



Лабораторийн ажил 9

Dart & FLutter basics

Гүйцэтгэсэн: О.Саруулгэрэл /22B1NUM1637/

Шалгасан: Г.Ууганбаяр

2025 он

Улаанбаатар хот

Dart	2
1. Introduction & Basics	2
Questions	3
2. Conditions and Loops	4
Questions	4
3. Functions	6
Questions	6
4. Collection	7
Questions	8
5. File Handling	9
Questions	10
6. OOP	11
Questions	11
7. Null Safety	13
Questions	13
8. Asynchronous Programming	14
Questions	15

Dart

1. Introduction & Basics

1.1. Dart гэж юу вэ?

Dart бол Google-ийн хөгжүүлсэн, хэрэглэгчийн интерфэйс бүтээхэд зориулагдсан, объект хандалтат програмчлалын хэл юм. Энэ нь Flutter фреймворктой хамт хэрэглэгддэг бөгөөд нэг кодоор Android, iOS, вэб, десктоп зэрэг олон платформд зориулсан аппликейшн хөгжүүлэх боломжийг олгодог.

1.2. Dart програмын бүтэц

Dart код нь ихэвчлэн `main()` функцээс эхэлдэг. Энэ нь програмын гол гүйцэтгэл эхлэх цэг юм.

```
void main() {
  print('Hello, Dart!');
}
```

1.3. Өгөгдлийн төрлүүд (Data Types)

Dart хэл нь төрөл тодорхойлогч (type-safe) шинж чанартай. Үндсэн өгөгдлийн төрлүүдэд дараах зүйлс орно:

- int
- double
- string
- bool
- var – автоматаар төрлийг тодорхойлдог хувьсагч
- Lists -> `List<String> fruits = ["apple", "banana", "mango"];`
- Maps -> `Map<String, String> capital = {"Mongolia": "Ulaanbaatar"};`
- Sets -> `Set<int> numbers = {1, 2, 3};`
- Runes(Unicode тэмдэгтүүд) -> `var heart = '\u2665'; // ♥`
- Null -> `String? Name;`

1.4. Хувьсагч (Variables)

Хувьсагчийг `var`, `final`, эсвэл `const` ашиглан зарлаж болно.

`var` – дахин утга оноож болдог

`final` – нэг удаа л утга өгнө

`const` – compile-time дээр тогтсон утгатай

1.5. Операторууд (Operators)

-expr -> Энэ оператор нь хувьсагч эсвэл тооны эсрэг утгыг авахад ашиглагдана.

```
int x = 5;
int y = -x; // y = -5
```

/ -> Үр дүн нь **double** төрөлтэй байна.

~/ -> Энэ нь бүхэл тоон хуваалт хийдэг оператор юм. Үр дүнг бутархай биш, зөвхөн **integer** (бүхэл тоо) хэлбэрээр авна.

1.6. User Input

Dart хэл дээр хэрэглэгчээс мэдээлэл авахын тулд dart:io санг ашигладаг. Энэ сан нь оролт/гаралт (input/output) үйлдэл хийх боломжийг олгодог.

stdin.readLineSync() -> хэрэглэгчийн оруулсан өгөгдлийг String? төрлөөр буцаадаг (nullable string).

1.7. String

"" эсвэл "" "" ашиглан олон мөртэй текст бичиж болно.

Questions

1. 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. Formula= (total bill amount) / number of people

```
1 import 'dart:io';
2
3 void main() {
4   print("Нийт төлбөрийг оруулна уу:");
5   double totalAmount = double.parse(stdin.readLineSync()!);
6
7   print("Нийт хүний тоог оруулна уу:");
8   int numberOfPeople = int.parse(stdin.readLineSync()!);
9
10  double splitAmount = totalAmount / numberOfPeople;
11
12  print("Хүн бүрийн төлөх дүн: ₮${splitAmount.toStringAsFixed(2)}₮");
13 }
```

