



INSTITUTO FEDERAL
PIAUI

Prof. Dr. Otilio Paulo
otilio.paulo@ifpi.edu.br

Programação para Dispositivos Móveis

Estruturas de Repetição

<https://dartpad.dartlang.org/>

Estruturas de Repetição

Em Dart, existem três estruturas de repetição, são elas:

- For;
- While;
- do...while.

Estrutura For

Estrutura de repetição for

- Denominamos “laço” (loop em inglês) a uma estrutura de repetição.
- As estruturas de repetição, executam a repetição de um conjunto de instruções enquanto uma determinada **condição é verdadeira**.

Estrutura For

Pseudocódigo – Estrutura de repetição Para

- Em pseudocódigo o laço for da linguagem Dart é equivalente ao comando Para.
- Seu funcionamento é simples, como veremos.

Estrutura For

Pseudocódigo:

```
Para (valor inicial até condição_final passo n) faça  
  Início  
    Instruções;  
  Fim;
```

onde:

- valor_inicial é uma instrução de atribuição do valor inicial do laço para a variável de controle.
- condição final é uma condição que controla o laço.
- passo é o incremento do laço.

Estrutura For

Utilização do comando for

O laço for é uma estrutura de repetição muito utilizada.

- É muito útil quando se sabe de antemão quantas vezes a repetição deverá ser executada.
- Este laço utiliza uma variável para controlar a contagem do loop, bem como seu incremento.
- Trata-se de um comando bem enxuto, já que própria estrutura faz a inicialização, incremento e encerramento do laço.

Relembrando Conceitos

valor será incrementado apos a operação :1
valor será decrementado apos a operação :1
valor será incrementado antes da operação :2
valor será decrementado antes da operação :0

```
void main() {  
    int a = 1;  
    int b = 1;  
    int c = 1;  
    int d = 1;  
    int valor = a++;  
  
    print('valor será incrementado apos a operação :$valor');  
    valor = b--;  
    print('valor será decrementado apos a operação :$valor');  
    valor = ++c;  
    print('valor será incrementado antes da operação :$valor');  
    valor = --d;  
    print('valor será decrementado antes da operação :$valor');  
}
```


Estrutura For

Sintaxe:

```
for(valor_inicial; condição_final; valor_incremento)
{
    instruções;
}
```

```
void main() {
    int contador; //variável de controle do loop
    for (contador = 1; contador <= 10; contador++) {
        print('$contador');
    }
}
```

1
2
3
4
5
6
7
8
9
10

Atividade

Na estrutura criada no exemplo anterior faça que a contagem seja regressiva.

```
10  
9  
8  
7  
6  
5  
4  
3  
2  
1
```

Atividade

```
void main() {  
    int contador; //variável de controle do loop  
    for (contador = 10; contador >= 1; contador--) {  
        print('$contador');  
    }  
}
```

Estrutura For

Caso queiramos usar um for para mostrar valores em uma lista devemos usar os seguinte comando:

```
void main(){  
    List produtos =[1,2,3,4,5];  
    for(var item in produtos){  
        print(item);  
    }  
}
```

Atividade

Crie uma lista e mostre ela através de um for como mostrado logo abaixo:

```
luis  
cesar  
rodrigo  
paulo
```

DART

```
void main(){  
    List nomes =['luis', 'cesar', 'rodrigo', 'paulo'];  
    for(var item in nomes){  
        print(item);  
    }  
}
```

Atividade

Faça um código usando o for que mostre o resultado a seguir:

Não é válido o uso de lista.

1
2
3
4
5
5
4
3
2
1

Atividade

```
void main() {  
    int contador;  
    for (contador = 1; contador <= 5; contador++) {  
        print('$contador');  
    }  
    for (contador = 5; contador >= 1; contador--) {  
        print('$contador');  
    }  
}
```

Console

1
2
3
4
5
5
4
3
2
1

Atividade

Faça um código usando o for com lista que mostre o resultado a seguir:

Uma lista criada não poderá ter mais de 5 elementos.

1
2
3
4
5
5
4
3
2
1

Atividade

```
void main() {  
    List numero = [1, 2, 3, 4, 5, 5, 4, 3, 2, 1];  
    for (var item in numero) {  
        print(item);  
    }  
}
```

Estrutura While

While – Esta instrução é usada quando não sabemos quantas vezes um determinado bloco de instruções precisa ser repetido. Com ele, a execução das instruções vai continuar até que uma condição seja verdadeira. A condição a ser analisada para a execução do laço de repetição deverá retornar um valor booleano.

```
while (teste condicional)
{
    //comandos; > serão executados enquanto o teste condicional for igual a
    verdadeiro (true)|
}
```

Estrutura While

While – Essa estrutura é uma mescla entre o for e o if.

```
void main() {  
    var salario = 1000;  
    while (salario < 5000) {  
        salario += 100;  
        print("Meu salário AINDA é de = R\$$salario ");  
    }  
}
```

