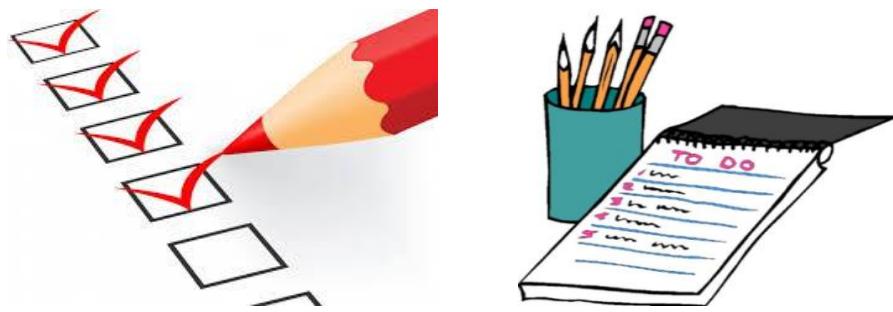


Listas



Função imprimeLista

✓ Exemplo

```
# Função
def imprimeLista (lista):
    for elemento in lista:
        print (elemento)
    return

# Teste da função
lteste = [1, [17, 22], 24]
imprimeLista (lteste)
```

1
[17, 22]
24

Função imprimeLista

- ✓ O que fazer para imprimir os elementos de sublistas em linhas separadas também?

- ✓ necessário identificar se o elemento é uma sublista

 type (elemento)

- ✓ se for, chamar recursivamente a função

 imprimeLista (elemento)

3

Função imprimeLista

- ✓ Exemplo

```
# Função
def imprimeLista (lista):
    for elemento in lista:
        if type (elemento) == list:
            imprimeLista (elemento)
        else:
            print (elemento)
    return
```

 == int
== float
== str
== bool

 1
17
22
24

4

Simulação parcial da execução

```
def imprimeLista (lista):
    for elemento in lista:
        if type(elemento) == list:
            imprimeLista (elemento)
        else:
            print (elemento)
    return
```

Variáveis

lista	elemento
[1, [17, 22], 24]	1

Impressão

1

5

Simulação parcial da execução

```
def imprimeLista (lista):
    for elemento in lista:
        if type(elemento) == list:
            imprimeLista (elemento)
        else:
            print (elemento)
    return
```

Variáveis

lista	elemento
[1, [17, 22], 24]	[17, 22]

Impressão

1

6

Simulação parcial da execução

```
def imprimeLista (lista):
    for elemento in lista:
        if type(elemento) == list:
            imprimeLista (elemento)
        else:
            print (elemento)
    return
```

Variáveis

lista	elemento
[1, [17, 22], 24]	[17, 22]

Impressão

1

7

Simulação parcial da execução

```
def imprimeLista (lista):
    for elemento in lista:
        if type(elemento) == list:
            imprimeLista (elemento)
        else:
            print (elemento)
    return
```

2ª instância

Variáveis

lista	elemento
[17, 22]	

Impressão

1

8

Simulação parcial da execução

```
def imprimeLista (lista):
    for elemento in lista:
        if type(elemento) == list:
            imprimeLista (elemento)
        else:
            print (elemento)
    return
```

Variáveis

lista	elemento
[17, 22]	

Impressão

1

9

Simulação parcial da execução

```
def imprimeLista (lista):
    for elemento in lista:
        if type(elemento) == list:
            imprimeLista (elemento)
        else:
            print (elemento)
    return
```

Variáveis

lista	elemento
[17, 22]	17

Impressão

1

10

Simulação parcial da execução

```
def imprimeLista (lista):
    for elemento in lista:
        if type(elemento) == list:
            imprimeLista (elemento)
        else:
            print (elemento)
    return
```

Variáveis

lista	elemento
[17, 22]	17

Impressão

1

11

Simulação parcial da execução

```
def imprimeLista (lista):
    for elemento in lista:
        if type(elemento) == list:
            imprimeLista (elemento)
        else:
            print (elemento)
    return
```

Variáveis

lista	elemento
[17, 22]	17

Impressão

1
17

12

Simulação parcial da execução

```
def imprimeLista (lista):
    for elemento in lista:
        if type(elemento) == list:
            imprimeLista (elemento)
        else:
            print (elemento)
    return
```

Variáveis

lista	elemento
[17, 22]	22

Impressão

1
17

13

Simulação parcial da execução

```
def imprimeLista (lista):
    for elemento in lista:
        if type(elemento) == list:
            imprimeLista (elemento)
        else:
            print (elemento)
    return
```

Variáveis

lista	elemento
[17, 22]	22

Impressão

1
17

14

Simulação parcial da execução

```
def imprimeLista (lista):
    for elemento in lista:
        if type(elemento) == list:
            imprimeLista (elemento)
        else:
            print (elemento)
    return
```

Variáveis

lista	elemento
[17, 22]	22

Impressão

1
17
22

15

Simulação parcial da execução

```
def imprimeLista (lista):
    for elemento in lista:
        if type(elemento) == list:
            imprimeLista (elemento)
        else:
            print (elemento)
    return
```

A blue arrow originates from the word "return" and points to the text "encerra a 2ª instância" located above the second "return" keyword.

