

CSS Grid

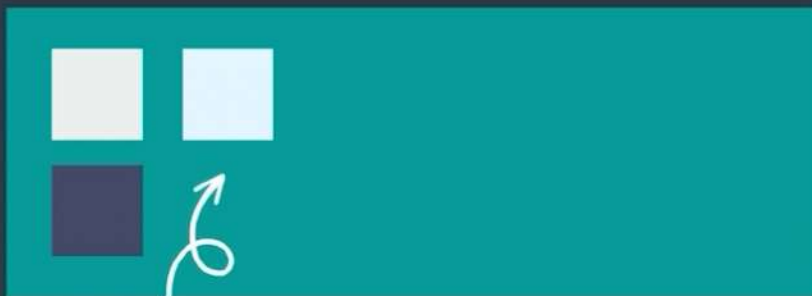
Setting a container's display to grid will make all children grid items

```
container {  
  display : grid;
```

```
}
```

hagupta7385@gmail.com

grid container



grid item

Grid Template

They define the lines & track sizing

grid-template-rows

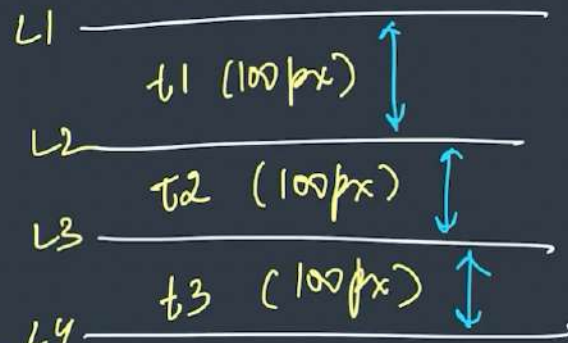
grid-template-columns

Grid Template

They define the lines & track sizing

grid-template-rows : 100px 100px 100px;

grid-template-columns



Grid Template

Repeat is used to divide all available space

grid-template-rows : repeat(count, 1fr)

grid-template-columns : repeat(count, 1fr)

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

grid-template-rows : 1fr 1fr

Grid Template

Repeat is used to divide all available space

`grid-template-rows : repeat(count, 1fr)`

1 fraction of 100% $\frac{\text{width}}{\text{available space}}$



Grid Gaps

They define the gaps between the lines

row-gap : 10px;

column-gap : 10px;

grid-gap : rowGap columnGap

Grid Columns

Defines an item's starting & ending position inside the column

grid-column-start : line_number

grid-column-end : line_number

grid-column : start_col / end_col

grid-column : start_col / span number

Grid Columns

Defines an item's starting & ending position inside the column

grid-column-start : line_number 1

grid-column-end : line_number 3

1 → 4

grid-column : start_col / end_col

grid-column : start_col / span number

1 / span 3

2/4



Grid Rows

Defines an item's starting & ending position inside the row

grid-row-start : line_number

grid-row-end : line_number

grid-row : start_row / end_row

grid-row : start_row / span number

Common Properties

- justify-items (container)
- justify-self (item)

} Horizontal

- align-items (container)
- align-self (item)

} Vertical

- place-items
- place-self

CSS Animations

To animate CSS elements

```
@keyframe myName {
```

```
  from { font-size : 20px; }
```

```
  to { font-size : 40px; }
```

```
}
```

initial
final

← template

CSS Animations

- animation-name

- ✓ animation-duration

- ✓ animation-timing-function

- ✓ animation-delay

- animation-iteration-count

- animation-direction

Animation Shorthand

animation : myName 2s linear 3s infinite normal

Media Queries

width (of viewport)

```
@media (min-width : 400px) {  
    div {  
        background-color : red;  
    }  
}
```

```
@media (max-width : 400px) {  
    div {  
        background-color : red;  
    }  
}
```

Media Queries

orientation (of viewport)

```
@media (orientation : landscape) {  
    div {  
        background-color : red;  
    }  
}
```


z-index

It decides the **stack level** of elements

Overlapping elements with a larger z-index cover those with a smaller one.

z-index : auto (0)

z-index : 1 / 2 / ...

z-index : -1 / -2 / ...

