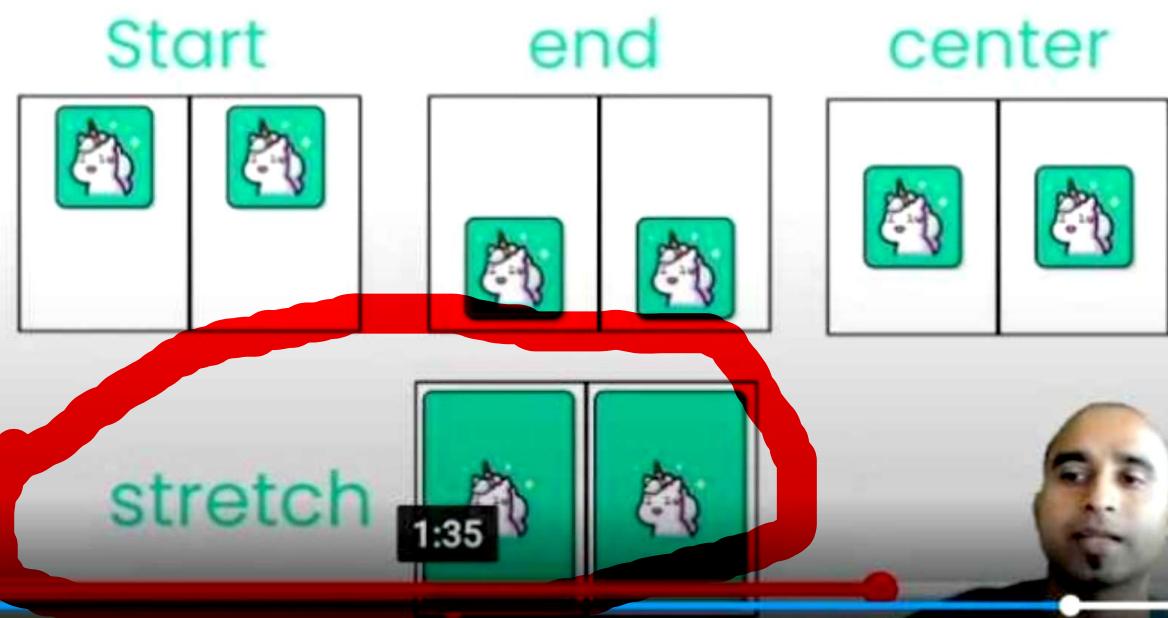


CSS Grid Batch 8.mp4

## Code Snippet (css):

```
.container {  
    display: grid; gap: 50px;  
    height: 100vh;   
    grid-template-rows: 200px 200px;  
    grid-template-columns: 200px 200px;  
    align-items: center;  
}
```

align-items -> Y Axis



## Code Snippet (css):

```
.container {  
    display: grid; gap: 50px;  
    height: 100vh;  
    grid-template-rows: 200px 200px;  
    grid-template-columns: 200px 200px;  
    justify-items : end;  
}
```

