



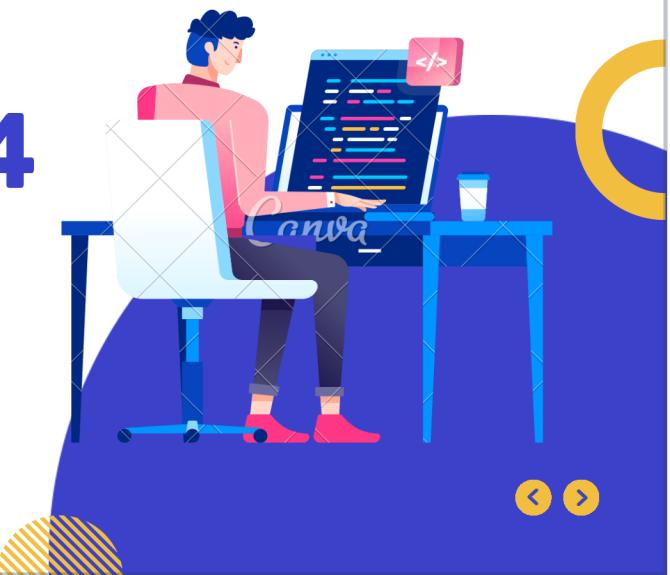
Search ...



GDG SESSION 4

BEGINNERS
TRACK

A Deep Dive into CSS





CSS Box Model (Recap)

Every HTML element is a rectangular box

Layout and spacing depend on the box model



Components

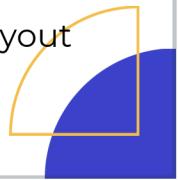
1. **Content:** actual text/image
2. **Padding:** space inside the border
3. **Border:** wraps padding and content
4. **Margin:** space outside the element



Important Note:

box-sizing: border-box;

Includes padding and border inside width/height (Preferred for layout work)





Display Types (Recap)

Search ...

Block Elements

- Start on a new line
- Take full available width

html

<div>, <p>, <section>, <header>

css

display: block;



Inline Elements

- Stay in the same line
- Width & height not respected

html

, <a>,

css

display: inline;



Inline-Block Elements

- Stay inline
- Width & height respected

css

display: inline-block;





WHAT IS FLEXBOX?

Flexible Box Layout, is a CSS module designed to simplify the process of creating responsive and dynamic layouts.

- One-dimensional layout system
- + • Controls layout along row OR column
- Ideal for navbars, cards, alignment



Key Concepts of Flexbox

Flex Container: The parent element that holds the items. `display: flex` or `display: inline-flex`.

Flex Items: The child elements inside the flex container

Axes:

Main Axis: The primary direction of items (horizontal by default).

Cross Axis: Perpendicular to the main axis (vertical by default).





- .flex-container becomes a flex container
- Child elements automatically become flex items
- flex: 1 makes each item take equal width
- justify-content controls horizontal spacing
- align-items controls vertical alignment
- gap controls space between items

```
1 .container {  
2     background-color: #f2f2f2;  
3     padding: 20px;  
4 }  
5  
6 .item {  
7     display: inline-block;  
8     width: 30%;  
9     background-color: #4CAF50;  
10    color: white;  
11    padding: 20px;  
12    text-align: center;  
13    margin-right: 3%;  
14 }
```

```
1 <!DOCTYPE html>  
2 <html>  
3 <head>  
4     <title>Flex Tutorial</title>  
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">  
6  
7     <style>  
8         .flex-container {  
9             display: flex;  
10            justify-content: space-between;  
11            align-items: center;  
12            gap: 15px;  
13            background-color: #f2f2f2;  
14            padding: 20px;  
15        }  
16  
17         .flex-item {  
18             background-color: #4CAF50;  
19             color: white;  
20             padding: 20px;  
21             flex: 1;  
22             text-align: center;  
23         }  
24     </style>  
25 </head>  
26 <body>  
27     <div><h1>Flex Tutorial</h1></div>  
28     <hr>  
29     <div class="flex-container">  
30         <div class="flex-item">Item 1</div>  
31         <div class="flex-item">Item 2</div>  
32         <div class="flex-item">Item 3</div>  
33     </div>  
34  
35 </body>  
36 </html>
```

“Flexbox does not change HTML. It changes how elements behave.”





FLEX-DIRECTION AND JUSTIFY-CONTENT

Search ...



