

Live Cohort

DAY - 15

**CSS Notes: Transform, Flexbox
(Detailed), and Hover Effects**

CSS Notes

1. Transform: translateX() and translateY()

The transform property lets you move, rotate, scale, or skew elements without affecting the normal layout.

1.1 translateX

Moves an element left or right along the X-axis.

```
1  div {  
2      transform: translateX(100px);  
3  }  
4
```

👉 Moves the element 100px to the right.
Use negative values to move it left.

1.2 translateY

Moves an element up or down along the Y-axis.

```
1  div {  
2      transform: translateY(50px);  
3  }  
4
```

👉 Moves the element 50px downward.
Use negative values to move it upward.

🧠 **Real-life Example:** Adjusting button or image positions in animations.

CSS Notes

2. Centering an Element with Position and Transform

You can center an element both horizontally and vertically using position: absolute and transform: translate.

💻 Example

```
1 <div class="parent">
2   <div class="child">Centered Box</div>
3 </div>
4
```

```
1 .parent {
2   position: relative;
3   height: 300px;
4   background: lightgray;
5 }
6
7 .child {
8   position: absolute;
9   top: 50%;
10  left: 50%;
11  transform: translate(-50%, -50%);
12  background: skyblue;
13  padding: 20px;
14 }
15
```



👉 The box is perfectly centered inside the parent.

💡 **Real-life Use:** Popups, modals, or loading screens.

3. Display Flex (Detailed)

Flexbox helps align and distribute elements efficiently inside a container.

💻 Basic Setup

```
1 <div class="container">
2   <div class="item">1</div>
3   <div class="item">2</div>
4   <div class="item">3</div>
5 </div>
6
```

CSS Notes



```
1 <div class="container">
2   <div class="item">1</div>
3   <div class="item">2</div>
4   <div class="item">3</div>
5 </div>
6
```



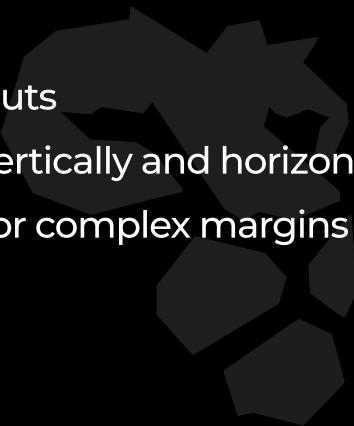
```
1 .container {
2   display: flex;
3   background: lightyellow;
4 }
5 .item {
6   background: lightblue;
7   margin: 5px;
8   padding: 20px;
9 }
```

👉 All items appear in a single horizontal row.

💡 Why Use Flexbox?

- Simplifies responsive layouts
- Easily aligns items both vertically and horizontally
- Eliminates need for float or complex margins

4. Align Items



Controls vertical alignment of flex items.

Value

Description

flex-start

Align items at the top

flex-end

Align items at the bottom

center

Align items in the vertical center

stretch

Items stretch to fill container (default)

CSS Notes

Example



```
1 .container {  
2   display: flex;  
3   align-items: center;  
4   height: 200px;  
5   background: lightgray;  
6 }  
7
```

👉 Commonly used for buttons, icons, or card alignment.

5. Justify Content

Controls horizontal alignment of items.

Value	Description
flex-start	Align left
flex-end	Align right
center	Centered
space-between	Equal space between items
space-around	Equal space around items
space-evenly	Equal space everywhere

CSS Notes

Example

```
1 .container {  
2   display: flex;  
3   justify-content: space-between;  
4 }  
5
```

👉 Use Case: Navigation bar links spaced evenly.

6. Flex Direction

Defines the direction in which flex items are placed.

Value	Description
row	Horizontal (default)
row-reverse	Reversed horizontally
column	Vertical stacking
column-reverse	Reversed vertical order

Example

```
1 .container {  
2   display: flex;  
3   flex-direction: column;  
4 }  
5
```

CSS Notes

7. Flex Wrap

Allows items to wrap onto multiple lines when space runs out.

Value	Description
nowrap	(default) All items on one line
wrap	Items move to next line if needed
wrap-reverse	Wraps in reverse order

Example

```
● ● ●  
1 .container {  
2   display: flex;  
3   flex-wrap: wrap;  
4 }  
5
```



👉 Use Case: Product cards wrapping on smaller screens.

8. Flex Shrink

Defines how much an item shrinks when space is limited.

Example

```
● ● ●  
1 .item {  
2   flex-shrink: 1;  
3 }  
4
```

- Simplifies responsive layouts
- Easily aligns items both vertically and horizontally
- Eliminates need for float or complex margins

CSS Notes

- Simplifies responsive layouts
- Easily aligns items both vertically and horizontally
- Eliminates need for float or complex margins

👉 Use Case: Preventing logos or images from shrinking.

9. Hover Effects

The :hover pseudo-class lets you style elements on mouse hover, adding interactivity.

💻 Example

```
● ● ●
1 button {
2   background-color: royalblue;
3   color: white;
4   padding: 10px 20px;
5   border: none;
6   cursor: pointer;
7   transition: 0.3s ease;
8 }
9 button:hover {
10   background-color: navy;
11   transform: scale(1.05);
12 }
13 }
```



👉 Use Case: Buttons that change color and size when hovered.

CSS Notes



Summary

Concept	Description
transform	Moves or scales elements visually
translateX / Y	Shifts element along X or Y axis
display: flex	Organizes items in flexible layouts
align-items	Controls vertical alignment
justify-content	Controls horizontal alignment
flex-direction	Defines arrangement direction
flex-wrap	Allows wrapping of elements



CSS Notes

flex-shrink

Controls how items shrink

:hover

Adds interactivity to elements

