



Função

É um conjunto de instruções que recebem entradas, fazem alguns cálculos específicos e produzem saída. As funções são criadas quando certas instruções ocorrem repetidamente no programa e uma função é criada para substituí-las. As funções facilitam a divisão do programa complexo em subgrupos menores e aumentam a capacidade de reutilização do código do programa.

Definindo a função:

```
function_name define ( Parametros) {  
    Corpo da funcao  
    return valor;  
}
```

- `function_name` define o nome da função.
- `return_type` define o tipo de dados em que a saída virá.
- `return value` define o valor a ser retornado da função.

Maneira que a função é chamada no escopo do programa

```
function_name (argument_list);
```

- `function_name` define o nome da função.
- `argument_list` é a lista dos parâmetros que a função requer.

Exemplo: função completa

```
int add(int a, int b)
{
    // Creating function
    int result = a + b;
    // returning value result
    return result;
}

void main()
{
    // Calling the function
    var output = add(10, 20);

    // Printing output
    print(output);
}
```

Run

Console

30

Observação:

Você deve observar que duas funções não podem ter o mesmo nome de função, embora sejam diferentes nos parâmetros.

Nota:

Você deve observar que também pode retornar diretamente string ou inteiro ou qualquer saída de expressão por meio da instrução return.

Funções com parâmetro opcional:

(Muito utilizado no Flutter)

Há também um sistema de parâmetro opcional no Dart que permite ao usuário fornecer alguns parâmetros opcionais dentro da função.

Parâmetro	Descrição
Parâmetro posicional opcional	Para especificá-lo, use colchetes ('[]')
Parâmetro nomeado opcional	Quando passamos este parâmetro, é obrigatório passá-lo enquanto passamos os valores. É especificado por colchetes ('{}').

Os parâmetros passados de maneira “Simples” que esteja somente entre os parênteses “()” utilizasse somente o tipo do parâmetro e o nome do parâmetro que seria o parâmetro posicional.

Os Parâmetro posicional opcional deve usar os colchetes “[]” e não precisa chamar eles quando passar como parâmetro.

Já os Parâmetro nomeado opcional deve se utilizar “{}” que quando for passar o parâmetro deve chamar pelo nome e atribuir o valor dela (nome : “Vitor”)

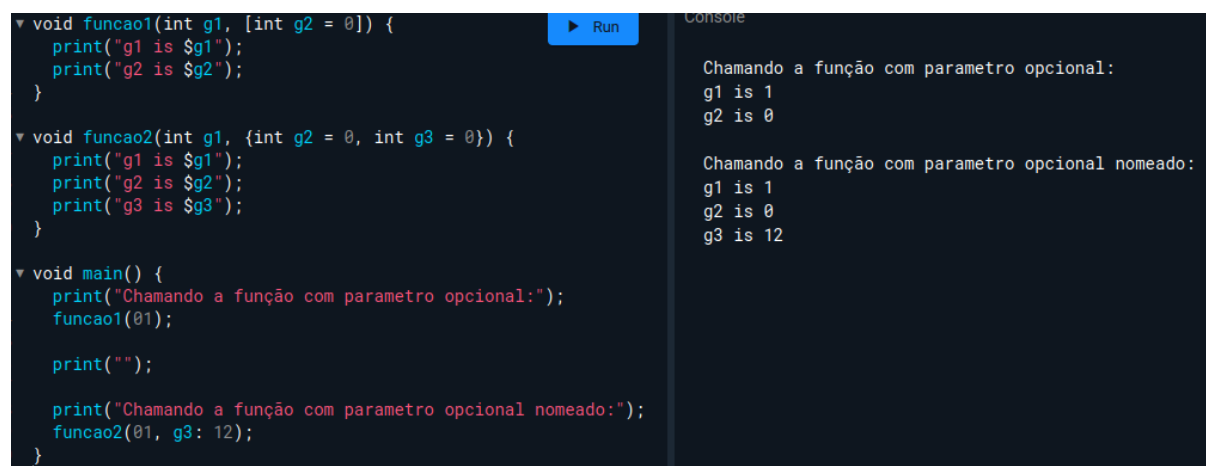
Dentro desses parâmetros também podemos utilizar como um parâmetro posicional requerindo ele sem precisar passar um valor padrão.

Ex:

```
void mostrarMensagem(  
    String nome, {  
        int idade = 0,  
        required String sexo,  
    }) {  
}
```

Ps: Quando passado parâmetros opcionais deve se passar o valor padrão se não for utilizado (Por causa do Null Safety das variaveis/objetos)

Exemplo:



The screenshot shows a code editor with the following Dart code:

```
void funcao1(int g1, [int g2 = 0]) {  
    print("g1 is $g1");  
    print("g2 is $g2");  
}  
  
void funcao2(int g1, {int g2 = 0, int g3 = 0}) {  
    print("g1 is $g1");  
    print("g2 is $g2");  
    print("g3 is $g3");  
}  
  
void main() {  
    print("Chamando a função com parametro opcional:");  
    funcao1(01);  
  
    print("");  
  
    print("Chamando a função com parametro opcional nomeado:");  
    funcao2(01, g3: 12);  
}
```

The console output on the right shows the following:

```
Chamando a função com parametro opcional:  
g1 is 1  
g2 is 0  
  
Chamando a função com parametro opcional nomeado:  
g1 is 1  
g2 is 0  
g3 is 12
```

Função recursiva:

A função recursiva são aquelas funções nas quais a função chama a si mesma. É uma boa maneira de evitar chamar repetidamente a mesma função para obter a saída.

Exemplo: função recursiva para série de Fibonacci.

```
1 // Computes the nth Fibonacci number.
2 int fibonacci(int n) {
3     // This is recursive function as it calls itself
4     return n < 2 ? n : (fibonacci(n - 1) + fibonacci(n - 2));
5 }
6
7 void main() {
8     var i = 20; // input
9     print('fibonacci($i) = ${fibonacci(i)}');
10 }
11
```

Run

Console

fibonacci(20) = 6765

Função Lambda:

Eles são a forma abreviada de representar uma função no Dart. Eles também são chamados de função de seta. Mas você deve notar que com a função lambda você pode retornar o valor para apenas uma expressão.

Exemplo

```
1
2 // Função Lambda em Dart
3 void gfg() => print("Olá Mundo");
4
5 void main() {
6     gfg(); // Chamando função Lambda
7 }
8
```

Run

Console

Olá Mundo