ESTRUTURA DE DECISÃO E LISTAS/VETORES/ARRAYS

Ruan Pablo Medeiros

ROTEIRO

- Estrutura Condicional
  - Escolha ... caso (match-case)
- Listas/Vetores/Arrays

PYTHON – MATCH CASE

```
op = input("Escolha entre [T]riângulo, [Q]uadrado, [R]etangulo, [L]osango ou [C]írculo")

match op:
    case "T":
        print("Você escolheu Triângulo")
    case "Q":
        print("Você escolheu Quadrado")
    case "C":
        print("Você escolheu Círculo")
    case "L":
        print("Você escolheu Losango")
    case default:
        print("opção invalida")
```

PYTHON – LISTAS/VETORES/ARRAYS

- Pode ser definido como sendo um conjunto de variáveis do mesmo tipo. São também conhecidos por listas ou array;
- Strings em algumas linguagens são classificados como vetores, já que são uma sequência de caracteres;
- Sua declaração normalmente segue o mesmo formato de variáveis, porém acrescenta-se o número de elementos desejados entre colchetes ([e]) para acessar os elementos do vetor.

VETOR

- idade alunos

[12 , 18, 34, 32, 27, 15, 78, 31, 41, 26]

- nome

['J', 'O', 'V', 'E', 'M', ' ', 'P', 'R', 'O', 'G', 'R', 'A', 'M', 'A', 'D', 'O', 'R']

- Assim como todas as variáveis, os vetores possuem um identificador que serve para poder acessar os dados do vetor.

PYTHON - VETOR

- Variável composta unidimensional
  - Contém espaço para armazenar vários valores;
  - É acessada através de um índice.
- A ideia de vetor é bem comum na matemática, algumas vezes denominada de variável subscrita Exemplo: x1,x2, x3, ... ,Xn
- O que vimos até agora são variáveis que só podem conter um único valor. Exemplo: nota = 8. 6
- No caso de vetores, uma variável acessa dados de múltiplos valores. Exemplo: x1, = 12, Xr 73, ...

PYTHON - LISTA

- Em outras linguagens de programação, listas são sinônimos de vetores ou arrays, não possuindo restrições que o Python impõe.
- Por outro lado, em Python, os valores das listas podem ser de qualquer tipo. Na grande maioria das linguagens, os valores dos arrays precisam ser sempre do mesmo tipo.
- lista = ['A', 1, 2, 'casa', 2.3]

PYTHON - LISTA

- Para acessar (ler ou escrever) cada membro de uma lista, basta informar a posição entre colchetes.
- Os elementos são enumerados de 0 (zero) até o número especificado como quantidade menos 1 (n-1).

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

PYTHON - LISTA

- É possível iterar por todos os valores de uma lista usando o comando for.

```
notas = [8.0, 5.5, 1.5)
for i in range(3):
        print (notas [i])
```

PYTHON - LISTA

- É preciso sempre criar a lista!
- Como não sabemos o que colocar em cada posição da lista, podemos criar uma lista com valores quaisquer e depois alterar

```
notas = [0.0, 0.0, 0.0]
notas[0] = float(input("Primeira nota: ")
notas[1] = float(input("Segunda nota: ")
notas[2] = float(input("Terceira nota: ")
```

PYTHON – LISTA

- Geralmente não sabemos o que colocar na lista e não sabemos qual a quantidade de elementos ela terá: para isso, criamos uma lista vazia.
- Depois adicionamos valores na lista usando append.

```
notas = []
notas.append(float(input('Primeira nota:')))
notas.append(float(input(Segundanota:'))
notas.append(float(input(Terceira nota:'))
```

PYTHON – CUIDADOS NO USO DE LISTAS

- Certifique-se de que esteja sempre acessado posições válidas da lista

```
notas = [8.0, 5.5, 1.5]
notas[1] = 8.5
notas[2] = 5.0
notas[3] = 1.5
```

IndexError: list assignment índex out of range

PYTHON – LISTA

- Para acessar (para ler ou escrever) cada membro de uma lista, basta informar a posição entre colchetes.
- Os elementos são enumerados de 0 (zero) até o número especificado como quantidade menos 1 (n-1).
- Python permite acesso à lista em ordem decrescente. Último elemento é -1.

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

PYTHON - LISTA

- Para manipular listas é sempre bom saber o tamanho dela. O comando **len (<lista>)** retorna o tamanho da lista, facilitando a iteração por todos os valores da lista.

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

PYTHON - LISTA

- Para iterar uma lista, também é possível fazer diretamente usando a própria lista no for.
- Iterando diretamente:

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

PYTHON - LISTA

- É possível anexar valores de uma lista em outra (concatenar) usando o operador'+':

```
>>> lista = [1,2,3,4]
>>> lista + [5,6]
[1, 2, 3, 4, 5, 6]
>>> lista + [4,5,6]
[1, 2, 3, 4, 4, 5, 6]
```

PYTHON - LISTA

- O operador de multiplicação '*' repete n vezes os elementos que estão na lista.
- lista * n equivale a lista + lista + ... + lista (n vezes)

```
>>> lista = [1,2,3,4]
>>> lista * 3
[1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4]
```

PYTHON - LISTA

- Em diversas situações, nas quais já sabemos de antemão qual será o tamanho da lista, é útil inicializar a lista com valor 0 (zero). Isso evita que seja preciso usar o comando append para adicionar valores.

```
>>> tamanho = 10
>>> lista = [0] * tamanho
>>> lista
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

PYTHON - EXEMPLO

- Criando uma lista com tamanho n informado pelo usuário, preenchendo com valores de 1 até n.

```
n = int(input('Qual o tamanho da lista?'))
lista = [0] * n
for i in range(n):
    lista [i] = i+1
print(lista)
```

EXERCÍCIOS

1. Fazer um algoritmo para ler um vetor de 8 números inteiros e imprimir na tela os conteúdos do vetor lido (mesma ordem).
2. Altere o exercício anterior para mostrar os números lidos na ordem inversa da leitura.
3. Altere o primeiro exercício para também mostrar a soma de seus elementos e apresentar quantos deles são positivos.
4. Escreva um algoritmo para ler nome e nota de uma turma de até 30 alunos . Depois exiba uma lista com o nome dos alunos com nota maior que 7.
5. Faça um programa para preencher com leitura um vetor de 5 posições e informe a posição em que um valor x (também lido do teclado) está no vetor. Caso o valor não seja encontrado, o programa deve imprimir o valor -1.