📗 Dart Basics Learning Outline (Week 1)

Day 1: Variables & Data Types

Keyword	Meaning	When to Use	Example
var	Type inferred from first value → cannot change type later	When you know the value but don't want to explicitly write the type	dart var name = "Samir"; // String name = 25; // ❌ Error
dynamic	Truly dynamic, type can change at any time	When type is unknown, or needs to change during runtime (not recommended often)	dart dynamic x = "Hello"; x = 42; // ✅ allowed
final	Value assigned **once** (runtime constant)	When value doesn't change after assignment, but only known at runtime	dart final city = "Dhaka"; // decided at runtime
const	**Compile-time constant**	When value is fixed and known before running the program	dart const pi = 3.1416; // must be known at compile-time
late	Initialize later, but **before using**	When you can't assign immediately, but will before usage	dart late String username; username = "Samir"; print(username);

Day 2: Data Types

int → whole numbers
double → decimal numbers
String → text
bool → true/false
dynamic → flexible type

Day 3: String Interpolation

String interpolation means **inserting variables or expressions directly inside a string** using $ or ${}. It makes string formatting easier and more readable.

```
String name = "Samir"; -------  int age = 20;
print("My name is $name and I am $age years old.");
print("Next year I will be ${age + 1} years old.");
```

Day 4: Multi-line Strings

```
String story = '''
Once upon a time in Dart world,
there was a programmer learning variables.
He practiced every day!
''';
print(story);
```

Day 5: Null Safety Basics

```
// Nullable string (can hold null)
String? nickname;
print(nickname); // Output: null

// Provide default if null (?? operator)
print(nickname ?? "No nickname"); // Output: No nickname

// Force non-null (!) → risky if value is null
nickname = "Nafis";
print(nickname!.toUpperCase()); // Output: NAFIS
```

👉 ? makes a variable nullable,

👉 ?? gives default value,

👉 ! says "I'm sure it's not null" (but can crash if wrong).