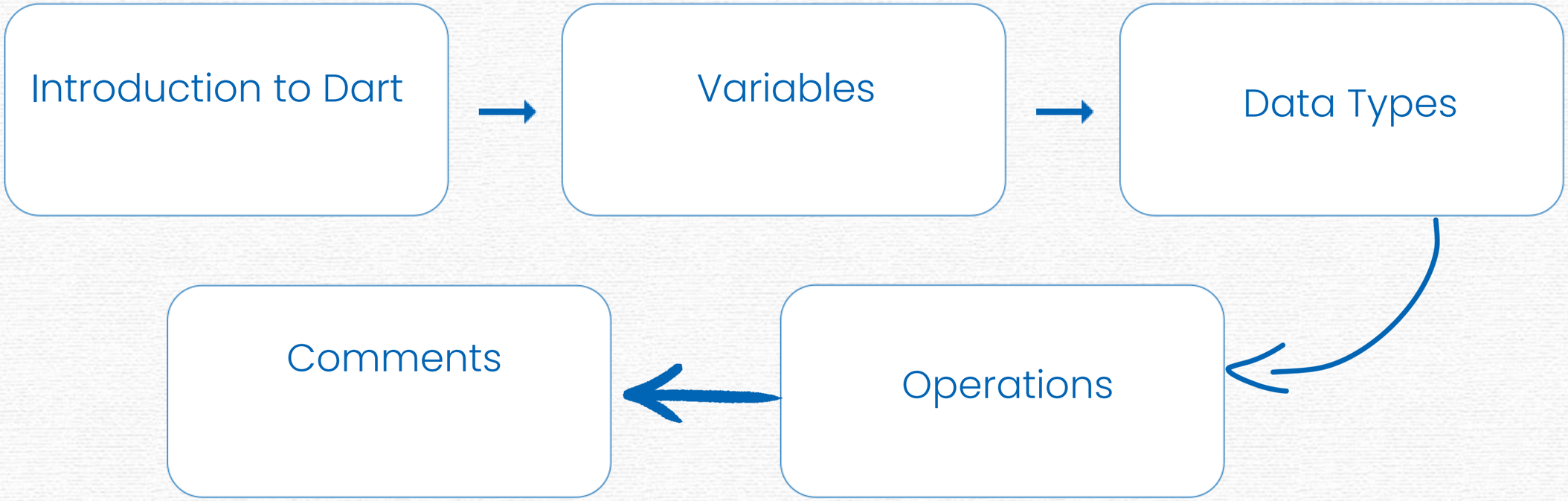
Mobile App

SESSION 1

Table of content



Mobile App Components

Mobile apps are built with two key components:

UI (User Interface): Focuses on design and use interaction.

Logic : Handles data processing and app functionality.

What is Dart

Dart is a programming language developed by Google, primarily used for building mobile, desktop, server, and web applications. It's the language behind Flutter, a popular framework for building cross-platform mobile apps

Why Dart

1.Cross-Platform Development

2.Fast Development with Hot Reload

3.High Performance

4.Strong Typing with Flexibility

5.Rich Standard Library and Package Ecosystem

print

The Starting Point How Dart Programs Begin

```dart
void main() {

  print('Welcome to Dart!');
}
```

Types of Variables

1.var: Type is inferred based on the assigned value.

```
var age = 25;  // Inferred as int
```

2.Explicit Types: Specify the type explicitly(int , double, string, boolean ,char).

```
String message = "Hello, Dart";
```

Types of Variables

3.final: Immutable after assignment, value assigned at runtime

```
final name = 'IEEE';
```

4.const: Compile-time constant, value fixed during compilation

```
const pi = 3.14;
```

Types of Variables

5.Dynamic: can hold any type of value.

```
dynamic variable = "Hello"; // Holds a string
variable = 42; // Now holds an integer
```

Final & const

Clear explaining the difference between FINAL and CONST:

Final Keyword

Meaning: The value of a final variable is set only once and cannot be reassigned after it's initialized. However, it is evaluated at runtime.