2. 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= (distance) / (speed)

```
1 void main() {
2     double distance = 25;
3     double speed = 40;
4
5     double timeInHours = distance / speed;
6     double timeInMinutes = timeInHours * 60;
7
8     print("Оффис руу явах хугацаа: ${timeInMinutes.toStringAsFixed(2)} минут");
9 }
10
```

2. Conditions and Loops

2.1. assert()

assert() нь алдааг шалгах зориулалттай тусгай оператор юм. Програм бичих явцад тодорхой нөхцөл үнэн байх ёстой гэсэн баталгааг тавихад ашиглагддаг. Хэрэв тухайн нөхцөл үнэн биш бол (false), програм алдаа өгч зогсдог.

```
void main() {
    int number = -5;
    assert(number >= 0, "Тоо эерэг байх ёстой!");
    print("Too: $number");
}
```

Questions

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

```
1 import 'dart:io';
2
3 void main() {
4     stdout.write("Эхний тоог оруулна уу: ");
5     int x1 = int.parse(stdin.readLineSync()!);
6
7     stdout.write("Хоёр дахь тоог оруулна уу: ");
8     int x2 = int.parse(stdin.readLineSync()!);
9
10    print("$x1 + $x2 = ${x2 + x1}");
11    print("$x1 - $x2 = ${x1 - x2}");
12    print("$x1 * $x2 = ${x2 * x1}");
13    if (x2 != 0) {
14        double division = x1 / x2;
15        print("$x1 / $x2 = ${division.toStringAsFixed(2)}");
16    } else {
17        print("Хуваах боломжгүй: Хоёр дахь тоо 0 байна.");
18    }
19 }
20
```

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

```
1 void main() {
2     for (int i = 1; i <= 100; i++) {
3         if (i != 41)
4             print(i);
5         else
6             continue;
7     }
8 }
9
```

3. Functions

3.1. Anonymous Function

Нэргүй функц гэдэг нь нэр өгөөгүй функц бөгөөд ихэвчлэн түр хугацаанд, эсвэл callback хэлбэрээр ашиглагддаг. Тодорхой нэр өгөх шаардлагагүй, шууд хувьсагчид хадгалах эсвэл функц дотор дамжуулах байдлаар ашиглана.

```
void main() {
  // Anonymous function
  var cube = (int number) {
    return number * number * number;
  };

  print("The cube of 2 is ${cube(2)}");
  print("The cube of 3 is ${cube(3)}");
}
```

3.2. Arrow Function

returnType functionName(parameters) => expression:

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

```
int add(int a, int b) => a + b;
```

Questions

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

```
1 void createUser(String name, int age, {bool isActive = true}) {}
```

2. 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.

```
1 int calculateArea({int length = 1, int width = 1}) {
2   return length * width;
3 }
```

4. Collection

4.1. List

```
List<String> names = ['Bat', 'Saraa', 'Davaa'];
```

Dart-ийн List нь динамик буюу хүссэн хэмжээгээр томрох/жижигрэх боломжтой. Гэхдээ List.filled() ашиглан тогтмол урттай жагсаалт мөн үүсгэж болдог.

```
var fixedList = List.filled(3, 0); // [0, 0, 0]
```

4.2. Set

Set нь давхардсан утгагүй элементүүдийг хадгалдаг цуглуулгын төрөл юм. List-тэй төстэй боловч нэг ижил утга нэгээс илүү удаа байж болохгүй.

```
Set <variable_type> variable_name = {};
```

4.3. Map

Map нь key-value буюу түлхүүр-утга хосоор өгөгдлийг хадгалдаг бүтэц юм. Python-ийн dict, JavaScript-ийн object-той төстэй.

```
void main(){
  Map<String, String> countryCapital = {
    'USA': 'Washington, D.C.',
    'India': 'New Delhi',
    'China': 'Beijing'
  };
  print(countryCapital);
}
```

