

Introdução à Programação

Aula 02 Variáveis e Expressões

Profa. Andreia Rodrigues



Agenda

— — —

- Comando de Atribuição
- Variáveis
- Constantes
- Expressões Aritméticas
- Expressões Relacionais
- Expressões Lógicas

Comando de Atribuição



Comando de Atribuição

— — —

O comando de atribuição é uma instrução em linguagens de programação que permite que um valor seja atribuído a uma variável.

É utilizado para armazenar valores e manipular dados durante a execução do programa.

Em geral, os comandos de atribuição são escritos na forma "variável = valor", onde a variável é o nome da variável que se deseja definir e o valor é o conteúdo que se deseja atribuir a ela.

Comando de Atribuição

— — —

A atribuição é uma operação fundamental em muitas linguagens de programação, incluindo C, Java, Python, entre outras.

O valor pode ser uma expressão ou outro tipo de dado que a linguagem de programação suporta.

Comando de Atribuição

— — —

Uma vez que uma variável é atribuída a um valor, esse valor pode ser usado em outras operações ou expressões dentro do programa, permitindo que os programas realizem cálculos e tomem decisões com base nos valores armazenados nas variáveis.

Variáveis



Variáveis

— — —

Variáveis são espaços de memória usados para armazenar valores em um programa de computador.

Elas permitem que os dados sejam armazenados em tempo de execução e acessados posteriormente durante a execução do programa.

Variáveis

— — —

As variáveis têm um nome, um tipo de dado que elas armazenam e um valor atual que pode ser atualizado ao longo do tempo durante a execução do programa.

A maioria das linguagens de programação exige que as variáveis sejam declaradas com um tipo de dado antes de serem usadas em um programa.



Variáveis

— — —

As variáveis permitem que os programas sejam mais dinâmicos e flexíveis, permitindo que os desenvolvedores armazenem dados e realizem cálculos com eles.

Além disso, elas permitem que os programas sejam escritos de maneira mais legível e modular, tornando o código mais fácil de entender e modificar.



Exemplos de Variáveis

— — —

Em Python, as variáveis são declaradas de maneira simples, atribuindo um valor a um nome de variável.

Não é necessário especificar o tipo de dados durante a declaração da variável.

O tipo de dados é inferido automaticamente pelo interpretador do Python com base no valor atribuído à variável.



Exemplos de Variáveis em Python

— — —

idade = 20 # número inteiro

peso = 60.5 # ponto flutuante

aprovado = True # booleano

nome = "João" # texto

numeros = [1, 2, 3, 4, 5] # lista

pessoa = {"nome": "João", "idade": 20, "altura": 1.75} # dicionário



Constantes



Constantes

— — —

Constantes são valores imutáveis que são atribuídos a uma variável em um programa e não podem ser alterados durante a execução do programa.

Esses valores são definidos pelo programador e permanecem fixos durante todo o processo de execução do programa.

As constantes são frequentemente usadas para representar valores fixos, como pi ou o número de segundos em um minuto.

Constantes

— — —

As constantes são uma parte importante da programação, pois elas ajudam a tornar o código mais legível, reduzem a possibilidade de erros e permitem que o programador altere facilmente os valores fixos em todo o programa.

As constantes são amplamente utilizadas em linguagens de programação como C, C++, Java e Python.



Constantes em Python

— — —

Em Python, as constantes são geralmente definidas como variáveis que não mudam de valor durante a execução do programa.

Embora a linguagem Python não possua suporte nativo para constantes, é uma prática comum nomear variáveis que não são destinadas a mudar como constantes, indicando que essas variáveis não devem ser alteradas no decorrer do programa.

Exemplos de Constantes em Python

— — —

Esses exemplos são apenas algumas das muitas possibilidades de constantes que podem ser definidas em um programa Python:

`PI = 3.14`

`TAXA_DE_JUROS = 0.05`

`NOME_APP = "Meu App Python"`

É importante lembrar que, em Python, essas variáveis ainda podem ser alteradas se o programador tentar fazê-lo, mas o uso de convenções de nomenclatura ajuda a indicar que elas não devem ser alteradas.

