1. What is flutter?

Flutter is an open-source UI toolkit or software development kit (SDK) developed by Google to build mobile, web, and desktop applications from a single codebase.

2. Advantages of Flutter or Features

- Cross-platform Development
- Hot Reloading and Hot Restart
- Similar to Native App performance
- Good Community Support
- Minimal Code
- Documentation
- Flexible UI

3. What are the types of widgets present in Flutter?

- **Stateless Widget:** A widget that does nothing is a Stateless Widget. In essence, they are static and don’t store any state. Thus, they don’t save values that may change. Icon, Text, and IconButton are some examples of Stateless widgets.

- **Stateful Widget:** A widget that does anything is a Stateful Widget. Stateful widgets are dynamic by nature, which means they can monitor changes and update the UI accordingly. Radio, Slider, CheckBox, and Image are some of the examples of Stateful widgets.

4. Talk about different build modes in Flutter.

- **Debug mode** is used to debug the application on the physical device, emulator, or simulator.
- **Profile mode** is used to analyze the performance of your app. Here, some extensions and tracing are enabled.
- **Release mode** is enabled to deploy your app. Here, assertions, service extensions, and debugging are disabled.

5. What is Tree shaking?

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.

6. What is pubspec.yaml file?

The pubspec.yaml file is used to define the dependencies of your Flutter project. This metadata information is written in the YAML language. This file can have the following fields Name, Version, Description, Homepage, Repository, Documentation, Dependencies, Environment, and more about the pubspec.yaml file.

7. What is Context?

Context is a handle to the location of the widget in the widget tree. It is a BuildContext instance that gets passed to the builder of a widget to know where it is inside the widget tree. It is nothing but a reference to the location of a Widget within the tree structure of all the built widgets.

8. Explain BuildContext

BuildContexts are used to identify or locate widgets in widget trees. Each widget has its BuildContext, i.e., one BuildContext per widget. We're using it to find references to other widgets and themes. In addition, you can utilize it to interact with widget parents and access widget data.

9. Explain Devtools

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.

10. What is the HTTP package?

The HTTP package is used in the Flutter project to provide a simple way to fetch data from the Internet.

11. How can we create HTTP requests in Flutter?

```
http.get('https://xyz.com/users/1');
```

It will return a Future <http.Response>.

12. What is the use of Navigation.push and Navigation.pop function?

The push method is used to add a route to the stack of routes managed by the navigator.

The pop method is used to remove the current route from the stack of routes managed by the navigator.

13. Explain the flutter widgets.

Each element on the Flutter app's screen is considered a widget. In Flutter, the UI is built using widgets.

14. What do you understand from hot reload and hot restart?

Hot reload means injecting the updated source code files into running Dart VM (Virtual Machine). The Hot reload process does not add only new classes but it also adds properties, fields, and methods to existing classes, and changes existing functions.

Hot restart works by resetting the app's current state to the app's initial state.

15. What do you understand from 'State'? What is the use of the setState() method?

The state is the data that your application currently showing.

State refers to the information that you can read synchronously when a widget is built and it might change during the lifetime of the widget.

The 'setState' method is used to notify the changed state of an internal object.

16. Explain the lifecycle of a Stateful Widget.

- createState
- initState

- didChangeDependencies
- build
- didUpdateWidget
- setState
- deactivate
- dispose

17. What is the Container class in Flutter?

The Container class provides the ability to create a widget with certain properties like padding, borders, height, width, border, etc.

18. What is the difference between SizedBox VS Container?

The Container in Flutter is a parent widget that can contain multiple child widgets and manage them efficiently through width, height, padding, background color, etc. If we have a widget that needs some background styling be it a color, shape, or size, we may wrap it in a container widget.

The SizedBox widget in Flutter is a box that comes in a specified size. Unlike Container, it does not allow us to set the color or decoration for the widget. We can only use it for sizing the widget passed as a child. It means it forces its child widget to have a specific width or height.

19. What are packages and plugins in Flutter?

A package is a group of similar types of classes, interfaces, and sub-packages. The packages and plugins help us to build the app without having to develop everything from packages.