Variáveis

lista	elemento
[17, 22]	22

Impressão

1
17
22

16

Simulação parcial da execução

```
def imprimeLista (lista):
    for elemento in lista:
        if type(elemento) == list:
            imprimeLista (elemento)
        else:
            print (elemento)
    return
```

retorna para a 1ª instância

próxima iteração: impressão do último elemento

Variáveis		Impressão
lista	elemento	
[1, [17, 22], 24]	[17, 22]	1 17 22

17

Simulação parcial da execução

```
def imprimeLista (lista):
    for elemento in lista:
        if type(elemento) == list:
            imprimeLista (elemento)
        else:
            print (elemento)
    return
```

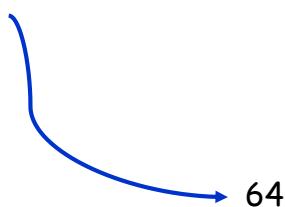
Variáveis		Impressão
lista	elemento	
[1, [17, 22], 24]	24	1 17 22 24

18

Exercício

- ✓ Escrever uma função que receba uma lista já preenchida com valores inteiros e retorne a soma de seus elementos
- ✓ Exemplo

$I = [1, 17, 22, 24]$



19

Algortimo

```
soma = 0
para cada elemento da lista
    soma = soma + elemento
retornar soma
```

20



```
# Função
def somaLista (lista):
    soma = 0
    for elemento in lista:
        soma += elemento
    return soma

# Teste da função
l = [1, 17, 22, 24]
print ('Soma: %d' %somaLista(l))
```



```
# Função
def somaLista (lista):
    soma = 0
    for elemento in lista:
        soma += elemento
    return soma

# Teste da função
l = [1, [17, 22], 24]
print ('Soma: %d' %somaLista(l))

TypeError: unsupported operand
type(s) for +=: 'int' and 'list'
```

```
# Função
def somaLista (lista):
    soma = 0
    for elemento in lista:
        if type(elemento) == list:
            soma += somaLista (elemento)
        else:
            soma += elemento
    return soma

# Teste da função
l = [1, [17, 22], 24]
print ('Soma: %d' %somaLista(l))
```

23

```
def somaLista (lista):
    soma = 0
    for elemento in lista:
        if type(elemento) == list:
            soma += somaLista (elemento)
        else:
            soma += elemento
    return soma
```

Variáveis

lista	soma	elemento
[1, [17, 22], 24]	0	1

24

Simulação parcial da execução

```
def somaLista (lista):
    soma = 0
    for elemento in lista:
        if type(elemento) == list:
            soma += somaLista (elemento)
        else:
            soma += elemento
    return soma
```

Variáveis

lista	soma	elemento
[1, [17, 22], 24]	0	1

25

Simulação parcial da execução

```
def somaLista (lista):
    soma = 0
    for elemento in lista:
        if type(elemento) == list:
            soma += somaLista (elemento)
        else:
            soma += elemento
    return soma
```

Variáveis

lista	soma	elemento
[1, [17, 22], 24]	1	1

26

Simulação parcial da execução

```
def somaLista (lista):
    soma = 0
    for elemento in lista:
        if type(elemento) == list:
            soma += somaLista (elemento)
        else:
            soma += elemento
    return soma
```

Variáveis

lista	soma	elemento
[1, [17, 22], 24]	1	[17, 22]

27

Simulação parcial da execução

```
def somaLista (lista):
    soma = 0
    for elemento in lista:
        if type(elemento) == list:
            soma += somaLista (elemento)
        else:
            soma += elemento
    return soma
```

Variáveis

lista	soma	elemento
[1, [17, 22], 24]	1	[17, 22]

28

Simulação parcial da execução

```
def somaLista (lista):
    soma = 0
    for elemento in lista:
        if type(elemento) == list:
            soma += somaLista (elemento)
        else:
            soma += elemento
    return soma
```

Variáveis

lista	soma	elemento
[17, 22]		

