Android Interview Questions

1) What is Android?

Android is an open-source, Linux-based operating system used in mobiles, tablets, televisions, etc.

2) Who is the founder of Android?

Andy Rubin.

3) Explain the Android application Architecture.

Following is a list of components of Android application architecture:

Services: Used to perform background functionalities.

Intent: Used to perform the interconnection between activities and the data passing mechanism.

Resource Externalization: strings and graphics.

Notification: light, sound, icon, notification, dialog box and toast.

Content Providers: It will share the data between applications.

4) What are the code names of android?

Aestro

Blender

Cupcake

Donut

Eclair

Froyo

Gingerbread

Honeycomb

Ice Cream Sandwich

Jelly Bean

KitKat

Lollipop

Marshmallow

5) What are the advantages of Android?

Open-source: It means no license, distribution and development fee.

Platform-independent: It supports Windows, Mac, and Linux platforms.

Supports various technologies: It supports camera, Bluetooth, wifi, speech, EDGE etc. technologies.

Highly optimized Virtual Machine: Android uses a highly optimized virtual machine for mobile devices, called DVM (Dalvik Virtual Machine).

6) Does android support other languages than java?

Yes, an android app can be developed in C/C++ also using android NDK (Native Development Kit). It makes the performance faster. It should be used with Android SDK.

7) What are the core building blocks of android?

The core building blocks of Android are:

Activity

View

Intent

Service

Content Provider

Fragment etc.

8) What is activity in Android?

Activity is like a frame or window in java that represents GUI. It represents one screen of android.

9) What are the life cycle methods of android activity?

There are 7 life-cycle methods of activity. They are as follows:

onCreate()

onStart()

onResume()

onPause()

onStop()

onRestart()

onDestroy()

10) What is intent?

It is a kind of message or information that is passed to the components. It is used to launch an activity, display a web page, send SMS, send email, etc. There are two types of intents in android:

Implicit Intent

Explicit Intent

11) How are view elements identified in the android program?

View elements can be identified using the keyword findViewById.

12) Define Android toast.

An android toast provides feedback to the users about the operation being performed by them. It displays the message regarding the status of operation initiated by the user.

13) Give a list of impotent folders in android

The following folders are declared as impotent in android:

AndroidManifest.xml

build.xml

bin/

src/

res/

assets/

14) Explain the use of 'bundle' in android?

We use bundles to pass the required data to various subfolders.

15) What is an application resource file?

The files which can be injected for the building up of a process are called as application resource file.

16) What is the use of LINUX ID in android?

A unique Linux ID is assigned to each application in android. It is used for the tracking of a process.

17) Can the bytecode be written in java be run on android?

No

18) List the various storages that are provided by Android.

The various storage provided by android are:

Shared Preferences

Internal Storage

External Storage

SQLite Databases

Network Connection

19) How are layouts placed in Android?

Layouts in Android are placed as XML files.

20) Where are layouts placed in Android?

Layouts in Android are placed in the layout folder.

21) What is the implicit intent in android?

The Implicit intent is used to invoke the system components.

22) What is explicit intent in android?

An explicit intent is used to invoke the activity class.

23) How to call another activity in android?

```
Intent i = new Intent(getApplicationContext(), ActivityTwo.class);

startActivity(i);
```

What is service in android?

A service is a component that runs in the background. It is used to play music, handle network transaction, etc.

25) What is the name of the database used in android?

SQLite: An opensource and lightweight relational database for mobile devices.

26) What is AAPT?

AAPT is an acronym for android asset packaging tool. It handles the packaging process.

27) What is a content provider?

A content provider is used to share information between Android applications.

28) What is fragment?

The fragment is a part of Activity by which we can display multiple screens on one activity.

29) What is ADB?

ADB stands for Android Debug Bridge. It is a command line tool that is used to communicate with the emulator instance.

30) What is NDK?

NDK stands for Native Development Kit. By using NDK, you can develop a part of an app using native language such as C/C++ to boost the performance.

31) What is ANR?

ANR stands for Application Not Responding. It is a dialog box that appears if the application is no longer responding.

32) What is the Google Android SDK?

The Google Android SDK is a toolset which is used by developers to write apps on Android-enabled devices. It contains a graphical interface that emulates an Android-driven handheld environment and allows them to test and debug their codes.

33) What is an APK format?

APK is a short form stands for Android Packaging Key. It is a compressed key with classes, UI's, supportive assets and manifest. All files are compressed to a single file is called APK.

34) Which language does Android support to develop an application?

Android applications are written by using the java (Android SDK) and C/C++ (Android NDK).

35) What is ADT in Android?

ADT stands for Android Development Tool. It is used to develop the applications and test the applications.

36) What is View Group in Android?

View Group is a collection of views and other child views. It is an invisible part and the base class for layouts.

37) What is the Adapter in Android?

An adapter is used to create a child view to present the parent view items.

38) What is nine-patch images tool in Android?

We can change bitmap images into nine sections with four corners, four edges, and an axis.

39) Which kernel is used in Android?

Android is a customized Linux 3.6 kernel.

40) What is application Widgets in Android?

Application widgets are miniature application views that can be embedded in other applications and receive periodic updates.

41) Which types of flags are used to run an application on Android?

Following are two types of flags to run an application in Android:

FLAG_ACTIVITY_NEW_TASK

FLAG_ACTIVITY_CLEAR_TOP

42) What is a singleton class in Android?

A singleton class is a class which can create only an object that can be shared by all other classes.

43) What is sleep mode in Android?

In sleep mode, CPU is slept and doesn't accept any commands from android device except Radio interface layer and alarm.

44) What do you mean by a drawable folder in Android?

In Android, a drawable folder is compiled a visual resource that can use as a background, banners, icons, splash screen, etc.

45) What is DDMS?

DDMS stands for Dalvik Debug Monitor Server. It gives the wide array of debugging features:

Port forwarding services

Screen capture

Thread and heap information

Network traffic tracking

Location data spoofing

46) Define Android Architecture?

The Android architecture consists of 4 components:

Linux Kernal

Libraries

Android Framework

Android Applications

47) What is a portable wi-fi hotspot?

