

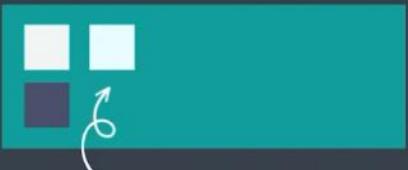
Day-10 css part-6

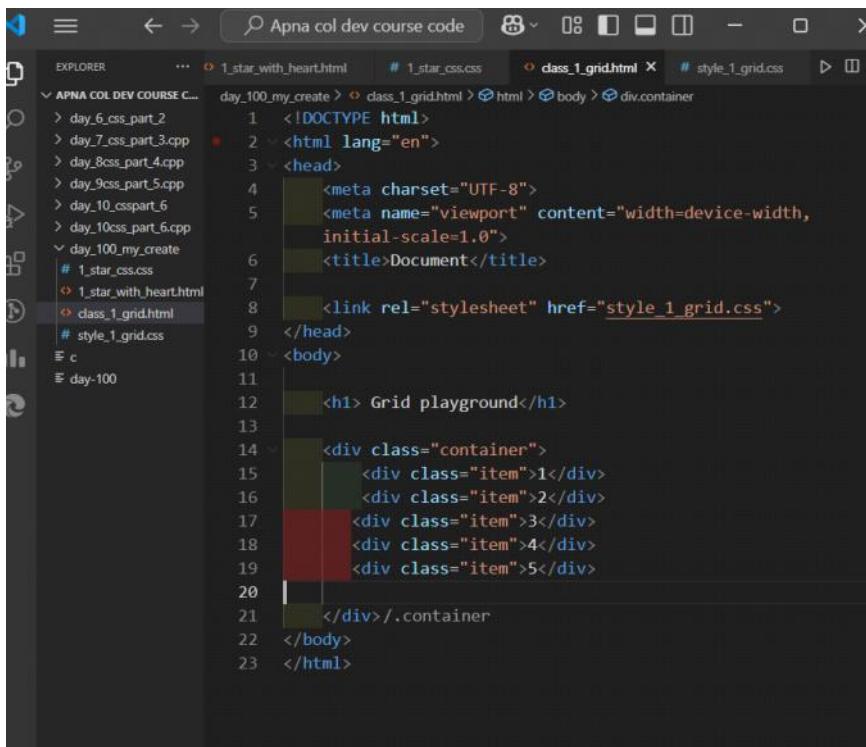
Sunday, May 18, 2025 12:03 AM

Name	Type
0. Topic List.jpg	JPG
01. What is Grid-.mp4	MP4
02. Grid Model.mp4	MP4
03. Grid Template.mp4	MP4
04. Grid Template (repeat).mp4	MP4
05. Grid Gaps.mp4	MP4
06. Grid Columns.mp4	MP4
07. Grid Rows.mp4	MP4
08. Grid Properties.mp4	MP4
09. Animation in CSS.mp4	MP4
10. Animation Shorthand.mp4	MP4
11. _ in Animation.mp4	MP4
12. Media Queries.mp4	MP4
13. Media Queries (Orientation).mp4	MP4
14. Pet Adoption Page.mp4	MP4
15. Download Starter HTML.pdf	Misc
16. z-index.mp4	MP4
17. CSS Part 6 (Qs).pdf	Misc

CSS Grid

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

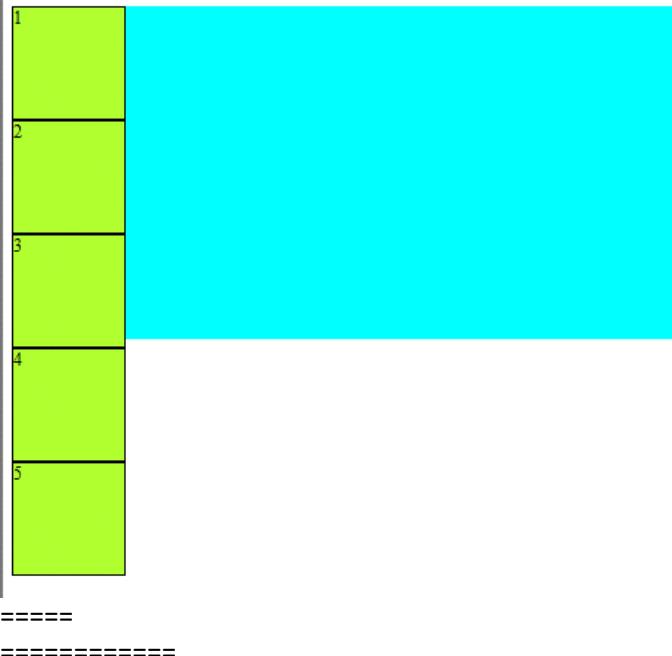
```
container {  
    display: grid;  
}  
  
grid container  
  
grid item
```



```
1 <!DOCTYPE html>  
2 <html lang="en">  
3 <head>  
4     <meta charset="UTF-8">  
5     <meta name="viewport" content="width=device-width,  
6         initial-scale=1.0">  
7     <title>Document</title>  
8     <link rel="stylesheet" href="style_1_grid.css">  
9 </head>  
10 <body>  
11     <h1> Grid playground</h1>  
12  
13     <div class="container">  
14         <div class="item">1</div>  
15         <div class="item">2</div>  
16         <div class="item">3</div>  
17         <div class="item">4</div>  
18         <div class="item">5</div>  
19     </div>/.container  
20  
21 </body>  
22 </html>
```

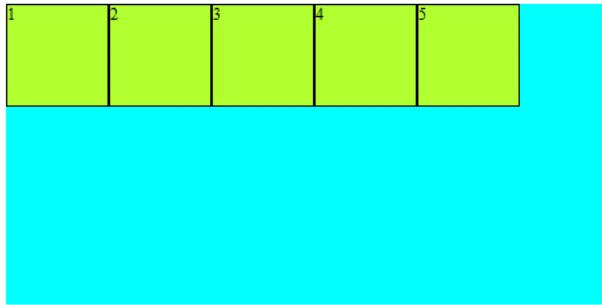
```
↳ 1_star_with_heart.html      # 1_star_css.css      ↳ class_1_grid.html
day_100_my_create > # style_1_grid.css > .item
1   .container {
2     width: 600px;
3     height: 300px;
4     background-color: aqua;
5   }
6
7   .item{
8     width: 100px;
9     height: 100px;
10    background-color: greenyellow;
11    border: 2px solid black;
12
13
14 }
```

Grid playground



When give :
Display: flex;

Grid playground



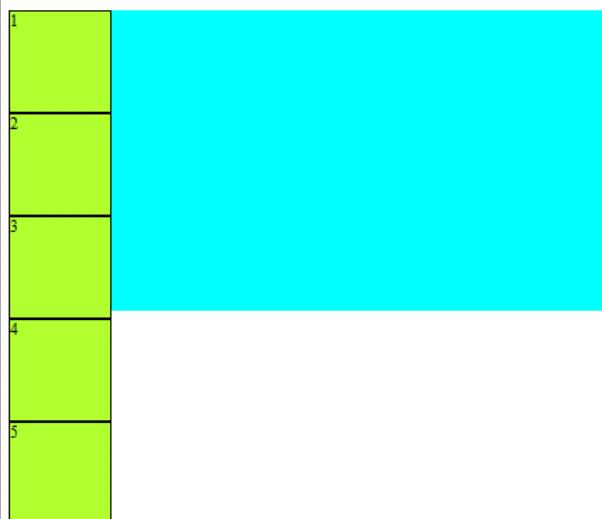
=====

When give:

Display:grid;

We can see it come to horizontal axis .

Grid playground



=====

=====

When give: height is reduce to 100px to 50px for item div

Grid playground



It take some space between them .

=====

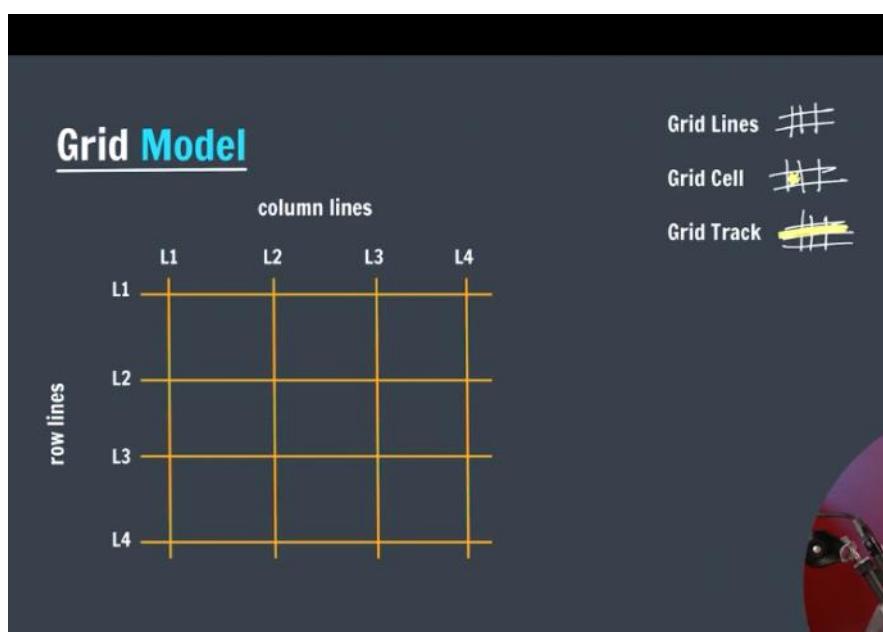
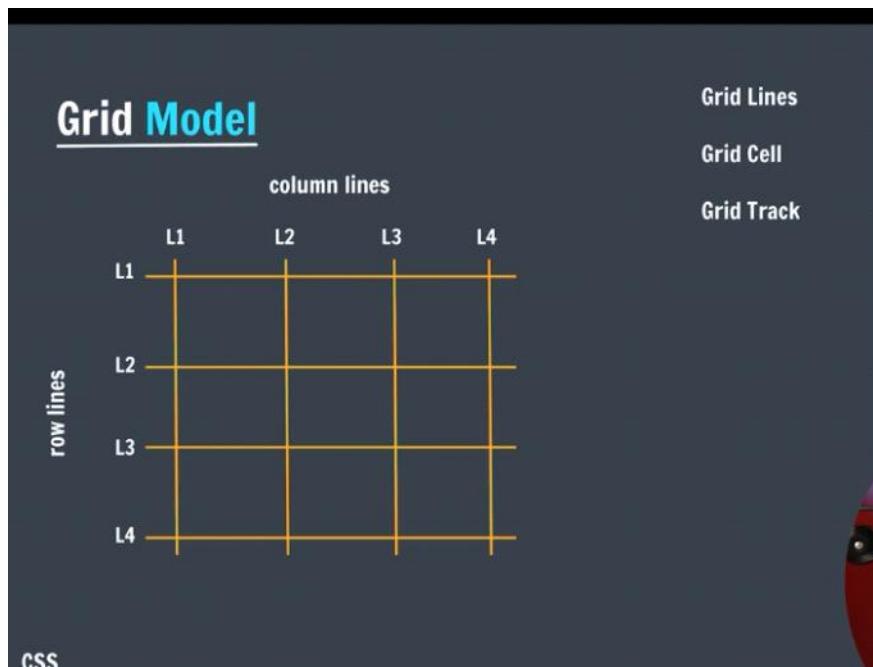
When width is 20px;

```
100_my_create > # style_1_grid.css > .item
1  v .container {
2      width: 600px;
3      height: 300px;
4      background-color: aqua;
5      display: grid;
6
7  }
8
9  v .item{
10     width: 100px;
11     height: 20px;
12     background-color: greenyellow;
13     border: 2px solid black;
14
15  }
16 }
```

Grid playground



Class 2: Grid Model



Grid line:

Purple color dot line indicate the grid line:

Di... 275 x 517 1 N. ⌂

Grid playground

```
<!DOCTYPE html>
<html lang="en">
  <head> ... </head>
  <body>
    <h1> Grid playground</h1>
    ... <div class="container"> ... </div> (grid)
      == $0
      <!-- Code injected by live-server -->
      <script> ... </script>
  html body div.container
```

Styles Computed Layout Event Listeners >

element.style { }

```
.container {
  width: 600px;
  height: 300px;
  background-color: #00FFFF;
  display: grid;
}
```

```
div {
  display: block;
  unicode-bidi: isolate;
}
```

div.container 600 x 300

Color #000000
Font 16px "Times New Roman"
Background #00FFFF

ACCESSIBILITY
Name generic
Role generic
Keyboard-focusable

=====

Class 3: Grid Template

They define the lines and track sizing.

