

Q1: Explain CSS Grid and how it differs from Flexbox. When would you use Grid over Flexbox?

Ans : **CSS Grid** is a layout system that allows you to create **two-dimensional layouts**—both rows and columns—on a web page. It helps place items precisely in a grid-like structure.

Difference from Flexbox:

- **Flexbox** → One-dimensional (works in a **row or column**). Good for aligning items in a line.
- **Grid** → Two-dimensional (works in **rows and columns**). Good for complete page or complex layouts.

When to use Grid over Flexbox:

- When you need precise control of **rows and columns**.
- When designing **full-page layouts** or **complex components**.
- Flexbox is better for simple alignment of elements in one direction.

Example of Grid Container and Items:

```
.container {  
  display: grid;  
  grid-template-columns: 100px 100px 100px;  
  grid-template-rows: 50px 50px;  
}
```

Q2: Describe the grid-template-columns, grid-template-rows, and grid-gap properties. Provide examples of how to use them ?

1. **grid-template-columns** → Defines the number and width of **columns** in the grid.

```
.container {  
  grid-template-columns: 100px 200px 100px;  
}
```

2. **grid-template-rows** → Defines the number and height of **rows** in the grid.

```
.container {  
  grid-template-rows: 50px 100px;  
}
```

3. **grid-gap** → Adds **space between rows and columns**.

```
.container {  
  grid-gap: 10px; /* 10px space between rows and columns */  
}
```

Example combining all three:

```
.container {  
  display: grid;  
  grid-template-columns: 150px 150px 150px;  
  grid-template-rows: 80px 80px;  
  grid-gap: 15px;  
}
```