

INTERVIEW QUESTION-ANSWERS OF FLUTTER

1. What is Flutter?

ANS: - Flutter is an open-source UI SDK by Google for building cross-platform applications from a single codebase. It allows developers to create apps for Android, iOS, web, and desktop with a focus on expressive and flexible UI.

2. What programming language does Flutter use?

ANS: - Flutter uses **Dart**, a modern, object-oriented, and class-based programming language developed by Google. Dart is designed for fast performance and ease of learning.

3. What are widgets in Flutter?

ANS: - Widgets are the fundamental building blocks of a Flutter application. Everything in Flutter, including layout and UI components, is a widget. Widgets are categorized as:

- **Stateless Widgets:** Immutable and do not change over time.
- **Stateful Widgets:** Maintain mutable state that can change during the app's lifecycle.

4. What is the Flutter Hot Reload feature?

ANS: - **Hot Reload** allows developers to instantly view changes made to the code without restarting the application. It improves development speed and debugging efficiency by retaining the current app state.

5. What are the types of widgets in Flutter?

ANS: - Widgets in Flutter can be categorized as:

- **Visible Widgets (Structural):** Container, Text, Image, Button.
- **Invisible Widgets (Functional):** Theme, MediaQuery, Navigator.

6. How does Flutter handle layouts?

ANS: - Flutter uses a flexible and hierarchical layout system. Common layout widgets include:

- **Column and Row:** Arrange children vertically and horizontally.
- **Stack:** Overlap children.
- **Container:** Add padding, margins, and constraints.
- **Expanded and Flexible:** Manage space within Row or Column.

7. What is Flutter's rendering engine?

ANS: - Flutter uses its own rendering engine called **Skia**, which allows it to render directly on the canvas without relying on native components. This ensures a consistent UI across platforms.

8. How does Flutter achieve cross-platform support?

ANS: - Flutter achieves cross-platform support by compiling its code into native ARM or x86 machine code using Dart's AOT (Ahead-of-Time) compilation and rendering UI directly via the Skia engine.

9. How do you manage state in Flutter?

ANS: - Flutter supports various state management approaches, including:

- **setState**: Built-in and simple for local state.
- **Provider**: Recommended by Flutter for reactive state management.
- **Bloc/Cubit**: Uses the BLoC (Business Logic Component) pattern.
- **Redux**: Inspired by the Redux library in JavaScript.
- **Riverpod**: A modern and flexible state management solution.

10. What is the role of keys in Flutter?

ANS: - Keys in Flutter preserve the state of widgets when they are moved within the widget tree. Common types of keys:

- **GlobalKey**: Maintains a reference to the widget throughout its lifecycle.
- **UniqueKey**: Ensures uniqueness for widgets.

11. What are Flutter's debugging tools?

ANS: -

- **DevTools**: Flutter's suite for debugging, profiling, and performance analysis.
- **Flutter Inspector**: Visualizes the widget tree and layouts.
- **Dart Analyzer**: Identifies code issues.
- **Logs & Print Statements**: Useful for tracking execution.

12. What are plugins and packages in Flutter?

ANS: -**Packages**: Reusable code modules shared on pub.dev.

Plugins: Packages with native platform integration (e.g., accessing device hardware like camera or GPS).