The portable wi-fi hotspot is used to share internet connection to other wireless devices.

48) Name the dialog box which is supported by Android?

Alert Dialog

Progress Dialog

Date Picker Dialog

Time picker Dialog

49) Name some exceptions in Android?

Inflate Exception

Surface.OutOfResourceException

SurfaceHolder.BadSurfaceTypeException

WindowManager.BadTokenException

50) What are the basic tools used to develop an Android app?

JDK

Eclipse+ADT plugin

SDK Tools

Flask Interview Questions and Answers

Q 1. What is Flask?

Answer: Flask is a microweb framework that provides an API to build up web applications. Flask's framework is also easier to learn because of its diversified working style. Flask is based on the WSGI (Web Server Gateway Interface) toolkit and the Jinja2 template engine. It is very flexible to implement a simple web application. Also, Flask provides visual debugging, which gives more control over the component.

Q 2. What are the features of Flask Python?

Answer:

Built-in web server and debugger

Compatibility with most of the latest technologies.

High scalability and flexibility for simple web applications.

Integrated support for unit testing

Securing cookies in client-side sessions

Dispatching RESTful request

Google App Engine compatibility

Unicode support

Web Server Gateway Interface(WSGI) compliance

Q 3. What is the difference between Flask and Django?

Answer:

Flask	**Django**
Flask is a WSGI framework	Django is a Full-stack web framework
It allows multiple types of databases.	It doesn't support multiple types of databases.
Use SQL Alchemy	Build-in ORM

Diversified Working Style	Monolithic Working Style
Arbitrary structure	Conventional Project Structure
It supports API	It does not have any support for API
It does not support Dynamic HTML pages	Django accepts Dynamic Pages.
It has support for Visual debug	No support for Visual Debug
It doesn’t offer a built-in bootstrapping tool.	Django-admin enables us to start building web applications without any external input,
URL dispatcher is a RESTful request.	URL dispatcher is Robust Documentation.

Q 4. What is the default host port and port of Flask?

Answer: The default local host of the flask is 127.0.0.1, and the default port is 5000.

Q 5. Which databases are Flask compatible with?

Answer: As a backend database, Flask supports SQLite and MySQL. DbAdapters are used to support various databases. It comes with an SQLDbAdapter that allows you to connect to a variety of SQL databases using Flask-SQLAlchemy, including MySQL, Oracle, PostgreSQL, SQLite, Sybase, Firebird, and others. It includes a MongoDbAdapter that allows you to connect to MongoDB databases using Flask-MongoEngine.

Q 6. why do we use Flask(__name__) in Flask?

Answer: The __name__ parameter is a Python built-in variable that is set to the name of the current module. When we pass __name__ as an argument to the Flask class constructor, it helps Flask to determine where to locate resources such as templates and static files.

Q 7. What is routing in Flask?

Answer: App Routing means mapping the URLs to a specific function that will handle the logic for that URL. Modern web frameworks use more meaningful URLs to help users remember the URLs and make navigation simpler.

So if our site’s domain was www.example.org and we want to add routing to “www.example.org/hello”, we would use “/hello”.

Q 8. What is Template Inheritance in Flask?

Answer: Template Inheritance is a powerful feature of Flask’s Jinja templating has a great feature called template inheritance. Jinja is a Python programming language web template engine. We’ve noticed that a website’s web pages all have the same footer, navigation bar, and other elements. Instead of creating the identical footer and navigation bar on each webpage separately, we utilize template inheritance, which allows us to generate the part that is common to all webpages (e.g. footer, navigation bar) only once and eliminates the need to write the HTML, head, and title tag many times.

Q 9. What does url_for do in Flask?

Answer: The url_for() method is used to generate a URL to a specific function dynamically. After the first argument, which is the name of the selected function, we can send any number of keyword arguments matching the variable portion of the URL. This function is useful since it allows us to create URLs dynamically instead of hard-coding them into the templates.

```
<a href=”{{ url_for(‘get_post_id’, post_id=post.id}}”>{{post.title}}<a>
```

View function for handling variables in routes.

```
@app.route(“/blog/post/<string:post_id>”)

def get_post_id(post_id):

   return post_id
```

Q 10. How do you handle cookies in a Flask?

Answer: The set_cookie() method on the response object in Flask is used to set cookies. The view function’s make response() method can be used to construct the response object. On the client’s PC, cookies are stored as text files. Cookies are used to track a user’s online actions and to provide recommendations based on the user’s preferences in order to improve the user’s online experience. Cookies are stored on the client’s machine by the server and are associated with the client’s request to that server in all subsequent transactions until the cookie’s lifetime expires or the cookie is erased by the server’s specific web page.

Q 11. How does file uploading work in Flask?

Answer: The process of sending binary or regular files to a server is known as file uploading. Flask makes it simple for us to upload files. All we need is an HTML form with multipart/form-data encryption enabled. The request.files[] Object is used by the server-side flask script to get the file from the request object. The file is saved to the chosen location on the server when it has been successfully uploaded. You can get the name of the target file by doing the following.

request.files['file'] = name.filename

Q 12. What is Flask-WTF, and what are its characteristics?

Answer: WTF, also known as WT Forms in Flask, is a type of interactive user interface. The WTF is a flask built-in module that lets you build forms in a different way in flask web apps. Flask-WTF is designed to be simple to connect with WTForms, and it works well with Flask-WTF. Flask WTF includes the following features:

Integration with web forms is available.

It comes with a CSRF token, it’s an extremely secure form.

CSRF protection on a global scale

Comes with the ability to integrate internationalization.

There’s also a Supporting Captcha

This module has a file uploader that works with Flask Uploads.

Q 13. How long can an identifier be in Flask Python?

Answer: In Flask Python, An identifier can be as long as you want, as python is case-sensitive so it will treat upper case and lower case letters differently. Also, there are some words that are reserved keywords that can’t be used by users. Some of them are listed below:

def, false, import, not, true, as, del, finally, in, or, try, assert, elseif, for, is, pass, while, break, else, from, lambda, print, with, class, except, global, none, raise, yield, continue, exec, if, nonlocal, return

There are also some standards that users must follow while naming an identifier. It should start with a character, underscore, or a letter from A to Z or a-z, and the remaining characters in the identifier’s name can be any of the following: A-Z or a-z, 0-9, or.