justify-items -> X Axis

Start



justify-items -> X Axis

center



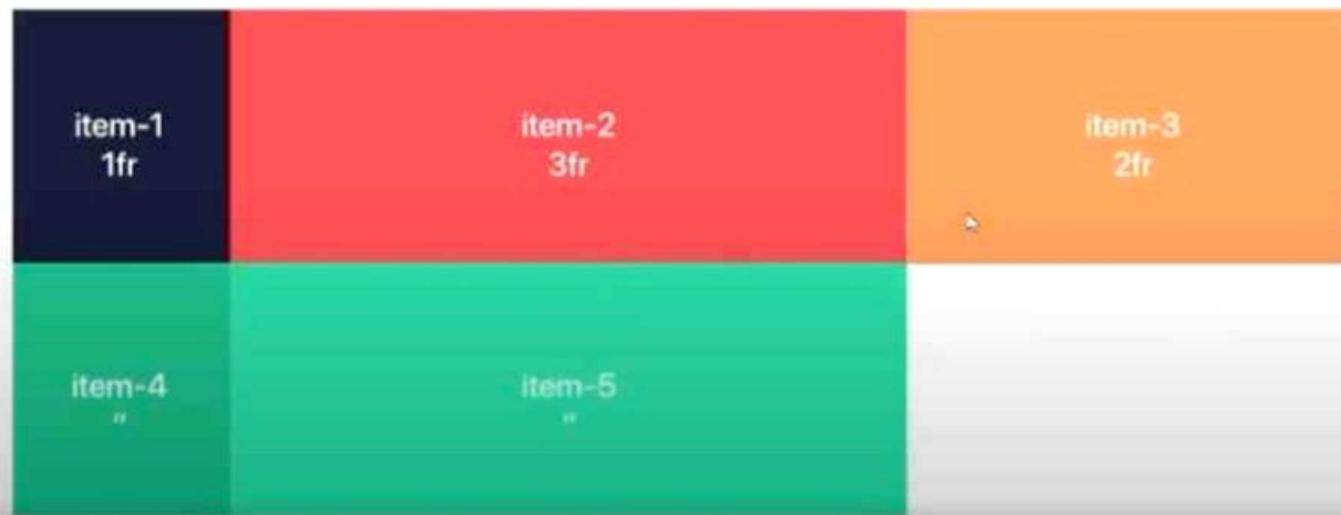
stretch



## CSS Grid Batch 8.mp4

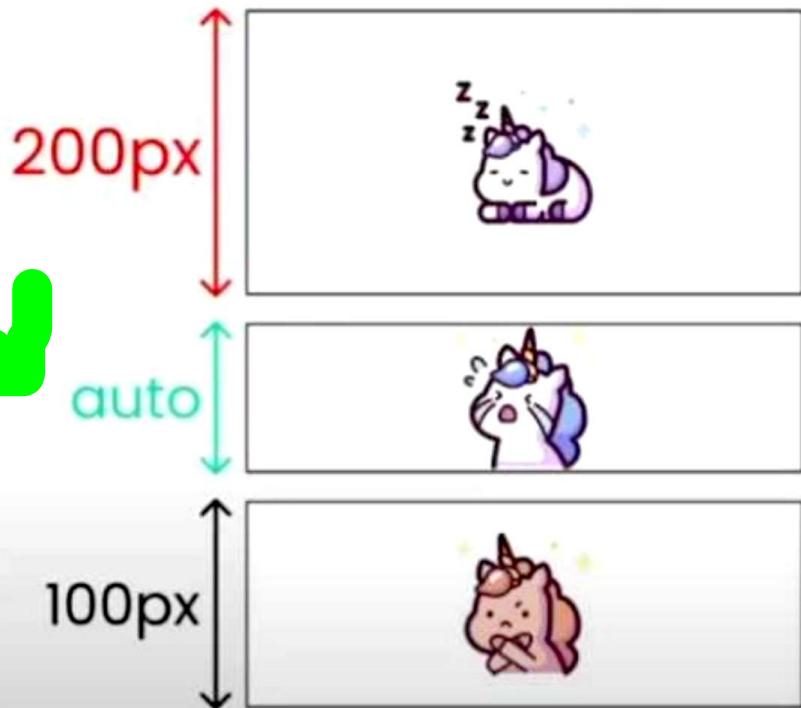
### Code Snippet (css):

```
.container{  
    display: grid;  
    grid-template-columns: 1fr 3fr 2fr;  
}
```

