

INSTITUTO  
FEDERAL  
Pará



Ricardo José Cabeça de Souza

[ricardo.souza@ifpa.com.br](mailto:ricardo.souza@ifpa.com.br)

- **FUNÇÕES E PROCEDIMENTOS**

- **FUNÇÕES**

- Também conhecidas como sub-rotinas, e **retorna um valor** quando são “chamadas”

- **PROCEDIMENTO**

- Diferem das funções apenas por **não retornarem resultado (valor)**

# • FUNÇÕES E PROCEDIMENTOS EM PYTHON

- Tudo é considerado função
- FORMATO:

**def NOME\_FUNÇÃO ( [parâmetros] ):**  
**Comandos**

## FORMATO:

1. Uma linha de cabeçalho começando com uma palavra reservada (**def**)
2. A palavra reservada **def** é seguida pelo **nome da função** e alguns *parâmetros* entre parênteses. A lista de parâmetros pode ser vazia ou conter qualquer número de parâmetros separados por vírgulas e terminando com dois pontos.
3. Um **corpo** consistindo de um ou mais comandos, cada um deles com a mesma tabulação – 4 espaços é o padrão em Python – em relação à linha de cabeçalho.

# • FUNÇÕES EM PYTHON

- Sabemos como usar várias funções, como len, int, float, print e input
- Podemos definir nossas próprias funções em Python
- Para definir uma nova função, utilizaremos a instrução **def**

- **FUNÇÕES/PROCEDIMENTO EM PYTHON**
  - Exibir um nome na tela

```
func.py - C:\Users\rrjjcss\AppData\Local\Programs\...  
File Edit Format Run Options Window Help  
#Função  
def exibir_fun():  
    nome=input("Informe seu nome:")  
    return nome  
print("Função: Boa noite %s" % (exibir_fun()))|  
Ln: 5 Col: 46
```

```
proc.py - C:\Users\rrjjcss\AppData\Local\Programs\Python\...  
File Edit Format Run Options Window Help  
#Procedimento  
def exibir_proc():  
    nome=input("Informe seu nome:")  
    print("Boa noite %s" % (nome))  
exibir_proc()  
Ln: 1 Col: 13
```

# • FUNÇÕES EM PYTHON

```
def soma(a,b): ❶
```

```
    print(a+b) ❷
```

```
soma(2,9) ❸
```

❶ A instrução **def** seguida pelo **nome da função**, no caso, **soma**

Após o nome e entre parênteses, especificamos o nome dos **parâmetros** que a função receberá (**a** e **b**)

Usamos : (dois pontos) após os parâmetros para indicar o início de um bloco

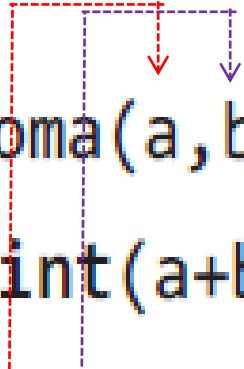
❷ Usamos a função **print** para exibir **a+b**

❸ Ocorre a **chamada** da função **soma**

Os **parâmetros** são substituídos na mesma ordem em que foram definidos

# • FUNÇÕES EM PYTHON

```
def soma(a,b):  
    print(a+b)  
soma(2,9)
```



❶ A instrução **def** seguida pelo **nome da função**, no caso, **soma**

Após o nome e entre parênteses, especificamos o nome dos **parâmetros** que a função receberá (**a** e **b**)

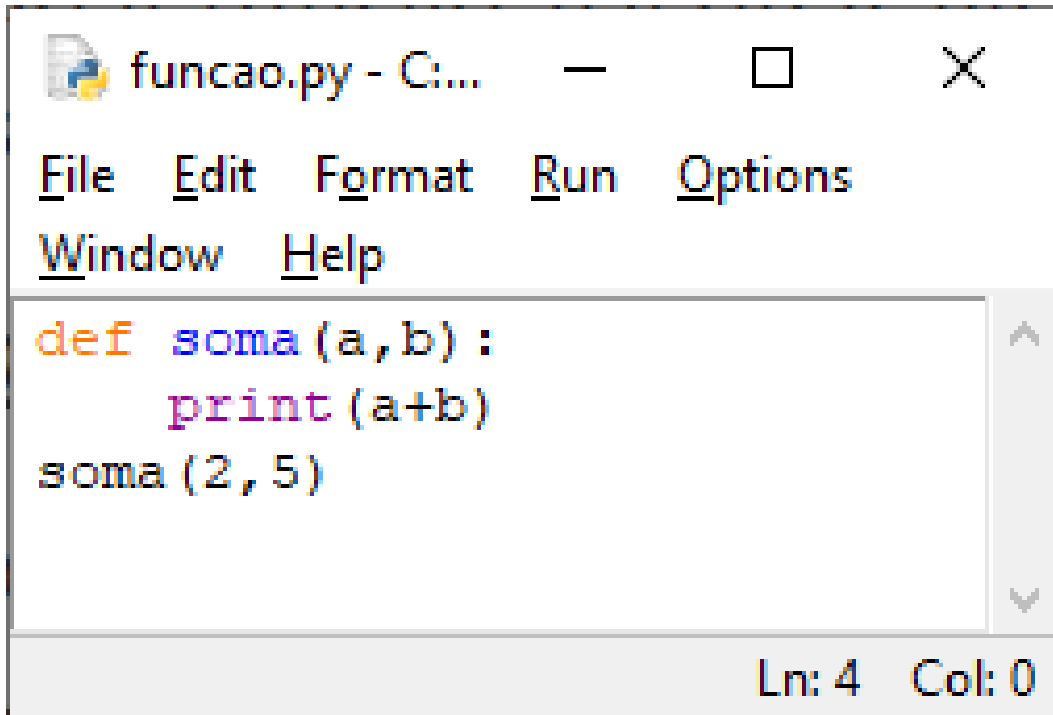
Usamos : (dois pontos) após os parâmetros para indicar o início de um bloco

