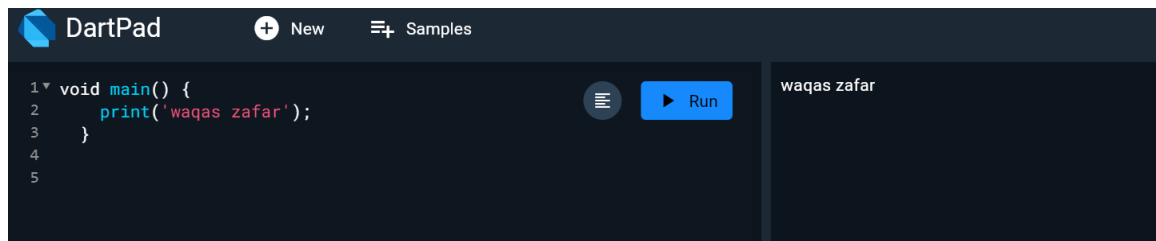


DART PROGRAMMING

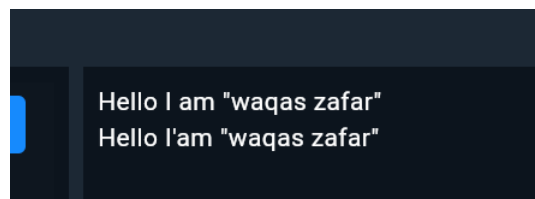
1. Write a program to print your name in Dart.

```
void main() {  
    print('waqas zafar');  
}
```



2. Write a program to print Hello I am “John Doe” and Hello I’am “John Doe” with single and double quotes.

```
void main() {  
    // Using double quotes with escape character for single quote  
    print('Hello I am "waqas zafar"');  
  
    // Using single quotes with escape character for single quote  
    print('Hello I\'am "waqas zafar"');}
```



3. Declare **constant type** of int set value 7.

```
void main() {  
    const int myConstant = 7;  
    print('The constant value is: $myConstant');  
}
```

The constant value is: 7

4. Write a program in Dart that finds simple interest. $\text{Formula} = (p * t * r) / 100$

```
void main() {  
    double principal = 1000;  
    double time = 4;  
    double rate = 5;  
    double simpleInterest = (principal * time * rate) / 100;  
    print('Simple Interest: $simpleInterest');  
}
```

Simple Interest: 200

5. Write a program to print a square of a number using user input.

```
void main() {  
    print('Enter a number: ');  
    String input = '7';  
    int number = int.parse(input);  
    int square = number * number;  
    print('The square of $number is $square');}
```

Enter a number:
The square of 7 is 49

6. Write a program to print full name of a from first name and last name using user input.

```
import 'dart:io';
```

```

void main() {
  print('Enter your first name: ');
  String firstName = stdin.readLineSync()!;

  print('Enter your last name: ');
  String lastName = stdin.readLineSync()!;

  String fullName = '$firstName $lastName';
  print('Your full name is: $fullName');
}

```

```

The Dart DevTools debugger and profiler
Enter your first name:
waqas
Enter your last name:
zafar
Your full name is: waqas zafar

Process finished with exit code 0

```

7. Write a program to find quotient and remainder of two integers.

```

import 'dart:io';

void main() {
  print('Enter the first integer: ');
  int firstNumber = int.parse(stdin.readLineSync()!);

  print('Enter the second integer: ');
  int secondNumber = int.parse(stdin.readLineSync()!);

  int quotient = firstNumber ~/ secondNumber;
  int remainder = firstNumber % secondNumber;

  print('Quotient: $quotient');
  print('Remainder: $remainder');
}

```

```

The Dart DevTools debugger and prof
Enter the first integer:
4
Enter the second integer:
7
Quotient: 0
Remainder: 4

```

8. Write a program to swap two numbers.

```
import 'dart:io';

void main() {
  print('Enter the first number: ');
  int firstNumber = int.parse(stdin.readLineSync()!);
  print('Enter the second number: ');
  int secondNumber = int.parse(stdin.readLineSync()!);
  int temp = firstNumber;
  firstNumber = secondNumber;
  secondNumber = temp;
  print('After swapping:');
  print('First number: $firstNumber');
  print('Second number: $secondNumber');
}
```

```
Enter the first number:
6
Enter the second number:
7
After swapping:
First number: 7
Second number: 6
```

