



# Engenharia de Software

## Computacional thinking with Python

Prof. Dr. Francisco Elânio

# Funções em Python

Mais cedo ou mais tarde você descobrirá que a base do seu programa está aumentando e, quando aumenta, a dificuldade do gerenciamento aumenta rapidamente.

O que começou com 20 linhas de código em python, aumentou para 500 linhas ou mais.

Quando isso acontece é hora de começar a pensar em quais estratégias você poderá usar para reduzir a complexidade da sua base de código.

# O que é uma Função?

As funções são definidas por relacionar constantes e variáveis para descrever fenômenos naturais e tecnológicos, estudadas em diversas áreas do conhecimento.

Se uma variável  $y$  depende de uma variável  $x$  de tal modo que cada valor de  $x$  determina exatamente um valor de  $y$ , dizemos que  $y$  é uma função de  $x$ .

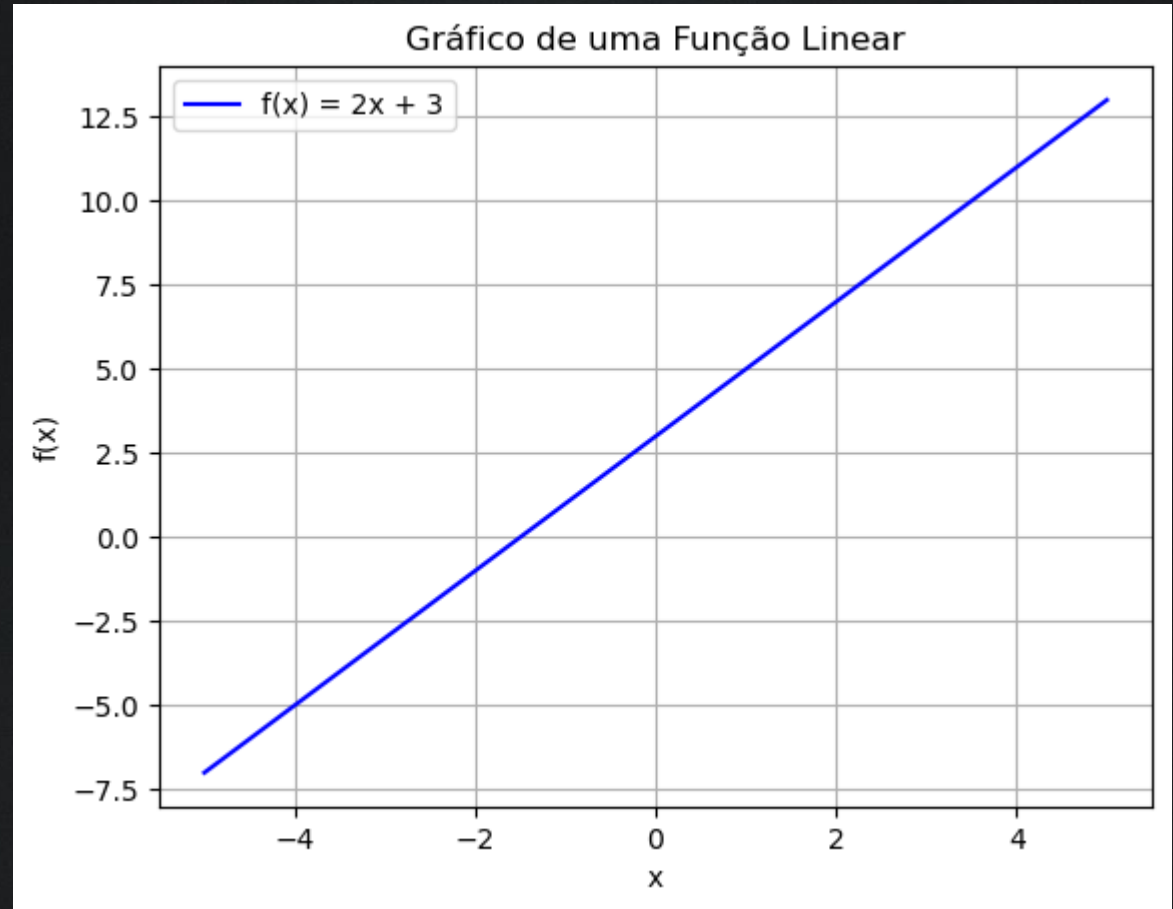
Uma função  $f$  é uma regra que associa uma única saída a cada entrada. Se a entrada for denotada por  $x$ , então a saída é denotada por  $f(x)$ .

