Q.I What is Dart, and what is its primary use?

Dart is a programming language developed by Google, primarily used for building web and mobile applications.

Q.2 What are the key features of Dart?

Key features include strong typing, ahead-of-time (AOT) compilation, asynchronous programming, and a rich standard library.

Q.3 Name some popular frameworks and platforms that use Dart.

Dart is used in frameworks like Flutter for mobile app development and AngularDart for web applications.

Q.4 What are the benefits of using Dart for web and mobile development?

Dart offers improved performance, productivity, and a consistent codebase for both web and mobile apps.

Q.5 Explain the difference between JIT and AOT compilation in Dart.

JIT (Just-ln-Time) compilation compiles code at runtime, while AOT (Ahead-Of-Time) compilation compiles code before execution for

Q.6 What is the main data type in Dart for representing integers?

The main data type for integers in Dart is int.

Q.7 How do you define a variable in Dart?

You can define a variable using the var, dynamic, or type-specific keywords like int, double, or String.

Q.8 What is type inference in Dart?

Type inference allows Dart to automatically determine the data type of a variable based on its value.

Q.9 How do you declare a constant in Dart?

Use the const keyword to declare a constant in Dart.

Q.IO Explain the purpose of the final keyword in Dart.

The final keyword is used to declare a variable whose value cannot be changed after it is assigned a value.

Q. 11 What is the difference between var and dynamic in Dart?

var infers the type of a variable from its initial value, while dynamic allows the variable to have a changing data type.

Q. 12 How do you comment code in Dart?

Use // for single-line comments and /\* \*/ for multi-line comments in Dart.

Q. 13 What are Dart data types for representing floating-point numbers?

Dart uses double to represent floating-point numbers.

Q. 14 How do you perform type casting or type conversion in Dart?

Use type casting syntax like (int) or .toString() to convert between data types in Dart.

Q. 15 Explain the purpose of the is keyword in Dart.

The is keyword is used for type checking to determine if an object is of a specific type.

Q.16 What is the Dart String interpolation syntax?

Dart uses ${variable} within strings to interpolate the value of a variable into a string.

Q. 17 How do you create a list (array) in Dart?

Lists in Dart can be created using square brackets, e.g., List numbers

Q.18 What is the difference between a List and a Set in Dart?

A List is an ordered collection with duplicate elements, while a Set is an unordered collection with unique elements.

Q. 19 Explain the purpose of the for and for-in loops in Dart.

The for loop is used for traditional counting loops, while for-in is used for iterating over collections like lists.

Q.20 How do you define a function in Dart?

Functions in Dart are defined using the function\_name(parameters) { .. } syntax.

Q.21 What is a named parameter in Dart functions?

Named parameters allow you to pass arguments to a function by name, enhancing readability and flexibility.

Q.22 How do you create a class in Dart?

You can create a class using the class keyword followed by the class name and properties/methods.

Q.23 What is the Dart constructor, and how is it defined?

A constructor initializes an object of a class. In Dart, a constructor is defined using the same name as the class.

Q.24 How do you implement inheritance in Dart?

Inheritance is achieved by using the extends keyword to create a subclass (child class) that inherits properties and methods from a

superclass (parent class).

Q.25 Explain the difference between async and await in Dart.

async marks a function as asynchronous, while await is used within an asynchronous function to wait for a Future to complete.

Q.26 What is a Future in Dart, and how is it used for asynchronous operations?

A Future represents a potentially long-running operation, and it can be used to handle asynchronous tasks.

Q.27 How do you handle exceptions in Dart?

Dart uses try, catch, and finally blocks to handle exceptions and error conditions.

Q.28 What is the assert statement in Dart, and when is it used?

assert is used for debugging and testing. It throws an error if a specified condition is not met.

Q.29 How can you import external libraries or packages in Dart?

You can use the import keyword followed by the package name to import external libraries or packages in Dart.

Q.30 What is the purpose of the async\* keyword in Dart?

async\* is used to define asynchronous generator functions, which can yield multiple values over time.

Q.31 What is a mixin in Dart, and how is it used in class composition?

A mixin is a way to reuse a class's code in multiple class hierarchies without inheritance, using the with keyword.

Q.32 How do you work with JSON data in Dart?

Dart provides libraries like dart:convert for encoding and decoding JSON data using json.decode() and json.encode().

Q.33 What is the purpose of the typedef keyword in Dart?

typedef is used to define custom function types, which can be used to declare functions with specific signatures.