20. Why is the Android and iOS folder in the Flutter project?

Android: This folder holds a complete Android project. It is used when you create the Flutter application for Android. When the Flutter code is compiled into the native code, it will get injected into this Android project, so that the result is a native Android application. **For Example**: When you are using the Android emulator, this Android project is used to build the Android app, which is further deployed to the Android Virtual Device.

iOS: This folder holds a complete Mac project. It is used when you build the Flutter application for iOS. It is similar to the Android folder, which is used when developing an app for Android. When the Flutter code is compiled into the native code, it will get injected into this iOS project, so that the result is a native iOS application.

21. What is Tween Animation?

In a tween animation, it is required to define the start and endpoint of animation. It means the animation begins with the start value, then goes through a series of intermediate values, and finally reaches the end value. It also provides the timeline and curve, which defines the time and speed of the transition. The widget framework provides a calculation of how to transition from the start to and endpoint.

22. What is Stream in Flutter?

A stream is like a pipe, you put a value on the one end, and if there's a listener on the other end that listener will receive that value. A Stream can have multiple listeners and all of those listeners will receive the same value when it's put in the pipeline. The way you put values on a stream is by using a StreamController.

23. Explain the difference between "??" and "?" Operators.

expr1 ?? expr2

The "??" operator is used to evaluate and return the value between two expressions based on the given condition.

condition ? expr1 : expr2

This operator first checks the condition, and if it is true, it will evaluate expr1 and return its value (if the condition is matched). Otherwise, it evaluates and returns the value of expr2.

24. What are keys in Flutter, and when to use it?

- Keys in Flutter are used as an identifier for Widgets, Elements, and SemanticsNodes. We can use it when a new widget tries to update an existing element; then, its key should be the same as the current widget key associated with the element.
- Keys should not be different amongst the Elements within the same parent.
- The subclasses of Key must be a GlobalKey or LocalKey.
- Keys are useful when we try to manipulate (such as adding, removing, or reordering) a collection of widgets of the same type that hold some state.

25. What is the difference between WidgetsApp and MaterialApp?

WidgetsApp is used for basic navigation. It includes many foundational widgets together with the widgets library that Flutter uses to create the UI of our app.

MaterialApp, along with the material library, is a layer that is built on the top of WidgetsApp and its library. It implements Material Design that provides a unified look and feel to our app on any platform.

26. What types of tests can you perform in Flutter?

Unit Tests: It tests a single function, method, or class. Its goal is to ensure the correctness of code under a variety of conditions. This testing is used for checking the validity of our business logic.

Widget Tests: It tests a single widget. Its goal is to ensure that the widget's UI looks and interacts with other widgets as expected.

Integration Tests: It validates a complete app or a large part of the app. Its goal is to ensure that all the widgets and services work together as expected.

27. What are Null-aware operators?

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 ?? operator returns the first expression if and only if it is not null.

- The ??= operator in Dart is used when we want to assign a value if and only if it is not null.
- The ? operator is used when we want to make sure that we don’t invoke a function of a null value. It will call a function if and only if the object is not null.

28. What is the use of Ticker in Flutter?

We use a ticker to tell how often our animation is refreshed in Flutter.

29. What is the use of Mixins?

Multiple inheritances are not supported by Dart. Thus, we need mixins to implement multiple inheritances in Flutter/Dart. The use of mixins makes it easy to write reusable class code in multiple class hierarchy levels.

30. What is state management?

Using it, states of various UI controls are centralized to handle data flow across an application. It can be a text field, radio button, checkbox, dropdown, toggle, form, and so on. In Flutter, state management can be categorized into two types as follows:

- **Ephemeral State:** Ephemeral state is also called UI state or local state, and it pertains to a particular widget. In other words, it is a state that is contained within the specific widget. By means of StatefulWidget, Flutter provides support for this state.
- **App State:** This is different from the ephemeral state since it is a state that we intend to share across different parts of the app and that we want to maintain between sessions. These types of states can thus be used globally. By means of scoped_model, Flutter provides support for this state.

31. Explain Flutter Provider.

The provider package is an easy-to-use package which is a wrapper around the InheritedWidgets that makes it easier to use and manage. It provides a state management technique that is used for managing a piece of data around the app.

32. What is await in Flutter?

Until the async method is finished, await interrupts the process flow.