

Vetores


Aula 2: Declaração e Inicialização de Vetores

Na segunda aula, aprofundaremos nosso conhecimento sobre vetores, focando na declaração e inicialização de vetores em Python. Além disso, exploraremos algumas técnicas úteis para trabalhar com vetores.

Declaração de Vetores em Python

Em Python, você pode declarar vetores usando colchetes e inicializá-los com valores. Veja como fazer isso:

python

 Copy code


```
# Declarando e inicializando um vetor de inteiros
numeros = [1, 2, 3, 4, 5]

# Declarando e inicializando um vetor de strings
frutas = ["maçã", "banana", "laranja"]
```

Acesso a Elementos de Vetores

Já aprendemos a acessar elementos de vetores na aula anterior. Lembre-se de que os índices começam em 0. Veja um exemplo:

python


 Copy code

```
frutas = ["maçã", "banana", "laranja"]
primeira_fruta = frutas[0] # Acesso ao primeiro elemento (maçã)
```

Tamanho de Vetores

Para descobrir o tamanho de um vetor (quantos elementos ele contém), você pode usar a função `len()` em Python:

python


 Copy code

```
frutas = ["maçã", "banana", "laranja"]  
tamanho = len(frutas) # Isso retornará 3
```

Adicionando elementos

Você pode adicionar elementos a um vetor usando o método `append()`. Por exemplo:

python


 Copy code

```
numeros = [1, 2, 3]  
numeros.append(4) # Adiciona o número 4 ao vetor
```

Removendo elementos

Você pode remover elementos de um vetor usando métodos como `pop()` ou `remove()`. Por exemplo:

python

 Copy code

```
numeros = [1, 2, 3, 4]  
numeros.pop(1) # Remove o elemento de índice 1 (o número 2)  
numeros.remove(4) # Remove o número 4 da lista
```

Estas são algumas das operações de manipulação mais comuns que você pode aplicar a vetores em Python. Elas são úteis ao trabalhar com dados dinâmicos e podem ser aplicadas em várias situações.