Grid Template

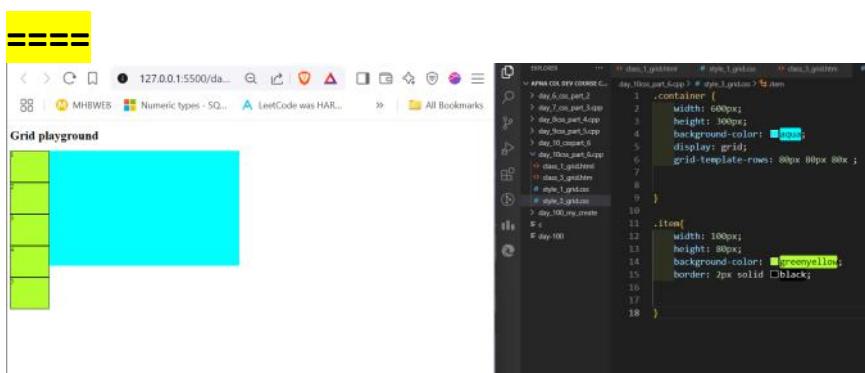
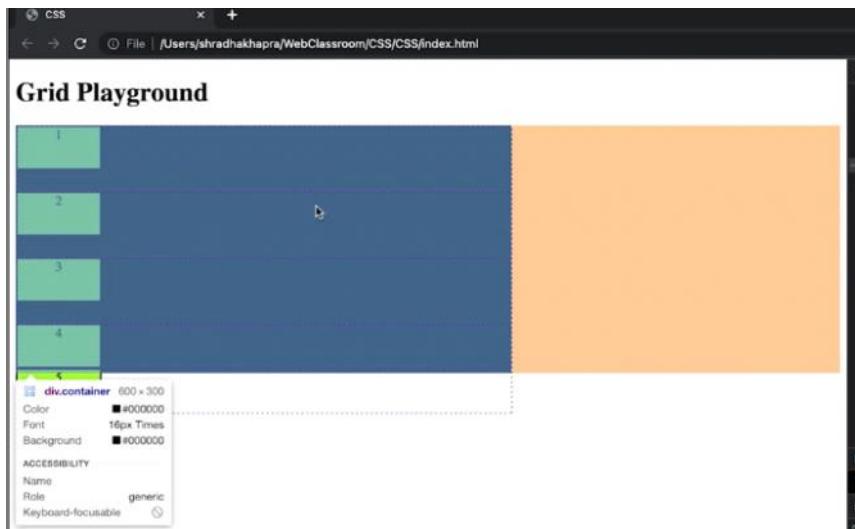
They define the lines & track sizing

grid-template-rows

grid-template-columns

```
< index.html      # style.css >

# style.css > .container
1   .container {
2     width: 600px;
3     height: 300px;
4     background-color: black;
5     display: grid;
6     grid-template-rows: 80px 80px 80px;
7   }
8
9   .item {
10    width: 100px;
11    height: 50px;
12    background-color: greenyellow;
13    border: 2px solid black;
14    text-align: center;
15 }
```



When height: 40px;



Grid playground



← →

🔍 Apna col dev course code



v

```
...  ↗ class_1_grid.html × # style_1_grid.css ↗ class_3_grid.htm ↗ style_3_grid.css
DEV COURSE C... day_10css_part_6.cpp > # style_3_grid.css > .item
ss_part_2.css
ss_part_3.css
ss_part_4.css
ss_part_5.css
csspart_6
ss_part_6.css
_grid.html
_grid.htm
_grid.css
_grid.css
_my_create
1   .container {
2     width: 600px;
3     height: 300px;
4     background-color: aqua;
5     display: grid;
6     grid-template-rows: 50px 50px 50px 50px;
7
8     grid-template-columns: 100px 100px 100px;
9
10    }
11
12
13   .item{
14     width: 100px;
15     height: 50px;
16     background-color: greenyellow;
17     border: 2px solid black;
18     text-align: center;
19
20
21 }
```

Dimensions: Responsive 617 x 517 100% No throttling

Grid playground

```

<!DOCTYPE html>
<html lang="en">
  <head> ...
  <body>
    <h1> Grid playground </h1>
    <div class="container"> ...
      <!-- Code injected by live-server -->
      <script> ... </script>
    </div>
  </body>
</html>

```

Styles Computed Layout Event Listeners

element.style { }

body { user agent stylesheet }

margin: 8px;

margin: 8px; border: 1px solid black; padding: 8px; width: 600px; height: 300px; background-color: #f0f0ff; display: flex; justify-content: center; align-items: center; font-size: 1em; font-weight: bold; color: black; font-family: sans-serif; }

grid-template-columns: 1fr 1fr 1fr; grid-template-rows: 1fr 1fr; gap: 10px; width: 600px; height: 300px; background-color: #f0f0ff; display: grid; grid-template-columns: 1fr 1fr 1fr; grid-template-rows: 1fr 1fr; gap: 10px; margin: 8px; border: 1px solid black; padding: 8px; width: 600px; height: 300px; background-color: #f0f0ff; display: flex; justify-content: center; align-items: center; font-size: 1em; font-weight: bold; color: black; font-family: sans-serif; }

== for row

Grid playground

```

<!DOCTYPE html>
<html lang="en">
  <head> ...
  <body>
    <div class="item">1</div>
    <div class="item">2</div>
    <div class="item">3</div>
    <div class="item">4</div>
    <div class="item">5</div>
  </div>
  <!-- Code injected by live-server -->
  <script> ... </script>
</body>
</html>

```

html body div.container

Styles Computed Layout Event Listeners

element.style { }

.container { style_3_grid_614105795:1 width: 600px; height: 300px; background-color: #aqua; display: grid; grid-template-rows: 50px 50px 50px 50px; grid-template-columns: 100px 100px 100px; }

== for column:

Grid playground

```

<!DOCTYPE html>
<html lang="en">
  <head> ...
  <body>
    <div class="item">1</div>
    <div class="item">2</div>
    <div class="item">3</div>
    <div class="item">4</div>
    <div class="item">5</div>
  </div>
  <!-- Code injected by live-server -->
  <script> ... </script>
</body>
</html>

```

html body div.container

Styles Computed Layout Event Listeners

element.style { }

.container { style_3_grid_614105795:1 width: 600px; height: 300px; background-color: #aqua; display: grid; grid-template-rows: 50px 50px 50px 50px; grid-template-columns: 100px 100px 100px; }

==

Take grid: 5 time column and 4 time row

127.0.0.1:5500/day... 278 x 517 100% No throttling

Grid playground

```

<!DOCTYPE html>
<html lang="en">
  <head> ...
  <body>
    <h1> Grid playground </h1>
    <div class="container"> ...
      <!-- Code injected by live-server -->
      <script> ... </script>
    </div>
  </body>
</html>

```

html body div.container

Styles Computed Layout Event Listeners

element.style { }

.container { style_my_3_grid_614105795:1 width: 600px; height: 300px; background-color: #aqua; display: grid; grid-template-columns: 100px 100px 100px 100px 100px; grid-template-rows: 50px 50px 50px 50px; }

.item { width: 100px; height: 50px; background-color: #greenyellow; border: 2px solid black; text-align: center; }

==

```

*** <div class="container"></div> == $0
<div class="item">1</div>
<div class="item">2</div>
<div class="item">3</div>
<div class="item">4</div>
<div class="item">5</div>
</div>

<script> ...</script>
</body>
</html>

```

html body div.container

Styles Computed Layout Event Listeners >

Filter :hover .cls + ☰ ☱

element.style { }

```

.container {
    width: 600px;
    height: 300px;
    background-color: aqua;
    display: grid;
    grid-template-rows: 50px 50px 50px 50px;
    grid-template-columns: 100px 100px 100px
                          100px 100px;
}

```

=====

Another way to declare grid:

Auto .

If I write 3 time auto means
3 row or 3 column.

```

height: 300px;
background-color: black;
display: grid;
grid-template-rows: 50px 50px 50px 50px 50px;
grid-template-columns: auto auto auto;
}

```

Grid Playground

div.container 600 x 300

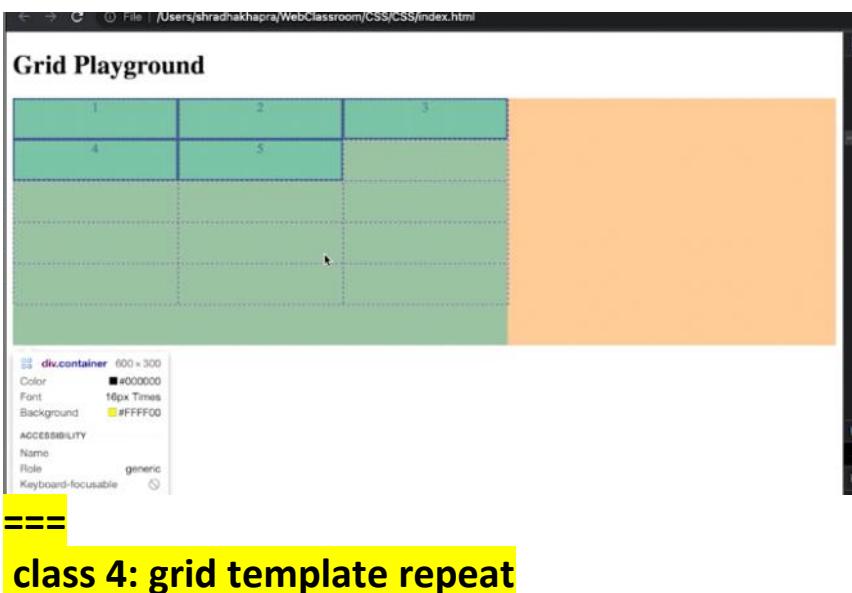
Color #000000
Font 16px Times
Background #FFFF00

ACCESSIBILITY
Name
Role generic
Keyboard-focusable

==

**When delete the width and height of item
Then every item take full grid cell .**

```
.container {  
    width: 600px;  
    height: 300px;  
    background-color: yellow;  
    display: grid;  
    grid-template-rows: 50px 50px 50px 50px 50px;  
    grid-template-columns: 200px 200px 200px;  
}  
  
.item {  
    /* width: 100px;  
     * height: 50px; */  
    background-color: greenyellow;  
    border: 2px solid black;  
    text-align: center;  
}
```





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

====

Repeat(3,1fr) ;

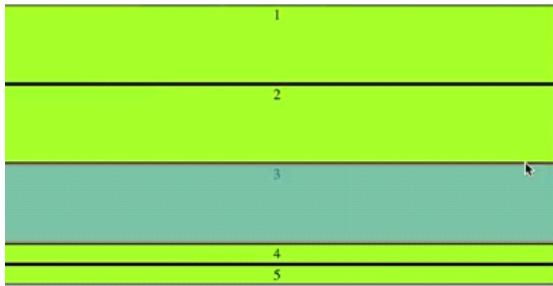
Means 1st 3 items take same space.

Other taking space don't know.

```
yle.css > .container
.container {
    width: 600px;
    height: 300px;
    background-color: yellow;
    display: grid;
    /* grid-template-rows: 50px 50px 50px 50px 50px;
    grid-template-columns: 200px 200px 200px; */
    grid-template-rows: repeat(3, 1fr);
}

.item {
    /* width: 100px;
    height: 50px; */
    background-color: greenyellow;
    border: 2px solid black;
    text-align: center;
}
```

Grid Playground



```
=====
5  display: grid;
6  /* grid-template-rows: 50px 50px 50px 50px 50px;
7  grid-template-columns: 200px 200px 200px; */
8  grid-template-rows: repeat(5, 1fr);
9 }
10
11 .item {
```

Playground

```
600x60
■ #000000
16px Times
ind #ADFF2F
ILITY
Aa 17.12
generic
i-focusable
```

```
index.html # style.css X
# style.css > .container
1   .container {
2     width: 600px;
3     height: 300px;
4     background-color: yellow;
5     display: grid;
6     /* grid-template-rows: 50px 50px 50px 50px 50px;
7     grid-template-columns: 200px 200px 200px; */
8     grid-template-rows: repeat(5, 1fr);
9     grid-template-columns: repeat(3, 1fr);
10 }
11
12 .item {
13   /* width: 100px;
14   height: 50px; */
15   background-color: greenyellow;
16   border: 2px solid black;
17   text-align: center;
18 }
```

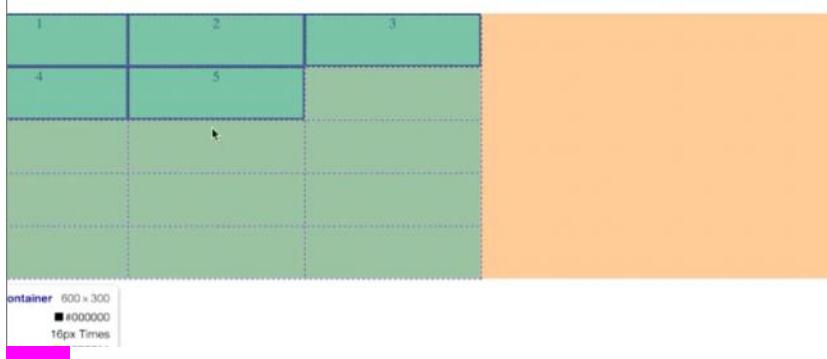
```
/* grid-template-rows: 50px 50px 50px 50px
grid-template-columns: 200px 200px 200px; */
grid-template-rows: repeat(5, 1fr);
grid-template-columns: repeat(3, 1fr);
```

When height and width are not given

Then all item take full cell space.

Answer:

Playground



Class 5: grid gaps

Grid Gaps

They define the gaps between the lines

row-gap

column-gap

grid-gap : rowGap columnGap

div.container {
 width: 600px;
 height: 300px;
 background-color: yellow;
 display: grid;
 grid-template-rows: 1fr 1fr 1fr 1fr;
 grid-template-columns: repeat(3, 1fr);
 row-gap: 10px;
}

```
style.css > .container  
1 .container {  
2     width: 600px;  
3     height: 300px;  
4     background-color: yellow;  
5     display: grid;  
6     grid-template-rows: 1fr 1fr 1fr 1fr;  
7     grid-template-columns: repeat(3, 1fr);  
8     row-gap: 10px;  
9 }
```

Given :

row gap : 10px;

Grid Playground

The screenshot shows a CSS playground interface. On the left, there's a preview area with a green grid container containing five items labeled 1 through 5. To the right of the preview is the HTML structure, which consists of a single `<div>` element with a class of `container`. The right side of the interface contains the CSS code editor.

```
5 display: grid;
6 grid-template-rows: 1fr 1fr
7 grid-template-columns: repeat(3, 1fr)
8 /* row-gap: 10px;
9  column-gap: 30px; */
10 grid-gap: 10px 30px;
11 }
12 }
```

Grid Playground

The screenshot shows a CSS playground interface. On the left, there's a preview area with a green grid container containing five items labeled 1 through 5. The background of the grid cells is purple. To the right of the preview is the HTML structure, which consists of a single `<div>` element with a class of `container`. The right side of the interface contains the CSS code editor.

