Vetores

Instituto de Computação

Leonardo Murta

leomurta@ic.uff.br

Exemplo Motivacional

Instituto de computação

- Programa para auxiliar a escrever “Parabéns!” nas melhores provas de uma disciplina com 3 alunos
  - Ler os nomes e as notas de 3 alunos
  - Calcular a média da turma
  - Listar os alunos que tiveram nota acima da média

Exemplo Motivacional

Instituto de computação

```
nome1 = input('Informe o nome do aluno 1: ')
nome2 = input('Informe o nome do aluno 2: ')
nome3 = input('Informe o nome do aluno 3: ')

nota1 = float(input(f'Informe a nota de {nome1}: '))
nota2 = float(input(f'Informe a nota de {nome2}: '))
nota3 = float(input(f'Informe a nota de {nome3}: '))

media = (nota1 + nota2 + nota3)/3
print(f'A média da turma é {media:.2f}.')

if nota1 > media:
    print(f'Parabéns {nome1}!')
if nota2 > media:
    print(f'Parabéns {nome2}!')
if nota3 > media:
    print(f'Parabéns {nome3}!')
```

E se fossem 40 alunos?

Instituto de computação

- É possível definir variáveis que guardam mais de um valor
- Essas variáveis são conhecidas por diferentes nomes:
  - *Arrays* (arranjos)
  - Variáveis compostas
  - Variáveis subscritas
  - Variáveis indexáveis
- Elas podem ser unidimensionais (vetores) ou multidimensionais (matrizes)

Vetores

Instituto de computação

- Variável composta **unidimensional**
  - Contém espaço para armazenar diversos valores
  - É acessada via um índice
- A ideia de vetor é comum na matemática, com o nome de variável subscrita
  - Exemplo: x_1, x_2, ..., x_n
- O que vimos até agora são variáveis com somente um valor
  - Exemplo: x = 123
- No caso de vetores, uma mesma variável guarda ao mesmo tempo múltiplos valores
  - Exemplo: $x_1 = 123$, $x_2 = 456$, ...

Listas

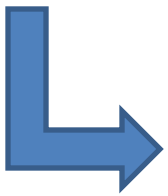
Instituto de computação

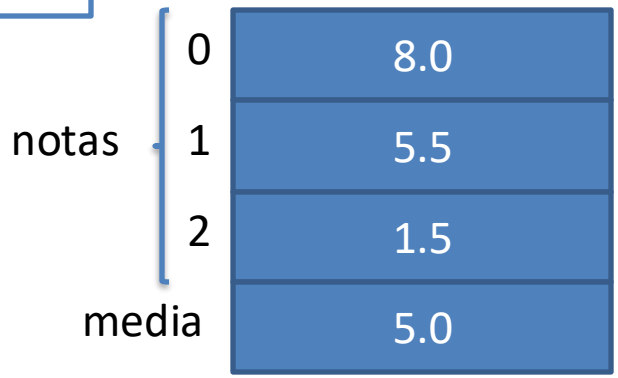
- Em Python, **vetores** são implementados por meio de **listas**
  - Colchetes delimitam o início e o fim da lista
  - Vírgula delimitam os elementos
- Os elementos de uma lista podem ser de qualquer tipo
- Exemplos
  - lista = ['A', 1, 2, 'Casa', 2.3]
  - notas = [10, 5, 6.7, 2, 7.5]

Utilização de listas

- Para acessar (ler ou escrever) uma posição da lista, basta informar a posição entre colchetes
  - A primeira posição zero
  - Similar ao que vimos para string

```
notas = [8.0, 5.5, 1.5]
media = (notas[0] + notas[1] + notas[2]) / 3
```





Utilização de listas

Instituto de computação

- Pode-se iterar por todos os seus valores usando um comando **for**
  - O comando **len()** informa o tamanho da lista
  - Similar ao que vimos para string

```
notas = [8.0, 5.5, 1.5]
for i in range(len(notas)):
    print(notas[i])
```