Q 14. What HTTP methods does Python Flask provide?

Answer: To handle HTTP requests, Flask uses a number of decorators. The HTTP protocol is the backbone of internet data communication. This HTTP protocol defines a number of techniques for obtaining information from a particular URL. The different HTTP methods are:

Request	Purpose
GET	The most widely used approach. The server responds with data after receiving a GET message.
POST	To submit HTML form data to the server, use this method. The server does not save the data supplied via the POST method.
PUT	Upload content to replace all current representations of the target resource.
DELETE	Deletes all current representations of the URL’s target resource.
HEAD	Retrieves the headers for a resource, without retrieving the resource itself.

Q 15. In Flask, what do you mean by template engines?

Answer: Template engines are used when we want to build web applications that are split into different components. It is used for server-side applications that are not created as APIs and run on a single server. Templates also make it possible to render the server-side data that must be provided to the application quickly, such as the body, navigation, footer, dashboard, and so on.

Ejs, Jade, Pug, Mustache, HandlebarsJS, Jinja2, and Blade are some popular template engines.

Q 16. What is the use of jsonify() in Flask?

Answer: Jsonify is one of the flask.json module’s functions. It converts data to JSON and encapsulates it in a response object with the mime-type application/JSON. It loads directly from the flask module instead of the flask itself. To put it another way, jsonify() is a Flask helper method for correctly returning JSON data. The application/JSON mime-type is set by jsonify(), whereas json.dumps() just deliver a JSON data string. This could have unanticipated ramifications.

The jsonify() function is useful in Flask apps because it automatically sets the correct response headers and content type for JSON responses, and allows you to easily return JSON-formatted data from your route handlers. This makes it easier and more convenient to create APIs that return JSON data.

Q 17. Explain How You Can Access Sessions In Flask?

Answer: The duration between when a client signs in and logs off of a server is referred to as a session. Flask session is a flask utility that gives server-side support for sessions in the flask application developed. It is a plugin that gives your application server-side session capability. The data that must be saved in the session is saved in a temporary directory on the server. When we need to save a significant amount of data between queries in Flask, we can use session objects.

```
from flask import Flask, render_template, redirect, request, session
# The Session instance is not used for direct access, you should always use flask.session
from flask_session import Session
app = Flask(__name__)
app.config["SESSION_PERMANENT"] = False
app.config["SESSION_TYPE"] = "filesystem"
Session(app)
@app.route("/login", methods=["POST", "GET"])
def login():
    if request.method == "POST":
        session["name"] = request.form.get("name")
        return redirect("/")
    return render_template("login.html")
@app.route("/logout")
def logout():
    session["name"] = None
    return redirect("/")
```

Q 18. Explain how one can one-request database connections in Flask?

Answer: Creating and closing database connections all the time is very inefficient, Because database connections encapsulate a transaction, you must ensure that the connection is only used by one request at a time. The Flask framework allows its users to request databases in three ways. They are:

before_request(): No parameters are given when these connections are invoked before making a request.

after_request(): After initiating a request, these connections are called, and a response is sent to the client.

teardown_request(): This decorator is called when an exception is raised or everything went well (the error parameter will be None).

Q 19. What is the g object? What distinguishes it from the session object?

Answer: g is a global namespace that can hold any data for a single app context. For example, a prior request handler might set g.user, which the route and other functions can access. In the flask, on the other hand, the session data is tracked using a session object, which is a dictionary object that has a key-value pair of the session variables and their associated values. We can save data for a particular browser using the session. The session data is carried over when a user of our Flask app performs more queries in the same browser.

Q 20. Mention how one can enable debugging in Flask Python?

Answer: When Debug is turned on, any changes to the application code are updated immediately in the development stage, eliminating the need to restart the server.

By setting the flag on the applications object

Bypassing the flag as a parameter to run. If the user enables debug support, the server will reload it when the code will change and the user doesn't have to restart after each change made in the code.

```
#Method 1
app.debug = True


#Method 2
app.run('host' = localhost, debug = True)
```

Q 21. What do you mean by the Thread-Local object in Flask Python?

Answer: A thread-local object is one that is connected to the current thread id and saved in a specialized structure. Internally, Flask Python uses thread-local objects so that the user does not have to transmit objects from one function to the next within a request to stay thread safe.

Q 22. How is memory managed in Flask Python?

Answer: In a flask, Memory allocation is managed by the Flask Python memory management. Also, It has an inbuilt garbage collector which recycles all unused memory to save up heap space. The Python interpreter's responsibility is to keep track of everything. Users can, however, use the core API to access some of the tools.

Q 23. What type of Applications can we create with Flask?

Answer: We may develop nearly any form of web application by utilizing Flask. It is so versatile and adaptable that it may be rapidly merged with other technologies. Flask, for example, can be used in concert with NodeJS serverless, AWS lambda, and other third-party services to create cutting-edge systems. We can also build Single Page Apps, RESTful API-based Apps, SAS Apps, Small to Medium Websites, Static Websites, Machine Learning Applications, Microservices, and Serverless Apps.

Q 24. How to create a RESTful application in Flask?

Answer: Flask Restful is a Flask plugin that allows you to create REST APIs in Python using Flask as the backend. To create a REST API, we have to do the following steps:

Import the modules and start up the program.

Creating the REST API endpoints

Define the request methods

Implement the endpoint handlers

Serialize Data

Error Handling

Test the endpoints using various tools like Postman

Q 25. What Is Flask Sijax?

Answer: Sijax is a Python/jQuery library that makes AJAX easy to use in web applications to your Flask applications.Flask Sijax also provides an easy way to send JSON data between the server and the client.

To install we can use the following command

```
pip install flask-sijax
```

Q 26. Why is Flask called a Microframework?