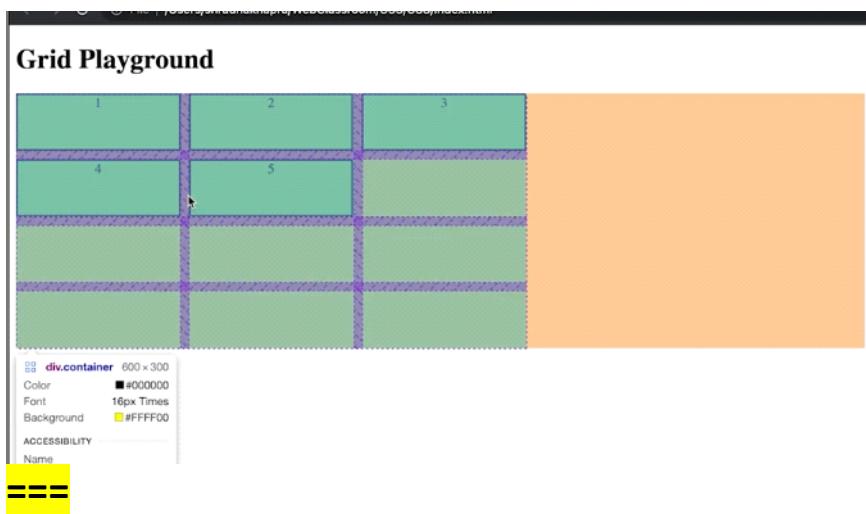
====

**Grid-gap: 10px;
Means row and column all take same gap.**

====

When delete the height and width

```
# style.css > .item
1   .container {
2     width: 600px;
3     height: 300px;
4     background-color: yellow;
5     display: grid;
6     grid-template-rows: 1fr 1fr 1fr 1fr;
7     grid-template-columns: repeat(3, 1fr);
8     /* row-gap: 10px;
9     column-gap: 30px; */
10    grid-gap: 10px;
11  }
12
13 .item {
14   /* width: 100px;
15   height: 50px; */
16   background-color: greenyellow;
17   border: 2px solid black;
18   text-align: center;
19 }
```



Class 6 : grid columns

Grid Columns

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

grid-column-start : line_number

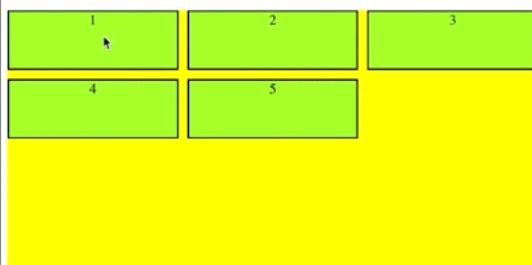
grid-column-end : line_number

grid-column : start_col / end_col

grid-column : start_col / span number

```
index.html      # style.css  X
# style.css > .item
 1   width: 600px;
 2   height: 300px;
 3   background-color: yellow;
 4   display: grid;
 5   grid-template-rows: 1fr 1fr 1fr 1fr;
 6   grid-template-columns: repeat(3, 1fr);
 7   /* row-gap: 10px;
 8   column-gap: 30px; */
 9   grid-gap: 10px;
10 }
11
12
13 .item {
14   /* width: 100px;
15   height: 50px; */
16   background-color: greenyellow;
17   border: 2px solid black;
18   text-align: center;
19 }
20
```

Grid Playground



====
First:

Give individual class of item one

```
body {
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
  grid-template-rows: 1fr 1fr;
}

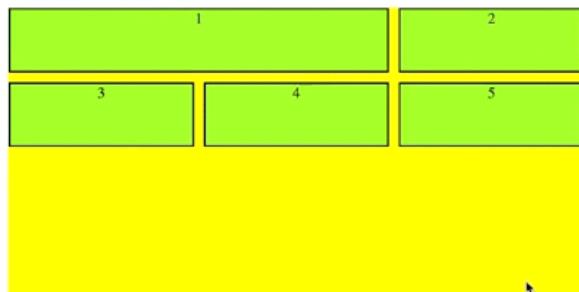
.item {
  width: 100px;
  height: 50px;
  background-color: greenyellow;
  border: 2px solid black;
  text-align: center;
}

.one {
  grid-column-start: 1;
  grid-column-end: 3;
}
```

```
11   }
12
13   .item {
14     /* width: 100px;
15     height: 50px; */
16     background-color: greenyellow;
17     border: 2px solid black;
18     text-align: center;
19   }
20
21   .one {
22     grid-column-start: 1;
23     grid-column-end: 3;
24   }
25 }
```

< → C File | /Users/shradhakapra/WebClassroom/CSS/CSS/index.html

Grid Playground



==

When give : width and height

```
.item {
  width: 100px;
  height: 50px;
  background-color: greenyellow;
}
```

Here 2nd cell space take 1st item . Though it's Width is small.

div.container 600 × 300
Color #000000
Font 16px Times
Background #FFFF00

==other way to right:

```
height: 50px; /*  
background-color: greenyellow;  
border: 2px solid black;  
text-align: center;  
}  
  
.one {  
/* grid-column-start: 1;  
grid-column-end: 4; */  
grid-column: 1/4  
}
```

body 998 × 358.44
Color #000000
Font 16px Times
Margin 8px

ACCESSIBILITY
Name
Role generic
Keyboard-focusable

=====

=====

Class 7: Grid Rows

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

Grid Rows

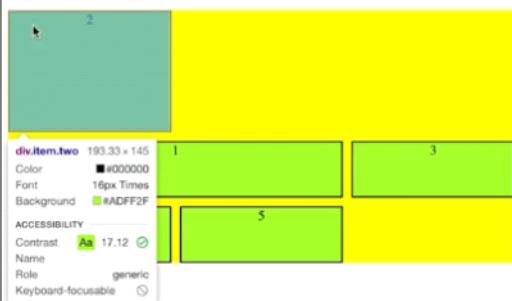
Defines an item's starting & ending position inside the row

grid-row-start : line_number 1

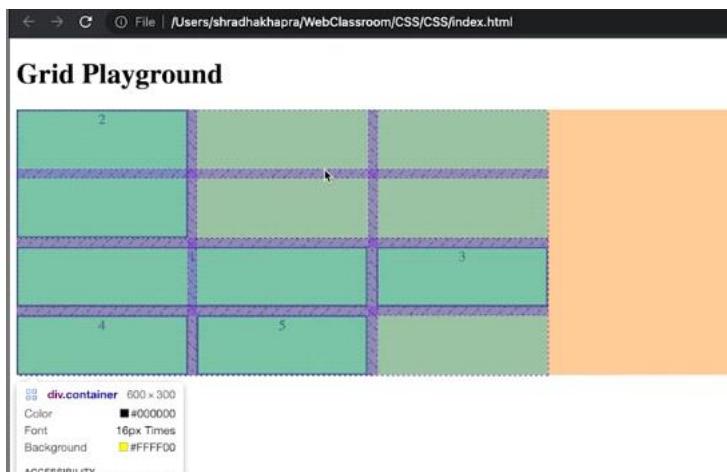
grid-row-end : line_number 3

```
19  }
20
21 .one {
22 /* grid-column-start: 1;
23 grid-column-end: 4; */
24 grid-column: 1 / span 2;
25 }
26
27 .two [
28 /* grid-row-start: 1;
29 grid-row-end: 3; */
30 grid-row: 1/
31 ]
32 }
```

Grid Playground



If inspect:



=====

=====

Class : 08

Common Properties

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

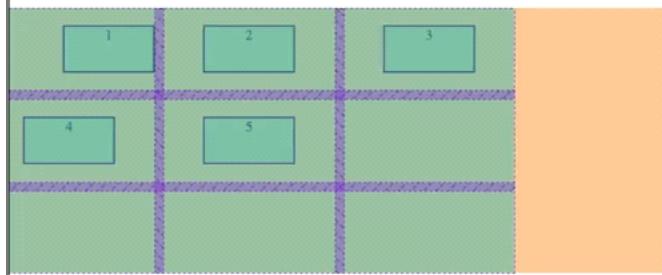
- align-items (container)
- align-self (item)] Vertical

- place-items
- place-self

```
# style.css > .container
1   .container {
2     width: 600px;
3     height: 300px;
4     background-color: yellow;
5     display: grid;
6     grid-template-rows: 1fr 1fr 1fr;
7     grid-template-columns: repeat(3, 1fr);
8     /* row-gap: 10px;
9     column-gap: 30px; */
10    grid-gap: 10px;
11    justify-items: center;
12    align-items: center;
13  }
14
15  .item {
16    width: 100px;
17    height: 50px;
18    background-color: greenyellow;
19    border: 2px solid black;
20    text-align: center;
21  }
22
23  .one {
24    justify-self: end;
```



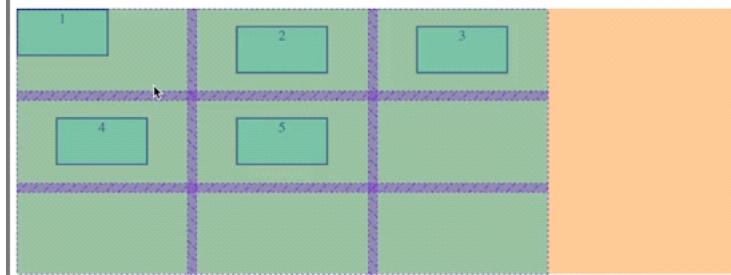
Grid Playground



```
div.container {  
    display: grid;  
    grid-template-columns: 1fr 1fr 1fr;  
    grid-template-rows: 1fr 1fr;  
    gap: 10px;  
}  
  
div.container div {  
    border: 1px solid black;  
    padding: 10px;  
    width: 100%;  
    height: 100%;  
}
```

```
22 }  
23  
24 .one {  
25   place-self: start;  
26 }  
27 /*
```

Grid Playground



```
div.container {  
    display: grid;  
    grid-template-columns: 1fr 1fr 1fr;  
    grid-template-rows: 1fr 1fr;  
    gap: 10px;  
}
```

=====

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

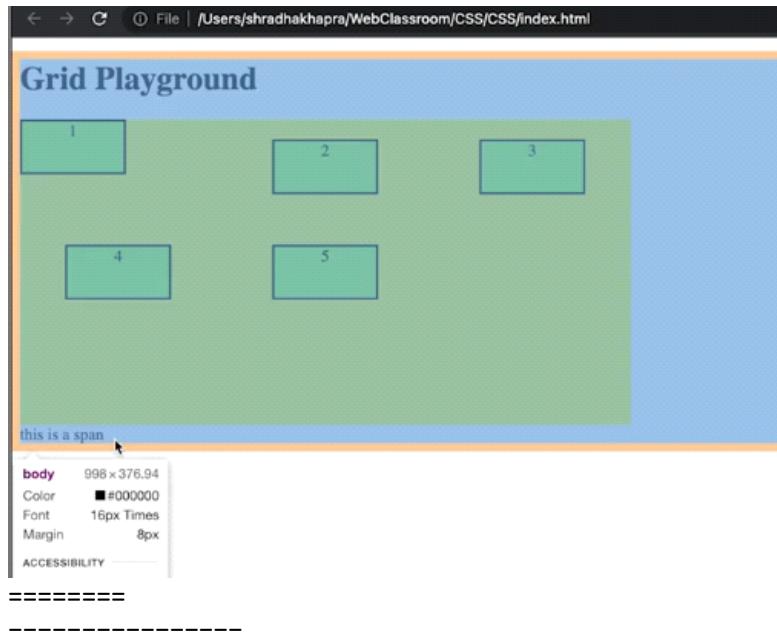
It works as block element for all items.