Criação de uma lista a partir de valores lidos do teclado

Instituto de computação

- Armazenar as notas de 3 alunos em uma lista. A nota de cada aluno será informada pelo teclado.

```
notas[0] = float(input('Digite a nota do primeiro aluno: '))
notas[1] = float(input('Digite a nota do segundo aluno: '))
notas[2] = float(input('Digite a nota do terceiro aluno: '))
```

```
Traceback (most recent call last):
  File "/home/leomurta/workspace/prog/teste.py", line 1, in <module>
    notas[0] = float(input('Digite a nota do primeiro aluno: '))
NameError: name 'notas' is not defined
```

Criação de uma lista a partir de valores lidos do teclado

Instituto de computação

- Como não sabemos o que colocar em cada posição da lista, vamos criar uma lista vazia

```
notas = []
notas[0] = float(input('Digite a nota do primeiro aluno: '))
notas[1] = float(input('Digite a nota do segundo aluno: '))
notas[2] = float(input('Digite a nota do terceiro aluno: '))
```

```
Traceback (most recent call last):
  File "/home/leomurta/workspace/prog/teste.py", line 2, in <module>
    notas[0] = float(input('Digite a nota do primeiro aluno: '))
IndexError: list assignment index out of range
```

Criação de uma lista a partir de valores lidos do teclado

Instituto de computação

- É necessário usar o comando **append** para adicionar novos elementos na lista

```
notas = []
notas.append(float(input('Digite a nota do primeiro aluno: ')))
notas.append(float(input('Digite a nota do segundo aluno: ')))
notas.append(float(input('Digite a nota do terceiro aluno: ')))
```

