Functions

Function declaration and invocation | Return types in functions | Arrow functions (=>)

What is Function in Programming?

- A function is a block of code designed to perform a specific task.
- It can take inputs (parameters), process them, and return an output.
- Functions help to break down complex problems into smaller, reusable parts.
- They are also known as methods, procedures, or subroutines in some languages.

```
function greet() {
  print("Hello, World!");
}
```

Why Are Functions Needed?

- **Reusability:** Write once, use multiple times.
- **Organization:** Keeps code modular and easier to read.
- **Debugging:** Errors are easier to find in smaller code blocks.
- **Maintainability:** Updating logic in one place updates all usages.
- **Team Collaboration:** Different team members can work on separate functions.

Example:
Instead of repeating the same calculation multiple times, define a single function and reuse it.

What is a Function in Dart?

- In Dart, a function is an object that represents an operation or behavior.
- You can store a function in a variable, pass it as a parameter, or return it from another function.
- Functions are defined using the void or return type, followed by a name and parameter list.

Example:

```dart
void greetUser(String name) {
  print('Hello, $name!');
}
```

Types of Functions in Dart

- **Built-in Functions:** Provided by Dart (e.g., `print()`, `main()`).
- **User-defined Functions:** Written by developers to perform specific tasks.
- **Anonymous Functions (Lambdas):** Functions without a name.
- **Arrow Functions:** Short-hand syntax for single-expression functions.
- **Higher-Order Functions:** Take functions as parameters or return functions.

Function Declaration and Invocation

Declaration: Writing a function.
Invocation (Call): Using the function to execute it.

Example:

```
// Declaration
void sayHello() {
  print('Hello!');
}

// Invocation
sayHello();
```

Return Types in Functions

- A function can **return a value** using the `return` keyword.
- The return type must match the declared type.
- If nothing is returned, use `void`.

Example:

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

void showMessage() {
  print('Task completed');
}
```

Arrow Functions (=>)

- Used for **short, single-line** functions.
- Syntax: `returnType functionName(parameters) => expression;`
- Automatically returns the expression result.

Example:

```dart
int add(int a, int b) => a + b;

void greet(String name) => print('Hi $name!');
```

Summary

- Functions organize code into **reusable blocks**.
- Dart supports **multiple types**: user-defined, anonymous, arrow, and more.
- **Return types** define output, while **arrow syntax** simplifies short functions.
- Functions make programs **modular**, **readable**, and **efficient**.