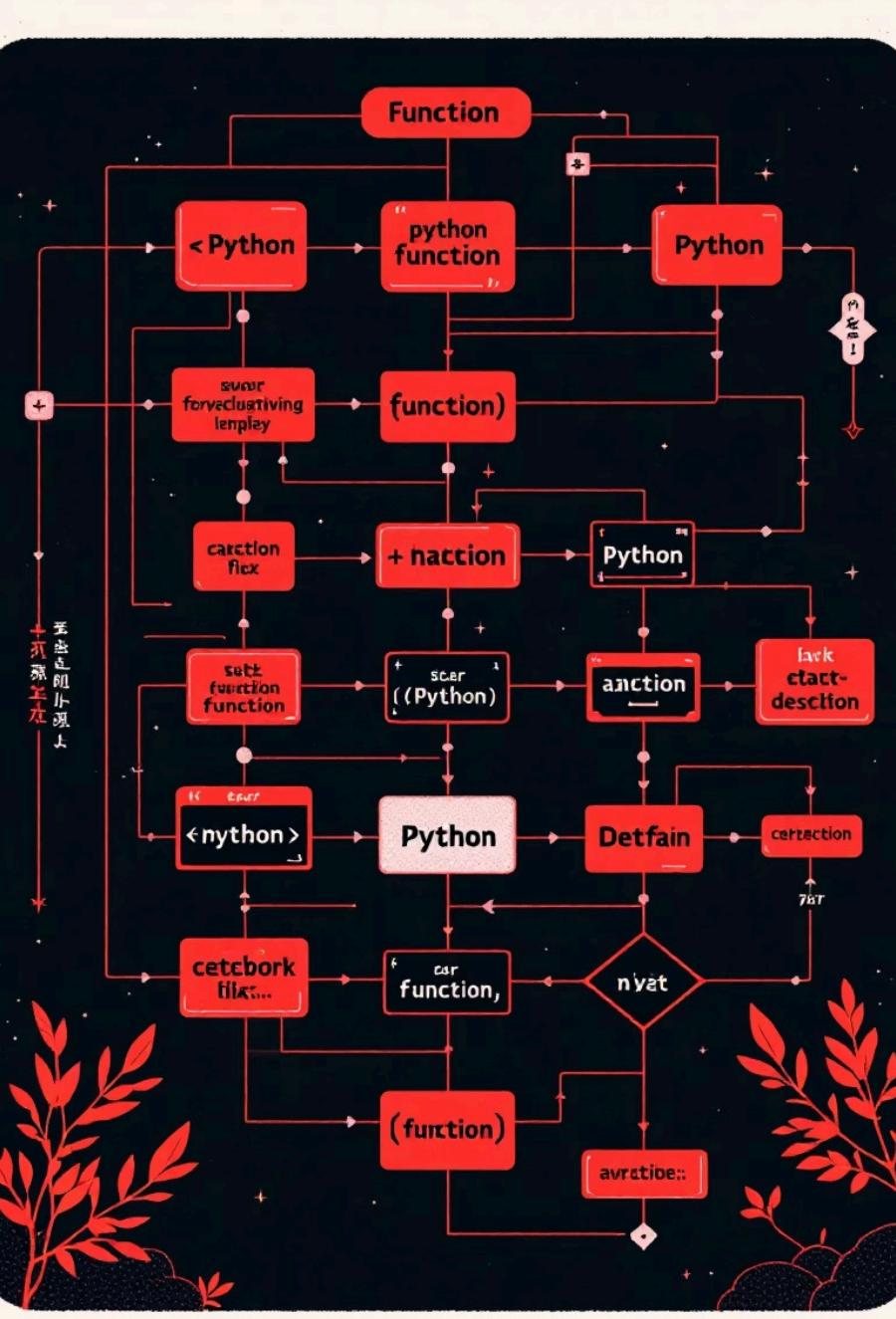


# Aula 04 - Funções I: Conceito, Aplicação e Escopos

Aprenda a criar funções reutilizáveis e organize melhor seu código Python





## Objetivo da Aula

Introduzir a criação de funções, promovendo a reutilização eficiente de código e compreensão dos escopos de variáveis.