Controls main axis direction

`flex-direction: row; /* default */`

`flex-direction: column;`

`row-reverse`

`column-reverse`

-
- `justify-content: flex-start;`
 - + `justify-content: center;`
 - `justify-content: space-between;`
 - `justify-content: space-around;`
 - `justify-content: space-evenly;`
-



align-items: Aligns items along the cross axis

flex-grow
flex-shrink
flex-basis

**flex: 1; /* grow shrink
basis */**

gap: 20px;

```
1 #items {  
2     align-items: stretch;    /* default */  
3     align-items: center;  
4     align-items: flex-start;  
5     align-items: flex-end;  
6 }  
7 }
```



FLEXBOX EXAMPLE – NAVBAR

```
1 .navbar {  
2   display: flex;  
3   justify-content: space-between;  
4   align-items: center;  
5 }  
6 .menu {  
7   display: flex;  
8   gap: 20px;  
9 }
```

FLEXBOX EXAMPLE – CARD ROW

```
1  
2 .cards {  
3   display: flex;  
4   gap: 20px;  
5 }  
6 .card {  
7   flex: 1;  
8   padding: 20px;  
9 }  
10
```



Typical Web Page Structure

A website's structure is typically organized into distinct sections to ensure clarity, usability, and accessibility.

These sections are defined using **semantic HTML elements**, which help both developers and assistive technologies understand the purpose of each part of the page.



The **header** is usually located at the top of the page and contains elements like the website's title, logo, and sometimes a tagline.





BASIC STRUCTURE OF A WEBSITE

Search ...



The **navigation bar** provides links to the main sections of the site, such as "Home," "About," or "Contact."

The **main content section** is the central area of the page, containing the unique content for that specific page.

The **sidebar** is an optional section that provides supplementary information, such as related links, advertisements, or author bios.

The **footer** is located at the bottom of the page and contains secondary information, such as copyright notices,

`<header>, <nav>, <main>,
<section class="content">Content</section> < >
<aside>, and <footer>`



PRACTICE SESSION +

Building Layout Together

Search ...



Task 1: Create a Navbar

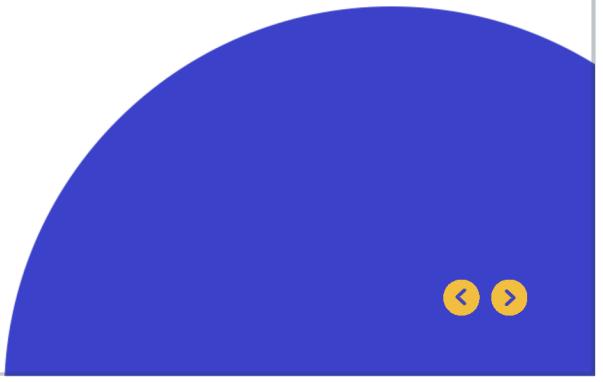
Goal: Create a horizontal navbar with logo on left, links in center, button on right

Requirements

- Logo on left
- Menu items on right
- Items aligned vertically center

Hints

- display: flex
- justify-content: space-between
- align-items: center





Search ...



TASK 2: CREATE CARD SECTION USING FLEXBOX

GOAL: CREATE 3 EQUAL-WIDTH CARDS IN A ROW

Requirements

- Section with 2 or 3 cards
- Cards in a row
- Equal width
- Space between cards

Hints

- display: flex
- gap: 20px;
- flex: 1;



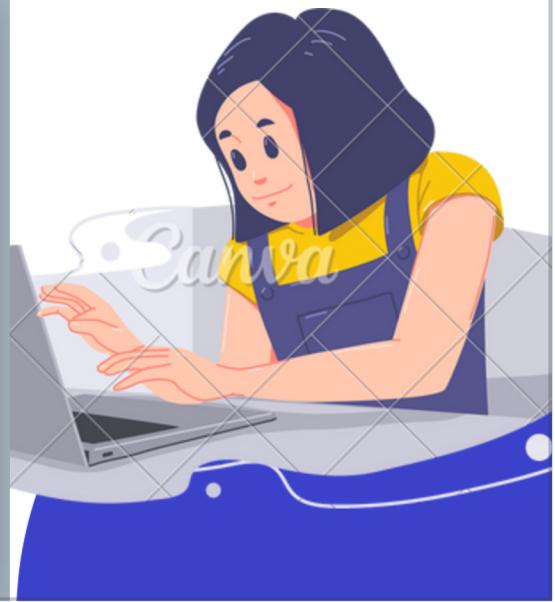


HOME PRACTICE TASKS

Search ...



```
1 Practice Tasks
2 Task 1: Navbar
3
4 Goal: Create a horizontal navbar with logo on left, links in center, button on right
5 Hints:
6
7   Use display: flex on navbar
8
9   Use justify-content: space-between
10
11  Style links as inline elements
12
13 Task 2: Card Section
14
15 Goal: Create 3 equal-width cards in a row and add hover card
16 Hints:
17
18   Use display: flex on container
19
20   Add gap for spacing
21
22   Make cards equal with flex: 1 on each card
23
24   Center content inside cards
25
26 Task 3: Responsive Gallery
27
28 Goal: Create a gallery that wraps to multiple rows
29 Hints:
30
31   Use flex-wrap: wrap
32
33   Set fixed width for image boxes
34
35   Use justify-content: center to center boxes
```





FINAL OUTPUT EXPECTATION

Search ...



Page should contain:

- Header
- Flexbox Navbar
- Card Section
- Image section
- Footer

