

# [CSS FLEXBOX]

Niraj\_Vadhiya

[WD - CSS AND CSS 3]

# [CSS Flexbox]

**Question 1: What is CSS Flexbox, and how is it useful for layout design? Explain the terms flex-container and flex-item.**

**Ans: -** CSS Flexbox (Flexible Box Layout) is a layout model designed to provide a more efficient way to lay out, align, and distribute space among items in a container, even when their size is unknown or dynamic. It is particularly useful for creating responsive layouts that adapt to different screen sizes and orientations.

## **Flex-Container:**

The parent element that holds the flex items.

Defined by setting the display property to flex or inline-flex.

## **Example:**

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



# [CSS Flexbox]

## **Flex-Item:**

The child elements of a flex-container.

These items are laid out according to the flexbox model.

## **Example:**

```
<div class="container">  
  <div class="item">Item 1</div>  
  <div class="item">Item 2</div>  
  <div class="item">Item 3</div>  
</div>
```

## **Benefits of Flexbox:**

**Alignment:** Easily align items horizontally and vertically.

**Order:** Change the order of items without altering the HTML structure.

**Flexibility:** Items can grow or shrink to fill available space.



# [CSS Flexbox]

**Responsiveness:** Create layouts that adapt to different screen sizes.

**Question 2: Describe the properties justify-content, align-items, and flex-direction used in Flexbox.**

**Ans: -**

**1. justify-content:**

Defines how the flex items are distributed along the main axis (horizontal by default).

**Values:**

flex-start: Items are packed at the start of the container.

flex-end: Items are packed at the end of the container.

center: Items are centered along the main axis.

space-between: Items are evenly distributed with the first item at the start and the last item at the end.

space-around: Items are evenly distributed with equal space around them.



# [CSS Flexbox]

space-evenly: Items are evenly distributed with equal space between them.

## Example:

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

## 2. align-items:

Defines how the flex items are aligned along the cross axis (vertical by default).

## Values:

flex-start: Items are aligned at the start of the cross axis.

flex-end: Items are aligned at the end of the cross axis.

center: Items are centered along the cross axis.



# [CSS Flexbox]

baseline: Items are aligned along their baseline.

stretch: Items are stretched to fill the container (default).

## **Example:**

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

## **3. flex-direction:**

Defines the direction in which the flex items are placed in the flex container.

## **Values:**

row: Items are placed in a row, from left to right (default).

row-reverse: Items are placed in a row, from right to left.



# [CSS Flexbox]

column: Items are placed in a column, from top to bottom.

column-reverse: Items are placed in a column, from bottom to top.

## **Example:**

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

## **Example of Flexbox in Action:**

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width,  
initial-scale=1.0">
```



# [CSS Flexbox]

```
<title>Flexbox Example</title>

<style>

  .container {

    display: flex;

    justify-content: space-between;

    align-items: center;

    height: 100vh;

    background-color: #f0f0f0;

  }

  .item {

    background-color: #007bff;

    color: white;

    padding: 20px;

    margin: 10px;

    border-radius: 5px;

  }

</style>

</head>

<body>
```





# [CSS Flexbox]

```
<div class="container">  
  <div class="item">Item 1</div>  
  <div class="item">Item 2</div>  
  <div class="item">Item 3</div>  
</div>  
</body>  
</html>
```

## **In this example:**

The container is a flex-container with `display: flex`.

The items are flex-items.

`justify-content: space-between` distributes the items evenly with space between them.

`align-items: center` centers the items vertically within the container.

