

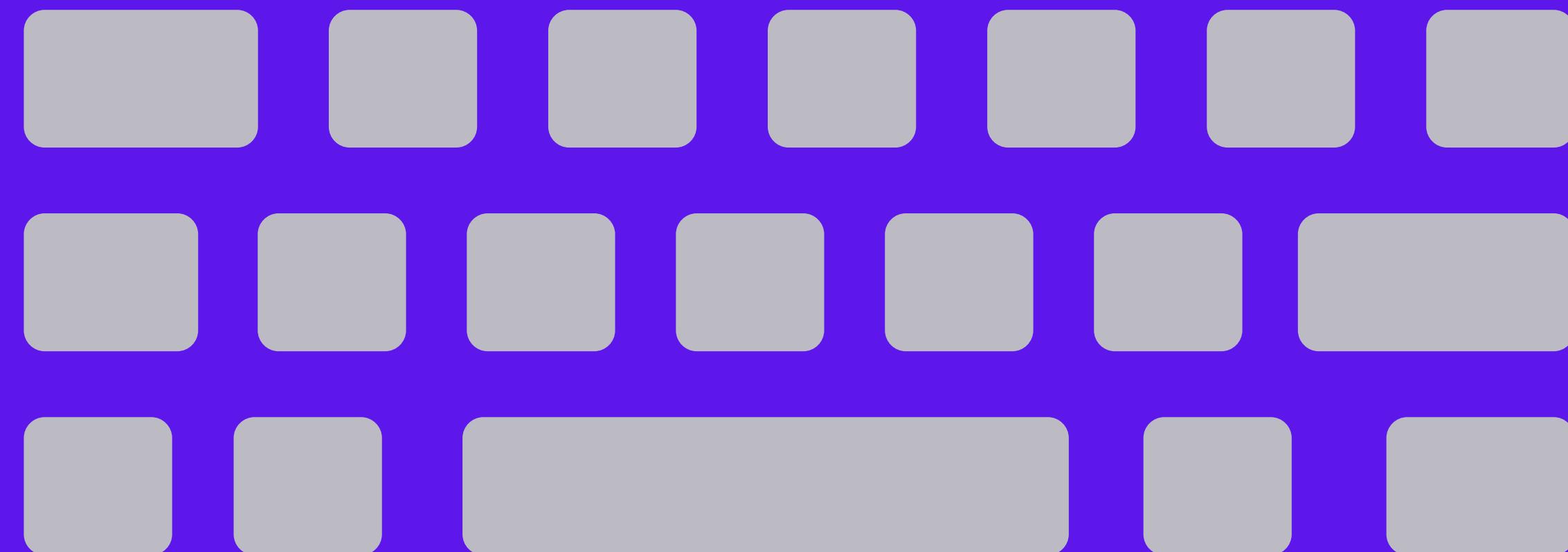


PROGRAMAÇÃO 101

dia 3

Comandos de Repetição

O que são Comandos de Repetição?



Definição e Importância

- **Definição:**
 - Comandos de repetição, ou loops, permitem que um bloco de código seja executado repetidamente até que uma condição seja atendida.
- **Necessidade:**
 - Automatizar tarefas repetitivas, processar coleções de dados e simplificar o código.

- **for Loop:**
 - Itera sobre uma sequência (lista, tupla, string) ou outros objetos iteráveis. (que podemos percorrer)
- **while Loop:**
 - Executa um bloco de código enquanto uma condição é verdadeira.

Comando for

- Iteração sobre uma Sequência:
 - Utilizado para percorrer elementos em uma sequência.

```
for numero in [1, 2, 3, 4, 5]:  
    print(numero)
```



Comando while

- Execução Enquanto uma Condição é Verdadeira:
 - Repetição baseada em uma condição lógica.

```
contador = 0
while contador < 5:
    print(contador)
    contador += 1
```



