

# Sistemas Embarcados

Aula 1 - Dart



# Introdução

Dart é uma linguagem de programação que foi criada pela Google pensando na praticidade de compilar seu software pra qualquer plataforma, seja mobile, desktop ou Web.

No caso de aplicações Web, o programa desenvolvido em Dart pode ser compilado como um código Javascript de maneira a ser suportado por navegadores. No caso de aplicações para arquiteturas ARM (smartphones) ou x64 (desktop) o programa desenvolvido em Dart será compilado para o código binário específico da arquitetura.

Dart.dev

Preview the future of Dart and Flutter with the [Dart 3 alpha release](#) and on-demand content from [Flutter Forward](#).



Google uses cookies to deliver its services, to personalize ads, and to analyze traffic. You can adjust your privacy controls anytime in your [Google settings](#). [Learn more.](#)

Okay

# Tipos de Dados Básicos

Os tipos de dados básicos do Dart são semelhantes aos tipos da linguagem C e estendido com algumas estruturas de Dados. São elas:

- Number (int e double)
- String
- Boolean (bool)
- List (também conhecido como Array ou Vetor)
- Set
- Map
- Rune (semelhante ao char do C)
- Symbol
- Null

Built-in Types

# Programa simples

```
// Define a function.
void printInteger(int aNumber) {
    print('The number is $aNumber.'); // Print to console.
}

// This is where the app starts executing.
void main() {
    var number = 42; // Declare and initialize a variable.
    printInteger(number); // Call a function.
}
```

# Outras brincadeiras com funções

## First Class Object

```
String upperMessage(String msg) ⇒ '!!! ${msg.toUpperCase()} !!!'
```

```
var loudify = upperMessage;  
assert(loudify('hello') = '!!! HELLO !!!'); // true
```

## Função Anônima

```
const list = ['apples', 'bananas', 'oranges'];  
const mappedList = list.map((item) {  
  return item.toUpperCase();  
}) // ['APPLES', 'BANANAS', 'ORANGES']
```

```
mappedList.forEach((item) {  
  print('$item: ${item.length}');  
});
```

# Listas

São semelhantes aos vetores/arrays da linguagem C, porém com uma sintaxe um pouco diferente e algumas funções built-in já implementadas

```
var list = [  
  'Car',  
  'Boat',  
  'Plane',  
];
```

```
var list = [1, 2, 3];  
assert(list.length == 3);  
assert(list[1] == 2);  
  
list[1] = 1;  
assert(list[1] == 1);
```

## Métodos da Lista

Strings também são consideradas vetores de caracteres (ou runes) em Dart. Além disso, Dart suporta o **spread operator**.

```
var list = [1, 2, 3];  
var list2 = [0, ...list];  
assert(list2.length == 4); // true
```

Outra mágica do Dart são as **collection if** e **collection for**

```
var nav = ['Home', 'Furniture', 'Plants', if (promoActive) 'Sale'];  
  
var listOfInts = [1, 2, 3];  
var listOfStrings = ['#0', for (var i in listOfInts) '#$i'];  
assert(listOfStrings[1] == '#1'); // true
```

# Sets ou Conjuntos

---

São parecidos com as listas, porém não aceitam elementos repetidos e não são ordenados. Sets funcionam como os conjuntos da matemática e permitem operações de interseção e união.

```
var halogens = {'fluorine', 'chlorine', 'bromine', 'iodine'}
var names = <String>{}
```

```
final characters1 = <String>{'A', 'B', 'C'};
final characters2 = <String>{'A', 'E', 'F'};
final intersectionSet = characters1.intersection(characters2)
print(intersectionSet); // {A}
```

# Maps

---

É uma estrutura de dado baseado em chave-valor que não é ordenado e costuma ser mais rápido pra procurar um valor do que as listas

```
var gifts = {
  // Key:    Value
  'first': 'partridge',
  'second': 'turtledoves',
  'fifth': 'golden rings'
};
```

```
var nobleGases = {
  2: 'helium',
  10: 'neon',
  18: 'argon',
};
```

```
assert(gifts['second'] == 'turtledoves'); // true
```

# Estruturas de decisão

## If e Else

```
if (isRaining()) {  
    you.bringRainCoat();  
} else if (isSnowing()) {  
    you.wearJacket();  
} else {  
    car.putTopDown();  
}
```

## Switch e Case

```
var command = 'OPEN';  
switch (command) {  
    case 'CLOSED':  
        executeClosed();  
        break;  
    case 'PENDING':  
        executePending();  
        break;  
    case 'APPROVED':  
        executeApproved();  
        break;  
    case 'DENIED':  
        executeDenied();  
        break;  
    case 'OPEN':  
        executeOpen();  
        break;  
    default:  
        executeUnknown();  
}
```



# Estruturas de repetição

## Loop For

```
var list = [];  
for (var i = 0; i < 3; i++) {  
    list.add(i++);  
    print(`$list[i]`);  
}  
  
for (final v in list) { // list = [1, 2, 3]  
    var temp = v++;  
    print(`$temp`);  
}
```

## forEach()

```
var collection = [1, 2, 3];  
collection.forEach(print); // 1 2 3
```

## While

```
while (!isDone()) {  
    doSomething();  
}
```

## Do - While

```
do {  
    printLine();  
} while (!atEndOfPage());
```

# Entrada de dados

Para ler dados do terminal digitado pelo usuário precisamos importar a biblioteca **dart:io** e usar a a função

```
`stdin.readLineSync()`
```

```
import 'dart:io';

void main() {
  print("Qual a sua idade?");
  String? input = stdin.readLineSync();
  if (input != null) {
    int idade = int.parse(input);
    print("Ano que vem sua idade será ${idade + 1} anos.");
  } else {
    print("Não foi possível calcular o valor da idade");
  }
}
```

# Exercícios

1. Desenvolva um programa em Dart para fazer conversão de moedas utilizando um map. Considere dolar igual a 5 reais e euro igual a 6 reais.

Solução

-----