Q.34 How do you create a single-line function (arrow function) in Dart?

Single-line functions can be created using the => syntax, e.g., (int a, int b) => a + b.

Q.35 What are optional and named parameters in Dart functions?

Optional parameters can be omitted when calling a function, while named parameters are specified using names when calling the

function.

Q.36 How do you create a custom exception in Dart?

You can create a custom exception by defining a class that extends the built-in Exception class.

Q.37 What is a static method in Dart, and how is it different from an instance method?

A static method belongs to the class itself, not to instances of the class, and is accessed using the class name.

Q.38 How do you handle asynchronous errors (exceptions) in Dart?

You can use the try-catch block to handle asynchronous errors when working with Future objects.

Q.39 Explain the concept of cascading (..) in Dart.

Cascading allows you to perform multiple operations on an object without repeating the object reference.

Q.40 What is the purpose of the async/await pattern in Dart?

The async/await pattern simplifies asynchronous code by making it look more like synchronous code, improving readability.

Q.41 How do you define and use an enumeration (enum) in Dart?

Enumerations are defined using the enum keyword and can be used to represent a fixed set of values.

Q.42 What is the difference between async and async\* in Dart function declarations?

async is used for asynchronous functions that return a single value or null, while async\* is used for asynchronous generator functions

that yield multiple values.

Q.43 How do you create a custom exception class in Dart?

To create a custom exception class, define a class that extends either Exception or another exception class.

Q.44 What is the with keyword used for in Dart?

The with keyword is used to include a mixin in a class, allowing the class to inherit the behaviors defined by the mixin.

Q.45 How do you define and use a factory constructor in Dart?

A factory constructor is defined using the factory keyword and is used to create an instance of a class based on specific conditions.

Q.46 Explain the purpose of the break and continue statements in Dart loops.

break is used to exit a loop prematurely, while continue is used to skip the current iteration and proceed to the next one.

Q.47 What is the role of the await for loop in Dart?

The await for loop is used to iterate over asynchronous data streams, waiting for each data item to be available.

Q.48 How do you create a custom iterable class in Dart?

To create a custom iterable class, define a class that implements the Iterable interface and provides custom iteration logic.

Q.49 What is the difference between an abstract class and an interface in Dart?

An abstract class can have both method implementations and abstract methods, while an interface defines a contract for classes to

implement.

Q.50 How can you work with asynchronous code in Dart without using async/await?

You can use Future and the .then() method to handle asynchronous code in Dart without async/await.

Q.51 How do you perform null checking in Dart to avoid null reference errors?

Use the null-aware operator ?. to safely access properties or methods on potentially null objects.

Q.52 What is a getter and setter in Dart, and how are they defined?

Getters and setters are used to read and write object properties. They are defined using the get and set keywords.

Q.53 How do you work with asynchronous streams in Dart?

Asynchronous streams can be worked with using the Stream and StreamController classes for handling and emitting data over time.

Q.54 What is the Dart event loop, and how does it relate to asynchronous programming?

The event loop manages asynchronous operations, ensuring that tasks are executed in a non-blocking manner.

Q.55 How can you handle timeouts in Dart when working with asynchronous code?

You can use the Future.timeout() method to handle timeouts when waiting for asynchronous operations to complete.

Q.56 Explain the concept of asynchronous error handling in Dart.

Asynchronous error handling involves using try-catch blocks and onError callbacks to handle errors in asynchronous code.

Q.57 What is the purpose of the as keyword in Dart?

The as keyword is used for type casting, allowing you to treat an object as an instance of a specific type.

Q.58 How do you define a constant constructor in Dart?

A constant constructor is defined using the const keyword before the constructor's name, ensuring that instances are compile-time

constants.

Q.59 What is the async\* generator function, and how is it used?

async\* is used to create asynchronous generator functions, which can yield values over time and pause execution between yields.

Q.60 How do you handle exceptions thrown in asynchronous code in Dart?

You can use a combination of try-catch blocks and the .catchError() method to handle exceptions in asynchronous code.

Q.61 What is the purpose of the typedef keyword when defining function types in Dart?

typedef is used to define custom function types, which can make code more readable and maintainable.

Q.62 How do you work with generic types in Dart?

Dart supports generic types, allowing you to write code that works with different data types without sacrificing type safety.

Q.63 What are futures and streams, and how do they relate to asynchronous programming in Dart?

Futures represent single values that may not be available yet, while streams represent sequences of values that arrive over time.

