

Estrutura de Dados e Programação Orientada a Objetos

Prof^a. Dr^a. Alana Marques
alanamm.prof@gmail.com

Objetivo da aula

Entender conceitos iniciais relacionados às estruturas de dados

O que é estrutura de dados?



Estrutura de dados (1)

É o modo particular de **armazenamento** e **organização** dos dados, a fim de melhorar a eficiência de suas manipulações

Estrutura de dados(2)

É uma **coleção** que pode ser caracterizada por sua organização e pelas operações que a definem

Problemas que podem ser resolvidos com estrutura de dados

Soluções (com algoritmos) que necessitem de uma coleção de valores que estão relacionados entre si

Exemplo inicial

Crie um programa que armazene uma sequência de valores (de tamanho dinâmico) **numa única variável, chamada registro**. Feito isso, mostre qual é o valor médio dessa sequência.

Estrutura de dados

Dados simples:

- padrão:
 - inteiro (int);
 - real (float);
 - caracter (str);
 - lógico (boolean).

Estrutura de dados

Dados simples:

- padrão:
 - inteiro (int);
 - real (float);
 - caracter (str);
 - lógico (boolean).

Dados estruturados:

- Estáticos:
 - arrays;
 - registros;
 - arquivos;
 - conjuntos;
 - cadeias.
- Dinâmicos:
 - filas;
 - pilhas;
 - listas encadeadas;
 - árvores;
 - grafos.

Estrutura de dados

Dados simples:

- padrão:
 - inteiro (int);
 - real (float);
 - caracter (str);
 - lógico (boolean).

Dados estruturados:

- Estáticos:
 - arrays;
 - registros;
 - arquivos;
 - conjuntos;
 - cadeias.
- Dinâmicos:
 - filas;
 - pilhas;
 - listas encadeadas;
 - árvores;
 - grafos.

Estrutura de dados

Dados simples:

- padrão:
 - inteiro (int);
 - real (float);
 - caracter (str);
 - lógico (boolean).

Dados estruturados:

- Estáticos:

- arrays;
- registros;
- arquivos;
- conjuntos;
- cadeias.

ou vetor

- Dinâmicos:

- filas;
- pilhas;
- listas encadeadas;
- árvores;
- grafos.

Arrays

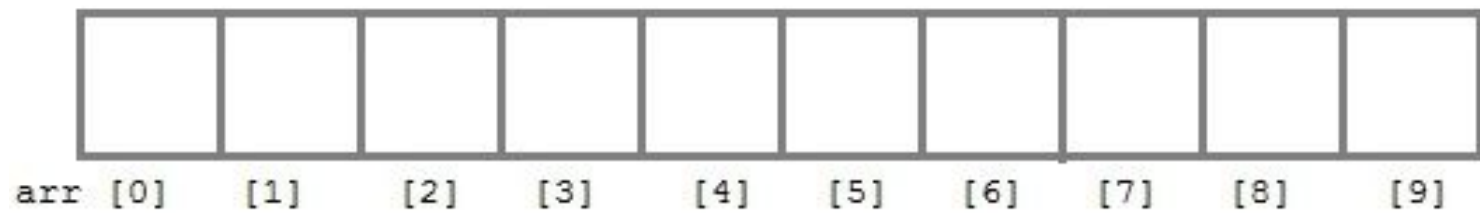
- conjunto **finito** de elementos;
- também conhecidos como vetores.
- Em Python:

```
valores = [1, 2, 3, 5]
```

