IEC037
Introdução à Programação de Computadores
Aula 13 – Tabelas em Python

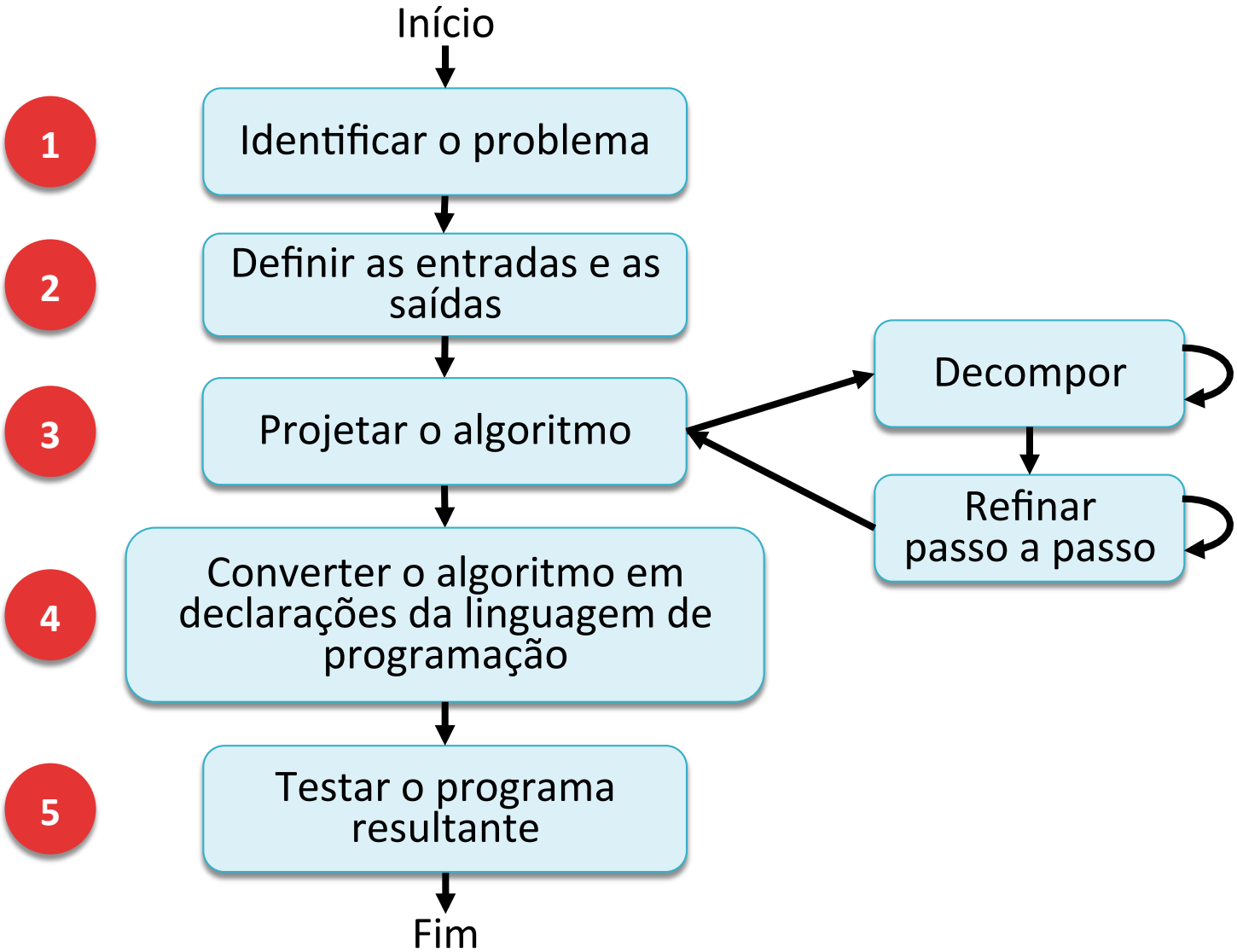
Professor: André Carvalho

Sala: 1211

E-mail:andre@icomp.ufam.edu.br

Página: http://iccufam.wordpress.com

Resolução de Problemas Algorítmicos



Conteúdo

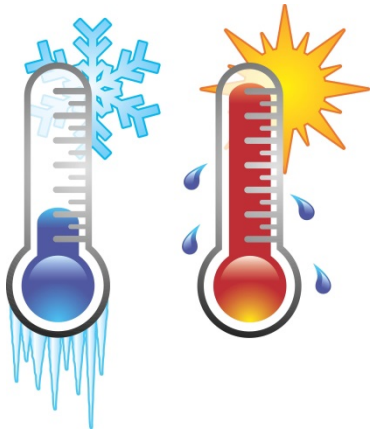
- O que é uma Tabela em Python?
- Como acessar elementos da Tabela?
- Entrada de valores
- Problemas envolvendo Tabelas

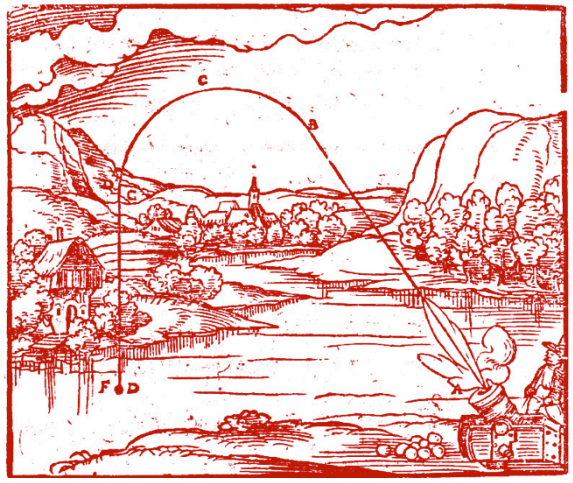
Conteúdo

- O que é uma Tabela em Python?
- Como acessar elementos da Tabela?
- Entrada de valores
- Problemas envolvendo Tabelas

Dados bidimensionais

- Algumas aplicações demandam que os dados sejam organizados em um formato de tabela bidimensional:
  - Temperatura medida em diferentes localidades e em diversos momentos.
  - Tempo de disparo de um projétil e as respectivas posições x e y no espaço.





Tabelas

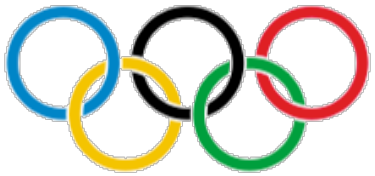
- Arranjos de dados organizados em linhas e colunas são chamados de matrizes, ou tabelas no Python.
- Python não tem uma estrutura específica para criar tabelas.
- Por isso, utilizamos uma lista de listas para criar uma estrutura de linhas e colunas.

Como criar uma tabela?

- Seja o quadro de medalhas dos sete primeiros colocados nas Olimpíadas de 2012.

	Ouro	Prata	Bronze
EUA	46	29	29
China	38	27	23
Grã-Bretanha	29	17	19
Rússia	24	26	32
Coreia do Sul	13	08	07
Alemanha	11	19	14
França	11	11	12

```
quadro = [
[46, 29, 29],
[38, 27, 23],
[29, 17, 19],
[24, 26, 32],
[13,  8,  7],
[11, 19, 14],
[11, 11, 12]
]
```



Como criar uma tabela?

- Tabelas são organizadas em linhas.
- Uma tabela é criada como uma lista de linhas de um arranjo tabular.

```
quadro = [
[46, 29, 29],
[38, 27, 23],
[29, 17, 19],
[24, 26, 32],
[13,  8,  7],
[11, 19, 14],
[11, 11, 12]
]
```

Linha

Quais as dimensões de uma tabela?

