CSS Layout Interactive Tutorial – Flexbox, Grid & Variables

Starter Code (Base HTML)

Start with this bare HTML structure. Students will add CSS step by step to see the effects live.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title>CSS Layout Tutorial</title>
  <style>
    /* CSS will be added step by step */
  </style>
</head>
<body>
  <h2>Flexbox Example</h2>
  <div class="flex-demo">
    <div>Item 1</div>
    <div>Item 2</div>
    <div>Item 3</div>
  </div>

  <h2>Grid Example</h2>
  <div class="grid-demo">
    <div>Card 1</div><div>Card 2</div><div>Card 3</div>
    <div>Card 4</div><div>Card 5</div><div>Card 6</div>
  </div>

  <h2>CSS Variables Example</h2>
  <div class="card">Card 1</div>
  <div class="card">Card 2</div>
  <div class="card">Card 3</div>
</body>
</html>
```

Step 1: Apply Flex & Grid Rules

Add the following CSS rules into the <style> tag. Observe how Flexbox and Grid arrange the items.

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

.grid-demo {
  display: grid;
  grid-template-columns: repeat(2, 1fr);
}
```

Interactive Task:

- Change flex-direction to 'row' and refresh – what happens?
- Change grid-template-columns to 'repeat(3, 1fr)' – how many columns do you see now?
- Add or remove cards inside .grid-demo and see how Grid auto-places them.

Step 1b: Add Media Queries

Enhance your layout for mid and large screens using min–max ranges.

```
@media (min-width: 600px) and (max-width: 899px) {
  .flex-demo { flex-direction: row; }
  .grid-demo { grid-template-columns: repeat(3, 1fr); }
}

@media (min-width: 900px) {
  .grid-demo { grid-template-columns: repeat(4, 1fr); }
}
```

Interactive Task:

- Resize your browser between 600px and 899px – see the switch to row layout and 3 columns.
- Resize above 900px – check for 4 columns.
- Try changing 600px to 500px – what effect does it have on when the layout switches?

Step 2: Add CSS Variables

Use variables to make your design consistent and easy to update.

```
:root {
  --brand-yellow: #f9d100;
  --text-color: #111;
  --card-bg: #fff;
  --radius: 10px;
}

body {
  background: var(--brand-yellow);
  color: var(--text-color);
}

.card {
  background: var(--card-bg);
  border-radius: var(--radius);
  padding: 12px;
  margin-bottom: 8px;
}
```

Interactive Task:

- Change --brand-yellow to a new color (e.g., lightblue) – the entire background changes.
- Change --radius to 20px – cards should have more rounded corners.
- Add a new variable --heading-size: 24px and use it for font-size.