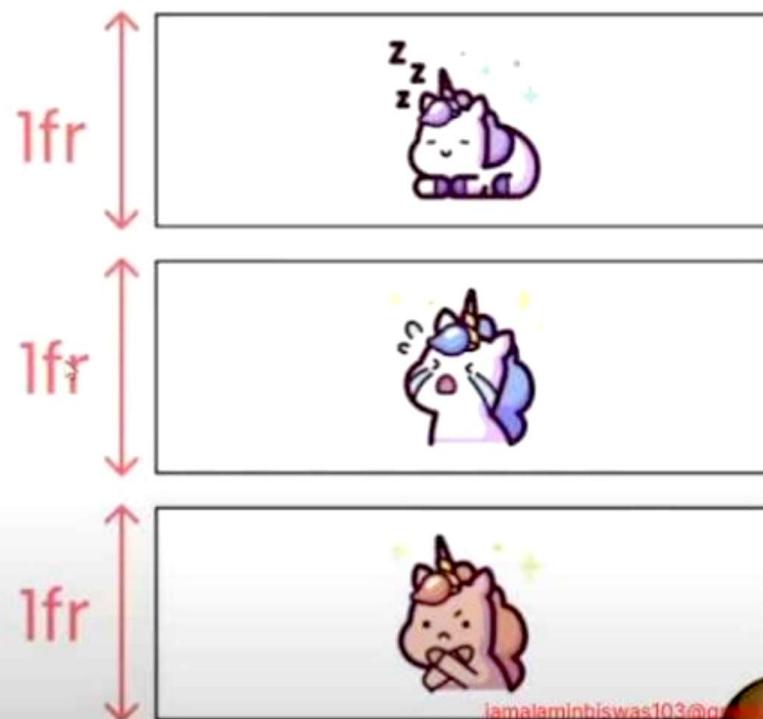


## CSS Grid Batch 8.mp4

grid-template-rows : 200px auto 100px

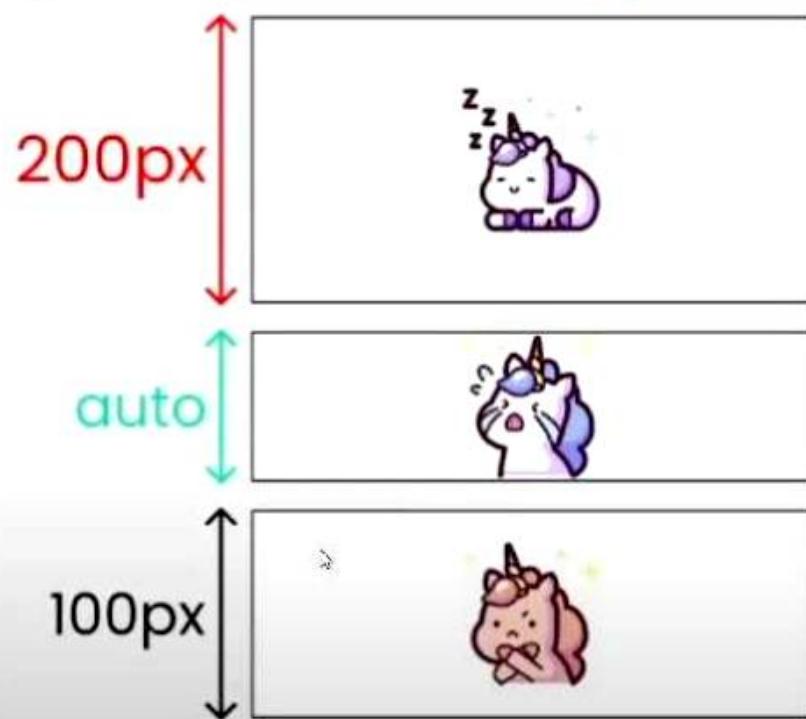


grid-template-rows : repeat(3, 1fr);



## CSS Grid Batch 8.mp4

grid-template-rows : 200px auto 100px



grid-template-rows : repeat(3, 1fr);



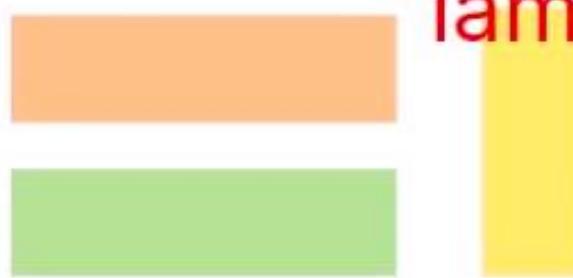
```
32 <body>
33   <header>
34     <h1>Responsive Website</h1>
35     <ol>
36       <li>Viewport meta tag</li>
37       <li>CSS Relative Unit</li>
38       <li>body max width and horizontal center align</li>
39       <li>image fluid</li>
40       <li>two column flex and use media query with flex
41         direction column</li>
42         <li>multi column: grid layout </li>
43         <li>menu responsive</li>
44     </ol>
45   </header>
46   <main>
```

```
    <div class="image-container">
```



