

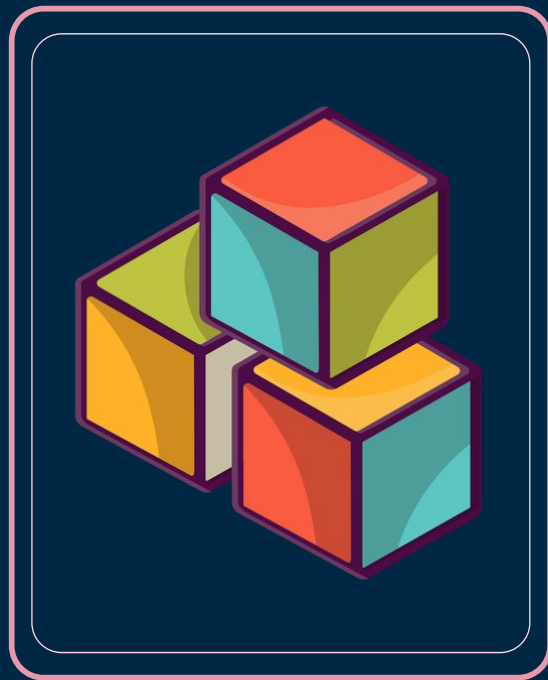
RACIOCÍNIO ALGORÍTMICO

Funções


```
def aula6(data):
```

FUNÇÕES

Em programação, uma função é um bloco de código nomeado que pode receber um conjunto de valores de entrada, realizar um conjunto de instruções e retornar um valor de saída opcional. As funções são usadas para dividir um programa em partes menores e independentes, o que facilita o desenvolvimento, a organização e a reutilização de código.




BENEFÍCIOS NA UTILIZAÇÃO



Dividir o programa em partes menores e mais gerenciáveis.

Modularidade




As alterações e correções de bugs se tornam mais fáceis.

Manutenção simplificada


Uma vez definida, pode ser chamada em qualquer parte do programa.

Reutilização de código



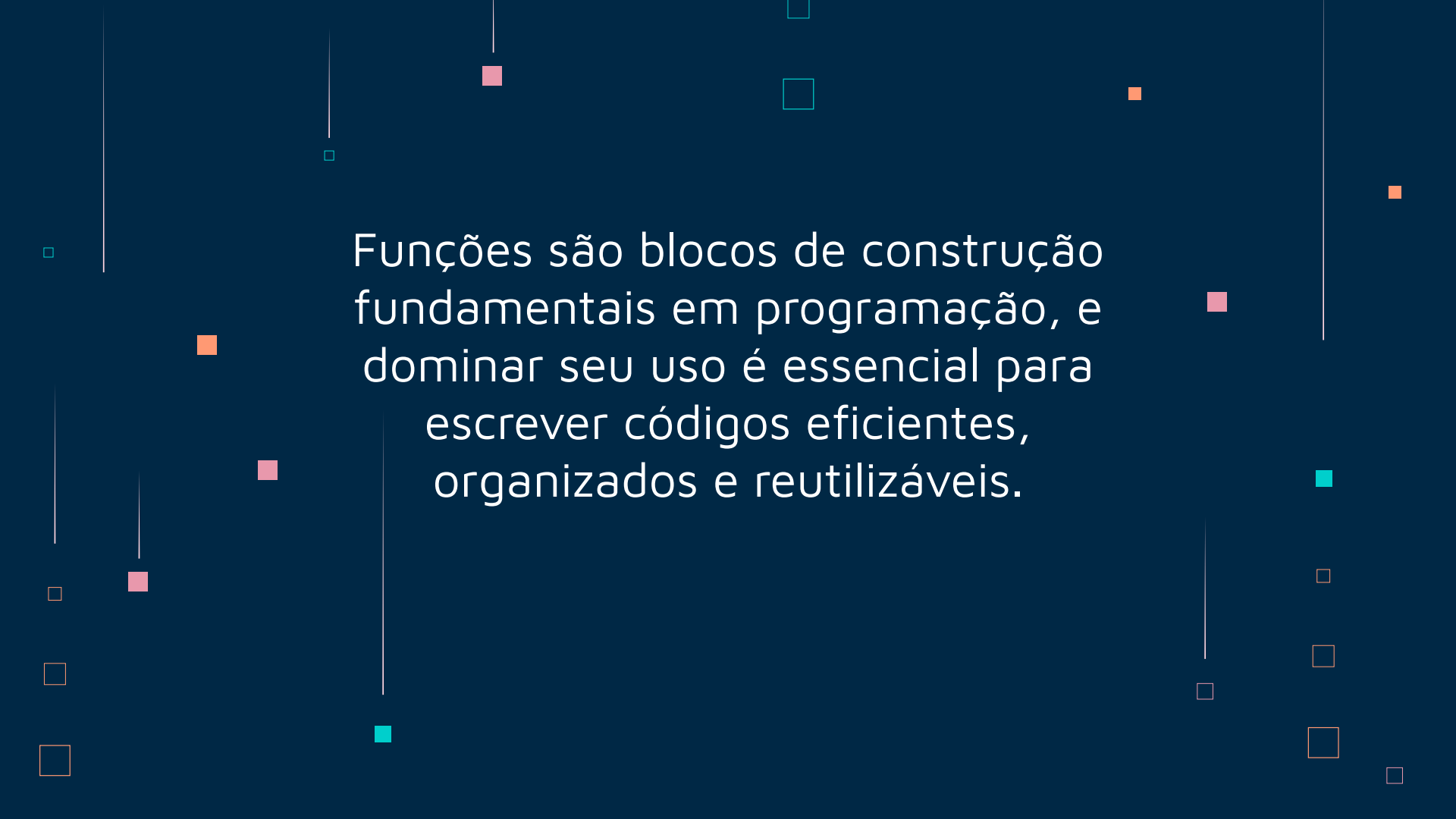
Permite abstrair detalhes complicados em um único bloco de código.

