

Escopo de Variáveis em Python

O escopo de variáveis em Python é fundamental para entender como as variáveis são acessíveis em diferentes partes do código. Python possui principalmente dois tipos de escopo: **local** e **global**.

Escopo Local:

Variáveis Locais:

Variáveis declaradas dentro de uma função têm escopo local. Isso significa que elas são visíveis apenas dentro do bloco de código da função onde foram definidas. Quando a função é chamada, essas variáveis são criadas e quando a execução da função é concluída, elas são destruídas.

Exemplo:

```
def exemplo_funcao():  
    variavel_local = 10  
    print(variavel_local)  
  
exemplo_funcao()  
# Output: 10  
  
# A variável local não é acessível fora da função  
# print(variavel_local) # Isso gerará um erro
```

Parâmetros de Função:

Os parâmetros de uma função também têm escopo local. Eles são criados quando a função é chamada e existem apenas durante a execução dessa chamada. Os parâmetros permitem que dados sejam passados para a função.

Exemplo:

```
def soma(a, b):  
    resultado = a + b  
    return resultado  
  
# a e b são parâmetros locais  
# resultado é uma variável local
```

Escopo Global:

Variáveis Globais:

Variáveis declaradas fora de qualquer função ou bloco de código têm escopo global. Isso significa que elas podem ser acessadas de qualquer parte do programa. No entanto, é necessário ter cuidado ao modificar

variáveis globais dentro de funções, pois isso pode levar a efeitos colaterais inesperados.

Exemplo:

```
variavel_global = 5

def exemplo_funcao():
    print(variavel_global)

exemplo_funcao()
# Output: 5

# A variável global é acessível em qualquer parte do código
print(variavel_global)
# Output: 5
```

Palavra-chave **global**:

Para modificar uma variável global dentro de uma função, é necessário usar a palavra-chave **global**. Isso informa ao interpretador que a variável não deve ser tratada como local, mas sim como global.

Exemplo:

```
variavel_global = 5

def modificar_variavel():
    global variavel_global
    variavel_global = 10

modificar_variavel()
print(variavel_global)
# Output: 10
```

Considerações Finais:

Entender o escopo de variáveis é crucial para escrever código claro e evitar problemas de acesso indevido. O uso adequado de variáveis locais e globais contribui para a modularidade e manutenção do código.

Ao escolher entre variáveis locais e globais, leve em consideração a clareza do código e evite o uso excessivo de variáveis globais, pois isso pode tornar o código mais difícil de entender e manter.

Espero que essas explicações adicionais forneçam uma compreensão mais profunda do escopo de variáveis em Python. Se houver mais dúvidas ou tópicos específicos que gostaria de abordar, sintá-se à vontade para perguntar!