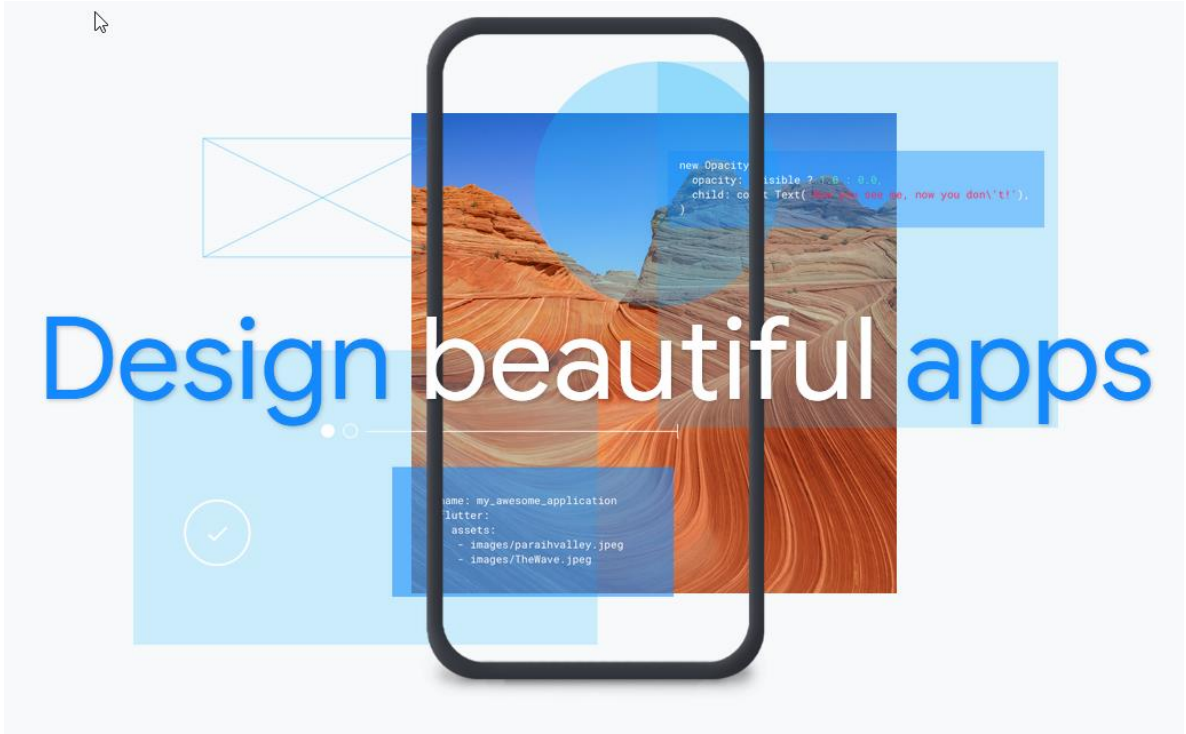
Transformando o futuro das pessoas
e as pessoas para o futuro.

#Senacfaz75

Senac

Desenvolvimento Mobile: Flutter



O que é o Flutter?

Flutter é um kit de desenvolvimento de interface de usuário (UI toolkit), de código aberto, criado pelo Google, que possibilita a criação de aplicativos compilados nativamente. Atualmente pode compilar para Android, iOS, Windows, Mac, Linux, Google *Fuchsia e Web.

O que é o Dart?

Os aplicativos Flutter são escritos na linguagem de programação Dart e fazem uso de muitos dos recursos mais avançados da linguagem.

No Windows, macOS e Linux, por meio do projeto semi-oficial Flutter Desktop Embedding, o Flutter é executado na máquina virtual Dart, que possui um mecanismo de compilação que ocorre em tempo de execução. Ao escrever e depurar um aplicativo, o Flutter usa a compilação JIT, permitindo o "hot reload", com a qual as modificações nos arquivos de origem podem ser injetadas em um aplicativo em execução. O Flutter estende isso com suporte para hot reload de widgets stateful, onde na maioria dos casos as alterações no código-fonte podem ser refletidas imediatamente no aplicativo em execução, sem a necessidade de uma reinicialização ou perda do Estado.

