FUNÇÕES

Funções pré-definidas, criando funções, escopo de funções, função como argumento

FUNÇÕES

- As linguagens de programação têm à sua disposição várias funções pré-definidas:
- EXEMPLO (Linguagem *Python*)
  - input()
  - print()
  - math.sqrt() (utilizando *import math*)
  - abs()

FUNÇÕES

- As linguagens de programação têm à sua disposição várias funções pré-definidas:
- EXEMPLO (Linguagem *Python*)
  - input()
  - print()
  - math.sqrt() (utilizando *import math*)

FUNÇÕES

- As linguagens de programação têm à sua disposição várias funções pré-definidas:
- EXEMPLO (Linguagem *Python*)
  - input()
  - print()
  - math.sqrt() (utilizando *import math*)
  - a s()

FUNÇÕES

- As linguagens de programação têm à sua disposição várias funções pré-definidas:
- EXEMPLO (Linguagem *Python*)
  - input()
  - print()
  - math.sqrt(16) (utilizando *import math*)
  - abs(-3)

FUNÇÕES

- As Funções Pré Definidas podem ser usadas diretamente em expressões:

- Exemplo:

```
h= math.sqrt (a + y**2 + 2 * abs(math.sin(y) ))
```

FUNÇÕES

- As Funções Pré Definidas podem ser usadas diretamente em expressões:
- Exemplo:

h= sqrt (a + pow (y,2) + 2 * sin (y))

Identificadores das FUNÇÕES

FUNÇÕES

Se houver necessidade o programador pode **definir** suas próprias **FUNÇÕES**

FUNÇÕES

- Permitem modularizar um programa.
- Facilitam o uso da abordagem dividir e conquistar no desenvolvimento de programas.
- Facilitam a reusabilidade de códigos.

FUNÇÕES

```
#declaração da função
def <nome>(<argumentos>):
    """
    Descreve input
    Descreve output
    """
    instrução
    instrução
    instrução
    instrução

    return <valorRetornado>

#chamada da função
variavel=<nome>(<arg1,arg2,...,argN)
```

FUNÇÕES

```
#declaração da função
def <nome>(<argumentos>):
    """
    Des
    Descr
    """
    instrução
    instrução
    instrução
    instrução

    return <valorRetornado>

#chamada da função
variavel=<nome>(<arg1,arg2,...,argN)
```

Palavra reservada

FUNÇÕES

```
#declaração da função
def <nome>(<argumentos>):
    """
    Descreve input
    Descreve output
    """
    instrução
    instrução
    instrução
    instrução

    return <valorRetornado>

#chamada da função
variavel=<nome>(<arg1,arg2,...,argN)
```

docstring (opcional!!)

FUNÇÕES

```
#declaração da função
def <nome>(<argumentos>):
    """
    Descreve input
    Descreve output
    """
    instrução
    instrução
    instrução
    instrução

    return <valorRetornado>

#chamada da função
variavel=<nome>(<arg1,arg2,...,argN)
```

Corpo da função

FUNÇÕES

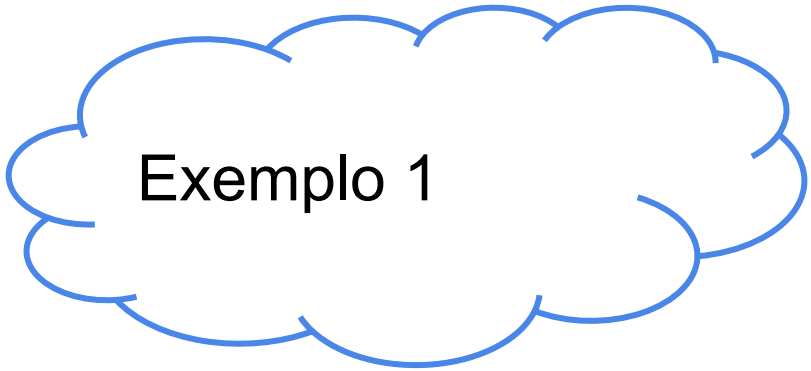
```
#declaração da função
def <nome>(<argumentos>):
    """
    Descreve input
    Descreve output
    """
    instrução
    instrução
    instrução
    instrução

    return <valorRetornado>

#chamada da função
variavel=<nome>(<arg1,arg2,...,argN)
```

