

Funções.

Macete (Dica)

Passo a Passo no VS Code:

Abra as Configurações de Snippets:

No Windows/Linux: Ctrl + Shift + P

No Mac: Cmd + Shift + P

Digite: "Configure User Snippets" e aperte Enter.

Escolha a Linguagem:

Digite "dart" na lista que aparecer e selecione o arquivo dart.json.

Cole este Código:

Apague o que estiver lá e cole este modelo (ou adicione-o):

JSON

```
{
  "Main Simples": {
    "prefix": "main",
    "body": [
      "void main() {",
      "  $0",
      "}"
    ],
    "description": "Cria uma função main sem argumentos"
  }
}
```

Na programação, uma função é um bloco de código que realiza uma tarefa específica. Em vez de escrever o mesmo código dez vezes, você o "guarda" dentro de uma função e a chama pelo nome sempre que precisar.

Por que usar funções?

- **Organização:** O código fica mais limpo.
- **Reutilização:** Escreva uma vez, use sempre.
- **Facilidade:** Se precisar consertar um erro, você mexe em apenas um lugar.

Anatomia de uma Função em Dart

Para criar uma função, seguimos esta estrutura:

```
tipoDeRetorno nomeDaFuncao(Tipo parametro1, Tipo parametro2) {  
    // Código que será executado  
    return valorDeSaida;  
}
```

- Tipo de Retorno: O tipo de dado que a função devolve (int, String, double). Se não devolver nada, usamos void.
- Nome: Como você chamará a função (ex: calcularMedia).
- Parâmetros: São os dados que a nossa função deve trabalhar
- return: A palavra que entrega o resultado final

Exemplos Práticos

Exemplo 1: Função sem retorno (void)

Usada quando queremos apenas que o computador execute algo, como imprimir na tela.

```
funcao01.dart X  
1 void saudar(){  
2     print ("Ola mundo");  
3 }  
4  
Run | Debug  
5 void main() {  
6     saudar();  
7 }  
8 }  
9
```

Exemplo 2: Função com parâmetros e retorno

Uma função se soma simples.

```
funcao02.dart X  
Run | Debug  
1 void main() {  
2     int soma = somar(10, 5);  
3     print("Resultado da soma $soma");  
4 }  
5  
6 int somar(int n1, int n2) {  
7     return n1 + n2;  
8 }  
9
```

Exercícios

Exercício 01: Calculadora de Área de um retângulo, com entrada de dados.

Enunciado: Crie uma função que receba a largura e a altura de um retângulo e retorne a sua área.

```
funcao03.dart X
1 // função que calcula área
2 import 'dart:io';
3
4 double areaQuadrado(double largura, double altura) {
5     return largura * altura;
6 }
7
8 void main() {
9     print("");
10    print("*****Cálculo Área Quadrado *****");
11    print("");
12
13    stdout.write("Digite a altura do quadrado: ");
14    String? entrada_altura = stdin.readLineSync();
15    double altura = double.parse(entrada_altura!);
16
17    stdout.write("Digite a largura do quadrado: ");
18    String? entrada_largura = stdin.readLineSync();
19    double largura = double.parse(entrada_largura!);
20
21    double resultado = calcularArea(largura, altura);
22    print("A área do retangulo é $resultado");
23 }
24
25 double calcularArea(double largura, double altura) {
26     return largura * altura;
27 }
28
```

```
String? entrada = stdin.readLineSync(); // O '?' aceita que pode vir nulo
double largura = double.parse(entrada!); // O '!' diz: "Eu garanto que não é nulo"
```

Exercício 02: Vamos criar um programa onde será digitado o nome e idade de uma pessoa, terá uma função que imprimirá o nome, idade e se esta pessoa é maior ou menor de idade.

```

1  import 'dart:io';
2
3  void verificarIdade(String nome, int idade) {
4      if (idade >= 18) {
5          String mensagem = "Ola $nome, sua idade é $idade, você é maior de idade";
6          print(mensagem);
7      } else {
8          String mensagem = "Ola $nome, sua idade é $idade, você é menor de idade";
9          print(mensagem);
10     }
11 }
12
13
14 void main() {
15     stdout.write("Digite nome da pessoa: ");
16     String nome = stdin.readLineSync();
17     stdout.write("Digite a idade da pessoa: ");
18     String? entradaIdade = stdin.readLineSync();
19     int idade = int.parse(entradaIdade!);
20     verificarIdade(nome, idade);
21 }
22

```

As **Arrow Functions** em Dart (frequentemente chamadas de sintaxe de expressão ou "fat arrow") são uma maneira concisa de escrever funções que contêm apenas **uma única expressão**.

1. Sintaxe Fundamental

A estrutura troca as chaves {} e a palavra-chave return pelo símbolo =>.

Comparação Visual

Função Tradicional:

```

funcao_arrow.dart 1
1  void main() {
2      print(somar(2, 5));
3  }
4
5  //Função Tradicional:
6  int somar(int a, int b) {
7      return a + b;
8  }
9
10
11 //Arrow Function:
12 int somar(int a, int b) => a + b;
13

```