Answer: Flask is termed “micro” because its main feature set is relatively limited: routing, request processing, and blueprint modules are all there is to it. Many capabilities, such as ORM, caching, and authentication, were available as optional extensions, but competing frameworks (such as Django) included them by default. The “small core + extensions” design makes it a “micro-” framework that is much easier to get started with and scale up.

Q 27. How to get a visitor IP address in Flask?

Answer: To get the visitor IP address in Flask we use method request.remote_addr Below is the implementation of it:

```
from flask import Flask, request

app = Flask(__name__)
```

```
@app.route('/')
def get_visitor_ip():
    visitor_ip = request.remote_addr
    return f"Visitor's IP address is: {visitor_ip}"


if __name__ == '__main__':
    app.run(debug=True)
```

Q 28. Which extension is used to connect to a database in Flask?

Answer: Extension enhances database management and interaction during development by eliminating the requirement to write raw SQL queries. PostgreSQL, SQLite, and MySQL are just a few of the RDBMSs Flask supports. The Flask-SQLAlchemy plugin is required to connect to databases.

Q 29. What is logging in to Flask?

Answer: Flask logging gives a lot of capability and flexibility to Flask application developers. It also allows developers to construct a sophisticated, event-logging system for Flask apps and includes all of the essential functions and classes. The same standardized Python logging framework is used in Flask. During logging, the Python modules can communicate and contribute.

Q 30. Explain Application Context and Request Context in Flask?

Answer:

Application Context :

The Application Context is the context in which the Flask application runs. It is created when the application starts and is destroyed when the application when us down. The application context stores the configuration and another global state of the application.

Request Context :

The Request Context is the context in which a request is processed. It is created when a request comes in and is destroyed when the request is completed. The Request Context stores information about the current request, such as the request method, URL, headers, and form data.

Q 31: What is Flask-SocketIO?

Answer: Flask-SocketIO is a Flask extension that provides real-time communication between clients and servers using WebSockets.

Q 32. What is Flask-Bcrypt?

Answer: Flask-Bcrypt is a Flask extension that provides password hashing and verification functionality for Flask applications.

Q33. What is Flask-JWT?

Answer: Flask-JWT is a Flask extension that provides JSON Web Token (JWT) authentication and authorization functionality for Flask applications.

Q 34. What is Flask-Assets?

Answer: Flask-Assets is a Flask extension that provides tools for managing and compiling static assets like CSS and JavaScript files.

Q 35. What is Flask-Migrate?

Answer: Flask-Migrate is a Flask extension that provides database migration functionality for Flask applications.

Q 36. What is Flask-Admin?

Answer: Flask-Admin is a Flask extension that provides a simple interface for building administrative interfaces for Flask applications. It allows you to quickly and easily create CRUD (Create, Read, Update, Delete) interfaces for your application’s models and data.

Q 37. What is Flask-SQLAlchemy?

Answer: Flask-SQLAlchemy is a Flask extension that provides an easy-to-use interface for working with SQL databases in Flask applications.

Q 38. How do you handle errors in Flask?

Answer: You can handle errors in Flask by using Flask’s error-handling functionality. This allows you to define custom error pages and handlers for different types of errors.

Q 39. What is a Flask blueprint?

Answer: A Flask blueprint is a way to organize your Flask application into smaller, modular components. Blueprints can define routes, templates, and static files, and can be registered with an application to create a larger, more complex application.

Overall, a successful Flask interview will test your knowledge of Flask and its associated technologies, as well as your ability to solve problems and work collaboratively. By reviewing common Flask interview questions and answers, you'll be better prepared to demonstrate your expertise in this popular Python web framework.

Q 40: What is Flask-RESTful?

Answer: Flask-RESTful is a Flask extension that provides tools for building RESTful APIs in Flask applications.

Flutter interview questions and answers

1.Why is Flutter preferred over other mobile app developing tools?

Flutter is a free and open-source mobile UI framework, introduced by Google in May 2017. Flutter enables you to develop a native mobile app using only one codebase. This implies you may design two separate apps using the same programming language and codebase (for iOS and Android).

Flutter is preferred over other mobile app development tools like Java and React Native due to a myriad of reasons. A few of them are:

Flutter supports cross-platform development

Programming in Flutter is extremely easy and flexible

The building process in Flutter is much faster than that in all its competitors

2.What are packages and plugins in Flutter?

A package is a set of classes, interfaces, and sub-packages that allow users to create modular code that can be shared easily. Instead of starting from scratch, using packages can help construct applications fast. In Flutter, you may use a package to add new widgets or functionality to an app.

On the other hand, a plugin is a piece of software that enhances your app's functionality. Plugins play a crucial role in the Flutter ecosystem.

3.What are the limitations of Flutter?

Since Flutter is still a relatively new tool, it has some limitations. They are as follows:

The number of third-party libraries is very small

The release size of Flutter is larger than expected

Flutter requires to be used along with another OOP language, Dart which cannot compete with other OOP languages such as Java, C#

Due to Flutter's limited complexity, mobile ad platforms do not support it

Flutter is still not used by a wide audience

4.Why does it generally take a long time to develop a Flutter app?

Because Flutter generates a device-specific IPA or APK file, building a Flutter application takes substantially longer the first time. This method, which normally takes a long time, uses Xcode and Gradle to build a file.

5.What are keys and how do you use them?

A set of IDs for Widgets, Elements and SemanticsNodes make up the key class. Keys are responsible for preserving the state of widgets when they are updated inside the widget tree. Keys may also be used to modify and rearrange collections of widgets of the same type and defined states.

Otherwise, keys may be superfluous to the code, even if they aren't harmful in any manner. Keys are useful when you need to edit a widget tree with stateful widgets, but not when the tree is entirely made up of stateless widgets.

6.What are the different types of streams in Dart?

Streams provide an asynchronous sequence of data. Asynchronous programming uses the idea of streams. They refer to a program's asynchronous succession of data occurrences. We put a value on one end and a listener on the other, similar to a pipe. Several listeners can be combined into a single stream, and when they are placed in the pipeline, they will all get the same data. It is possible to utilize the SteamController to establish new streams or manage existing ones. There are two types of streams:

Single Subscription Streams These streams convey events in chronological order. They're thought of as individual sequences inside a greater totality. When the sequence in which events are received matters, such as when reading a file, these streams are employed. Throughout the sequence, there can only be one listener, and the event will not be triggered if there isn't one.