## Grid

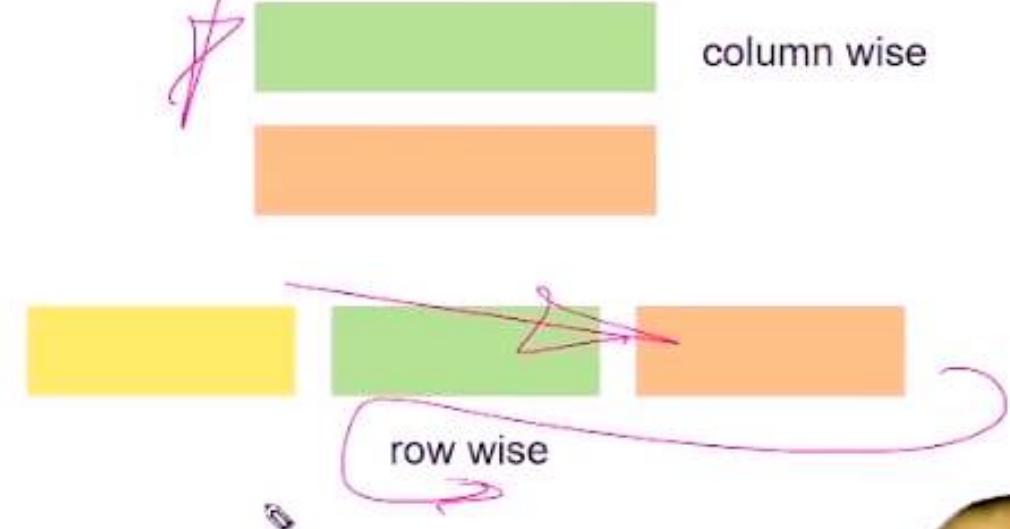
Multi dimensional layout



## Flexbox

One dimensional layout

iamalaminbiswas103@gmail.com





## Grid

Focuses on content placement  
Layout First

- Provides a mechanism to divide **available viewport** for layout into columns and rows.
- Each item can be positioned anywhere in a grid cell.

