

Dart – Functions

Girma B.



Function

- Function is a set of codes that together perform a specific task.
- It is used to break the large code into smaller modules and reuse it when needed.
- Functions make the program more readable and easy to debug.
- It improves the modular approach and enhances the code reusability.

Function

- Suppose, we write a simple calculator program where we need to perform operations number of times when the user enters the values.
- We can create different functions for each calculator operator. By using the functions, we don't need to write code for adding, subtracting, multiplying, and divide again and again.
- We can use the functions multiple times by calling.

Function - why?

- It increases the module approach to solve the problems.
- It enhances the re-usability of the program.
- We can do the coupling of the programs.
- It optimizes the code.
- It makes debugging easier.
- It makes development easy and creates less complexity.

Function - Definition

- A function can be defined by providing the name of the function with the appropriate parameter and return type. A function contains a set of statements which are called function body. The syntax is given below.

- Syntax:

```
return_type func_name (parameter_list):  
{  
    //statement(s)  
    return value;  
}
```

Function - Example

```
int mul(int a, int b){  
    int c;  
    c = a+b;  
    print("The sum is:${c}");  
}
```

Function - Parameters

- When a function is called, it may have some information as per the function prototype is known as a parameter (argument).
- The number of parameters passed and data type while the function call must be matched with the number of parameters during function declaration. Otherwise, it will throw an error.
- Parameter passing is also optional, which means it is not compulsory to pass during function declaration. The parameter can be two types.
- Actual Parameter - A parameter which is passed during a function definition is called the actual parameter.
- Formal Parameter - A parameter which is passed during a function call is called the formal parameter.

Function - Parameters


- When a function is called, it may have some information as per the function prototype is known as a parameter (argument).
- The number of parameters passed and data type while the function call must be matched with the number of parameters during function declaration. Otherwise, it will throw an error.
- Parameter passing is also optional, which means it is not compulsory to pass during function declaration. The parameter can be two types.
- Actual Parameter - A parameter which is passed during a function definition is called the actual parameter.
- Formal Parameter - A parameter which is passed during a function call is called the formal parameter.

Function - Parameters

- A function always returns some value as a result to the point where it is called. The return keyword is used to return a value.
- The return statement is optional. A function can have only one return statement. The syntax is given below.

```
int sum (int a, int b){  
    .....  
    .....  
    return result;  
}
```

```
var c = sum(30,20)
```



Function - Anonymous

- Dart also provides the facility to specify a nameless function or function without a name. This type of function is known as an anonymous function, lambda, or closure.
- An anonymous function behaves the same as a regular function, but it does not have a name with it. It can have zero or any number of arguments with an optional type annotation.
- We can assign the anonymous function to a variable, and then we can retrieve or access the value of the closure based on our requirement.

Function - Anonymous

- An Anonymous function contains an independent block of the code, and that can be passed around in our code as function parameters. The syntax is as follows.

Syntax:

```
(parameter_list) {  
    statement(s)  
}
```

Function - Anonymous

```
void main() {  
    var list = ["Ram","Raj","Rani","Ritu"];  
    print("Example of anonymous function");  
    list.forEach((item) {  
        print('${list.indexOf(item)}: $item');  
    });  
}
```

Thank you