When to use: Use final when the value is known only at runtime.

final

```dart
void main() {
  final currentTime = DateTime.now(); // Value is set at runtime
  print(currentTime);

  // Uncommenting the line below will cause an error
   currentTime = DateTime.now(); // Error: currentTime is final and cannot be reassigned
}
```

The value of currentTime is determined when the program runs, not at compile time.

const

Const Keyword

Meaning: The value of a const variable is a compile-time constant. It must be known and fixed at compile time.

When to use: Use const for values that are immutable and known before runtime.

Shared Instances: When multiple const objects have the same value, they share the same memory instance (canonicalized).

const

```dart
void main() {
  const pi = 3.14159; // Value is known at compile time
  print(pi);

  // Uncommenting the line below will cause an error
   pi = 3.14; // Error: pi is const and cannot be reassigned
}
```

final & const

Feature	final	const
When Assigned	Value is set only once at runtime.	Value is set at compile time.
Mutability	Cannot be reassigned after initialization.	Cannot be reassigned after initialization.
Evaluation Time	Runtime.	Compile-time.
Examples of Use	Dynamic values like DateTime.now().	Fixed values like math.pi.
Shared Instances	Each instance is unique.	Instances with the same value are shared.

Data Types

Primitive Data Types:

int

double

string

bool

char (Dart doesn't have a char type, but it uses String for single characters):

```
String letter = 'A';  // Dart doesn't have a distinct `char` type
```

Null Safety

What is Null Safety?

Ensures variables are non-nullable by default, preventing null reference errors.

Nullable vs Non-Nullable:

Non-Nullable: Cannot be null. Example: int age = 25;

Nullable: Can be null. Use ?. Example: int? age = null;

Null Safety

1.Nullable Type Declaration (?): Used to declare variables that can hold null values.

```
String? name;  // Nullable String
```

2.Null Assertion Operator (!): Forces a nullable variable to be treated as non-null (use cautiously).

```
String? name = "Dart";
print(name!.length);  // Throws error if name is null.
}
```

Null Safety

3.Null-Coalescing Operator (??): Provides a default value if the expression is null.

```
String? name;
print(name ?? "Default Name");  // Outputs: "Default Name" if name is null.
```

Operations

1.Arithmetic operations:

These operations are used to perform mathematical calculations.

+ (Addition)

- (Subtraction)

***** (multiplication)

/ (division “**returns a double**”)

~/ (division “**returns a int**”)

% (Modulus “**remainder of division**”)

operations

2.Relational (Comparison) Operations

These operations are used to compare values, resulting in a boolean value (true or false).

(==) Equal to

```
bool isEqual = (5 == 5);  // true
```

(!=) Not Equal

```
bool isNotEqual = (5 != 3);  // true
```

<,>,>= , <=

Operations

3.Assignment Operators:

These operations are used to assign values to variables (assign the right side to the left side)

```
double number = 5;
number += 3;  // number = number + 3 -> 8

number -= 3;  // number = number - 3 -> 5

number *= 3;  // number = number * 3 -> 15

number /= 2;  // number = number / 2 -> 7.5

number %= 2;  // number = number % 2 -> 1
```

Operations

4.Logical Operators :

These operations are used to combine boolean expressions.

&& : Logical **AND** (both conditions must be true)

```
bool res = (5 > 3 && 2 < 4);  // true
```

Operations

|| : Logical OR (at least one condition must be true)

```
bool res = (5 > 3 && 2 < 4);  // true
```

! : Logical NOT (inverts the truth value)

```
bool result = !(5 > 3);  // false
```

comments

Single-line Comment: Starts with // and extends to the end of the line.

Multi-line Comment: Starts with /* and ends with */.

```
// This is a single-line comment
int age = 20;

/*
  This is a multi-line comment.
  It can span multiple lines.
*/
```

Exercise

Write a Dart program that does the following in one statement:

Declares a variable for a user's name, birth year, and the current year.
Calculates the user's age.
Checks if the user is an adult (18 or older).
Prints a greeting message, the user's age, and whether they are an adult.

Exercise

The Solution

```
void main() {
  var name = 'Alshimaa';
  var birthYear = 2004;
  var currentYear = 2025;

  var age = currentYear - birthYear;
  var isAdult = age >= 18 ? 'Yes' : 'No';

  print("Hello, $name. You are $age years old. Adult: $isAdult");
}
```

Task

Presentation

teams

tHANK YOU

SEE YOU NEXT TIME