Controle de Loop

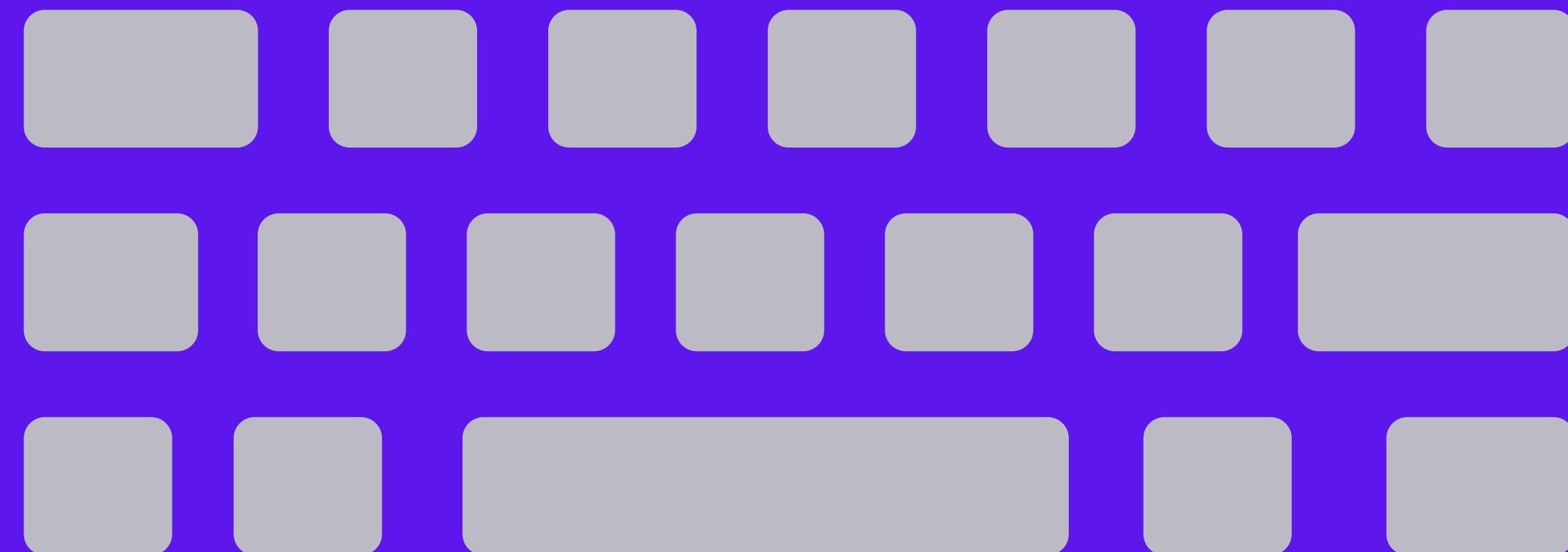
- **break:**
 - Encerra imediatamente o loop.
- **continue:**
 - Pula para a próxima iteração do loop.



dia 3

Funções

O que são Funções em Python?



Definição e Importância

- **Definição:**

- Funções são blocos de código reutilizáveis, projetados para realizar uma tarefa específica. Elas são executadas somente quando chamadas.

```
def saudacao(nome):  
    return "Olá, " + nome + "!"
```



Definição e Importância

Funções evitam repetições desnecessárias!

Sem função:

```
print("\nParabéns pra você!")
print("Nesta data querida!")
print("Muitas felicidades!")
print("Muitos anos de vida!")
print("\nParabéns pra você!")
print("Nesta data querida!")
print("Muitas felicidades!")
print("Muitos anos de vida!")
print("\nParabéns pra você!")
print("Nesta data querida!")
print("Muitas felicidades!")
print("Muitos anos de vida!")
```

Com função:

```
def parabens():
    print("\nParabens pra voce!")
    print("Nesta data querida!")
    print("Muitas felicidades!")
    print("Muitos anos de vida!")

parabens()
parabens()
parabens()
```



Estrutura Básica de uma Função

- **Declaração com def:**
 - Define o nome e os parâmetros da função.
- **Corpo da Função:**
 - Contém as instruções executadas quando a função é chamada.
- **InSTRUÇÃO return:**
 - Retorna um valor da função.
- **Acesso a variáveis:**
 - Variáveis declaradas dentro da função não podem ser acessadas de fora dela.



Chamada de Função

- **Passagem de Argumentos:**
 - Argumentos são valores fornecidos à função durante a chamada.

```
mensagem = saudacao("Alice")
```



Parâmetros e Argumentos

- **Parâmetros:**
 - Variáveis na definição da função.
- **Argumentos:**
 - Valores passados para a função durante a chamada.

```
def adicao(a, b):  
    return a + b  
  
resultado = adicao(3, 5)
```



Funções com Valor Padrão

- **Atribuição de Valor Padrão aos Parâmetros:**
 - Permite que a função seja chamada sem fornecer argumentos para esses parâmetros.

```
def saudacao_personalizada(nome, saudacao="Olá"):  
    return saudacao + ", " + nome + "!"
```



PROGRAMAÇÃO 101