29

Simulação parcial da execução

```
def somaLista (lista):
    soma = 0
    for elemento in lista:
        if type(elemento) == list:
            soma += somaLista (elemento)
        else:
            soma += elemento
    return soma
```

Variáveis

lista	soma	elemento
[17, 22]		

30

Simulação parcial da execução

```
def somaLista (lista):
    soma = 0
    for elemento in lista:
        if type(elemento) == list:
            soma += somaLista (elemento)
        else:
            soma += elemento
    return soma
```

Variáveis

lista	soma	elemento
[17, 22]	0	

31

Simulação parcial da execução

```
def somaLista (lista):
    soma = 0
    for elemento in lista:
        if type(elemento) == list:
            soma += somaLista (elemento)
        else:
            soma += elemento
    return soma
```

Variáveis

lista	soma	elemento
[17, 22]	0	17

32

Simulação parcial da execução

```
def somaLista (lista):
    soma = 0
    for elemento in lista:
        if type(elemento) == list:
            soma += somaLista (elemento)
        else:
            soma += elemento
    return soma
```

Variáveis

lista	soma	elemento
[17, 22]	0	17

33

Simulação parcial da execução

```
def somaLista (lista):
    soma = 0
    for elemento in lista:
        if type(elemento) == list:
            soma += somaLista (elemento)
        else:
            soma += elemento
    return soma
```

Variáveis

lista	soma	elemento
[17, 22]	17	17

34

Simulação parcial da execução

```
def somaLista (lista):
    soma = 0
    for elemento in lista:
        if type(elemento) == list:
            soma += somaLista (elemento)
        else:
            soma += elemento
    return soma
```

Variáveis

lista	soma	elemento
[17, 22]	17	22

35

Simulação parcial da execução

```
def somaLista (lista):
    soma = 0
    for elemento in lista:
        if type(elemento) == list:
            soma += somaLista (elemento)
        else:
            soma += elemento
    return soma
```

Variáveis

lista	soma	elemento
[17, 22]	17	22

36



Simulação parcial da execução

```
def somaLista (lista):
    soma = 0
    for elemento in lista:
        if type(elemento) == list:
            soma += somaLista (elemento)
        else:
            soma += elemento
    return soma
```

Variáveis

lista	soma	elemento
[17, 22]	39	22

37



Simulação parcial da execução

```
def somaLista (lista):
    soma = 0
    for elemento in lista:
        if type(elemento) == list:
            soma += somaLista (elemento)
        else:
            soma += elemento
    return soma
```

A blue arrow points from the 'return soma' line to the right, with the text 'retorna 39 e encerra a 2ª instância' written next to it.

Variáveis

lista	soma	elemento
[17, 22]	39	22

38

Simulação parcial da execução

```
def somaLista (lista):
    soma = 0
    for elemento in lista:
        if type(elemento) == list:
            soma += somaLista (elemento)
        else:
            soma += elemento
    return soma
```

retorna para
a 1^a instância
39

Variáveis		
lista	soma	elemento
[1, [17, 22], 24]	1	[17, 22]

39

Simulação parcial da execução

```
def somaLista (lista):
    soma = 0
    for elemento in lista:
        if type(elemento) == list:
            soma += somaLista (elemento)
        else:
            soma += elemento
    return soma
```

Variáveis		
lista	soma	elemento
[1, [17, 22], 24]	40	[17, 22]

40

Simulação parcial da execução

```
def somaLista (lista):
    soma = 0
    for elemento in lista:
        if type(elemento) == list:
            soma += somaLista (elemento)
        else:
            soma += elemento
    return soma
```

Variáveis

lista	soma	elemento
[1, [17, 22], 24]	40	24

41

Simulação parcial da execução

```
def somaLista (lista):
    soma = 0
    for elemento in lista:
        if type(elemento) == list:
            soma += somaLista (elemento)
        else:
            soma += elemento
    return soma
```

Variáveis

lista	soma	elemento
[1, [17, 22], 24]	40	24

42



Simulação parcial da execução

```
def somaLista (lista):
    soma = 0
    for elemento in lista:
        if type(elemento) == list:
            soma += somaLista (elemento)
        else:
            soma += elemento
    return soma
```

Variáveis

lista	soma	elemento
[1, [17, 22], 24]	64	24

43



Simulação parcial da execução

```
def somaLista (lista):
    soma = 0
    for elemento in lista:
        if type(elemento) == list:
            soma += somaLista (elemento)
        else:
            soma += elemento
    return soma
```

A blue arrow points from the 'return soma' line to the text 'retorna 64 e encerra a função'.

retorna 64
e encerra a
função

Variáveis

lista	soma	elemento
[1, [17, 22], 24]	64	24

44