9. Write a program in Dart to remove all whitespaces from String.

```
void main() {
  String text = ' Hello World ';
  String trimmedText = text.replaceAll(' ', '');

  print('Original string: $text');
  print('String after removing whitespace: $trimmedText');
}
```

10.

```
The Dart DevTools debugger and profiler is available
5 Original string: Hello World
4 String after removing whitespace: HelloWorld
0
Process finished with exit code 0
```

11. Write a Dart program to convert String to int.

```
void main() {
  String numberString = '12345';
  int number = int.parse(numberString);

  print('String: $numberString');
  print('Integer: $number');
}
```

```
[Debug service available at http://127.0.0.1:524
↑
↓ The Dart DevTools debugger and profiler is availa
String: 12345
Integer: 12345
Process finished with exit code 0
>
```

12. Suppose, you often go to restaurant with friends and you have to split amount of bill.

Write a program to calculate split amount of bill.

$\text{amount} / \text{number of people}$

Formula= (total bill

```
void main() {
  double totalBill = 100.0;
  int numberOfPeople = 4;
  double splitAmount = totalBill / numberOfPeople;

  print('Total bill amount: $totalBill');
  print('Number of people: $numberOfPeople');
  print('Split amount per person: $splitAmount');
}
```

```
The Dart DevTools debugger and profiler is a
Total bill amount: 100.0
Number of people: 4
Split amount per person: 25.0

Process finished with exit code 0
```

13. Suppose, your distance to office from home is 25 km and you travel 40 km per hour.

Write a program to calculate time taken to reach office in minutes.

Formula= $(\text{distance}) / (\text{speed})$.

```
void main() {
  double distance = 25.0;
  double speed = 40.0;

  double timeHours = distance / speed;
  double timeMinutes = timeHours * 60;

  print('Distance to office: $distance km');
  print('Speed: $speed km/h');
  print('Time taken to reach office: ${timeMinutes.toStringAsFixed(2)}
minutes');
}
```

```
The Dart DevTools debugger and profiler is available
Distance to office: 25.0 km
Speed: 40.0 km/h
Time taken to reach office: 37.50 minutes

Process finished with exit code 0
```

14. Write a dart program to check if the number is odd or even.

```
void main() {
  int number = 25;

  if (number % 2 == 0) {
    print('$number is even.');
```

```
  } else {
    print('$number is odd.');
```

15. Write a dart program to check whether a character is a vowel or consonant.

```
void main() {
  String character = 'v';

  character = character.toLowerCase();

  if (character == 'a' || character == 'e' || character == 'i' || character == 'o' || character == 'u') {
    print('$character is a vowel.');
```

```
  } else {
    print('$character is a consonant.');
```

16. Write a dart program to check whether a number is positive, negative, or zero.

```
void main() {
  double number = -5.0;

  if (number > 0) {
    print('$number is positive.');
```

```

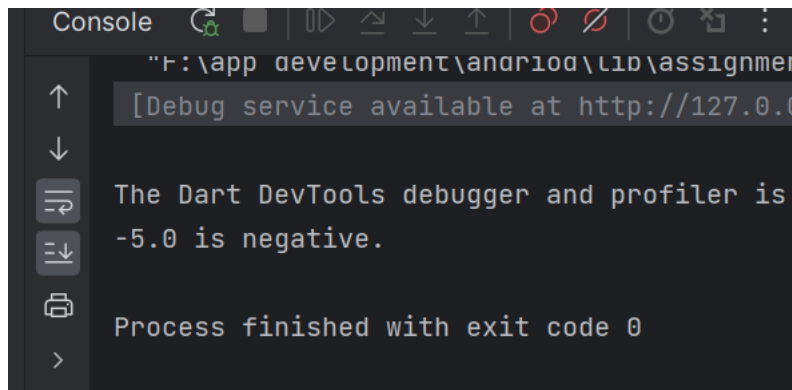
    print('$number is negative.');
```

```

  } else {
    print('$number is zero.');
```

```

  }
}
```