❷ Usamos a função **print** para exibir **a+b**

❸ Ocorre a **chamada** da função **soma**

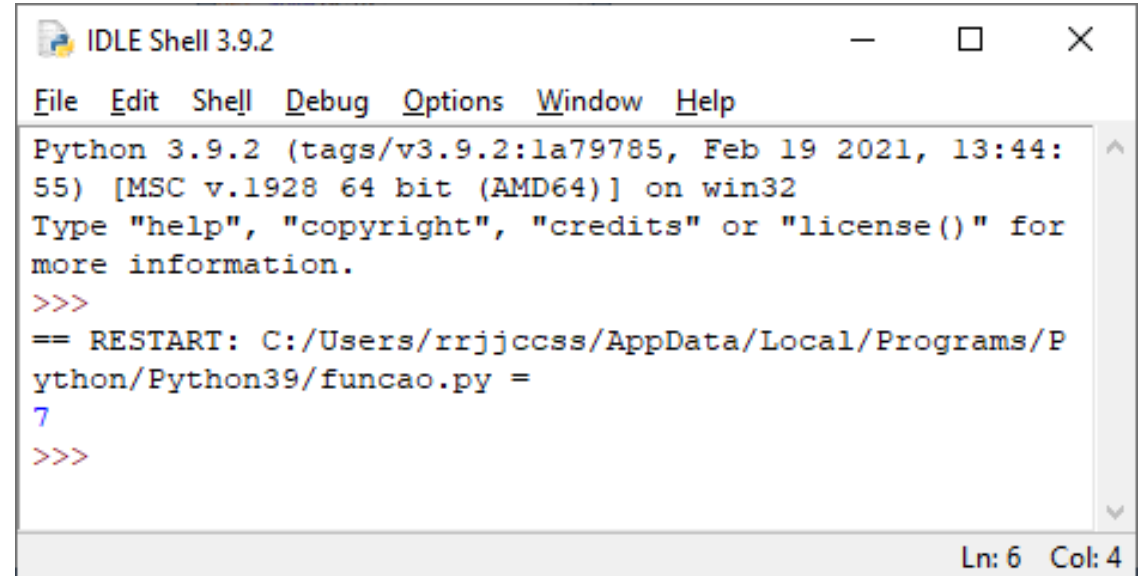
Os **parâmetros** são substituídos na mesma ordem em que foram definidos

# • FUNÇÕES EM PYTHON



```
def soma(a,b):  
    print(a+b)  
soma(2,5)
```

Ln: 4 Col: 0



```
IDLE Shell 3.9.2  
File Edit Shell Debug Options Window Help  
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
== RESTART: C:/Users/rrjjccss/AppData/Local/Programs/Python/Python39/funcao.py =  
7  
>>>
```

Ln: 6 Col: 4





```
def soma(a,b): ❶  
    print(a+b) ❷  
soma(2,9) ❸
```

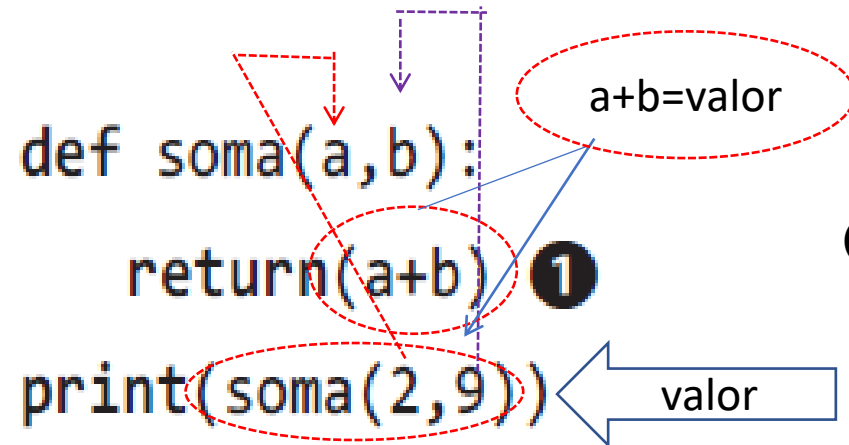
## • FUNÇÕES EM PYTHON

- Funções são especialmente interessantes para isolar uma tarefa específica em um trecho de programa
- Permite que a solução de um problema seja reutilizada em outras partes do programa, sem precisar repetir as mesmas linhas
- O exemplo acima utiliza **dois parâmetros e imprime sua soma**
- Essa função **não retorna** valores

# • FUNÇÕES EM PYTHON

- Definição do **retorno** de um **valor**

```
def soma(a,b):  
    return(a+b)  
print(soma(2,9))
```