As versões de lançamento dos aplicativos Flutter são compiladas com a compilação antecipada (AOT) no Android e no iOS, possibilitando o alto desempenho do Flutter em dispositivos móveis.

*Fuchsia é um sistema operacional atualmente sendo desenvolvido pelo Google. Ao contrário de sistemas operacionais anteriores desenvolvidos pelo Google, como o Chrome OS e o Android, que são baseados no kernel Linux, Fuchsia é baseado em um novo microkernel chamado Zircon (o nome anterior era Magenta), derivado do Little Kernel, que foi destinado para sistemas embarcados e é principalmente escrito em C. Fuchsia foi projetado para ser executado em uma infinidade de dispositivos, incluindo telefones celulares e computadores pessoais.

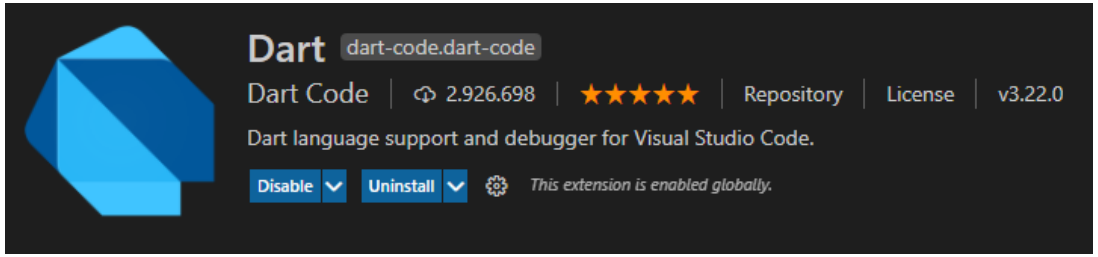
Instalações:

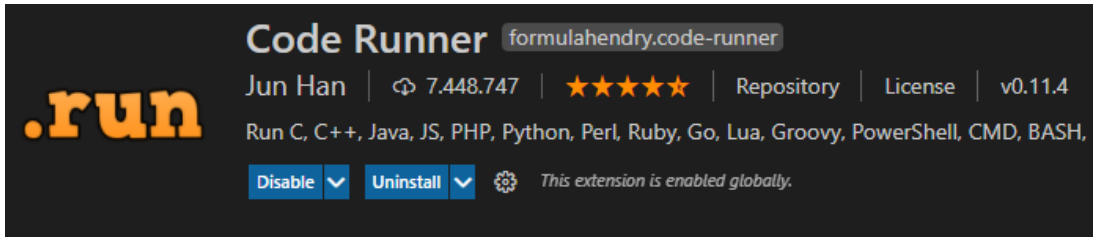
Java SE
https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html

Flutter
https://flutter.dev/docs/get-started/install/windows

OBS: Colocar a pasta bin do Flutter no Path do Windows

Extensões VS Code





Null Safety

A partir do Dart 2.12, temos a possibilidade de usar o Null Safety para facilitar nossas aplicações, dificultando erros de valores nulos. Aprenderemos a identificar erros de valores nulos, aplicar no nosso projeto do ByteBank, atualizar o Dart para a nova versão e migrar um projeto inteiro! Assim podemos criar códigos complexos com mais facilidade!

Erros de valor Null

Existem infinitos erros que podem acontecer por conta de um valor nulo:

- Você pode estar esperando um dado do seu back-end, mas ele não existe ainda...
- Você pode criar uma lista que muda de tamanho de acordo com a quantidade de produtos...