% in Animation

```
@keyframe myName {
```

```
  0% { font-size : 20px; }
```

```
  50% { font-size : 30px; }
```

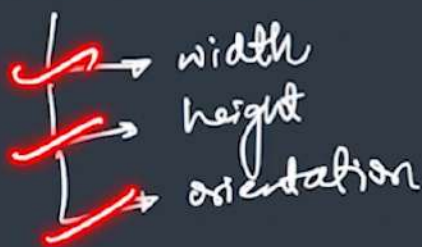
```
  100% { font-size : 40px; }
```

```
}
```

Media Queries

media features - width (of viewport)

device / environment



```
@media (width : 400px) {  
  div {  
    background-color : red;  
  }  
}
```

Media Queries

width (of viewport)

```
@media (min-width : 400px) {  
  div {  
    background-color : red;  
  }  
}
```

```
@media (max-width : 400px) {  
  div {  
    background-color : red;  
  }  
}
```

Media Queries

orientation (of viewport)

```
@media (orientation : landscape) {
```

```
  R (liv)
```

```
    background-color : red;
```

```
  }
```

```
}
```

z-index

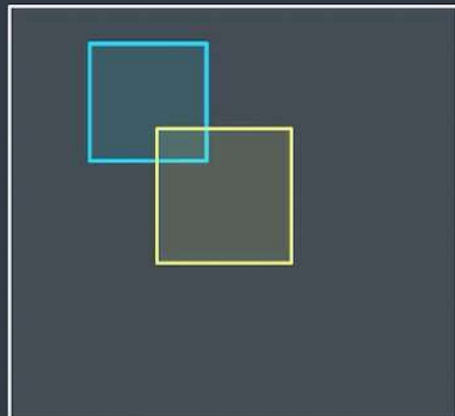
It decides the **stack level** of elements

Overlapping elements with a larger z-index cover those with a smaller one.

z-index : auto (0)

z-index : 1 / 2 / ...

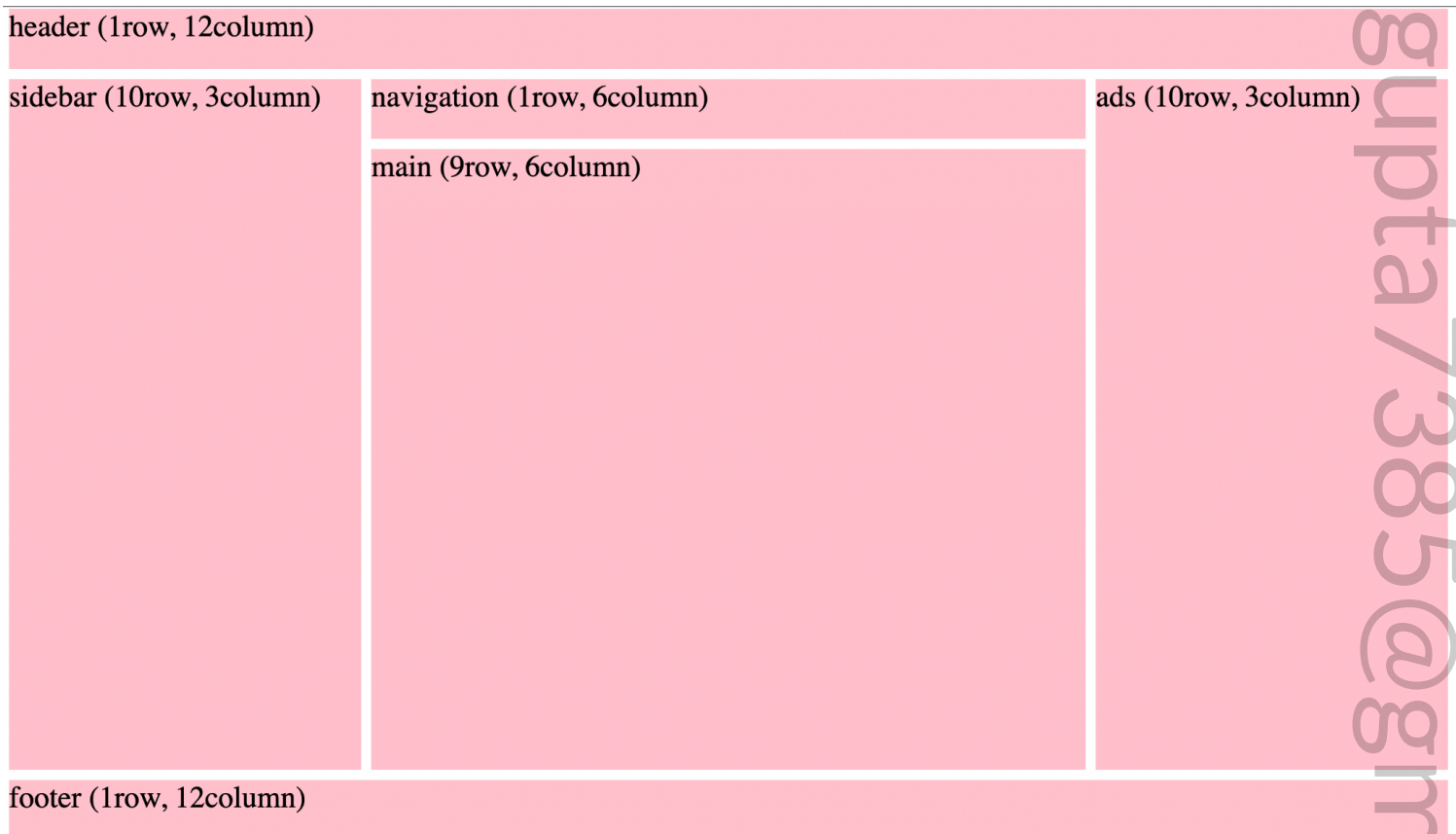
z-index : -1 / -2 / ...