Here span take new line

```

<body>
  <h1>Grid Playground</h1>
  <div class="container">
    <div class="item one">1</div>
    <div class="item two">2</div>
    <div class="item">3</div>
    <div class="item">4</div>
    <div class="item">5</div>
  </div>
  <span>this is a span</span>
</body>
</html>

```



If we create :

Display: inline-grid;

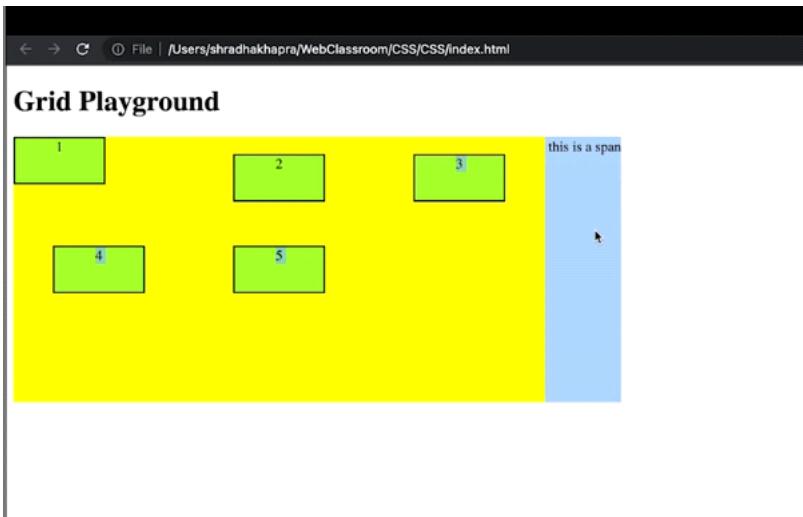
Then new element take space.

```

# style.css > ⚒ .container
1   .container {
2     width: 600px;
3     height: 300px;
4     background-color: yellow;
5     display: inline-grid;
6     grid-template-rows: inline-block
7     grid-template-columns: inline-flex
8     /* row-gap: 10px */ inline-flexbox

```

o/p:



=====

=====

Class :9

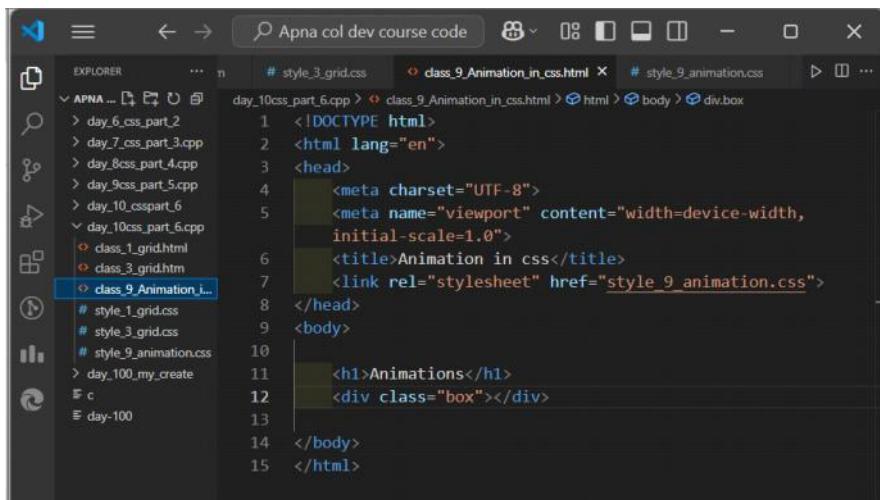
```
@keyframe myName {  
    from { font-size : 20px; }  
    to { font-size : 40px; }  
}
```

Here, myname is animation name.

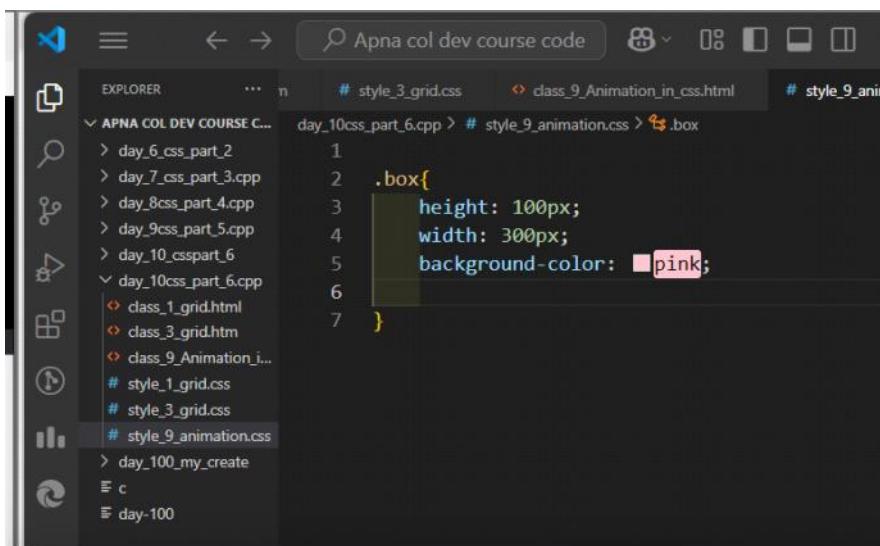
From "start state"

To "final state".

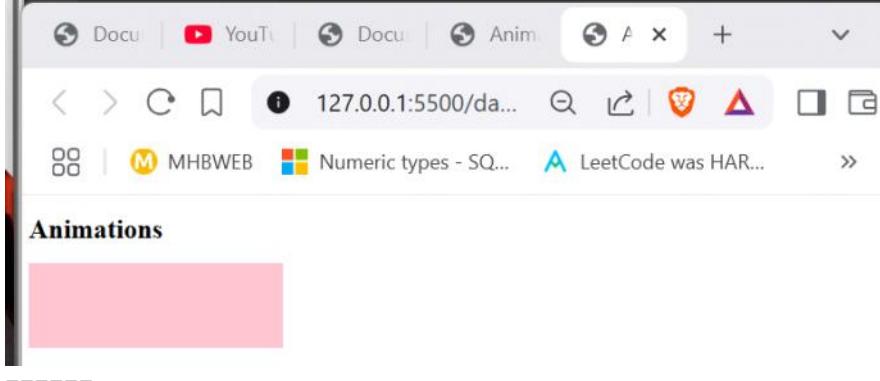
=====



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Animation in css</title>
    <link rel="stylesheet" href="style_9_animation.css">
</head>
<body>
    <h1>Animations</h1>
    <div class="box"></div>
</body>
</html>
```



```
.box{
    height: 100px;
    width: 300px;
    background-color: pink;
```



CSS Animations

- animation-name
- ✓ animation-duration
- ✓ animation-timing-function
- animation-delay
- animation-iteration-count
- animation-direction

====

CSS Animations

```
div {  
    animation-name: fontAnimation;  
    animation-duration: 3s;  
}  
  
• animation-name  
✓ animation-duration  
✓ animation-timing-function  
✓ animation-delay  
• animation-iteration-count 1/2/3... infinite  
• animation-direction  
    ↗ normal  
    ↗ reverse  
    ↗ alternate
```

seacrh:

Animation dircetion MDN:

The screenshot shows the MDN Web Docs page for CSS `animation-direction`. It features a sidebar with navigation links like 'Try it', 'Syntax', 'Formal definition', etc. The main content includes a 'Try it' section with a 'CSS Demo: animation-direction' interface where users can select from 'normal', 'reverse', 'alternate', or 'alternate-reverse'. Below this is a 'Syntax' section with code examples:

```

1  .box{
2      height: 100px;
3      width: 300px;
4      background-color: pink;
5
6      /* properties for animation */
7
8      animation-name: fontAnimation;
9      animation-duration: 2s;
10     animation-timing-function: ease-in;
11     animation-delay: 0;
12     animation-iteration-count: 3; /* how many time same
13     proces repeat*/
14
15     animation-direction: normal;
16
17     /* animation-direction : normal/ alternate /alternate
18     reverse / reverse */
19
20 }
21
22 /* Syntax : for creating animation
23
24 @keyframe fontAnimation {
25
26     from {
27         /* starting condition
28     }
29     to {
30         /* ending condition
31     }
32 }
33

```

=====
=====

The screenshot shows a code editor with several tabs open. The active tab contains CSS code for a box with an animation:

```

1  .box{
2      height: 100px;
3      width: 300px;
4      background-color: pink;
5
6      /* properties for animation */
7
8      animation-name: fontAnimation;
9      animation-duration: 2s;
10     animation-timing-function: ease-in;
11     animation-delay: 0;
12     animation-iteration-count: 3; /* how many time same
13     proces repeat*/
14
15     animation-direction: normal;
16
17     /* animation-direction : normal/ alternate /alternate
18     reverse / reverse */
19
20 }
21
22 /* Syntax : for creating animation
23
24 @keyframe fontAnimation {
25
26     from {
27         /* starting condition
28     }
29     to {
30         /* ending condition
31     }
32 }
33

```

```
31     } /* ending condition
32 }
33 }
34
35 */
36
37 @keyframes fontAnimation {
38   from {
39     font-size: 20px;
40   }
41   to {
42     font-size:40px;
43   }
44 }
45
46 }
```

O/P:

Animations

This is a box



Size increasing

Animations

This is a box



=====

=====

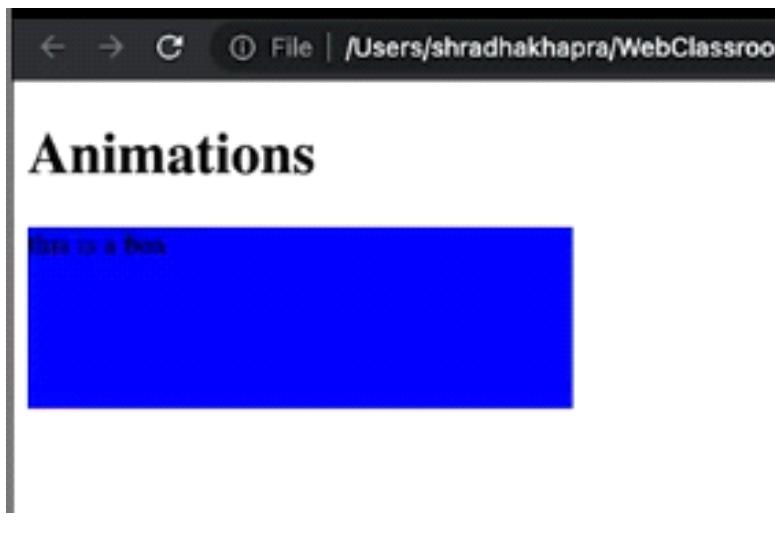
NEXT Animation:

```
3 .box{  
6     background-color: pink;  
7  
8     /* properties for animation */  
9  
10    /* animation-name: fontAnimation;  
11  
12    animation-name: colorAnimate;  
13    animation-duration: 2s ;  
14    animation-timing-function: ease-in  
15    animation-delay: 0;  
16    animation-iteration-count: 3; /* h  
17    process repeat*/  
18    animation-direction: normal;  
19  
20    /* animation-direction : normal/ a  
21    reverse / reverse */  
22  
23  
24 }
```

```
22  
23 @keyframes colorAnimate {  
24     from {  
25         background-color: green;  
26     }  
27     to {  
28         background-color: blue;  
29     }  
30 }
```

o/p:

Color change bule to green



=====

Give:

Animation-name: widthAnimate;

In box class.

```
63  }
64
65 /* work:
66 when run:
67 change width 20px to 400px left to right
68 continuously 3 times
69 */
70 @keyframes widthAnimate
71 {
72 from {
73   width: 10px;
74 }
75 to {
76   width: 400px;
77 }
78
79 }
```

====

```
  animation-name: widthAnimate;
  animation-duration: 3s;
  animation-timing-function: 3ms;
  animation-delay: 0; 3s;
  animation-iteration-count: infinite;
  animation-direction: normal;
}
```

When set

Animation-iteration-count: infinte;

Animation-duration : 3s;

It animate 3 s continuously.

=====

Class : 10 Animaiton shorthand property

Animation Shorthand

animation : myName 2s linear 3s infinite normal

```
=====
style.css > ...
5  /* animation-name: widthAnimate;
6  animation-duration: 3s;
7  animation-timing-function: ease-in;
8  animation-delay: 0;
9  animation-iteration-count: infinite;
10 animation-direction: normal; */

11 animation: widthAnimate 3s ease-in 0s infinite normal;
12 }

13 @keyframes widthAnimate {
14   from {
15     width: 10px;
16   }
17   to {
18     width: 400px;
19   }
20 }
```

when I want to more property set in a Single syntax or @ keyframe sector::

```
10   /* animation iteration count: infinite,
11    animation-direction: normal; */

12   animation: widthAnimate 3s ease-in 0s 2 normal;
13 }

14 @keyframes widthAnimate {
15   from {
16     width: 10px;
17     background-color: green;
18   }
19   to {
20     width: 400px;
21     background-color: blue;
22     border-radius: 15px;
23   }
24 }
```

Class : 11

% in Animation