# Função Linear

$$f(x) = ax + b$$

Exemplo:

$$f(x) = 2x + 3$$





# Função Quadrática

$$f(x) = ax^2 + bx + c$$

Exemplo:

$$f(x) = x^2 - 4x + 3$$

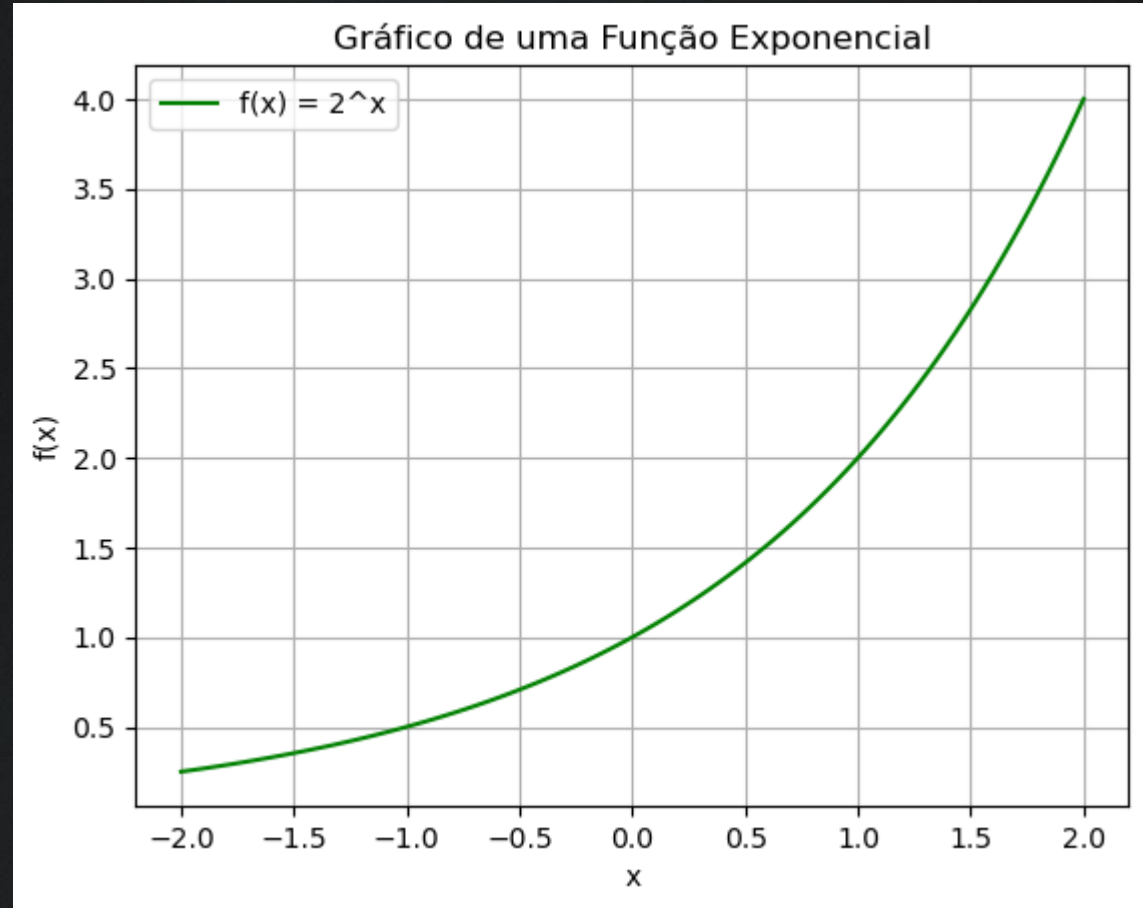


# Função Exponencial

$$f(x) = a^x$$

Exemplo:

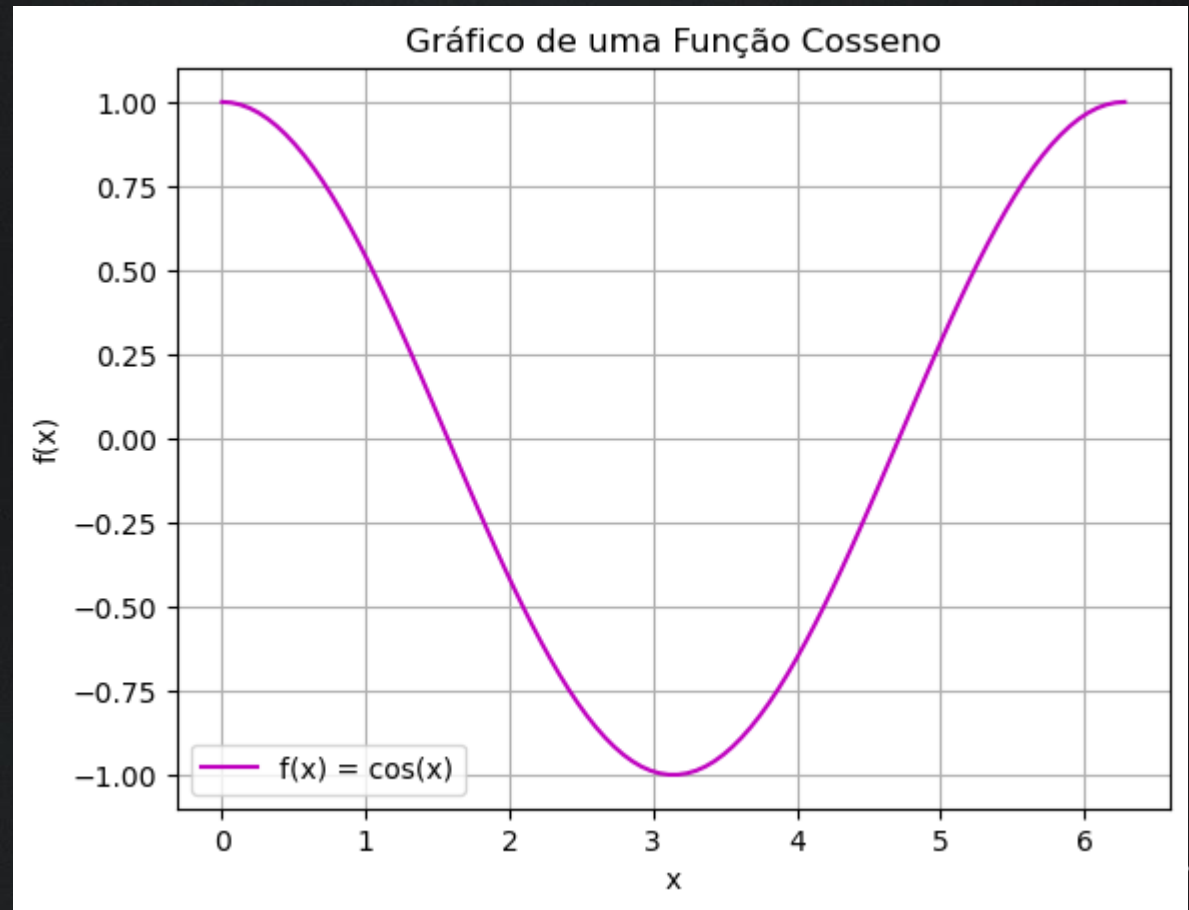
$$f(x) = 2^x$$



# Função Cosseno

Exemplo:

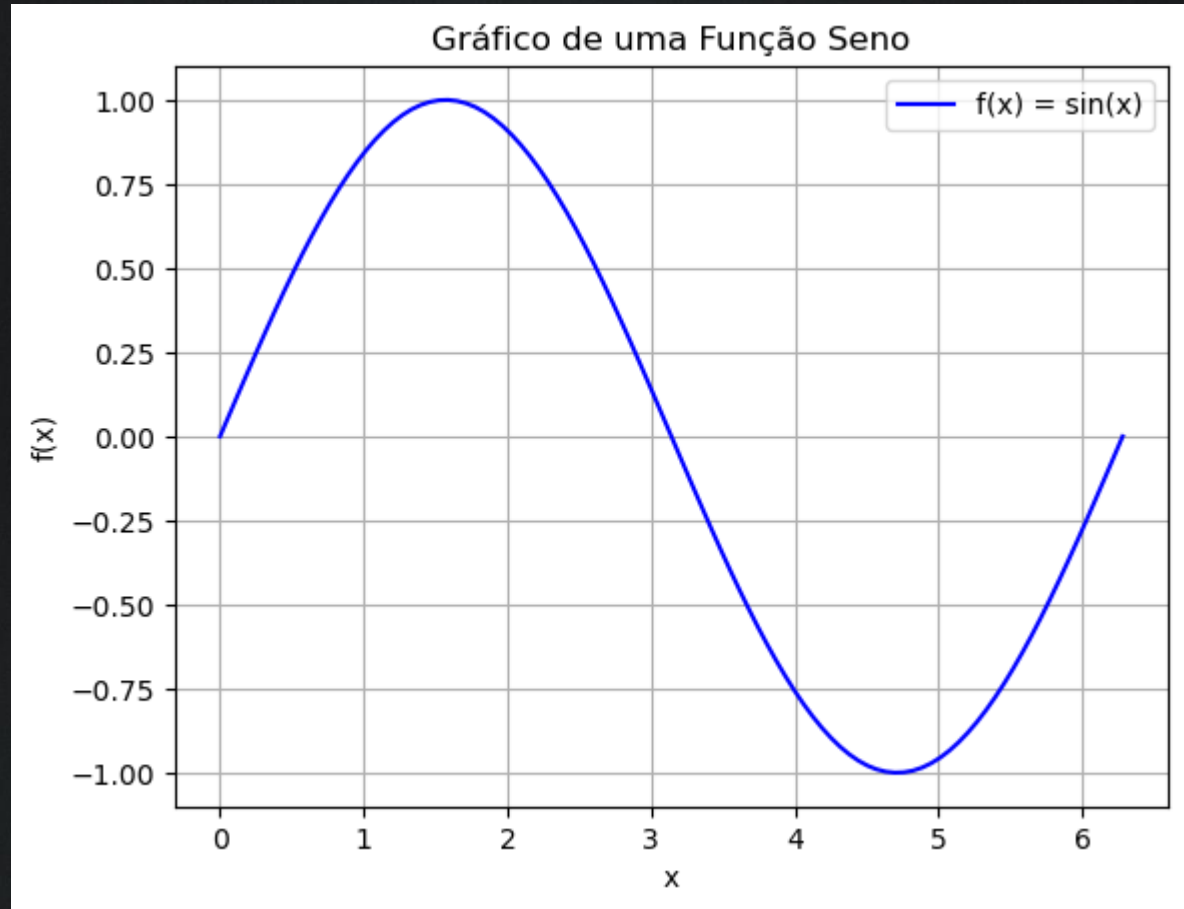
$$f(x) = \cos(x)$$



# Função Seno

Exemplo:

$$f(x) = \text{sen}(x)$$





# Anatomia das Funções

## 1. As funções introduzem duas palavras-chave novas def e return:

def

return

A palavra-chave def nomeia a função e detalha os argumentos que a função pode ter. Enquanto return é opcional, e ela é usada para passar de volta um valor para o código que chamou a função.

## 2. As funções podem aceitar dados de argumento

Uma função pode aceitar os dados de argumento (entrada para uma função).

## 3. As funções tem código e documentação

O código é recuado um nível abaixo da linha def e deve incluir comentários onde fazem sentido.

# Anatomia das Funções

A linha def nomeia a função e lista qualquer argumento

```
def descrição do nome(argumento opcional):
```

```
    """Documentação."""
```

```
    Seu código para função vai aqui!
```

```
    Seu código para função vai aqui!
```

```
    Seu código para função vai aqui!
```