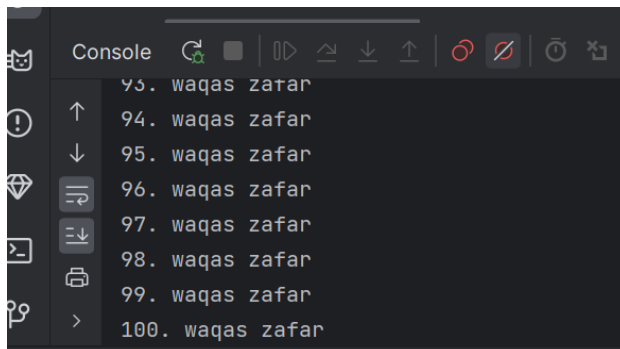
17. Write a dart program to print your name 100 times.

```

void main() {
  String name = 'waqas zafar';
  for (int i = 1; i <= 100; i++) {
    print('$i. $name');
```

```

  }
}
```



18. Write a dart program to calculate the sum of natural numbers.

```

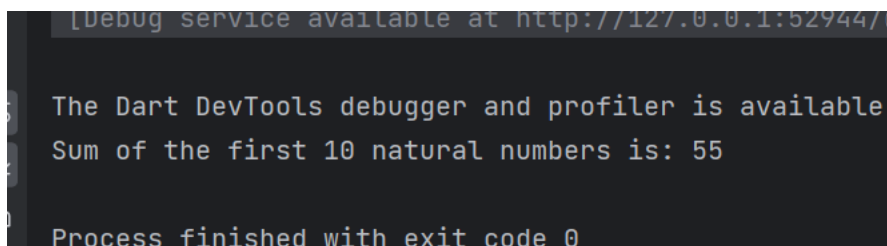
void main() {
  int n = 10;
  int sum = 0;

  for (int i = 1; i <= n; i++) {
    sum += i;
  }

  print('Sum of the first $n natural numbers is: $sum');
```

```

}
```



19. Write a dart program to generate multiplication tables of 5.

```
void main() {  
  int number = 5;  
  int limit = 10;  
  
  for (int i = 1; i <= limit; i++) {  
    print('$number x $i = ${number * i}');  
  }  
}
```

The Dart DevTools debugger and

```
5 x 1 = 5  
5 x 2 = 10  
5 x 3 = 15  
5 x 4 = 20  
5 x 5 = 25  
5 x 6 = 30  
5 x 7 = 35  
5 x 8 = 40  
5 x 9 = 45  
5 x 10 = 50
```

20. Write a dart program to generate multiplication tables of 1-9.

```
void main() {  
  for (int i = 1; i <= 9; i++) {  
    for (int j = 1; j <= 10; j++) {  
      print('$i x $j = ${i * j}');  
    }  
    print('');  
  }  
}
```

```
9 x 1 = 9  
9 x 2 = 18  
9 x 3 = 27  
9 x 4 = 36  
9 x 5 = 45  
9 x 6 = 54  
9 x 7 = 63  
9 x 8 = 72  
9 x 9 = 81  
9 x 10 = 90
```

21. Write a dart program to create a simple calculator that performs addition, subtraction, multiplication, and division.

```
void main() {  
  double num1 = 10;
```