Fonte: Kako(Caio couto Moreira). Alura. Disponível em: https://www.alura.com.br/artigos/flutter-null-safety

Aula 1 – Hello dart e tipos de variáveis!

hellodart.dart > ...

```dart
//É preciso criar uma classe main para rodar o código Dart
Run | Debug
main() {
  //print é o comando de saída em Dart
  print('Hello dart!');
  print('Como vai?');

  //Tipos de variáveis em Dart
  //Dart faz inferência de tipo (Genérico)
  var nome = 'John Doe';
  //nome é do tipo String
  //Se eu tentar colocar outro valo nessa variável que não seja String
  //vai dar erro
  //var nome = 10; //Tipo já foi definido como String (ERRO)

  /**
   * Podemos também definir uma variável
   * já com o seu tipo
   */

  //Variável do tipo String
  String novoNome = 'Jane Doe'; //Estou tipando minha variável
  String CPF = '999.999.999.99-99';

  //Definindo um tipo numérico inteiro
  int ano = 1970;

  //Definindo um número decimal
  double altura = 1.77;

  //Definindo um tipo booleano (Verdadeiro ou falso)
  bool vf = true;
  bool fv = false;

  //Saída Interpolada, ao contrário do javascript
  //Não é reciso as {} e `` para interpolar valores
  print('Seu nome: $novoNome'); //Lembra PHP
  print('Nascimento: $ano');
  print('Sua altura: $altura');
  print('Resultado VF: $vf');
  print('Resultado FV: $fv');
}
```

```
Hello dart!
Como vai?
Seu nome: Jane Doe
Nascimento: 1970
Sua altura: 1.77
Resultado VF: true
Resultado FV: false

[Done] exited with code=0 in 1.359 seconds
```

Aula 1 – Estruturas de Dados Arrays (arrays.dart)

```dart
Run | Debug
void main() {
  //Arays são estruturas que armazenam mais de um valor
  //a uma variável

  //Criando um array
  var compras = ['Macarrão', 'Feijão', 'Pão', 'Manteiga'];

  //Exibindo o array
  print('Lista de compras: $compras');

  //Exibindo os itens pelo índice
  print('Nome do primeiro produto: ${compras[0]}');
  print('Nome do segundo produto: ${compras[1]}');
  print('Nome do terceiro produto: ${compras[3]}');

  //Item out of range
  //print('Nome do primeiro produto: ${compras[5]}');

  //Acessando e alterando um valor do array
  compras[0] = 'Arroz';
  //Exibindo os itens pelo índice
  print('Nome do primeiro produto: ${compras[0]}');

  //Criando array numérico
  var pares = [0, 2, 4, 6];

  //Exibindo o array
  print('Lista de números pares $pares');
  print('------------------------------------------------------');

  //Métodos utilizados em arrays
  /**
   * first() : Retorna o primeiro elemento do Array
   * last() : Retorna o último elemento do Array
   * isEmpty() : Retorna true se a lista está vazia, caso contrário, false.
   * length() : Retorna o tamanho do Array
   */

  var listaNomes = ['José Maria', 'Pedro da Silva', 'Cristina Pereira'];

  print('Primeiro nome: ${listaNomes.first}');
  print('Último nome: ${listaNomes.last}');
  print(
      'O Array está vazio? ${listaNomes.isEmpty}');
  print('Tamanho do Array: ${listaNomes.length}');
}
```

```
Lista de compras: [Macarrão, Feijão, Pão, Manteiga]
Nome do primeiro produto: Macarrão
Nome do segundo produto: Feijão
Nome do terceiro produto: Manteiga
Nome do primeiro produto: Arroz
Lista de números pares [0, 2, 4, 6]
------------------------------------------------------
Primeiro nome: José Maria
Último nome: Cristina Pereira
O Array está vazio? false
Tamanho do Array: 3
Exited
```