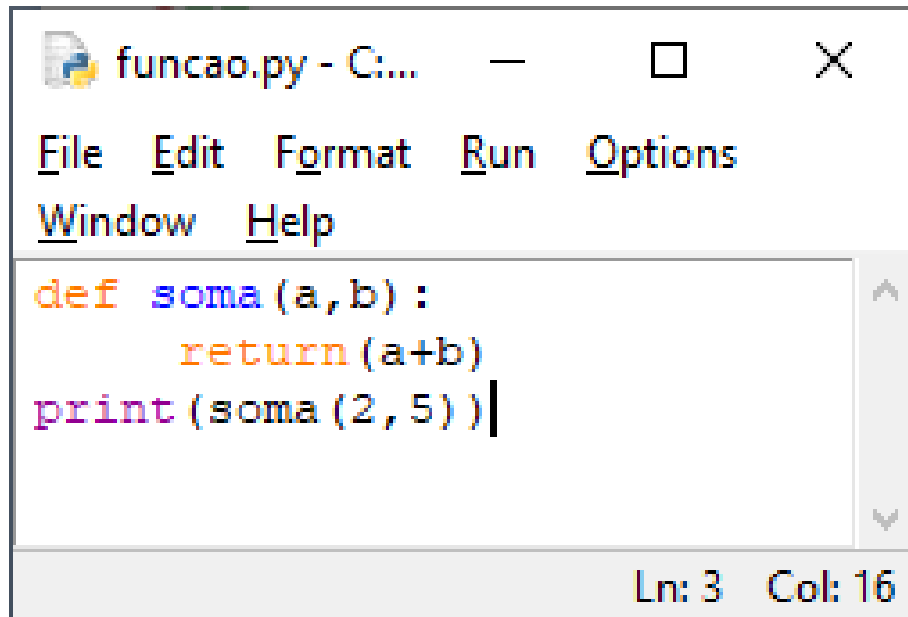
a+b=valor

1

valor

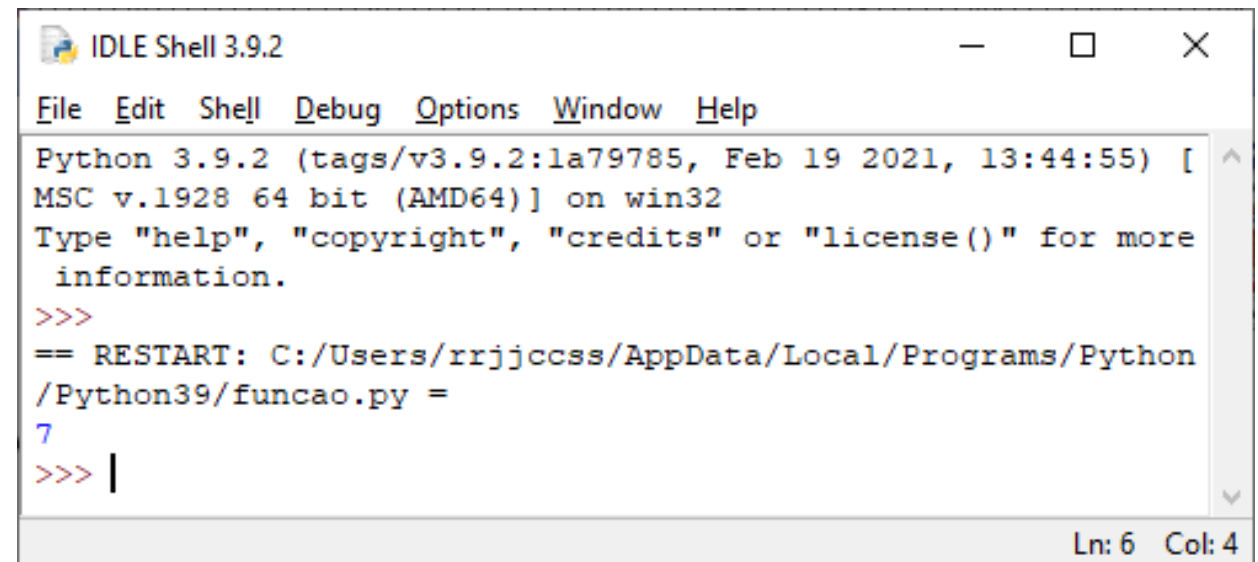
❶ A instrução **return** indica o **valor** a retornar

# • FUNÇÕES EM PYTHON



```
def soma(a,b):  
    return(a+b)  
print(soma(2,5))
```

Ln: 3 Col: 16

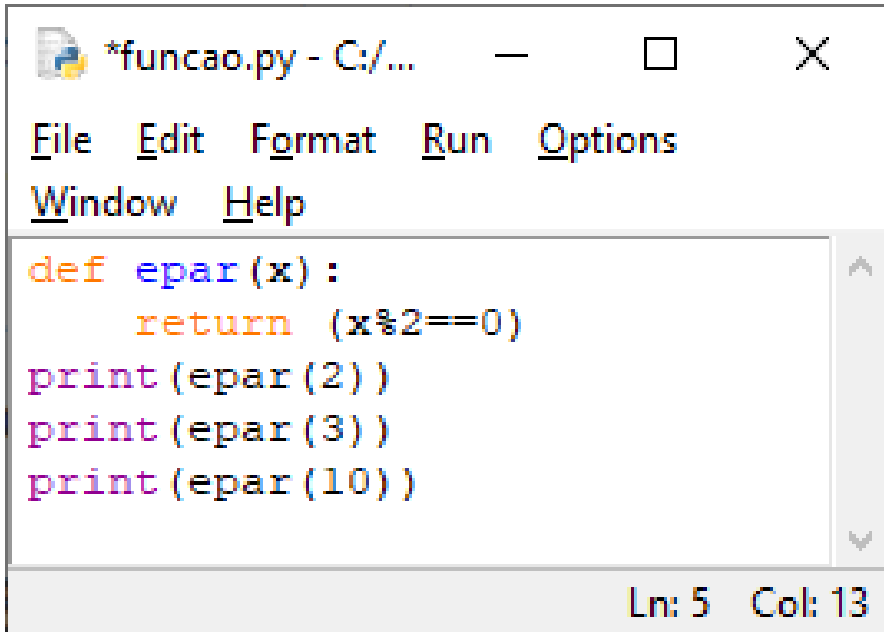


```
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
== RESTART: C:/Users/rrjjccss/AppData/Local/Programs/Python/Python39/funcao.py =  
7  
>>> |
```