- Número de linhas (comprimento da tabela):

```
NLIN = len(quadro)
```

- Número de colunas (comprimento de uma das linhas):

```
NCOL = len(quadro[0])
```

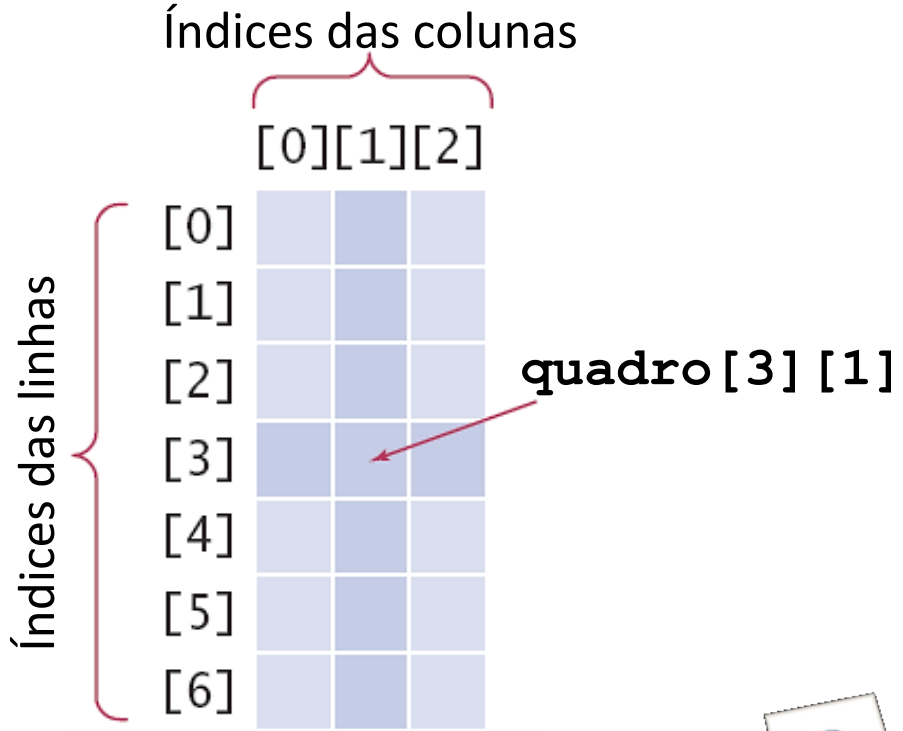
Conteúdo

- O que é uma Tabela em Python?
- **Como acessar elementos da Tabela?**
- Entrada de valores
- Problemas envolvendo Tabelas

Como acessar elementos da Tabela?

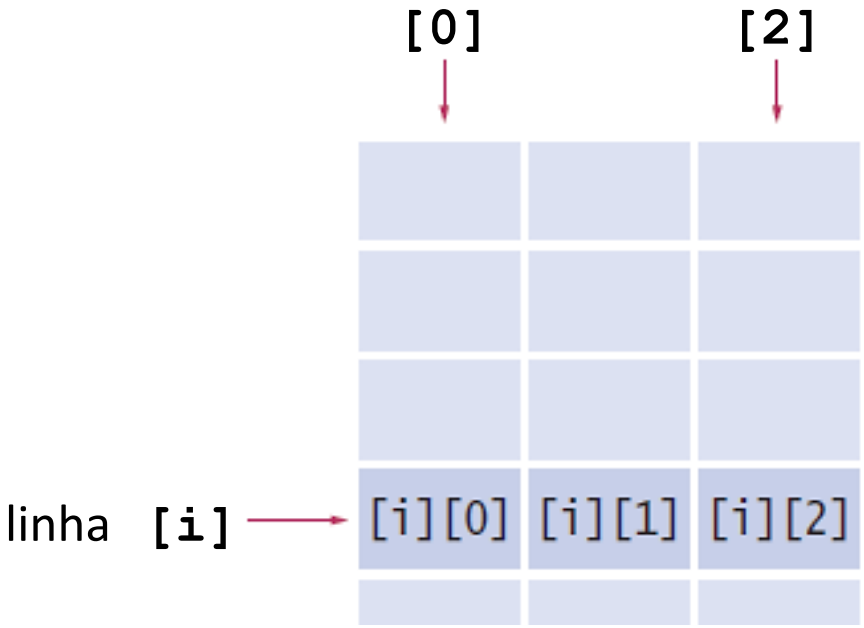
- Indique o índice da linha e o índice da coluna, cada um entre colchetes, e nessa ordem.

```
x = quadro[3][1]
```



Computando o total de uma linha

- Qual o total de medalhas de um país `i`?



