

+ Código

+ Texto

```
def inverter_posicoes(numero):
```

```
    """
```

Esta função inverte os valores de número inteiro no espaço delimitado pelos separadores unidades do sistema decimal. Por exemplo, se for enviado o número:

```
4, a função retorna 4
34, a função retorna 43
234, a função retorna 432
4.567, a função retorna 4.765
34.567, a função retorna 43.765
234.567, a função retorna 432.765
123.456.789, a função retorna 321.654.987
```

Estrutura da função:

- 1 - Trata o número recebido, de modo que strings ou inteiros possam ser enviados, mas somente é feita inversão na parte inteira do número;
- 2 - Quebra o número em partes para efetuar inversão;
- 3 - Efetua a inversão conforme lógica acima;

Lógica da função:

- i - Ler um número ABC.DEF.GHI qualquer
- ii - Transformar este número em uma lista de strings ['A','B','C','D','E','F','G']
- iii - Inverter a lista ['I','H','G','F','E','D','C','B','A']
- iv - Converter em subset de 3 em 3 [['I','H','G'],['F','E','D'],['C','B','A']]
- v - Inverter novamente [['C','B','A'], ['F','E','D'], ['I','H','G']]
- vi - Converter em lista simples ['C','B','A', 'F','E','D', 'I','H','G']
- vii - Gerar o número de saída CBA.FED.IHG

Input: numero (int) -> número inteiro em que se deseja realizar a operação

Output: numero invertido (str) -> string com o número invertido com pontuação

```
    """
```

```
# 1 - Tratar o número, adicionando separadores de unidades para impressão
# e garantir que possa ser enviado float, int ou str
numero_formatado = formatar_numero(numero)
```

```
# 2 - Quebrar o número em uma lista separada de 3 em 3, pois temos sempre
#três casas decimais no sistema de numeração decimal
numero_quebrado = quebrar_numero(numero_formatado)
```

```
# 3 - Inverter a lista separada em 3 em 3 e gerar número
numero_invertido = inverter_lista(numero_quebrado)
```

```
return numero_invertido
```

```
def formatar_numero(numero):
```

```
    """ Esta função recebe um int, float ou str e retorna um inteiro
    contendo pontos como separador de unidades. Por exemplo, se enviarmos
    '1234', 1234.00, '1234.00' ou 1234 a função retorna 1.234
```

Input: numero (int,str) numero que irá receber formatação

Return: int formatado conforme descrito acima

```
    """
```

```
    if isinstance(numero) == int:
```

```

1. type(numero) == int:
    #números inteiros permanecem como estão
    numero_formato_int = numero
elif type(numero) == str or type(numero) == float:
    #esta conversão garante que strings de floats serão convertidas em , '123456.00' v
    numero_formato_int = int(float(numero))
else:
    #se tiver um formato diferente, retorna um erro
    return ValueError("Operação não permitida neste formato de dado.")
#retorna o número_formato_int 123456789 em 123.456.789
return format(numero_formato_int, 'd').replace(",", ".")

def quebrar_numero(numero):
    """Esta função quebra um número fornecido em uma lista de strings onde cada
    componenente é um item da string.

    Input:  numero (str) a ser transformado em uma lista de strings e invertido
    Return: lista_em_trios, uma lista de strings com ordem inversa ao número fornecido
    """

    #transforma 123456 em ['1', '2', '3', '4', '5', '6']
    lista_numero = [i for i in numero]
    #transforma ['1', '2', '3', '4', '5', '6'] em ['6', '5', '4', '3', '2', '1']
    lista_numero_invertida = list(reversed(lista_numero))
    lista_em_trios = list(separar_em_trios(lista_numero_invertida, 3))

    return lista_em_trios

def separar_em_trios(lista, n):
    """Esta função recebe uma lista e quebra a mesma em subsets de tamanho n
    Inputs: lista (list) a ser quebrada em subsets
           n      (int) tamanho do subset
    Return: um gerador, deve ser convertido em list para ser usado
    """

    #remove os pontos de separação que possam haver na string
    while '.' in lista: lista.remove('.')
    # transforma ['6', '5', '4', '3', '2', '1'] em [['6', '5', '4'], ['3', '2', '1']]
    for i in range(0, len(lista), n):
        yield (lista[i:i + n])

def inverter_lista(lista):
    """Esta função recebe uma lista de listas do número a ser invertido
    Input:  lista (list) no formato [['6', '5', '4'], ['3', '2', '1']]
    Return: numero_invertido (str) '654.321'
    """

    #converte [['6', '5', '4'], ['3', '2', '1']] em [['3', '2', '1'], ['6', '5', '4']]
    lista_invertida = list(reversed(lista))
    #cria uma string vazia para contarmos o tamanho da nossa lista
    numero_invertido = ""
    #transforma lista de listas em lista simples
    #[['3', '2', '1'], ['6', '5', '4']] vira ['3', '2', '1', '6', '5', '4']
    lista_simples = sum(lista_invertida, [])
    #percorre a lista simples já na ordem correta da saída e monta uma string numero_inver
    for i in lista_simples:
        numero_invertido += str(i)
    #formata o número invertido para ficar com as seprações e facilitar leitura
    numero_invertido = formatar_numero(numero_invertido)

```

```

        numero_invertido = numero_invertido * 10 + int(numero % 10)
        numero = numero // 10
    return numero_invertido

def testar():
    """Esta função executa vários testes automatizados para confirmar funcionamento
    da função
    """
    assert inverter_posicoes(234567) == '432.765'
    assert inverter_posicoes(1) == '1'
    assert inverter_posicoes(12.0) == '21'
    assert inverter_posicoes(123.00) == '321'
    assert inverter_posicoes('1234') == '1.432'
    assert inverter_posicoes(12345) == '21.543'
    assert inverter_posicoes('123456.00') == '321.654'
    assert inverter_posicoes('1234567') == '1.432.765'
    assert inverter_posicoes(12345678.00) == '21.543.876'
    assert inverter_posicoes('123456789.0') == '321.654.987'
    assert inverter_posicoes(1234567890) == '1.432.765.098'
    print('Testes ok!')

def mostrar_testes():
    """Esta função retorna a impressão dos números enviados a função e a saída da função,
    para melhor visualização da operação da função
    """
    for i in [234567, 1, 12.0, 123.00, '1234', 12345, 123456, '1234567', 12345678.00, 123456789.0, '1234567890']:
        print(f'Número inserido: {formatar_numero(i)}')
        print(f"Número invertido: {inverter_posicoes(i)}")
        print('-----')

testar()
mostrar_testes()

```

```

Testes ok!
Número inserido: 234.567
Número invertido: 432.765
-----
Número inserido: 1
Número invertido: 1
-----
Número inserido: 12
Número invertido: 21
-----
Número inserido: 123
Número invertido: 321
-----
Número inserido: 1.234
Número invertido: 1.432
-----
Número inserido: 12.345
Número invertido: 21.543
-----
Número inserido: 123.456
Número invertido: 321.654
-----
Número inserido: 1.234.567
Número invertido: 1.432.765
-----
Número inserido: 12.345.678

```

Número invertido: 21.543.876

Número inserido: 123.456.789

Número invertido: 321.654.987

Número inserido: 1.234.567.890

Número invertido: 1.432.765.098

✓ 0s conclusão: 17:26