Q.64 What is the purpose of the async modifier in Dart function declarations?

The async modifier indicates that a function is asynchronous and may pause execution to wait for asynchronous operations.

Q.65 How do you use the await for loop to process asynchronous streams in Dart?

The await for loop is used to iterate over an asynchronous stream, awaiting values as they become available.

Q.66 What is the difference between null and undefined in Dart?

In Dart, null represents the absence of a value, while undefined is not a recognized value; it typically occurs when a variable is not

initialized.

Q.67 How do you create and use an enumerated type (enum) in Dart?

You can create an enum by using the enum keyword, and it can be used to define a set of named constant values.

Q.68 What is the late keyword in Dart, and when is it used?

The late keyword is used to declare a variable that will be initialized at a later point, indicating that it won't be null.

Q.69 How do you implement method cascading in Dart?

Method cascading is achieved using the .. operator, which allows you to chain multiple method calls on the same object.

Q.70 What is a factory constructor in Dart, and why would you use it?

A factory constructor is used to create instances of a class with specific logic, such as returning cached objects or singletons.

Q.71 How do you define and use a static variable in Dart?

Static variables are defined using the static keyword and are associated with the class rather than instances of the class.

Q.72 What is the purpose of the Future.delayed function in Dart?

Future.delayed is used to create a future that completes after a specified duration, allowing you to introduce delays in asynchronous

code.

Q.73 How do you work with files and directories in Dart?

Dart provides the File and Directory classes for working with files and directories, including reading, writing, and managing file

systems.

Q.74 What is the role of the await keyword in Dart asynchronous code?

await is used to pause the execution of an asynchronous function until a Future completes and returns a value.

Q.75 What is the difference between async/await and using callbacks in Dart for asynchronous programming?

async/await provides a more sequential and readable way to handle asynchronous operations compared to callback-based

approaches.

Q.76 How do you handle concurrent asynchronous tasks in Dart?

Dart provides utilities like Future.wait and Future.forEach to manage and await the completion of multiple asynchronous tasks

concurrently.

Q.77 What is the async\* generator function used for in Dart?

async\* is used to create asynchronous generator functions that yield values over time while allowing for non-blocking operations.

Q.78 How do you use the Future.error constructor in Dart?

Future.error is used to create a future that immediately completes with an error. It can be helpful for error handling and testing.

Q.79 What is the completer class in Dart, and how is it used?

A Completer is used to manually create and complete Future objects, providing control over when a future completes.

Q.80 What are the benefits of using Dart for web and mobile development compared to other languages?

Dart offers improved performance, a rich standard library, and the Flutter framework for building cross-platform apps.

Q.81 What does the appBar widget in Flutter contains

The appBar widget in Flutter contains elevation, title and brightness

Q.82 Which OS does Dart language supports

Dart language supports Linux, macOS and Windows operating system

Q.83 How can you initialize final instance variables in Dart language

Final instance variables in Dart language can be initialized at the variable declaration, by a constructor parameter and in the

constructor's initializer list

Q.84 What does the const keyword in Dart language is used for

The const keyword in Dart language is used to declare constant variables, create constant values and declare constructors that create

constant values

Q.85 Which methods can be abstract in Dart language

The methods which can be abstract in Dart language are Instance, getter and setter

Q.86 What qualifies the usage of deferred loading in Dart language

The deferred loading in Dart language is to reduce a web app's initial startup time, to perform A/B testing—trying out alternative

implementations of an algorithm and to load rarely used functionality, such as optional screens and dialogs.

Q.87 What is valid for deferred loading in Dart language

The applicable details for deferred loading in Dart language is- a deferred library's constants aren't constants in the importing file, you

can't use types from a deferred library in the importing file and Dart language implicitly inserts loadLibrary() into the namespace that

you define using deferred as namespace.

Q.88 Which operator is valid for the int class in Dart language

The operator valid for the int class in Dart language are the \* operator, >> operator and ceil().

Q.89 What is a valid method on Stream which process the stream and return a result

The methods on Stream which process the stream and return a result are isEmpty, contains and elementAt.

Q.90 What is valid for Dart language's static methods

The Dart language's static methods are the member class instead of its object, static methods are also identifies as class methods

and we can access static methods using the class name.

Q.91 What will be printed by the following statements:

var cities = ('los angeles', 'san francisco', 'new york],•

print(cities.firstWhere((c) c.length < 1 0));

Only new york will be printed

Q.92 The Scaffold widget in Flutter, is able to occupy

The Scaffold widget in Flutter, is able to occupy the whole device screen.