Ln: 6 Col: 4

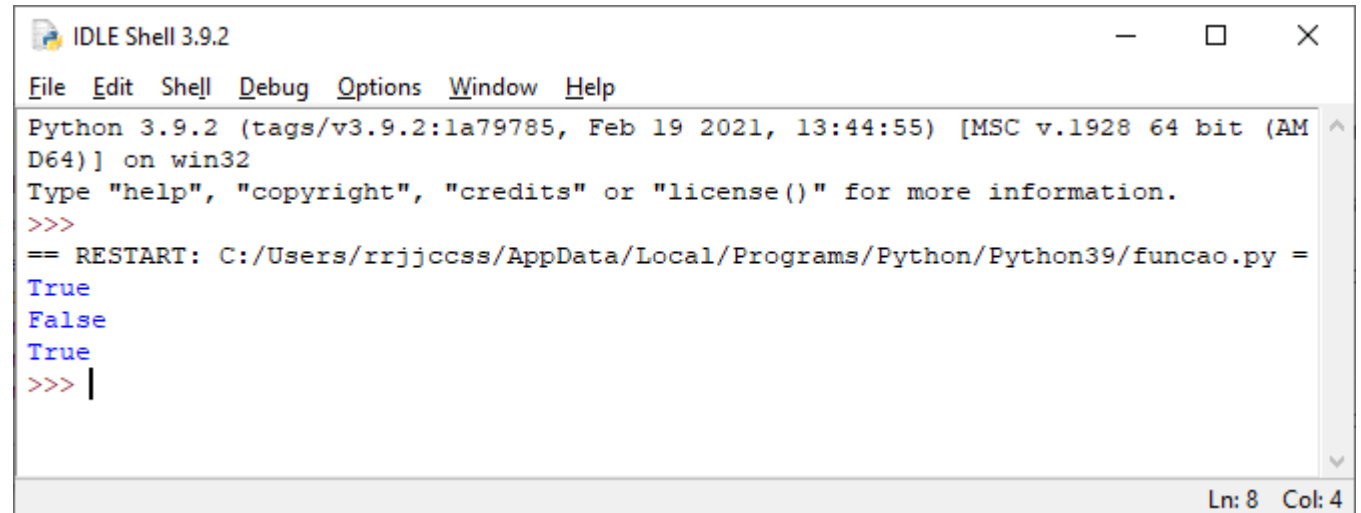
# • FUNÇÕES EM PYTHON

- Função que retorne verdadeiro ou falso



```
def epar(x):  
    return (x%2==0)  
print(epar(2))  
print(epar(3))  
print(epar(10))
```

Ln: 5 Col: 13



```
IDLE Shell 3.9.2  
File Edit Shell Debug Options Window Help  
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
== RESTART: C:/Users/rrjjccss/AppData/Local/Programs/Python/Python39/funcao.py =  
True  
False  
True  
>>> |
```

Ln: 8 Col: 4

# • FUNÇÕES EM PYTHON

- Definir uma função para retornar a palavra **par** ou **ímpar**

```
*funcao.py - C:\Users\rrjjccss\AppData\Loc...
File Edit Format Run Options Window Help

def epar(x):
    return (x%2==0)
def par_ou_impar(x):
    if epar(x):
        return "par"
    else:
        return "ímpar"
print(par_ou_impar(4))
print(par_ou_impar(5))

Ln: 9 Col: 22
```

```
IDLE Shell 3.9.2
File Edit Shell Debug Options Window Help

Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:\Users\rrjjccss\AppData\Local\Programs\Python\Python39\funcao.py =
par
ímpar
>>> |

Ln: 7 Col: 4
```

# • FUNÇÕES EM PYTHON

- Pesquisa em uma lista

```
def pesquise(lista, valor):  
    for x,e in enumerate(lista):  
        if e == valor:  
            return x ❶  
    return None ❷  
  
L=[10, 20, 25, 30]  
print(pesquise(L, 25))  
print(pesquise(L, 27))
```

A função **pesquise** recebe dois parâmetros: a lista (L) e o valor a pesquisar. Se o valor for encontrado, retornaremos o valor de sua posição em ❶. Caso não seja encontrado, retornaremos **None** em ❷. Observe que, se retornarmos em ❶, tanto **for** quanto o **return** em ❷ são completamente ignorados.

**enumerate:** função que pode receber como entrada uma **lista** e irá retornar um objeto do tipo **enumerate**, que poderá ser percorrido pelo **for**

# • FUNÇÕES EM PYTHON

```
funcao.py - C:\Users\rrjjccss\AppData\Loca...
File Edit Format Run Options Window Help

def pesquisa(lista, valor):
    for x, e in enumerate(lista):
        if e == valor:
            return x
    return None

L=[10,20,25,30]
print(pesquisa(L,25))
print(pesquisa(L,27))
```

Ln: 8 Col: 14

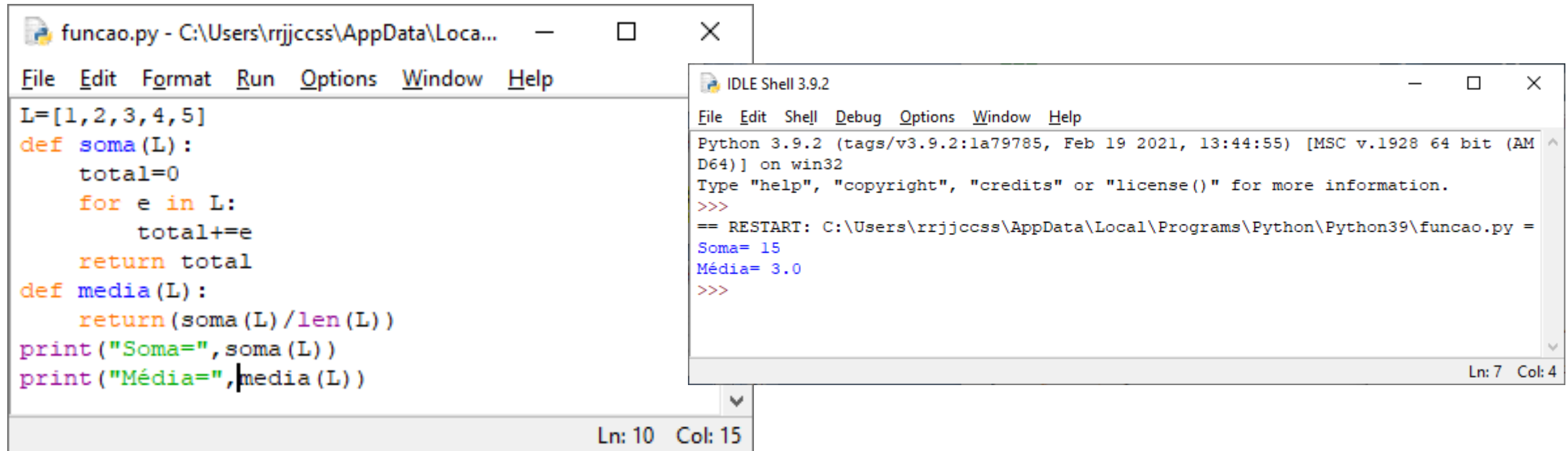
```
IDLE Shell 3.9.2
File Edit Shell Debug Options Window Help

Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:\Users\rrjjccss\AppData\Local\Programs\Python\Python39\funcao.py =
2
None
>>> |
```

