ScopeTreasure Courses

DART COURSES · LESSON D.3 · CODE SCOPE SERIES

Variables & Data Types

int, double, String and bool — labelled boxes for your data, var, final vs const, and $ string interpolation.

Introduction

An app is data at work: names, marks, prices, yes-or-no answers. Dart keeps every piece of data in a **variable** — a labelled box — and every box has a **type** that says what kind of thing may go inside. Get variables right and every later lesson (loops, functions, classes, Flutter screens) becomes easy, because all of them move data in and out of these boxes. *(Roman Urdu: Variable aik label wala dabba hai — type batata hai ke dabbe mein kis qisam ki cheez jayegi.)*

Key points

- The four types you will use daily:
  - `int` — whole numbers: `int age = 20;`
  - `double` — decimal numbers: `double height = 5.9;`
  - `String` — text, always in quotes: `String name = 'Ali';`
  - `bool` — only `true` or `false`: `bool isStudent = true;`
- You **declare** a variable by writing its type, a name, `=`, and the value: `int marks = 92;`. After that, use just the name — `print(marks);` shows the *value*, while `print('marks');` shows the word "marks". That one difference (quotes or no quotes) decides everything.

  (Roman Urdu: Naam baghair quotes ke likho to dabbe ki value aati hai; quotes ke sath wohi lafz chhap jata hai.)
- `var` lets Dart guess the type from the first value: `var city = 'Sahiwal';` becomes a String forever. After that, `city = 5;` is an **error** — a box never changes its type.
- `final` = the value is set once and never changes, but it can be decided when the program runs (for example, from user input). `const` = the value must be known *before* the program runs (like `const pi = 3.14159;`).
- **String interpolation** puts values inside text: `$name` for a simple variable, `${maths + urdu}` when a small calculation is needed inside the text. Use this instead of joining pieces with `+`.
- Variable names use **lowerCamelCase**: first word small, next words capitalised — `studentName`, `totalMarks`. Names must describe the thing (`marks` yes, `x` no), because you will read this code again after a month.

Code examples

Example 1 — The four types in one program

```
void main() {
  String name = 'Ali';
  int age = 20;
  double height = 5.9;
  bool isStudent = true;

  print(name);
  print(age);
  print(height);
  print(isStudent);
}
```

What it does: Four boxes, one of each type, each printed on its own line. Notice the values wear no quotes in the code except the String — text is the only type that needs them.

What you will see: four lines — `Ali`, `20`, `5.9`, `true`. Try `int age = 'twenty';` and Dart refuses to run: a word cannot live in a number box. That strictness is Dart protecting you, and it is called **type safety**.

Example 2 — Interpolation: values inside a sentence

```
void main() {
  String name = 'Sara';
  int maths = 92;
  int urdu = 85;

  print('My name is $name');
  print('Total marks: ${maths + urdu}');
}
```

What it does: `$name` drops the variable's value straight into the sentence. `${ ... }` runs a small calculation first (92 + 85) and drops the *answer* in. No `+` joining, no commas — the sentence reads like normal English.

What you will see: two lines — `My name is Sara` and `Total marks: 177`. If you wrote `'My name is name'` (no `$`), the console would show the word "name" — the `$` is what makes it a value, not a word.

Example 3 — var, final and const together

```
void main() {
  var city = 'Sahiwal'; // Dart khud samjha: yeh String hai
  city = 'Multan'; // OK — ab bhi String hi hai
  print(city);

  final country = 'Pakistan'; // aik dafa set, phir change nahi
  const pi = 3.14159; // pehle se maloom value

  print(country);
  print(pi);
}
```

What it does: `var` guesses String from the first value, so a new city *name* is fine but a number would be an error. `final` locks the country after one setting. `const` locks pi at the value everyone already knows.

What you will see: three lines — `Multan`, `Pakistan`, `3.14159`. Now try adding `country = 'India';` — Dart stops you before running, because `final` was a promise. In app code, that promise prevents a whole family of bugs.

Quick recap

- Variable = labelled box; type = what may live in the box: `int`, `double`, `String`, `bool`.
- `print(marks)` shows the value; `print('marks')` shows the word. Quotes decide.
- `var` = Dart guesses the type once; it never changes after that.
- `final` = set once at run time. `const` = known before run time.
- `$name` and `${calculation}` = interpolation — values inside sentences.

Common mistakes

1. **Putting a value in the wrong type of box** — `int age = '20';` looks harmless but the quotes make it a String, and Dart errors. Numbers do not wear quotes; text always does.
2. **Changing a var to another type later** — `var score = 10;` then `score = 'ten';` is an error. `var` is not "any type forever"; it is "type decided on line one."
3. **Joining sentences with + instead of interpolation** — `print('Name: ' + name + ', Age: ' + age);` even *fails* when age is an int, and is hard to read when it works. Write `print('Name: $name, Age: $age');` and let Dart do the joining.

Practice

1. Make variables for a shop item: name (String), price (int), discount percent (double), in stock (bool). Print all four with labels, using interpolation in one sentence each.

2. Write a program with `final` for your university name and `const` for the days in a week. Then try to change both — read Dart's two different error messages and note which keyword stopped you.

Practice quiz

1. Which Dart type holds a decimal number like 5.9?

- int
- double
- String
- bool

Answer: double

2. What happens after var city = 'Sahiwal'; if you then write city = 5;?

- It works, var means any type forever
- It is an error, var fixes the type from the first value and it never changes
- It changes city to a number and forgets the text
- It prints a warning but runs

Answer: It is an error, var fixes the type from the first value and it never changes

3. What is the difference between final and const?

- There is no difference
- final is set once when the program runs, const must be known before the program runs
- const can change later, final cannot
- final is for numbers only

Answer: final is set once when the program runs, const must be known before the program runs

---

Watch the video lesson: https://youtu.be/CCdrT2Hi9Pc

Read online: https://scopetreasure.com/notes/dart-variables-data-types · More free tools and notes at https://scopetreasure.com