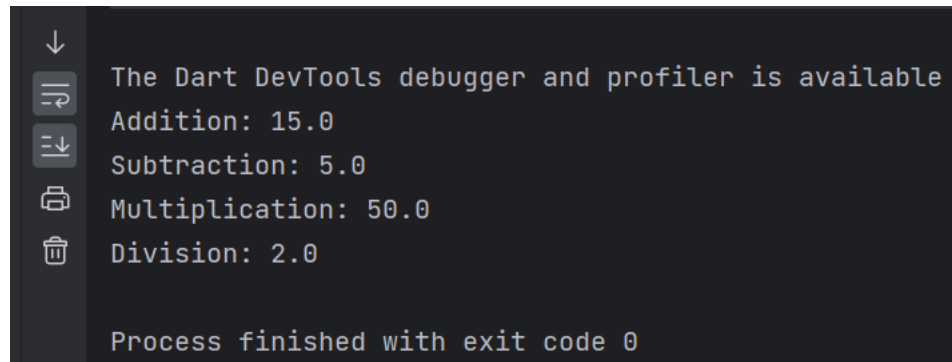
```

double num2 = 5;

print('Addition: ${add(num1, num2)}');
print('Subtraction: ${subtract(num1, num2)}');
print('Multiplication: ${multiply(num1, num2)}');
print('Division: ${divide(num1, num2)}');
}

double add(double a, double b) => a + b;
double subtract(double a, double b) => a - b;
double multiply(double a, double b) => a * b;
double divide(double a, double b) => a / b;

```



The Dart DevTools debugger and profiler is available

Addition: 15.0

Subtraction: 5.0

Multiplication: 50.0

Division: 2.0

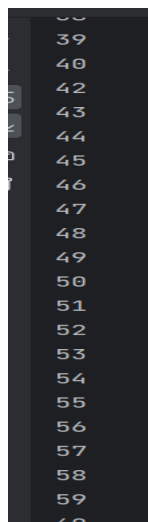
Process finished with exit code 0

22. Write a dart program to print 1 to 100 but not 41.

```

void main() {
  for (int i = 1; i <= 100; i++) {
    if (i != 41) {
      print(i);
    }
  }
}

```



39

40

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

23. Write a program in Dart to print your own name using function.

```

void main() {
  printName('waqas zafar');
}

```

```
void printName(String name) {
  print('My name is $name');
}
```

The Dart DevTools debugger and prof

My name is waqas zafar

Process finished with exit code 0

24. Write a program in Dart to print even numbers between intervals using function.

```
void main() {
  printEvenNumbers(1, 20);
}

void printEvenNumbers(int start, int end) {
  print('Even numbers between $start and $end:');
  for (int i = start; i <= end; i++) {
    if (i % 2 == 0) {
      print(i);
    }
  }
}
```

Even numbers between 1 and 20:

2

4

6

8

10

12

14

16

18

20

25. Create a function called greet that takes a name as an argument and prints a greeting message. For example, greet("John") should print

"Hello, John".

```
void main() {
  greet('John');
}

void greet(String name) {
  print('Hello, $name');
}
```

```
[Debug service available at http:
↓
≡ The Dart DevTools debugger and pro
≡↓ Hello, John
🖨 Process finished with exit code 0
```

26. Write a program in Dart that generates random password.

27. Write a program in Dart that find the area of a circle using function. Formula: $\pi * r * r$

```
import 'dart:math';

void main() {
  double radius = 5.0;
  double area = calculateArea(radius);
  print('Area of the circle: $area');
}

double calculateArea(double radius) {
  return pi * radius * radius;
}
```

```
5 The Dart DevTools debugger and profiler is available at:
Area of the circle: 78.53981633974483
Process finished with exit code 0
```

28. Write a program in Dart to reverse a String using function.

```
void main() {
  String text = 'Hello, World!'; // Example string to reverse
  String reversedText = reverseString(text);
  print('Original String: $text');
  print('Reversed String: $reversedText');
}

String reverseString(String text) {
  return text.split('').reversed.join('');
}
```

```
↑
↓ The Dart DevTools debugger and profiler is available at:
≡ Original String: Hello, World!
≡↓ Reversed String: !dlroW ,olleH
🖨 Process finished with exit code 0
🗑
```

29. Write a program in Dart to calculate power of a certain number. For e.g $5^3=125$

```
import 'dart:math';

void main() {
  int base = 5;
  int exponent = 3;
  int? result = calculatePower(base, exponent) as int?;
  print('$base^$exponent = $result');
}

num calculatePower(int base, int exponent) {
  return pow(base, exponent);
}
```

```
The Dart DevTools debugger and profiler
5^3 = 125

Process finished with exit code 0
```

