

Variáveis em Python

Índice

[Introdução](#)

[Tipos de Dados](#)

[Atribuição de Valores](#)

[Reatribuição de Valores](#)

[Nomenclatura de Variáveis](#)

[Escopo de Variáveis](#)

[Variáveis Locais e Globais](#)

[Conversão de Tipos](#)

Explicações Adicionais

- **Introdução:** Explica o que são variáveis e seu papel em Python.
- **Tipos de Dados:** Detalha os diferentes tipos de dados que uma variável pode ter.
- **Atribuição de Valores:** Mostra como atribuir valores a variáveis.
- **Reatribuição de Valores:** Explica como reatribuir valores a variáveis.
- **Nomenclatura de Variáveis:** Fornece diretrizes para nomear variáveis em Python.
- **Escopo de Variáveis:** Discute o conceito de escopo e visibilidade das variáveis.
- **Variáveis Locais e Globais:** Explica a diferença entre variáveis locais e globais.
- **Conversão de Tipos:** Demonstra como converter entre diferentes tipos de dados.

Introdução

Em Python, uma variável é um espaço de armazenamento que tem um nome simbólico e é associado a um valor. As variáveis permitem que você armazene, modifique e recupere dados durante a execução de um programa.

Tipos de Dados

As variáveis em Python podem armazenar diferentes tipos de dados, incluindo:

- **Inteiros:** Números inteiros, como `5` ou `-3`.
- **Float:** Números de ponto flutuante, como `3.14` ou `-0.001`.
- **Strings:** Sequências de caracteres, como `"Olá, Mundo!"`.
- **Booleanos:** Valores `True` ou `False`.
- **Listas:** Sequências de valores, que podem ser de diferentes tipos, como `[1, 2, 3]` ou `["a", "b", "c"]`.
- **Dicionários:** Estruturas de dados que armazenam pares chave-valor, como `{"nome": "João", "idade": 30}`.
- **Sets:** Coleções de valores únicos e não ordenados, como `{1, 2, 3}`.

Atribuição de Valores

Para atribuir um valor a uma variável, você usa o operador de atribuição `=`.

```
# Código
nome = "Maria" # Atribui uma string à variável 'nome'
idade = 25     # Atribui um inteiro à variável 'idade'
altura = 1.70  # Atribui um float à variável 'altura'

print(nome)
print(idade)
print(altura)

# Saída
# Maria
# 25
# 1.70
```

Reatribuição de Valores

Você pode reatribuir um novo valor a uma variável a qualquer momento. A reatribuição altera o valor armazenado na variável.

```
# Código
valor = 10
print(valor) # Saída: 10

valor = 20
print(valor) # Saída: 20
```

Nomenclatura de Variáveis

Os nomes das variáveis devem seguir algumas regras:

- Devem começar com uma letra ou sublinhado.
- Podem conter letras, números e sublinhados.
- Não podem ser palavras reservadas.

```
# Código
nome_usuario = "Ana" # Nome de variável válido
usuariol = "Carlos"  # Nome de variável válido
lusuario = "João"    # Nome de variável inválido (não pode começar com um número)
```

Escopo de Variáveis

O escopo de uma variável determina onde ela pode ser acessada. Variáveis definidas dentro de uma função são locais a essa função, enquanto variáveis definidas fora de funções são globais.

```
# Código
variavel_global = "Eu sou global"

def funcao():
    variavel_local = "Eu sou local"
    print(variavel_global) # Acessa variável global
    print(variavel_local) # Acessa variável local

funcao()
print(variavel_global) # Acessa variável global
# print(variavel_local) # Gera um erro, pois 'variavel_local' não está definida fora da

# Saída
# Eu sou global
# Eu sou local
# Eu sou global
```

Variáveis Locais e Globais

Em Python, o escopo de uma variável pode ser local ou global. Entender a diferença é crucial para evitar erros e garantir que seu código funcione conforme o esperado.

Variáveis Globais

Variáveis globais são definidas fora de qualquer função e podem ser acessadas de qualquer lugar no código. Elas têm um escopo global e são úteis quando você precisa de uma variável acessível em múltiplas funções.

```
# Código
variavel_global = "Eu sou global"

def mostra_variavel():
    print(variavel_global) # Acessa a variável global

mostra_variavel()
print(variavel_global) # Também acessa a variável global

# Saída
# Eu sou global
# Eu sou global
```

Variáveis Locais

Variáveis locais são definidas dentro de uma função e só podem ser acessadas dentro dessa função. Elas são úteis para armazenar valores temporários e específicos para a função.

```
# Código
def funcao_local():
    variavel_local = "Eu sou local"
    print(variavel_local) # Acessa a variável local

funcao_local()
# print(variavel_local) # Gera um erro, pois 'variavel_local' não está definida fora da

# Saída
# Eu sou local
```

Variáveis Locais e Globais no Mesmo Contexto

Se você tentar modificar uma variável global dentro de uma função, deve usar a palavra-chave `global`

para indicar que está se referindo à variável global.

```
# Código
variavel_global = "Valor inicial"

def modificar_global():
    global variavel_global
    variavel_global = "Valor modificado"

modificar_global()
print(variavel_global) # Saída: Valor modificado

# Saída
# Valor modificado
```

Conversão de Tipos

Você pode converter uma variável de um tipo de dado para outro usando funções de conversão.

```
# Código
numero = "42" # String
numero_inteiro = int(numero) # Converte para inteiro
print(numero_inteiro) # Saída: 42

decimal = 3.14 # Float
decimal_str = str(decimal) # Converte para string
print(decimal_str) # Saída: 3.14
```