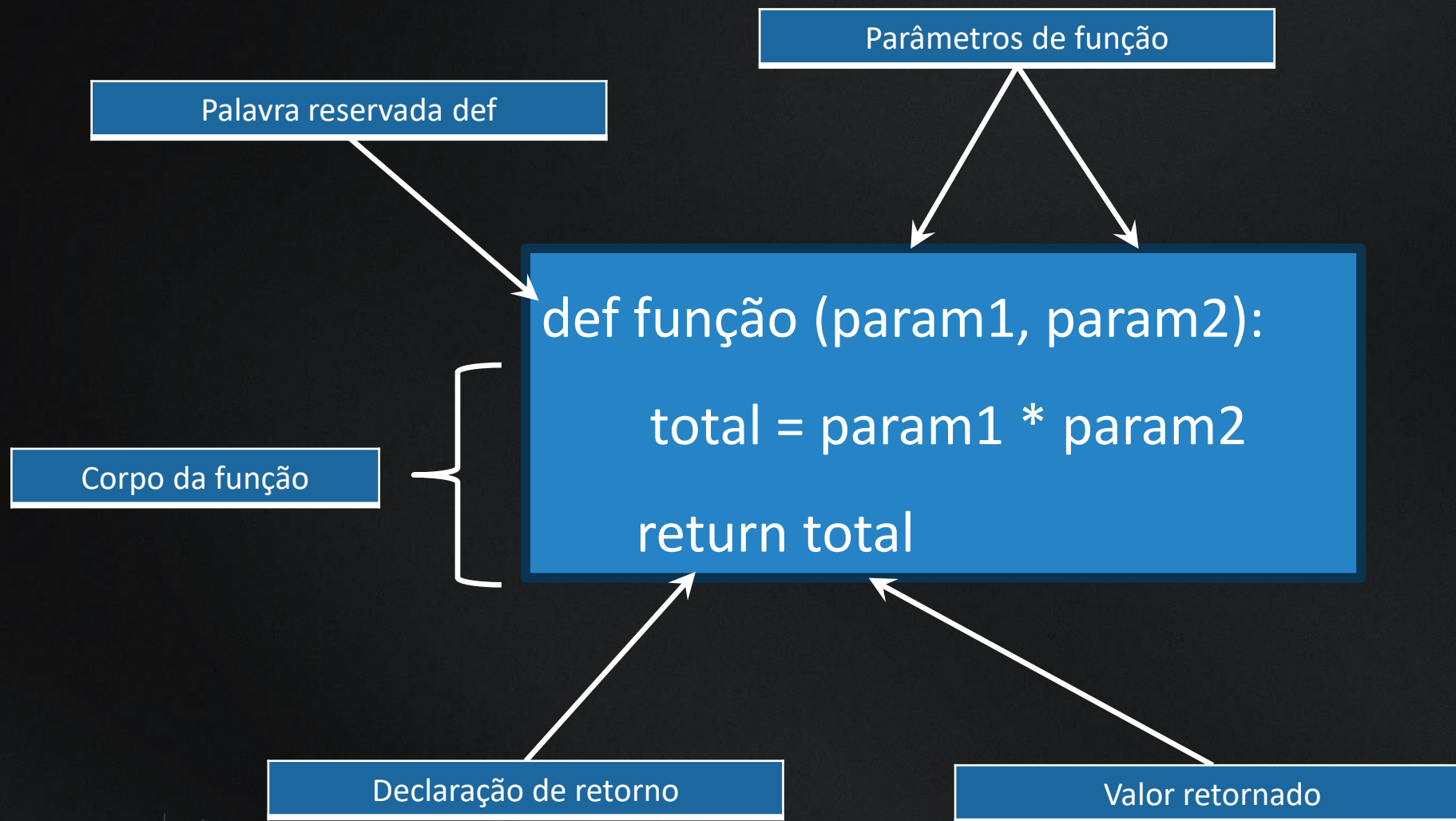
```
    return valor
```

Modelo de função útil

Descreve a finalidade da função

O código fica nesta parte

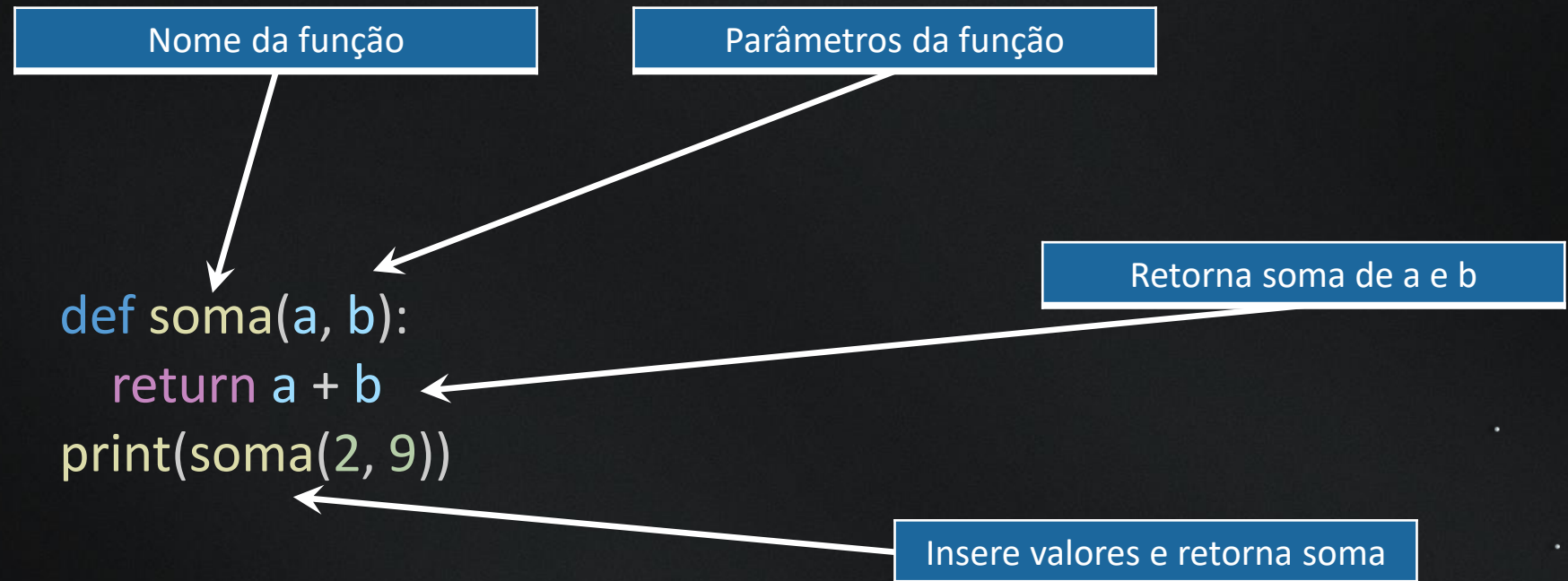
# Anatomia das Funções



# Retornando uma função em Python

Para retornar uma função em Python é necessário inserir o nome da função.

Para o exemplo abaixo, soma() e dois valores entre o parêntese.





