

FATIAMENTO DE LISTAS

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS • MATUTINO

DEPARTAMENTO DE TECNOLOGIA DA INFORMAÇÃO

ALGORITMOS E LÓGICA DE PROGRAMAÇÃO



PROFº JOSÉ PAULO CISCATO • LUCIO NUNES DE LIRA

FATIAMENTO DE LISTAS?

Existem situações em que precisamos trabalhar com apenas uma parte de uma lista ou de uma *string*. Para isso linguagem Python tem o recurso de ***fatiamiento***.

Veremos nos próximos slides exemplos de como usá-lo.

Mostra diversos fatiamentos de uma string.

```
nome = 'Megan Fox'
```

```
print(nome)
```

```
print(nome[0:5])
```

```
print(nome[0:4])
```

```
print(nome[6:9])
```

```
print(nome[6:8])
```

```
===== RESTART:
Megan Fox
Megan
Mega
Fox
Fo
>>>
```

Saída

FATIAMENTO DE LISTAS?

- Ao digitarmos `nome[0:5]` queremos a *string* desde o 1º caractere (índice 0) até o 4º caractere (índice 4).
- Note que o intervalo é aberto à direita, ou seja, quando colocamos `[A:B]` é a partir de A até o anterior de B.
- Podemos fazer diversas combinações para conseguir diferentes resultados.

Mostra diversos fatiamentos de uma string.

```
nome = 'Megan Fox'
```

```
print(nome)
```

```
print(nome[:5])
```

```
print(nome[:4])
```

```
print(nome[6:])
```

```
print(nome[:])
```

```
===== RESTART:
Megan Fox
Megan
Mega
Fox
Megan Fox
>>>
```

Saída

FATIAMENTO DE LISTAS?

- Veja que ao omitir o índice inicial em `nome[:5]` estabelecemos que *é a partir do início*;
- Analogamente, quando omitimos o índice final em `nome[6:]` definimos que *é até o final*;
- Quando omitimos ambos os índices, como em `nome[:]`, é o mesmo que definir *é a partir do início* e *é até o final*.

FATIAMENTO DE LISTAS?

- Também podemos usar **índices negativos**, indicando posições a partir da direita, ou seja, partindo do final da *string*;
- O índice **-1** significa o último; o índice **-2** significa o penúltimo; e assim por diante;
- Veremos alguns **exemplos** no próximo slide.

Mostra diversos fatiamentos de uma string.

```
nome = 'Megan Fox'
```

```
print(nome)
```

```
print(nome[:-1])
```

```
print(nome[:-2])
```

```
print(nome[-1:])
```

```
print(nome[-2:])
```

```
===== RESTART:
Megan Fox
Megan Fo
Megan F
x
ox
>>>
```

Saída

FATIAMENTO DE LISTAS?

- Vale lembrar que tudo que fizemos até agora também é válido para **qualquer tipo de lista**, e não apenas com *strings*;
- Tudo ficou claro? Se sobrou alguma dúvida pergunte durante as aulas e também vá ao **Plantão de Dúvidas**;
- Agora é hora de **exercitar** o fatiamento de listas!

EXERCÍCIOS RESOLVIDOS

1. Leia um nome com sobrenomes e exiba-o completo, exceto pelo primeiro nome.

```
nome = input('Nome: ')
i = 0
while nome[i] != ' ':
    i += 1
print(nome[i+1:])
```

EXERCÍCIOS RESOLVIDOS

2. Dada uma lista de números naturais, terminada pelo número -1, exiba a lista dos primeiros 5 valores lidos.

```
lista = []  
n = int(input('Número: '))  
while n != -1:  
    lista.append(n)  
    n = int(input('Número: '))  
print(lista[:5])
```

EXERCÍCIOS PROPOSTOS

1. Leia um nome com sobrenomes e exiba-o completo, exceto pelo último nome. Use apenas um laço.
2. Dada uma lista de números naturais, terminada pelo número zero, exiba a lista dos últimos cinco valores lidos. Use apenas um laço, que será para a leitura dos elementos.
3. Construa uma função que receba uma frase e exiba-a a partir do segundo caractere até o penúltimo. Não use laços de repetição.

EXERCÍCIOS PROPOSTOS

4. Dada uma lista, exiba apenas a primeira metade dos elementos. Use apenas um laço, que será para a leitura dos elementos.
5. Dada uma lista, exiba apenas a segunda metade dos elementos. Use apenas um laço, que será para a leitura dos elementos.
6. Dada uma lista, inverta as duas metades. Exemplo:
`lista=[1, 2, 3, 4, 5, 6] → [4, 5, 6, 1, 2, 3]`. Use no máximo uma lista auxiliar e apenas um laço, que será para a leitura dos elementos.