



Flutter Course

Responsive App in Flutter

Task 7



Fatma Abdelghany Mohammed



Responsive Design

Responsive Design is a development and design technique that allows websites or applications to 'adapt' to different screen sizes, without affecting user experience or usability.

Responsive Design even optimizes the user's browsing experience by making an application responsive according to the device.



Create A Responsive Flutter App

1. Media Query

Media Query can be used to get the real-time size (width/height) and orientation (portrait/landscape) of the window screen. It suggests the orientation and size of the app.





Create A Responsive Flutter App

2. Layout Builder

Layout Builder is just a simplified version of Media Query. The main difference between Media Query and Layout Builder is that Media Query is based on the full context of the screen rather than just the size of a particular widget; on the other hand, Layout Builder determines the maximum width and height of a specific widget only.

Layout Builder class provides the Box Constraints object that can be used for determining the `maxWidth` and `maxHeight` of the widget.





Create A Responsive Flutter App

3. Orientation Builder

Orientation Builder class can be used to determine a widget's current orientation.

The Orientation Builder widget is similar to the Layout Builder class. One can get the Orientation object using the builder property of OrientationBuilder class. For example, you can use the OrientationBuilder class to change the number of columns of GridView.





Create A Responsive Flutter App

4. Expanded and Flexible Widgets

Expanded and Flexible Widgets are two widgets that can be used inside Row, Column, or Flex to give their children the flexibility in order to expand to fill the available space. The only difference is that the Expanded widget requires the child to fill all the available space, whereas Flexible does not. Also, Expanded and Flexible widgets can be used to get a responsive Flutter UI that works with percentages instead of hardcoded values.





Create A Responsive Flutter App

5. Aspect Ratio widget

The Aspect Ratio widget can be used to size the child to a specific aspect ratio. When developing an app, one can neglect its size but must consider the aspect ratio. The AspectRatio widget thus assists you at this point by sizing the child value to a specific aspect ratio and ensures responsive design in Flutter.

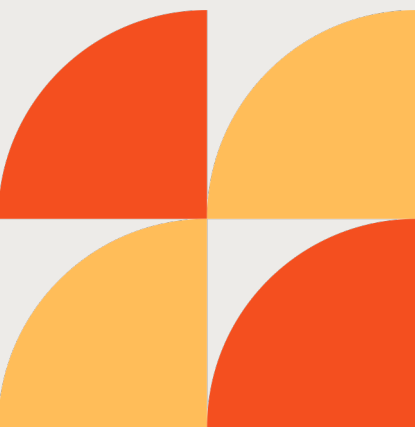


Create A Responsive Flutter App

6. Fractionally Sized Widgets

Fractionally Sized Widgets are useful when the designs are calling for dimensions that are relative. For example, when a button is occupying 80% of the app's width whereas the margin takes only 10%, fractionally sized box Widgets can help with that.

You can even wrap this Fractionally Sized Box in a flexible widget so that it can play well with a row or a column.





Create A Responsive Flutter App

7. CustomMultiChildLayout

`CustomMultiChildLayout` is a widget that utilizes a delegate to specify the size and position of multiple children within it. This delegate is responsible for defining the layout constraints of each child and determining their positioning within the parent widget. While the delegate can define the size of the parent, it cannot be dependent on the size of its children.

`CustomMultiChildLayout` is a suitable choice when there are intricate connections between the positioning and size of various widgets.

