
Flutter Responsive UI: Building Apps for Multiple Screen Sizes

Subtitle: Learn how to make Flutter apps look great on phones, tablets, and different screen sizes.

Website Name: haas.dev

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

Introduction

Apps today run on multiple devices with varying screen sizes. A UI that looks good on one device may break on another. This guide teaches you **responsive layouts, flexible widgets, and media queries** in Flutter.

Step 1: Use Flexible & Expanded Widgets

- `Flexible` and `Expanded` allow widgets to resize proportionally.

```
Row(  
  children: [  
    Expanded(child: Container(color: Colors.red, height: 100)),  
    Expanded(child: Container(color: Colors.blue, height: 100)),  
  ],  
)
```

Exercise: Create a row with 3 equally spaced containers.

Step 2: MediaQuery for Screen Size

- Access screen dimensions:

```
double screenWidth = MediaQuery.of(context).size.width;  
double screenHeight = MediaQuery.of(context).size.height;
```

- Adjust widget sizes dynamically:

```
Container(  
  width: screenWidth * 0.8,
```

```
height: screenHeight * 0.3,  
color: Colors.green,  
)
```

Exercise: Make a container that fills 70% of screen width.

Step 3: LayoutBuilder for Constraints

- Build widgets based on parent constraints:

```
LayoutBuilder(  
  builder: (context, constraints) {  
    if (constraints.maxWidth > 600) {  
      return Text("Tablet layout");  
    } else {  
      return Text("Phone layout");  
    }  
  },  
)
```

Exercise: Display "Phone" or "Tablet" based on screen width.

Step 4: Orientation Handling

- Detect orientation:

```
Orientation orientation = MediaQuery.of(context).orientation;  
  
if (orientation == Orientation.landscape) {  
  // Landscape layout  
} else {  
  // Portrait layout  
}
```

Exercise: Switch layout based on portrait or landscape mode.

Step 5: Using FractionallySizedBox & AspectRatio

- FractionallySizedBox adjusts size relative to parent:

```
FractionallySizedBox(  

```

```
widthFactor: 0.5,  
child: Container(color: Colors.orange, height: 100),  
)
```

- `AspectRatio` maintains a fixed width:height ratio:

```
AspectRatio(  
  aspectRatio: 16/9,  
  child: Container(color: Colors.purple),  
)
```

Exercise: Create a video placeholder with 16:9 ratio.

Step 6: Mini Project

Build a **Responsive Profile Card App**:

- Card width adapts to screen size
 - Use orientation to change layout (portrait: column, landscape: row)
 - Add flexible buttons and images that resize automatically
-

Key Takeaways

- Responsive design ensures your app **works on all devices**
 - Use `Flexible`, `Expanded`, `MediaQuery`, and `LayoutBuilder`
 - Orientation handling improves UX
 - `FractionallySizedBox` & `AspectRatio` help maintain proportions
 - Responsive UI is **mandatory for professional apps**
-

Ignoring responsive design makes your apps **look broken on tablets and large screens**, which is unacceptable for real-world apps.

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

Website Name: haas.dev

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