Palavra reservada

FUNÇÕES

```
#declaração da função
def <nome>(<argumentos>):
    """
    Descreve input
    Descreve output
    """
    instrução
    instrução
    instrução
    instrução

    return <valorRetornado>

#chamada da função
variavel=<nome>(<arg1,arg2,...,argN)
```

FUNÇÕES




```
#declaração da função
def fat(n):
   f=1
   for(i in range(n):
        f=f*i
   return f
#chamada da função
y=fat(5)
```

FUNÇÕES

- Todas as variáveis criadas são variáveis locais
  - Variáveis locais: existem apenas dentro da função onde são criadas.

- As variáveis na lista de argumentos da função também são variáveis locais.

- As variáveis na lista de argumentos permitem a troca de informações entre funções.

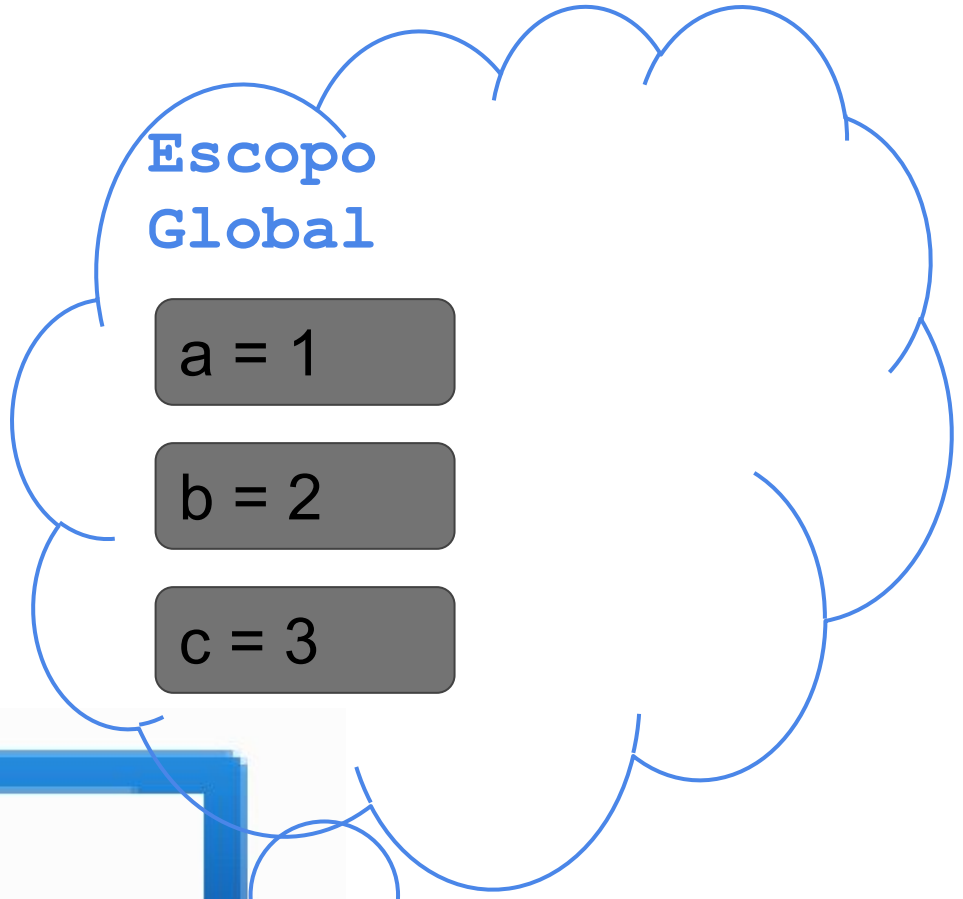
FUNÇÕES - Escopo das Variáveis

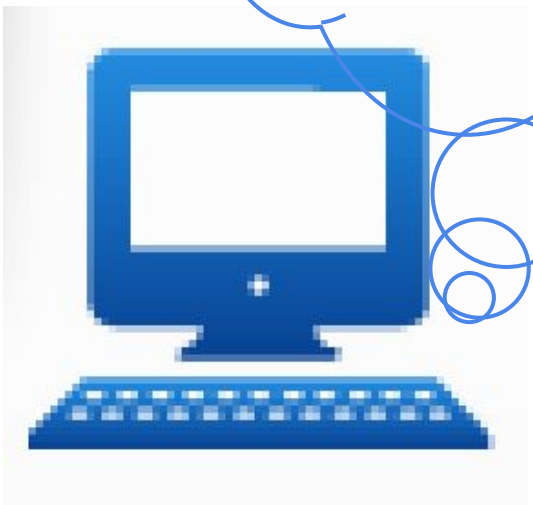
```
#declaração da função

def f(x):
    a=b=c=a=x
    print(a,b,c)
    return a+b+c

a,b,c=1,2,3
#chamada da função
d=f(4)
print(d)
print(a,b,c)
```

Exemplo 2

FUNÇÕES - Escopo das Variáveis

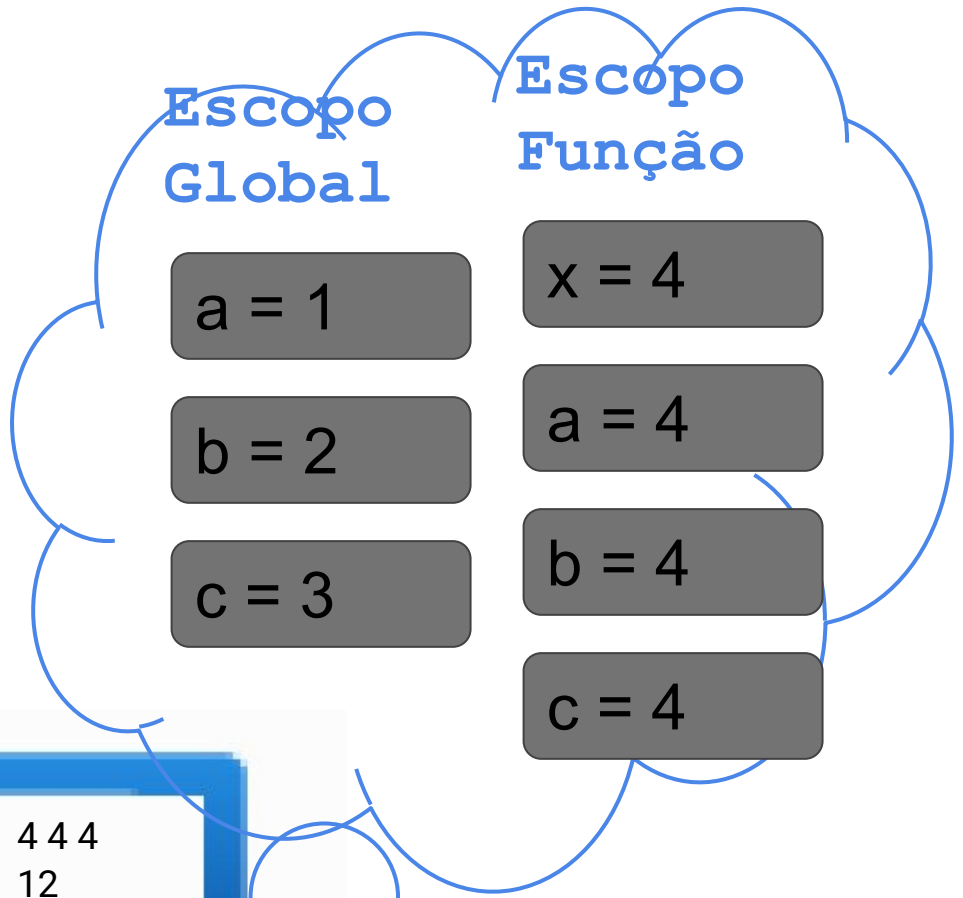
```
#declaração da função

def f(x):
    a=b=c=a=x
    print(a,b,c)
    return a+b+c

a,b,c=1,2,3
#chamada da função
d=f(4)
print(d)
print(a,b,c)
```