30. Write a function in Dart named add that takes two numbers as arguments and returns their sum.

```
void main() {
  int num1 = 10;
  int num2 = 5;
  int sum = add(num1, num2);
  print('Sum: $sum');
}

int add(int a, int b) {
  return a + b;
}
```

```
The Dart DevTools debugger and profiler i
Sum: 15

Process finished with exit code 0
```

31. Write a function in Dart called maxNumber that takes three numbers as arguments and returns the largest number.

```
import 'dart:math';

void main() {
  int num1 = 10;
  int num2 = 20;
  int num3 = 15;
  int max = maxNumber(num1, num2, num3);
}
```

```

    print('Max Number: $max');
  }

  int maxNumber(int a, int b, int c) {
    return max(a, max(b, c));
  }

```

```

The Dart DevTools debugger and profiler i
Max Number: 20

Process finished with exit code 0

```

32. Write a function in Dart called `isEven` that takes a number as an argument and returns `True` if the number is even, and `False` otherwise.

```

void main() {
  int number = 6; // Change the number to test different cases
  bool result = isEven(number);
  print('$number is even: $result');
}

bool isEven(int number) {
  return number % 2 == 0;
}

```

```

The Dart DevTools debugger and profiler
6 is even: true

Process finished with exit code 0

```

33. Write a function in Dart called `createUser` with parameters `name`, `age`, and `isActive`, where `isActive` has a default value of `true`.

```

void main() {
  createUser('zafar', 30);
  createUser('waqas', 25, isActive: false);
}

void createUser(String name, int age, {bool isActive = true}) {
  print('Name: $name, Age: $age, Active: $isActive');
}

```

```
Console [Debug service available at http://127.0.0.1:8080]
↑
↓ The Dart DevTools debugger and profiler is available at http://127.0.0.1:8080
↺ Name: zafar, Age: 30, Active: true
↻ Name: waqas, Age: 25, Active: false
Print
Process finished with exit code 0
```

34. Write a function in Dart called `calculateArea` that calculates the area of a rectangle. It should take `length` and `width` as arguments, with a default value of 1 for both.

Formula: `length * width`.

```
void main() {
  double length = 5.0;
  double width = 3.0;
  double area = calculateArea(length, width);
  print('Area of rectangle: $area');
}

double calculateArea(double length, double width) {
  return length * width;
}
```

```
5 The Dart DevTools debugger and profiler is available at http://127.0.0.1:8080
↓ Area of rectangle: 15.0
```

35. Create a list of names and print all names using list.

```
void main() {
  List<String> names = ['Alice', 'Bob', 'Charlie', 'David', 'Eve'];
  print('List of Names:');
  for (String name in names) {
    print(name);
  }
}
```

```
The Dart DevTools debugger and profiler is available at http://127.0.0.1:8080
List of Names:
Alice
Bob
Charlie
David
Eve
```

36. Create a set of fruits and print all fruits using loop.

```
void main() {
  Set<String> fruits = {'Apple', 'Banana', 'Orange', 'Mango'};
  print('Set of Fruits:');
  for (String fruit in fruits) {
```

```

    print(fruit);
  }
}

```

```

The Dart DevTools debugger
Set of Fruits:
Apple
Banana
Orange
Mango

```

37. Create a program that reads list of expenses amount using user input and print total.

```

import 'dart:io';

void main() {
  List<double> expenses = [];
  double total = 0.0;

  print('Enter expenses amount (press Enter to finish):');
  while (true) {
    String input = stdin.readLineSync()!;
    if (input.isEmpty) break;
    double expense = double.parse(input);
    expenses.add(expense);
    total += expense;
  }

  print('Total expenses: $total');
}

```

```

The Dart DevTools debugger and profiler is available at: https://dart.dev/tools
Enter expenses amount (press Enter to finish):
400
699
6767
234234
5656
453
234554545

Total expenses: 234802754.0

```

38. Create an empty list of type string called days. Use the add method to add names of 7 days and print all days.

```

void main() {
  List<String> days = [];
}