```
@keyframe myName {  
    0% { font-size : 20px; }  
    50% { font-size : 30px; }  
    100% { font-size : 40px; }  
}
```

Stage :1
Stage: 2
Stage: 3

=====

4 stage animation :

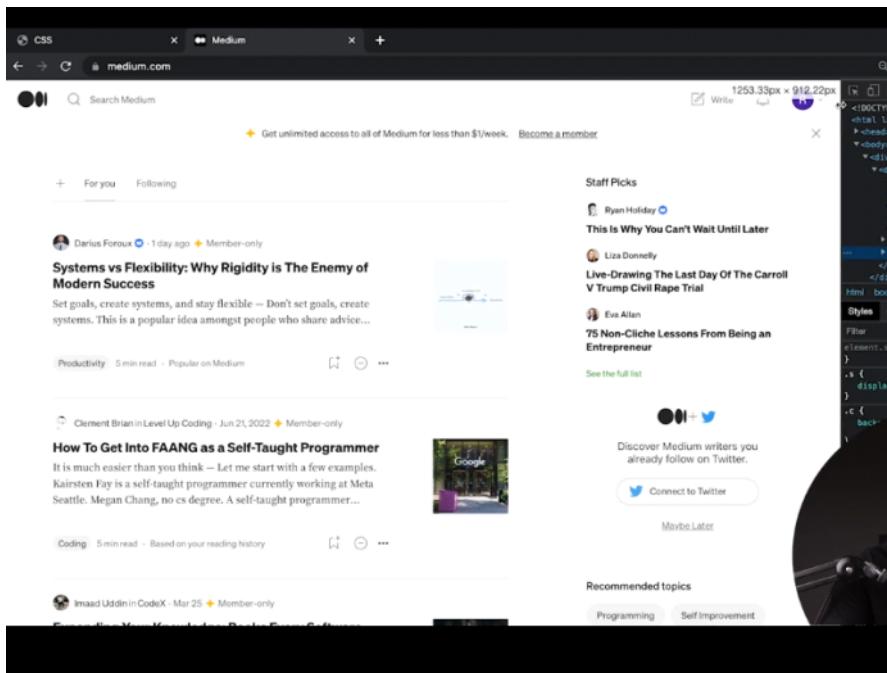
```
105  
106  
107  
108 @keyframes colAnimate2 {  
109     0% {  
110         background-color: green;  
111     }  
112     50% {  
113         background-color: yellow;  
114     }  
115     80%{  
116         background-color: red;  
117     }  
118     100% {  
119         background-color: blue;  
120     }  
121 }  
122 }
```

=====

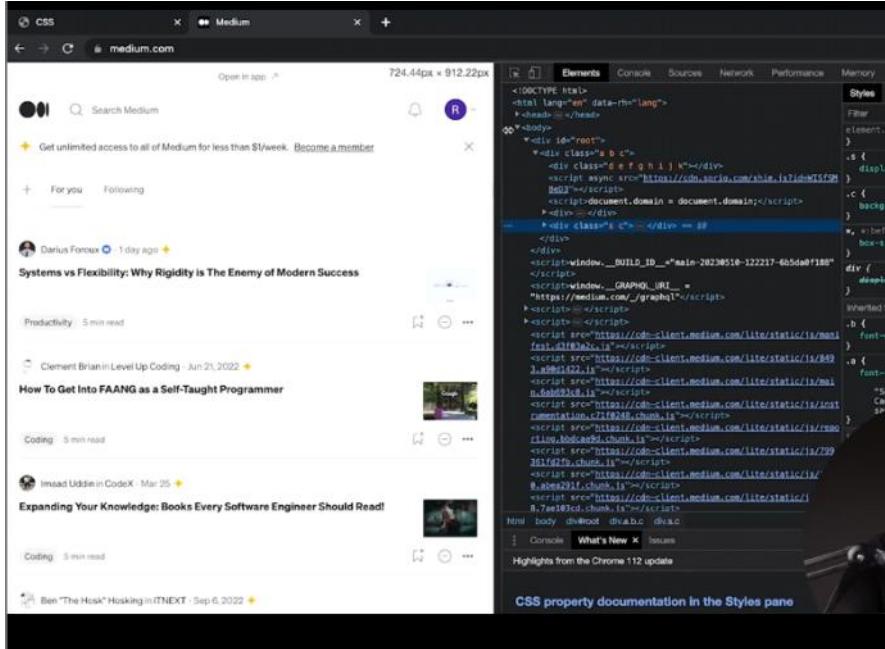
Class: 12 Media Queries



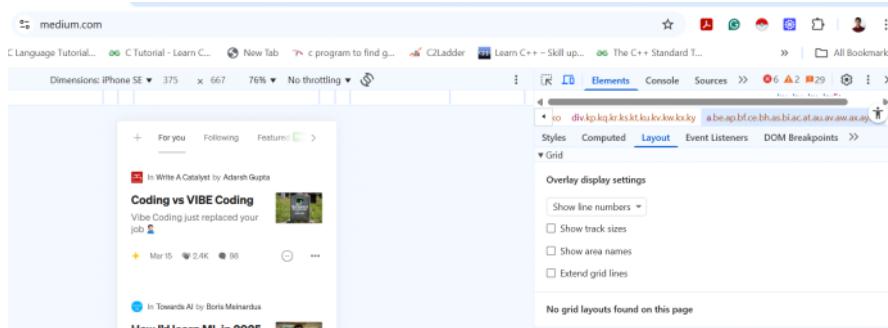
Iphone, samsung, window
Same phone landscape, horizontal, ve
Rtical,
Go: [open medium.com](https://open.medium.com)



When I decrease the website size then see:



Note: from inspect we can see many different type
 Dimension : where we can test diff type devices it's
 Visualization .



Here, if I click left button of the element,
 we can see dimension and can check many type form
 For diff devices.

667

Responsive

- ✓ iPhone SE
- iPhone XR
- iPhone 12 Pro
- iPhone 14 Pro Max
- Pixel 7
- Samsung Galaxy S8+
- Samsung Galaxy S20 Ultra
- iPad Mini
- iPad Air
- iPad Pro
- Surface Pro 7
- Surface Duo
- Galaxy Z Fold 5
- Asus Zenbook Fold
- Samsung Galaxy A51/71
- Nest Hub
- Nest Hub Max

Edit...

Dimensions: Samsung Galaxy A51/71 ▾ 412 × 914 50% ▾

+ For you Following Featured **New** >

In Write A Catalyst by Adarsh Gupta

Coding vs VIBE Coding

Vibe Coding just replaced your job

Media Queries

media features - width (of viewport)

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

Condition:

If width : 400px

Then b-g color will be : red;

Seach:

Media feature MDN

The screenshot shows a dark-themed browser window displaying the Mozilla Developer Network (MDN) documentation for CSS. The URL in the address bar is developer.mozilla.org/en-US/docs/Web/CSS/@media. The page title is "CSS" and the sub-section title is "@media". The left sidebar contains navigation links for CSS, References, Guides, Plus, and Blog. The main content area is titled "References > CSS > @media". A sidebar on the right lists various media features with their descriptions. The "device-width" feature is highlighted with a blue background and white text. Other listed features include device-aspect-ratio, device-height, display-mode, dynamic-range, forced-colors, and others like font-feature-values, font-palette-values, import, keyframes, layer, namespace, page, property, supports, functions, types, guides, animations, and backgrounds-and-borders.

- device-aspect-ratio
- device-height
- device-width
- display-mode
- dynamic-range
- forced-colors

device / environment

- ↳ width
- ↳ height
- ↳ orientation → landscape portrait

=====

Condition

```
index.html    # style.css  X

# style.css > {} @media (width: 400px) > h1
1   h1 {           }
2     background-color: yellow;
3     border: 2px solid black;
4   }
5
6   @media (width: 400px) {
7     h1 {
8       background-color: red;
9       color: white;
10    }
11  }
```

o/p:

When 393.33 px width

The screenshot shows a browser's developer tools with the CSS panel open. The title bar says "393.33px x 912.22px". The main area displays the HTML structure of the page:

```
<!DOCTYPE html>
...<html lang="en"> = $0
  ><head> = </head>
  ><body>
    <h1>Media Queries Demo</h1>
  </body>
</html>
```

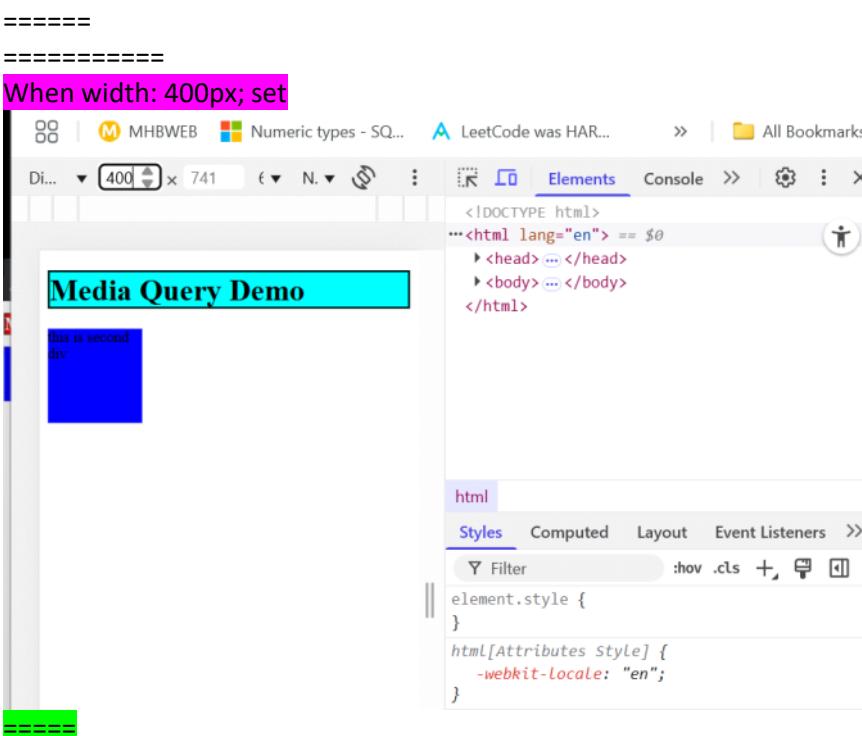
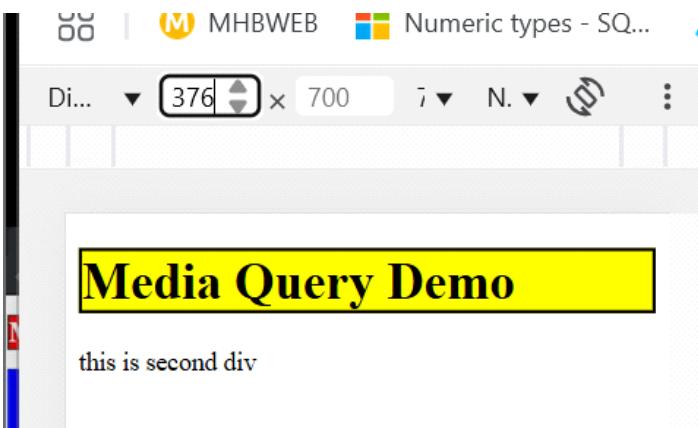
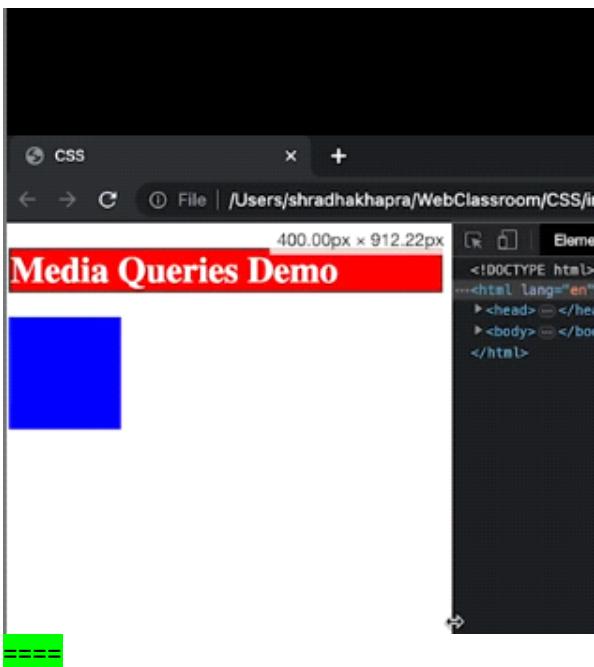
When 400px width : then b-g color: red;
And color: white;



```
=====
8  /* if width == 400px , then it's color will be aqua */
0 @media (width:400px){
1   h1{
2     background-color: aqua;
3   }
4
5   div {
6     height: 100px;
7     width: 100px;
8     background-color: blue;
9   }
0
1
2 }
```

```
8 </head>
9 <body>
10
11 <h1>Media Query Demo</h1>
12
13
14
15 <div>this is second div</div>
16
17 </body>
18 </html>
```

O/P:



Media Queries

width (of viewport)

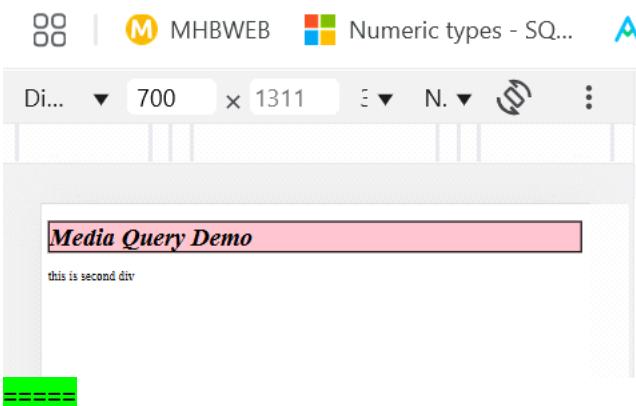
```
@media (min-width : 400px) {  
    div {  
        background-color : red;  
    }  
}  
  
@media (max-width : 400px) {  
    div {  
        background-color : red;  
    }  
}
```

here:

Width: 0 to 400px color: red;

If width: 400 px to more ;
Color: red;

=====



A screenshot of a code editor window titled "Apna col dev course code". The active tab is "# style12_media_query.css". The code displays several media query rules for an H1 element:

```
... to11_Animation_in_css.html    class_12_Media_Query.html    # style12_media_query.css    ...  
...  
7  /* media query condition */  
8  
9  /* if width == 400px , then it's color will be aqua */  
10 @media (width:400px){  
11  h1{  
12      background-color: aqua;  
13  }  
14  
15  div {  
16      height: 100px;  
17      width: 100px;  
18      background-color: blue;  
19  }  
20  
21  }  
22 /* min-width: 700px; */  
23  
24 @media (min-width: 700px) {  
25  h1 {  
26      background-color: greenyellow;  
27      font-style: italic;  
28  }  
29  
30 /* */  
31 @media (max-width: 700px){  
32  h1 {  
33      background-color: pink;  
34  }  
35  
36 }  
37  
38 }  
39
```



A screenshot of a browser developer tools interface. The left panel shows the DOM tree for a page titled "Media Query Demo". The right panel shows the "Elements" tab with the CSS styles for the H1 element:

