**Para saber mais: escopo de uma variável**

Em Python, o escopo de uma variável é definido pela região do código onde ela pode ser acessada. No caso de uma função, o escopo pode ser dividido em duas categorias: **escopo global** e **escopo local**.

O escopo global é o espaço no qual uma variável pode ser acessada por qualquer função ou código que esteja sendo executado no programa. Já o escopo local é o espaço no qual a variável pode ser acessada apenas pela função em que foi definida.

O problema de escopo ocorre quando uma variável é definida dentro do escopo de uma função e, em seguida, é referenciada fora do escopo da função. Nesse caso, o Python gera uma mensagem de erro, indicando que a variável não foi definida (NameError).

Abaixo segue um exemplo que ilustra esse comportamento. Inicialmente, vamos criar uma variável x externa a função soma(), na qual definimos uma outra variável y e, por fim, imprimimos a soma das duas variáveis.

x = 7

def **soma**():

y = 9

print(x + y)

Copiar código

Note que o x é a nossa variável definida no escopo global e o y a variável definida no escopo local da função soma(). Quando tentamos executar a nossa função, a soma é realizada normalmente:

soma()

Copiar código

**Saída:**

16

Copiar código

No entanto, o Python gera um erro quando tentamos imprimir a soma de x e y fora do escopo da função, pois a variável y existe apenas dentro da função soma().

print(x + y)

Copiar código

**Saída:**

---------------------------------------------------------------------------

NameError Traceback (most recent **call** **last**)

<ipython-input-4-f09a7b03ddbf> **in** <**module**>

----> 1 print(x + y)

NameError: name 'y' **is** **not** defined

Copiar código

Para corrigir esse erro, podemos tornar y uma variável global ou retornar seu valor na função e atribuí-lo a uma variável externa. No próximo vídeo, vamos aprender como proceder nesse tipo de situação.