

# Untitled

November 21, 2022

## 0.1 Questão 01: Refatore o código utilizando compreensão de lista

```
[4]: import random as r
    lista = []
    for item in range(10):
        lista.append(r.randrange(1,100))
    print(lista)
```

[4, 40, 50, 68, 28, 75, 73, 17, 55, 17]

## 0.2 Questão 02: Refatore o código utilizando compreensão de lista

```
[9]: lista = [4, 40, 50, 68, 28, 75, 73, 17, 55, 17]
    nova_lista = []
    for itens in zip(lista, lista[1:], lista[2:]):
        x = itens[0]
        for item in itens:
            if item < x:
                x = item
        nova_lista.append(x)
    print(nova_lista)
```

[4, 40, 28, 28, 28, 17, 17, 17]

## 0.3 Questão 03: Refatore o código utilizando map, filter, reduce. Analise quais dessas funções são necessárias e as utilize

```
[184]: lista = [4, 40, 50, 68, 28, 75, 73, 17, 55, 17]
    nova_lista = []
    for itens in zip(lista, lista[1:], lista[2:]):
        x = itens[0]
        for item in itens:
            if item < x:
                x = item
        nova_lista.append(x)
    print(nova_lista)
```

[4, 40, 28, 28, 28, 17, 17, 17]

#### 0.4 Questão 04: Refatore o código utilizando compreensão de lista

```
[185]: import re
lista = ['QWT997z', 'PoP', 'Raquete-X67', 'XYZw&', 'F99-bx', 'Hall']

for item in lista.copy():
    if re.fullmatch("[A-Z].*[a-z]", item) == None:
        lista.remove(item)

print(lista)
```

```
['QWT997z', 'F99-bx', 'Hall']
```

#### 0.5 Questão 05: Refatore o código utilizando map, filter, reduce. Analise quais dessas funções são necessárias e as utilize

```
[186]: import re
lista = ['QWT997z', 'PoP', 'Raquete-X67', 'XYZw&', 'F99-bx', 'Hall']

for item in lista.copy():
    if re.fullmatch("[A-Z].*[a-z]", item) == None:
        lista.remove(item)

print(lista)
```

```
['QWT997z', 'F99-bx', 'Hall']
```

#### 0.6 Questão 06: Refatore o código utilizando compreensão de lista

```
[59]: import re
lista = ['QWT997z', 'PoP', 'Raquete-X67', 'Eu Quero Passar', 'Robo F99-bx', 'Hall']
nova_lista = []
for item in lista:
    itens = re.findall("[A-Z][a-z][a-z]+", item)
    nova_lista.extend(itens)

print(nova_lista)
```

```
['Raquete', 'Quero', 'Passar', 'Robo', 'Hall']
```

#### 0.7 Questão 07: Refatore o código utilizando map, filter, reduce. Analise quais dessas funções são necessárias e as utilize

```
[ ]: import re
lista = ['QWT997z', 'PoP', 'Raquete-X67', 'Eu Quero Passar', 'Robo F99-bx', 'Hall']
nova_lista = []
```

```

for item in lista:
    itens = re.findall("[A-Z][a-z][a-z]+", item)
    nova_lista.extend(itens)

print(nova_lista)

```

**0.8 Questão 08:** Refatore o código utilizando map, filter, reduce. Analise quais dessas funções são necessárias e as utilize

```

[61]: lista = [
    {
        'produto': 'A',
        'quantidade': 10
    },
    {
        'produto': 'B',
        'quantidade': 2
    },
    {
        'produto': 'C',
        'quantidade': 30
    },
    {
        'produto': 'D',
        'quantidade': 17
    },
]

qtd = 0
for item in lista:
    qtd += item['quantidade']

qtd /= len(lista)
print(qtd)

```

14.75

**0.9 Questão 09:** Refatore o código utilizando map, filter, reduce. Analise quais dessas funções são necessárias e as utilize

```

[69]: lista = [4, 40, 50, 68, 28, 75, 73, 17, 55, 17]
mm = max(lista) / min(lista)
qtd = 0
for item in lista:
    if item > mm:
        qtd += 1
print(qtd)