## Flexbox

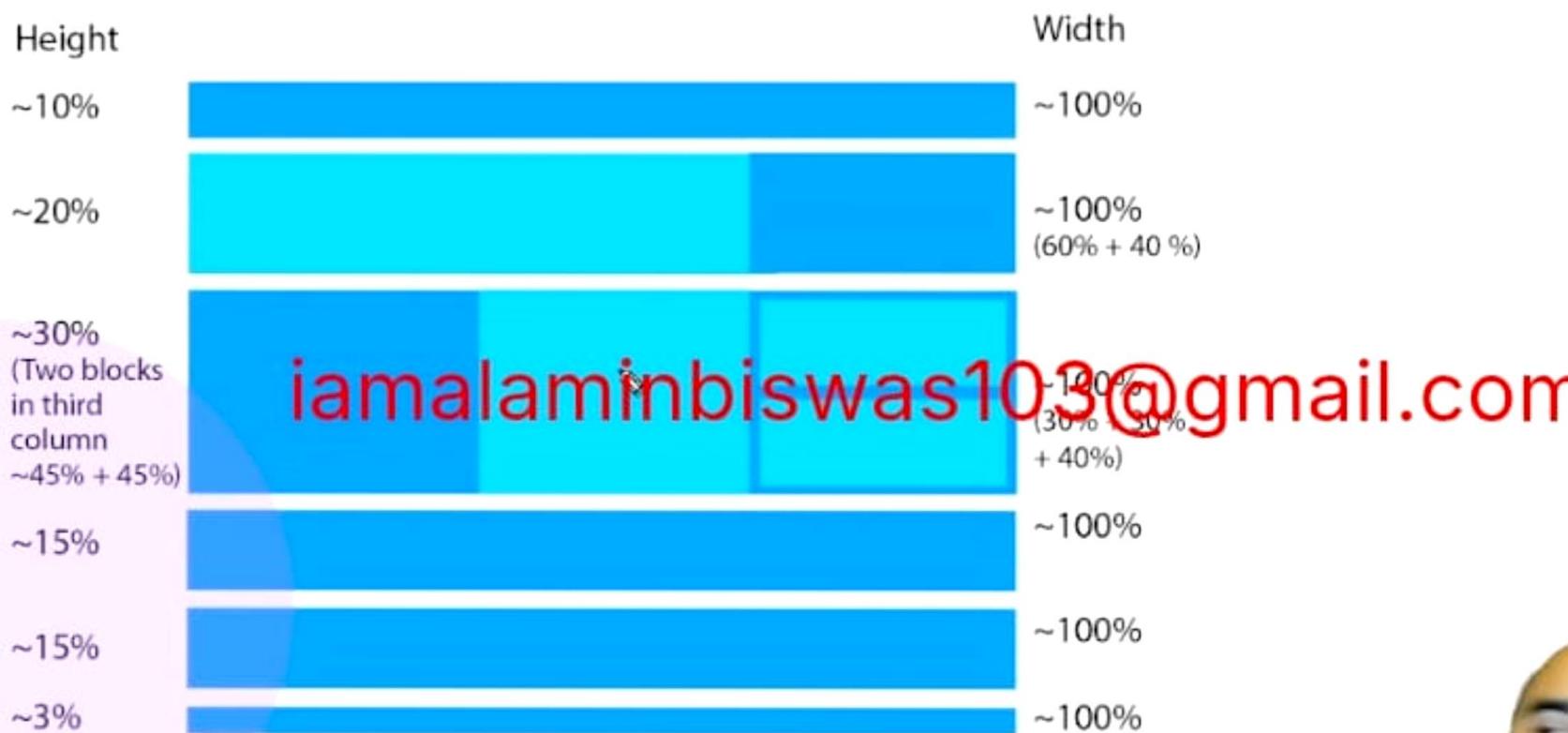
Focuses on content flow  
Content First

Widths (or heights) of flex items are determined by the content of the item.

- Spacing and aligning items are flexible.



# Complex design is easier with css grid





## When to use Flexbox

- You have a small design to implement
- You need to align elements
- You need a content-first design



iamalaminbiswas103@gmail.com



# When to use Gridiamalaminbiswas103@gmail.com

- You have a complex design to implement
- You need to have a gap between block elements
- You need to overlap elements
- You need a layout-first design



## Media Query is -

- The key part of responsive web design
- Used to change styling depending on the size of the viewport
- Used to check width and height of the viewport
- Used to check orientation (is the tablet/phone in landscape or portrait mode?)
- Used to check resolution



```
103@gmail.com
  media (orientation and breakpoint expressions) {
    CSS-Code;
  }
```