4.4. Where

where() функц нь жагсаалт, set зэрэг цуглуулга дээр ашиглагддаг бөгөөд тодорхой нөхцөлд нийцсэн элементүүдийг шүүж авахад хэрэглэгддэг.

```
Iterable<E> where(
  bool test(
    E element
  )
)
```

Example: Filter only odd number form list

```
void main() {
  List<int> numbers = [2, 4, 6, 8, 10, 11, 12, 13, 14];

  List<int> oddNumbers = numbers.where((number) => number.isOdd).toList();
  print(oddNumbers);
}
```


Questions

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

```
1 void main() {
2     Map<String, String> contact = {
3         "name": "Saruul",
4         'phone': '99119911',
5         'city': 'Ulaanbaatar',
6     };
7
8     var result = contact.keys.where((key) => key.length == 4);
9     print("4 үсэртэй түлхүүрүүд: ${result.toList()}");
10 }
11
```

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

```
1 import 'dart:io';
2
3 void main() {
4     List<String> tasks = [];
5
6     while (true) {
7         print("Хийх ажлын жагсаалт");
8         print("1. Шинэ ажил нэмэх");
9         print("2. Ажил устгах");
10        print("3. Ажлуудыг харах");
11        print("4. Гаргах");
12        stdout.write("Сонголтоо оруулна уу: ");
13        String? choice = stdin.readLineSync();
14    }
```

```

16     case '1':
17         stdout.write("хийх ажилаа оруулна уу: ");
18         String? task = stdin.readLineSync();
19         if (task != null && task.isNotEmpty) {
20             tasks.add(task);
21             print("Ажил нэмэгдлээ");
22         }
23         break;
24
25     case '2':
26         stdout.write("Устгах ажилын дугаар (0-${tasks.length - 1}): ");
27         int index = int.parse(stdin.readLineSync()!);
28         if (index >= 0 && index < tasks.length) {
29             print("Устгасан: ${tasks[index]}");
30             tasks.removeAt(index);
31         } else {
32             print("Буруу дугаар!");
33         }
34         break;
35
36     case '3':
37         print("Ажлын цагсаалт");
38         if (tasks.isEmpty) {
39             print("Ажил алга байна.");
40         } else {
41             for (int i = 0; i < tasks.length; i++) {
42                 print("$i: ${tasks[i]}");
43             }
44         }
45         break;
46
47     case '4':
48         return;
49
50     default:
51         print("Буруу сонголт. Дахин оролдоно уу.");
52 }
53 }
54 }

```

5. File Handling

файл унших (Read file) - readAsStringSync()

файл руу бичих (Write file) - writeAsStringSync()

файл устгах (Delete file) - deleteSync()

void main() {

File file = File('example.txt');

String content = file.readAsStringSync(); // Бүх агуулгыг уншина

file.writeAsStringSync('Сайн байна уу!\nЭнэ бол шинэ текст.');

if (file.existsSync()) {

```
file.deleteSync();
print('Файл устлаа.');
```

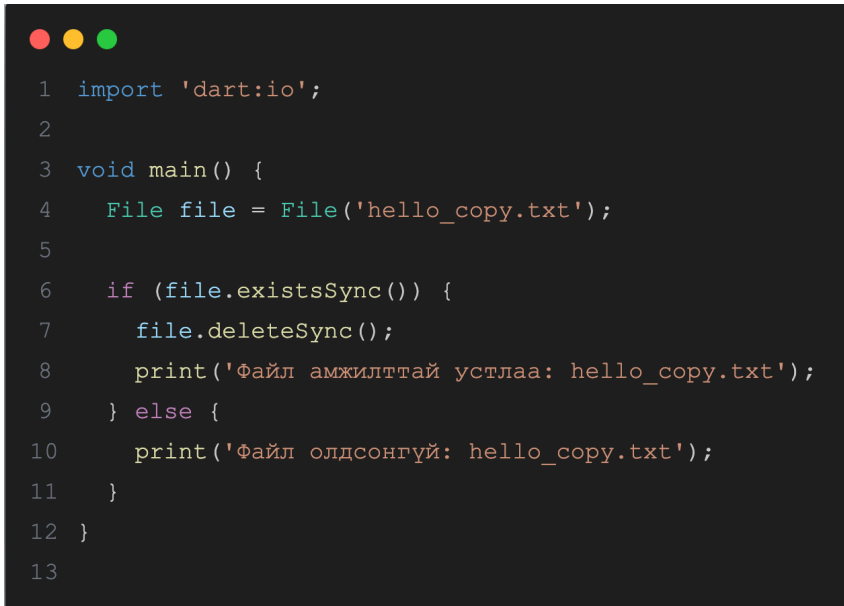
```
} else {
  print('Файл олдсонгүй.');
```

```
}

print("Файлын агуулга:");
print(content);
}
```