```

### 0.10 Questão 10: Refatore o código utilizando compreensão de lista

```
[187]: lista = [
    {
        'produto': 'A',
        'quantidade': 10
    },
    {
        'produto': 'B',
        'quantidade': 2
    },
    {
        'produto': 'C',
        'quantidade': 30
    },
    {
        'produto': 'D',
        'quantidade': 17
    },
]

for idx, item in enumerate(lista.copy()):
    if item['quantidade'] > 10:
        lista.remove(item)
    else:
        lista[idx] = item['produto']

print(lista)
```

```
['A', 'B']
```

### 0.11 Questão 11: Refatore o código utilizando map, filter, reduce. Analise quais dessas funções são necessárias e as utilize

```
[73]: lista = [
    {
        'produto': 'A',
        'quantidade': 10
    },
    {
        'produto': 'B',
        'quantidade': 2
    },
    {
```

```

        'produto': 'C',
        'quantidade': 30
    },
    {
        'produto': 'D',
        'quantidade': 17
    },
]

for idx, item in enumerate(lista.copy()):
    if item['quantidade'] > 10:
        lista.remove(item)
    else:
        lista[idx] = item['produto']

print(lista)

```

['A', 'B']

## 0.12 Questão 12: Refatore o código utilizando compreensão de lista

```

[188]: lista = [4, 40, 50, 68, 28, 75, 73, 17, 55, 17]
nova_lista = []
for a, b in zip(lista, lista[1:]):
    m = 1
    for q in range(2, max(a,b)):
        if a%q == b%q == 0:
            m = q
            break
    if m == 1:
        nova_lista.append(a)

print(nova_lista)

```

[28, 75, 73, 17, 55]

## 0.13 Questão 13: Refatore o código utilizando map, filter, reduce. Analise quais dessas funções são necessárias e as utilize

```

[189]: lista = [4, 40, 50, 68, 28, 75, 73, 17, 55, 17]
nova_lista = []
for a, b in zip(lista, lista[1:]):
    m = 1
    for q in range(2, max(a,b)):
        if a%q == b%q == 0:
            m = q

```

```

        break
    if m == 1:
        nova_lista.append(a)

print(nova_lista)

```

[28, 75, 73, 17, 55]

#### 0.14 Questão 14: Refatore o código utilizando compreensão de lista

```

[190]: tupla = ('A', 'A', 'B', 'C', 'C', 'C', 'D', 'D', 'C', 'A', 'E')
        frequencia = {}
        for item in tupla:
            frequencia[item] = frequencia.get(item, 0) + 1

        print(frequencia)

```

{'A': 3, 'B': 1, 'C': 4, 'D': 2, 'E': 1}

#### 0.15 Questão 15: Refatore o código utilizando map, filter, reduce. Analise quais dessas funções são necessárias e as utilize

```

[191]: tupla = ('A', 'A', 'B', 'C', 'C', 'C', 'D', 'D', 'C', 'A', 'E')
        frequencia = {}
        for item in tupla:
            frequencia[item] = frequencia.get(item, 0) + 1

        print(frequencia)

```

{'A': 3, 'B': 1, 'C': 4, 'D': 2, 'E': 1}

#### 0.16 Questão 16: Refatore o código utilizando map, filter, reduce. Analise quais dessas funções são necessárias e as utilize

```

[78]: lista = [4, 40, 50, 68, 28, 75, 73, 17, 55, 17]
        r = True
        for item in lista:
            r &= item % 2 == 0

        print(r)

