

Abhishek Tyagi
Flutter and Dart
Assignment - 4
28 Sept. 2023

NETWORK BULLS
WHERE CAREERS FLY

11

Ans 1) In flutter, a widget is a fundamental building block used to construct user interfaces. Widgets are a central concept in flutter and are used to represent everything you see on the screen, from simple elements like buttons and text to complex layouts and entire screens. Importance:

- * Reusable components which can encapsulate UI elements and behaviors.
- * Composability, i.e., nest widgets within other widgets to build complex interfaces.
- * Hot Reload: It allows you to make changes to code and see results instantly in running application. This is possible because of flutter UI being composed of widgets.
- * Performance, dynamic UI.

There are 2 types of widgets:

- * Stateless: Immutable. Example: Text, Icon, Image.
- * Stateful: Mutable. They maintain a separate mutable state object that can be modified. E.g.: TextField, Checkbox, custom widgets that have dynamic behaviors.

Ans 2) Material Design is a design language and visual framework developed by Google. It can be used to create visually appealing and consistent UI using Google's design principles.

- Widgets and components : Flutter provides a wide range of them to allow developers to easily create UI elements such as buttons, text fields, cards etc.
- Material Theme : Includes specifications for colors, typography, and other design elements, ensuring a consistent and visually pleasing look and feel across the app.
- Customization : Highly customizable to fit various colors, fonts etc.
- Cross platform
 - Elevation and shadows
 - Adaptive layouts

Ans3 MaterialApp is a top level widget that is used to define and config the characteristics and behaviors of app. It sets the overall theme, navigation and structure for app. It includes several properties and functionalities like title, theme, routes etc. It is typically used as root widget of a Flutter app and wraps entire widget tree.
Differences between Scaffold and containers;

* Scaffold is primarily used for defining overall structure and layout of a screen or page, including app bar, body content and other structural elements.

(3)

Container is used for customization & decoration of individual widgets or groups of widgets. It allows you to apply padding, margin, colors etc to its child widget.

* Scaffold provides a predefined layout structure with specific areas for app bar, body content etc. contained is a single widget that can contain and style other widgets as child.

Ans 4) It is a crucial configuration file to list dependencies that your project relies on. It provide metadata about project such as name, description, author, version no., homepage URL etc. It is useful for documentation purposes. You can also declare assets (media files) that are part of your project in it.

Ans 5) main()

- Entry point for Dart programs/flutter apps
- Usually defined in a dart file called main.dart
- Standard dart function
- Perform any necessary setup and non-flutter specific tasks before starting flutter framework
- Executed before runApp()

- Initializes and runs Flutter framework to render app's UI.
- Typically called within 'main()' function in some file where it is defined.
- Flutter-specific function.
- Initializes flutter framework and specifies root widget for app's UI.
- Executes after main(), initializing flutter framework.

Ans6) Stateless widget

- Immutable.
- Does not rebuild itself automatically when data changes.
- 'build' method called when widget is created.
- Generally more performant as it does not need to manage separate state object.
- No lifecycle methods.
- Does not manage its own state.
- Used for static or unchanging UI elements.

Stateful widget

- Mutable.
- Rebuilds itself when its internal state changes.
- 'build' method can be called multiple times.
- Can be less performant due to overhead of managing a separate state object.

- Provides lifecycle methods like `initState`, `didUpdateWidget`, `dispose` etc.
- Manages own state using 'state' object.
- Used for UI elements that need to change in response to changing data.

Ans7) Use 'mainAxisAlignment' when you want to control how child widgets are aligned along main axis of a container as by default common values include start, center, end, spaceBetween, spaceEvenly. Use 'crossAxisAlignment' when you want to control how child widgets are aligned along cross axis of a container as layout. Common values include: stretch, center, end, stretch, baseline.

Ans8) • The Expanded widget is typically used inside a 'Row', 'Column' or 'Flex' to indicate that a child should take up the available remaining space along main axis. It ensures that a child widget expands to fill all available space, pushing other widgets if necessary.

• The flexible 'widget' is more versatile, allowing you to specify a flex factor, which determines how remaining space is distributed.

among flexible widgets. By assigning different flex factors to flexible widgets, you can control the proportion of space they occupy relative to each other. If parents not fine-grained control over space distribution compound to 'expanded'.

Row, Column

- Layout widgets used to arrange child widgets horizontally & vertically respectively.
- Commonly used for standard layouts such as navigation forms, lists etc.

center, used to center-align its child widget within available space.

- Placeholder
- Temporary widget during development process.
- It displays a simple "X" as others placeholders text, helping developer visualize where other widgets will be placed in layout.

Padding

- Used to add space around its child widget. Can be specified for all sides - top, bottom, right, left.
- Commonly used for adding margins and spacing around UI elements.

Theme

- Used to define and apply a consistent visual theme across an app.
- It allows you to specify attributes like colors, typography and shapes that apply to various UI elements throughout an app.

Ansible code + output in separate pdf file