Questions

1. Write a dart program to delete the file “hello_copy.txt”. Make sure you have created the file “hello_copy.txt”.



```
1 import 'dart:io';
2
3 void main() {
4   File file = File('hello_copy.txt');
5
6   if (file.existsSync()) {
7     file.deleteSync();
8     print('Файл амжилттай устлаа: hello_copy.txt');
```

```
9   } else {
10    print('Файл олдсонгүй: hello_copy.txt');
```

```
11  }
12 }
13
```

2. Write a dart program to store name, age, and address of students in a csv file and read it.

```
1 import 'dart:io';
2
3 void main() {
4   File file = File('students.csv');
5
6   file.writeAsStringSync('Dorjoo,20,Ulaanbaatar\n', mode: FileMode.append);
7   file.writeAsStringSync('Bataa,22,Darkhan\n', mode: FileMode.append);
8   file.writeAsStringSync('Saraa,19,Erdenet\n', mode: FileMode.append);
9
10  print('CSV файл амжилттай үүслээ.');
```

6. OOP

Questions

1. 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 of CokeBottle. Instantiate CokeBottle using the factory constructor and call the open() on the object.

```
1 class IBottle {
2   void open() {}
3   factory IBottle() {
4     return CokeBottle();
5   }
6 }
7
8 class CokeBottle implements IBottle {
9   @override
10  void open() {
11    print("Coke bottle is opened");
12  }
13 }
14
15 void main() {
16   IBottle bottle = IBottle();
17   bottle.open();
18 }
19
```

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

```
1 import 'dart:io';
2
3 class Question {
4   String question;
5   List<String> options;
6   int correctAnswer;
7
8   Question(this.question, this.options, this.correctAnswer);
9 }
10
11 void main() {
12   List<Question> quiz = [
13     Question("1. Dart-г аль компани бүтээсэн бэ?", [
14       "Apple",
15       "Google",
16       "Microsoft",
17     ], 2),
18     Question("2. Flutter ямар хэл дээр бичигдсэн бэ?", [
19       "Dart",
20       "Java",
21       "C#",
22     ], 1),
23     Question("3. 5 + 3 = ?", ["6", "7", "8"], 3),
24   ];
25
26   int score = 0;
27
28   for (var q in quiz) {
29     print(q.question);
30     for (int i = 0; i < q.options.length; i++) {
31       print("${i + 1}. ${q.options[i]}");
32     }
33
34     stdout.write("Таны хариулт (тоог оруулна уу): ");
35     int? answer = int.tryParse(stdin.readLineSync()!);
36
37     if (answer == q.correctAnswer) {
38       print("Зөв!\n");
39       score++;
40     } else {
41       print("Буруу. Зөв хариулт: ${q.options[q.correctAnswer - 1]}\n");
42     }
43   }
44
45   print("Та нийт $score/${quiz.length} оноо авлаа!");
```

7. Null Safety

1. Null assertion operator (!)

"Энэ хувьсагч null биш" гэж хүчээр ашиглаж байгаа.

2. Default утга өгөх (?? оператор)

String? Name;

String finalName = name ?? "Зочин"; // Хэрвээ name нь null бол "Зочин" гэж авна

3. Null-aware operator (?.)

Хэрвээ хувьсагч null биш бол дараагийн үйлдлийг гүйцэтгэнэ.