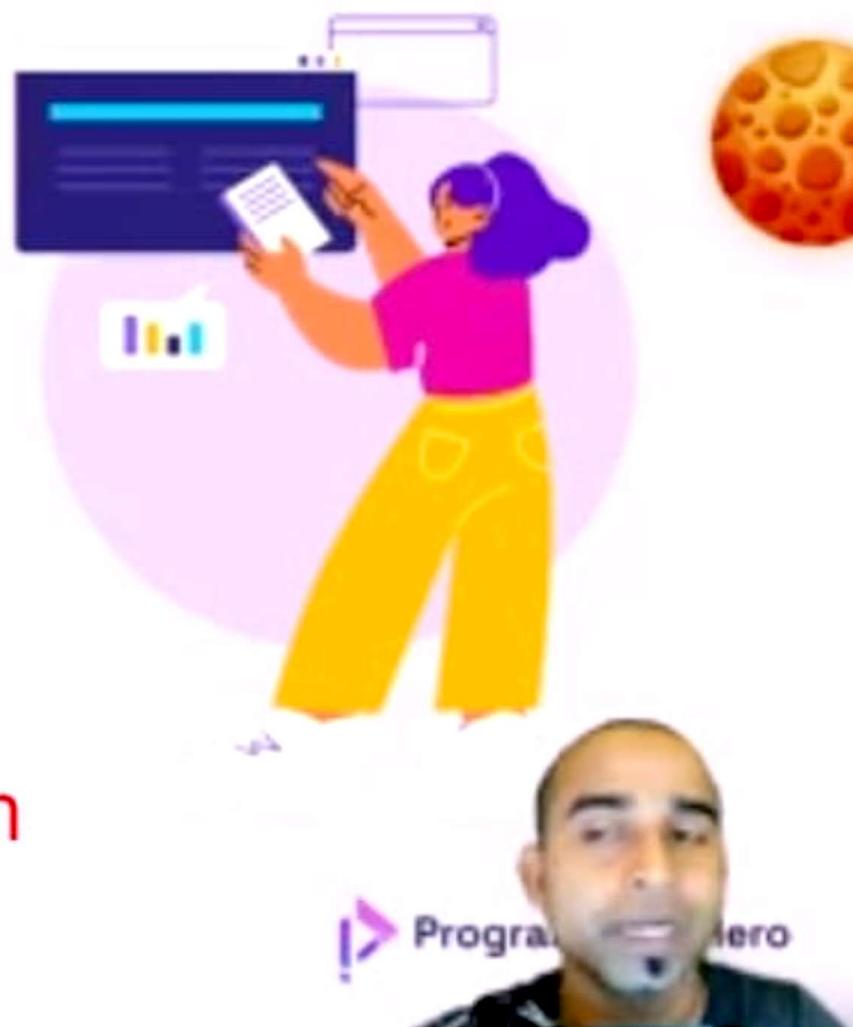
## 7 things to make website responsive -

- Use viewport Meta tag
- CSS Relative Measuring Unit
- Make Image fluid (use % as width).
- Body max width and center position

iamalaminbiswas103@gmail.com

Two column using flex layout and media

query



## Photo gallery with CSS Grid

Code Snippet (html):

```
<section class="gallery">  
  <div class="card">  
    *** *** *** ***  
  </div>  
  
  <div class="card">  
    *** *** *** *** ***  
  </div>  
  
  *** *** ***  
</section>
```

Code Snippet (css):

```
.gallery {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  gap: 10px;  
}
```

Popular Collection



Flex Tripod

\$100.40

• 100%  
• 100%  
• 100%



Microphone

\$100.20

• 100%  
• 100%  
• 100%



Smartie

\$100.00

• 100%  
• 100%  
• 100%



Chrom

\$100.00

• 100%  
• 100%  
• 100%



Light Setup

\$100.00

• 100%  
• 100%  
• 100%



Photograph set

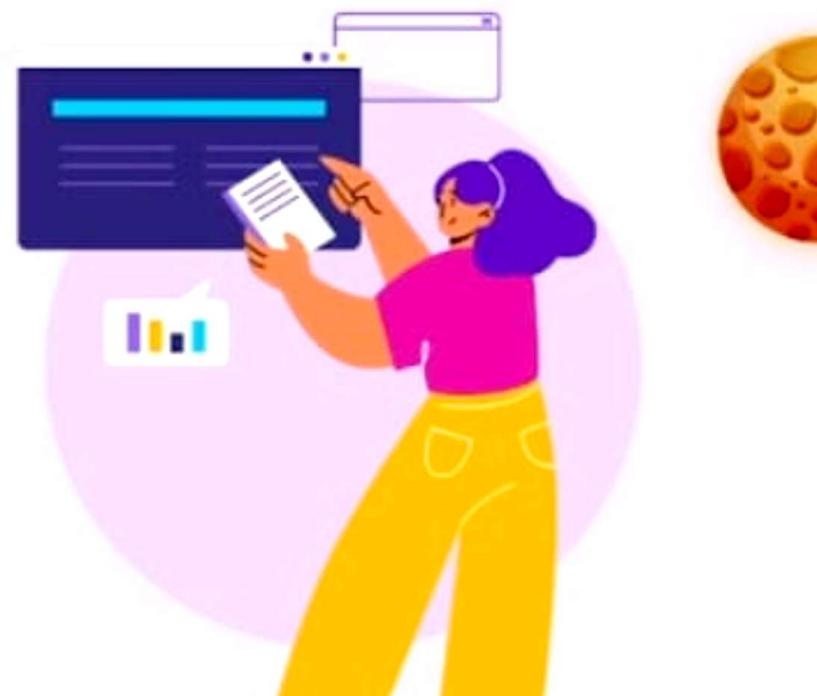
\$100.00

• 100%  
• 100%  
• 100%

iamalaminbiswas103@gmail.com

## Tips of the day -

- Focus on making images responsive.
- Implement responsive typography.
- Divide your website in small parts & make these responsive one-by-one.