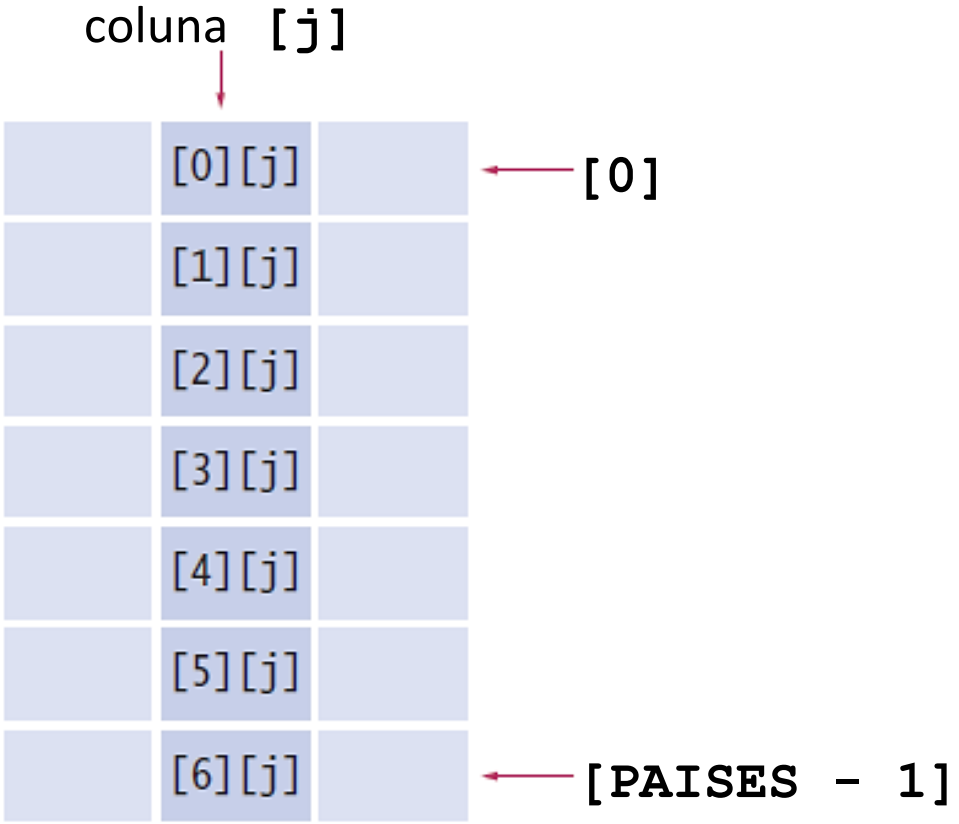
Computando o total de uma linha :: Script

- Qual o total de medalhas da Rússia (linha 3)?

```
# Constantes (facilitam manutencao)
PAISES   = len(quadro)
MEDALHAS = len(quadro[0])

total = 0

# Processa a j-esima coluna na linha 3
for j in range(MEDALHAS):
    total = total + quadro[3][j]

print(total)
```

Computando o total de uma coluna

- Qual o total de medalhas do tipo `j` entre os sete países?



Computando o total de uma coluna :: Script

- Qual o total de medalhas de ouro entre os sete primeiros colocados?

```
# Constantes (facilitam manutencao)
PAISES   = len(quadro)
MEDALHAS = len(quadro[0])

ouro = 0

# Processa a i-esima linha na coluna 0
for i in range(PAISES):
    ouro = ouro + quadro[i][0]

print(ouro)
```

Seleção de elementos de uma linha

- Seleção de todos os elementos da linha **i**:

```
x = quadro[i][:]
```

- Seleção dos **n** primeiros elementos da linha **i**:

```
x = quadro[i][:n]
```

- Seleção dos **n** últimos elementos da linha **i**:

```
x = quadro[i][-n:]
```

Seleção de elementos de uma coluna

- Tabelas são listas de linhas. Por isso, os elementos de uma coluna em específico devem ser manipulados um a um, através de um laço.
- Portanto, nenhum dos comandos abaixo terão o efeito desejado para a coluna `j`:

```
x = quadro[:][j]
```