```

```

    days.addAll(['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday',
'Saturday', 'Sunday']);
    print('Days of the Week:');
    print(days);
}

```

```

The Dart DevTools debugger and profiler is available at: http://127.0.0.1:8080
Days of the Week:
[Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday]
Process finished with exit code 0

```

39. Add your 7 friend names to the list. Use where to find a name that starts with alphabet a.

```

void main() {
    List<String> friends = ['waqas', 'zafar', 'Abdullah', 'faseeh', 'Alyan',
'Adam', 'Anna'];
    print('Friends List:');
    print(friends);

    String startsWithA = friends.firstWhere((friend) => friend.startsWith('A')
|| friend.startsWith('a'), orElse: () => '');
    if (startsWithA.isNotEmpty) {
        print('Name starting with A: $startsWithA');
    } else {
        print('No name starting with A found.');
```

```

}
}

The Dart DevTools debugger and profiler is available at: http://127.0.0.1:8080
Friends List:
[waqas, zafar, Abdullah, faseeh, Alyan, Adam, Anna]
Name starting with A: Abdullah

```

40. Create a map with name, address, age, country keys and store values to it. Update country name to other country and print all keys and values.

```

void main() {
    Map<String, dynamic> person = {
        'name': 'fuzail',
        'address': '123 Main Street',
        'age': 20,
        'country': 'pakistan',
    };

    person['country'] = 'Canada';

    print('Map Keys and Values:');
    person.forEach((key, value) {
        print('$key: $value');
    });
}

```



```
[debug service available at http://127.0.0.1]

The Dart DevTools debugger and profiler is av
Map Keys and Values:
name: fuzail
address: 123 Main Street
age: 20
country: Canada
```

41. Create a map with name, phone keys and store some values to it. Use where to find all keys that have length 4.

```
void main() {
  Map<String, String> contacts = {
    'waqas': '123-456-7890',
    'fuzail': '234-567-8901',
    'alyan': '345-678-9012',
  };

  Set<String> keysWithLength4 = contacts.keys.where((key) => key.length ==
6).toSet();
  print('Keys with Length 6:');
  print(keysWithLength4);
}
```

```
The Dart DevTools debugger and profiler is
Keys with Length 6:
{fuzail}
```

42. Create a simple to-do application that allows user to add, remove, and view their task.

```
import 'dart:io';

void main() {
  List<String> tasks = [];

  while (true) {
    print('Enter a task (or type "exit" to quit):');
    String input = stdin.readLineSync()!;
    if (input.toLowerCase() == 'exit') {
      break;
    } else {
      tasks.add(input);
      print('Task added.');
```

```

The Dart DevTools debugger and profiler is available
Enter a task (or type "exit" to quit):
write
Task added.
Enter a task (or type "exit" to quit):
read
Task added.
Enter a task (or type "exit" to quit):
destroy
Task added.
Enter a task (or type "exit" to quit):
exit
Tasks:
[write, read, destroy]

```

43. Write a dart program to create a class Laptop with properties [id, name, ram] and create 3 objects of it and print all details.

```

class Laptop {
  int id;
  String name;
  int ram;

  Laptop(this.id, this.name, this.ram);

  @override
  String toString() {
    return 'ID: $id, Name: $name, RAM: $ram GB';
  }
}

void main() {
  Laptop laptop1 = Laptop(1, 'HP Pavilion', 8);
  Laptop laptop2 = Laptop(2, 'Dell Inspiron', 16);
  Laptop laptop3 = Laptop(3, 'MacBook Pro', 32);

  print('Laptop Details:');
  print(laptop1);
  print(laptop2);
  print(laptop3);
}

```

```

The Dart DevTools debugger and profiler is available
Laptop Details:
ID: 1, Name: HP Pavilion, RAM: 8 GB
ID: 2, Name: Dell Inspiron, RAM: 16 GB
ID: 3, Name: MacBook Pro, RAM: 32 GB

```

44. Write a dart program to create a class House with properties [id, name, price]. Create a constructor of it and create 3 objects of it. Add them to the list and print all details.

```

class House {
  int id;

