

**INSTITUTO FEDERAL**

Catarinense

Campus Camboriú

# AULA 1 (Arrays)

**Professora: Lidiane Visintin**

*lidiane.visintin@ifc.edu.br*

**Professor: Rafael de Moura Speroni**

*rafael.speroni@ifc.edu.br*

# Objetivo:



- Compreender o conceito de Array.

# Programa sem Array

```
nro = 10

for i in range(0, nro):
    n1 = float(input("Informe a primeira nota do aluno: "))
    n2 = float(input("Informe a segunda nota do aluno: "))

    media = (n1+n2)/2
    print(f"A média do aluno é: {media}")
```

Mas e se eu precisasse mostrar todas as médias após sair desta estrutura de repetição? Isso seria possível?

# Definição

---

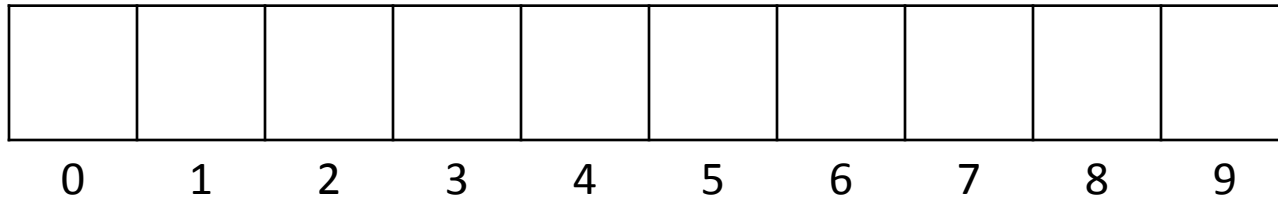
Em programação de computadores, **um arranjo(array)** é uma estrutura de dados que armazena uma coleção de elementos de tal forma que cada um dos elementos possa ser identificado por, pelo menos, **um índice** ou **uma chave**. Essa estrutura de dados também é conhecida como **variável indexada, vetor e matriz**.

**Arrays em Python** são estruturas **homogêneas**, exclusivamente sobre **valores numéricos** (int, float, entre outros).

# Como representar um vetor?

---

Vetor **média**



# Requisitos



Para fazer uso da biblioteca **NumPy** é preciso:

Em todos os sistemas operacionais ( Windows, macOS e Linux):

- Instale o [Anaconda](#) (instala todos os pacotes que você precisa e todas as várias outras ferramentas).
- Ou, utilize o comando **pip install numpy** no terminal.

# Como declarar um vetor?

---

```
import numpy as np

# define o tamanho do array
N = 5

#preenche a estrutura com zeros
vetor = np.zeros(N)
```

# Como mostrar o conteúdo de um vetor?

---

```
import numpy as np

# define o tamanho do array
N = 5

#preenche a estrutura com zeros
vetor = np.zeros(N)

# mostra os valores armazenados na estrutura
print(vetor)
```



# Como preencher com valores do usuário?

```
import numpy as np

# define o tamanho do array
N = 5

#preenche a estrutura com zeros
vetor = np.zeros(N)

#preenche o vetor com valores do tipo float
for i in range(N):
    vetor[i] = float(input(f'Informe um valor para V[{i}]):

# mostra os valores armazenados na estrutura
print(vetor)
```

# Como preencher com valores do usuário?

```
import numpy as np
```

arrays possuem tamanhos determinados

```
# define o tamanho do array
```

```
N = 5
```

é preciso iniciar com zeros(0) neste caso

```
#preenche a estrutura com zeros
```

```
vetor = np.zeros(N)
```

precisamos de estrutura de repetição para percorrer cada posição do array.

```
#preenche o vetor com valores do tipo float
```

```
for i in range(N):
```

atribuindo valores float para o array

```
    vetor[i] = float(input(f'Informe um valor para V[{i}]):
```

```
# mostra os valores armazenados na estrutura
```

```
print(vetor)
```

exibindo o conteúdo do array;

# Outra forma de mostrar os valores?

```
import numpy as np

# define o tamanho do array
N = 5

#preenche a estrutura com zeros
vetor = np.zeros(N)

#preenche o vetor com valores do tipo float
for i in range(N):
    vetor[i] = float(input(f'Informe um valor para V[{i}]):

# outra forma de mostrar
for i in range(N):
    print(f'V[{i}] = {vetor[i]} ')
```

# Outra forma de mostrar os valores?

```
import numpy as np

# define o tamanho do array
N = 5

#preenche a estrutura com zeros
vetor = np.zeros(N)

#preenche o vetor com valores do tipo float
for i in range(N):
    vetor[i] = float(input(f'Informe um valor para V[{i}]):

# outra forma de mostrar
for i in range(N):
    print(f'V[{i}] = {vetor[i]} ')
```

para exibir o conteúdo do array  
posição a posição é preciso fazer uso  
de uma estrutura de repetição

# Métodos especiais

---

```
# mostra o tipo da estrutura
print(type(vetor))

soma2 = vetor.sum() # somatório
media = vetor.mean() # média
desvio = vetor.std() # desvio padrão
max = vetor.max() # o maior valor
min = vetor.min() # o menor valor
argmax = vetor.argmax() # retorna a posição que contém o
maior valor da estrutura
argmin = vetor.argmin() # retorna a posição que contém o
menor valor da estrutura
```

# Referências



## Referências Básicas

FORBELLONE, André Luiz Villar; EBERSPÄCHER, Henri Frederico. Lógica de programação: a construção de algoritmos e estruturas de dados. 3. ed. Pearson Prentice Hall. 2005

MANZANO, José Augusto N. G; OLIVEIRA, Jayr Figueiredo de.. Algoritmos: lógica para desenvolvimento de programação de computadores.. 27. ed.. Érica. 2014

## Referências Complementares

DOWNEY, Allen B. **Pense em Python**. 2ª Ed. Novatec. 2016

MENEZES, Nilo Ney de Coutinho. **Introdução a programação com Python**. 3ª Ed. Novatec. 2019

CORMEN, Thomas H et al. **Algoritmos: teoria e prática**. 2. ed. Elsevier, Campus,. 2002

## Referências na Internet

<https://docs.python.org/3/>

<https://www.w3schools.com/python/default.asp>