Atividade

Construa uma estrutura em que uma idade seja armazenada em uma variável. Em uma estrutura comparativa com while seja incremento valor a esta variável até que a condição ser maior de 17 anos seja validada.

```
5 é menor que 18 anos
6 é menor que 18 anos
7 é menor que 18 anos
8 é menor que 18 anos
9 é menor que 18 anos
10 é menor que 18 anos
11 é menor que 18 anos
12 é menor que 18 anos
13 é menor que 18 anos
14 é menor que 18 anos
15 é menor que 18 anos
16 é menor que 18 anos
17 é menor que 18 anos
temos 18 anos de idade
```

Estrutura While

```
void main() {  
    int idade=1;  
    while(idade<18){  
        print(idade.toString()+' é menor que 18 anos');  
        idade++;  
    }  
    print ('temos $idade anos de idade');  
}
```

```
5 é menor que 18 anos  
6 é menor que 18 anos  
7 é menor que 18 anos  
8 é menor que 18 anos  
9 é menor que 18 anos  
10 é menor que 18 anos  
11 é menor que 18 anos  
12 é menor que 18 anos  
13 é menor que 18 anos  
14 é menor que 18 anos  
15 é menor que 18 anos  
16 é menor que 18 anos  
17 é menor que 18 anos  
temos 18 anos de idade
```

Estrutura Do While

Do/While – O do/while tem quase o mesmo funcionamento que o while, a diferença é que com o uso dele teremos os comandos executados ao menos uma única vez.

```
do  
{  
  
    //comandos;  
  
} while (condicao);
```

Estrutura Do While

Do/While – O do/while tem quase o mesmo funcionamento que o while, a diferença é que com o uso dele teremos os comandos executados ao menos uma única vez.

```
void main() {  
    int i = 0;  
  
    while(i <= 10) {  
        print(i);  
        ++i;  
    }  
    print('final');  
}
```

```
0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
final
```


Estrutura Do While

Quando a condição se torna falsa ele não entra no laço.

```
void main() {  
    int i = 0;  
  
    while(i >= 10) {  
        print(i);  
        ++i;  
    }  
    print('final');  
}
```

final

Estrutura Do While

Do/While – O do/while tem quase o mesmo funcionamento que o while, a diferença é que com o uso dele teremos os comandos executados ao menos uma única vez.

```
void main() {  
    int i = 0;  
  
    do {  
        print(i);  
        ++i;  
    } while (i <= 10);  
}
```

0
1
2
3
4
5
6
7
8
9
10

Estrutura Do While

Quando a condição do laço é falsa ele sai mas imprime uma vez.

```
void main() {  
    int i = 0;  
  
    do {  
        print(i);  
        ++i;  
    } while(i != 1);  
}
```

0

Atividade

Usando o Do while em um algoritmo devemos obter os seguintes resultados:

Quando uma variável recebe um valor, então o programa imprime a frase a quantidade de vezes o valor desta variável.

Caso receba valor zero ou negativo ainda mostrará frase uma vez.

```
Você é um vencedor!  
Você é um vencedor!  
Você é um vencedor!  
Você é um vencedor!  
Você é um vencedor!
```

```
Você é um vencedor!
```

Atividade

Você é um vencedor!
Você é um vencedor!
Você é um vencedor!
Você é um vencedor!
Você é um vencedor!

```
void main() {  
    int i = 0;  
    int frase = 5;  
    do {  
        print('Você é um vencedor!');  
        ++i;  
    } while (i < frase);  
}
```

Atividade

Usando qualquer estratégia aprendida vamos criar um algoritmo que mostre uma contagem de 0 a 30 somente mostrando números pares.

0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30

Atividade

```
void main() {  
    for (int pares = 0; pares <= 30; pares += 2) {  
        print(pares);  
    }  
}
```

```
void main() {  
  
    List number = [0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28,  
30];  
    for (var item in number) {  
        print(item);  
    }  
}
```

 RUN

0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30

Atividade - Desafio

Usando qualquer estratégia aprendida vamos criar um algoritmo que possui um número inteiro e mostre os números pares e ímpares separados, da contagem de 1 até esse inteiro.

Caso este número for menor que 1 então a mensagem 'fora do padrão' deverá aparecer.

Número iniciado foi 10

Contagem Pares

2

4

6

8

10

Contagem Impares

1

3

5

7

9

Atividade - Desafio

```
void main() {  
    int number = 10;  
    print('Número iniciado foi $number');  
    if (number < 1) {  
        print('fora do padrão');  
    }  
    print('Contagem Pares');  
    if (number % 2 == 0) {  
        for (int i = 2; i <= number; i += 2) {  
            print(i);  
        }  
    }  
    print('Contagem Impares');  
    for (int i = 1; i <= number; i += 2) {  
        print(i);  
    }  
}
```