Aula 1 – Operdaores Aritméticos (oparitimeticos.dart)

```dart
Run | Debug
void main() {
  //Operadores aritméticos
  // + - * / %

  int a = 20;
  int b = 5;

  //Operador de Soma
  var soma = a + b;

  //Operador de Subtração
  var subt = a - b;

  //Operador de Multiplicação/Produto
  var produto = a * b;

  //Operador de divisão
  var divisao = a / b;

  //operador resto da divisão
  var restoDiv = a % b;

  //Saída
  print('Soma de $a + $b = $soma');
  print('Subtração de $a - $b = $subt');
  print('Produto de $a * $b = $produto');
  print('Divisão de $a / $b = $divisao');
  print('Resto da Divisão de $a % $b = $restoDiv');

  //Operdador resumido
  int num = 100;
  num += 20;
  num -= 10; //Pode ser * /

  //Saída
  print('Número: $num');

  //Incremento e decremento
  num++;
  num--;

  //Saída
  print('Número: $num');
}
```

```
Soma de 20 + 5 = 25
Subtração de 20 - 5 = 15
Produto de 20 * 5 = 100
Divisão de 20 / 5 = 4.0
Resto da Divisão de 20 % 5 = 0
Número: 110
Número: 110

[Done] exited with code=0 in 1.345 seconds
```

Aula 1 – Operadores Relacionais (oprelaconais.dart)

```dart
Run | Debug
void main() {
  //Operdaores Relacionais
  /**
   * ==, != (Igual e diferente)
   * >, < (Maior e menor)
   * >=, <= (Maio igual e Menor igual)
   */

  //Declarando variáveis
  int a = 20;
  int b = 5;

  //Verificando as variáveis
  print('$a = $b? Resultado: ${a == b}');
  print('$a ≠ $b? Resultado: ${a != b}');
  print('$a > $b? Resultado: ${a > b}');
  print('$a < $b? Resultado: ${a < b}');
  print('$a ≥ $b? Resultado: ${a >= b}');
  print('$a ≤ $b? Resultado: ${a <= b}');

  //Podemos atribuir esses resultados a uma variável
  bool igual = a == b;
  print('Verificação de igualdade: $igual');
}

```

```
20 = 5? Resultado: false
20 ≠ 5? Resultado: true
20 > 5? Resultado: true
20 < 5? Resultado: false
20 ≥ 5? Resultado: true
20 ≤ 5? Resultado: false
Verificação de igualdade: false

[Done] exited with code=0 in 1.508 seconds
```

Aula 1 – Operadores Lógicos (oplogicos.dart)

```dart
Run | Debug
void main() {
  //Operadores Lógicos
  /**
   * && (E) --> V && V = V
   * || (OU) --> F || F = F
   * ! Negação
   */

  //Declarando as variáveis
  int a = 20;
  int b = 5;
  int c = 7;

  //Verificando Verdadeiro
  bool proposicao1 = a > b;
  bool proposicao2 = b < c;

  //Saída
  //        V         V
  print('$a > $b && $b < $c - Resposta: ${proposicao1 && proposicao2}');
  //        V         F
  print('$a > $b || $b > $c - Resposta: ${proposicao1 || proposicao2}');

  //Verificando o Falso
  bool proposicao3 = a < b;
  bool proposicao4 = b > c;

  //Saída
  print('$a < $b && $b > $c - Resposta: ${proposicao3 && proposicao3}');
  print('$a > $b || $b > $c - Resposta: ${proposicao4 || proposicao4}');

  //Negar um valor
  bool v = true;
  bool f = false;

  //Saída
  print('Negando o V: ${!v}');
  print('Negando o F: ${!f}');
}
```

```
20 > 5 && 5 < 7 - Resposta: true
20 > 5 || 5 > 7 - Resposta: true
20 < 5 && 5 > 7 - Resposta: false
20 > 5 || 5 > 7 - Resposta: false
Negando o V: false
Negando o F: true

[Done] exited with code=0 in 1.404 seconds
```