Broadcast Streams These broadcasts provide subscribers with information about upcoming events. Subscribers can instantly begin listening to events after subscribing to them. These are flexible streams that allow several listeners to listen at the same time. Furthermore, even after canceling a previous membership, one may listen again.

7.What is pubspec.yaml file?

While creating a Flutter project, a particular type of file is always included at the top of the project. This file is known as the pubspec.yaml file, also called ‘pubspec’. This file contains information about a project's dependencies, such as packages and their versions, typefaces, and so on. It ensures that the package version is the same the next time you create the project. You can also place restrictions on the app. This project's configuration file will be used a lot while working with the Flutter project. This specification is written in YAML, a human-readable markup language.

8.What is Flutter?

Flutter is an open source software development kit (SDK) UI used for cross-platform applications from a single codebase. Flutter is primarily used for creating natively compiled, multi-platform applications across iOS and Android seamlessly.

9.Flutter is developed by which company?

Google

10.What was the first version of Flutter which ran on an Android operating system?

Sky is the first version of Flutter which ran on Android OS. It was announced at the 2015 Dart developer summit with the stated goal of being able to render consistently at 120 frames per second.

11.Can you tell us the four main elements of Flutter?

The four main elements of Flutter are:

Flutter engine

Widgets

Design-specific widgets

Foundation Library

12.When was the first version of Flutter released?

The first version of Flutter was released in May 2017, but it was first announced in 2015. In December 2018, Google released Flutter 1.0. In December 2019, Flutter 1.12 was released.

13.Flutter is written in which language?

Flutter was written in the Dart language. While writing and debugging an application, Flutter runs in the Dart virtual machine, which features a just-in-time execution engine.

14.What does the acronym SDK stand for?

Software Development Kit is a set of software tools and programs provided by software and hardware vendors that developers can use to develop applications for specific platforms. An SDK helps developers easily integrate their apps with a vendor's services. SDKs can include APIs(Application Programming Interfaces), sample code, documentation, and other resources that help developers create software applications.

15.Can you name some best editors for Flutter development?

Some popular IDEs (Integrated Development Environment) for Flutter include the following:

Android Studio

IntelliJ Idea

Emac

Visual Studio

Codemagic

16.Which operator is used to evaluate and return values between two expressions?

This “??” operator is used to evaluate and return values between two expressions. This operator first checks the expression 1 and, if it is non-null, returns its value; otherwise, it will evaluate and return the value of expression 2.

17.Can you tell us how many kinds of widgets there are in Flutter?

There are two main types of widgets in Flutter. These include:

StatelessWidget- It does not have any state information. It is static throughout its lifecycle. Examples are Row, Text, Column, and Container.

StatefulWidget- It has state information. It contains two classes: the state object and the Widget. It is dynamic because it can change the inner data during the Widget's lifetime. Examples are Radio, Form, Checkbox, and TextField.

18.Can you name the command used to compile the release mode?

This command “Flutter run -- release” is used to compile the release mode. Release mode for a web app means that the app is compiled with the dart2js compiler for best performance.

19.Name the different types of build modes in Flutter.

There are three types of build modes in Flutter. These include:

Debug- It is used to test the apps. On Android Studio, you can find a green play button on the top panel. “Flutter run”

Profile- In this mode, some debugging ability is maintained - enough to profile your app's performance, and also, it has the performance as the release mode. “Flutter run -- profile”

Release- It is used for deploying the app on marketplaces. “Flutter run - - release”

20.Which widget in Flutter is a box that comes with a different size?

SizedBox is a widget in Flutter that allows us to specify an arbitrary size for a widget. It is basically used to add empty space between widgets. SizedBox also supports specifying a specific aspect ratio for its child.

21.Suppose you have to represent real world behavior in Flutter, which animation would you use?

The choice of animation in Flutter would depend on the specific behavior that needs to be represented. Flutter provides several animation options that can be used to represent real-world behavior. These include Physics-based animation, Tween animation, Curved animation, and Hero animation.

22.Name some popular apps that use Flutter.

Some of the most popular apps that employ Flutter include:

Google Ads

Hamilton

KlasterMe

Reflectly

23.Which folder is used to write Android apps in Flutter?

Android Folder is used to write Android apps in Flutter. The Android Folder contains files and folders required for running the application, and these files are autogenerated during the creation of a Flutter project.

24.What is the use of the Await function?

The Await function is an asynchronous function. Its job is to wait until it gets the final value. Await is used with async functions and that run asynchronously, giving users the choice to wait for the asynchronous mode to finish before continuing.

25.Can you tell us which function compiles and updates the app?

In Flutter, there are several functions and commands that can be used to compile and update the app. However, the most commonly used function for this purpose is hot reload.

hot reload is a development feature in Flutter that allows you to update your app's code in real-time, without restarting the entire app.

26.Can you tell which function is accountable for starting the program?

main () function is used to start a program. This function is highly critical as, without it, one cannot write any function.

27.What is the purpose of animation controller class in Flutter?

The Animation Controller class in Flutter is used to control and manage animations. It provides a way to start, stop, or pause an animation, and allows you to define the duration, speed, and direction of an animation.

The primary purpose of the Animation Controller class is to define an animation that can be used in your app's user interface. The class provides methods for defining and manipulating animations, such as setting the duration and the curve used to control the animation's progress.

28.How can you test a single Widget?

Using the Widget tests technique, you can make sure that various portions of the user interface work as intended without the need for a physical device or simulator. This technique is the best way to isolate small parts of your app and find out whether your code is behaving as expected

29.Can you use WidgetsApp for basic navigation?

Yes, a material app widget builds a navigator, which manages a stack of widgets identified by strings, also known as routes objects and gives you two ways for managing the stack.

30.Which widget allows us to refresh the screen?