```
Media Query Demo  
this is second div  
...  
h1 {  
  background-color: pink;  
}  
/* for setting an range */  
@media (min-width:200px) and ( max-width:300px){  
  h1 {  
    background-color: yellow;  
  }  
}
```



Class : 13 Orientation

Media Queries

orientation (of viewport)

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

Two type orientation :

1. landscape

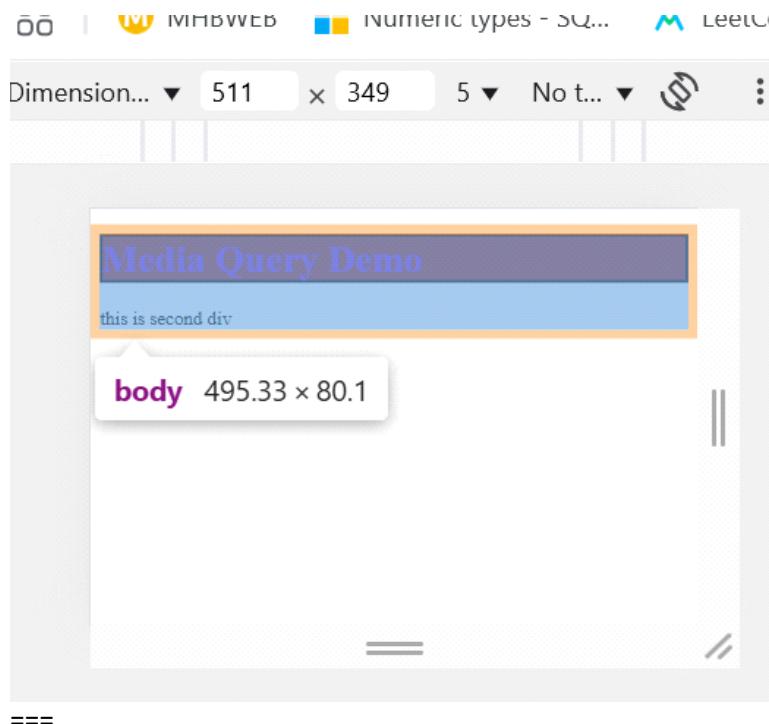
2.



Here I can rotate , and can see landscape mode.

```
50
51 /* class: 13 Media if landscape then color: aqua */
52
53 @media (orientation: landscape) {
54   h1{
55     background-color: brown;
56     color: blueviolet
57   }
58
59 }
60
```

When landscape mode:



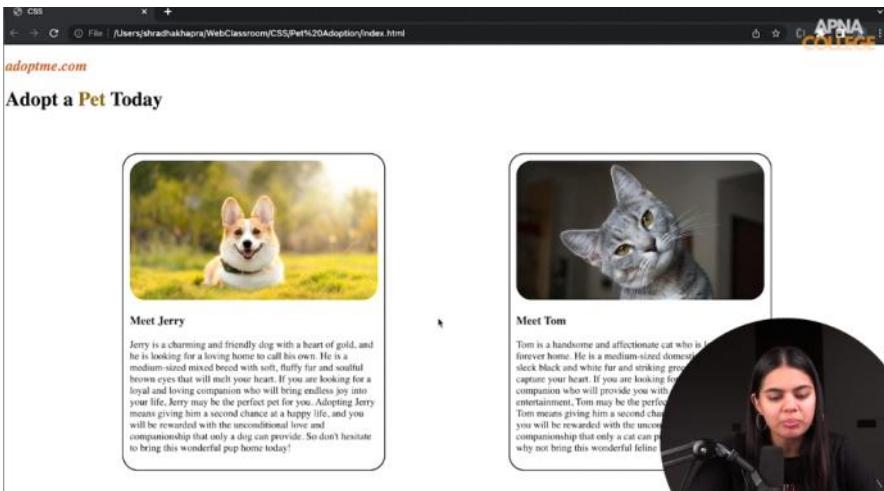
...  LeetCode



This button is help to convert landscape ,

=====

Class: 14: Pet Adoption Page
(do this class again)



==

```

tyle12_media_query.css   class_14_pet_adoption_page.html x  jerry.png  tom.png  # style_14_pet_adoption.css  # style_9_anim  ...
day_10css_part_6.cpp > class_14_pet_adoption_page.html > i.css  jerry.png  tom.png  # style_14_pet_adoption.css  # style_9_anim  ...
1 1<!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>CSS : create Pet Adoption Page</title>
7   <link rel="stylesheet" href="style_14_pet_adoption.css">
8 </head>
9 <body>
10 <h2><i>adoptme.com</i></h2>
11 <h1>Adopt a <span>Pet </span> Today</h1>
12 <br><br>
13 <div id="pets">
14   <div class="card">
15     
16     <h3>Meet Tom</h3>
17     <p>
18       Tom is a charming and friendly dog,
19       it always looking for a loving home to call his own. He is ...lorem .
20       Lorem, ipsum dolor sit amet consectetur adipisicing elit. Deserunt cum quos fuga
21       dolores enim, sit beatae molestiae officia. Itaque vitae deserunt vel libero
22       repellendus ipsam tempora temporibus sapiente delectus officia.
23     </p>
24   </div>
25   <div class="card">
26     
27     <h3>Meet Jerry </h3>
28     <p>
29       Jerry is a most wonderful rat.
30       Lorem ipsum dolor sit amet consectetur adipisicing elit. Possimus reiciendis
31       velit numquam obcaecati! Iusto hic fugiat quibusdam voluptate obcaecati eligendi
32       eum accusantium itaque veniam, est debitis iste magnam labore eos.
33   </div>
34 </div>
35
36
37
38
39 </p>

```

```
repellendus ipsam tempora temporibus sapiente delectus officia.  
29      </p>  
30  </div>/.card  
31  
32  <div class="card">  
33      
34    <h3>Meet Jerry </h3>  
35  
36    <p> Jerry is a most wondering rat.  
37      Lorem ipsum dolor sit amet consectetur adipisicing elit. Possimus reiciendis  
38      velit numquam obcaecati! Iusto hic fugiat quibusdam voluptate obcaecati eligendi  
39      eum accusantium itaque veniam, est debitis iste magnam labore eos.  
40    </p>  
41  </div>/.card  
42  
43  </div>/#pets  
44  
45  <h4> a gift of new life</h4>  
46  
47  
48  
49 </body>  
50 </html>  
51
```

====

o/p:



Meet Tom

Tom is a charming and friendly dog, it always looking for a loving home to call his own. He is ...lorem . Lorem, ipsum dolor sit amet consectetur enim, sit beatae molestiae officia. Itaque vitae deserunt vel libero repellendus ipsam tempora temporibus sapiente delectus officia.



Meet Jerry

Jerry is a most wondering rat. Lorem ipsum dolor sit amet consectetur adipisicing elit. Possimus reiciendis velit numquam obcaecati! Iusto h eum accusantium itaque veniam, est debitis iste magnam labore eos.

```
day_10css_part_0.cpp > style_14_pet_adoption.css > ...
1
2 .card {
3     width:400px;
4     border:2px solid black;
5     border-radius:25px;
6     padding: 10px;
7     margin:10px;
8 }
9
10 img {
11     width:100%;
12     border-radius: 25px;
13 }
14
15 h2 {
16     color: bisque
17 }
18
19 span {
20     color: blueviolet
21 }
22
23
24 h4 {
25     text-align:right;
26     color: aqua
27 }
28
29
30 #pets {
31     display: flex;
32     justify-content: space-evenly;
33 }
34
```

```
<!DOCTYPE html>
...<html lang="en"> == $0
  > <head> ... </head>
  > <body> ...
</html>
```

html

Styles Computed Layout >

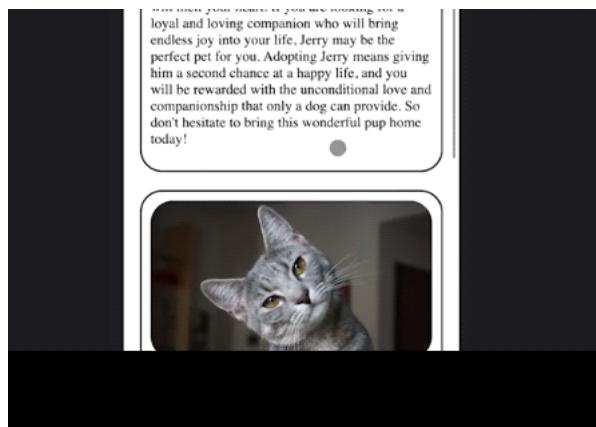
element.style {
}

```
html[Attributes style] {  
  -webkit-locale: "en";  
}  
:root {  
  user agent styles  
  view-transition-name: root;  
}  
html {  
  user agent styles  
  display: block;  
}
```

margin -
border -
padding -

##important
From inspect we see:

When width: 375 px;
Then . Two card come one after another



==
Change: _____
We havve to right:

```

27 }
28
29
30 ∵ #pets {
31   display: flex;
32   justify-content: space-evenly;
33 }
34
35
36 ∵ @media (max-width:375px){      /* 375 px for iphone ES */
37 ∵   #pet {
38     flex-wrap:wrap;
39   }
40 }
41
42 ∵ /* wrap: if we have more element then it take
43   into next line. */
44

```

Right



Class : 15 pdf

Download Starter HTML

Compulsory & Important

Please use this Github link to download the starting HTML file(s) that you will need to write code in this section.

Please download files before watching the lecture.

For pet Adoption Activity of CSS Part 6, there are few files you need to download.

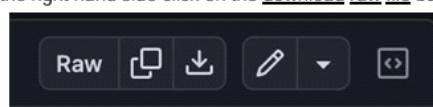
Prerequisites - If you don't have a Github account, Please sign up on <https://github.com/> using any email address. And keep the account logged in.

Please download all the HTML files in this folder :

[https://github.com/apna-college/Delta/tree/main/CSS/Chapter6%20\(Pet%20Adoption\)/Pet%20Adoption](https://github.com/apna-college/Delta/tree/main/CSS/Chapter6%20(Pet%20Adoption)/Pet%20Adoption)

To download each file (as mentioned in the chapter) :

- Click on file name
- On the right hand side click on the download raw file button.



====

Class : 16

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 / ...



When element are overlap one another.
Like one book is put upon another book.

Here, 1 , 3 book 2 are same lavel.
Book 2 are 2nd level .

****like stack****

z-index

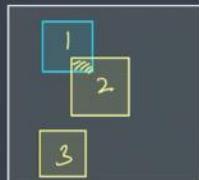
It decides the **stack level** of elements

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 / ...



APNA
COLLEGE

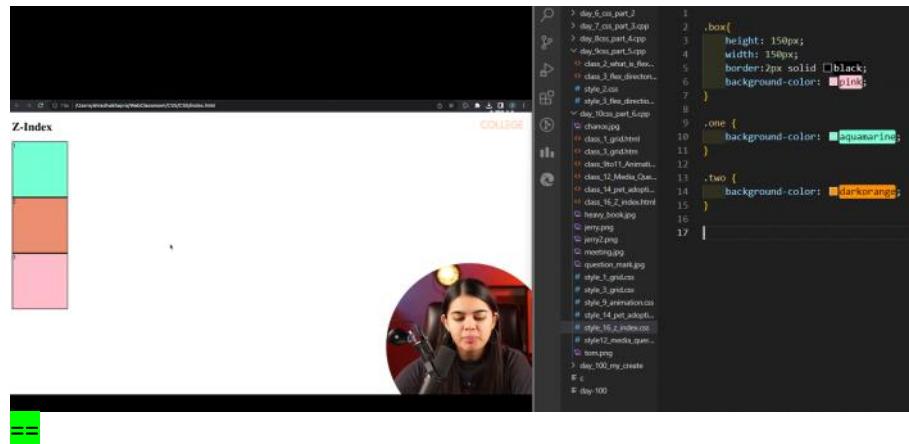
CSS

==

```

# style_14_pet_adoption.css      ◊ class_16_Z_index.html × # style_16_z_index.css
\10css_part_6.cpp > ◊ class_16_Z_index.html > html > body > div.box.three
1
2  <!DOCTYPE html>
3  <html lang="en">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width,
initial-scale=1.0">
7      <title>Z-index </title>
8      <link rel="stylesheet" href="style_16_z_index.css">
9  </head>
10 <body>
11
12     <h1><u>Z-Index</u></h1>
13     <div class = "box one"> 1</div>
14     |<div class = "box two"> 2</div>
15     |<div class = "box three"> 3</div>
16
17 </body>
18 </html>

```

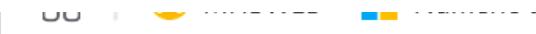


When add this:

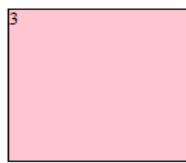
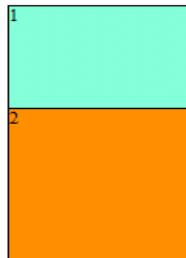
```