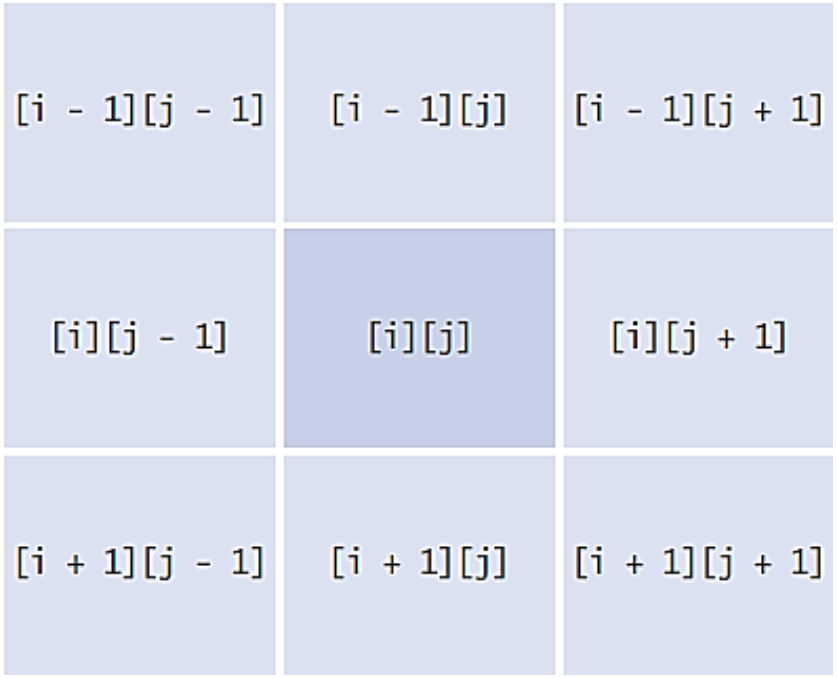
```
x = quadro[:n][j]
```

```
x = quadro[-n:][j]
```

