



100 Flutter Interview Questions

part - 3

#flutterdaily



Sayed Abdulaziz



@sayedabdulaziz

21) What is the tree shaking in Flutter?

Tree shaking is an optimization technique to remove the unused module in the bundle during the build process. It is a dead code elimination technique used to optimize the code.



22) What are DevTools in Flutter?

DevTools in Flutter are a set of tools used for performance management and debugging. With these tools, you can inspect the UI layout, diagnose the UI performance issues, perform source-level debugging, view general log & diagnostics information, and more. This tool is still in preview release but you can test the alpha version of this tool by clicking the “beaker” icon in the upper-right corner of DevTools.



23) What is The Flex widget in Flutter?

The Flex widget allows you to control the axis along which the children are placed (horizontal or vertical). This is referred to as the main axis. If you know the main axis in advance, then consider using a Row (if it's horizontal) or Column (if it's vertical) instead, because that will be less verbose.



24) What are the differences between expanded and flexible widgets?

Flexible is use to resize the widgets in rows and columns. It's mainly used to adjust the space of the different child widgets while keeping the relation with their parent widgets.

Meanwhile, Expanded changes the constraints sent to the children of rows and columns; it helps to fill the available spaces there. Therefore, when you wrap your child in an Expanded widget it fills up the empty spaces.



25) Material Vs Cupertino Widget?

Cupertino widgets are used to build an iOS-like app and MaterialApp is used to build an Android (Material) app.

Material widgets implement the Material design language for iOS, Android, web, and desktop.

Cupertino widgets implement the current iOS design language based on Apple's Human Interface Guidelines.

The Material Design language was created for any platform, not just Android. When you write a Material app in Flutter, it has the Material look and feels on all devices, even iOS. If you want your app to look like a standard iOS-styled app.

you would use the Cupertino library. You can technically run a Cupertino app on either Android or iOS, but (due to licensing issues) Cupertino won't have the correct fonts on Android. For this reason, use an iOS-specific device when writing a Cupertino app.



26) What is a Null Aware Operator?

Null-aware operators in dart allow you to make computations based on whether or not a value is null. It's shorthand for longer expressions. A null-aware operator is a nice tool for making nullable types usable in Dart instead of throwing an error.

The most common use of the Null aware operator is when a developer wants to parse JSON data from the server and after parsing JSON, the user can check whether the JSON is empty or not using the IF-Else condition.



27) What is Form, textfield and textFormField?

If you making a Form where you require to save, reset, or validate operations- use TextFormField. Else For Simple user input capture TextField is sufficient. TextFormField, which integrates with the Form widget.

This is a convenience widget that wraps a TextField widget in a FormField. A Form ancestor is not required. The Form simply makes it easier to save, reset, or validate multiple fields at once. To use without a Form, pass a GlobalKey to the constructor and use GlobalKey.currentState to save or reset the form field.



28) Difference between StreamBuilder and FutureBuilder?

FutureBuilder is used for one-time response, like taking an image from the Camera, getting data once from the native platform (like fetching device battery), getting file reference, making an HTTP request, etc.

On the other hand, StreamBuilder is used for fetching some data more than once, like listening for a location update, playing music, stopwatch, etc.

Both StreamBuilder and FutureBuilder have the same behavior: They listen to changes on their respective object. And trigger a new build when they are notified of a new value.



29) What is the Future?

A Future is used to represent a potential value, or error, that will be available at some time in the future. Receivers of a Future can register callbacks that handle the value or error once it is available.

To perform asynchronous operations in Dart, you can use the Future class and the `async` and `await` keywords. A future (lower case "f") is an instance of the Future (capitalized "F") class.

A future represents the result of an asynchronous operation and can have two states: uncompleted or completed.



30) What is the Difference between synchronous operation and synchronous function also Difference between asynchronous operation and asynchronous function?

- synchronous operation: A synchronous operation blocks other operations from executing until it completes.
- synchronous function: A synchronous function only performs synchronous operations.
- asynchronous operation: Once initiated, an asynchronous operation allows other operations to execute before it completes.
- asynchronous function: An asynchronous function performs at least one asynchronous operation and can also perform synchronous operations.



Like & follow



Share post



Thank you



@sayedabdulaziz