# Exemplos da Função def em Python

Exemplos para declarar uma função de soma, subtração, multiplicação e divisão que recebe dois números como parâmetros e o imprime na tela.

```
def soma(a, b):  
    return a + b  
print(soma(2, 9))
```

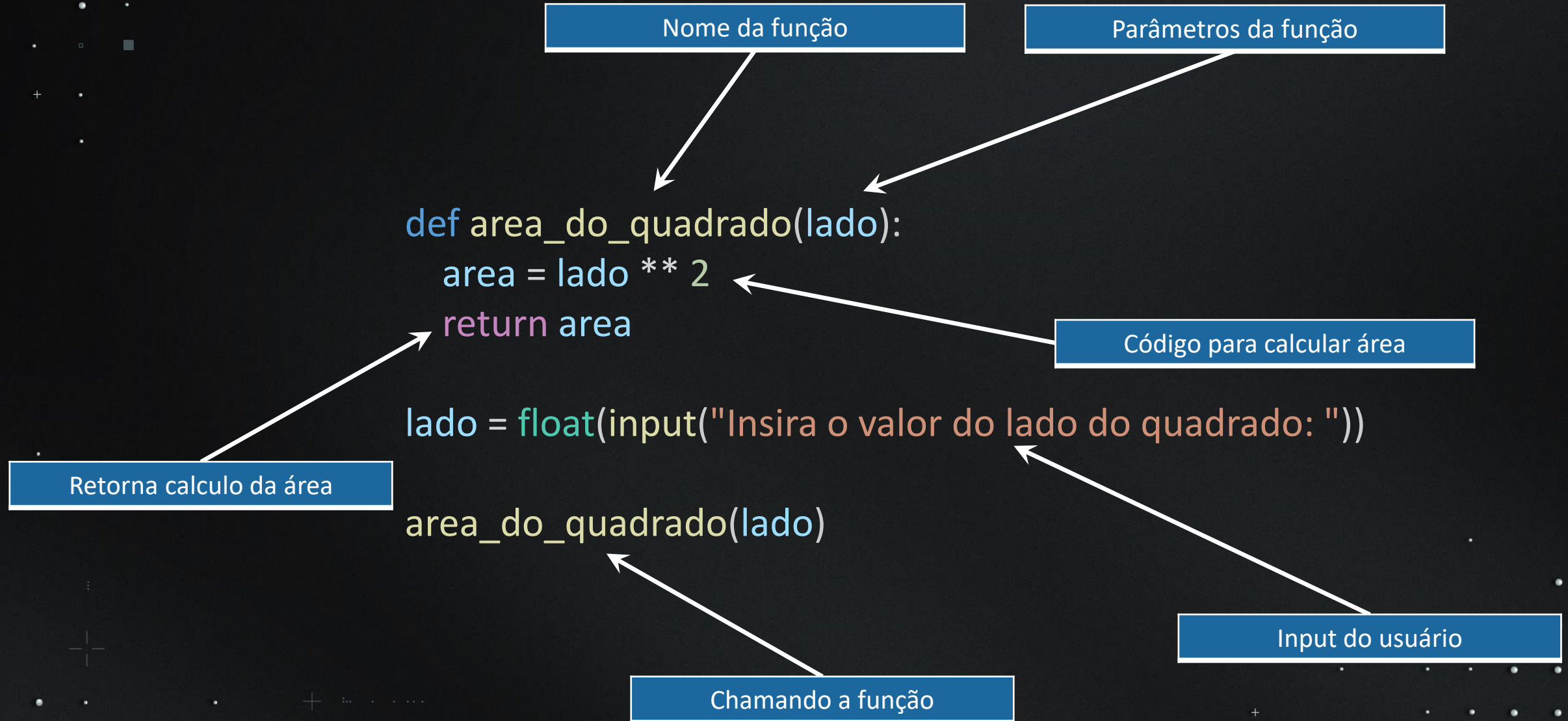
```
def soma(a, b):  
    return a - b  
print(soma(5, 2))
```

```
def soma(a, b):  
    return a * b  
print(soma(3, 4))
```

```
def soma(a, b):  
    return a / b  
print(soma(24, 8))
```