```

False

### 0.17 Questão 17: Refatore o código utilizando compreensão de lista

```
[192]: lista = [
    {
        'produto': 'A',
        'quantidade': 10,
        'valor': 5.67
    },
    {
        'produto': 'B',
        'quantidade': 2
    },
    {
        'produto': 'C',
        'quantidade': 30
    },
    {
        'produto': 'D',
        'quantidade': 17
    },
    {
        'nome': 'José',
        'email': 'jose@email.com'
    },
]

for item in lista.copy():
    if {'quantidade', 'produto'} != item.keys():
        lista.remove(item)

print(*lista, sep='\n')
```

```
{'produto': 'B', 'quantidade': 2}
{'produto': 'C', 'quantidade': 30}
{'produto': 'D', 'quantidade': 17}
```

### 0.18 Questão 18: Refatore o código utilizando map, filter, reduce. Analise quais dessas funções são necessárias e as utilize

```
[103]: lista = [
    {
        'produto': 'A',
        'quantidade': 10,
        'valor': 5.67
    },
    {
        'produto': 'B',
```

```

        'quantidade': 2
    },
    {
        'produto': 'C',
        'quantidade': 30
    },
    {
        'produto': 'D',
        'quantidade': 17
    },
    {
        'nome': 'José',
        'email': 'jose@email.com'
    },
]

```

```

for item in lista.copy():
    if {'quantidade', 'produto'} != item.keys():
        lista.remove(item)

print(*lista, sep='\n')

```

```

{'produto': 'B', 'quantidade': 2}
{'produto': 'C', 'quantidade': 30}
{'produto': 'D', 'quantidade': 17}

```

## 0.19 Questão 19: Refatore o código utilizando compreensão de lista

```

[148]: import random as r
saida = True
while type(saida) is bool:
    lista = []
    for item in range(5):
        lista.append(r.randrange(1,20))

    check = True
    for x, y in zip(lista[1:], lista):
        check &= x > y

    saida = lista if check else check

print(saida)

```

```
[1, 8, 11, 12, 17]
```



**0.20 Questão 20:** Refatore o código utilizando map, filter, reduce. Analise quais dessas funções são necessárias e as utilize

```
[ ]: import random as r
saida = True
while type(saida) is bool:
    lista = []
    for item in range(5):
        lista.append(r.randrange(1,20))

    check = True
    for x, y in zip(lista[1:], lista):
        check &= x > y

    saida = lista if check else check

print(saida)
```

**0.21 Questão 21:** Refatore o código utilizando recursividade, para eliminar o while.

```
[ ]: import random as r
saida = True
while type(saida) is bool:
    lista = []
    for item in range(5):
        lista.append(r.randrange(1,20))

    check = True
    for x, y in zip(lista[1:], lista):
        check &= x > y

    saida = lista if check else check

print(saida)
```

**0.22 Questão 22:** Refatore o código utilizando recursividade, para eliminar o while.

```
[183]: r = 1
k = 1
while k <= 10:
    k += 1
    r += k

print(r)
```

**0.23 Questão 23: Refatore o código utilizando compreensão de lista**

```
[193]: lista = [13, 2, 4, 5, 4, 7, 5]

for idx in range(len(lista)-1):
    if lista[idx] > lista[idx+1]:
        lista[idx], lista[idx+1] = lista[idx+1], lista[idx]

print(lista)
```

[2, 4, 5, 4, 7, 5, 13]

**0.24 Questão 24: Refatore o código utilizando map, filter, reduce. Analise quais dessas funções são necessárias e as utilize**

```
[194]: lista = [13, 2, 4, 5, 4, 7, 5]

for idx in range(len(lista)-1):
    if lista[idx] > lista[idx+1]:
        lista[idx], lista[idx+1] = lista[idx+1], lista[idx]

print(lista)
```

[2, 4, 5, 4, 7, 5, 13]

**0.25 Questão 25: Refatore o código utilizando recursividade, para eliminar o while.**

```
[178]: lista = [13, 2, 4, 5, 4, 7, 5]

for idx in range(len(lista)-1):
    if lista[idx] > lista[idx+1]:
        lista[idx], lista[idx+1] = lista[idx+1], lista[idx]

print(lista)
```

[2, 4, 5, 4, 7, 5, 13]