2
3  .two {
4      background-color: #darkorange;
5
6      /* add below element */
7
8      position: relative;
9      bottom:50px;
0  }
1
o/p:

```

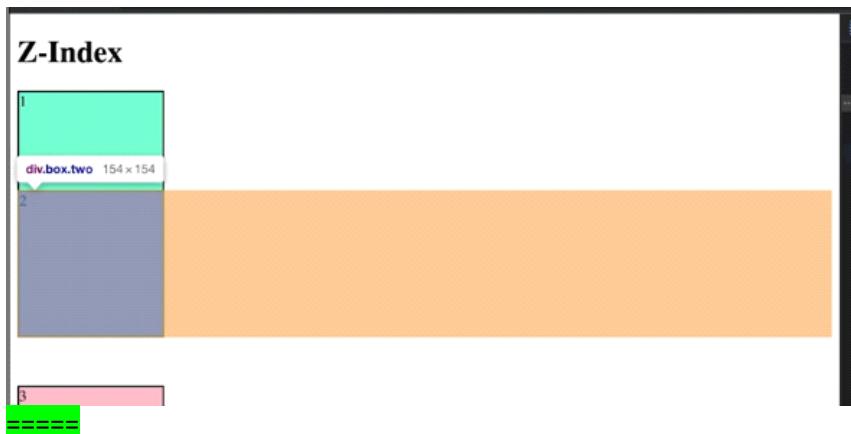


Z-Index



==== here overlap the orange box.

Here orange box upper side of stack
And green box down of the orange box.



Why we use Z-index property?

Ans: when we want to show some element come to level 2 of stack

And some level come stack level 1 and

Some level come stack level 3 . Then we use Z-index property.

====

Question : if two element are overlap which element will be up or level2
Of stack ?

Answer: which Z-index => value is greater setting , this element
Will be upper side and overlap to the level 1.

z-index

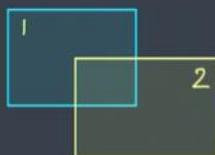
It decides the **stack level** of elements

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 / ...



Here,

By default:

Every element has `z-index: auto(0);` so all are same level.

And

Question: for which element , we can set Z-index property ?

Ans:

If position is => static , default, then can't set z-index property.

When ,

We give *****`position : relative ;`*****

*****then can set z-index property.

Note:*****

which element come upper level ?

Answer: which element z-index value is big.*****

=====

EXAMPLE:

z-index : auto (0)

(+ve) *z-index : 1 / 2 / ... upper*

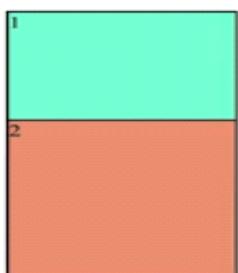
(-ve) *z-index : -1 / -2 / ... lower*

```
.one {  
    background-color: aquamarine;  
    position: relative;  
    z-index: -1;  
}  
                                ↗ auto
```

o/p:

← → C File | /Users/shradhakhapra/WebClassroom/

Z-Index



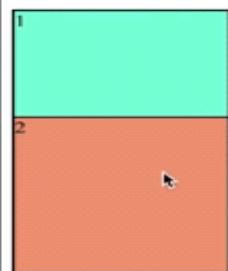
====

```
.one {  
    background-color: aquamarine;  
    position: relative;  
    z-index: 1;  
}
```

```
.two {  
    background-color: darksalmon;  
    position: relative;  
    bottom: 50px;  
    z-index: 2;  
}
```

o/P:

Z-Index



```
7
8   .one {
9     background-color: aquamarine;
10    position: relative;
11    z-index: 1;
12  }
13
14  .two {
15    background-color: darksalmon;
16    position: relative;
17    bottom: 50px;
18    z-index: 0;
19  }
```

=====

Question : if set both z-index same value , what will be ?

Answer:

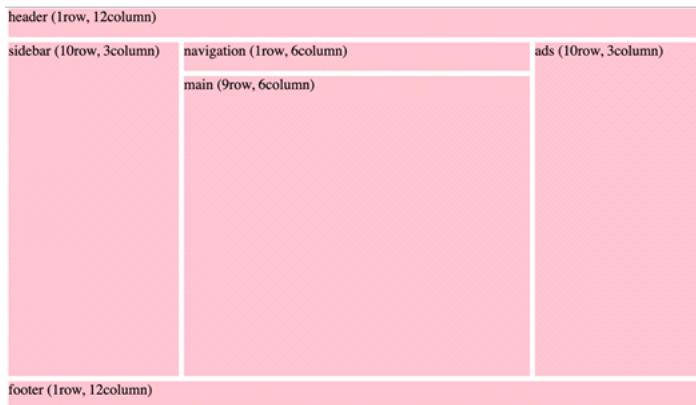
=====

Pdf question

CSS (Part 6)

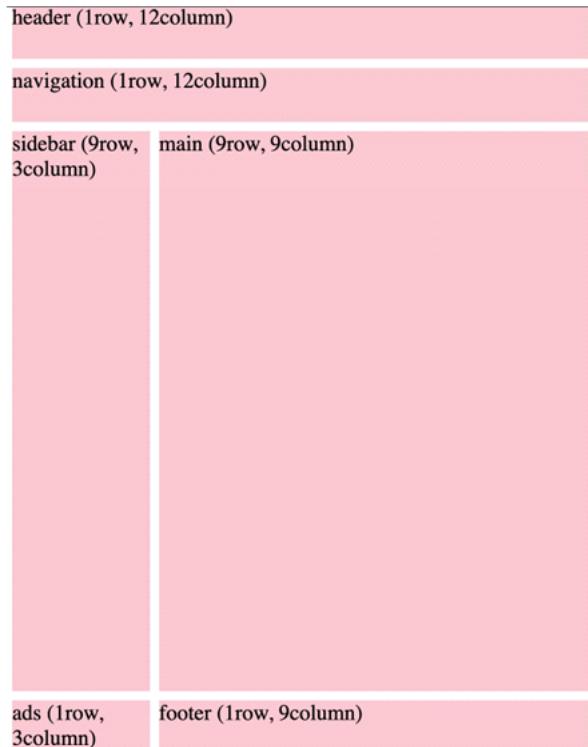
Practice Questions

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.



```
<!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>
```



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

Loader



HTML 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 Code :

```
=====
```

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>
```

CSSCode

```
.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

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

HTML Code will remain the same as answer 1.

CSS Code

```
.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);  
}  
}
```

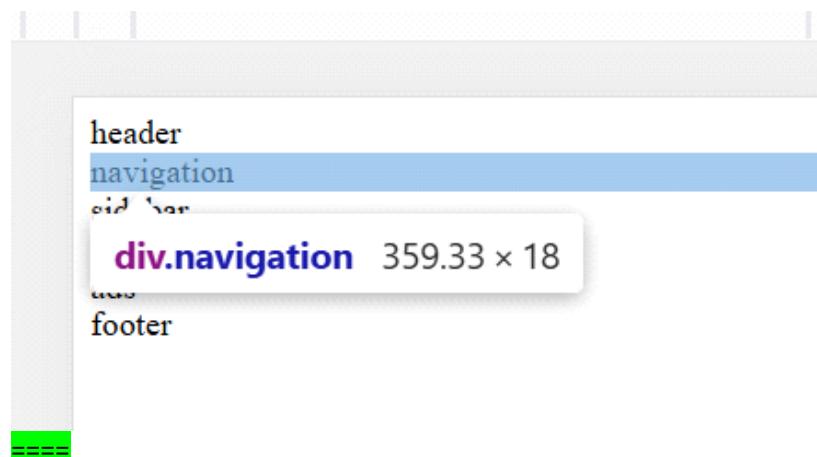
=====

Solve the practice question pdf 17 and 18

Answer : 1

```
1  <!-- question 1: create following layout using css grid: -->
2
3
4  <!DOCTYPE html>
5  <html lang="en">
6  <head>
7      <meta charset="UTF-8">
8      <meta name="viewport" content="width=device-width,
9          initial-scale=1.0">
9      <title>css: grid learning</title>
10
11     <link rel="stylesheet" href="style_17_18_question_solve.
12         css">
12
13 </head>
14 <body>
15
15     <div class="container">
16
17         <div class="header">header</div>
18         <div class="navigation"> navigation</div>
19         <div class="sidebar"> sidebar</div>
20
21         <div class="main"> main</div>
22         <div class="ads"> ads</div>
23
24         <div class="footer">footer</div>
25
26
27     </div>/.container
28 </body>
29 </html>
```

o/p:



When create jjsut container

dimension... ▾ 375 x 667 7. ▾ No t... ▾

Elements

```
<!-- question 1: create following layout using css grid: -->
<!DOCTYPE html>
<html lang="en"> scroll
  <head> ... </head>
  <body>
    <div class="container"> grid
      <div class="header">header
      </div>
      <div class="navigation"> navigation</div> == $0
      <div class="sidebar"> sidebar
      </div>
      <div class="main"> main</div>
      <div class="ads"> ads</div>
      <div class="footer">footer
      </div>
    </div>
  </body>
<!-- Code injected by live-server -->
<script> ... </script>
</html>
```

div.container 359.33 x 667.33

header
navigation
sidebar
main
ads
footer

=====

header

div.sidebar 359.33 x 46.44

sidebar

main

ads

footer

=====

header
navigation
sidebar
main
ads
footer

Body

header
navigation
sidebar
main
ads
footer

layout using css grid: -->

```
<!DOCTYPE html>
<html lang="en"> scroll
  <head> ... </head>
  <body>
    <div class="container"> grid
      <div class="header">header
      </div>
      <div class="navigation"> navigation</div> == $0
      <div class="sidebar"> sidebar
      </div>
      <div class="main"> main</div>
      <div class="ads"> ads</div>
      <div class="footer">footer
      </div>
    </div>
  </body>
<!-- Code injected by live-server -->
<script> ... </script>
</html>
```

====

When add:

```
21
22 .container div {
23
24     background-color: pink;
25     font-size: 1.5rem;
26 }
```



```
<!-- question 1: create following layout using css grid: -->
<!DOCTYPE html>
<html lang="en">
  <head> ... </head>
  <body>
    <div class="container">grid
      <div class="header">header
      </div>
      ...
      <div class="navigation">navigation</div> == $0
      <div class="sidebar"> sidebar
      </div>
      <div class="main"> main</div>
      <div class="ads"> ads</div>
      <div class="footer">footer
      </div>
    </div>
    <!-- Code injected by live-server -->
    <script> ... </script>
  </body>
</html>
```

html body div.container div.navigation

When add

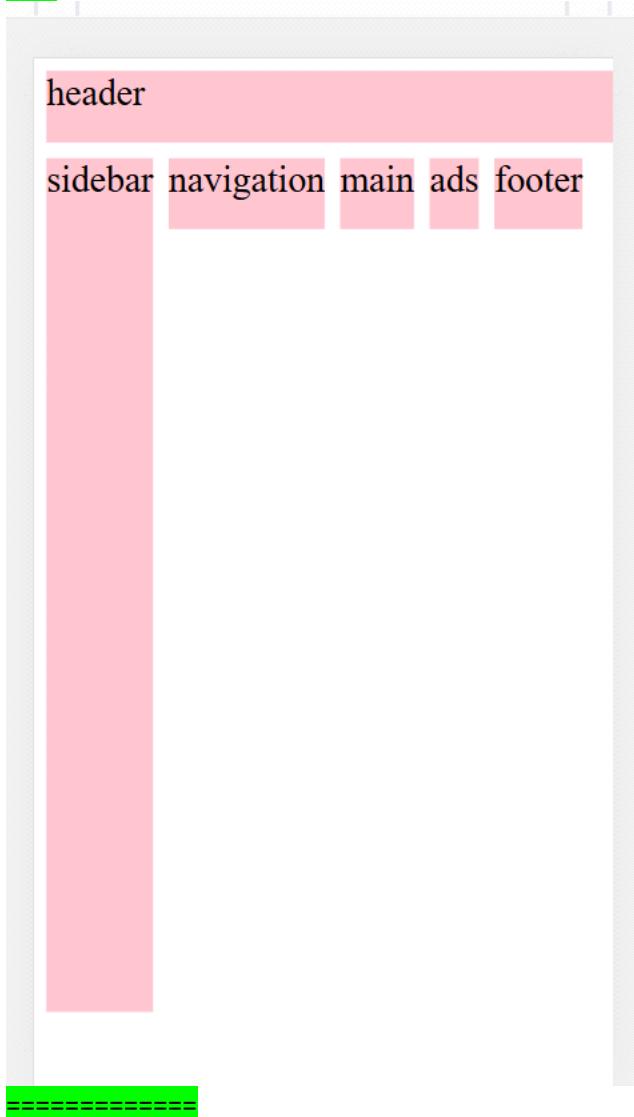
```
.header {
  grid-column: 1 /span 12;
}
```

o/p

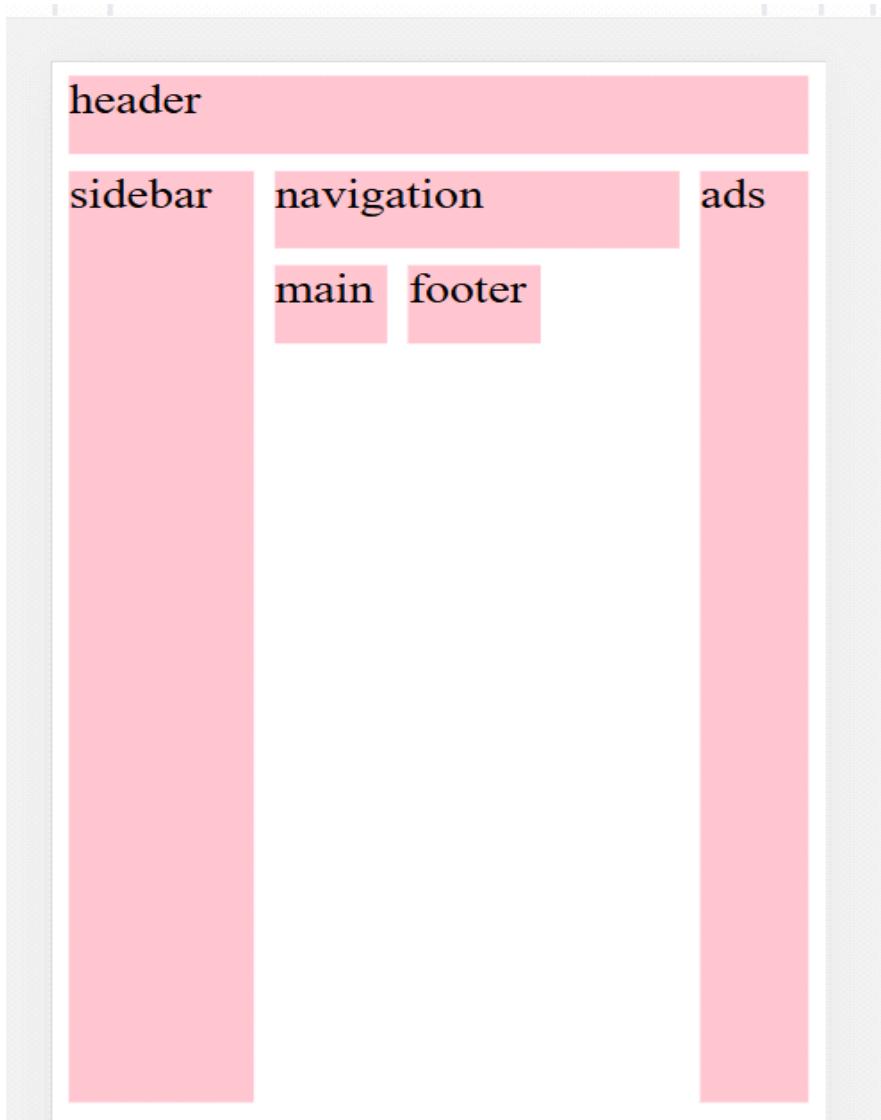
When add"