# Criando função para área de um quadrado



# Exercícios

1. Crie uma função para calcular a força em Newton de acordo com os valores de entrada da massa e aceleração inseridos pelo usuário.
2. Crie uma função para calcular o volume de um cubo com os valores de largura, altura e comprimento que devem ser inseridos pelo usuário.
3. Crie uma função para calcular a área de uma sala. O usuário deve inserir as dimensões em metros.

# 1. Calcula a força em Newton

```
def calcular_forca(massa, aceleracao):  
    forca = massa * aceleracao  
    return forca
```

```
massa = float(input("Insira a massa (em kg): "))  
aceleracao = float(input("Insira a aceleração (em m/s^2): "))
```

```
forca_resultante = calcular_forca(massa, aceleracao)
```

```
print("A força resultante é de", forca_resultante, "Newton (N)")
```

## 2. Crie uma função para calcular volume de um cubo

```
def volume_do_cubo(comprimento, largura, altura):  
    volume = comprimento * largura * altura  
    return volume
```

```
comprimento = float(input("Insira o comprimento do cubo: "))  
largura = float(input("Insira a largura do cubo: "))  
altura = float(input("Insira a altura do cubo: "))
```

```
volume = volume_do_cubo(comprimento, largura, altura)
```

```
print("O volume do cubo com comprimento", comprimento, "largura", largura, "e altura",  
      altura, "é", volume)
```



### 3. Crie uma função para calcular a área de uma sala

```
def area_da_sala():  
    comprimento = float(input("Insira o comprimento da sala (em metros): "))  
    largura = float(input("Insira a largura da sala (em metros): "))  
    area = comprimento * largura  
    return area
```

```
area_da_sala = area_da_sala()  
print("A área da sala é", area_da_sala, "metros quadrados")
```



*A sabedoria preserva a  
vida de quem a possui.*

# Referências

ASCENCIO, A. F. G, CAMPOS, E. A. V. Fundamentos da Programação de Computadores: algoritmos, Pascal, C/C++ e Java, 2ª Edição, São Paulo: Pearson 2007.

SALVETTI, Dirceu Douglas; BARBOSA, Lisbete Madson. Algoritmos. São Paulo: Pearson, 2004.

LACEY, Nichola. Python by Example: Learning program in 150 challenges. University Printing House, Cambridge CB2 8BS, United Kingdom, 2019.