```

```

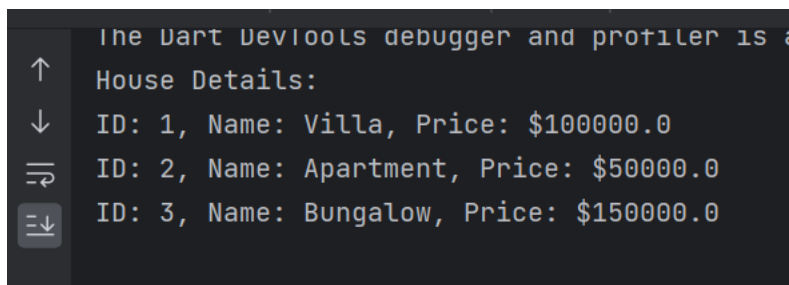
String name;
double price;

House(this.id, this.name, this.price);
}

void main() {
  List<House> houses = [
    House(1, 'Villa', 100000),
    House(2, 'Apartment', 50000),
    House(3, 'Bungalow', 150000),
  ];

  print('House Details:');
  houses.forEach((house) => print('ID: ${house.id}, Name: ${house.name},
Price: \${house.price}'));
}

```



```

The Dart DevTools debugger and profiler is a
House Details:
ID: 1, Name: Villa, Price: $100000.0
ID: 2, Name: Apartment, Price: $50000.0
ID: 3, Name: Bungalow, Price: $150000.0

```

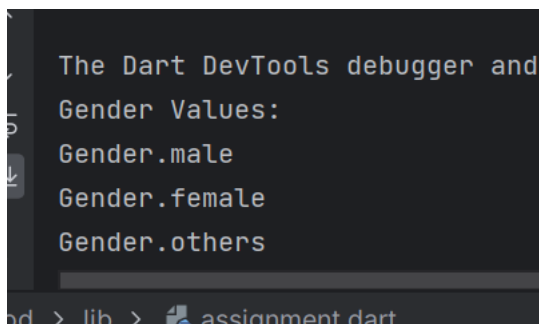
45. Write a dart program to create an enum class for gender [male, female, others] and print all values.

```

enum Gender {
  male,
  female,
  others,
}

void main() {
  print('Gender Values:');
  Gender.values.forEach((gender) => print(gender));
}

```



```

The Dart DevTools debugger and
Gender Values:
Gender.male
Gender.female
Gender.others

```

46. Write a dart program to create a class Animal with properties [id, name, color]. Create another class called Cat and extends it from Animal. Add new properties sound in String. Create an object of a Cat and print all details.

```

class Animal {
  int id;
  String name;
  String color;

  Animal(this.id, this.name, this.color);
}

class Cat extends Animal {
  String sound;

  Cat(int id, String name, String color, this.sound) : super(id, name,
color);
}

void main() {
  Cat myCat = Cat(1, 'Whiskers', 'White', 'Meow');
  print('Cat Details:');
  print('ID: ${myCat.id}, Name: ${myCat.name}, Color: ${myCat.color}, Sound:
${myCat.sound}');
}

```

```

The Dart DevTools debugger and profiler is available
Cat Details:
ID: 1, Name: Whiskers, Color: White, Sound: Meow

```

47. Write a dart program to create a class Camera with private properties [id, brand, color, price]. Create a getter and setter to get and set values. Also, create 3 objects of it and print all details.

```

class Camera {
  int _id;
  String _brand;
  String _color;
  double _price;

  Camera(this._id, this._brand, this._color, this._price);

  int get id => _id;
  set id(int value) => _id = value;

  String get brand => _brand;
  set brand(String value) => _brand = value;

  String get color => _color;
  set color(String value) => _color = value;

  double get price => _price;
  set price(double value) => _price = value;
}

void main() {
  Camera camera1 = Camera(1, 'Canon', 'Black', 500);
  Camera camera2 = Camera(2, 'Nikon', 'Silver', 600);
  Camera camera3 = Camera(3, 'Sony', 'White', 700);
}