String? name = "Anu";

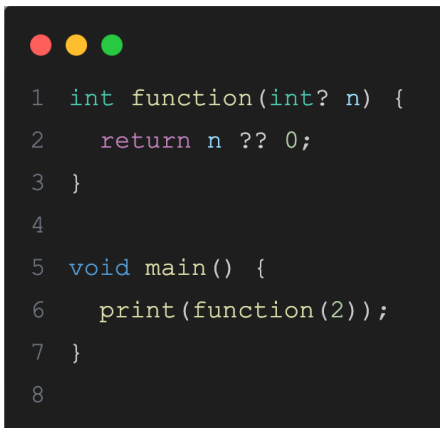
print(name?.toUpperCase()); // ANU

name = null;

print(name?.toUpperCase()); // null

Questions

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



```

1  int function(int? n) {
2    return n ?? 0;
3  }
4
5  void main() {
6    print(function(2));
7  }
8

```

2. 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.

```

1 import 'dart:math';
2
3 int? generateRandom() {
4   Random random = Random();
5   return random.nextBool() ? 100 : null;
6 }
7
8 void main() {
9   int status = generateRandom() ?? 0;
10  print("Status: $status");
11 }
12

```

8. Asynchronous Programming

1. Future

Future нь ирээдүйд нэг утга буцаах буюу дуусах амлалт юм.

```

Future<String> fetchData() {
  return Future.delayed(Duration(seconds: 2), () => 'Өгөгдөл ирлээ!');
}

```

Future.delayed() – 2 секунд хүлээгээд утга буцаана.

2. async/await

async – функцийг асинхрон болгон

await – тухайн Future-г хүлээж, дараа нь үргэлжлүүлнэ

3. then() – Future дээр callback ажиллуулна

```

fetchData().then((value) {
  print("then: $value");
});

```

await шиг хүлээхгүй, харин амжилттай ирсний дараа тухайн блок ажиллана.

4. catchError() – алдаа барихад ашиглана

```

Future<int> divide(int a, int b) async {
  if (b == 0) throw Exception("0-ээр хувааж болохгүй!");
  return a ~/ b;
}

```

```

void main() {
  divide(10, 0).then((value) {
    print("Хариу: $value");
  }).catchError((e) {
    print("Алдаа: $e");
  });
}

```

```
});
}
```

5. Stream

Цаг хугацааны явцад олон удаагийн өгөгдөл дамжуулах боломжтой объект юм. Товчхондоо, Stream бол “өгөгдлийн урсгал” бөгөөд энэ нь нэг дор биш, алхам алхмаар ирдэг.
await for эсвэл listen() ашиглана

Questions

1. Write a Dart program that takes a list of integers as input, multiplies each integer by 2 asynchronously, and then prints the modified list.



```

1 import 'dart:io';
2 import 'dart:async';
3
4 void main() async {
5   print("Тоонуудоо таслалаар бич: ");
6   String? input = stdin.readLineSync();
7
8   List<int> numbers =
9     input!.split(',').map((e) => int.parse(e.trim())).toList();
10
11   List<int> doubled = await Future.delayed(
12     Duration(seconds: 1),
13     () => numbers.map((n) => n * 2).toList(),
14   );
15
16   print("Үр дүн: $doubled");
17 }
18
```

2. Write a Dart program that takes a string as input, reverses the string asynchronously, and then prints the reversed string.


```
1 import 'dart:io';
2
3 Future<String> reverseStringAsync(String input) async {
4   return await Future.delayed(Duration(seconds: 1), () {
5     return input.split('').reversed.join();
6   });
7 }
8
9 void main() async {
10   stdout.write("Текст оруулна уу: ");
11   String? text = stdin.readLineSync();
12
13   print("Урвуулж байна...");
14   String reversed = await reverseStringAsync(text!);
15   print("Урвуу текст: $reversed");
16 }
17
```