

# Funções em Python

## Índice

[Introdução](#)

[Definição de Funções](#)

[Chamando Funções](#)

[Parâmetros e Argumentos](#)

[Valor de Retorno](#)

[Escopo das Funções](#)

[Funções Lambda](#)

## Explicações Adicionais

- **Introdução:** Explica o conceito básico de funções e seu uso em Python.
- **Definição de Funções:** Detalha como criar funções em Python.
- **Chamando Funções:** Mostra como chamar funções definidas.
- **Parâmetros e Argumentos:** Explica como passar parâmetros para funções e como os argumentos são usados.
- **Valor de Retorno:** Discute como as funções podem retornar valores e como usar esses valores.
- **Escopo das Funções:** Explica o escopo das variáveis dentro das funções e como elas interagem com variáveis globais.
- **Funções Lambda:** Introduz funções lambda e como usá-las.

## Introdução

Funções em Python são blocos de código reutilizáveis que realizam uma tarefa específica. Elas ajudam a modularizar o código, tornando-o mais organizado e reutilizável.

## Definição de Funções

Para definir uma função em Python, você usa a palavra-chave `def` seguida pelo nome da função e parênteses. O corpo da função é definido por um bloco indentado.

```
# Código
def saudacao(nome):
    """Exibe uma saudação personalizada."""
    print(f"Olá, {nome}!")

# Exemplo de chamada da função
saudacao("Maria") # Saída: Olá, Maria!
```

## Chamando Funções

Após definir uma função, você pode chamá-la em qualquer lugar do seu código. Para chamar uma função, use seu nome seguido por parênteses. Se a função requer parâmetros, passe-os dentro dos parênteses.

```
# Código
def somar(a, b):
    """Retorna a soma de dois números."""
    return a + b

resultado = somar(5, 3)
print(resultado) # Saída: 8
```

## Parâmetros e Argumentos

Parâmetros são variáveis listadas entre parênteses na definição da função. Argumentos são os valores que você passa para esses parâmetros quando chama a função.

```
# Código
def saudacao(nome, idade):
    """Exibe uma saudação com nome e idade."""
    print(f"Olá, {nome}! Você tem {idade} anos.")

saudacao("Pedro", 25) # Saída: Olá, Pedro! Você tem 25 anos.
```

## Valor de Retorno

Funções podem retornar valores usando a palavra-chave `return`. Se uma função não tiver uma instrução `return`, ela retorna `None` por padrão.

```
# Código
def multiplicar(x, y):
    """Retorna o produto de dois números."""
    return x * y

produto = multiplicar(4, 5)
print(produto) # Saída: 20
```

## Escopo das Funções

Dentro de uma função, variáveis definidas são locais e só podem ser acessadas dentro dessa função. No entanto, uma função pode acessar variáveis globais definidas fora dela.

```
# Código
variavel_global = "Eu sou global"

def mostrar_variavel():
    variavel_local = "Eu sou local"
    print(variavel_global) # Acessa variável global
    print(variavel_local) # Acessa variável local

mostrar_variavel()
print(variavel_global) # Acessa variável global
# print(variavel_local) # Gera um erro, pois 'variavel_local' não está definida fora da

# Saída
# Eu sou global
# Eu sou local
# Eu sou global
```

## Funções Lambda

Funções lambda são pequenas funções anônimas definidas com a palavra-chave `lambda`. Elas são usadas para criar funções de uma única linha e são úteis em casos onde uma função simples é necessária temporariamente.

```
# Código
# Função lambda para somar dois números
soma = lambda x, y: x + y

print(soma(10, 5))  # Saída: 15
```