RefreshIndicator Widget enables us to refresh the screen. When the user pulls down on the widget, the onRefresh callback is triggered, which typically involves fetching new data from a server or updating the UI in some way.

31.Can you state a few examples of stateless widget?

Some examples of a stateless widget include the following:

Text - This displays a string of text with a single style.

Container: This widget can contain other widgets and provide padding, margins, and other layout properties.

Icon - This is used for the list of available material icons that can be used with this class.

32.Explain the term “Tree shaking” in Flutter.

Tree shaking is a method of removing the unused module in the bundle during the development process. Tree shaking serves as a sort of optimization technique that optimizes the code by removing the dead code.

While importing or exporting codes, there might be dead codes hanging around. Removing these dead codes reduces the code size which in turn improves the performance of the application.

33.What is used to import packages for your project?

pubspec.yaml file is used to import packages in the Dart code. It allows us to set the constraints for the application. This file contains project dependencies, general project settings, and project assets.

34.What’s the use of Navigation.push in Flutter?

The role of Navigation.push in Flutter is to add a route to a bundle of other routes which are managed by the navigator.

35.Why HTTP package is used in Flutter?

HTTP package is used in Flutter for making HTTP requests to web servers. It allows Flutter developers to send HTTP requests and receive HTTP responses from APIs or web servers. Flutter apps need to communicate with APIs or web servers to fetch data or send data to the server. The HTTP package provides a convenient and easy-to-use way to make HTTP requests in Flutter.

36.Explain profile mode in Flutter?

The profile mode in Flutter is used to test the functionality of an app while launching it. It compiles and launches your app almost identically to release mode but with additional functionality to allow debugging performance problems.

37.What's the role of BuildContext in Flutter?

In Flutter, the BuildContext is an object that provides access to the location of a widget in the widget tree hierarchy and to various services such as Theme, MediaQuery, and Navigator. The BuildContext is used by widgets to access the properties of their parent widget, such as its size, position, and theme. It is is also used to navigate between screens using the Navigator widget.

38.Can you tell us which class is responsible for implementing the basic material design visual layout structure for an app in Flutter?

The scaffold class is a widget in Flutter that is used to implement the basic material design visual layout structure. The scaffold class makes it faster to create a general-purpose mobile application. Moreover, it contains almost everything we need to create a functional and responsive Flutter application

39.Can you tell us what will be the default return type, if you don’t specify the return type for a function?

In Flutter, Dart is the programming language used to develop the framework. In Dart, if you don't specify the return type for a function, the default return type will be dynamic. This means, if you declare a function in Flutter without specifying a return type, Dart assumes that the function can return any type of value and assigns the dynamic type to the return value by default.

40.What’s the name of a constructor for a class in Flutter?

The name of a constructor for a class is the same as that of the class itself. Constructor is a special strategy that is used to create objects with predetermined values for the properties. There are three types of constructors in Flutter: named constructor, standard constructor, and factory constructor.

41.What’s the core of the Flutter layout mechanism?

The main core of the Flutter layout mechanism is the Widgets. The widgets are the building blocks of the user interface, and they are arranged in a tree-like structure known as the widget tree. The image, text, icon, and even the layout of your application are all widgets.

42.Which widget can be used to display one after another in scroll format?

The ListView Widget is used to display a large number of items as a scrollable list. ListView is an advanced version of the Column widget that automatically provides scrolling when the list of items won’t fit on the screen.

43.What is a layout in Flutter?

A layout in Flutter refers to how widgets are arranged on the screen. The layout is determined by the constraints passed down from the parent widget, and it determines the position and size of the child widget.

44.Can you tell us how we can test a single widget in Flutter?

Widget tests allow building and interacting with widgets in a test environment. It is used to check whether the Widget works as expected or not. Hence, using this testing UI component, you can test a single widget.

45.What’s the latest version of Flutter?

3.7 is the latest version of Flutter. Along with an improved framework, this new version also has great new features like enhanced material 3 support, cascading menus and menu bars,impeller preview, custom context menus, and DevTools updates among others .

46.Is Flutter a front-end framework or back-end?

Flutter is primarily a front-end framework. It provides tools and widgets to build beautiful and engaging user interfaces, and it supports various platforms such as iOS, Android, and the web. However, the framework does provide some back-end functionalities. Flutter applications can communicate with web APIs or connect to backend services through third-party packages or plugins.

47.What's the function of the future in Dart?

A future function is used to predict a potential error or value that may occur in the future. It can be completed with a value or an error. Here, developers can plug call backs for each case. Some programming languages use “promise” for the same function.

48.What's the role of image.network() constructor in Flutter?

The Image.network() constructor is a widget in Flutter that allows you to display images from the network in your application. It is used to load and display images from a URL. Example: Image.network('https://piccol.photos/259?image=8','')

49.Can you tell us what you know about Rune In Dart?

In dart programming language strings are a sequence of UTF-16 which is a 16 bit unicode Transformation Format. Rune refers to an integer that is used to describe any Unicode code point.

50.Is Flutter free?

Yes. It is a free and open source mobile UI framework created by Google. Here, with only one codebase, you can create native mobile applications.

51.Why do we use const keyword in Flutter?

We use the keyword to declare constants. Basically, when we know the value of a variable at compile time. This means the compiler knows in advance what value to store .

52.What are some operators commonly used in Dart?

Here are some of the operators that are commonly used to build apps in Dart:

= (assignment operator): Used to assign a value to a variable. ?? (null-aware operator): Used to check whether a variable is null or not and provide a default value if it is null. ! (bang operator): Used to assert that an expression is not null. . (dot operator): Used to access properties and methods of an object.

53.Why do we use a ticker in Flutter?

Tickers are used in Flutter as they provide a secure way of updating the UI at 60 frames per second. It is a class that listens to the frameCallback and calls a tick function that forwards the elapsed duration between the current frame and last frame to the ticker listener.

54.How can you reduce execution time for running code?

It will depend on the size and adaptability of the app packages. According to which we can integrate just-in-time compilers to run code which will enhance the app's performance. Dart code execution is faster than many other languages where Ahead-of-time compilers also shorten execution times. And, for merging sequential streams, integrating these compilers helps in run-time reduction and functionality increase in a variety of applications.