Aula 1 – Condicional If Else (condicionais.dart)

```dart
Run | Debug
void main() {
  //Condicionais
  //Declaração de variáveis
  int a = 10;
  int b = 5;

  //Condicional simples, Else opcional
  if (a > b) {
    print('Informação Verdadeira!');
  } else {
    print('Informação Falsa');
  }

  //Declarando Variável
  String nome = 'john';

  //Condicional
  if (nome != 'Jane') {
    print('Os nomes são diferentes!');
  } else {
    print('Os nomes são iguais!');
  }

  //Condicionais encadeadas
  //Testando mais de uma condição
  //Declarando variável
  double media = 4;

  //Condicional
  if (media >= 7.5 && media <= 10) {
    print('O aluno passou de ano!');
  } else if (media >= 5 && media < 7.5) {
    print('Aluno em recuperação!');
  } else {
    print('Aluno Reprovado!');
  }

}
```

```
Informação Verdadeira!
Os nomes são diferentes!
Aluno Reprovado!
Exited
```

Aula 1 – Condicional Switch (condswitch.dart)

```dart
Run | Debug
void main() {
  print('1 - Numero par');
  print('2 - Maior número');
  print('3 - Sair');

  int opcao = 2;

  switch (opcao) {
    case 1:
      //Declaração
      int numero = 10;

      //Condicional
      if (numero % 2 == 0) {
        print('O número $numero é par!');
      } else {
        print('O número $numero é ímpar!');
      }
      break;

    case 2:
      //Declaração
      int a = 3;
      int b = 3;

      //Condicional
      if (a > b) {
        print('O número $a é maior que o número $b!');
      } else if (a < b) {
        print('O número $a é menor que o número $b!');
      } else {
        print('Os números são iguais!');
      }
      break;

    default:
      print('Fora do intervalo de opções!');
  }
}
```

```
1 - Numero par
2 - Maior número
3 - Sair
Os números são iguais!
Exited
```

Aula 1 – Loop While (while.dart)

```dart
//Loop While
Run | Debug
/**
 * O loop while executa as instruções cada vez que a
 * condição especificada é avaliada como verdadeira. Em outras palavras,
 * o loop avalia a condição antes que o bloco de código seja executado
 */
void main() {
  //Declarando uma Flag
  int contador = 0;

  while (contador <= 20) {
    print('Número: $contador');

    //Incrementando o contaador para o loop não ficar infinito
    contador++;

    //condicional para quebra o Loop
    //Podemos usar também o continue
    if (contador == 10) {
      print('Loop interrompido!');
      break;
    }
  }
}
```

```
Número: 0
Número: 1
Número: 2
Número: 3
Número: 4
Número: 5
Número: 6
Número: 7
Número: 8
Número: 9
Loop interrompido!
Exited
```

Aula 1 – Loop For (for.dart)

```dart
//Loop For
Run | Debug
/**
 * O laço For é uma implementação de um loop definido.
 * O loop for executa o bloco de código por um determinado número de vezes.
 * Ele pode ser usado para iterar sobre um conjunto fixo de valores,
 * como uma matriz.
 */

void main() {
  //Declaração
  int contador = 20;

  for (var i = 0; i < contador; i++) {
    print('Número: $i');
    if (i == 10) {
      print('Loop interrompido!');
      break;
      //Podemos usar o continue também
    }
  }
  print('------------------------------------------------');

  //Varrendo array
  //Definindo o Array
  var nomes = ['Bete', 'Ana', 'Pedro', 'João', 'Maria'];

  for (var i = 0; i < nomes.length; i++) {
    print('Nome: ${nomes[i]}');
  }
  print('------------------------------------------------');

  //Simulando Post de Filmes
  var filmes = [
    'Matrix',
    'Uma vida iluminada',
    'Teoria de Tudo',
    'Divertidamente'
  ];

  for (var filme in filmes) {
    print(filme);
  }
}
```

```
Número: 0
Número: 1
Número: 2
Número: 3
Número: 4
Número: 5
Número: 6
Número: 7
Número: 8
Número: 9
Número: 10
Loop interrompido!
------------------------------------------------
Nome: Bete
Nome: Ana
Nome: Pedro
Nome: João
Nome: Maria
------------------------------------------------
Matrix
Uma vida iluminada
Teoria de Tudo
Divertidamente

[Done] exited with code=0 in 2.107 seconds
```

