

Função

Objetivos de aprendizagem

- Explicar o significado e propósito de uma função
- Reconhecer a definição de uma função e de uma chamada de função
- Explicar o funcionamento de programa que usa a mesma função diversas vezes.
- Escolher bons dados para testar programas que incluam funções.
- Criar programa que define funções e que chama funções.

```
def mostrar_mensagem(nome_pessoa):  
    print(nome_pessoa, ", seja bem vinda!")  
    print("Aprenda o poder das funções")  
  
def main():  
    print("Olá, estudante")  
    mostrar_mensagem("Cris")  
  
main()
```

1. Considere o código do programa acima e responda:

a. Qual palavra-chave do Python indica que a linha de código se refere à definição de uma função?

A palavra-chave que indica que a linha uma função é o def.

b. Quais são duas assinaturas de função daquele código? Dica: Uma assinatura de função informa o nome da função e seus parâmetros.

As duas assinaturas são mostrar_mensagem(nome_pessoa) e main().

c. O nome da função consta na assinatura da função? Qual é o nome de cada uma das funções do programa? Sim consta, os nome de cada uma delas é mostrar_mensagem e main.

d. Execute o programa. O que ele produz de resultado?

O resultado que ele produz é “Olá, estudante” “Cris, seja bem vinda” “Aprenda o poder das funções”

e. Que comando você acrescentaria ao programa para mostrar as duas últimas linhas de texto do resultado novamente? Onde você acrescentaria esse comando no programa?

O comando que seria acrescentado é main().

2. Examine o seguinte programa.

```
import math  
  
def calcular_área(raio):  
    área = math.pi * raio ** 2  
    print("Área do círculo de raio %d é de %.2f" % (raio, área))  
    return área  
  
def main():  
    raio = int(input("Informe o raio do círculo: "))
```

```
calcular_área(raio)
```

```
main()
```

- a. Identifique no programa quais são as definições de função e quais são as chamadas de função.
- b. A chamada de função e a definição da função para *calcular_área* contêm um nome de variável entre parênteses. A variável na chamada de função é o que chamamos de **argumento**. A variável na definição da função é chamada de **parâmetro**. O que é o parâmetro na definição da função *calcular_área*? Qual é o seu propósito?

O parâmetro da função *calcular_área* é *raio*, e seu propósito identificar posteriormente um argumento para a função funcionar.

- c. Neste exemplo, o nome do parâmetro da definição de função e o argumento na chamada de função *calcular_área* possuem o mesmo nome. Isso é necessário?

Não é necessário, pois um argumento pode ser um variável de nome diferente do parâmetro.

- d. Execute o programa. Verifique sua resposta para a pergunta anterior (c), alterando o nome da variável *raio* para *número* na função *main*. Não altere o nome do parâmetro na definição da função *calcular_área*. O programa ainda funciona?

Sim ainda funciona.

- e. Adicione no programa uma linha de código que chama a função *calcular_área* e que usa o valor 6 como argumento. Execute o programa e veja se ele funciona corretamente.

- f. Adicione uma função ao programa para calcular e mostrar o diâmetro de um círculo, considerando o raio como parâmetro. Coloque esta definição de função antes da função *main*.

- g. Adicione uma linha de código para chamar a função criada no item anterior (cálculo do diâmetro de um círculo). Use o raio informado pelo usuário como argumento para esta chamada de função.

3. Examine o seguinte programa:

```
def somar_números(número1, número2):
    soma = número1 + número2
    print("%d + %d = %d" % (número1, número2, soma))
    return soma

def subtrair_números(número1, número2):
    subtração = número1 - número2
    print("%d - %d = %d" % (número1, número2, subtração))
    return subtração

def main():
    número1 = int(input("Informe um número inteiro entre 1 e 10: "))
    número2 = int(input("Informe um número inteiro entre 1 e 10: "))
    somar_números(número1, número2)
    subtrair_números(número1, número2)

main()
```

- a. Execute o programa com os dados abaixo. Mostre o resultado para cada conjunto de dados.

Conjunto de dados	Operando 1	Operando 2	Resultado
-------------------	------------	------------	-----------

1	2	6	8 e -4
2	3	8	11 e -5
3	34	1	35 e 33
4	-4	5	1 e -9

b. Quais problemas você observou quando você forneceu os conjuntos de dados 3 e 4?

4. Escreva uma função que desenhe um sapo. Chame a função para ver se funciona. Veja um exemplo de sapo abaixo.

```
@..@
(----)
( >__< )
^^  ~~  ^^
```

5. Altere (expanda) o programa do exercício anterior para produzir o seguinte resultado. Sugestão: altere a função para que ela tenha como argumento a mensagem a ser mostrada.

```
Um sapo...
    @..@
    (----)
    ( >__< )
    ^^  ~~  ^^
```

```
Dois sapos...
    @..@
    (----)
    ( >__< )
    ^^  ~~  ^^
```

```
Três sapos...
    @..@
    (----)
    ( >__< )
    ^^  ~~  ^^
```

```
Quatro sapos...
    @..@
    (----)
    ( >__< )
    ^^  ~~  ^^
```