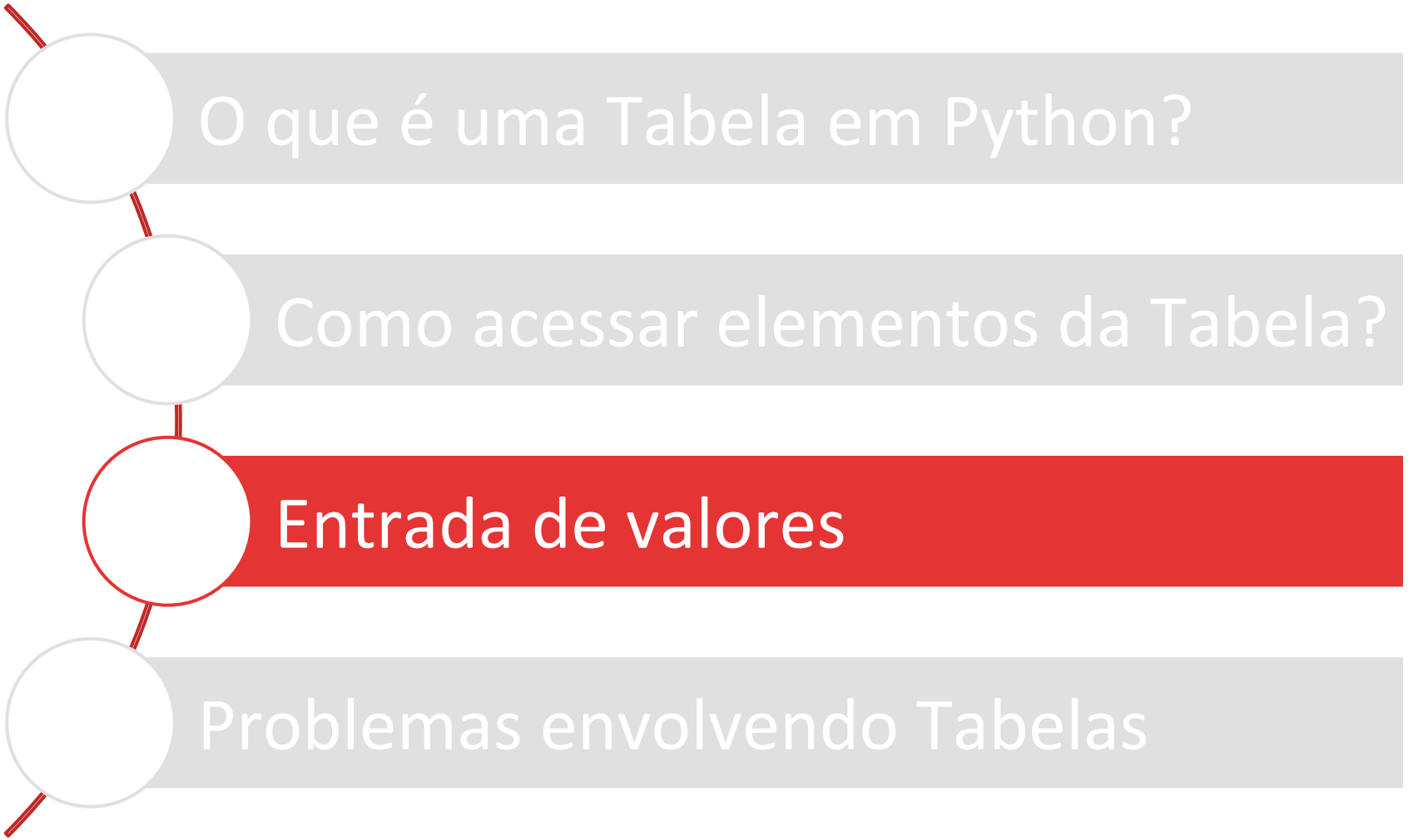
Localizando elementos adjacentes

- Alguns programas que trabalham com tabelas precisam de localizar os elementos adjacentes a outro elemento `[i][j]`.

Cuidado para não obter índices negativos



Conteúdo


O que é uma Tabela em Python?
Como acessar elementos da Tabela?
Entrada de valores
Problemas envolvendo Tabelas

Entrada de valores

:: Via teclado

```
mat  = []       # Cria uma tabela vazia
NLIN = int(input("Qual o no. de linhas? "))
NCOL = int(input("Qual o no. de colunas? "))

for i in range(NLIN):
    mat.append([])  # Anexa nova linha vazia
    for j in range(NCOL):
        valor = int(input("Valor do elemento: "))
        mat[i].append(valor)

# Imprime tabela
print(mat)
```

Entrada de valores

:: Valores aleatórios

```
from random import *

mat  = []       # Cria uma tabela vazia
NLIN = int(input("Qual o no. de linhas? "))
NCOL = int(input("Qual o no. de colunas? "))

for i in range(NLIN):
    mat.append([])  # Anexa nova linha vazia
    for j in range(NCOL):
        mat[i].append(randint(0, 99))

# Imprime tabela
print(mat)
```

Entrada de valores

- Se a matriz for quadrada, como os códigos podem ser simplificados?
- Dica:

```
NCOL = NLIN
```

Conteúdo

- O que é uma Tabela em Python?
- Como acessar elementos da Tabela?
- Entrada de valores
- **Problemas envolvendo Tabelas**

Problema 01

Determinar:

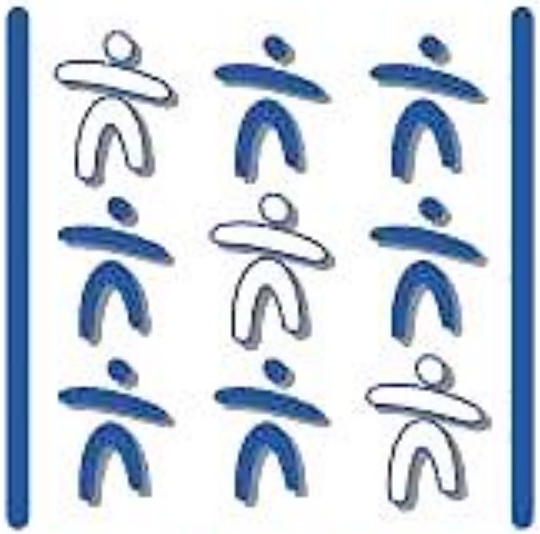
1. `M[3][0]`
2. `M[4][2]`
3. `M[1][3]`
4. `M[5][M[0][2]]`
5. `M[M[3][1]][1]`
6. `M[4][(M[1][2]+M[3][0])]`