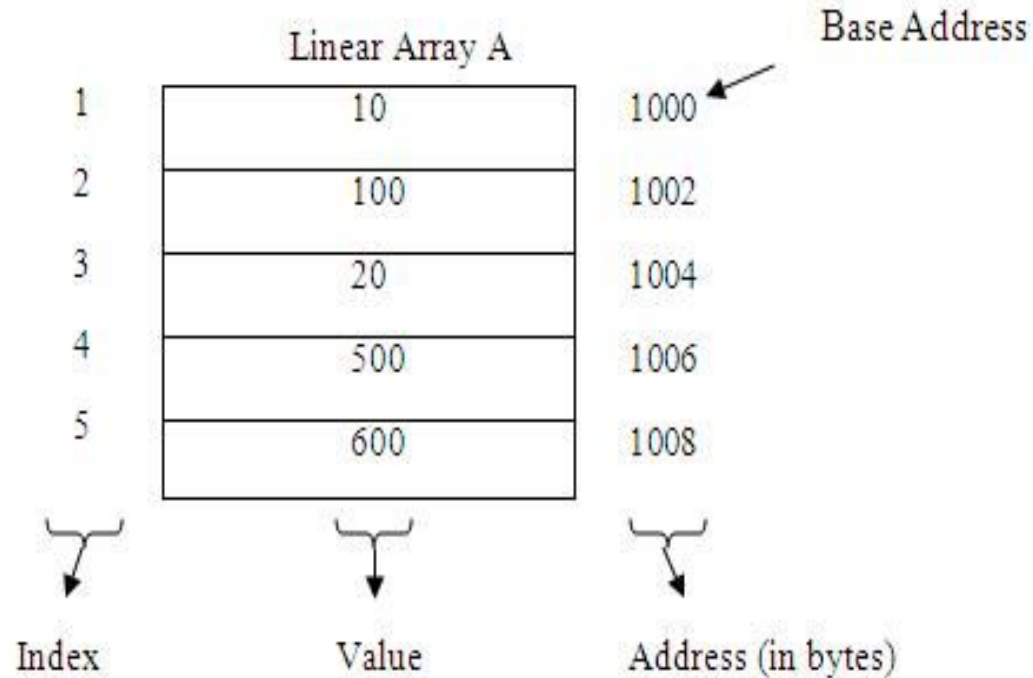
```
nomes = ["Aline", "Eduardo", "José", "Mateus"]
```

```
mistura = ["joão", 15, 1.78, "brasileiro", True]
```

Array



Array na memória RAM



Funções importantes

- Saber o tamanho de um vetor;
- Percorrer todos os valores de um vetor;
- Acessando elementos de um vetor;
- Operações com vetores;
- Concatenação de vetores;
- Remoção de elementos dos vetores;
- Listar elementos específicos dos vetores.

Funções importantes

- **Saber o tamanho de um vetor;**
- Percorrer todos os valores de um vetor;
- Acessando elementos de um vetor;
- Operações com vetores;
- Concatenação de vetores;
- Remoção de elementos dos vetores;
- Listar elementos específicos dos vetores.

Saber o tamanho de um vetor

```
def main():  
  
    universidades = ["IESP", "UFCG", "UFPB", "IFPB"]  
  
    tamanho = len(universidades)  
  
    print(tamanho)  
  
main()
```

Funções importantes

- Saber o tamanho de um vetor;
- **Percorrer todos os valores de um vetor;**
- Acessando elementos de um vetor;
- Operações com vetores;
- Concatenação de vetores;
- Remoção de elementos dos vetores;
- Listar elementos específicos dos vetores.

Percorre todos os valores do vetor

```
def main():  
  
    universidades = ["UEPB", "UFCG", "UEPB", "IFPB"]  
  
    for item in universidades:  
  
        print(item)  
  
main()
```

Exercício A

Crie um array (vetor) com os seguintes valores: **0, 1, 2, 3 e 4**. Em seguida, percorra o array e mostre cada valor dele multiplicado por 4.

Funções importantes

- Saber o tamanho de um vetor;
- Percorrer todos os valores de um vetor;
- **Acessando elementos de um vetor;**
- Operações com vetores;
- Concatenação de vetores;
- Remoção de elementos dos vetores;
- Listar elementos específicos dos vetores.

Acessando elementos de um vetor

```
def main():  
  
    universidades = ["UEPB", "UFCG", "UEPB", "IFPB"]  
  
    print(universidades[0])  
  
    print(universidades[2])  
  
main()
```

Acessando elementos de um vetor

```
def main():  
  
    universidades = ["IESP", "UFCG", "UFPB", "IFPB"]  
  
    print(universidades[0])  
  
    print(universidades[2])  
  
main()
```

Não há índices negativos!

Acessando elementos de um vetor

```
def main():  
    0           1           2           3  
    universidades = ["IESP", "UFCG", "UFPB", "IFPB"]  
  
    print(universidades[0])  
  
    print(universidades[2])  
  
main()
```

Não há índices negativos!

Exercício B

Num programa python, crie um array com os seguintes valores “**iesp**”, “**2019.1**”, “**SI**”, “**ED**”. Em seguida, imprima **apenas** o segundo valor do array (é o valor 2019.1).

Funções importantes

- Saber o tamanho de um vetor;
- Percorrer todos os valores de um vetor;
- Acessando elementos de um vetor;
- **Operações com vetores;**
- Concatenação de vetores;
- Remoção de elementos dos vetores;
- Listar elementos específicos dos vetores.