Você aprenderá sobre parâmetros, argumentos e como retornar valores de dentro das funções.

Ao final desta aula, você será capaz de criar funções personalizadas que tornarão seu código mais limpo e organizadamente estruturado.

# O que é uma Função?



## Bloco Reutilizável

Um conjunto de código que executa uma tarefa específica e pode ser usado várias vezes



## Organização

Ajuda a estruturar, reutilizar e simplificar a lógica do programa



## Retorno de Valores

Pode devolver informações processadas através do **return**

```
11 python :  
11 def sabit function, and teep()  
2 | athon of the frist)  
13 | -+ candions,trat" > with ():  
30 | < a Conricts, "Def), >=> , no;  
15 deattting();  
4) | <`= outhon ( function of function)">  
47 | wuthon is scar(s):  
18 | -the dontrins, action:  
13 LINI:  
14 hetr = i/ dop;
```

### ⓘ Definindo Funções com **def**

```
def nome_da_funcao(parâmetros):  
    # bloco de código  
    return variavel_a_ser_retornada
```

# Criando Funções

## Sintaxe Clara

Use `def` seguido de um nome descritivo

## Nome Descritivo

Escolha nomes que expliquem claramente o propósito da função

## Chamada da Função

Lembre-se sempre de `chamar` sua função para executá-la

### Funções sem parâmetro

```
def saudacao():
    print("Olá, bem-vindo!")
```

### Funções com parâmetro

```
def saudacao_personalizada(nome):
    print(f"Olá, {nome}! Seja bem-vindo!")
```

② **Atividade Prática:** Crie uma função que receba dois números e mostre a soma deles quando a função for chamada.

```
fir paraiammer, (e:  
    print 1y  
        function { funcion {= i6 daraumers } })  
        <>> air lobj ;  
    } co lobj;
```

# Argumentos e Parâmetros

1

## Parâmetros

Variáveis declaradas na **definição** da função

2

## Argumentos

Valores reais passados quando a função é **chamada**

3

## Retorno

Use **return** para devolver valores processados

```
def soma(a, b): # a e b são parâmetros  
    return a + b  
  
print(soma(3, 5)) # 3 e 5 são argumentos
```

- ⌚ **Prática:** Crie uma função que faça uma saudação personalizada com nome e idade do usuário.

# Escopo de Variáveis



## Escopo Local

Variável criada **dentro** da função só existe naquele contexto



## Escopo Global

Variável criada **fora** pode ser usada em todo o programa

```
x = 10 # variável global
```

```
def teste():
    y = 5 # variável local
    print(x + y) # x é acessível, y apenas aqui
```

- ✖ **Dica Didática:** Atenção a erros comuns como ausência de return ou uso inadequado de variáveis globais.

