



---

## **GUIA DE LABORATÓRIO 3.2 - FUNÇÕES E ÂMBITOS (*SCOPES*)**

### **RESOLUÇÃO DOS EXERCÍCIOS DE REVISÃO**

---

- 1. FUNÇÃO** → Genericamente, uma função é um conjunto de operações/instruções, com um propósito bem definido e que poderemos querer reutilizar em diversas situações. Normalmente, a esse conjunto de instruções damos um nome, e é através deste nome que nos referimos à função no resto do programa. Em determinadas linguagens de programação não é obrigatório que uma função devolva ou retorne um valor.

Do ponto de vista da linguagem Python, uma função é um bloco de código com um nome, lista de parâmetros e um valor de retorno, sendo que esse valor de retorno pode ser um conjunto de valores. Uma lista de parâmetros é uma sequência de 0 ou mais parâmetros (o que é um parâmetros? vem já a seguir...)

**PARÂMETRO** → Um parâmetro é uma variável *local* à função e que serve para que uma outra parte do código comunique com a função à qual o parâmetro pertence.

**ARGUMENTO** → Valor passado para um parâmetro. Normalmente, a associação entre PARÂMETRO e ARGUMENTO é feita "posicionalmente".

**ARGUMENTO COM NOME** → É um argumento que é associado ao parâmetro respectivo através do nome deste.

**PARÂMETRO OPCIONAL** → Parâmetro que possui um valor por omissão.

- 2.** (após termos feito `import math`)

```
elevado_a = math.pow
```

- 3.**

```
y = 10
def f(x):
    print(x + y)
f(3)
def g():
    y = 20
    f(3)
g()
```

13

13

<pre> y = 5 def func1():     def f(x):         print(x + y)     f(10) def func2():     def f(x):         print(x + y)     y = 50     f(10) func1() func2() </pre>	<pre> 15 60 </pre>
<pre> y = 10 def func3():     def f1():         y = 1     def f2():         nonlocal y         y = 2     def f3():         global y         y = 3     y = 0     f1()     print(y)     f2()     print(y)     f3()     print(y) func3() print(y) </pre>	<pre> 0 2 2 3 </pre>

4. **LAMBDA** → Permite definir uma função anónima. Em Python, uma *lambda* (ie, uma função anónima) apenas pode conter uma instrução e essa instrução deve ser uma expressão, ou seja, deve produzir um valor. Uma **FUNÇÃO ANÓNIMA** é uma função sem nome.
5. Não.
6. Devolvem um dicionário com os nomes definidos globalmente (ie, ao nível do módulo) e localmente (ao nível da função que está a ser definida).
7. TPC
- 8.

**8.1** a -> [3, 5]

**8.2**   `b -> [False, False, False, True, False, False, False, True]`

**8.3**   `c -> ('i', 'a', 'a', 'a')`

**8.4**   `d -> 'E/o'`