TÉCNICO LISBOA

Fundamentos de Programação - 2018/2019 Aula Prática 05 (30 minutos) Turno 6ª feira 15:30-17:00
Nome:
Número:
Data:
Curso:

Capítulo 5 - Listas

Escreva uma função em Python com o nome `duplica_elementos` que recebe uma lista e devolve a lista obtida da lista original em que todos os elementos são duplicados. Não necessita verificar a validade dos argumentos. Por exemplo,

```
>>> duplica_elementos(['a', ['b', 'c'], 5])
['a', 'a', ['b', 'c'], ['b', 'c'], 5, 5]
>>> duplica_elementos([])
[]
```

Solução 1:

```
def duplica_elementos(lst):
    res = []
    for e in lst:
        res = res + [e, e]
    return res
```

Solução 2:

```
def duplica_elementos(lst):
   return [e for e in lst for n in range(2)]
```

TÉCNICO LISBOA

Fundamentos de Programação - 2018/2019 Aula Prática 05 (30 minutos) Turno 2ª feira 10:30-12:00
Nome:
Número:
Data:
Curso:

Capítulo 5 - Listas

Escreva uma função em Python com o nome `remove_repetidos` que recebe uma lista e devolve a lista obtida da lista original em que todos os elementos repetidos foram removidos. A sua função deve utilizar o operador `del`. Não necessita verificar a validade dos argumentos. Por exemplo,

```
>>> remove_repetidos([2, 4, 3, 2, 2, 2, 3])
[2, 4, 3]
>>> remove_repetidos([2, 5, 7])
[2, 5, 7]
```

Solução 1:

```
def remove_repetidos(l):
    for i in range(len(l)-1, 1, -1):
        if l[i] in l[0:i-1]:
            del(l[i])
    return l
```

Solução 2 (sem utilizar o operador del):

```
def remove_repetidos(lista):
    newlista = []
    for e in lista:
        if e not in newlista:
            newlista += [e]
    return newlista
```

TÉCNICO LISBOA

Fundamentos de Programação - 2018/2019 Aula Prática 05 (30 minutos) Turno 5ª feira 10:30-12:00
Nome:
Número:
Data:
Curso:

Capítulo 5 - Listas

Escreva uma função chamada `substitui` que recebe uma lista `lst`, e dois valores `velho` e `novo`, e que devolve a lista que resulta de substituir em `lst` todas as ocorrências de `velho` por `novo`. Não necessita verificar a validade dos argumentos. Por exemplo,

```
>>> substitui([1, 2, 3, 2, 4], 2, 'a')
[1, 'a', 3, 'a', 4]
```

Solução 1:

```
def substitui_nd(lst, velho, novo):
    # versão não destrutiva
    res = []
    for e in lst:
        if e == velho:
            res = res + [novo]
        else:
            res = res + [e]
    return res
```

Solução 2:

```
def substitui_d(lst, velho, novo):
    # versão destrutiva
    for i in range(len(lst)):
        if lst[i] == velho:
            lst[i] = novo
    return lst
```

Solução 3:

```
def substitui(lst, velho, novo):
   return [e if e != velho else novo for e in lst]
```

TÉCNICO LISBOA

Fundamentos de Programação - 2018/2019 Aula Prática 05 (30 minutos) Turno 4ª feira 08:00-09:30
Nome:
Número:
Data:
Curso:

Capítulo 5 - Listas

Escreva uma função chamada `posicoes_lista` que recebe uma lista e um elemento, e devolve uma lista contendo todas as posições em que o elemento ocorre na lista. Por exemplo,

```
>>> posicoes_lista(['a', 2, 'b', 'a'], 'a')
[0, 3]
```

Solução 1:

```
def posicoes_lista(lst, el):
    res = []
    for i in range(len(lst)):
        if lst[i] == el:
            res = res + [i]
    return res
```

Solução 2:

```
def posicoes_lista(lst, el):
    return [i for i in range(len(lst) if lst[i] == el]
```

TÉCNICO LISBOA

Fundamentos de Programação - 2018/2019 Aula Prática 05 (30 minutos) Turno 6ª feira 14:00-15:30
Nome:
Número:
Data:
Curso:

Capítulo 5 - Listas

Escreva uma função chamada `crescimento` que recebe uma lista de inteiros, e devolve a lista obtida da lista original em que todos os inteiros são multiplicados pela posição que ocupam na lista. Não necessita verificar a validade dos argumentos. Por exemplo,

```
>>> crescimento([3, -4, -3, 1, 7])
[0, -4, -6, 3, 28]
```

Solução 1:

```
def crescimento(lista):
    res = []
    for i in range(len(lista)):
        res += [i*lista[i]]
    return res
```

Solução 2:

```
def crescimento(lista):
    return [lista[i]*i for i in range(len(lista))]
```

TÉCNICO LISBOA

Fundamentos de Programação - 2018/2019 Aula Prática 05 (30 minutos) Turno 6ª feira 12:30-14:00
Nome:
Número:
Data:
Curso:

Capítulo 5 - Listas

Escreva uma função chamada `retifica` que recebe uma lista de inteiros, e devolve a lista obtida da lista original em que todos os inteiros negativos foram substituídos pelo índice corresponente à sua posição na lista. Não necessita verificar a validade dos argumentos. Por exemplo,

```
>>> retifica([3, -4, -3, 1, 7])
[3, 1, 2, 1, 7]
```

Solução 1:

```
def retifica(lista):
    res = []
    for i in range(len(lista)):
        if lista[i] < 0:
            res += [i]
        else:
            res += [lista[i]]
    return res
```

Solução 2:

```
def retifica(lst):
   return [e if (e > 0) else i for i,e in enumerate(lst)]
```

TÉCNICO LISBOA

Fundamentos de Programação - 2018/2019 Aula Prática 05 (30 minutos) Turno 2ª feira 09:00-10:30
Nome:
Número:
Data:
Curso:

Capítulo 5 - Listas

Escreva uma função chamada `multiplica_vs` que recebe uma lista de inteiros (`vetor`) e um inteiro (`escalar`), e devolve uma lista contendo o produto do `vetor` pelo `escalar`. Não necessita verificar a validade dos argumentos. Por exemplo,

```
>>> multiplica_vs([3, 4, 0, 1, 7], 2)
[6, 8, 0, 2, 14]
```

Solução 1:

```
def multiplica_vs(vetor, escalar):
    res = []
    for e in vetor:
        res += [e*escalar]
    return res
```

Solução 2:

```
def multiplica_vs(lst, s):
    return [s*e for e in lst]
```