

Question 1: Explain CSS Grid and how it differs from Flexbox. When would you use Grid over Flexbox?

CSS Grid is a layout system in CSS that lets you arrange elements in rows and columns (like a table, but more powerful).

It gives you full control of both horizontal and vertical alignment at the same time.

Flexbox vs Grid

Flexbox works best for one-dimensional layouts arranging items in a row OR in a column.

Grid works best for two-dimensional layouts

arranging items in rows AND columns at the same time.

When to use Grid?

Use Grid when you need a page layout with rows + columns (like a photo gallery or web page structure).

Use Flexbox when you only need to arrange items in one line (like a navbar or button group).

Question 2: Describe the grid-template-columns, grid-template-rows, and grid-gap properties.

Provide examples of how to use them.

1. grid-template-columns → Defines how many columns and their sizes.

Example:

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

This means: 1st column = 100px, 2nd = 200px, 3rd = takes remaining space.

2. grid-template-rows → Defines row heights.

Example:

```
.container {  
  display: grid;  
  grid-template-rows: 100px 150px auto;  
}
```

This means: 1st row = 100px, 2nd = 150px, 3rd = takes remaining space.

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

Example:

```
.container {  
  display: grid;  
  grid-template-columns: 100px 100px 100px;  
  grid-gap: 10px;  
}
```

This makes a 3-column grid with 10px gap between items.

Easy Summary:

- Grid = rows + columns together.
- Flexbox = row OR column.
- grid-template-columns = column sizes.
- grid-template-rows = row sizes.
- grid-gap = space between items.