



Funções

Algoritmos e Programação em Python

Prof. Fabio Fernando Kobs, Dr.

Funções - Conceito

- Em Python, uma função é uma sequência de comandos que executa alguma tarefa e que tem um nome.
- A sua principal finalidade é nos ajudar a organizar programas em pedaços que correspondam a como imaginamos uma solução do problema.
- A sintaxe de uma definição de função é:

```
def NomeDaFunção (parâmetros):    # cabeçalho da função
```

Comandos ou instruções
- O nome da função deve seguir a regra de identificadores permitidos. Os parâmetros especificam qual dado ou informação a função deve receber para que possa ser usada ou executada.

Funções - Exemplo

- Exemplo de uma nova função com dois argumentos separados por vírgula:

```
def soma(a, b):    # soma é o nome da função
    print(a + b)    # bloco da função com recuo à direita
```

```
soma(2,9)          # chama a função soma, enviando os valores 2 e 9, onde
soma(7,8)          # o valor 2 será recebido pelo nome a e 9 pelo nome b
soma(10,15)
```

Funções – Retorno de valor

- Para retornar um valor de uma função utiliza-se a instrução **return**. Por exemplo, deixa-se na função os cálculos e ao final retorna-se **o resultado** para impressão, ou seja, retorna somente **UM** valor (número, booleano ou texto).
- Exemplo de uma nova função com retorno de um valor:

```
def soma(a, b):    # soma é o nome da função  
    return a + b  # bloco da função com recuo à direita
```

```
print(soma(2,9))  # imprime o valor retornado da função soma
```

- Pode-se dizer que **return** marca o fim da execução da função.

Funções – Retorno de valor

- Exemplo de retorno *string* se valor é par ou ímpar:

```
def par_ímpar(x):  
    if x % 2 == 0:  
        return “Par”  
    else:  
        return “Ímpar”
```

```
print(par_ímpar(4))
```

```
print(par_ímpar(5))
```



Funções – Exercícios

1. Escreva uma função que retorne o maior de dois números.
Valores esperados:
 $\text{m\acute{a}ximo}(5,6) \Rightarrow 6$
 $\text{m\acute{a}ximo}(2,1) \Rightarrow 2$
 $\text{m\acute{a}ximo}(7,7) \Rightarrow 7$
2. Escreva uma função que receba o lado de um quadrado e retorne sua área ($A = \text{lado}^2$).
3. Escreva uma função que receba a base e a altura de um triângulo e retorne sua área ($A = (\text{base} \times \text{altura})/2$).
4. Escreva uma função que receba dois números e retorne True se o primeiro número for múltiplo do segundo.

Funções – Pesquisa em uma lista

- Exemplo:

```
def pesquise(lista, valor):    #a função recebe dois parâmetros
    if valor in lista:
```

```
        print (“Valor encontrado”)
```

```
    else:
```

```
        print(“Valor não encontrado”)
```

```
#####
```

```
L = [10, 20, 25, 30]
```

```
pesquise(L, 25)    # envia dois parâmetros para a função
```

```
pesquise(L, 27)
```

Funções – Cálculo da média

- Exemplo:

```
def média(L):  
    soma = 0  
    for e in L:  
        soma += e  
    return soma/len(L)  
#####  
L = [10,20,25,30]  
print(média(L))
```

Funções – Cálculo do fatorial

- Exemplo:

```
def fatorial(n):
```

```
    fat = 1
```

```
    while n > 1:
```

```
        fat *= n
```

```
        n -= 1
```

```
    return fat
```

```
#####
```

```
print(fatorial(5))
```