Operações com vetores

- Leitura
- Criação
- Inserção de dados
- Remoção de dados
- Reordenamento de dados

Operações com vetores

- Leitura
- Criação
- Inserção de dados
- Remoção de dados
- Reordenamento de dados

Leitura - 1ª forma

```
def main():  
    tam = int(input("Quantos elementos no array?"))  
    conjunto = []  
    for i in range(0, tam):  
        elemento = input("elemento "+str(i+1)+": ")  
        conjunto.append(elemento)  
    print(conjunto)  
main()
```

Função append

Adiciona um elemento em um array existente

```
info = [123, 'xyz', 'zara', 'abc']  
info.append(2009)  
print ("Array atualizado : ", info)
```

```
Array atualizado : [123, 'xyz', 'zara', 'abc', 2009]
```

Leitura - 2ª forma

```
def main():  
    a = [x for x in input("Digite os elementos da array,  
separados por espaço: ").split()]  
    print(a)  
  
main()
```

Leitura - 2ª forma

```
def main():  
    a = [x for x in input("Digite os elementos do array,  
separados por espaço: ").split()]  
    print(a)  
  
main()
```

Existem inúmeras formas de implementar a leitura de um array!

Exercício C

Crie um programa que o usuário digite uma sequência de valores (de tamanho dinâmico) **numa única variável, chamada registro**. Em seguida, mostre os valores armazenados.

Operações com array

- Leitura
- **Criação**
- Inserção de dados
- Remoção de dados
- Reordenamento de dados

Criação da array (1)

```
conjunto = []
```

```
valores = [1,2,3,5]
```

```
nomes = ["Alana","Aline","José","Mateus"]
```

```
conjunto = ["joão", 15, 1.78, "brasileiro", True]
```

Criação da array (2)

```
conjunto = list()
```

```
valores = list((1,2,3,5))
```

```
nomes = list(("Alana","Aline","José","Mateus"))
```

```
conjunto = list(("joão", 15, 1.78, "brasileiro", True))
```

Exercício D

Elabore um programa que cria um array de nomes de países e outro array com nomes de cidades. Cada uma das criações devem ser feitas com métodos diferentes.

Operações com vetores

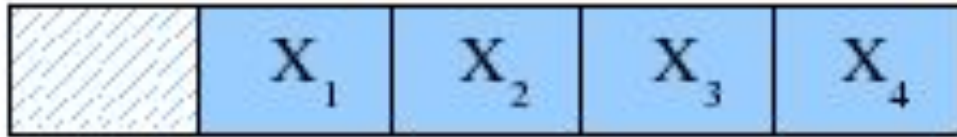
- Leitura
- Criação
- **Inserção de dados**
- Remoção de dados
- Reordenamento de dados

Inserção de valores no vetores

```
def main():  
    conjunto = [123, 'alana', 'ifpb', 'RC']  
    conjunto.append('2019.2')  
    print(conjunto)  
    conjunto.insert(0, 'ED')  
    print(conjunto)  
  
main()
```

Função append

Alocação sequencial



Função insert

Adiciona um elemento **numa posição específica** num array existente

```
def main():  
    conjunto = list((123, 'alana', 'ifpb', 'RC'))  
    conjunto.append('2019.2')  
    conjunto.insert(0, 'ED')  
    print(conjunto)
```

```
main()
```

```
['ED', 123, 'aline', 'iesp', 'SI', '2018.1']
```

Exercício E

Crie um programa no qual o usuário digitará 5 nomes. Todos os cinco nomes serão salvos no vetor chamado “registros”. Em seguida:

- a) Mostre o que o usuário digitou na tela;
- b) O programa pedirá outro nome, que será adicionado no fim do vetor “registros”;
- c) O programa pedirá outro nome, que será adicionado na 2ª posição do vetor “registros”;
- d) Imprima o resultado na tela.

Extra: para saber a posição de um valor no vetor

```
nomes = ["Sarah", "Alana", "Aline", "Maria",  
"Alberto"]
```

```
pos = nomes.index("Aline")
```

```
print(pos)
```

Exercício F

Crie um programa no qual o usuário digitará 5 nomes num vetor (OBRIGATÓRIO TER SEU NOME INCLUÍDO NO VETOR). Em seguida:

- a) mostre os valores na tela.
- b) adicione seu sobrenome na posição do seu nome no array.
- c) imprima o resultado na tela.