Ln: 7 Col: 4

# • FUNÇÕES EM PYTHON

- Cálculo da soma e média de uma lista



The image shows a Python IDE window titled 'funcao.py' and an 'IDLE Shell 3.9.2' window. The code in the editor defines a list L, a function 'soma' to calculate the sum, and a function 'media' to calculate the average. The shell shows the execution results: Soma= 15 and Média= 3.0.

```
funcao.py - C:\Users\rrjjcss\AppData\Loca...  
File Edit Format Run Options Window Help  
L=[1,2,3,4,5]  
def soma(L):  
    total=0  
    for e in L:  
        total+=e  
    return total  
def media(L):  
    return(soma(L)/len(L))  
print("Soma=",soma(L))  
print("Média=",media(L))  
Ln: 10 Col: 15
```

```
IDLE Shell 3.9.2  
File Edit Shell Debug Options Window Help  
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
== RESTART: C:\Users\rrjjcss\AppData\Local\Programs\Python\Python39\funcao.py =  
Soma= 15  
Média= 3.0  
>>>  
Ln: 7 Col: 4
```



# • FUNÇÕES EM PYTHON

## • Cálculo do fatorial

```
funcao.py - C:\Users\rrjjccss\AppData\Local\...  
File Edit Format Run Options Window Help  
x=int(input("Informe um número inteiro >= 0:"))  
def fatorial(n):  
    fat=1  
    while n>1:  
        fat*=n  
        n-=1  
    return fat  
print("O fatorial de %d é %d"%(x,fatorial(x)))  
Ln: 1 Col: 41
```

```
IDLE Shell 3.9.2  
File Edit Shell Debug Options Window Help  
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
== RESTART: C:\Users\rrjjccss\AppData\Local\Programs\Python\Python39\funcao.py =  
Informe um número inteiro >= 0:5  
O fatorial de 5 é 120  
>>> |  
Ln: 7 Col: 4
```

- **FUNÇÕES EM PYTHON**
- Calcular salário

```
salario.py - C:/Users/rrjjcss/AppData/Local/Programs/Python/Python39/salario.py (3.9.2)
File Edit Format Run Options Window Help

def calcular_pagamento(qtd_horas, valor_hora):
    horas = float(qtd_horas)
    taxa = float(valor_hora)
    if horas <= 40:
        salario=horas*taxa
    else:
        h_excd = horas - 40
        salario = 40*taxa+(h_excd*(1.5*taxa))
    return salario
str_horas= input('Digite as horas: ')
str_taxa=input('Digite a taxa: ')
total_salario = calcular_pagamento(str_horas,str_taxa)
print('O valor de seus rendimentos é R$%.2f'% total_salario)
```

```
IDLE Shell 3.9.2
File Edit Shell Debug Options Window Help

Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/rrjjcss/AppData/Local/Programs/Python/Python39/salario.py =
Digite as horas: 44
Digite a taxa: 20
O valor de seus rendimentos é R$920.00
>>>
```

Ln: 8 Col: 4

Ln: 13 Col: 46

# • FUNÇÕES EM PYTHON

- Parâmetros nomeados

Mesmo que venhamos a **trocar a ordem dos argumentos** na chamada da função, ela será executada corretamente da mesma forma, pois os parâmetros estão sendo identificados por um nome e não pela sua posição.

```
*imc.py - C:/Users/rrjjccss/AppData/Local/Program...
File Edit Format Run Options Window Help
#Sem nomeação parâmetros
def calculo_imc(peso, altura):
    print("IMC=%.2f"%(peso / altura ** 2))

calculo_imc(75, 1.68)

Ln: 1 Col: 24
```

```
IDLE Shell 3.9.2
File Edit Shell Debug Options Window Help
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: C:/Users/rrjjccss/AppData/Local/Programs/Python/Python39/imc.py ===
IMC=26.57
>>> |

Ln: 6 Col: 4
```

```
imc.py - C:/Users/rrjjccss/AppData/Local/Programs...
File Edit Format Run Options Window Help
#Com nomeação parâmetros
def calculo_imc(peso, altura):
    print("IMC=%.2f"%(peso / altura ** 2))

calculo_imc(peso=75, altura=1.68)

Ln: 5 Col: 33
```

```
IDLE Shell 3.9.2
File Edit Shell Debug Options Window Help
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: C:/Users/rrjjccss/AppData/Local/Programs/Python/Python39/imc.py ===
IMC=26.57
>>>

Ln: 6 Col: 4
```

- **VARIÁVEIS LOCAIS(INTERNAS) E GLOBAIS(EXTERNAS)**
  - Quando usamos funções, começamos a trabalhar com variáveis **internas ou locais** e com variáveis **externas ou globais**
  - A diferença entre elas é a **visibilidade** ou **escopo**

## • VARIÁVEIS LOCAIS (INTERNAS)

- Uma variável local a uma função existe apenas dentro dela, sendo normalmente inicializada a cada chamada
- Não podemos acessar o valor de uma variável local fora da função que a criou e, por isso, passamos parâmetros e retornamos valores nas funções, de forma a possibilitar a troca de dados no programa

# • VARIÁVEIS LOCAIS (INTERNAS)

```
local.py - C:/Users/rrjjccss/AppData/Local/Programs/Python/P...
File Edit Format Run Options Window Help
a=5
b=2
def soma(x,y):
    s=x+y
    return s
print("Soma de %d e %d é igual a %d"%(a,b,soma(a,b)))
```