```
33  
34 .sidebar {  
35   grid-column: 1 /span 3;  
36   grid-row: 2 /span 10;  
37 }
```

o/p:



```
43
44 .ads {
45   grid-column: 10 /span 3;
46   grid-row: 2/span 10;
47 }
```

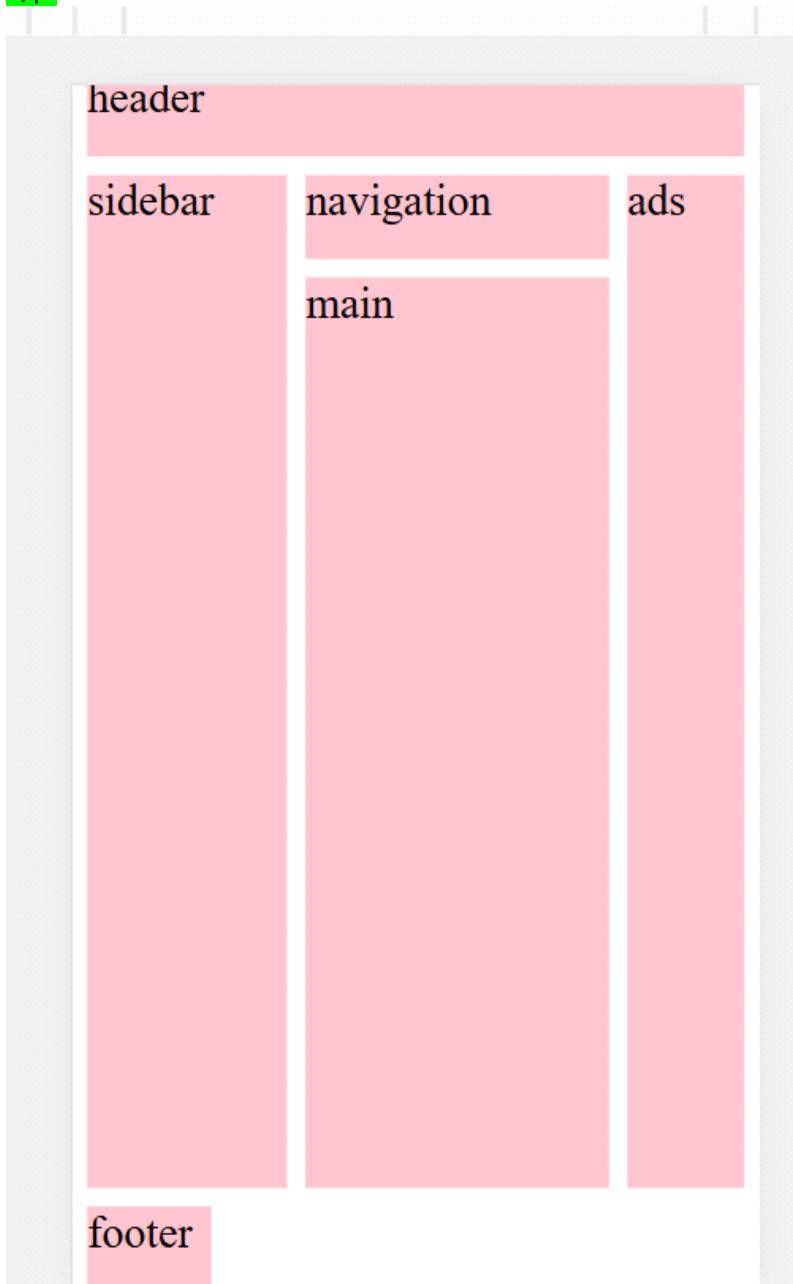


====

When add main:

```
8
9   .main {
0     grid-column: 4 /span 6;
1     grid-row: 3 / span 9;
2 }
```

o/p



====

```
<!-- question 1: create following layout using css grid: -->   
<!DOCTYPE html>   
<html lang="en">   
  <head> ... </head>  
  <body>  
    <div class="container">   
      <div class="header">header</div>  
    </div>  
    ...  
    <div class="navigation">  
      navigation</div> == $0  
      <div class="sidebar"> sidebar</div>  
      <div class="main"> main</div>  
      <div class="ads"> ads</div>  
      <div class="footer">footer</div>  
    </div>  
    <!-- Code injected by live-server -->  
    <script>...</script>  
  </body>  
</html>
```

html body div.container div.navigation

Styles Computed Layout >>

Filter :hov .cls +, ☰

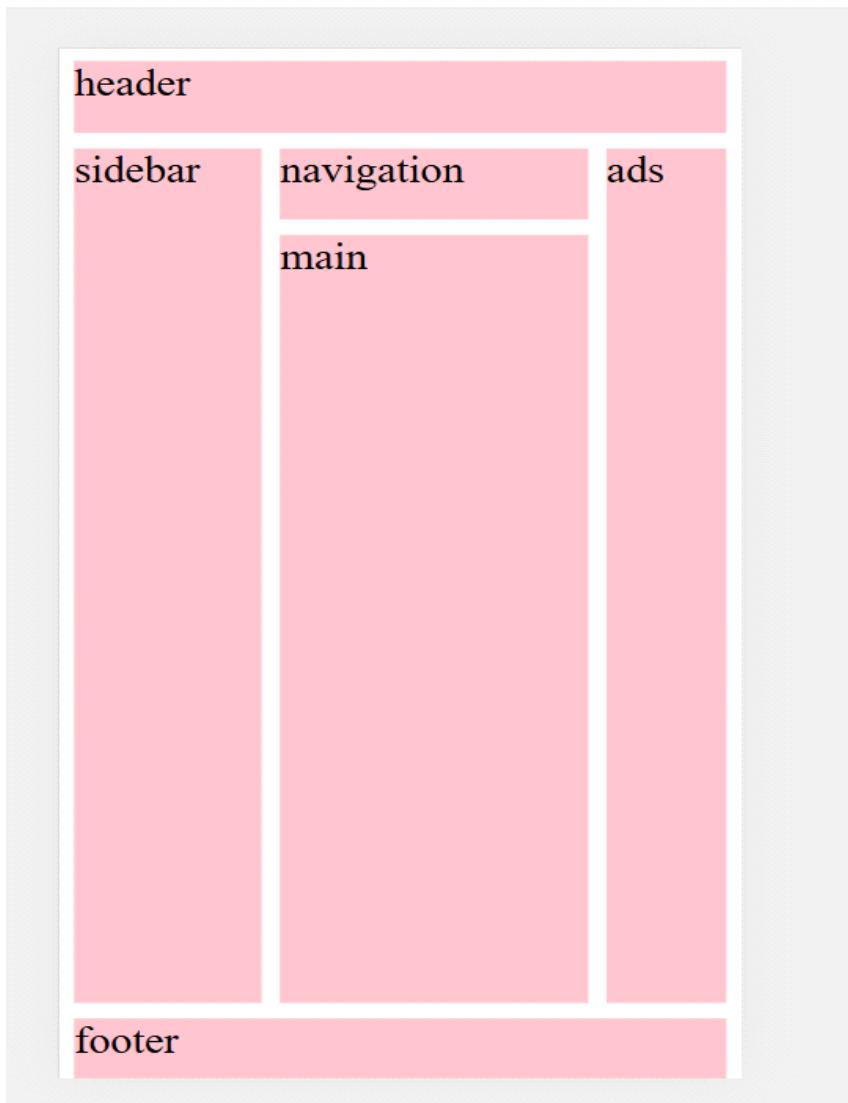
```
54 .footer {  
55   grid-column:1 /span 12;  
56 }
```

o/P,
footer take full 12 column width .



=====

Final answer:



=====

Question 2 answer:

=====

Question: 3

```
... solve.css # style_18_ans2.css class_18_quest_3ans.html # style_18_ans_3.css ...
... day_10css_part_6.cpp > class_18_quest_3ans.html > body > div.loader
cpp 1 <!DOCTYPE html>
2 <html lang="en">
ectio... 3 <head>
5 <meta charset="UTF-8">
6 <meta name="viewport" content="width=device-width,
initial-scale=1.0">
7 <title>create a loader</title>
8 <link rel="stylesheet" href="style_18_ans_3.css">
n </head>
9 <body>
10 <h1> Loader </h1>
11 <div class="loader"> </div>
12 </body>
13 </html>
14
15
```

```
1
2
3 .loader {
4     border: 16px solid #f3f3f3;
5     /* create border around the loader div */
6
7     border-top: 16px solid goldenrod;
8     /* explain: */
9
10    border-radius: 50%;
11    width: 120px;
12    height: 120px;
13 }
```

o/p:



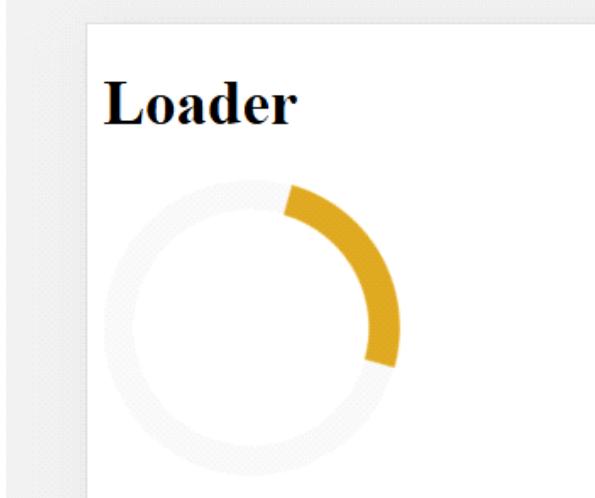
The screenshot shows a browser window displaying a simple HTML page. The page contains a single heading element `<h1> Loader </h1>` and a single `<div>` element with the class `.loader`. The `.loader` div has a `border-radius: 50%`, making it a circle. It also has a `border: 16px solid #f3f3f3` and a `border-top: 16px solid goldenrod`, creating a thick orange border at the top and a thin grey border on the sides and bottom. Inside the circle, there is a green semi-circular progress bar starting from the top-left. The browser's status bar at the bottom shows the dimensions of the body element as `body 359.33 x 210.1`.

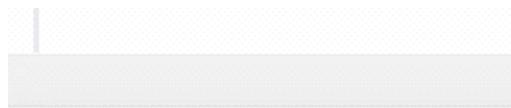
```
<!DOCTYPE html>
<html lang="en">
  <head> ... </head>
  ... <body> ... </body> == $0
</html>
```

```
2
3 .loader {
4     border: 16px solid #f3f3f3;
5     /* create border around the loader div
6
7     border-top: 16px solid goldenrod;
8     /* explain: */
9
10    border-radius: 50%;
11    width: 120px;
12    height: 120px;
13    animation: spin 2s linear infinite;
14 }
15
16
17
18
19 @keyframes spin {
20     0% {
21         /* SET rotation to 0 degree */
22         transform: rotate(0deg);
23     }
24
25     100% {
26         /* set rotation to 360 degree */
27         transform: rotate(360deg);
28     }
29 }
30
31 }
32
```

When right this :

It will spin contiunously





Loader



=====

The screenshot shows a browser window with the following details:

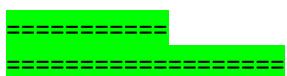
- The title bar says "div.loader 211.79 x 211.79".
- The main content area displays the word "Loader" above a blue diamond-shaped icon with a yellow border.
- The browser's developer tools are open, showing the HTML code on the right side:

```
<!DOCTYPE html>
<html lang="en">
  <head> ... </head>
  <body> == $0
    <h1> Loader </h1>
    <div class="loader"> </div>
    <!-- Code injected by live-server -->
    <script> ... </script>
  </body>
</html>
```

=====

What is this ?

Answer:



FINISH THIS DAY-10 CSS PART-6
MOST IMPORTANT COMPLICATED DAY .

====