# Boas Práticas com Funções



## Nomes Descritivos

Use nomes claros para funções e parâmetros que expliquem sua finalidade



## Funções Concisas

Mantenha funções curtas e objetivas, focando em uma tarefa específica



## Evite Repetição

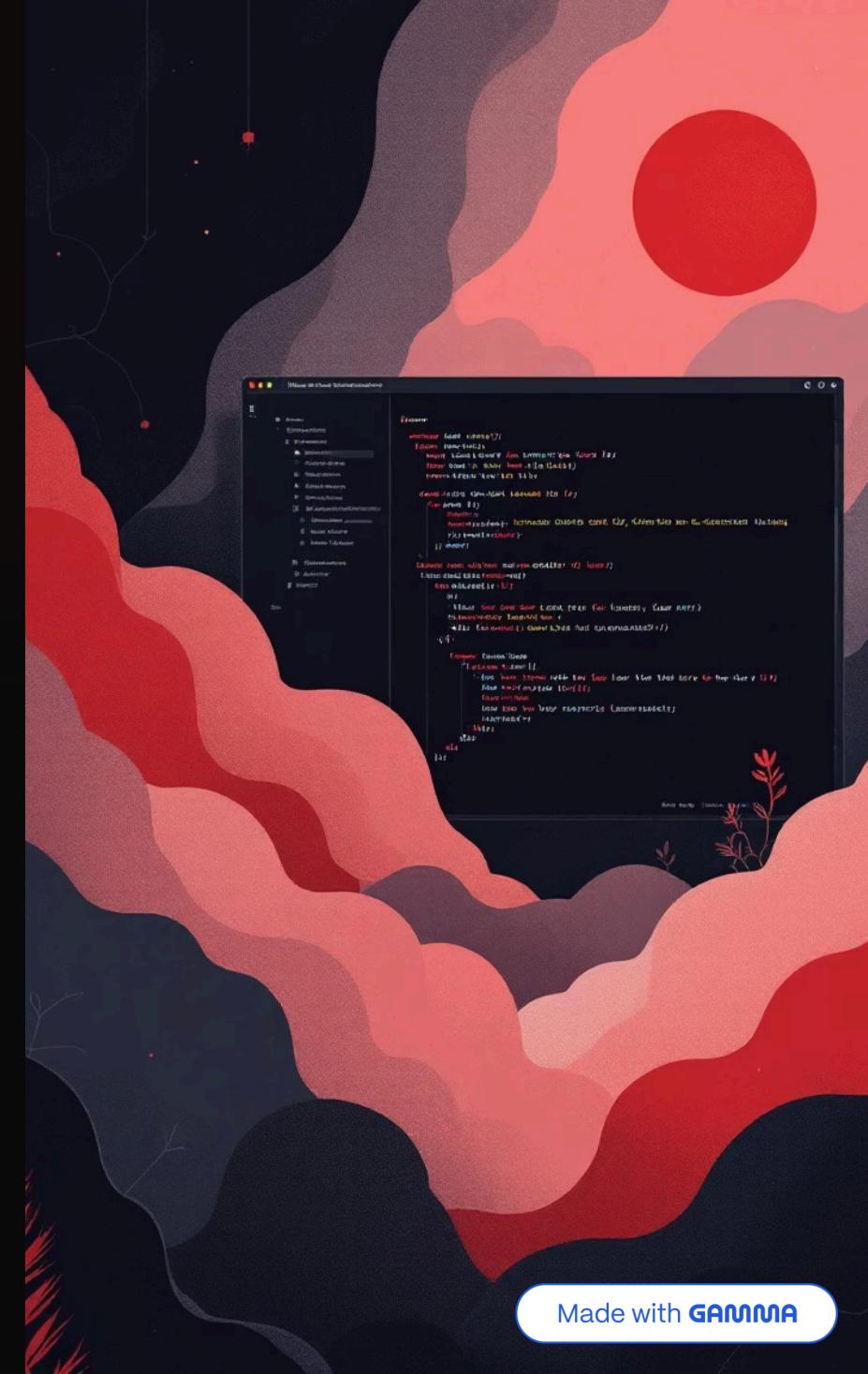
Elimine código duplicado criando funções reutilizáveis



## Use Return

Utilize `return` adequadamente para devolver valores processados

- ② **Prática:** Crie uma função que receba um número e retorne o dobro dele. Salve esta informação em uma variável global.



# Funções Lambda



## Introdução às Funções Lambda

Funções **anônimas** ideais para operações simples e expressões rápidas



### Sintaxe Simples

Uma linha de código para operações básicas



### Uso Prático

Perfeitas para cálculos rápidos e transformações simples

```
quadrado = lambda x: x**2  
print(quadrado(4)) # Resultado: 16
```

**Exemplo prático:** Use lambda para calcular o **dobro** de qualquer número de forma rápida e eficiente.

# Prática e Exercícios



## Funções Matemáticas

Crie funções para soma, subtração, multiplicação e divisão



## Saudação Personalizada

Função que recebe nome e idade, retornando mensagem customizada



## Calculadora de IMC

Calcule o Índice de Massa Corporal e classifique a situação



## Lambda Rápidas

Use funções lambda para operações matemáticas de soma, subtração, divisão e multiplicação.

**⚠ Desafio Final:** Criar uma função que calcule a média de 3 notas e retorne se o aluno está aprovado ou reprovado (média  $\geq 7.0$ ).

