

## day-7-LiB

# Introduction to Dart Programming: A Beginner's Guide

## Overview of Today

- ☐ Understanding Dart and its purpose
  - ☐ Core syntax and concepts
  - ☐ Variables, data types, and operators
  - ☐ Control flow and functions
  - ☐ Building and running simple Dart programs
- 

## Resources that'll help you

- [Dart in 100 seconds](#) - A great video to get an idea of Dart language in 100 seconds
  - [Dart Documentation](#) - Official Dart programming language docs
  - [DartPad](#) - Online Dart code editor
  - [Dart Tutorial in less than 2 hrs](#) - Dart Programming Tutorial from freecodecamp in case you prefer longer tutorials
  - [Dart Cheatsheet](#) - Quick reference for syntax
- 

## Understanding Dart

Dart is a client-optimized programming language developed by Google. It's primarily used for building mobile, desktop, web, and server applications, and is the language used for Flutter framework for Cross-platform development.

## Core Concepts

1. [Everything is an Object](#):
  - All values are objects

- Even numbers, functions, and null are objects
- All objects inherit from the Object class

## 2. Strong Type System:

- Static type checking
- Type inference
- Generic types

## 3. Null Safety:

- Variables can't contain null by default
- Explicit null handling
- Safer code with fewer runtime errors

# Basic Syntax Overview

```
// Variables and Data Types
String name = 'John';
int age = 25;
double height = 1.75;
bool isStudent = true;
var dynamicType = 'This type is inferred';

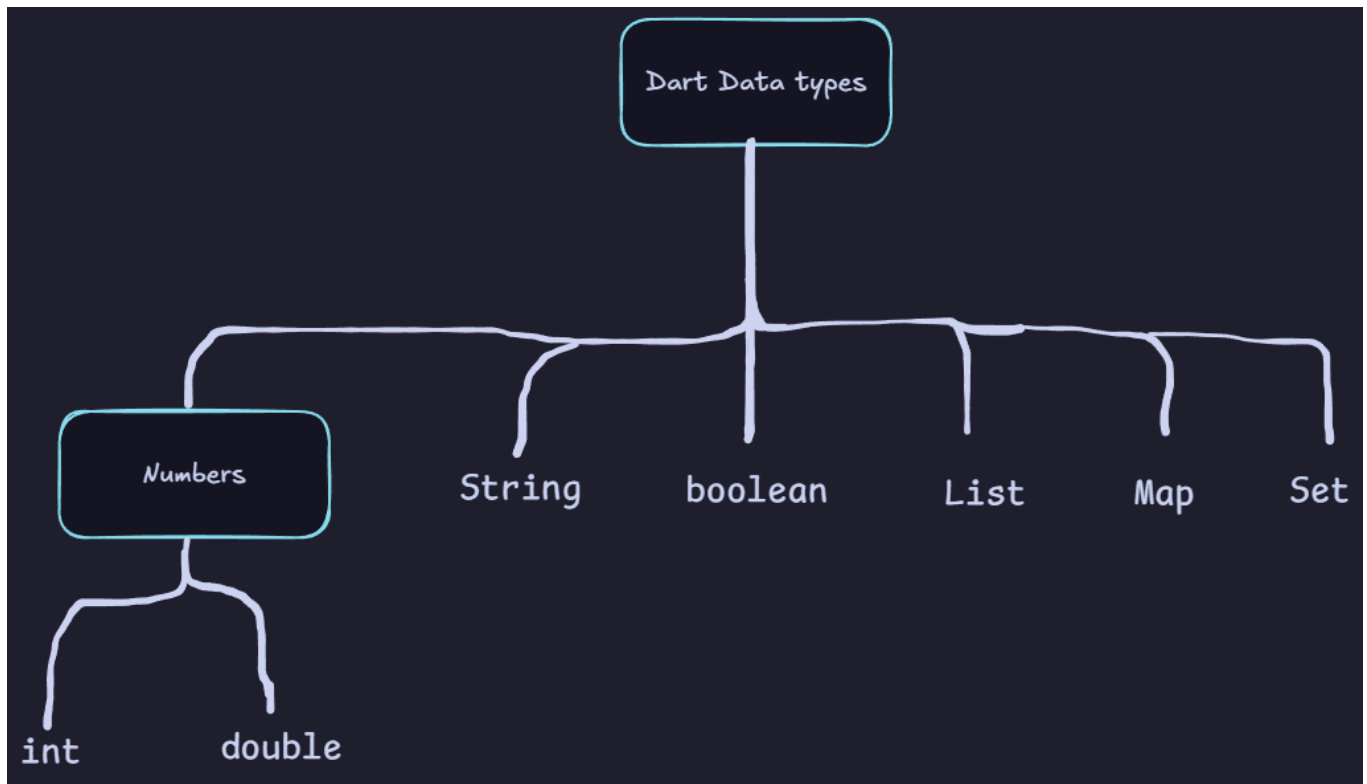
// Lists
List<String> fruits = ['apple', 'banana', 'orange'];

// Maps
Map<String, int> scores = {
  'math': 90,
  'science': 85
};

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

// Arrow Functions
int multiply(int a, int b) => a * b;
```