CSS (Part 6)

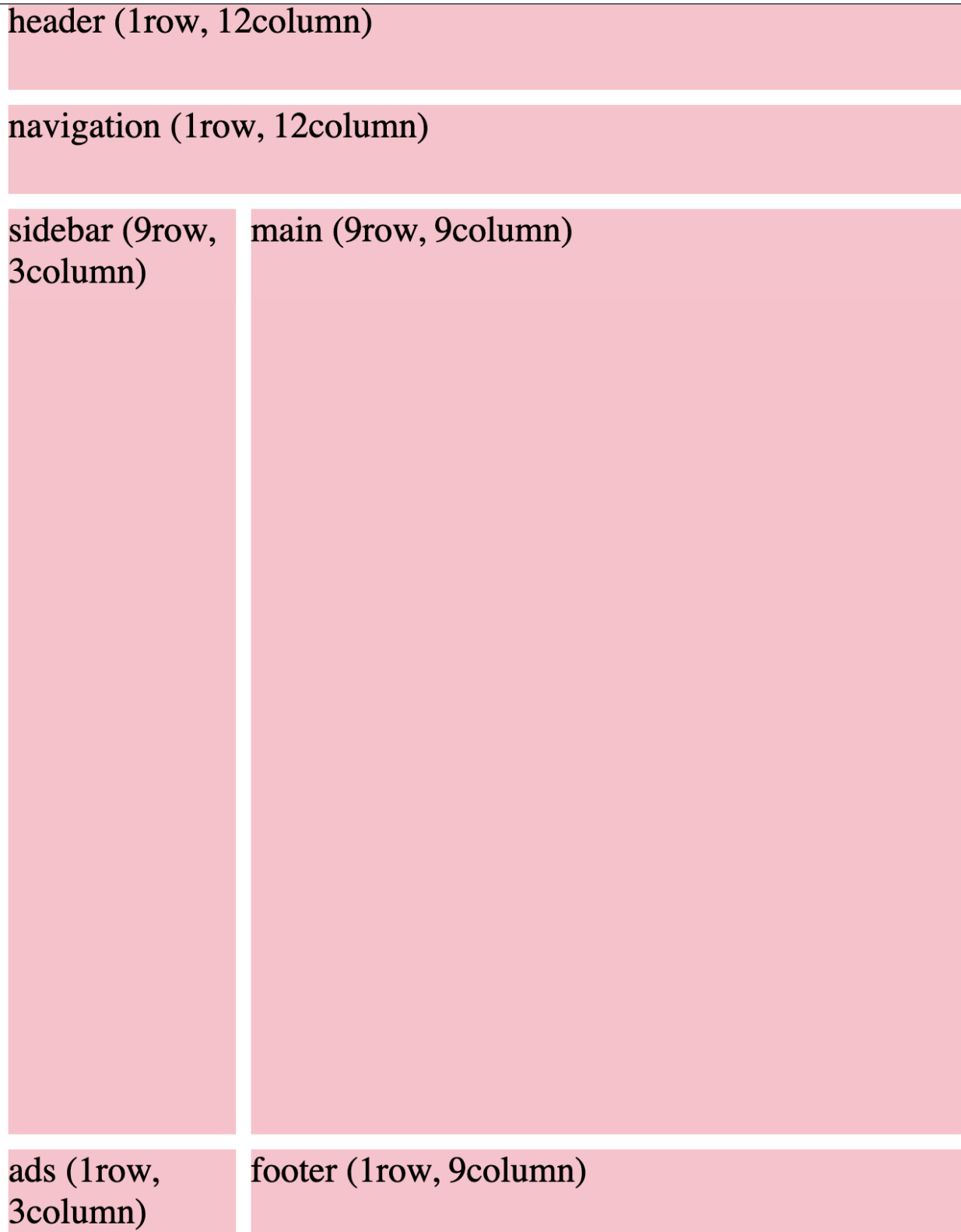
Practice Questions

Qs1. Create the following layout using CSS Grid :



- Divide the grid into 12 rows & 12 columns.
- Give a gap of 10px between each row and column
- Set the sizing of individual boxes as shown in the image

Qs2. Use Media Queries to change the above layout the given one when the viewport width goes below 720px.



Qs3. Try to complete this code to create a web page loader using CSS animations.

Loader



HTML Code :

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width,
initial-scale=1.0" />
<title>CSS</title>
<link rel="stylesheet" href="style.css" />
</head>
<body>
<h1>Loader</h1>
<div class="loader"></div>
</body>
</html>
```


CSS Code :

```
.loader {  
border: 16px solid #f3f3f3;  
border-top: 16px solid goldenrod;  
border-radius: 50%;  
width: 120px;  
height: 120px;  
animation: spin 2s linear infinite;  
}  
  
@keyframes spin {  
0% {  
/* Set rotation to 0 degrees */  
}  
100% {  
/* Set rotation to 360 degrees */  
}  
}
```

CSS (Part 1)

Practice Solutions

Ans 1

HTML Code

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width,
initial-scale=1.0" />
<title>CSS</title>
<link rel="stylesheet" href="style.css" />
</head>
<body>
<div class="container">
<div class="header">header</div>
<div class="navigation">navigation</div>
<div class="sidebar">sidebar</div>
<div class="main">main</div>
<div class="ads">ads </div>