Aula 2 – Funções (funcao_void.dart)

```dart
//Main() é uma função de inicialização de código dart
//Essa função é obrigatória
//Essa função é void, não retorna valor

//Exemplos
void mostrarNome() {
  print('John Doe');
}

void linha() {
  print('--------------------------------');
}

//Função com parâmetro
void calcularValores(String operador, double a, double b) {
  if (operador == '+') {
    double soma = a + b;
    print('Operador \"$operador\" a soma de $a + $b = $soma');
  } else if (operador == '*') {
    double produto = a * b;
    print('Operador \"$operador\" o produto de $a x $b = $produto');
  } else if (operador == '-') {
    double subt = a - b;
    print('Operador \"$operador\" a subtração de $a - $b = $subt');
  } else if (operador == '/') {
    double divisao = a / b;
    if (b == 0) {
      print('Divisão inválida!');
    } else {
      print('Operador \"$operador\" o produto de $a / $b = $divisao');
    }
  }
}

```

```dart
void main() {
  //Chamando a função
  mostrarNome();
  linha();

  calcularValores('+', 10, 5);
  linha();

  calcularValores('-', 3, 5);
  linha();

  calcularValores('*', 30, 2);
  linha();

  calcularValores('/', 5, 10);
  linha();
}

```

```
John Doe
--------------------------------
Operador "+" a soma de 10.0 + 5.0 = 15.0
--------------------------------
Operador "-" a subtração de 3.0 - 5.0 = -2.0
--------------------------------
Operador "*" o produto de 30.0 x 2.0 = 60.0
--------------------------------
Operador "/" o produto de 5.0 / 10.0 = 0.5
--------------------------------

[Done] exited with code=0 in 1.336 seconds
```

Aula 2 – Funções (funcao_retorno.dart)

```dart
//Funções com retorno

//Declaração
//Foi importado assim que eu chamei a função linha()
import 'funcao_void.dart';

double calcularMedia(double n1, double n2, double n3, double n4) {
  double media = (n1 + n2 + n3 + n4) / 4;
  return media;
}

double calcularDivisao(double n1, double n2) {
  double div = (n1 / n2);
  return div;
}

 //Função Resumida
  double calcularPorcentagem(percent, valor) => (percent * valor) / 100;

Run | Debug
void main() {
  double media = calcularMedia(10, 10, 10, 10);
  double divisao = calcularDivisao(10, 5);

  //Saída
  print('Média Aritmética');
  print('A média é $media');
  linha();
  print('A divisão é $divisao');
  linha();

////////////////////////////////////////////////////////////////////////

  //Declaração de variáveis
  double percent = 5;
  double valor = 100;

  //Executa a função e guarada o valor em uma variável
  double percentual = calcularPorcentagem(percent, valor);

  //Imprime 10% de 100
  print('$percent% de $valor = $percentual');
}
```

```
Média Aritmética
A média é 10.0
--------------------------------
A divisão é 2.0
--------------------------------
5.0% de 100.0 = 5.0
Exited
```

Aula 2 – Funções (funcao_param_opcional.dart)

```dart
//Parâmetro opcional

import 'dart:math';

Run | Debug
void main() {
  //declarar limite
  int numero = sortearNumero(1000);
  //Saída
  print('O número sorteado foi: $numero');

  ////////////////////////////////////////////////////
  int valor1 = 10;
  int valor2 = 20;

  //Segundo parâmetro é opcional
  int soma = somarValores(valor1, valor2);

  //Saída
  print('Resultado dos valores somados: $soma');
}

//Utilizamos os colchetes para determinar o parâmetro opcional
int sortearNumero([int limite = 3]) {
  //Se nenhum valor limite for passado
  //Será assumido o 3
  return Random().nextInt(limite);
}

//1 parâmetro obrigatório e 1 opcional
int somarValores(int v1, [int v2 = 0]) {
  //Se não informar o valor 2
  //o parâmetro assume 0
  print('Valor 1: $v1');
  print('Valor 2: $v2');
  return v1 + v2;
}
```

```
O número sorteado foi: 778
Valor 1: 10
Valor 2: 20
Resultado dos valores somados: 30
Exited
```