Ln: 7 Col: 0

**Escopo da função**

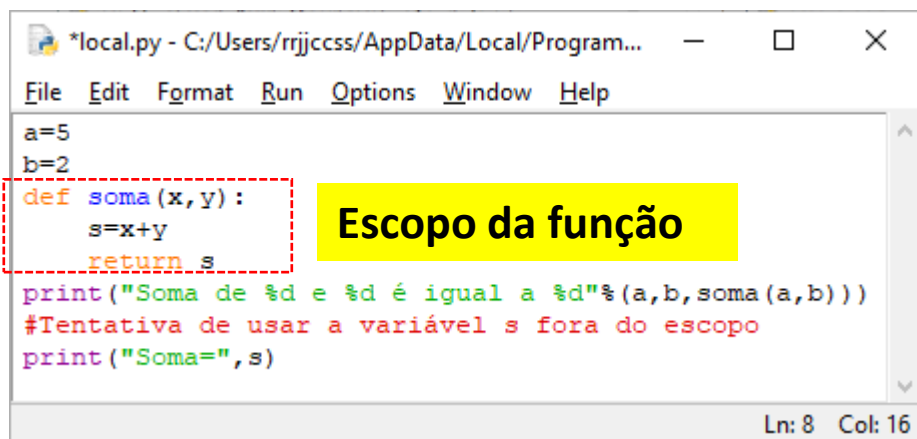
```
IDLE Shell 3.9.2
File Edit Shell Debug Options Window Help
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/rrjjccss/AppData/Local/Programs/Python/Python39/local.py ==
Soma de 5 e 2 é igual a 7
>>> |
```

Ln: 6 Col: 4

s é um exemplo de variável **local(interna)** pois está dentro do escopo da função soma.

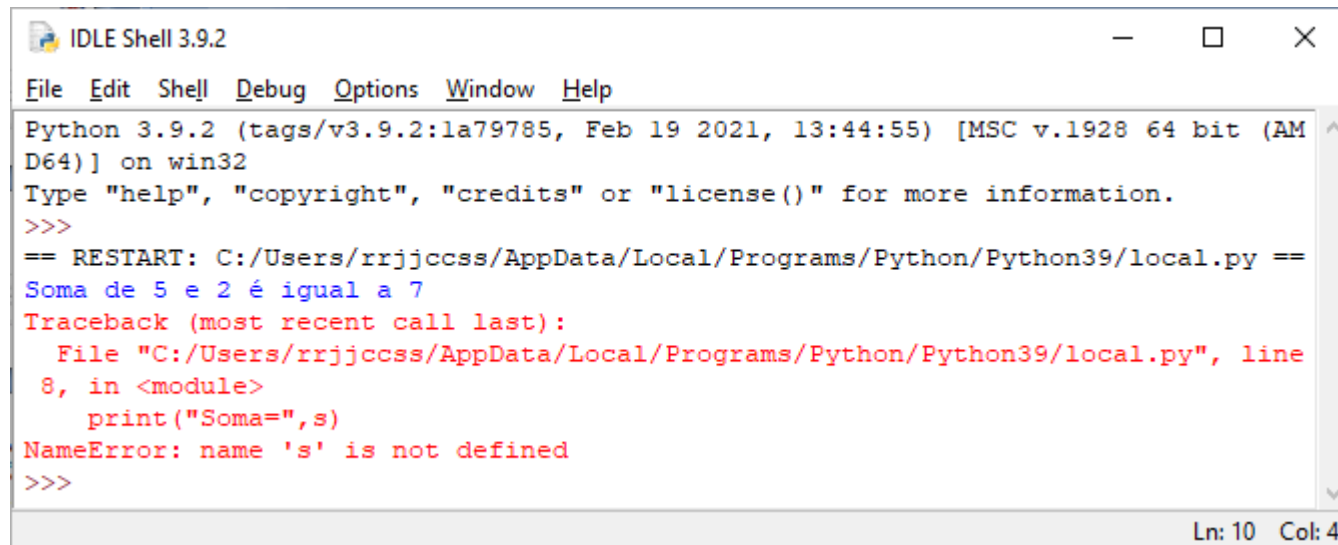
# • VARIÁVEIS LOCAIS (INTERNAS)

`s` é um exemplo de variável **local(interna)** pois está dentro do escopo da função `soma`. Se tentarmos usar a variável `s` fora do escopo da função, a variável não estará definida.



```
*local.py - C:/Users/rrjjccss/AppData/Local/Program...
File Edit Format Run Options Window Help
a=5
b=2
def soma(x, y):
    s=x+y
    return s
print("Soma de %d e %d é igual a %d"%(a,b,soma(a,b)))
#Tentativa de usar a variável s fora do escopo
print("Soma=",s)
Ln: 8 Col: 16
```

Escopo da função



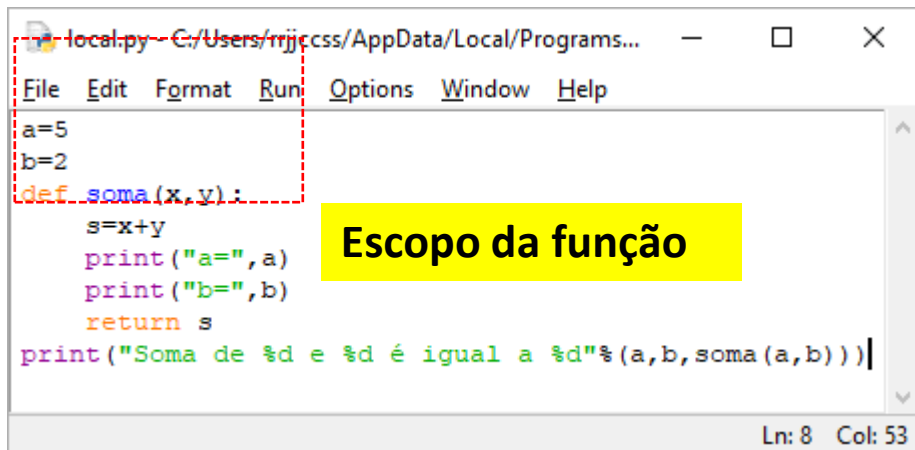
```
IDLE Shell 3.9.2
File Edit Shell Debug Options Window Help
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/rrjjccss/AppData/Local/Programs/Python/Python39/local.py ==
Soma de 5 e 2 é igual a 7
Traceback (most recent call last):
  File "C:/Users/rrjjccss/AppData/Local/Programs/Python/Python39/local.py", line 8, in <module>
    print("Soma=",s)
NameError: name 's' is not defined
>>>
Ln: 10 Col: 4
```



