

O que são funções?

Função é um bloco de código identificado por um nome e pode receber uma lista de parâmetros, esses parâmetros podem ou não ter valores padrões. Usar funções torna o código mais legível e possibilita o reaproveitamento de código. Programar baseado em funções, é o mesmo que dizer que estamos programando de maneira estruturada.

Exemplo

```
def exibir_mensagem():  
    print("Olá mundo!")  
  
def exibir_mensagem_2(nome):  
    print(f"Seja bem vindo {nome}!")  
  
def exibir_mensagem_3(nome="Anônimo"):  
    print(f"Seja bem vindo {nome}!")  
  
exibir_mensagem()  
exibir_mensagem_2(nome="Guilherme")  
exibir_mensagem_3()  
exibir_mensagem_3(nome="Chappie")
```

Retornando valores

Para retornar um valor, utilizamos a palavra reservada **return**.
Toda função Python retorna **None**.

Exemplo

```
def calcular_total(numeros):  
    return sum(numeros)  
  
def retorna_antecessor(numero):  
    antecessor = numero - 1  
  
    return antecessor, sucessor  
  
calcular_total([10, 20, 34]) # 64  
retorna_antecessor(10) # (9)
```

Escopo local e escopo global

Python trabalha com escopo local e global, dentro do bloco da função o escopo é local. Portanto, alterações ali feitas em variáveis locais serão perdidas quando o método terminar de ser executado. Para usar objetos globais utilizamos a palavra-chave **global**, que informa ao interpretador que a variável que está sendo manipulada no escopo local é global. Essa **NÃO** é uma boa prática e deve ser evitada.

Exemplo

```
salario = 2000

def salario_bonus(bonus):
    global salario
    salario += bonus
    return salario

salario_bonus(500) # 2500
```

EXERCÍCIOS

1. Crie uma função chamada `imprimeOla()`. Essa função deverá imprimir a mensagem “Olá” na tela.
2. Crie uma função chamada `imprimeMsg(mensagem)`. Essa função deverá receber 1 parâmetro “mensagem” do tipo (STR) e imprimir esse parâmetro recebido na tela.
3. Crie uma função chamada `maiorNumero(num1, num2)`. Essa função deverá receber 2 parâmetros do tipo (INT) e imprimir o maior número. Se os números forem iguais, imprime o primeiro parâmetro.
4. Crie uma função chamada `verificaPar(num)`. Essa função deverá receber 1 parâmetro do tipo (INT) e deverá retornar uma STR. Se o número for par, o retorno da função deverá ser “PAR”, caso contrário o retorno deverá ser “IMPAR”.
5. Crie uma função chamada `soma(num1, num2)`. Essa função deverá receber 2 parâmetros do tipo INT E deverá retornar a soma dos dois números recebidos.
6. Crie uma função chamada `areaCirculo(raio)`. Essa função deverá receber 1 parâmetro do tipo FLOAT E deverá retornar a área do circulo. FORMULA: ($\text{ÁREA} = \text{PI} \times \text{Raio}^2$)