

Computação I - Python

Aula 8: Estrutura de Repetição: for

Uso do for para percorrer todas as posições de dados indexáveis

Apresentado por: Rafael Machado Andrade

Produção DCC-UFRJ

Metodologia de referência <https://doi.org/10.5753/wei.2016.9683>



Estrutura de Repetição *for*

Faça uma função que receba uma lista contendo números variados e retorne uma lista contendo todos os números negativos dessa lista.

Estrutura de Repetição *for*

Faça uma função que receba uma lista contendo números variados e retorne uma lista contendo todos os números negativos dessa lista.

Versão utilizando utilizando **while**:

```
def separaNegativosWhile(lista_numeros):  
    '''Funcao que recebe uma lista contendo numeros variados e retorna  
    uma lista contendo todos os numeros negativos da lista recebida.  
    list -> list'''  
  
    negativos = []  
    contador = 0  
    while contador < len(lista_numeros):  
        if lista_numeros[contador] < 0:  
            list.append(negativos, lista_numeros[contador])  
        contador = contador + 1  
    return negativos
```

Ao utilizar o **while**, é necessário uma variável contadora para o teste booleano do **while** e referenciar cada elemento contido na lista através da própria lista, ao se especificar seu índice.

Estrutura de Repetição *for*

Faça uma função que receba uma lista contendo números variados e retorne uma lista contendo todos os números negativos dessa lista.

Versão utilizando utilizando **for**:

```
def separaNegativosFor(lista_numeros):  
    '''Funcao que recebe uma lista contendo numeros variados e retorna  
    uma lista contendo todos os numeros negativos da lista recebida.  
    list -> list'''  
  
    negativos = []  
    for numero in lista_numeros:  
        if numero < 0:  
            list.append(negativos, numero)  
    return negativos
```

Ao utilizar o **for**, não é utilizada variável contadora, e cada elemento contido na lista é referenciado diretamente pela variável criada no comando **for**.

Estrutura de Repetição *for*

Faça uma função que receba uma lista contendo números variados e retorne uma lista contendo todos os números negativos dessa lista.

Versão utilizando utilizando **while**:

```
def separaNegativosWhile(lista_numeros):  
    '''Funcao...'''  
    negativos = []  
    contador = 0  
    while contador < len(lista_numeros):  
        if lista_numeros[contador] < 0:  
            list.append(negativos, lista_numeros[contador])  
            contador = contador + 1  
    return negativos
```

Versão utilizando utilizando **for**:

```
def separaNegativosFor(lista_numeros):  
    '''Funcao...'''  
    negativos = []  
    for numero in lista_numeros:  
        if numero < 0:  
            list.append(negativos, numero)  
    return negativos
```

Estrutura de Repetição *for*

Faça uma função que receba uma lista contendo números variados e retorne uma lista contendo tuplas, onde cada tupla possui 2 valores, sendo o primeiro valor a posição de um número e o segundo o próprio número, para todos os números negativos dessa lista.

Estrutura de Repetição *for*

Faça uma função que receba uma lista contendo números variados e retorne uma lista contendo tuplas, onde cada tupla possui 2 valores, sendo o primeiro valor a posição de um número e o segundo o próprio número, para todos os números negativos dessa lista.

Versão utilizando utilizando **while**:

```
def separaNegativosWhile(lista_numeros):  
    '''Funcao...'''  
    negativos = []  
    contador = 0  
    while contador < len(lista_numeros):  
        if lista_numeros[contador] < 0:  
            list.append(negativos, (contador, lista_numeros[contador]))    #linha alterada  
            contador = contador + 1  
    return negativos
```

Ao utilizar o **while**, é necessário uma variável contadora para o teste booleano do **while** e referenciar cada elemento contido na lista através da própria lista, ao se especificar seu índice.

Estrutura de Repetição *for*

Faça uma função que receba uma lista contendo números variados e retorne uma lista contendo tuplas, onde cada tupla possui 2 valores, sendo o primeiro valor a posição de um número e o segundo o próprio número, para todos os números negativos dessa lista.

Versão utilizando utilizando **for**:

```
def separaNegativosFor2(lista_numeros):  
    '''Funcao...'''  
    negativos = []  
    posicao = 0  
    for numero in lista_numeros:  
        if numero < 0:  
            list.append(negativos, (posicao, numero))  
            posicao = posicao + 1  
    return negativos
```

#linha acrescentada

#linha alterada

#linha acrescentada

Ao utilizar o **for**, não é utilizada variável contadora, e cada elemento contido na lista é referenciado diretamente pela variável criada no comando **for**.

Estrutura de Repetição *for*

Faça uma função que receba uma lista contendo números variados e retorne uma lista contendo tuplas, onde cada tupla possui 2 valores, sendo o primeiro valor a posição de um número e o segundo o próprio número, para todos os números negativos dessa lista.

Versão utilizando utilizando **while**:

```
def separaNegativosWhile(lista_numeros):  
    '''Funcao... '''  
    negativos = []  
    contador = 0  
    while contador < len(lista_numeros):  
        if lista_numeros[contador] < 0:  
            list.append(negativos, (contador, lista_numeros[contador]))    #linha alterada  
        contador = contador + 1  
    return negativos
```

Versão utilizando utilizando **for**:

```
def separaNegativosFor2(lista_numeros):  
    '''Funcao... '''  
    negativos = []  
    posicao = 0    #linha acrescentada  
    for numero in lista_numeros:  
        if numero < 0:  
            list.append(negativos, (posicao, numero))    #linha alterada  
            posicao = posicao + 1    #linha acrescentada  
    return negativos
```

Autores

- **João C. P. da Silva** ▶ Lattes
- **Carla Delgado** ▶ Lattes
- **Ana Luisa Duboc** ▶ Lattes

Colaboradores

- **Anamaria Martins Moreira** ▶ Lattes
- **Fabio Mascarenhas** ▶ Lattes
- **Leonardo de Oliveira Carvalho** ▶ Lattes
- **Charles Figueiredo de Barros** ▶ Lattes
- **Fabício Firmino de Faria** ▶ Lattes

Computação I - Python

Aula 8: Estrutura de Repetição: for

Uso do for para percorrer todas as posições de dados indexáveis

Apresentado por: Rafael Machado Andrade

Produção DCC-UFRJ

Metodologia de referência <https://doi.org/10.5753/wei.2016.9683>

