



Dart Básico - Aula 2



Os arrays são estruturas que servem para guardar dados, e organizá-los. Seu objetivo é ser um espaço fixo na memória do computador que armazena elementos.

Esses elementos podem ser acessados por um tipo de indicação, que chamamos de índice. Você pode guardar seus chapéus na gaveta 1, suas calças na gaveta 2 e as meias na gaveta 3, e sempre que precisar de calças, chapéus ou meias vai saber em que gaveta buscar.



- Tipo mais simples array
- Index começa por 0

```
var listNotas = [9, 7.5, 8];  
  
listNotas[1] = 9;  
  
for(var i = 0; i < listNotas.length; i++) {  
    total += listNotas[i];  
}
```

```
List<String> nomes = [];  
  
List<int> palpites = [0, 7, 32]  
  
// Adicionando novos itens a lista  
nomes.add('Laura');  
  
nomes.add('Olívia');  
  
  
// Removendo itens de uma lista  
palpites.remove(7);  
  
// Removendo itens de uma lista, na posição X  
palpites.removeAt(1);  
  
palpites[1] = 3;  
  
palpites.length; // Lendo o tamanho atual da lista
```

Exercício 6



Vamos criar um sistema de cálculo de notas? A aplicação deve armazenar uma quantidade N de notas e calcular a média a qual o aluno chegou. A nota para passar de ano é 7.



Exercício 7



Crie uma lista de valores inteiros. Essa lista deve ser populada utilizando somente números aleatórios.

Após popular essa lista, remova todos os itens pares dessa lista, deixando somente os valores ímpares.

Em uma nova lista, devem ficar somente os valores pares.



Exercício 8



Crie uma lista de valores inteiros. Essa lista deve ser populada utilizando somente números aleatórios.

Após popular essa lista, escreva uma função que ordene essa lista em ordem crescente, do menor para o maior.

Essa função deve receber por parâmetro uma lista de inteiros, e retornar uma nova lista de inteiros, ordenada.



Exercício 9



Escreva uma função que receba duas listas por parâmetro, e que retorne um terceiro array, com somente as diferenças entre as duas listas.

Ex:

```
lista1 = [7, 8, 9, 10];
```

```
lista2 = [7, 10, 14];
```

```
listaResultado = [8, 9, 14];
```



Exercício 10

Crie uma lista de valores inteiros. Essa lista deve ser populada utilizando somente números aleatórios.

Após popular essa lista, escreva uma função que retorne o terceiro menor número dessa lista.

Ex:

[7, 12, 14, 3, 22, 9]

o menor número dessa lista é 3

o segundo menor número dessa lista é 7

o terceiro menor número dessa lista é 9, que é o resultado esperado.





- Coleção desordenada de itens únicos

```
var tecnologiasMobile = {'Flutter', 'React Native', 'Kotlin', 'Swift'};

tecnologiasMobile.add("Xamarin");

for (String tecnologia in tecnologiasMobile)
    print(tecnologia);
```

```
// declarando um set
var lista = <String>{};

Set<String> lista = {};

lista.add("Item 1"); // Adicionando um novo item ao Set

lista.remove("Item 1"); // Removendo um item do Set

lista.elementAt(0); // Buscando um elemento do set em uma posição específica
```

Exercício 11

Utilizando Sets, vamos criar um programa que, a partir de uma lista de pessoas, faz o sorteio de um amigo secreto.

Cada pessoa na lista deverá receber um amigo secreto, não podendo receber a si mesmo.

Também é necessário verificar se a quantidade de pessoas presentes na lista é par, pois caso não seja, não será possível realizar o sorteio.





- Coleção de itens com chave e valor
- Chave e valor podem ser de qualquer tipo de objeto

```
var mapaNotas = Map<String, double>();

mapaNotas["Prova 1"] = 9;
mapaNotas["Prova 2"] = 7.5;
mapaNotas["Prova 3"] = 8.5;

mapaNotas.forEach((chave, valor) => print("$chave - $valor"));
```

```
Map<String, dynamic> usuario = {  
    "nome": "Fernando Duwe",  
    "idade": 33,  
    "nota": 14.13  
};
```

```
var usuario2 = {  
    "nome": "Usuário 2",  
    "idade": 27,  
    "nota": 33,  
};
```

```
usuario["nota2"] = 3;  
usuario["nota"] = 12;
```

```
print(usuario);
```

```
usuario.remove("nota2");
```

```
print(usuario);
```

```
Map<String, dynamic> usuario = {  
    "nome": "Fernando Duwe",  
    "idade": 33,  
    "nota": 14.13  
};  
  
// listando todas as chaves e valores de um mapa  
for(MapEntry<String, dynamic> item in usuario.entries) {  
    print(item.key);  
    print(item.value);  
}
```

```
// Guardando uma lista de mapas

void addContato(String nome, String telefone, List<Map<String, dynamic>>
contatos) {
    contatos.add({
        "nome": nome,
        "telefone": telefone,
    });
}

void main() {
    List<Map<String, dynamic>> telefones = [];

    addContato("Pessoa 1", "(47) 994324-3434", telefones);
    addContato("Pessoa 2", "(47) 24234-42343", telefones);

    for (Map<String, dynamic> item in telefones) {
        print ("${item['telefone']} - ${item['nome']}");
    }
}
```


Exercício 12

Vamos criar uma lista de tarefas. Guardaremos em cada tarefa o descritivo da tarefa e se ela foi concluída ou não. Você deverá criar as seguintes funções:

```
addTarefa();
```

```
concluirTarefa();
```

```
listarTarefasConcluidas();
```

```
listarTodasAsTarefas();
```



Exercício 13


Vamos criar um sistema de login. Você deverá guardar uma lista de usuários e senhas. Crie uma rotina de cadastro de usuário e uma rotina de login, que retornará verdadeiro se o login foi bem sucedido.





Dart suporta a utilização de spread operators (...). Podemos utilizar os spread operators para inserir os itens de um array em um outro array, veja:

```
var list = [1, 2, 3];  
  
var list2 = [0, ...list];
```



Ao adicionar novos itens nos nossos arrays, podemos também realizar verificações, para controlar a inserção de itens.

```
void main() {  
    bool promoActive = true;  
  
    var nav = ['Home', 'Furniture', 'Plants', if (promoActive) 'Outlet'];  
  
    var listOfInts = [1, 2, 3];  
  
    var listOfStrings = ['#0', for (var i in listOfInts) '#$i'];  
  
    var intsFromFor = [for (int i = 0; i < 100; i++) i ];  
  
    print(intsFromFor);  
}
```