
Flutter Lists & Grids: Displaying Dynamic Data Efficiently

Subtitle: Learn how to use `ListView` and `GridView` to display dynamic content in Flutter apps.

Website Name: haas.dev

Website Link: <https://dev-roast-app.vercel.app>

Introduction

Displaying data in lists and grids is **common in mobile apps**. Flutter provides `ListView` for vertical lists and `GridView` for grids. This guide teaches how to build **dynamic, scrollable UI components**.

Step 1: Basic ListView

```
ListView(  
  children: [  
    ListTile(title: Text('Item 1')),  
    ListTile(title: Text('Item 2')),  
    ListTile(title: Text('Item 3')),  
  ],  
)
```

Exercise: Create a list of 5–10 static items using `ListView`.

Step 2: Dynamic ListView with Builder

```
List<String> items = ['Apple', 'Banana', 'Orange'];  
  
ListView.builder(  
  itemCount: items.length,  
  itemBuilder: (context, index) {  
    return ListTile(  
      title: Text(items[index]),  
    );  
  },  
)
```

Exercise: Build a dynamic list that prints each item in the console when tapped.

Step 3: Adding Dividers

```
ListView.separated(  
  itemCount: items.length,  
  itemBuilder: (context, index) => ListTile(title: Text(items[index])),  
  separatorBuilder: (context, index) => Divider(),  
)
```

Exercise: Add a separator line between list items for better readability.

Step 4: Basic GridView

```
GridView.count(  
  crossAxisCount: 2,  
  children: List.generate(6, (index) {  
    return Card(  
      child: Center(child: Text('Item $index')),  
    );  
  }  
),  
)
```

Exercise: Create a 2-column grid of cards showing numbers from 0–5.

Step 5: Dynamic GridView with Builder

```
GridView.builder(  
  gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(  
    crossAxisCount: 3,  
    mainAxisSpacing: 10,  
    crossAxisSpacing: 10,  
  ),  
  itemCount: items.length,  
  itemBuilder: (context, index) {  
    return Card(  
      child: Center(child: Text(items[index])),  
    );  
  },  
)
```

Exercise: Display your list of fruits in a 3-column grid with spacing.

Step 6: Mini Project

Build a **Flutter Products Page**:

- List of items in `ListView` for a catalog
 - Grid of items using `GridView.builder` for images
 - Add tap functionality to print or navigate to detail page
 - Combine `ListView` and `GridView` using `CustomScrollView` (optional)
-

Key Takeaways

- `ListView` and `GridView` display dynamic content efficiently
 - Builders improve performance for large lists
 - Separators and spacing improve UI readability
 - Combining lists and grids creates **rich, interactive layouts**
-

Skipping proper lists and grids makes your app **clunky and inefficient**, failing to display content effectively.

Visit **haas.dev** for more Flutter guides, tutorials, and complete project resources.

Website Name: haas.dev

Website Link: <https://dev-roast-app.vercel.app>