55.What is Dart?

Dart is a general-purpose, object-oriented programming language that was initially developed by Google in 2011. It is designed to be fast, efficient, and scalable, making it ideal for building a wide range of applications, including web and mobile apps, server-side applications, and command-line tools.

56.What process will you use to reduce widget rebuild?

On rebuilding, the state of the widget changes. This, however, helps the user to see the UI reflect state changes. At this point recreating sections of the user interface that do not need to be changed is unnecessary.

To avoid the needless rebuilding of the widgets one can divide the widget tree into small individual widgets each having its own build process. Here const constructor can be used to inform Flutter that the dont need to be rebuilt.

1.Tell me about Flutter Provider.

Widgets are used to build the provider. Inside the provider, one can use all the objects as if they are a part of it with the new widget subclasses it creates. Primarily, it works on the concept of PUB-SUB which means one provider with many subscribers.

2.What are the advantages of using Flutter?

Flutter has many advantages:

Fast development: With the help of Flutter, you can save a lot of time while developing software. Hot reload is a flutter feature with which you can make changes to your code and see the results in real time.

Flexible user interface: Flutter provides your application with a sleek, beautiful, and modern look that is sure to impress your users.

Provides native performance: Flutter applications run faster and smoother on devices.

Widgets: When building your application in Flutter, you get access to a wide range of pre-loaded widgets that can be utilized in your app.

Huge community support: If you want to reach a global audience, then Flutter is for you because it supports internationalization. With it, you can create an application that is available in multiple languages.

3.What is the purpose of the Navigator in Flutter and how is it used?

Navigator is a widget in Flutter that manages a stack of screens or "routes" in an application. It is used to navigate between different screens and to manage the state of the navigation stack. Navigator is typically used in conjunction with MaterialApp or CupertinoApp to create a navigation hierarchy for an application.

4.Can you tell us some examples of App State?

Some examples of App State include:

Login info

Notifications in a social networking app

User preferences

5.Explain Streams in Flutter?

Future and Stream are classes in Dart that define asynchronous programming. It behaves asynchronously as it doesn't return the next event when requested. Instead, it informs when the next event will be returned when ready.

6.What is the lifecycle of a StatefulWidget?

A StatefulWidget’s lifecycle is as follows:

createState

initState

didChangeDependencies

build

didUpdateWidget

setState

deactivate

dispose

7.Define the term Cookbook.

A cookbook provides specific programming solutions for some common problems experienced by developers during the development process. Each recipe can be used independently during the development process.

8.What do you mean by Container class?

The container class is the convenience widget that enables positioning, sizing, and painting of widgets. A container class can include multiple widgets and it enables developers to manage those widgets according to their convenience.

9.What is Flutter Architecture?

Flutter architecture is comprised of three layers:

Flutter Framework Layer: This is the highest layer in the Flutter architecture and includes the widgets and the Material Design widgets. It also includes APIs for animations, gestures, and routing.

Engine Layer: This is the core layer of the Flutter architecture that contains the rendering engine, which handles graphics and text rendering. It also includes the Skia graphics library, used to draw graphics on the screen.

Platform Layer: This is the lowest layer in the Flutter architecture and provides access to the native platform APIs. It includes plugins and packages that allow Flutter to interact with the device's hardware and software, such as the camera, location services, and sensors.

10.Can you give some reasons why Flutter is better than its alternatives?

Flutter has several advantages over its alternatives. Here are some of the main reasons why Flutter is a great choice for building mobile apps:

Flutter offers a hot-reload feature that enables developers to see changes in their code instantly without having to wait for the app to recompile.

Flutter allows you to build apps that work seamlessly on multiple platforms, including Android, iOS, and the web.

Flutter offers a rich set of customizable widgets, allowing developers to create beautiful and responsive designs that adapt to different screen sizes and orientations.

Flutter uses a compiled programming language, Dart, which is optimized for performance and runs natively on the device.

11.Explain how you will deploy a Flutter app to the Google Play/App Store?

For deploying Flutter apps to the Google Play Store or App Store, you will need to know the guidelines suggested by each. In both cases, it's important to make sure your app meets the requirements and guidelines of the respective app store before submitting it for review. This includes things like app content, user data privacy, user interface, and more.

Deploying to Google Play Store:

Create a Google Play Developer account

Create a signing key

Configure your app

Submit your app for review

Publish your app

Deploying to App Store:

Create an Apple Developer account

Create a signing certificate and provisioning profile

Configure your app

Submit your app for review

Publish your app

12.Explain the Primary Axis and Cross Axis Alignment.

An alignment defines how a row or a column is aligned with respect to the horizontal or vertical axis. For example, for a row its main axis is horizontal and so its cross axis will be vertical.

PrimaryAxisAlignment: the rows run horizontally, and the columns run vertically.

CrossAxisAlignment: the rows run vertically, and the columns run horizontally.

13.Can you tell us a way to access screen size in the future?

You can access the screen size using the MediaQuery class. The MediaQuery class provides information about the current device's screen size and orientation.

14.What's the difference between container and sizedBoxe?

Container and SizedBox are both widgets in Flutter that can be used to size and position child widgets, but they have some differences.

The Container widget is a multi-purpose widget that can be used to apply decoration, padding, margin, and constraints to a child widget. The Container widget can also be used as a layout widget to position its child widget within its bounds.

The SizedBox widget, on the other hand, is a simple widget that has a fixed width and height.The SizedBox widget is useful when you need to add fixed-size gaps in between widgets or to set a minimum or maximum size for a widget.

15.When do you use release mode and what's the function of release mode?

A release mode is used when one needs to optimize and produce the codes without any debug data. The release mode is used for deploying the app. The command used to compile the release mode is Flutter run --release

16.Can you name some popular database packages in Flutter?

Here are two widely used packages in Flutter:

Firebase: It is used to use or modify the cloud database

Sqflite: It is used to modify the SQLite database.

17.What are the advantages of a Flutter Inspector?

