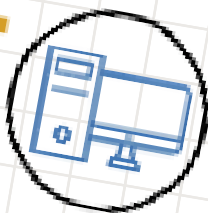


CODE
LAB TEEN

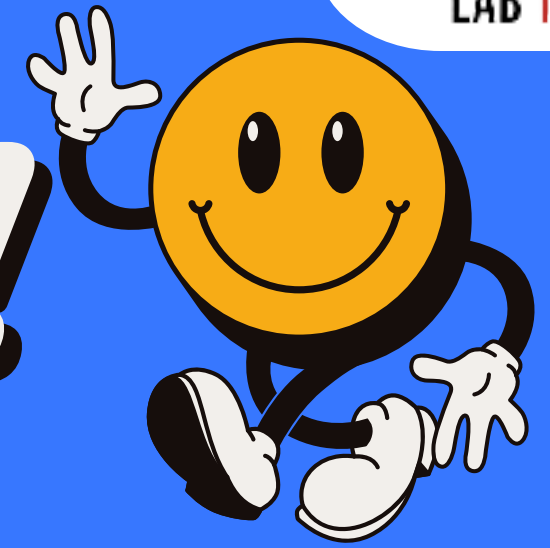


PYTHON AULA 13





BEM - VINDOS!



AGENDA

Hoje vamos revisar for e também vetores e resolver alguns exercícios.

REVISÃO-FOR

A estrutura de repetição for em Python é usada para percorrer sequências, permitindo que você execute um bloco de código várias vezes sem precisar ficar pensando em atualizar a variável, pois o próprio laço faz isso automaticamente.

REVISÃO-ESTRUTURA FOR

ESTE É O MÉTODO QUE
DEFINE QUAIS NÚMEROS
SERÃO UTILIZADOS

AQUI ESTÃO OS NÚMEROS QUE
DETERMINAM A REPETIÇÃO

```
for i in range(0, 10, 2)
```