	0	1	2	3
0	1	2	3	4
1	5	-5	3	0
2	1	1	1	1
3	-3	1	0	0
4	-4	0	1	1
5	-1	-1	-2	-2

1.	-3	4.	-2
2.	1	5.	-5
3.	0	6.	-4

Problema 02

- Escreva um script que imprima uma matriz quadrada de dimensão N contendo:
  - 1 nos elementos abaixo da diagonal principal
  - 0 na diagonal principal
  - -1 nos elementos acima da diagonal principal



Problema 02
:: Projetar algoritmo

coluna [j]

linha [i]

[0][0]	[0][1]	[0][2]	[0][3]
[1][0]	[1][1]	[1][2]	[1][3]
[2][0]	[2][1]	[2][2]	[2][3]
[3][0]	[3][1]	[3][2]	[3][3]

Diagonal principal: `i = j`

Elementos acima: `i < j`

Elementos abaixo: `i > j`

Problema 02
:: Script – Criação da matriz

```
N = int(input("Dimensao da matriz:"))

# Criacao da matriz quadrada
mat = []
for i in range(N):
    linha = [0] * N
    mat.append(linha)
```

Problema 02

:: Script – Preenchimento da matriz

```
# Preenchimento da matriz
for i in range(N):
    for j in range(N):
        # Verifica se termo estah ABAIXO
        if (i > j):
            mat[i][j] = 1
        # Verifica se termo estah ACIMA
        elif (i < j):
            mat[i][j] = -1
        # Elementos da diagonal principal
        else:
            mat[i][j] = 0
```

Problema 03

- Criar uma matriz quadrada 4×4 na qual cada elemento é a soma dos índices de sua posição na matriz.

Problema 03
:: Script

```
N = 4
mat = []
for i in range(N):
    # Anexar nova linha i
    mat.append([])
    for j in range(N):
        # Anexar elemento na linha i
        mat[i].append(i+j)
```

Problema 04
:: Matriz transposta

- Achar a transposta de uma matriz M×N fornecida.

Problema 04
:: Projetar algoritmo

Matriz **a**

[0][0]	[0][1]	[0][2]
[1][0]	[1][1]	[1][2]
[2][0]	[2][1]	[2][2]
[3][0]	[3][1]	[3][2]

`t[i][j] = a[j][i]`

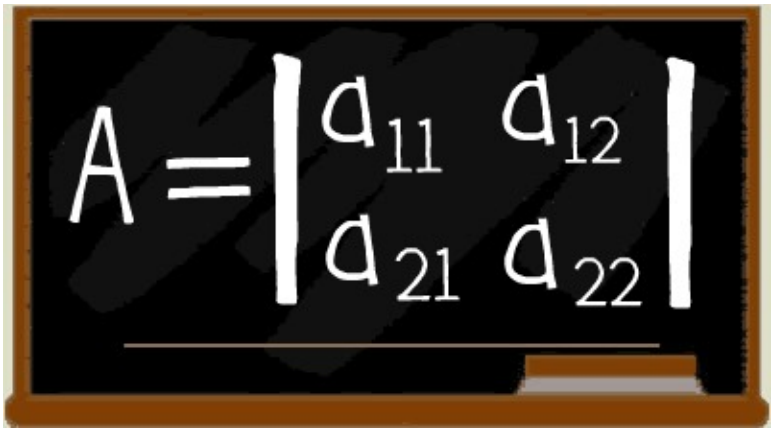
Matriz **t**

a[0][0]	a[1][0]	a[2][0]	a[3][0]
a[0][1]	a[1][1]	a[2][1]	a[3][1]
a[0][2]	a[1][2]	a[2][2]	a[3][2]

Problema 04
:: Script

```
# Constantes
NLIN = len(a)
NCOL = len(a[0])

t = []      # Matriz transposta vazia

for i in range(NCOL):
    # Anexar nova linha i
    t.append([])
    for j in range(NLIN):
        # Anexar elemento a[j][i] na linha i
        t[i].append(a[j][i])
```

Problema 05
:: Determinante



- Criar uma matriz 2×2 com números inteiros aleatórios no intervalo [-5;+5].
- Qual o determinante da matriz obtida?

