



# COMPLETE CSS NOTES

## INTRODUCTION TO FLEXBOX FOR ALIGNMENT & STRUCTURE

- Flexbox is a powerful layout system in CSS that helps in aligning and distributing elements efficiently.

### 1. display: flex

Defines a flex container, making its child elements flexible.

Example:

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

### 2. flex-direction

Specifies the direction of the flex items.

- `row` (default) → Items align horizontally.
- `column` → Items align vertically.

Example:

```
.container {  
  display: flex;  
  flex-direction: column;  
}
```



# COMPLETE CSS NOTES

## INTRODUCTION TO FLEXBOX FOR ALIGNMENT & STRUCTURE

### 3. flex-wrap

Controls whether flex items wrap onto multiple lines.

- `nowrap` (default) → Items stay on a single line.
- `wrap` → Items wrap to the next line if needed.

Example:

```
.container {  
  display: flex;  
  flex-wrap: wrap;  
}
```

### 4. flex-shrink

Determines how much a flex item can shrink if needed.

Example:

```
.item {  
  flex-shrink: 2; /* Shrinks twice as fast as others */  
}
```



# COMPLETE CSS NOTES

## INTRODUCTION TO FLEXBOX FOR ALIGNMENT & STRUCTURE

### 5. justify-content

Aligns flex items along the main axis.

- `flex-start` → Items start from the beginning.
- `center` → Items are centered.
- `flex-end` → Items align at the end.
- `space-between` → Equal space between items.
- `space-around` → Equal space around items.

Example:

```
.container {  
  display: flex;  
  justify-content: center;  
}
```



# COMPLETE CSS NOTES

## INTRODUCTION TO FLEXBOX FOR ALIGNMENT & STRUCTURE

### 6. align-items

Aligns flex items along the cross-axis.

- `flex-start` → Items align at the top.
- `center` → Items align in the middle.
- `flex-end` → Items align at the bottom.

Example:

```
.container {  
  display: flex;  
  align-items: center;  
}
```



# PROJECT EXERCISE : TWO-COLUMN LAYOUT WITH FLEXBOX

