length** called fr, which is short for “**fraction**”. We can use any unit in CSS

like **px, em,** and **rem** but we suggest: **fr** is easy!

EXPLAIN



05 MIN

.grid-container{ display: grid;

**Grid Template Column**

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

.grid-container{

display: grid;

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

OR

.grid-container{

display: grid;

grid-template-columns: repeat(4, 1fr); }

EXPLAIN



05 MIN

.grid-container{ display: grid;

**Grid Template Column**

grid-template-columns: 1fr 1fr 2fr; }

ACTIVITY

20 MIN

EX 1

EX 2

**Activity 2**

Define the layout like this picture with HTML that teacher provide Using ***grid-template-columns***

1fr 1fr

2fr 1fr 2fr

EXPLAIN

05 MIN

**Grid Template Area**

header

Content 1

sidebar

Content 2

footer

**Grid area** is the easy ways in **CSS Grid layout** to define the **column** of **layout faster**.

EXPLAIN



05 MIN

<HTML>

<div class="grid-container">

**Example**

<header class="header">Header</header> <aside class="sidebar">SideBar</aside> <div class="content1">Content 1</div> <div class="content2">Content 2</div> <footer class="footer">Footer</footer> </div>

<CSS>

.grid-container{

display: grid;

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

'header header'

'sidebar content1'

'sidebar content2'

'footer footer';

}

.header{ grid-area: header; }

.sidebar{

grid-area: sidebar; }

.content1{

grid-area: content1; }

ACTIVITY

20 MIN

**Activity 3**

Define the layout like this picture with HTML that teacher provide Using ***grid-template-areas***

******

HOMEWORK



https://cssgridgarden.com/