Q.93 What is a horizontal bar that is mainly displayed at the top of the Scaffold widget in Flutter

The horizontal bar shown at the top of the Scaffold widget in Flutter is the appBar.

Q.94 What displays the main content in the Scaffold widget in Flutter?

body shows the main content in the Scaffold widget in Flutter

Q.95 What is a slider panel that is displayed at the side in Flutter?

The slider panel shown at the side in Flutter is the drawer

Q.96 Which Dart language tool compiles Dart language code to native x64 machine code, producing either a standalone executable

or a snapshot in Dart language SDK

Dart language2native compiles Dart language code to native x64 machine code

Q.97 What is the name of Dart language-to-JavaScript compiler (used only for web development) in Dart language SDK

Dart language2js is the Dart language-to-JavaScript compiler

Q.98 In Dart language, uninitialized variables have an initial value of?

The initial value of uninitialized variables is null

Q.99 What is the type assigned to instance variables in Dart language

The instance variables in Dart language are final.

Q.IOO When should final instance variables in Dart language be initialized?

The final instance variables in Dart language are initialized before the constructor body starts

Q. 101 What type of constant does a const variable in Dart language is?

A const variable in Dart language is compile-time constant

Q.102 What is the size in bits of integer values in Dart language?

The size in bits of integer values in Dart language is 64 bits

Q.103 Why you are suitable as Dart programmer?

As a Dart programmer, I am having extensive experience in developing both web and mobile application with requisite skills including:

communication, problem solving and coping under pressure which is of importance for Dart programmer role.

Q.104 Do you feel satisfied with your role as Dart programmer?

I feel satisfied as Dart programmer as I am able to better use my programming and logic skills for developing reliable Dart programs

quickly which helps company to manage their processes more efficiently.

Q.105 How you keep yourself updated of new trends in Dart language?

Dart language has been developed by Google and has a defined release cycle. I update myself by attending Dart language related

seminars, tutorials and certifications as available online or offline.

Q.106 What are your strengths as a Dart programmer?

As a Dart programmer I am having extensive experience on developing both mobile applications for Android and iPhone as well as

developing cross browser web applications. I also have the requisite managerial skills for managing Dart development related

projects.

<https://mindmajix.com/dart-interview-questions>

**What are the Features of Dart?**

### What are the ways to execute a Dart Program?

### What are the data types in the Dart language?

### What is type-checking in Dart?

### What is the use of typedef in Dart?

### Does Dart have a syntax for declaring interfaces?

### What are Streams in Dart?

### 14. What is a pub in Dart?

### What is the file extension of Dart?

### Does Dart Support Comment?

### What Are The Various Types Of Operators In Dart?

### Does Dart support overloading?

### What are various string functions in Dart?

### . What are the core libraries in Dart?

### . What are the supplementary packages published by Dart?

### . What are the platforms used in Dart?

### What is Lambda Function?

### What are Getters and Setters?

### . What Is Method Overriding In Dart?

### . What is a constructor in Dart?

### What is polymorphism in Dart?

### What is the Dart final?

### 31. What is a double dot in Dart?

### 32. What is encapsulation in Dart?

### 33. What is an optional parameter in Dart?

### 34. What are streams in Dart?

**36. What are the types of Inheritance in Dart?**

* **Single Inheritance:** This inheritance occurs when a class inherits from a single parent class.
* **Multiple Inheritance:** This inheritance occurs when a class inherits from more than one parent class. This is not supported by Dart.
* **Multi-Level Inheritance:** This inheritance occurs when a class inherits another child class.
* **Hierarchical Inheritance:** More than one class has the same parent class under hierarchical inheritance.

### 35. What is Null -aware operator in Dart?

### 37. What are the Snapshots in Dart?

### 38. What is Dart Strings?

<https://codingwithtashi.medium.com/dart-interview-questions-and-answers-866d5c6a5141>

. **What are the data types in Dart.**

**3. Is dart Object Oriented? If so explain.**

**4. Explain an interface in dart with an example?**

**5. What is the difference between final and const?**

**6. What is sound null safe in dart?**

**7. What is the late keyword used for?**

**8. How to run Dart application?**

**10. What “is” and “as” keywords in dart?**

**11. How do you catch an error in the Dart program?**

**12. What is a constructor? Name types of constructor in dart?**

**14. What are “async”, “await ”and “Future”?**

**2. What are the differences between Object and dynamic?**

**3. What is assert in dart?**

**4. What is Cascade notation in dart?**

**5. 6. What is “mixin”? When do we use it?**

**7. What is isolate?**

**8. Is Method Overloading possible in dart? If not, what is the alternative approach?**

1. **How to achieve “isolate ”in dart?**
2. **Implement the extension method with an example**

**4. Does dart uses VM, if so, how?**

<https://coderpad.io/interview-questions/dart-interview-questions/>

What is the purpose of using the final keyword in Dart?