iamalaminbiswas103@gmail.com



## Task 1: Design a responsive flex layout



Task 2: Design a responsive grid layout  
iamalaminbiswas103@gmail.com



### Task 3: Adapt images for different devices



iamalaminbiswas103@gmail.com



## Task 4: Design responsive layout - II



Post Title Will Be Here

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Read More



Post Title Will Be Here

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Read More

iamalaminbiswas103@gmail.com



Dimensions: Phone (2) 480x800 100% 100% Re-sizing 0%



Post Title Will Be Here

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Read More



## Task 5: Design responsive layout - I



The `grid-template-areas` property in CSS Grid Layout is used to specify areas within the grid layout by referencing names assigned to grid items using the `grid-area` property.

Therefore, the correct answer is D: To define the content of the columns and rows in the grid.

The purpose of the `order` property in Flexbox is to (B) **Specifies the order of flex items within a container.**

### Explanation:

The `order` property in CSS Flexbox allows you to control the visual order of flex items within a flex container, independent of their order in the source code. A lower `order` value will appear earlier in the layout, while a higher `order` value will appear later.

**Question: What does `flex-direction: column;` do?**

**A Aligns items in a row:**

This is the default behavior of `flex-direction: row;`.

**B Stacks items vertically:**

This is the correct answer. `flex-direction: column;` arranges flex items in a vertical column, stacking them one below the other.

**C Reverses item order:**

This is achieved by `flex-direction: row-reverse;` or `flex-direction: column-reverse;`.

**D Wraps items to new lines:**

This is controlled by the `flex-wrap` property, specifically `flex-wrap: wrap;`.

**Answer: B Stacks items vertically**

## Module 7

layout — ~~परिप्रेक्षा~~

### 7.1 — CSS measuring-unit

em — relative size (unit)

— Parent ~~से~~ — font-size change → ~~परिप्रेक्षा~~

vmax — Viewport max

vmin — Viewport min

\* Percentage — Parent & other connected

\* vw — viewport width

→ ~~जोड़ जाय वैल्य उसी- width as many~~

\* vh  
→ viewport height

→ ~~जोड़ जाय वैल्य उसी- height as many~~

vmin, vmax

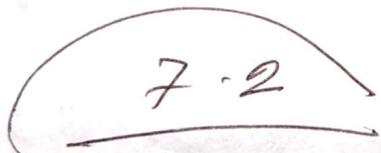
viewport at maximum profit after

Device limit about max parallel vmax

is always maximum size of the

Device

vmin at same over



flex-direction: row-reverse

**flex-wrap: wrap**

**flex-wrap: wrap-reverse**

required — input field

vh — ~~margin~~  
explore grid — ,

(7-4)  
CSS Grid Layout

grid ~~only one column available~~

grid-template-columns: ~~100px 100px 100px~~ (min

grid-template-columns: 200px 100px 10px;

justify-content: space-around 36r

grid-template-columns: repeat(2, 100px);

repeat(2, 100px) 100px

36px

grid-template-columns: repeat

(2, 1fr);

(3, 1fr);

(6, 1fr) 2fr;

(80px, 2) footer 80px: auto auto 266px 266px

now

fraction (1/8)

: 200px auto 200px

justify-content: flex-end

justify-content: flex-end

(flex-end)  
arcos

7-5.

numbers → element - border

grid → gap

column → gap

row (left, right)

30px 40px

→ grid rows columns: 30px repeat (6, 50px)

grid margin

10px

x9000 et voilà x9000:

striped parts bootstrap breakpoints

→ m1 m2 m3 m4 m5

① media query answer request (m1)

font-size: 10px  
font-size: 20px

→ m1 m2 m3 m4 m5