

Semana 25 - Aula 1

Tópico Principal da Aula: Vetores (Arrays Unidimensionais)

Subtítulo/Tema Específico: Conceitos e Prática de Implementação de Vetores em Python

Código da aula: [SIS]ANO1C1B4S25A1

Objetivos da Aula:

- Compreender o conceito de **vetor** como uma estrutura de dados fundamental na programação.
- Aplicar a sintaxe de listas do Python para a criação e manipulação de vetores.
- Desenvolver a capacidade de iterar sobre os elementos de um vetor para resolver problemas práticos, como o gerenciamento de estoque.

Recursos Adicionais (Sugestão, pode ser adaptado):

- Caderno para anotações;
- Acesso ao laboratório de informática e/ou internet.

Exposição do Conteúdo:

Referência do Slide: Slide 06 - Construindo o conceito Vetores

- **Definição:** Um vetor, em programação (e implementado como Lista no Python), é uma coleção ordenada de elementos acessíveis por um índice. Ele representa uma estrutura de dados unidimensional, na qual os elementos são dispostos sequencialmente, permitindo que cada valor seja identificado de forma única pela sua posição (que, em Python, começa em zero).
- **Aprofundamento/Complemento (se necessário):** Embora a definição clássica de vetor (em matemática ou em linguagens como C) exija que todos os elementos sejam do mesmo tipo (**homogêneo**), em Python, a lista nativa (**list**) que o representa é **heterogênea**, podendo armazenar números, textos e até outras listas simultaneamente. Para manipulação de grandes volumes de dados numéricos e garantir alta eficiência, utiliza-se a biblioteca NumPy, que trabalha com arrays estritamente homogêneos.
- **Exemplo Prático: Controle de Vendas Diárias**
Um vetor pode registrar o total de vendas realizadas a cada dia da semana.
- **exemplo 1**

```
atividades_praticas_S25 > exemplo_1.py > ...
1 # criando uma lista
2 vendas_semana = [150.75, 200.50, 300.25, 400.00, 250.80]
3 dias_semana = ['Segunda', 'Terça', 'Quarta', 'Quinta', 'Sexta']
4
5 # acessando a vendas de quinta-feira
6 print(f"Vendas de {dias_semana[3]}: R$ {vendas_semana[3]}")
7
8 # Calculando a média de vendas da semana
9 media_vendas = sum(vendas_semana) / len(vendas_semana)
10 print(f"Média de vendas da semana: R$ {media_vendas:.2f}")
11
```

- **Link de Vídeo 1:** [Curso de Python 08 - aprendendo sobre vetores | Prime Cursos Grátis.](#)
- **Link de Vídeo 2:** [O que são Vetores e Listas? - YouTube.](#)

Semana 25 - Aula 2

Tópico Principal da Aula: Matrizes (Arrays Bidimensionais)

Subtítulo/Tema Específico: Conceitos e Prática de Implementação de Matrizes (Listas Aninhadas) em Python

Código da aula: [SIS]ANO1C1B4S25A2

Objetivos da Aula:

- Compreender o conceito de **matriz** como uma estrutura de dados bidimensional.
- Implementar matrizes em Python utilizando a estrutura de **listas aninhadas** (lista de listas).
- Desenvolver habilidades para acessar e manipular elementos em uma matriz usando dois índices (linha e coluna).

Recursos Adicionais (Sugestão, pode ser adaptado):

- Caderno para anotações;
- Acesso ao laboratório de informática e/ou internet.