```

```

    print('Camera Details:');
    print('Camera 1: ID=${camera1.id}, Brand=${camera1.brand},
Color=${camera1.color}, Price=\${camera1.price}');
    print('Camera 2: ID=${camera2.id}, Brand=${camera2.brand},
Color=${camera2.color}, Price=\${camera2.price}');
    print('Camera 3: ID=${camera3.id}, Brand=${camera3.brand},
Color=${camera3.color}, Price=\${camera3.price}');
}

```

The Dart DevTools debugger and profiler is available at: [https://dart.dev/tools](#)

```

Camera Details:
Camera 1: ID=1, Brand=Canon, Color=Black, Price=$500.0
Camera 2: ID=2, Brand=Nikon, Color=Silver, Price=$600.0
Camera 3: ID=3, Brand=Sony, Color=White, Price=$700.0

```

48. Create an interface called **Bottle** and add a method to it called **open()**. Create a class called **CokeBottle** and implement the **Bottle** and print the message “Coke bottle is opened”. Add a factory constructor to **Bottle** and return the object

```

class CokeBottle implements Bottle {
  @override
  void open() {
    print('Coke bottle is opened');
  }
}

class Bottle {
  void open() {}
}

void main() {
  Bottle bottle = CokeBottle();
  bottle.open();
}

```

The Dart DevTools debugger
Coke bottle is opened

49. Create a simple quiz application using oop that allows users to play and view their score.

```
import 'dart:io';

class Quiz {
  List<String> questions = [
    'Question 1: What is Dart?',
    'Question 2: What is Flutter?',
    'Question 3: What is null safety?',
  ];

  List<String> answers = [
    'A programming language',
    'A UI toolkit',
    'A feature in Dart for safer code',
  ];

  int score = 0;

  void playQuiz() {
    for (int i = 0; i < questions.length; i++) {
      print(questions[i]);
      print('Enter your answer:');
      String userAnswer = stdin.readLineSync()!;
      if (userAnswer.toLowerCase() == answers[i].toLowerCase()) {
        print('Correct!');
        score++;
      } else {
        print('Incorrect. Correct answer: ${answers[i]}');
      }
    }
    print('Quiz completed. Your score: $score / ${questions.length}');
  }
}

void main() {
  Quiz quiz = Quiz();
  quiz.playQuiz();
}
```

```
Question 1: What is Dart?
Enter your answer:
A programming language
Correct!
Question 2: What is Flutter?
Enter your answer:
A UI toolkit
Correct!
Question 3: What is null safety?
Enter your answer:
A feature in dart for safer code
Correct!
Quiz completed. Your score: 3 / 3
```

50. What is the purpose of the “?” operator in Dart null safety?

The “?” operator in Dart null safety is used for null-aware access. It allows accessing properties or calling methods on objects that might be null without causing a null pointer exception. If the object is null, the expression evaluates to null instead of throwing an error.

51. Create a late variable named **address**, assign a **US** value to it and print it.

```
late String address;
void main() {
  address = '123 Main Street, USA';
  print('Address: $address');
}
```

52. How do you declare a nullable type in Dart null safety?

In Dart null safety, you can declare a nullable type by appending “?” to the type.

```
int? nullableInt;
void main() {
  nullableInt = null;
  print('Nullable Int: $nullableInt');
}
```

53. Write a function that accepts a nullable int parameter and returns 0 if the value is null using null coalescing operator “??”.

```
int returnZeroIfNull(int? value) {
  return value ?? 0;
}

void main() {
  int? nullableValue = null;
  print('Value or 0 if null: ${returnZeroIfNull(nullableValue)}');
}
```

54. Write a function named **generateRandom()** in dart that randomly returns **100** or **null**. Also, assign a return value of the function to a variable named **status** that can’t be null. Give status a default value of 0, if **generateRandom()** function returns null.

```
import 'dart:math';

int? generateRandom() {
  Random random = Random();
  return random.nextBool() ? 100 : null;
}

void main() {
  int status = generateRandom() ?? 0;
  print('Status: $status');
}
```

```
The Dart DevTools
Status: 100
```