Abstração



Possuem nomes significativos para diferentes partes do código.

Legibilidade e clareza

The background is a dark blue gradient. It is decorated with various geometric elements: thin white vertical lines, small squares in teal, orange, and pink, and larger squares with thin orange outlines. These elements are scattered across the slide, some aligned with the text and others floating independently.

Funções são blocos de construção fundamentais em programação, e dominar seu uso é essencial para escrever códigos eficientes, organizados e reutilizáveis.

Palavra chave para **definir** uma função

```
def nome_da_funcao(parâmetros):  
    # CÓDIGO AQUI  
    return algum_valor
```

Nome qualquer (**intuitivo**) que representa o que a função executa

```
def nome_da_funcao(parâmetros):  
    # CÓDIGO AQUI  
    return algum_valor
```

Qualquer valor(es) necessário(s) para executar o código

```
def nome_da_funcao(parâmetros):  
    # CÓDIGO AQUI  
    return algum_valor
```

```
def nome_da_funcao(parâmetros):  
    # CÓDIGO AQUI  
    return algum_valor
```

Implementação: código que será executado por essa função


```
def nome_da_funcao(parâmetros):  
    # CÓDIGO AQUI  
    return algum_valor
```

Qualquer tipo de dado que será retornado
para onde a função foi chamada

The background is a dark blue gradient. It features several vertical white lines of varying lengths. Scattered throughout are small squares in light blue, pink, orange, and teal. Some squares are solid, while others are outlined.

Parâmetros e Retorno SÃO OPCIONAIS

```
def nome_da_funcao(parâmetros):  
    # CÓDIGO AQUI  
    return algum_valor
```

DEFINIÇÃO

```
nome_da_funcao(parâmetros)
```

CHAMADA

Vamos pegar o seguinte trecho de código:

```
x = 4  
y = 2  
soma = x + y  
  
print(soma)
```

Output:

6



1. Função sem parâmetros e sem retorno:

```
def soma():  
    x = 4  
    y = 2  
    soma = x + y  
    print(soma)
```

```
soma()
```

Output:

6



1. Função com parâmetros e sem retorno:

```
def soma(x, y):  
    soma = x + y  
    print(soma)  
  
soma(4, 2)
```

Output:

6



1. Função com parâmetros e com retorno:

```
def soma(x, y):  
    return x + y  
  
print(soma(4, 2))
```

Output:

6



EXERCÍCIOS

1. Escreva uma função chamada "imprimir_nome" que imprime o seu nome.
2. Escreva uma função chamada "maior" que receba três números como parâmetros e retorne o maior entre eles.
3. Escreva uma função chamada "criar_vetor" que retorna um vetor preenchido com zeros de tamanho 5.
4. Escreva uma função chamada "media" que receba uma lista de números como parâmetro e retorne a média desses números.
5. Escreva uma função chamada "inverter" que receba uma string como parâmetro e imprime a string invertida.