Aula 2 – Funções (funcao_param_nomeado_1.dart)

```dart
//Parâmetros nomeados em função
//ajuda na clareza do código
//Evitando confusão com os parâmetros posicionais

Run | Debug
void main() {
  //chamando a função com parâmetros nomeados
  exibirCadastro(salario: 1500, funcionario: 'John Doe', funcao: 'Gerente');
}

//Funções
exibirCadastro(
  //Utilizamos required por conta da nova funcionalidade do Dart
  //Assunto complexo: non-null
    {required String funcao,
    required double salario,
    required String funcionario}) {

  print('Nome do Funcionário: $funcionario');
  print('Função: $funcao');
  print('Salário: R\$$salario');
  return 'Tudo ok!';
}
```

```
Nome do Funcionário: John Doe
Função: Gerente
Salário: R$1500.0
Exited
```

Aula 2 – Funções (funcao_param_nomeada_2.dart)

```dart
//Função anônima e parâmetros opcionais
//Função com parâmetros opcionais
void avaliarFilme(String nomeFilme, {categoria, nota}) {

  //Verificar se o valor está nulo
  var n = nota ?? 0;
  var cat = categoria ?? 'Sem categoria';

  //Imprime os Dados
  print('Nome do Filme: $nomeFilme');
  print('Categoria: $cat');
  print('Nota: $n');

}

Run | Debug
void main() { // função obrigatória

  //Executar a função avaliarFilme()
  //avaliarFilme('Matrix');
  avaliarFilme('Matrix', categoria: 'Ficção', nota: 10);

}
```

```
Nome do Filme: Matrix
Categoria: Ficção
Nota: 10
Exited
```

Aula 2 – Funções (funcao_anonima.dart)

```dart
//Funções anônimas: Funções que podem ser
//armazenadas dentro de variáveis

import 'dart:math';

Run | Debug
void main() {
  //Criando e definindo o tipo

  //Minha função é do tipo int
  int Function(int, int) soma = somarFunction;

  //Minha função é do tipo int/double/String/Dynamic
  String Function(String, String) mostraTexto = mostrarTexto;

  //Uma outra forma de criar a função anônima
  //Minha função é do tipo int/double/String/Dynamic
  dynamic Function(double, double) potencia = (base, expoente) {
    return pow(base, expoente);
  };//Não esquecer

  //mais uma forma de criar esse tipo de função
  var produto = (double a, int b) {
    return a * b;
  }; //Não esquecer

  //Saída
  print('A soma dos valores é ${soma(10, 10)}');
  print('Frase: ${mostraTexto("Olá", " Mundo!")}');
  print('Resultado da potência é:  ${potencia(2, 3)}');
  print('Resultado da multiplicação é:  ${produto(10, 3)}');
}

//Funções
int somarFunction(int a, int b) {
  return a + b;
}

String mostrarTexto(String a, String b) {
  return a + b;
}
```

```
A soma dos valores é 20
Frase: Olá Mundo!
Resultado da potência é:  8.0
Resultado da multiplicação é:  30.0
Exited
```

Aula 2 – Funções (funcao_arrow.dart)

```dart
//Arrow function é uma forma reduzida de representar
//um função. Contra: são limitadas, não podem ser utilidas
//em todas as situações

Run | Debug
void main() {
  //Função anônima
  var produto = (int a, int b) {
    return a * b;
  };

  //Arrow Function
  //1 sentença de código apenas
  var divisao = (double a, double b) => a / b;
  var modulo = (double c, double d) => c % d;
  var media = (double n1, double n2, double n3) => (n1 + n2 + n3) / 3;

  //Saída
  print('O valor do produto é: ${produto(10, 5)}');
  print('O valor da divisão é: ${divisao(50, 2)}');
  print('O valor da divisão é: ${modulo(5, 3)}');
  print('A média calculada é: ${media(4, 5, 6)}');
}
```

```
O valor do produto é: 50
O valor da divisão é: 25.0
O valor da divisão é: 2.0
A média calculada é: 5.0
Exited
```