ESTA É A VARIÁVEL
QUE SERÁ
ATUALIZADA A
CADA ITERAÇÃO
DO LOOP FOR

REVISÃO-ESTRUTURA FOR

O SEGUNDO NÚMERO É
O LIMITE QUE A
VARIÁVEL NÃO
ALCANÇARÁ

```
for i in range (0, 10, 2)
```

O PRIMEIRO NÚMERO
INDICA O VALOR INICIAL
DA VARIÁVEL

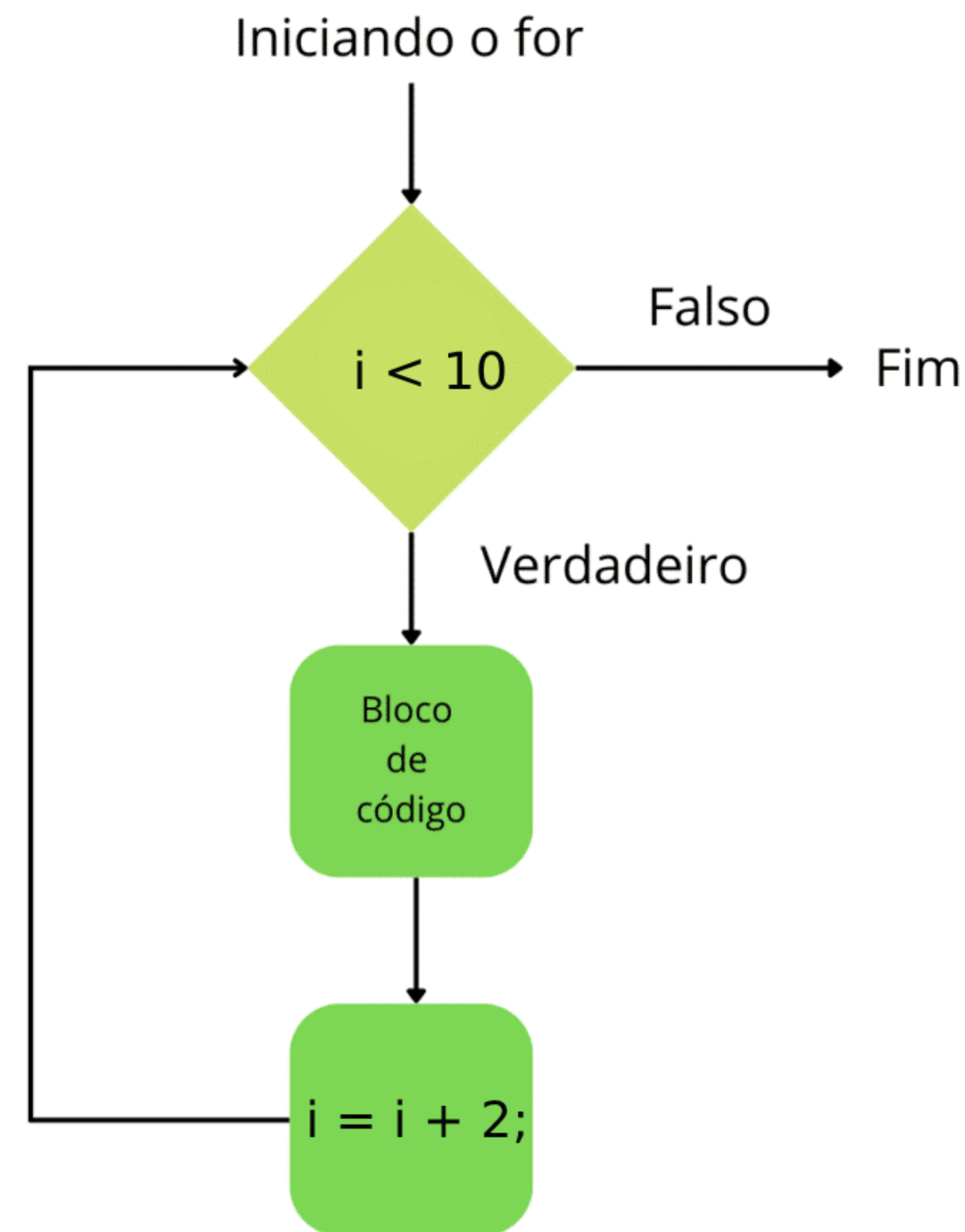
O TERCEIRO NÚMERO
INDICA O INCREMENTO
DA VARIÁVEL A CADA
ITERAÇÃO

Caso não coloque o
terceiro número, o
programa considera
como se existisse o
número 1 no lugar.

REVISÃO-FOR

```
for i in range (0, 10, 2)
```

Nesse código podemos ver que a variável “i” vai começar sendo 0 e cada vez que o For se repetir vai somar +2 na variável “i”, enquanto “i” for menor do que 10



REVISÃO - VETORES



Vetores, são uma estrutura de dados usada para armazenar uma sequência de elementos do mesmo tipo em uma única variável.



Várias variáveis



Um vetor

REVISÃO-VETORES



Para usarmos um vetor, antes temos que lembrar sobre o “índice”. O índice de um vetor em Python é o número que identifica a posição de cada elemento dentro do vetor. Isso permite acessar, modificar ou remover um elemento específico do vetor. Os índices em Python começam em 0, ou seja, o primeiro elemento tem índice 0, o segundo tem índice 1, e assim por diante.

LETRA	P	Y	T	H	O	N
ÍNDICE	0	1	2	3	4	5

REVISÃO - VETORES



Relembrando que em Python temos uma coisa bônus, também podemos usar índices negativos. Eles começam do fim e vão até o início do vetor.

VETOR	10	20	3	15	50
ÍNDICE	-5	-4	-3	-2	-1

REVISÃO-VETORES



Relembrando que para usarmos os vetores em Python, temos que instalar e importar uma biblioteca chamada *numpy*:

```
pip install numpy
```

```
import numpy as np
```

Lembre-se que temos que usar Google Colab.

REVISÃO - EXEMPLOS VETORES

Nesse exemplo criamos uma variável chamada `vetor` e fizemos ela receber uma função da biblioteca `np`. A função `np.zeros()` é usada para criar um vetor preenchido com zeros. Colocamos entre parênteses a quantidade de zeros que queríamos.

```
import numpy as np
vetor = np.zeros(5)
print(vetor)
```

```
[0. 0. 0. 0. 0.]
```

Essa função é útil quando você precisa inicializar um vetor com valores zerados antes de realizar cálculos.

