

18 Reunião 18: 29/OUT/2020



Figure 1: Fonte: Doodle do dia das criança 2020

18.1 Reuniões passadas

18.1.1 Comando for ... in range()

```
var = inicio
while var < fim:
    |
    | executa bloco com var = início, início+passo
    | início+2*passo, ..., ... até fim (EXCLUSIVE)
    |
    |
var += passo
```

tem o mesmo efeito que

```
for var in range(início, fim, passo):
    |
    | executa bloco com var = início, início+passo
    | início+2*passo, ..., ... até fim (EXCLUSIVE)
    |
```

```
for i in range(2, 10, 1): # (ini, fim, passo)
    print(i, end="")
```

Imprime: 2 3 4 5 6 7 8 9

```
for j in range(3, 9, 1): # (ini, fim, passo)
    print(i, end="")
```

Imprime: 3 4 5 6 7 8

```
for i in range(9, 2, -1): # (ini, fim, passo)
    print(i)
```

Imprime ?????

```
for i in range(-1, 7, 4): # (ini, fim, passo)
    print(i, end="")
```

Verem {0, 1, ..., 15} fim 16

Imprime: ?????

Se não especificar, `início` = 0

Se não especificar, `passo` = 1

```
for i in range(10): # (fim)
    print(i, end="")
```

Imprime: 0 1 2 3 4 5 6 7 8 9

```
for i in range(2, 10): # (ini, fim)
    print(i, end="")
```