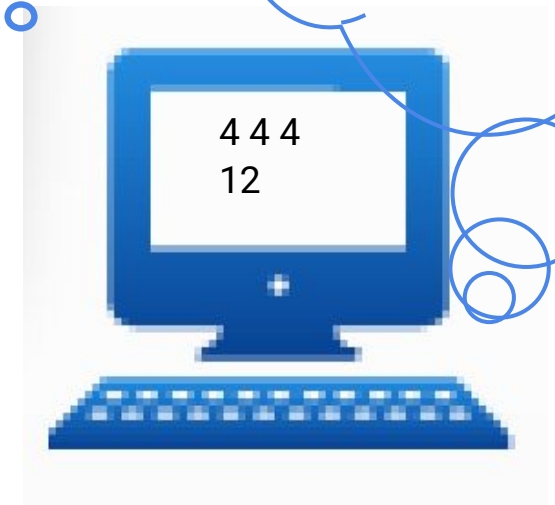




FUNÇÕES - Escopo das Variáveis

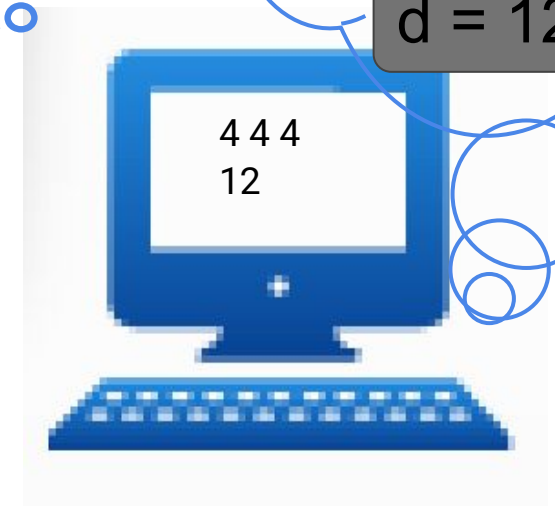
```
#declaração da função

def f(x):
    a=b=c=a=x
    print(a,b,c)
    return a+b+c

a,b,c=1,2,3
#chamada da função
d=f(4)
print(d)
print(a,b,c)
```