Exposição do Conteúdo:

Referência do Slide: Slide 06 - Construindo o conceito Matrizes

- **Definição:** Uma **matriz** é uma estrutura de dados **bidimensional** que organiza informações em formato de tabela, composta por **linhas** e **colunas**. No Python, é implementada como uma **lista de listas** (listas aninhadas), onde a lista externa contém as linhas, e cada lista interna contém os elementos das colunas. O acesso é feito pela dupla de índices: `matriz[linha][coluna]`.
- **Aprofundamento/Complemento (se necessário):** Matrizes são cruciais para a representação de dados tabulares (como planilhas), sistemas lineares em matemática, e manipulação de imagens (onde cada pixel é um elemento da matriz). Para manipulação eficiente em Python, a biblioteca NumPy é frequentemente utilizada, fornecendo o objeto `ndarray` de duas dimensões. É importante notar que copiar matrizes de listas nativas exige cuidado para evitar que a cópia compartilhe referências de memória com a original.
- **Exemplo Prático:** Quadro de Notas de Alunos
Uma matriz pode armazenar as notas de três alunos em quatro disciplinas.
- exemplo 2

```

atividades_praticas_S25 > exemplo_2.py > ...
1  # Matriz 3 x 4 (3 Linhas = Alunos; 4 colunas = notas)
2
3  ▾ matriz_notas = [
4      ... [7.5, 8.0, 9.0, 6.5], ... # Notas do Aluno 1 (Linha 0)
5      ... [6.0, 7.5, 7.0, 8.5], ... # Notas do Aluno 2 (Linha 1)
6      ... [9.0, 9.5, 8.5, 8.0] ... # Notas do Aluno 3 (Linha 2)
7  ]
8
9  # acessando a 3 nota do segundo aluno
10 print(f"Terceira nota do Aluno 2: {matriz_notas[1][2]}")
11
12 # Calculando a média do primeiro Aluno(media da linha 0)
13 media_aluno_1 = sum(matriz_notas[0]) / len(matriz_notas[0])
14 print(f"Media do Aluno 1: {media_aluno_1:.1f}")
15

```

- - **Link de Vídeo 1:** [Exercício Python #086 - Matriz em Python | Curso em Vídeo](#)ANO1C1B4S25A2.pdf].
 - **Link de Vídeo 2:** [Como Criar e Manipular Matrizes Em Python: Guia Completo para Iniciantes - Awari](#).

Semana 25 - Aula 3

Tópico Principal da Aula: Operações Vetorizadas

Subtítulo/Tema Específico: Eficiência e Uso da Biblioteca NumPy para Cálculos em Vetores e Matrizes

Código da aula: [SIS]ANO1C1B4S25A3

Objetivos da Aula:

- Definir o conceito de **operações vetorizadas** e contrastá-lo com o uso de *loops* tradicionais.
- Reconhecer a importância da biblioteca **NumPy** para a eficiência computacional em manipulação de vetores e matrizes.
- Aplicar operações matemáticas a coleções de dados de forma concisa e otimizada utilizando a vetorização.

Recursos Adicionais (Sugestão, pode ser adaptado):

- Caderno para anotações;
- Acesso ao laboratório de informática e/ou internet.

Exposição do Conteúdo:

Referência do Slide: Slide 06 - Construindo o conceito Operações vetorizadas

- **Definição: Vetorização** é uma técnica que permite a aplicação de uma operação (soma, multiplicação, função trigonométrica, etc.) a **todos os elementos** de um array (**vetor** ou **matriz**) de forma simultânea e interna, **sem a necessidade de escrever loops de repetição** (for/while). Isso é possível porque o processamento é delegado a código de baixo nível (C ou Fortran), altamente otimizado.
- **Aprofundamento/Complemento (se necessário):** A principal biblioteca de vetorização em Python é o **NumPy**. Ao utilizar o objeto `ndarray` do NumPy, ganhamos eficiência significativa (**performance**) e clareza no código (**legibilidade**). Por exemplo, para somar dois vetores, em vez de iterar sobre cada par de elementos, simplesmente fazemos `vetor_a + vetor_b`. Essa técnica é essencial em áreas como Data Science, Machine Learning e processamento de sinais, onde o tempo de execução é crítico.
- Exemplo Prático: Conversão de Unidade (Celsius para Fahrenheit)
Converter uma lista de temperaturas de Celsius para Fahrenheit ($F = C \times 1.8 + 32$).
- exemplo 3

```
atividades_praticas_S25 > exemplo_3.py > ...
1  import numpy as np
2
3  # temperaturas em celsius (ndarray do Numpy)
4  celsius = np.array([0, 10, 20, 30, 40, 50])
5
6  # operação vetorizada: convertendo para fahrenheit
7  temperaturas_fahrenheit = (celsius * 1.8) + 32
8
9  print(f"Temperaturas em Celsius: {celsius}")
10 print(f"Temperaturas em Fahrenheit: {temperaturas_fahrenheit}")
11
12 # Exemplo de soma vetorizada
13 precos_antigos = np.array([100, 200, 300])
14 aumento = 50
15 preco_novos = precos_antigos + aumento
16 print(f"Preços antigos: {precos_antigos}")
17 print(f"Preços novos após aumento de {aumento}: {preco_novos}")
18
```