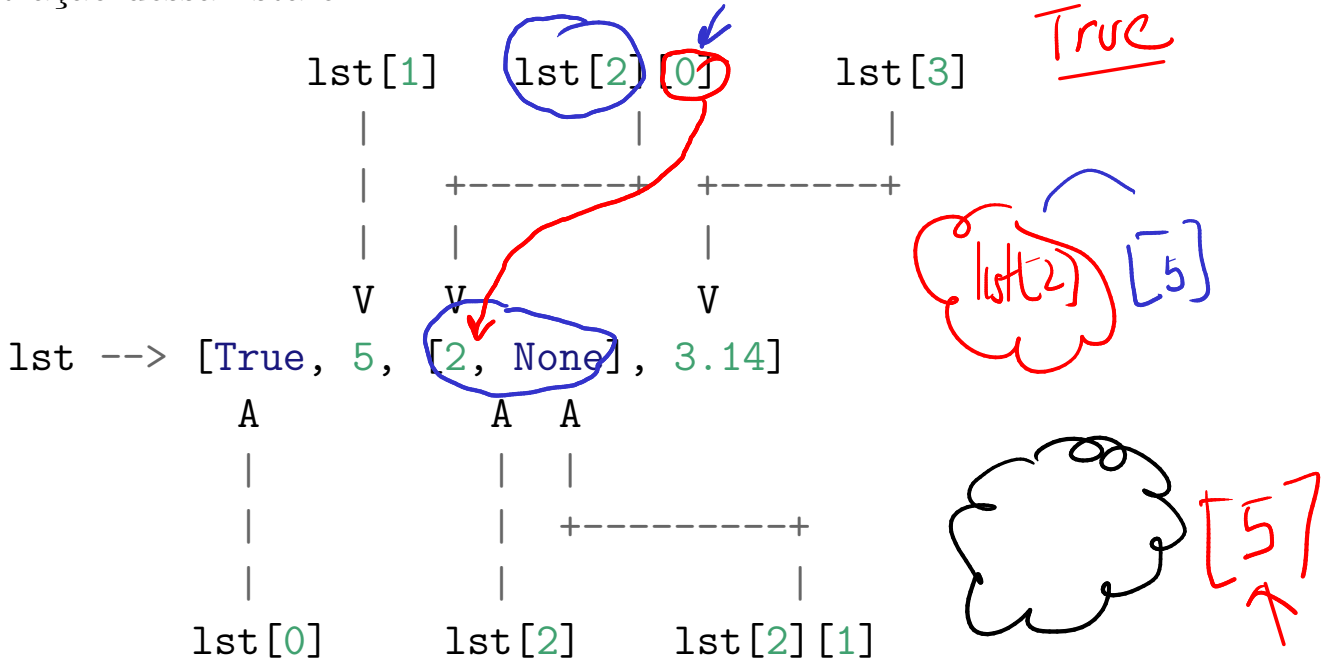
Imprime: 2 3 4 5 6 7 8 9

18.1.2 Listas

Considere a lista

`lst = [True, 5, [2, None], 3.14]`

ilustração dessa lista é



```
len(lista) = 4
```

```
len(lista[2]) = 2
```

```
# percorre do início para o fim
```

```
n = len(lst)
```

```
for i in range(0, len(lst), 1):
```

```
    print(i, ":", lst[i])
```

```
# percorre do fim para o início
```

```
for i in range(0, len(lst), 1):
```

```
    print(i, ":", lst[i])
```

```
# criar uma lista
lst = []
for i in range(6):
    lst = lst + [i] # concatenação
```

```
# criar uma lista
uns = 5 * [1]
```

```
# criar uma lista
primos = [2, 3, 5, 7, 11, 13, 17]
```

18.2 Hoje

- mais funções com listas
- operador `in`
- apelidos versus clones
- operador `is`

18.3 Exercício: filtro embaralhado

Dados $n > 0$ e uma sequência com de n números inteiros, imprimir os números que ocorrem na sequência eliminando as repetições.

18.3.1 Exemplos

Digite n: 10
 Digite o 1o. número: 1
 Digite o 2o. número: 2
 Digite o 3o. número: 3
 Digite o 4o. número: 3
 Digite o 5o. número: 2
 Digite o 6o. número: 1
 Digite o 7o. número: 4
 Digite o 8o. número: 1234
 Digite o 9o. número: 234
 Digite o 10o. número: 1000000
 Lista de números sem repetições:
 1 2 3 4 1234 234 1000000

Consertem o programa

- Tabulada
- ordem



Lista

vá para o link no e-mail ou no chat

not a char e

o mesmo que a char == Fase

Digite n: 10
 Digite o 1o. número: 123
 Digite o 2o. número: 12
 Digite o 3o. número: -6
 Digite o 4o. número: 7
 Digite o 5o. número: 29
 Digite o 6o. número: 49
 Digite o 7o. número: 128
 Digite o 8o. número: 256
 Digite o 9o. número: 512
 Digite o 10o. número: 29
 Lista de números sem repetições:
 123 12 -6 7 29 49 128 256 512

18.3.2 Solução embaralhada

A seguir está uma solução para este exercício. Infelizmente este programa caiu no chão e as tabulações foram perdidas e os comandos foram embaralhados. A sua tarefa será consertar o programa colocando os comandos na ordem certa e inserindo a tabulação.

```
def main():
    '''
    Programa que lê um inteiro  $n > 0$  e uma sequência de
     $n$  números inteiros e imprime a sequência eliminando
    as repetições.
    '''
    num = int(input(f"Digite o {i+1}o. número: "))
    achou = pertence(numeros, num)

    # inicialmente a lista de números esta vazia
    numeros = []

    numeros += [num]

    #-----
main()

# leia cada número da sequencias e coloque na lista `numeros`
# apenas se ele nao é repitado
for i in range(0, n, 1): # range(ini, fim, passo)
    if not achou:

        if item == lst[j]:
            achou = True
            j = j + 1

# imprima lista sem repetições
```



```

print("Lista de números sem repetições: ")

#-----
def pertence(lst, item):
    '''(list, int) -> bool

    Recebe uma lista `lst` de itens e um `item` e
    retorna True se o item é um elemento de lst
    '''
    achou = False
    j = 0
    while j < len(lst) and not achou:
        return achou

    for i in range(0, len(numeros), 1): # range(ini, fim, passo)
        print(f"{numeros[i]}", end=" ")
        print("\n") # muda de linha

    # leia a quantidade de número na sequência
    n = int(input("Digite n: "))