What is the difference between a List and a Set in Dart?

What are some advantages of using Dart for mobile app development compared to other programming languages?

**Question:**What is a constructor in Dart? Explain with an example.

**Question:**What are named parameters in Dart? Explain with an example.

**Question:**What is the purpose of the super keyword in Dart? Provide an example.

What is the purpose of the async and await keywords in Dart? Explain with an example.

Explain the concept of “asynchronous programming” in Dart and how it helps in building efficient and responsive applications.

What is the purpose of Dart’s async/await syntax? How does it improve asynchronous programming?

https://in.indeed.com/career-advice/interviewing/dart-interview-questions

### 5. Explain polymorphism in Dart.

### 6. When can you use generics?

### 8. What is the use of @Deprecated annotation in Dart?

### 10. Explain the types of inheritance in Dart.

<https://www.adaface.com/blog/dart-interview-questions/>

[**Dart Interview Questions For Freshers**](https://www.adaface.com/blog/dart-interview-questions/#s0)

1. [What is the difference between a var, dynamic and final variable in Dart?](https://www.adaface.com/blog/dart-interview-questions/#q13)
2. [How does Dart handle null safety?](https://www.adaface.com/blog/dart-interview-questions/#q14)
3. [How do you declare and initialize a list in Dart?](https://www.adaface.com/blog/dart-interview-questions/#q15)
4. [How do you declare and initialize a map in Dart?](https://www.adaface.com/blog/dart-interview-questions/#q16)
5. [How do you create and use classes and objects in Dart?](https://www.adaface.com/blog/dart-interview-questions/#q17)
6. [What is the difference between a constructor and a factory constructor in Dart?](https://www.adaface.com/blog/dart-interview-questions/#q18)
7. [How do you use the switch statement in Dart?](https://www.adaface.com/blog/dart-interview-questions/#q19)
8. [How do you use the for loop in Dart?](https://www.adaface.com/blog/dart-interview-questions/#q20)
9. [How do you use the while and do-while loops in Dart?](https://www.adaface.com/blog/dart-interview-questions/#q21)
10. [How do you use the break and continue keywords in Dart?](https://www.adaface.com/blog/dart-interview-questions/#q22)
11. [How do you use the try-catch statement in Dart?](https://www.adaface.com/blog/dart-interview-questions/#q23)

[**Dart Intermediate Interview Questions**](https://www.adaface.com/blog/dart-interview-questions/#s1)

1. [How do you use inheritance in Dart?](https://www.adaface.com/blog/dart-interview-questions/#q24)
2. [How do you use polymorphism in Dart?](https://www.adaface.com/blog/dart-interview-questions/#q25)
3. [How do you handle exceptions in Dart?](https://www.adaface.com/blog/dart-interview-questions/#q26)
4. [How do you use the assert statement in Dart?](https://www.adaface.com/blog/dart-interview-questions/#q27)
5. [How do you use the final keyword in Dart?](https://www.adaface.com/blog/dart-interview-questions/#q28)
6. [How do you use the const keyword in Dart?](https://www.adaface.com/blog/dart-interview-questions/#q29)
7. [How do you use the new keyword in Dart?](https://www.adaface.com/blog/dart-interview-questions/#q30)
8. [How do you use the .. cascade notation in Dart?](https://www.adaface.com/blog/dart-interview-questions/#q31)
9. [How do you parse and convert JSON data in a Dart application?](https://www.adaface.com/blog/dart-interview-questions/#q32)
10. [How do you use the => fat arrow notation in Dart?](https://www.adaface.com/blog/dart-interview-questions/#q33)
11. [What is the purpose of the async and await keywords in Dart?](https://www.adaface.com/blog/dart-interview-questions/#q34)
12. [How do you create and use streams in Dart?](https://www.adaface.com/blog/dart-interview-questions/#q35)