# Data Types in Dart



## Building Simple Dart Programs

### Setting Up Dart

#### Tip

Use DartPad for quick experimentation without local setup

For doing anything Dart-only, we suggest using [Dartpad](#), no need to install Dart SDK or anything else other than VSC since it'll come with Flutter SDK later.

### Creating Your First Dart Program

- I. Create a new main function:

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

### Control Flow Examples

```

// If statements
void checkAge(int age) {
    if (age >= 18) {
        print('Adult');
    } else {
        print('Minor');
    }
}

// Loops
void countToFive() {
    for (int i = 1; i <= 5; i++) {
        print(i);
    }
}

// Switch statements
String getGrade(int score) {
    switch (score) {
        case 90:
            return 'A';
        case 80:
            return 'B';
        default:
            return 'F';
    }
}

```

## Functions and Parameters

```

// Required parameters
String greet(String name) {
    return 'Hello, $name!';
}

// Optional parameters
void introduce(String name, [String? title]) {
    if (title != null) {
        print('This is $title $name');
    } else {
        print('This is $name');
    }
}

```

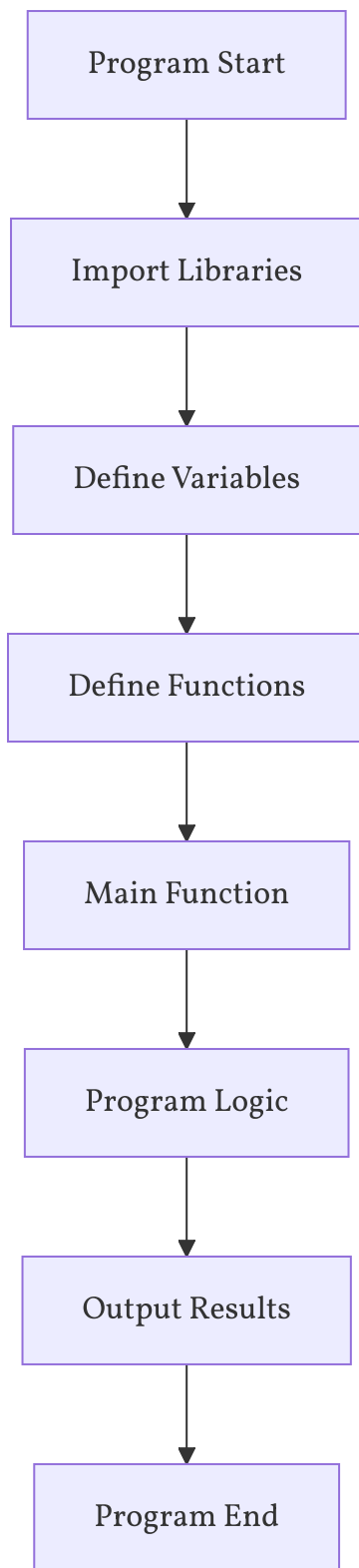
```
// Named parameters
void createUser({
  required String name,
  required int age,
  String? country
}) {
  print('User: $name, Age: $age, Country: ${country ?? "Unknown"}');
}
```

---

## Common Patterns and Best Practices

- + Use meaningful variable names
  - + Implement proper null safety
  - + Remember to Format code
  - + Add documentation comments
  - + Follow Dart style guide
  - Avoid using 'dynamic' unless necessary
  - Don't ignore nullable values
  - Avoid redundant type declarations
- 

## Basic Program Structure Flow



## Example Program Structure

This won't work on DartPad since they don't support `dart:io` library

```
// 1. Imports  
import 'dart:io';
```

```
// 2. Global variables (if needed)
const int maxAttempts = 3;

// 3. Functions
void processInput(String input) {
    // Function logic
}

// 4. Main program
main() {
    // Program logic
    String? userInput = stdin.readLineSync();
    if (userInput != null) {
        processInput(userInput);
    }
}
```

---

## Common Error Handling

```
try {
    // Code that might throw an error
    int result = int.parse('abc');
} on FormatException {
    print('Invalid format');
} catch (e) {
    print('An error occurred: $e');
} finally {
    print('This always runs');
}
```

---

## Tasks

- ☐ Open up Dartpad
- ☐ Create a simple calculator program
- ☐ Implement a program that works with lists and maps

☐ Create a function that uses different parameter types