```

18.3.3 Solução

```
def main():  
    '''  
    Programa que lê um inteiro  $n > 0$  e uma sequência de  
     $n$  números inteiros e imprime a sequência eliminando  
    as repetições.  
    '''  
    # leia a quantidade de número na sequência  
    n = int(input("Digite n: "))  
  
    # inicialmente a lista de números esta vazia  
    numeros = []  
  
    # leia cada número da sequencias e coloque na lista `numeros`  
    # apenas se ele nao é repitado  
    for i in range(0, n, 1): # range(ini, fim, passo)  
        num = int(input(f"Digite o {i+1}o. número: "))  
        achou = pertence(numeros, num)  
        if not achou:  
            numeros += [num]  
  
    # imprima lista sem repetições  
    print("Lista de números sem repetições: ")  
    for i in range(0, len(numeros), 1): # range(ini, fim, passo)  
        print(f"{numeros[i]}", end=" ")  
    print("\n") # muda de linha  
  
#-----  
def pertence(lst, item):  
    '''(list, int) -> bool  
  
    Recebe uma lista `lst` de itens e um `item` e
```

```
    returna True se o item é um elemento de lst
    '''
    achou = False
    j = 0
    while j < len(lst) and not achou:
        if item == lst[j]:
            achou = True
            j = j + 1
    return achou
```

```
#-----
main()
```

18.3.4 Operador `in` list

O operador `in` pode ser usado para verificar se um item esta na lista

```
item in lst
```

é `True` se `item` é um elemento da lista `lst` e `False` em caso contrário. Por exemplo

```
5 in [2, 3, 5, 7, 11] == True
True in [True] == True
True in [False] == False
True in [1, 2, 5, None] == False
None in [1, 2, 5, None] == True
```

18.3.5 Solução mais Pythoniana

```
def main():  
    '''  
    Programa que lê um inteiro  $n > 0$  e uma sequência de  
     $n$  números inteiros e imprime a sequência eliminando  
    as repetições.  
    '''  
    # leia a quantidade de número na sequência  
    n = int(input("Digite n: "))  
  
    # inicialmente a lista de números esta vazia  
    numeros = []  
  
    # leia cada número da sequencias e coloque na lista `numeros`  
    # apenas se ele nao é repitado  
    for i in range(0, n, 1): # range(ini, fim, passo)  
        num = int(input(f"Digite o {i+1}o. número: "))  
        # achou = pertence(numeros, num)  
        # if not achou:  
        if num not in numeros: # VEJA AQUI  
            numeros += [num]  
  
    # imprima lista sem repetições  
    print("Lista de números sem repetições: ")  
    for i in range(0, len(numeros), 1): # range(ini, fim, passo)  
        print(f"{numeros[i]}", end=" ")  
    print("\n") # muda de linha
```

18.4 Exercício: filtro *de luxe*

Dados $n > 0$ e uma sequência com de n números inteiros, imprimir os números que ocorrem na sequência e junto com o número de vezes que cada um ocorre.

18.4.1 Exemplo

Digite o tamanho da sequência: 10

Digite o 1o. número: 1

Digite o 2o. número: -2

Digite o 3o. número: 123

Digite o 4o. número: 1

Digite o 5o. número: -2

Digite o 6o. número: 123

Digite o 7o. número: -2

Digite o 8o. número: 0

Digite o 9o. número: 1

Digite o 10o. número: 4

índice!