Operações com vetores

- Leitura
- Criação
- Inserção de dados
- **Remoção de dados**
- Reordenamento de dados

Remoção de valores nos vetores

Remover algum(us) item(ns) do array:

```
def main():  
    conjunto = [123, 'alana', 'ifpb', 'RC']  
    conjunto.remove(123)  
    print(conjunto)  
  
main()
```

```
['alana', 'ifpb', 'RC']
```

Função remove()

Remove item(ns) de um array

```
def main():  
    animal = ['gato', 'cão', 'coelho', 'galinha']  
    animal.remove('coelho')  
    print( animal)  
main()
```

Remoção de valores no array

Remover todos os valores do array:

```
def main():  
    conjunto = [123, 'alana', 'ifpp', 'RC']  
    conjunto.clear()  
    print(conjunto)  
  
main()
```

[]

Função clear()

Remove todos os dados do array:

```
def main():  
    animal = ['gato', 'cão', 'coelho', 'galinha']  
    animal.clear()  
    print( animal)  
main()
```

Exercício G

Crie um array que o conteúdo seja nomes das cores. Em seguida remova apenas a cor “vermelho”, se houver.

Operações com vetores

- Leitura
- Criação
- Inserção de dados
- Remoção de dados
- **Reordenamento de dados**

Reordenamento de dados (1)

Ordena os valores dos dados no array:

```
def main():  
    numeros = [10, 4, 2, 0, 1, 3, 4, 5]  
    print(numeros)  
    numeros.sort()  
    print(numeros)  
  
main()
```

Reordenamento de dados (2)

Ordena os valores dos dados no array de ordem decrescente:

```
def main():  
    numeros = [10, 4, 2, 0, 1, 3, 4, 5]  
    print(numeros)  
    numeros.sort(reverse = True)  
    print(numeros)
```

Exercício H

Crie um array referente aos anos de nascimento das 5 pessoas mais próximas a você. Em seguida:

- a) Ordene o array na ordem crescente e mostre o resultado;
- b) Ordene o array na ordem decrescente e mostre o resultado;

Funções importantes

- Saber o tamanho de um vetor;
- Percorrer todos os valores de um vetor;
- Acessando elementos de um vetor;
- Operações com vetores;
- **Concatenação de vetores;**
- Remoção de elementos dos vetores;
- Listar elementos específicos dos vetores.

Concatenação de arrays

```
def main():  
  
    array1 = [1, 2, 3, 4]  
  
    array2 = [10, 9, 8, 7]  
  
    array3 = array1 + array2  
  
    print(array3)  
  
main()
```


Concatenação de arrays

```
def main():
```

```
    array1 = [1, 2, 3, 4]
```

```
    array2 = [10, 9, 8, 7]
```

```
    array3 = array1 + array2
```

```
    print(array3)
```

```
main()
```

```
[1, 2, 3, 4, 10, 9, 8, 7]
```

Funções importantes

- Saber o tamanho de um vetor;
- Percorrer todos os valores de um vetor;
- Acessando elementos de um vetor;
- Operações com vetores;
- Concatenação de vetores;
- **Remoção de elementos dos vetores;**
- Listar elementos específicos dos vetores.

Remoção de elementos do array

```
def main():  
  
    array1 = [10,20,30,40]  
  
    print(array1)  
  
    array1.pop(0)  
  
    print(array1)  
  
main()
```

Remoção de elementos do array

```
def main():
```

```
    array1 = [10,20,30,40]
```

```
    print(array1)
```

```
    array1.pop(0)
```

```
    print(array1)
```

```
main()
```

```
[10,20,30,40]
```

```
[20,30,40]
```

Remoção de elementos do array

```
def main():
```

```
    array1 = [10,20,30,40]
```

```
    print(array1)
```

```
    array1.pop(0)
```

índice



```
    print(array1)
```

```
main()
```

```
[10,20,30,40]
```

```
[20,30,40]
```

Funções importantes

- Saber o tamanho de um vetor;
- Percorrer todos os valores de um vetor;
- Acessando elementos de um vetor;
- Operações com vetores;
- Concatenação de vetores;
- Remoção de elementos dos vetores;
- **Listar elementos específicos dos vetores.**

Listar elementos específicos dos vetores

```
def main():
```

```
    array = [10,20,30,40]
```

```
    print("último elemento: ", array[-1])
```

```
    print("2° e 3° elementos: ", array[1:2])
```

```
main()
```

```
último elemento: 40
```

```
2° e 3° elemento: 20,30
```

Atividade

Crie uma agenda eletrônica que contenha dois vetores: um array com o nome dos seus amigos e outro com as suas respectivas datas de aniversário.

Tente inserir e remover as informações da agenda de maneira dinâmica (apagando pelo nome ou posição, inserindo n usuários por vez caso queira, etc...).

Quando terminar, mande o código para o classroom (até quarta-feira).

Dúvidas?



alanamm.prof@gmail.com