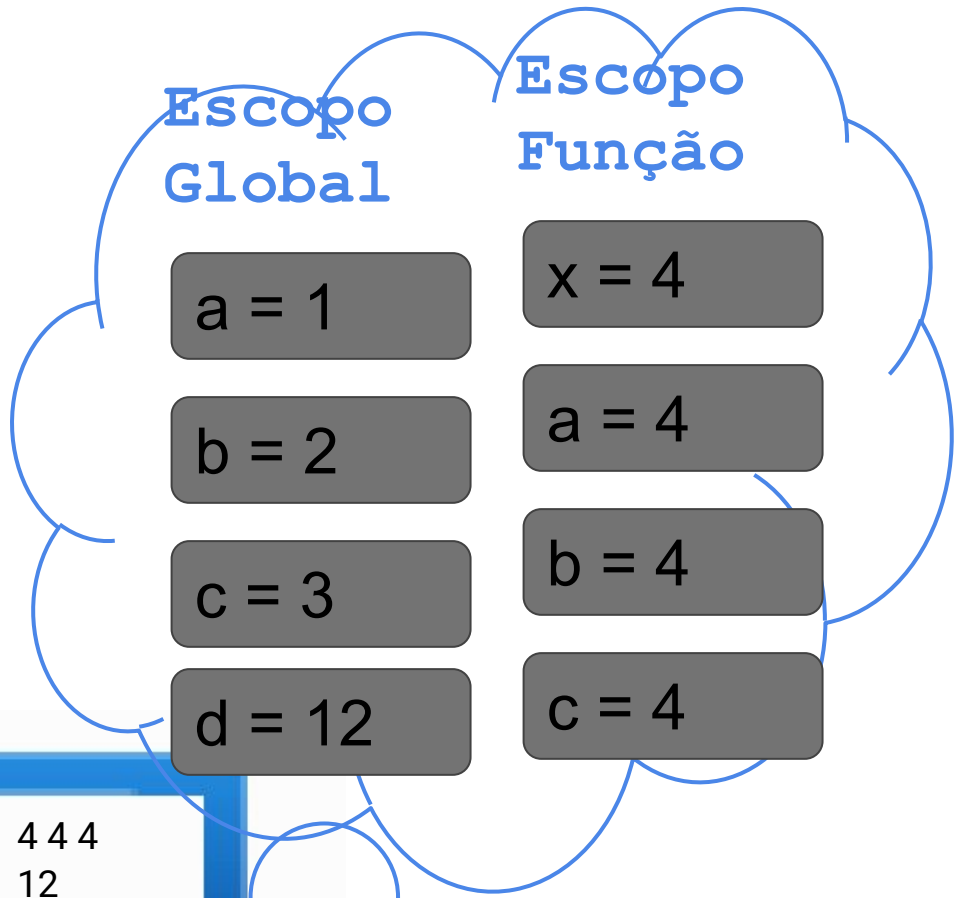




FUNÇÕES - Escopo das Variáveis

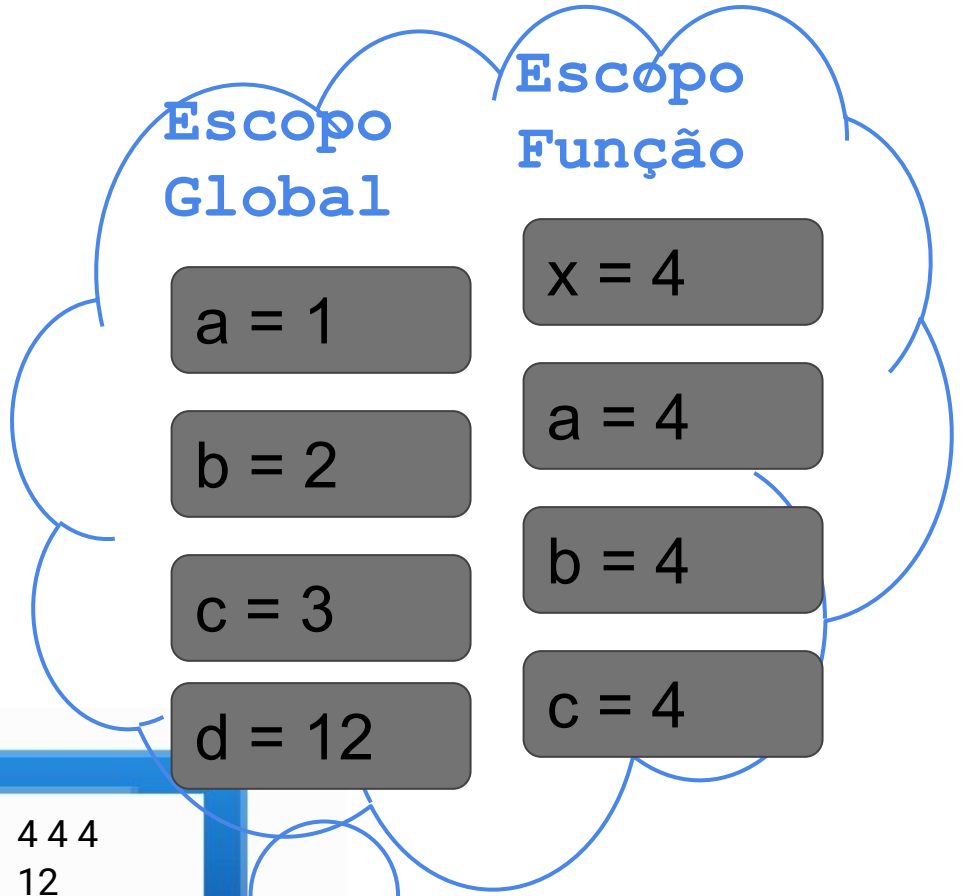
```
#declaração da função

def f(x):
    a=b=c=a=x
    print(a,b,c)
    return a+b+c

a,b,c=1,2,3
#chamada da função
d=f(4)
print(d)
print(a,b,c)
```