REVISÃO - EXEMPLOS VETORES

Na saída, nós temos [0. 0. 0. 0. 0.], isso significa que temos 5 números do tipo *float*, pois tem pontos depois do número. O padrão dessa função é sempre retornar números com pontos, mas nós podemos mudar isso:

```
import numpy as np
vetor = np.zeros(5)
print(vetor)
```

```
[0. 0. 0. 0. 0.]
```

```
import numpy as np
vetor = np.zeros(5, dtype=int)
print(vetor)
```

```
[0 0 0 0 0]
```

REVISÃO - EXEMPLOS VETORES

Nesse exemplo criamos um vetor que vai de 0 até um número específico usando a função `np.arange()`. A função `np.arange()` gera uma sequência de números dentro de um intervalo definido. Nesse caso cria um vetor que vai de 0 até 4:

```
import numpy as np
vetor = np.arange(5, dtype=int)
print(vetor)
```

```
[0 1 2 3 4]
```

REVISÃO - EXEMPLOS VETORES

Caso você queira usar um dos números em específico, temos que colocar entre colchetes o índice depois do nome da variável, por exemplo:

```
import numpy as np
vetor = np.arange(5, dtype=int)
print(vetor[1])
```

1

Valor do índice

Lembrando que também podemos usar os índices negativos

REVISÃO - EXEMPLOS VETORES

```
import numpy as np
vetor = np.zeros(5, dtype=int)
vetor[0]=2
vetor[1]=7
vetor[2]=8
vetor[3]=10
vetor[4]=-2
print(vetor)
```

```
[ 2  7  8 10 -2]
```

Relembrando também que podemos inserir qualquer valor nos índices.

```
import numpy as np

vetor = np.array([1,2,3,4])
```

REVISÃO - EXEMPLOS VETORES

Podemos também percorrer os valores de um *array* e acessar cada índice usando as estruturas de repetição. Para isso, usamos a *variável* de iteração como o índice do vetor. Assim, em cada iteração nós teremos o valor do vetor em um dos índices.

```
import numpy as np

vetor = np.array([1,2,3,4])

for i in range(4):
    print(f"índice: {i} valor: {vetor[i]}")
```

```
índice: 0 valor: 1
índice: 1 valor: 2
índice: 2 valor: 3
índice: 3 valor: 4
```

Nesse exemplo o For do código começa com $i = 0$ e a cada iteração do loop é adicionada 1, assim conseguimos percorrer o vetor até a quarta posição dele

REVISÃO - EXEMPLOS VETORES

Lembrando que nós também podemos usar o *input* para adicionarmos o valor no vetor:

```
import numpy as np
vetor = np.zeros(5, dtype=int)
vetor[0]=int(input("Valor do índice 0: "))
print(vetor)
```

```
Valor do índice 0: 10
[10  0  0  0  0]
```

REVISÃO - EXEMPLOS VETORES

```
import numpy as np  
vetor = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9])  
for i in range(0, 10, 2):  
  
    valor_atual = vetor[i]  
    print(f'Valor {valor_atual}')
```

```
Valor 1  
Valor 3  
Valor 5  
Valor 7  
Valor 9
```

Nesse exemplo o for do código começa com $i = 0$ e a cada iteração do loop é adicionada 2, assim percorremos o vetor com os índices pares (0,2,4,6,8)

REVISÃO - EXEMPLOS VETORES

Também podemos executar funções com os valores do vetor como no exemplo que temos somando todos os valores do vetor.

```
import numpy as np

vetor = np.array([1,2,3,4])
soma = 0
for i in range(4):
    valor_atual = vetor[i]
    soma += valor_atual

print(soma)
```

10

Para isso, criamos um for que percorra todo o vetor e criamos uma variável chamada soma e a cada interação adicionamos o valor do vetor nessa variável

OBRIGADO!

Contem para gente o que você achou da aula de hoje:



<https://forms.gle/7KACyH95T1oVaSEq5>

CODELAB TEEN