Expressões Aritméticas



Expressões Aritméticas

— — —

Expressões aritméticas são fórmulas matemáticas que consistem em números e operadores aritméticos para realizar cálculos numéricos.

As operações aritméticas comuns incluem:

- Adição (+)
- Subtração (-)
- Multiplicação (*)
- Divisão (/)
- Exponenciação (**)

Expressões Aritméticas

— — —

As expressões aritméticas seguem uma ordem de precedência, onde as operações com maior precedência são realizadas primeiro, seguidas pelas operações com menor precedência.

As expressões aritméticas são amplamente utilizadas em programação para realizar cálculos numéricos e manipular dados numéricos.



Exemplo de Expressão Aritmética

— — —

Um exemplo de expressão aritmética com diferentes operadores, operandos e precedências pode ser o seguinte:

$$a = 5$$

$$b = 3$$

$$c = 2$$

$$\text{resultado} = (a + b) * c ** 2 / (a - b)$$

Exemplo de Expressão Aritmética

— — —

Nesse exemplo, a expressão consiste em diferentes operadores aritméticos, operandos e precedências:

- Os operandos são as variáveis a , b e c , que têm valores atribuídos anteriormente.
- Os operadores são $+$, $-$, $*$, $/$ e $**$, que realizam, respectivamente, as operações de adição, subtração, multiplicação, divisão e exponenciação.
- A expressão dentro dos parênteses é resolvida primeiro, pois tem maior precedência. Nesse caso, $a + b$ é igual a 8.
- Em seguida, é realizada a exponenciação de c por 2 (o operador $**$ tem maior precedência que $*$ e $/$). O resultado é 4.
- Depois, a multiplicação de 8 por 4 é realizada, resultando em 32.
- Por fim, a divisão de 32 pelo resultado de $a - b$ é realizada, onde $a - b$ é igual a 2. O resultado final é 16.

Expressões Relacionais



Expressões Relacionais

— — —

Expressões relacionais são usadas em programação para comparar valores ou variáveis e retornar um resultado verdadeiro ou falso (booleano).

As expressões relacionais são compostas por operadores relacionais, como "igual a", "maior que", "menor que", "maior ou igual a", "menor ou igual a" e "diferente de", que são usados para comparar valores.



Exemplo de Expressão Relacional

— — —

x = 10

y = 5

z = 10

x == y # retorna False, pois x não é igual a y

x > y # retorna True, pois x é maior que y

y <= z # retorna True, pois y é menor ou igual a z

x != z # retorna False, pois x é igual a z



Exemplo de Expressão Relacional

— — —

Nesse exemplo, a variável x tem valor 10, a variável y tem valor 5 e a variável z tem valor 10.

As expressões relacionais comparam essas variáveis usando operadores relacionais, retornando um resultado booleano (True ou False) dependendo da comparação.

Por exemplo, a expressão $x == y$ compara se x é igual a y e retorna False, pois x não é igual a y .

Já a expressão $x > y$ compara se x é maior que y e retorna True, pois x é maior que y .

Expressões Lógicas



Expressões Lógicas

— — —

Expressões lógicas são usadas em programação para combinar expressões relacionais e retornar um resultado verdadeiro ou falso (booleano).

As expressões lógicas são compostas pelos operadores lógicos "e" (and), "ou" (or) e "não" (not), que são usados para combinar as expressões relacionais.

Exemplo de Expressão Lógica

— — —

x = 10

y = 5

z = 3

retorna True, pois ambas as expressões relacionais são verdadeiras
(x > y) and (y > z)

retorna True, pois a primeira expressão relacional é verdadeira
(x > y) or (y < z)

retorna True, pois a expressão relacional é falsa e a negação é verdadeira
not(x == y)



`print(expressão)`

Utilize esse comando para exibir na tela o resultado de uma variável, constante ou expressão.

Exercício

Escreva um programa em Python que atribua o valor da soma de 5 e 7 a uma variável chamada "soma".



Exercício

Escreva uma expressão aritmética em Python que calcule a área de um retângulo, dadas sua largura e altura como constantes.



Exercício

Escreva uma expressão lógica em Python que informe se um número inteiro é ímpar.



Avaliação



Encerramento

- Revisão
- Exercício Complementar
- Próxima Aula