<div class="footer">footer </div>
</div>
</body>
</html>
```

snehasupta7385@gmail.com

CSS Code

```
.container {  
margin: 0;  
padding: 0;  
height: 100vh;  
display: grid;  
grid-template-columns: repeat(12, 1fr);  
grid-template-rows: repeat(12, 1fr);  
grid-gap: 10px;  
}  
  
.container div {  
background-color: pink;  
font-size: 1.5rem;  
}  
  
.header {  
grid-column: 1 / span 12;  
}  
  
.sidebar {  
grid-column: 1 / span 3;  
grid-row: 2 / span 10;  
}  
  
.navigation {
```

```
grid-column: 4 / span 6;
}

.ads {
grid-column: 10 / span 3;
grid-row: 2 / span 10;
}

.main {
grid-column: 4 / span 6;
grid-row: 3 / span 9;
}

.footer {
grid-column: 1 / span 12;
}
```

Ans 2

HTML Code will remain the same as answer 1.

CSS Code

```
@media (max-width: 720px) {
.header {
grid-column: 1 / span 12;
}
```

```
.navigation {  
grid-column: 1 / span 12;  
grid-row: 2;  
}  
  
.sidebar {  
grid-column: 1 / span 3;  
grid-row: 3 / span 9;  
}  
  
.main {  
grid-column: 4 / span 9;  
grid-row: 3 / span 9;  
}  
  
.ads {  
grid-column: 1 / span 3;  
grid-row: 12;  
}  
  
.footer {  
grid-column: 4 / span 9;  
grid-row: 12;  
}  
}
```

Ans 3

Complete CSS:

```
.loader {  
border: 16px solid #f3f3f3;  
border-top: 16px solid goldenrod;  
border-radius: 50%;  
width: 120px;  
height: 120px;  
animation: spin 2s linear infinite;  
}  
  
@keyframes spin {  
0% {  
transform: rotate(0deg);  
}  
100% {  
transform: rotate(360deg);  
}  
}
```