Sumário

[1.Python 2](#_Toc151194224)

[1.1 Propriedades 2](#_Toc151194225)

[1.1.1 Funções 2](#_Toc151194226)

[1.2 Strings 2](#_Toc151194227)

[1.3 Lista [ ] 2](#_Toc151194228)

[1.4 Tupla ( ) 2](#_Toc151194229)

[1.5 IF - ELIF – ELSE 2](#_Toc151194230)

[1.6 While 2](#_Toc151194231)

[1.7 For 2](#_Toc151194232)

[1.8 Funções 2](#_Toc151194233)

[1.8.1 Definir uma Função 2](#_Toc151194234)

[1.8.2 Escopo da Função 2](#_Toc151194235)

[1.8.3 return 3](#_Toc151194236)

[1.8.4 \*args (Argumentos não nomeados) 4](#_Toc151194237)

[2. SQL 5](#_Toc151194238)

[2.1 Propriedades 5](#_Toc151194239)

[2.1.1 Tipos Primitivos 5](#_Toc151194240)

[2.2 Create Database 6](#_Toc151194241)

[2.3 Create Table 6](#_Toc151194242)

[2.4 Insert Into 7](#_Toc151194243)

[2.5 Alter Table 8](#_Toc151194244)

[2.5.1 Renomear tabela 8](#_Toc151194245)

[2.5.2 Adicionar uma coluna na tabela 8](#_Toc151194246)

[2.6 Select 9](#_Toc151194247)

# 1.Python

## 1.1 Propriedades

### 1.1.1 Funções

## 1.2 Strings

## 1.3 Lista [ ]

## 1.4 Tupla ( )

## 1.5 IF - ELIF – ELSE

## 1.6 While

## 1.7 For

## 1.8 Funções

### 1.8.1 Definir uma Função

### 1.8.2 Escopo da Função

Escopo significa o local onde aquele código pode atingir.

Existe o escopo global e local.

O escopo global é o escopo onde todo o código é alcançável.

O escopo local é o escopo onde apenas nomes do mesmo local

podem ser alcançados.

x = 1

def escopo1():

    x = 2

    def escopo2():

        x = 3

        print(x)

    print(x)

    escopo2()

print(x)

escopo1()

print(x)

Saída

1

2

3

1

O valor da variável fora da função não altera o valor da variável dentro da função, e vice-versa. É possível alterar variáveis de escopo externo em escopo interno usando a propriedade global.

def escopo1():

    global x

    x = 2

    def escopo2():

        global x

        x = 3

        print(x)

    print(x)

    escopo2()

print(x)

escopo1()

print(x)

Saída

1

2

3

3

### 1.8.3 return

Retorna os valores das funções

def soma(*x*, *y*):

    if *x* > 10:

        return [10, 20]

    return *x* + *y*

soma1 = soma(2, 2)

soma2 = soma(3, 3)

print(soma1)

print(soma2)

print(soma(11, 55))

saída

4

6

[10, 20]

### 1.8.4 \*args (Argumentos não nomeados)

args - Argumentos não nomeados

\* - \*args (empacotamento e desempacotamento)

Lembre-te de desempacotamento:

x, y, \*resto = 1, 2, 3, 4

print(x, y, resto)

def soma(\**args*):

    total = 0

    for numero in *args*:

        total += numero

    return total

print(soma(1, 2, 3))

saída

6