Aula 2 – Funções (funcao_tipo_dinâmico.dart)

```dart
//Dart é altamente tipado
//Cuidado ao criar suas variáveis
Run | Debug
void main() {

  //Criando tipos dinâmicos
  dynamic a = 1000;
  dynamic b = ' Mundo!';

  //Chamada da função e atribuição em uma variável
  dynamic uniao = concatenar(a, b);

  //Saída
  print('A união dos valores \"$a\" e \"$b\" é: $uniao');
}

//Função para juntar os valores de a e b
dynamic concatenar(param1, param2) {

  //método toString faz casting
  print(param1.toString() + param2.toString());

  //Retornando os valores
  return param1.toString() + param2.toString();
}
```

```
1000 Mundo!
A união dos valores "1000" e " Mundo!" é: 1000 Mundo!
Exited
```

Aula 2 – Funções (funcao_param1.dart)

```dart
//As funções pode também receber como parâmetros outroas funções

//Importando a biblioteca matemática
import 'dart:math';

Run | Debug
void main() {
  //Calcular par ou ímpar
  var par = () => print('Número Par');
  var impar = () => print('Número Ímpar');

  //Chamndo a função calcularParImpar
  calcularParImpar(par, impar);
}

//Funções
void calcularParImpar(Function calcularPar, Function calcularImpar) {
  //Criando número randômico
  //O VsCode importa da Biclioteca dar:math automáticamente
  var numero = Random().nextInt(50);
  print('Número randômico: $numero');
  //Verificando
  if (numero % 2 == 0) {
    calcularPar();
  } else {
    calcularImpar();
  }
}
```

```
Número randômico: 1
Número Ímpar
Exited
```

Aula 2 – Funções (funcao_param2.dart)

```dart
//Importando a biblioteca matemática
import 'dart:math';

//Criando uma função para encontrar o Delta
double calcularDelta(double a, double b, double c) {
  //Fórmula do Delta = b² - 4ac
  double delta = (b * b) - (4 * a * c);
  return delta;
}

//Criando uma função para encontrar os valores de X1 e X2
//Passo os valores das incógnitas e a função do Delta
dynamic calcularEquacao(double a, double b, double c, Function calcularDelta) {
  //Lista para a solucao
  var solucao = [];

  //Calculando as raízes
  var x1 = ((-b) + sqrt(calcularDelta(a, b, c))) / (2 * a);
  var x2 = ((-b) - sqrt(calcularDelta(a, b, c))) / (2 * a);

  //Inserindo os valores na lista
  solucao.add(x1);
  solucao.add(x2);
  return solucao;
}

Run | Debug
void main() {
  //Declarando as incógnitas
  double a = 1;
  double b = 2;
  double c = -15;

  //Chamando a função com parâmetros misturados
  var equacao = calcularEquacao(a, b, c, calcularDelta);

  //Saída
  print(equacao);
}
```

```
[3.0, -5.0]
Exited
```

Aula 2 – Funções (funcao_ret_funcao.dart)

```dart
//Com uma função retornando outra função
//podemos fazer com que alguma parte do nosso código
//seja executada em um outro momento
double Function(double) calcularJuros(double a) {

  //Função interna
  return (double b) { //Retornando uma função anônima
    return a * b;
  };

}

Run | Debug
void main() {

  //Passando os dois valores, a e b
  print(calcularJuros(2)(20));

  //Colocando a função em uma variável para otimizar o processo
  //passando 1 valor
  var juros = calcularJuros(.10);

  //Saída passando o valor de b
  print(juros(1500));
  print(juros(1000));
  print(juros(10000));
}
```

```
40.0
150.0
100.0
1000.0
Exited
```