Problema 05
:: Script

```
# Constantes
NLIN = 2
NCOL = 2

# Cria matriz 2x2 com valores aleatorios em [-5;5]
from random import *
mat = []
for i in range(NLIN):
    mat.append([])
    for j in range(NCOL):
        # Anexar no. aleatorio na linha i
        mat[i].append(randint(-5,5))

det = mat[0][0]*mat[1][1] - mat[0][1]*mat[1][0]
```

Problema 06

- Oito estudantes responderam a um teste de dez perguntas. As respostas são armazenadas em uma tabela. Cada linha da figura abaixo registra respostas de um aluno para as questões.

	0	1	2	3	4	5	6	7	8	9
Student 0	A	B	A	C	C	D	E	E	A	D
Student 1	D	B	A	B	C	A	E	E	A	D
Student 2	E	D	D	A	C	B	E	E	A	D
Student 3	C	B	A	E	D	C	E	E	A	D
Student 4	A	B	D	C	C	D	E	E	A	D
Student 5	B	B	E	C	C	D	E	E	A	D
Student 6	B	B	A	C	C	D	E	E	A	D
Student 7	E	B	E	C	C	D	E	E	A	D

Problema 06

- O gabarito é armazenado em uma lista de uma dimensão.
- Considerando que cada questão vale um ponto, qual a nota de cada aluno?

0	1	2	3	4	5	6	7	8	9
D	B	D	C	C	D	A	E	A	D

Problema 06
:: Projetar algoritmo

- Definir tabela de respostas (**`resp`**)
- Definir lista de gabarito (**`gab`**)
- Definir uma lista de notas (**`notas`**), com número de elementos igual à da tabela **`resp`**
- Para cada aluno (linha) na tabela **`resp`**:
  - Para cada resposta (coluna) na tabela **`resp`**:
    - Comparar resposta do aluno com gabarito
    - Se forem iguais, incrementar posição correspondente na lista **`notas`**
    - Caso contrário, não faz nada
  - Imprimir nota do aluno

Problema 06
:: Script – definições

```
# Respostas dos alunos 'as questoes
resp = [
['A', 'B', 'A', 'C', 'C', 'D', 'E', 'E', 'A', 'D'],
['D', 'B', 'A', 'B', 'C', 'A', 'E', 'E', 'A', 'D'],
['E', 'D', 'D', 'A', 'C', 'B', 'E', 'E', 'A', 'D'],
['C', 'B', 'A', 'E', 'D', 'C', 'E', 'E', 'A', 'D'],
['A', 'B', 'D', 'C', 'C', 'D', 'E', 'E', 'A', 'D'],
['B', 'B', 'E', 'C', 'C', 'D', 'E', 'E', 'A', 'D'],
['B', 'B', 'A', 'C', 'C', 'D', 'E', 'E', 'A', 'D'],
['E', 'B', 'E', 'C', 'C', 'D', 'E', 'E', 'A', 'D']]

# Gabarito das questoes
gab = ['D', 'B', 'D', 'C', 'C', 'D', 'A', 'E', 'A',
'D']
```

Problema 06
:: Script – cálculo da nota

```
# Constantes
NALUNOS = len(resp)     # No. de alunos
NQUEST = len(resp[0])   # No. de questoes

# Lista das notas de cada aluno
notas = [0] * NALUNOS

# Percorre cada linha (aluno)
for i in range(NALUNOS):
    # Pontua um aluno i, verificando cada questao
    for j in range(NQUEST):
        if (resp[i][j] == gab[j]):
            notas[i] = notas[i] + 1

    print("Nota do aluno", i, ":", notas[i])
```

Referências bibliográficas

- Menezes, Nilo Ney Coutinho (2010). Introdução à Programação com Python. Editora Novatec.

- HETLAND, Magnus Lie (2008). Beginning Python: From Novice to Professional. Springer eBooks, 2ª edição. Disponível em: http://dx.doi.org/10.1007/978-1-4302-0634-7.
- Horstmann, Cay & Necaise, Rance D. (2013). Python for Everyone. John Wiley & Sons.
- Liang, Y. D. (2013). Introduction to Programming Using Python. Pearson

Dúvidas?