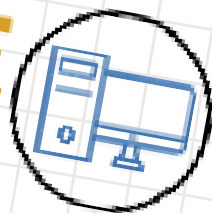


CODE
LAB TEEN

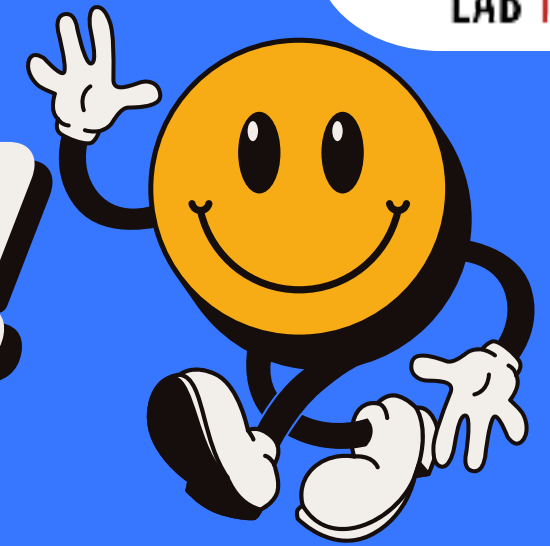


PYTHON AULA 12





BEM - VINDOS!



AGENDA

Hoje vamos começar a aprender sobre vetores, mais conhecidos como *arrays*.

VETORES



Em um projeto de programação grande, é comum ter muitas variáveis do mesmo tipo para guardar informações. As vezes, guardamos tantos dados que fica difícil e longo escrever tudo, além de confuso. Para resolver esse problema, nós usamos vetores para guardar os dados.



Várias variáveis



Um vetor

VETORES



Um vetor guarda várias informações de um mesmo tipo dentro dele, para que assim em vez de ter várias variáveis temos apenas uma que guarda os valores que precisamos.

Várias variáveis	0	1	2	3	4	5
Vetor	0, 1, 2, 3, 4, 5					

VETORES



Para usarmos um vetor, antes temos que aprender a contar o “índice” das coisas. Um índice é onde alguma coisa está guardada, porém ao contrário da vida real, nós começamos a contar no zero.

Vamos usar de exemplo a palavra “IDADE”, e vamos contar e mostrar qual índice é qual letra.

VETORES



IDADE

LETRA	I	D	A	D	E
ÍNDICE	0	1	2	3	4

VETORES



Podemos ver que nesse exemplo cada letra representa um índice, porém quando nós usamos números, cada número é um índice no vetor.

VETOR	10	20	3	15	50
ÍNDICE	0	1	2	3	4

VETORES



Mas, 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

VETORES



Vamos colocar a mão na massa! Para usarmos os vetores em Python, temos que instalar uma biblioteca chamada *numpy*:

```
pip install numpy
```

Note que nós apenas conseguimos fazer isso no Google Colab, por isso recomendamos o uso dele nessa aula.

VETORES



Para criarmos o nosso vetor, temos que importar a biblioteca depois de instalar ela:

```
import numpy as np
```

Da forma que foi escrita, nós importamos a biblioteca *numpy* para o nosso programa e chamamos ela de *np* no nosso código

VETORES



Agora nós podemos criar o vetor! Para o nosso primeiro exemplo, vamos criar um vetor com 5 números, e cada um deles vai ser igual a zero.

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

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

VETORES



Vamos explicar a sintaxe do código agora. Nós criamos uma variável chamada `vetor` e fizemos ela receber uma função da biblioteca `np`. Essa função se chama “zeros”, e colocamos entre parênteses a quantidade de zeros que queríamos.

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

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

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]
```

VETORES



Nós também podemos criar um vetor com números que vão de 0 até o número que queremos, só que em vez de usar o método “zeros”, nós usamos *arange*:

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

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

VETORES



Caso você queira usar um dos números em específico, nós temos que saber o índice daquele número. Sabendo o índice do número, se quisermos usá-lo nós temos que colocar entre colchetes o índice depois do nome da variável, como por exemplo:

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

1

VETORES



Também podemos usar os índices negativos, porém devemos tomar cuidado para não acessarmos o número que não queremos.

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

4

ERRADO

```
import numpy as np
vetor = np.arange(5, dtype=int)
print(vetor[-4])
```

1

CERTO

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]
```

Como normalmente nós não queremos valores só iguais a zero ou de zero até um número X, nós inserimos os valores nos índices.

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]
```

VETORES



Podemos também percorrer os valores de um *array* e acessar cada índice usando os *loops*(FOR e WHILE). Para isso, usamos a *variável* de iteração como o índice do vetor. Assim, em cada iteração nós teremos o valor 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
```

VETORES



O mesmo resultado
pode
ser obtido com o loop
WHILE.

Consegue fazer um
código usando o
WHILE?


No exemplo anterior, a cada iteração do loop, é exibido o valor armazenado no índice *i* do vetor. A tabela abaixo mostra quais valores serão exibidos baseado no *i*:

vetor[i]	1	2	3	4
i	0	1	2	3

VETORES



Suponham agora que queremos somar todos os valores de um vetor e obter o resultado do total. Como poderíamos fazer isso?




Basta utilizar um **LOOP** para percorrer o vetor e somar os valores de cada iteração

```
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 uma variável chamada soma e adicionamos o valor do vetor a ela em cada iteração

EXERCÍCIO

No vetor a seguir qual seria o índice do número 18?

[0,1,18,67,100]

OBRIGADO!

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



<https://forms.gle/uSEa4NHTvCh7fZRX8>