ScopeTreasure Courses

DART COURSES · LESSON D.2 · CODE SCOPE SERIES

Dart Setup + Pehla Program

DartPad in the browser (zero install) or Dart on your own computer — write, save and run your very first Dart program.

Introduction

Before you can practise Dart, you need a place to write and run it. Dart gives you two doors: **DartPad**, a free website where Dart runs in your browser with zero installation, and the **Dart SDK** on your own computer for serious work later. In this lesson you set up both ideas, write your first real program, and learn the three habits every programmer follows: write → run → read the result. *(Roman Urdu: DartPad browser mein chalta hai — kuch install kiye baghair aaj hi shuru karein.)*

Key points

- **DartPad** (dartpad.dev) is the beginner's best friend: code on the left, a blue **Run** button on top, results (the console) on the right. No download, no setup, works on any computer.
- On your **own computer**, you install the **Dart SDK** (from dart.dev) once. Then any file ending in `.dart` — for example `hello.dart` — runs with the command `dart run hello.dart`.
  (Roman Urdu: File ke naam ke aakhir mein .dart lazmi hai, warna computer use Dart nahi samjhega.)
- Every Dart program starts inside `void main() { }` — think of it as the door of the house: execution enters from here, line by line, top to bottom.
- Every statement ends with a **semicolon** `;` — it tells Dart "this instruction is finished." It is the most-forgotten character in beginner code.
- `//` **starts a comment** — a note for humans. Dart ignores everything after `//` on that line. Good programmers write comments to explain *why*, not *what*.
- `print()` shows a line on the console. Text goes inside single quotes: `print('Assalam o Alaikum');`.
- The loop of your life from today: type code → **Run** → read the console → fix → run again.

Code examples

Example 1 — The first program, with comments

```
// Yeh mera pehla Dart program hai
// Dart hamesha main() se shuru hota hai

void main() {
  print('Assalam o Alaikum, duniya!');
}
```

What it does: The two lines starting with `//` are comments — Dart skips them completely. The real work is one line: `print` shows the text on the console.

What you will see: in DartPad's console (right side), one line — `Assalam o Alaikum, duniya!` If you delete the semicolon and run again, DartPad underlines the error in red and refuses to run — that red message is your teacher, not your enemy. Read it; it usually names the exact line.

Example 2 — Many lines, in order

```
void main() {
  print('Name: Ali Raza');
  print('City: Sahiwal');
  print('Goal: Flutter developer');
}
```

What it does: Three `print()` statements run one after another, in the exact order written. Dart never shuffles your lines — the program is a recipe followed step by step.

What you will see: three lines on the console, in the same order: Name, then City, then Goal. Put the Goal line first in the code and run again — the console order changes with it. You control the order; Dart only follows.

Example 3 — The same program on your own computer

```
dart --version
dart run hello.dart
```

What it does: Save Example 1 in a file named `hello.dart`, open a terminal in that folder, and run these two commands. The first command proves the Dart SDK is installed (it answers with its version number). The second runs your file — `dart run` reads `hello.dart`, starts at `main()`, and prints to the terminal.

What you will see: a version line like `Dart SDK version: 3.x.x`, then the same greeting line as DartPad — `Assalam o Alaikum, duniya!` Same language, same program; only the stage changed.

Quick recap

- DartPad = zero-install practice in the browser. Dart SDK = the same language on your own computer.
- A Dart file ends in `.dart` and runs with `dart run file.dart`.
- Every program starts at `main()`; every statement ends with `;`.
- `//` = comment (for humans, ignored by Dart). `print()` = show a line on the console.

Common mistakes

1. **Forgetting the semicolon** — the number-one beginner error. Dart stops and points at the line. Before asking anyone, check the end of the line above the error for a missing `;`.
2. **Writing text without quotes** — `print(Ali);` asks Dart for a *box named* Ali, which does not exist. Text always wears quotes: `print('Ali');`.
3. **Saving the file with a wrong name** — `hello.txt` or `hello.dart.txt` will not run as Dart. The name must end in `.dart`, exactly like HTML files had to end in `.html`.

Practice

1. In DartPad, write a program that prints your name, your class/semester and your university on three separate lines. Run it.
2. Add two comment lines at the top with today's date and the lesson number. Then remove one semicolon on purpose, run, and read the error message carefully — then fix it and run again.

Practice quiz

1. What is DartPad, and what does it need?

- A paid app you must install
- A browser tool: code on the left, Run on top, console on the right, no download or setup
- A phone emulator only
- A Dart database

Answer: A browser tool: code on the left, Run on top, console on the right, no download or setup

2. How do you run a Dart file called hello.dart on your own computer?

- dart hello.dart
- dart run hello.dart
- run hello.dart
- open hello.dart in a browser

Answer: dart run hello.dart

3. What does the semicolon ; do at the end of a Dart statement?

- It starts a comment
- It tells Dart this instruction is finished
- It prints the line
- It is decoration and can be left out

Answer: It tells Dart this instruction is finished

Watch the video lesson: https://youtu.be/7OJLTLhX-ss

Read online: https://scopetreasure.com/notes/dart-setup-first-program · More free tools and notes at https://scopetreasure.com