# • VARIÁVEIS GLOBAIS (EXTERNAS)

- Uma variável global é definida **fora de uma função**, pode ser vista por todas as funções do módulo e por todos os módulos que importam o módulo que a definiu

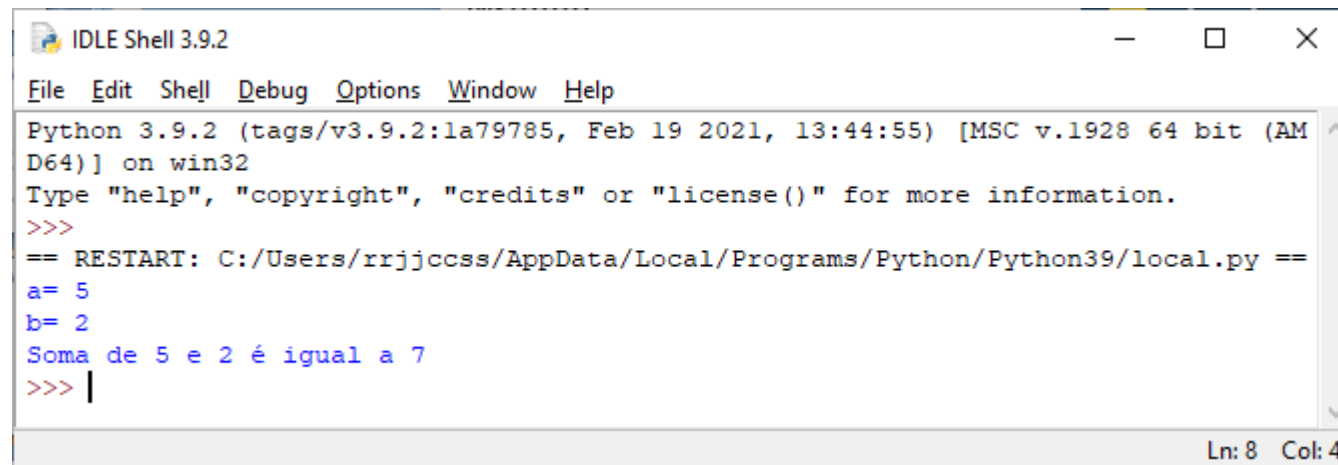
**a e b** são exemplos de variáveis **globais(externas)** pois está fora do escopo da função soma e podem ser usadas fora ou dentro da função.



```
local.py - C:/Users/rrjjccss/AppData/Local/Programs...  
File Edit Format Run Options Window Help  
a=5  
b=2  
def soma(x,y):  
    s=x+y  
    print("a=",a)  
    print("b=",b)  
    return s  
print("Soma de %d e %d é igual a %d"%(a,b,soma(a,b)))
```

Ln: 8 Col: 53

Escopo da função



```
IDLE Shell 3.9.2  
File Edit Shell Debug Options Window Help  
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
== RESTART: C:/Users/rrjjccss/AppData/Local/Programs/Python/Python39/local.py ==  
a= 5  
b= 2  
Soma de 5 e 2 é igual a 7  
>>> |
```

Ln: 8 Col: 4



## • CUIDADOS COM VARIÁVEIS LOCAIS/GLOBAIS

- Variáveis globais devem ser **utilizadas o mínimo possível** em seus programas, pois dificultam a leitura e violam o **encapsulamento** da função
- Uma variável global pode ser alterada por qualquer função, tornando a tarefa de saber quem altera seu valor realmente mais trabalhosa
- O encapsulamento é comprometido porque a função depende de uma variável externa, ou seja, que não é declarada dentro da função nem recebida como parâmetro

- **CUIDADOS COM VARIÁVEIS LOCAIS/GLOBAIS**
  - Um bom uso de variáveis globais é **guardar valores constantes** e que devem ser acessíveis a todas as funções do programa, como o nome da empresa
  - Elas também são utilizadas em nosso programa principal e **para inicializar o módulo com valores iniciais**
  - Tente utilizar variáveis globais apenas para configuração e com **valores constantes**

# • CUIDADOS COM VARIÁVEIS LOCAIS/GLOBAIS

```
a=5 ❶  
def muda_e_imprime():  
    a=7 ❷  
    print("A dentro da função: %d" % a)  
print("a antes de mudar: %d" % a) ❸  
muda_e_imprime() ❹  
print("a depois de mudar: %d" % a) ❺
```

Em ❶, criamos a variável global “a”.

Em ❷, temos a variável local da função, também chamada “a”, recebendo 7.

Em ❸ e ❺, imprimimos o valor da variável global “a” e, em ❹, o valor da variável local “a”.

Para o computador, essas variáveis são completamente diferentes, embora tenham o mesmo nome.

Em ❹, podemos acessar seu conteúdo porque realizamos a impressão dentro da função.

Da mesma forma, a variável “a” que imprimimos é a variável local, valendo 7.

Essa variável local deixa de existir no final da função e explica porque não alteramos a variável global “a”.

# • CUIDADOS COM VARIÁVEIS LOCAIS/GLOBAIS

```
local.py - C:/Users/rrjjccss/AppData/Local/Programs...  
File Edit Format Run Options Window Help  
a=5  
def muda_e_imprime():  
    a=7  
    print("A dentro da função: %d" % a)  
print("a antes de mudar: %d" % a)  
muda_e_imprime()  
print("a depois de mudar: %d" % a)  
|
```

Ln:

```
IDLE Shell 3.9.2  
File Edit Shell Debug Options Window Help  
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
== RESTART: C:/Users/rrjjccss/AppData/Local/Programs/Python/Python39/local.py ==  
a antes de mudar: 5  
A dentro da função: 7  
a depois de mudar: 5  
>>> |
```

Ln: 8 Col: 4

- **CUIDADOS COM VARIÁVEIS LOCAIS/GLOBAIS**
  - Se quisermos modificar uma **variável global** dentro de uma função, devemos informar que estamos usando uma variável global antes de inicializá-la, na primeira linha de nossa função
  - Essa definição é feita com a instrução **global**

# • CUIDADOS COM VARIÁVEIS LOCAIS/GLOBAIS

```
a=5  
  
def muda_e_imprime():  
    global a ❶  
    a=7 ❷  
    print("A dentro da função: %d" % a)  
print("a antes de mudar: %d" % a)  
muda_e_imprime()  
print("a depois de mudar: %d" % a)
```

Agora, em ❶, estamos trabalhando com a variável “a” global. Assim, quando fizemos a=7 em ❷, trocamos o valor de “a” global.