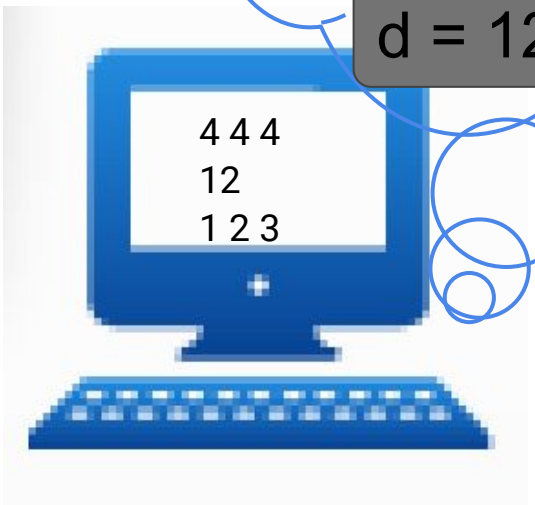




FUNÇÕES - Escopo das Variáveis

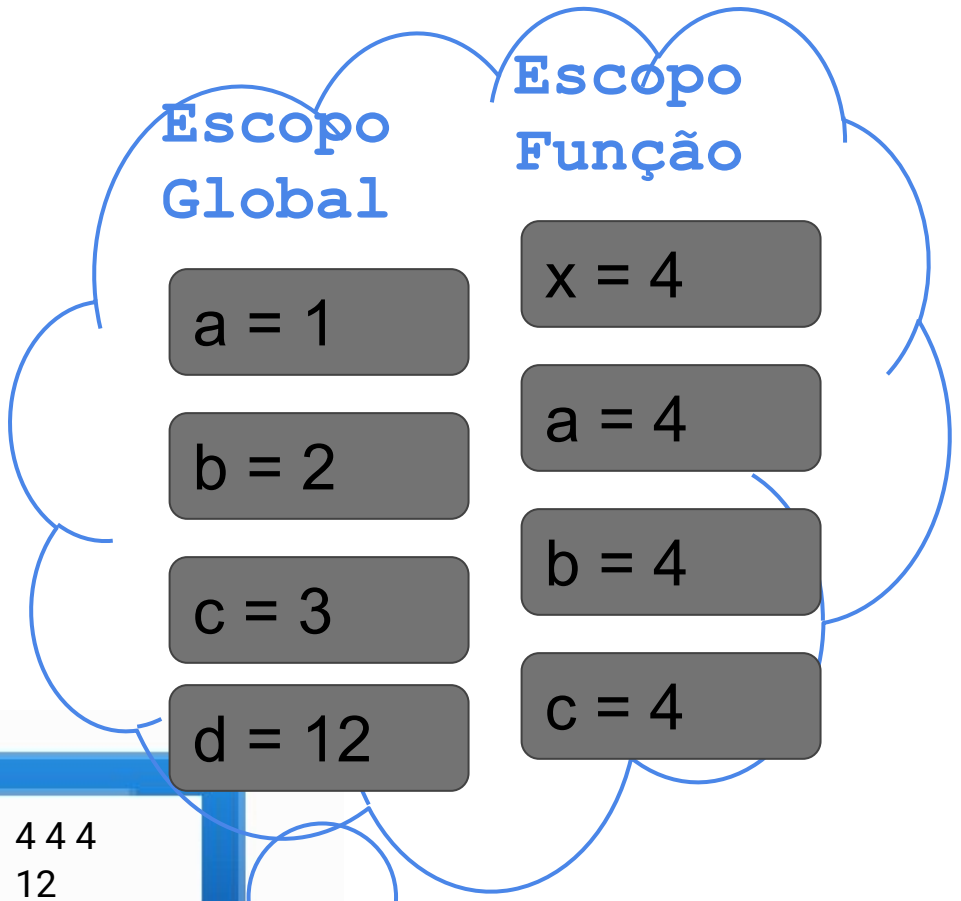
```
#declaração da função

def f(x):
    a=b=c=a=x
    print(a,b,c)
    return a+b+c

a,b,c=1,2,3
#chamada da função
d=f(4)
print(d)
print(a,b,c)
```