[**Dart Interview Questions For Experienced**](https://www.adaface.com/blog/dart-interview-questions/#s2)

1. [How do you work with Futures in Dart?](https://www.adaface.com/blog/dart-interview-questions/#q36)
2. [How do you handle async and synchronous code execution in Dart?](https://www.adaface.com/blog/dart-interview-questions/#q37)
3. [How do you handle errors in asynchronous code in Dart?](https://www.adaface.com/blog/dart-interview-questions/#q38)
4. [How do you use the listen and cancel methods in Stream?](https://www.adaface.com/blog/dart-interview-questions/#q39)
5. [How do you handle real-time data streams in a Flutter application?](https://www.adaface.com/blog/dart-interview-questions/#q40)

<https://www.remoterocketship.com/advice/10-Dart-interview-questions-and-answers-in-2023>

**1. How do you debug a Dart application?**

### 2. What is the difference between a Future and a Stream in Dart?

**3. How do you handle asynchronous programming in Dart?**

**4. What is the purpose of the Isolate class in Dart?**

### 5. How do you handle errors in Dart?

### 6. What is the difference between a static and dynamic type in Dart?

### 7. How do you optimize a Dart application for performance?

### 8. What is the difference between a function and a method in Dart?

### 9. How do you handle memory management in Dart?

<https://www.acte.in/dart-interview-questions-and-answers>

<https://www.fullstack.cafe/blog/dart-interview-questions>

## ***Q4*:**

## Differentiate between required and optional parameters in Dart

## ***Q5*:**

## Explain the different types of Streams?

## ***Q21*:**

## What is a difference between these operators ?? and ?.

**Senior**

[Top 67 Flutter Interview Questions](https://www.fullstack.cafe/interview-questions/flutter" \o "Flutter Interview Questions)**[Flutter](https://www.fullstack.cafe/interview-questions/flutter" \o "Flutter Interview Questions)**[67](https://www.fullstack.cafe/interview-questions/flutter" \o "Flutter Interview Questions)

**Answer**

**??**

* It is a **null-aware operator** which returns the expression on its left unless that expression’s value is null, in which case it evaluates and returns the expression on its right:

print(1 ?? 3); // <-- Prints 1.

print(null ?? 12); // <-- Prints 12.

**?.**

* It is a **conditional property access** which is used to guard access to a property or method of an object that might be null, put a question mark (?) before the dot (.):
* You can chain multiple uses of ?. together in a single expression:

myObject?.someProperty?.someMethod()

The preceding code returns null (and never calls someMethod()) if either myObject or myObject.someProperty is null.

<https://www.firehire.ai/interview-questions/dart>

<https://cloudfoundation.com/blog/dart-interview-questions/>

fluuetr AND DART MIXED.

<https://www.besanttechnologies.com/flutter-and-dart-interview-questions-and-answershttps://www.slajobs.com/flutter-and-dart-interview-questions-and-answers/>

<https://www.mytectra.com/interview-question/dart-programming-interview-questions-and-a>

Dart oops important questions

dart opps interview questions

ChatGPT

ChatGPT

Sure, here are some object-oriented programming (OOP) interview questions tailored for Dart:

1. **What is object-oriented programming (OOP), and how does Dart support it?**
2. **Explain the pillars of object-oriented programming and how they are implemented in Dart.**
3. **What is a class in Dart, and how do you define one?**
4. **What are objects in Dart, and how are they created?**
5. **Explain the concepts of inheritance, encapsulation, and polymorphism in Dart. Provide examples of each.**
6. **What is constructor overloading, and how do you achieve it in Dart?**
7. **Describe the difference between abstract classes and interfaces in Dart. When would you use each?**
8. **What is method overriding in Dart, and how is it implemented?**
9. **Explain the concept of access modifiers in Dart.**
10. **What are static members in Dart, and how are they different from instance members?**
11. **How do you implement method overloading in Dart?**
12. **What is the purpose of the this keyword in Dart?**
13. **What is a factory constructor in Dart, and when would you use it?**
14. **Explain the concept of mixins in Dart and provide an example.**
15. **How does Dart support multiple inheritance, and what are the limitations?**
16. **What is composition, and how is it implemented in Dart?**
17. **What are getters and setters in Dart, and how do you define them?**
18. **Explain the super keyword in Dart. When and how would you use it?**
19. **What is method hiding in Dart, and how do you achieve it?**
20. **Describe the concept of method cascades in Dart and provide an example of their usage.**