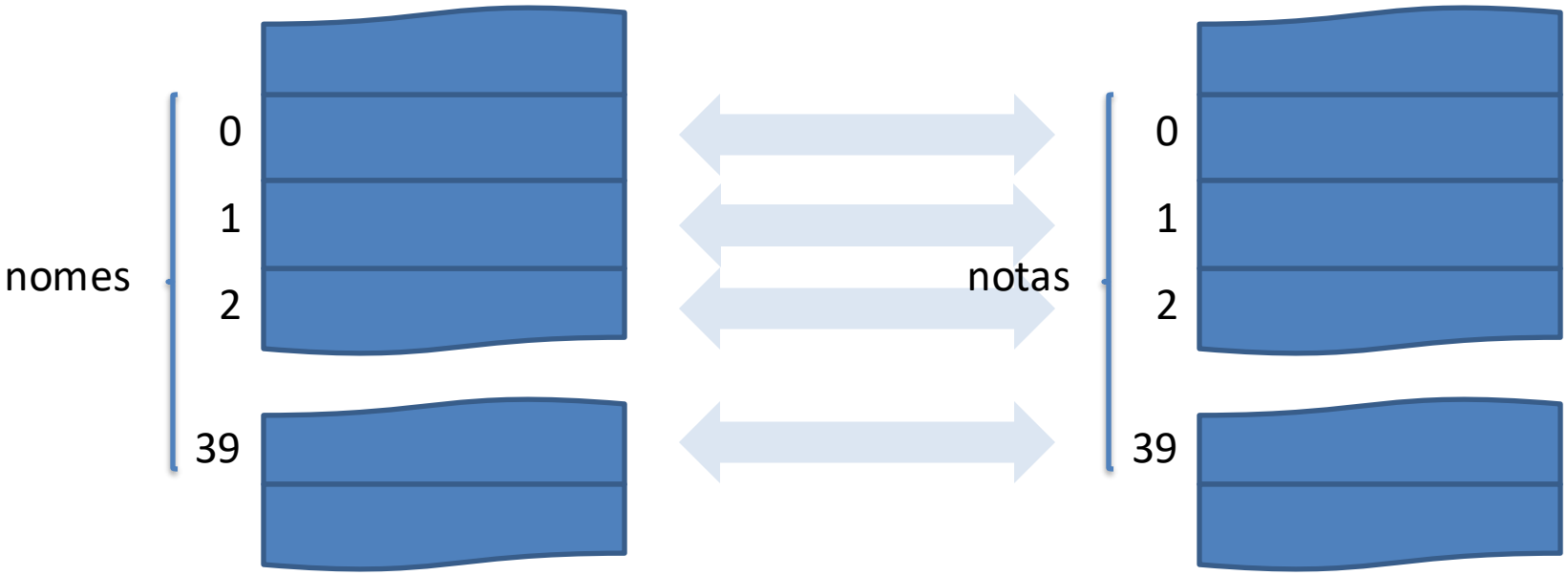
Exemplo

Instituto de computação

```
notas = []
print(notas)        → []
notas.append(8)
print(notas)        → [8]
notas.append(9)
print(notas)        → [8, 9]
notas.append(6)
print(notas)        → [8, 9, 6]
notas[1] = 10
print(notas)        → [8, 10, 6]
```

Retomando: E se fossem 40 alunos?

- Criaríamos dois vetores (nomes e notas) de 40 posições
- Vincularíamos a posição *N* do vetor de nomes à posição *N* do vetor de notas



Retomando: E se fossem 40 alunos?

Instituto de computação

```
num_alunos = 40
nomes = []
notas = []
soma = 0

for i in range(num_alunos):
    nomes.append(input(f'Informe o nome do aluno {i+1}: '))
    notas.append(float(input(f'Informe a nota de {nomes[i]}: ')))
    soma = soma + notas[i]

media = soma / num_alunos
print(f'A média da turma é {media:.2f}.')

for i in range(num_alunos):
    if notas[i] > media:
        print(f'Parabéns {nomes[i]}!')
```

Exercícios

Instituto de computação

1. Faça um programa que leia dois vetores de 3 posições, que representam forças sobre um ponto no espaço 3D, e escreva a força resultante
   - Dica: força resultante é a soma dos valores das coordenadas correspondentes nos dois vetores: $(x_1 + x_2)$, $(y_1 + y_2)$, $(z_1 + z_2)$
2. Faça um programa que preencha por leitura um vetor de 10 posições, e conta quantos valores diferentes existem no vetor.
3. Faça um programa que preencha por leitura um vetor de 5 posições, e informe a posição em que um valor x (lido do teclado) aparece pela primeira vez no vetor. Caso o valor x não seja encontrado, o programa deve imprimir o valor -1

Exercícios

Instituto de computação

4. Um dado é lançado 50 vezes, e o valor correspondente é armazenado em um vetor. Faça um programa que simule o lançamento do dado e determine o percentual de ocorrências de face 6 do dado dentre esses 50 lançamentos.
5. Faça um programa que leia um vetor **vet** de 20 posições. O programa deve gerar, a partir do vetor lido, um outro vetor **pos** que contenha apenas os valores inteiros positivos de **vet**. A partir do vetor **pos**, deve ser gerado um outro vetor **semdup** que contenha apenas uma ocorrência de cada valor de **pos**.

Exercícios

Instituto de computação

6. Leia um vetor de 10 posições e ordene o vetor, usando 3 algoritmos diferentes (crie um programa para cada algoritmo)
   a. Insertion Sort
   b. Selection Sort
   c. Bubble Sort

   - Revisite a aula 3, de Introdução à Programação, para relembrar os algoritmos
   - Em cada alternativa, conte o número de comparações realizadas, e imprima o número de comparações junto com o vetor ordenado
   - Observe qual dos algoritmos executou a ordenação com o menor número de comparações

Referências

- Slides feitos em conjunto com Vanessa Braganholo e Aline Paes

Vetores

Instituto de Computação

Leonardo Murta

leomurta@ic.uff.br