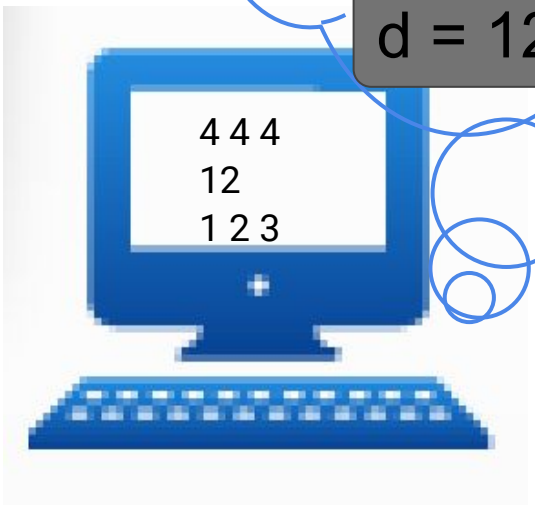




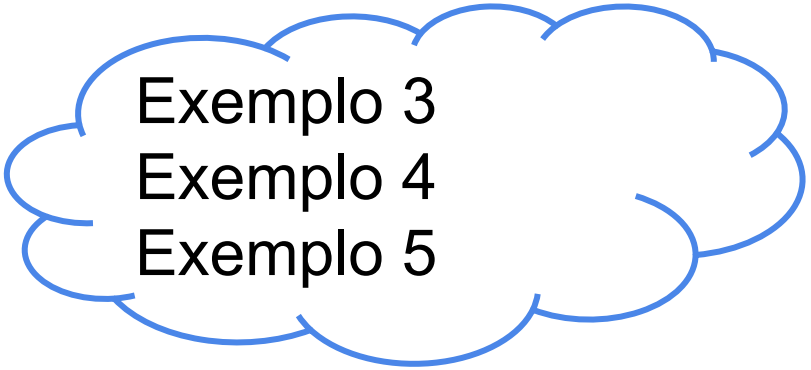
FUNÇÕES - Escopo das Variáveis

```
#declaração da função

def f(x):
    a=b=c=a=x
    print(a,b,c)
    return a+b+c

a,b,c=1,2,3
#chamada da função
d=f(4)
print(d)
print(a,b,c)
```





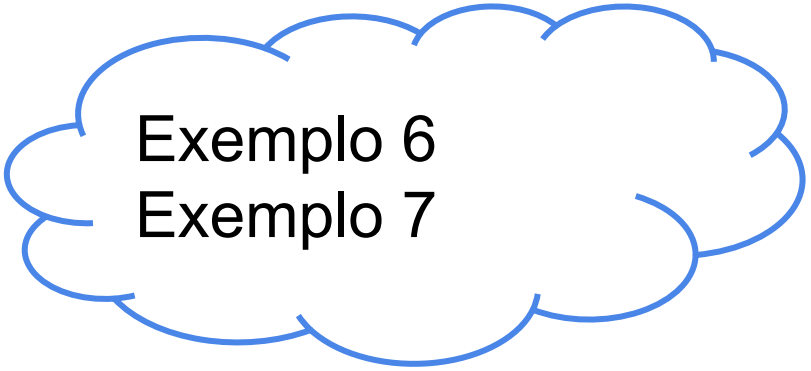
FUNÇÕES - Escopo das Variáveis



FUNÇÕES - Return

- O comando **return** é responsável por encerrar a execução da função e retornar o valor desejado.

- Uma função sem o comando return retorna None que indica a ausência de valor retornado.



FUNÇÕES - função como argumento

```
def fat(n):
    f=1
    for i in range(1,n+1):
        f=f*i
    return f

def areaCirculo(r):
    return math.pi*(r**2)

def soma(valor,func):
    print(valor, func(5))
    return valor+func(5)

print(soma(fat(5),areaCirculo))
print(soma(areaCirculo(5),fat))
```