The advantages of Flutter inspector are as follows:

Widget mode: With this button, you can choose different widgets from your application and inspect them one at a time. After choosing any Widget from the Widget Tree you can view the blueprint of that Widget with the help of the Layout Explorer tab, or you can check different properties and nested widgets of the selected Widget with the help of the Details Tree tab.

Refresh Tree: Whenever you make any changes in your application, and you hot reload those changes, they are not immediately reflected in your Flutter Inspector tool. To see the changes, you will have to click the Refresh Tree button.

Slow animation: You can reduce the speed of animation between layouts.

Invert Oversized Image: This will help to find oversized images. If developers want to create an application and they want to run it as fast as possible, but with heavy size and high-resolution images, it affects the performance of the application, so it will find the image and reduce the size of that image.

Debug paint- forms a border around each Widget.

18.Define factory constructors.

A factory constructor invokes another constructor, which is allowed to return an instance of a derived class, null or an existing instance. Factory constructors are defined using the factory keyword.

19.List the responsibilities of FlutterActivity.

The roles of a FlutterActivity are:

Configuring the appearance of the status bar

Displaying Android launch screen

Displaying flutter splash screen

Choosing an execution path for the Dart app bundle

Adding transparency, which is an optional responsibility.

Saving and restoring instance state.

20.Name the different types of grid view in Flutter.

The different types of grid views in Flutter are:

count()

custom()

builder()

extent()

21.How would you implement a custom transition between screens in Flutter?

You can use the PageRouteBuilder class to create custom transitions between screens. This class allows you to define custom animations, such as sliding or fading, for the page transition.

22.Can you explain how to use the Provider package for state management in Flutter?

The Provider package is a state management library that allows you to share data between widgets efficiently. You can use the Provider package to create a data model, expose it using a provider, and then access it in your widgets using the Provider.of method.

23.What is the difference between Expanded and Flexible?

Flexible takes the least space needed to fit in a child widget. On the other hand, expanded takes the rest of the size in the widget.

24.Can you tell us some differences between const and final in Flutter?

The only difference between const and final is that the const variables are evaluated at compile-time and are immutable whereas final variables are evaluated at runtime and can only be set once.

25.How would you optimize the performance of a Flutter app?

Some techniques to optimize the performance of a Flutter app include using the const keyword to make widgets immutable, avoiding unnecessary widget rebuilds, using the Provider package for efficient state management, and minimizing the number of expensive operations in the build method.

So, if your app is small, you can pass your data using the constructor of the widget class, but for a larger app, this is not an easy task. Unknowingly we use inherited widgets in many places while developing the flutter app.Theme.of(context), Navigator.of(context), and MediQuery.of(context).

26.Can you describe how to implement internationalization in a Flutter app?

Flutter provides the Intl package, which allows you to add support for multiple languages to your app. You can use the package to define messages in different languages, format numbers, dates, and times, and apply pluralization rules based on the language.

27.Differentiate between setState and Provider?

The setState() is used for managing the local state. Calling the setState() function notifies the framework about the change in the state of the object and that may affect the user interaction in the subtree.

Whereas, provider is a state management technique in Flutter that allows widgets to access data from a central location (i.e., a "provider"). Providers can be used to manage the application state, such as user authentication or data fetched from an API.

28.Is it possible to nest a Scaffold? Why or why not?

Yes, it is possible. Scaffold is a widget, so you can put it anywhere you want. Scaffold provides APIs like Drawer, SnackBar, BottomNavigationBar, FloatingActionButton, etc. However, it is not necessary to nest scaffolds as they can be used on their own.

29.Can you explain how to use streams and futures in Flutter? Also, tell us some similarities and differences between Future and Stream?

Streams and futures are powerful tools for handling asynchronous operations in Flutter. You can use a stream to listen for a series of events, such as user input or network responses, and react to them in real-time. A future, on the other hand, represents a value that may not be available yet and allows you to execute asynchronous code and retrieve the result when it's ready.

Similarities between future and stream:

Both work asynchronously

Both have some potential value

Differences between future and stream:

Future has one response whereas a stream can have multiple responses.

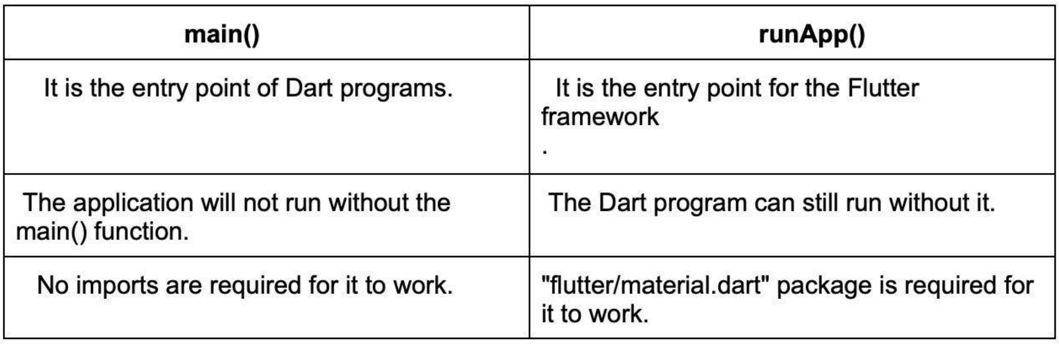
Futures are used in HTTP calls.

A Stream is a series of futures.

30.Can you tell us the difference between WidgetsApp and MaterialApp?

The main difference between WidgetsApp and MaterialApp is that the widget offers basic navigation, whereas MaterialApp is built on top of WidgetsApp.

31.Can you state some difference between runApp() and main()?



main()	runApp()
It is the entry point of Dart programs.	It is the entry point for the Flutter framework .
The application will not run without the main() function.	The Dart program can still run without it.
No imports are required for it to work.	"flutter/material.dart" package is required for it to work.

32.What’s the need of mixins?

Multiple inheritances are not supported in Dart. Hence, you would need mixins to use multiple inheritance in Flutter, as they allow you to write reusable class code in multiple class hierarchies.