Exemplo sem Funções

Fazer um algoritmo para calcular o número de alunos que tiraram nota acima da nota média
da turma. Imprimir a nota destes alunos. As notas são fornecidas via teclado e a turma tem N alunos.

```
turma = []
```

```
# leitura
```

```
while True:
```

```
    nota = float(input('Nota [-1 para fim]: '))
```

```
    if nota == -1:
```

```
        break
```

```
    turma.append(nota)
```

```
# soma
```

```
soma = cont = 0
```

```
for x in turma:
```

```
    soma += x
```

```
# impressão
```

```
if soma > 0:
```

```
    media = soma / len(turma)
```

```
    print(f'Notas maiores que a média: {media}')
```

```
    for x in turma:
```

```
        if x >= media:
```

```
            print(x)
```

```
            cont += 1
```

```
    print(cont, 'alunos com notas iguais ou maiores que a média')
```

```
else:
```

```
    print('Nenhuma nota digitada.')
```

Exemplo com Funções

```
def Leitura(turma):
```

```
    while True:
```

```
        nota = float(input('Nota [-1 para fim]: '))
```

```
        if nota == -1:
```

```
            break
```

```
        turma.append(nota)
```

```
def Soma(turma):
```

```
    soma = 0
```

```
    for x in turma:
```

```
        soma += x
```

```
    return soma
```

```
def Média(turma):
```

```
    soma = Soma(turma)
```

```
    cont = 0
```

```
    if soma > 0:
```

```
        media = soma / len(turma)
```

```
        print(f'Notas maiores que a média: {media}')
```

```
        for x in turma:
```

```
            if x >= media:
```

```
                print(x)
```

```
                cont += 1
```

```
        print(cont, 'alunos com notas iguais ou  
maiores que a média')
```

```
    else:
```

```
        print('Nenhuma nota digitada.')
```

```
#####
```

```
turma = []
```

```
Leitura(turma)
```

```
Média(turma)
```

Parâmetros Opcionais

- Nem sempre será necessário passar todos os parâmetros para uma função, preferindo utilizar um valor previamente escolhido como padrão.

- Exemplo 1:

```
def barra():  
    print("*" * 40)
```

- Exemplo 2:

```
def barra(n=40, caractere="*"):  
    print(caractere * n)
```

- Exemplo 3:

```
>>> barra(10)                                # faz com que n seja 10  
*****  
  
>>> barra(10, "-")
```

Parâmetros Opcionais – Exemplo

```
def soma(a, b, imprime=False)
    s = a + b
    if imprime:
        print(s)
    return s
```

Testar as seguintes instruções e escrever o resultado:

```
>>> soma(2,3)
```

```
>>> soma(3,4,True)
```

```
>>> soma(5,8,False)
```

Desempacotamento de Parâmetros

- Pode-se criar funções que recebem um número indeterminado de parâmetros utilizando listas de parâmetros.
- Exemplo:

```
def soma(*args):  
    s = 0  
    for x in args:  
        s += x  
    return s
```

```
#####
```

```
soma(1,2)
```

```
soma(2)
```

```
soma(5,6,7,8)
```

```
soma(9,10,20,30,40)
```

Exercícios

5. Faça um programa, com uma função que necessite de três argumentos, e que forneça a soma desses três argumentos.
6. Faça um programa, com uma função que necessite de um argumento. A função retorna o caractere 'P', se seu argumento for positivo, e 'N', se seu argumento for negativo ou 'Z' caso zero.
7. Faça um programa com uma função chamada `valorImposto`. A função possui dois parâmetros formais: `taxaImposto`, que é a quantia de imposto sobre vendas expressa em porcentagem; e o `valor da venda`, que é o valor total dos itens vendidos. A função retorna o valor do imposto embutido na venda.
8. Faça um programa com uma função que retorne a quantidade de dígitos de um número inteiro digitado.
9. Faça um programa com uma função que retorne o reverso de um número inteiro informado. Por exemplo: 127 -> 721.
10. Faça um programa para imprimir:

```
1
1 2
1 2 3
....
1 2 3 ... n
```

para um n informado pelo usuário. Use uma função que receba um valor n inteiro e imprima até a n ésima linha.



Referências

- MENEZES, Nilo Ney Coutinho. **Introdução à Programação com Python: Algoritmos e lógica de programação para iniciantes**. 2 ed. 5 reimp. São Paulo: Novatec, 2017.