	1	π
	-2	π
→	123	π
	0	1
	4	1

valor		frequência
1	0	3
-2	1	3
123	2	2
0	3	1
4	4	1

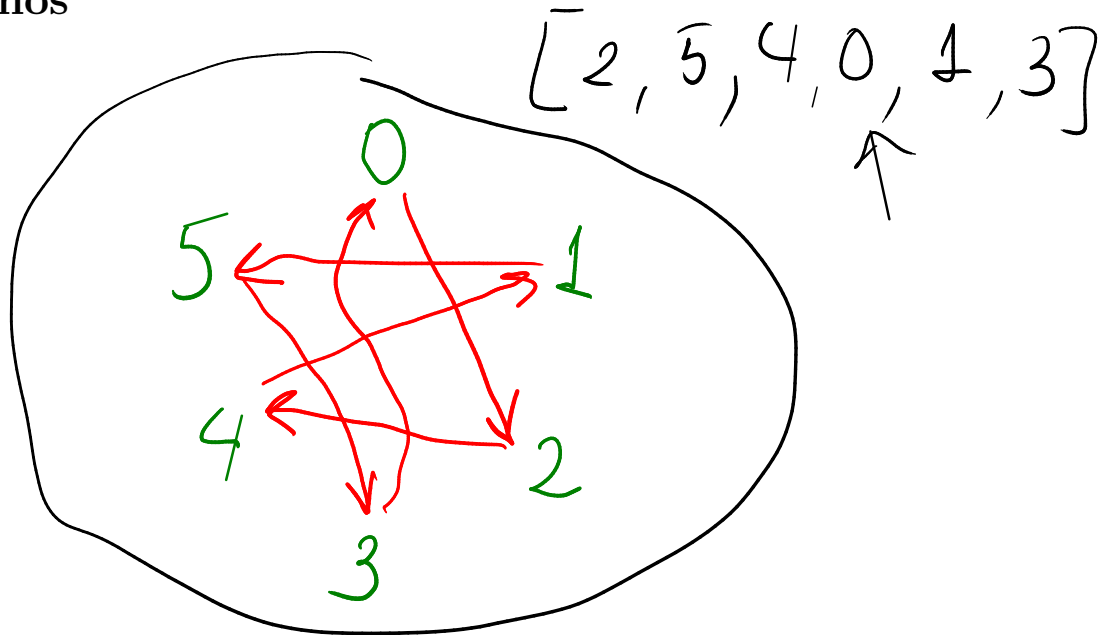
$\text{valor}[i] \rightarrow \text{freq}[i]$

5 7 0 1 1 2 6 9 4

↙ ↙ ↙ ↙ ↙
↑ ↑ ↑ ↑ ↑
↑

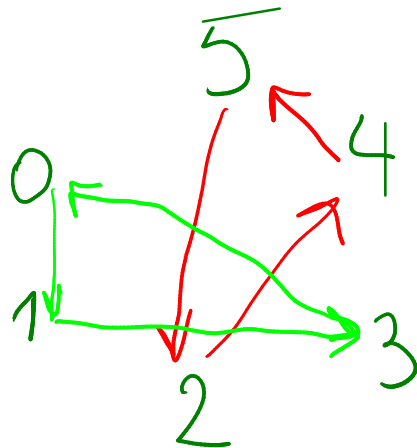
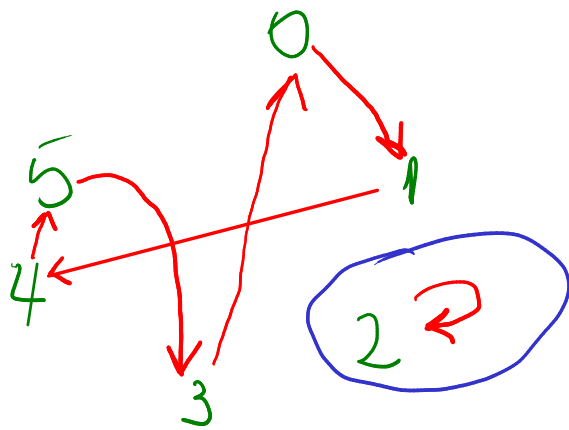
random.seed(27)

18.4.2 Rascunhos



amig-de = $[1, 4, 2, 0, 5, 3]$

0 1 2 3 4 5



$[1, 3, 4, 0, 5, 2]$

0 1 2 ③ 4 5

18.4.3 Solução

```
def main():
    '''
    Dados n e uma sequencia com n números inteiros,
    conta e imprime o número de vezes que cada
    número ocorre na sequência.
    '''
    n = int(input("Digite o tamanho da sequência: "))
    numeros = []
    frequencia = []
    for i in range(n):
        num = int(input("Digite um número: "))
        j = indice(num, numeros)
        if j == None:
            numeros += [num]
            frequencia += [1]
        else:
            frequencia[j] += 1

    print("      valor frequência")
    m = len(numeros)
    for i in range(m):
        print(f"{numeros[i]:8} {frequencia[i]:6}")

#-----
def indice(item, lista):
    '''(objeto, lista) -> int ou None
    RECEBE um objeto 'item' e uma lista 'lista'.
    RETORNA o indice da posição em que item ocorre na lista.
    Caso item não ocorra na lista a funcao RETORNA None
    '''
    ind = None
```



```

# procure item na lista
n = len(lista)
for i in range(n):
    if item == lista[i]:
        ind = i
return ind

```

#-----

```

# inicio do programa
main()

```

