

[Get started](#)

[Samples & tutorials](#)

[Development](#)

▼ [User interface](#)

[Introduction to widgets](#)

▼ [Building layouts](#)

[Layouts in Flutter](#)

[Tutorial](#)

[Creating adaptive and responsive apps](#)

[Building adaptive apps \(NEW\)](#)

[Understanding constraints](#)

[Box constraints](#)

[Adding interactivity](#)

[Assets and images](#)

▶ [Navigation & routing](#)

▶ [Animations](#)

▶ [Advanced UI](#)

[Widget catalog](#)

▶ [Data & backend](#)

▶ [Accessibility & internationalization](#)

▶ [Platform integration](#)

▶ [Packages & plugins](#)

▶ [Add Flutter to existing app](#)

▶ [Tools & features](#)

▶ [Migration notes](#)

[Testing & debugging](#)

[Performance & optimization](#)

[Deployment](#)

[Resources](#)

[Reference](#)

[Who is Dash?](#)

[Widget index](#)

[API reference](#)

[flutter CLI reference](#)

[Package site](#)

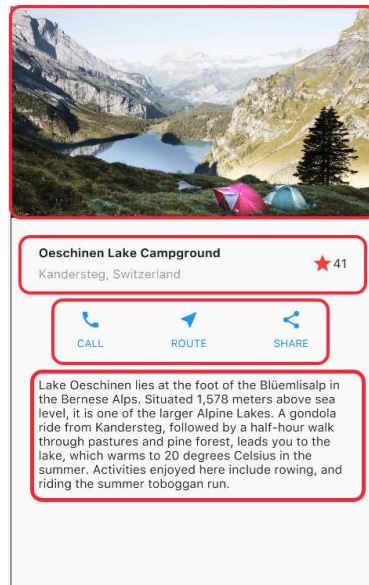
```
16 | +           title: Text('Flutter layout demo'),
17 | 17 |         ),
18 | 18 |         body: Center(
19 | 19 |           child: Text('Hello World'),
```

## Step 1: Diagram the layout

The first step is to break the layout down to its basic elements:

- Identify the rows and columns.
- Does the layout include a grid?
- Are there overlapping elements?
- Does the UI need tabs?
- Notice areas that require alignment, padding, or borders.

First, identify the larger elements. In this example, four elements are arranged into a column: an image, two rows, and a block of t

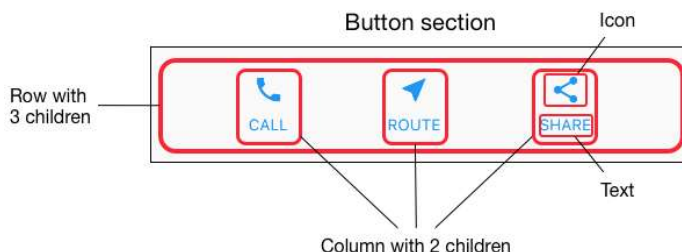


Column elements (circled in red)

Next, diagram each row. The first row, called the Title section, has 3 children: a column of text, a star icon, and a number. Its first child, the column, contains 2 lines of text. That first column takes a lot of space, so it must be wrapped in an Expanded widget.



The second row, called the Button section, also has 3 children: each child is a column that contains an icon and text.



Once the layout has been diagrammed, it's easiest to take a bottom-up approach to implementing it. To minimize the visual confu of deeply nested layout code, place some of the implementation in variables and functions.

## Step 2: Implement the title row

First, you'll build the left column in the title section. Add the following code at the top of the `build()` method of the `MyApp` class:

```
lib/main.dart (titleSection)
```