UNINASSAU | ser educacional

Desenvolvimento para Dispositivos Móveis

Estruturas de repetição e Vetores em DART

Profº: Joseph Donald

Contatos:
(83) 98228-8607
@josephdonald
030106382@uninassau.edu.br

"Se você tem uma maçã e eu tenho outra; e nós trocamos as maçãs, então cada um terá sua maçã. Mas se você tem uma ideia e eu tenho outra, e nós as trocamos; então cada um terá duas ideias."

George Bernard Shaw

DART oferece três tipos de estruturas de repetição para iterar sobre uma determinada lista de elementos:

- **for**
- **while**
- **do-while**

Estrutura de repetição “for”:

É usada quando se sabe a quantidade de iterações que serão necessárias. A sua sintaxe é composta de três partes: a inicialização de uma variável de controle; a condição que a variável de controle deve atender para a execução continuar; e o incremento ou decremento da variável de controle.

UNINASSAU

Estruturas de repetição em DART

Desenvolvimento Mobile
Profº: Joseph Donald

Exemplo de código:

```
for (int i = 0; i < 5; i++) {
  print(i);
}
```

Saída esperada:

```
0
1
2
3
4
```

Estrutura de repetição “while”:

É usada quando não se sabe a quantidade de iterações necessárias, mas se sabe uma condição que deve ser atendida para a execução continuar. A sua sintaxe é composta de uma condição que deve ser verdadeira para que o bloco de código dentro dela seja executado.

UNINASSAU

Estruturas de repetição em DART

Desenvolvimento Mobile
Profº: Joseph Donald

Exemplo de código:

```
int i = 0;
while (i < 5) {
  print(i);
  i++;
}
```

Saída esperada:

```
0
1
2
3
4
```

Estrutura de repetição “do-while”:

É semelhante a estrutura “while”, com a diferença de que o bloco de código dentro dela é executado pelo menos uma vez, mesmo que a condição não seja verdadeira.

UNINASSAU

Estruturas de repetição em DART

Desenvolvimento Mobile
Profº: Joseph Donald

Exemplo de código:

```
int i = 0;
do {
  print(i);
  i++;
} while (i < 5);
```

Saída esperada:

```
0
1
2
3
4
```

UNINASSAU

DESAFIO 1

Desenvolvimento Mobile
Profº: Joseph Donald

- Crie um programa que peça um número ao usuário e exiba a tabuada desse número de 1 a 10.
- ✅ Use um loop for para multiplicar o número.
- ✅ Exiba o resultado no formato:

  n x 1 = resultado

Em DART, temos três tipos de vetores:

- **Listas**
- **Conjuntos**
- **Mapas**

Listas:

- As listas são uma coleção ordenada de elementos do mesmo tipo, que podem ser acessados por um índice numérico.
- Os elementos podem ser adicionados, removidos e modificados a qualquer momento. A sua sintaxe é composta por colchetes [ ], e os elementos são separados por vírgulas.

```
List<String> frutas = ['maçã', 'banana', 'laranja'];
print(frutas); // saída esperada: [maçã, banana, laranja]
```

Conjuntos:

- Os conjuntos são uma coleção não ordenada de elementos únicos do mesmo tipo, que não possuem índices numéricos e não permitem elementos duplicados.
- Os elementos podem ser adicionados e removidos a qualquer momento. A sua sintaxe é composta por chaves { }, e os elementos são separados por vírgulas.

```
Set<int> numeros = {1, 2, 3, 4, 5};

print(numeros); // saída esperada: {1, 2, 3, 4, 5}
```

Mapas:

- Os mapas são uma coleção de pares chave-valor, onde cada chave é única e é usada para acessar um valor associado.
- Os elementos podem ser adicionados, removidos e modificados a qualquer momento. A sua sintaxe é composta por chaves {}, e os pares chave-valor são separados por vírgulas e dois pontos :

```
Map<String, String> telefones = {'João': '123456', 'Maria': '654321', 'Pedro': '789456'};
print(telefones); // saída esperada: {João: 123456, Maria: 654321, Pedro: 789456}
```

Funções em DART

- Para criar funções em Dart, usamos a palavra-chave void para indicar que a função não retorna nenhum valor, ou um tipo de retorno caso a função retorne algum valor.
- Em seguida, escrevemos o nome da função, seguido de parênteses que podem conter parâmetros de entrada separados por vírgulas. Por fim, escrevemos o corpo da função, delimitado por chaves { }. O corpo da função contém o código que será executado quando a função for chamada.

UNINASSAU

Funções em DART

Desenvolvimento Mobile
Profº: Joseph Donald

- Função com retorno

```
int soma(int a, int b) {
  int resultado = a + b;
  return resultado;
}
```

- Função sem retorno

```
void saudacao(String nome) {
  print("Olá, $nome! Como você está?");
}
```

UNINASSAU

DESAFIO 2

Desenvolvimento Mobile
Profº: Joseph Donald

- Elabore um algoritmo em DART que calcule o IMC de dona Clarabela. Estes são os dados dela:
  - Altura: 1,63
  - Peso: 45 kg

- Depois de encontrar o índice de massa corporal, o algoritmo deve classificar a condição física de dona Clarabela de acordo com a seguinte tabela:

Condição	Situação
IMC abaixo de 20	Abaixo do peso
IMC de 20 até 25	Peso Normal
IMC de 25 até 30	Sobre Peso
IMC de 30 até 40	Obeso
IMC de 40 e acima	Obeso Mórbido