# • CUIDADOS COM VARIÁVEIS LOCAIS/GLOBAIS

```
*local.py - C:/Users/rrjjccss/AppData/Local/Program...
File Edit Format Run Options Window Help
a=5
def muda_e_imprime():
    global a
    a=7
    print("A dentro da função: %d" % a)
print("a antes de mudar: %d" % a)
muda_e_imprime()
print("a depois de mudar: %d" % a)
Ln: 3 Co
```

```
IDLE Shell 3.9.2
File Edit Shell Debug Options Window Help
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/rrjjccss/AppData/Local/Programs/Python/Python39/local.py ==
a antes de mudar: 5
A dentro da função: 7
a depois de mudar: 7
>>> |
Ln: 8 Col: 4
```



- **PARÂMETROS OPCIONAIS**
- Imprimir uma barra na tela

```
local.py - C:\Users\rrjjccss\AppData\Local\Programs\Python\Python39\local.py (3.9.2)
File Edit Format Run Options Window Help

def barra():
    print("***60")
print("Exibindo uma barra na tela:")
print()
barra()

Ln: 5 Col: 7
```

```
IDLE Shell 3.9.2
File Edit Shell Debug Options Window Help
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:\Users\rrjjccss\AppData\Local\Programs\Python\Python39\local.py ==
Exibindo uma barra na tela:

*****

>>> |

Ln: 8 Col: 4
```

```
*local.py - C:\Users\rrjjccss\AppData\Local\Programs\Python\Python39\local.py (3.9.2)*
File Edit Format Run Options Window Help

#Usando parâmetros opcionais
def barra(n=40, caractere="*"):
    print(caractere * n)
print("Exibindo uma barra na tela com parâmetros opcionais:")
print()
barra(10, "-")

Ln: 1 Col: 28
```

```
IDLE Shell 3.9.2
File Edit Shell Debug Options Window Help
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:\Users\rrjjccss\AppData\Local\Programs\Python\Python39\local.py ==
Exibindo uma barra na tela com parâmetros opcionais:

-----

>>> |

Ln: 8 Col: 4
```



# • FUNÇÕES CHAMANDO OUTRAS FUNÇÕES

- Cada uma das funções que a gente escreve pode ser usada e chamada pelas outras funções

```
soma_q.py - C:/Users/rrjjccss/AppData/Local/P...
File Edit Format Run Options Window Help
def quadrado(x):
    y=x*x
    return y
def soma_quadrados(x,y,z):
    a=quadrado(x)
    b=quadrado(y)
    c=quadrado(z)
    return a+b+c
a=float(input("A="))
b=float(input("A="))
c=float(input("A="))
print("Soma=%2f"%(soma_quadrados(a,b,c)))
Ln: 13 Col: 0
```

```
IDLE Shell 3.9.2
File Edit Shell Debug Options Window Help
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/rrjjccss/AppData/Local/Programs/Python/Python39/soma_q.py ==
A=5
A=6
A=2
Soma=65.00
>>> |
Ln: 9 Col: 4
```

# • FUNÇÕES CHAMANDO OUTRAS FUNÇÕES

```
fun_fun.py - C:/Users/rrjjcss/AppData/Local/Programs/Python/Python39/fun_fun.py (3.9.2)
File Edit Format Run Options Window Help
def imprime_lista(L, fimpressão, fcondição):
    for e in L:
        if fcondição(e):
            fimpressão(e)
def imprime_elemento(e):
    print("Valor: %d" % e)
def épar(x):
    return x % 2 == 0
def éimpar(x):
    return not épar(x)
L=[1,7,9,2,11,0]
imprime_lista(L, imprime_elemento, épar)
print()
imprime_lista(L, imprime_elemento, éimpar)

IDLE Shell 3.9.2
File Edit Shell Debug Options Window Help
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/rrjjcss/AppData/Local/Programs/Python/Python39/fun_fun.py =
Valor: 2
Valor: 0

Valor: 1
Valor: 7
Valor: 9
Valor: 11
>>> |

Ln: 12 Col: 4
```

# REFERÊNCIAS

- MENEZES, Nilo Ney Coutinho. **Introdução a Programação com Python**: algoritmos e lógica de programação para iniciantes. 2. ed. São Paulo: Novatec, 2014. 324 p.
- GOOGLE. **Perguntas frequentes**. Disponível em <<https://research.google.com/colaboratory/faq.html>>. Acesso em 20/03/2021.
- RODRIGUES, Sigdri Ferreira. **Google Colab - Guia do Iniciante**. Disponível em <<https://medium.com/machina-sapiens/google-colab-guia-do-iniciante-334d70aad531>>. Acesso em 20/03/2021.
- CAELUM. **O que é python?** Disponível em <<https://www.caelum.com.br/apostila-python-orientacao-objetos/o-que-e-python/#python>>. Acesso em 03/04/2020.
- PYTHON.ORG. **Python no Windows**. Disponível em <<https://docs.python.org/pt-br/3/faq/windows.html>>. Acesso em 03/04/2020.
- EDITORA MELHORAMENTOS. **Compilar**. Disponível em <<https://michaelis.uol.com.br/moderno-portugues/busca/portugues-brasileiro/COMPILAR/>>.
- BANCZEC, Jhonathan Paulo. **Introdução à Python**. Disponível em <<https://pt.slideshare.net/jhoonb/introduo-python-mdulo-1>>. Acesso em 04/04/2020.
- LIMA, Stefani. **Funções Built-in do Python**. Disponível em <<https://stefanilima.com.br/2018/04/27/funcoes-built-in-do-python/>>. Acesso em 04/04/2020.