- - **Link de Vídeo 1:** [Como usar vetores e matrizes em Python | Biblioteca NumPy | Synapse Data Science](#)ANO1C1B4S25A3.pdf].
 - **Link de Vídeo 2:** [Vetores e Loops. A vetorização em Python, com destaque... | by Maurício Marques | Medium.](#)

Semana 25 - Aula 4

Tópico Principal da Aula: Transposição de Matrizes

Subtítulo/Tema Específico: Transposição de Matrizes para Reorganização de Dados

Código da aula: [SIS]ANO1C1B4S25A4

Objetivos da Aula:

- Definir o conceito matemático de **transposição de matrizes**.
- Aplicar a transposição para reorganizar dados em cenários práticos.
- Implementar a transposição de forma eficiente em Python, utilizando o atributo `.T` ou a função `transpose()` do NumPy.

Recursos Adicionais (Sugestão, pode ser adaptado):

- Caderno para anotações;
- Acesso ao laboratório de informática e/ou internet.

Exposição do Conteúdo:

Referência do Slide: Slide 06 - Construindo o conceito Transposição de Matrizes

- **Definição:** A **transposição de matrizes** (AT) é a operação que consiste em **inverter as linhas e colunas** de uma matriz original. O elemento que estava na posição (i,j) (linha i , coluna j) passa para a posição (j,i) na matriz transposta. Uma matriz 2×3 (2 linhas e 3 colunas) torna-se uma matriz 3×2 (3 linhas e 2 colunas) após a transposição.
- **Aprofundamento/Complemento (se necessário):** A transposição é fundamental em Álgebra Linear e tem vasta aplicação em computação. No **Machine Learning** e **Análise de Dados**, é comum que dados sejam lidos em um formato (ex: variáveis como linhas) e precisem ser transpostos para o formato exigido por um algoritmo (ex: variáveis como colunas). Usando a biblioteca NumPy, a transposição é uma operação muito simples e eficiente, realizada com o atributo `.T` ou a função `np.transpose()`.
 - **Exemplo 4: Reorganização de Dados de Clientes** Uma matriz armazena o Nome, Idade e Cidade de três clientes. A transposição organiza esses dados de forma que as colunas sejam os clientes.

Python

```
atividades_praticas_S25 > exemplo_4.py > ...
1  import numpy as np
2
3
4  # Matriz 3x3 (Linhas = Clientes, Colunas = Dados: Nome, Idade, Cidade)
5  clientes_dados = np.array([
6      ["Ana", 25, "SP"], .....# Cliente 1
7      ["Beto", 35, "RJ"], .....# Cliente 2
8      ["Cida", 45, "MG"], .....# Cliente 3
9  ])
10
11
12  print("Matriz Original (Clientes como Linhas):\n", clientes_dados)
13
14
15  # Transposição para organizar os dados (Clientes como Colunas)
16  dados_transpostos = clientes_dados.T
17
18
19  print("\nMatriz Transposta (Dados como Linhas):\n", dados_transpostos)
20  # Linha 0: Nomes (Ana, Beto, Cida)
21  # Linha 1: Idades (25, 35, 45)
22  # Linha 2: Cidades (SP, RJ, MG)
23
```

•

- **Link de Vídeo 1:** [Matriz 07: matriz transposta | Matemática no Papel.](#)
 - **Link de Vídeo 2:** [Transposição de matriz - Stack Overflow em Português \(Foco em NumPy\).](#)
-