



Fundamentos de Programação

Operadores e EXPRESSÕES

Expressões Aritméticas

Aula 06

Post It da Aula Passada

Escrita de Dados

- Dispositivo de escrita (saída) padrão
 - Tela
 - Terminal
- A função *print*
 - Argumentos
 - Uso de vírgulas

Leitura de Dados

- Dispositivo de leitura (entrada) padrão
 - Teclado
- A função *input*
 - Argumentos
 - Atribuição
 - Casting

Expressões Aritméticas

Expressões aritméticas consistem em expressões matemáticas onde os **operadores são aritméticos e os operandos são numéricos**

Sendo assim, o resultado de uma expressão aritmética também é sempre numérico: um dado do tipo inteiro ou real

Entretanto, antes de compreendermos as expressões aritméticas em si, precisamos conhecer os operadores aritméticos disponíveis

Operadores Aritméticos

Operadores aritméticos são responsáveis pela execução de operações matemáticas em um computador

Operadores aritméticos podem ser binários ou unários:

- Unários: utilizados na inversão do estado de um dado numérico (positivo para negativo ou negativo para positivo)
- Binários: utilizados em operações matemáticas como adição, subtração, multiplicação, divisão, radiciação, exponenciação...

Operadores Aritméticos

Operador	Descrição	Tipo
+	Manutenção de sinal	Unário
-	Inversão de sinal	Unário
=	Atribuição	Binário

Operadores Aritméticos

Operador	Descrição	Tipo
+	Adição	Binário
-	Subtração	Binário
*	Multiplicação	Binário
/	Divisão	Binário
//	Divisão inteira (piso)	Binário
%	Divisão inteira (resto)	Binário
^	Exponenciação	Binário
√	Radiciação	Binário

Expressões Aritméticas

“As expressões aritméticas são realizadas a partir do **relacionamento existente entre variáveis e constantes numéricas** com a utilização de operadores aritméticos”

Manzano e Oliveira

Expressões Aritméticas

5 + 5

C O A C

a * 10

V O A C

a ^ b

V O A V

c = 5 + a * 2

V O A C O A V O A C

a = 3.1415 * r ^ 2

V O A C O A V O A C

Precedência de Operadores

A precedência indica a **ordem que os operadores são processados** em uma expressão aritmética

De maneira geral, podemos considerar que:

- 1.** Operadores unários precedem os demais
- 2.** Operadores binários têm diferentes precedências entre si

Precedência de Operadores

Operador	Descrição	Tipo	Precedência
+	Manutenção de sinal	Unário	1 (maior)
-	Inversão de sinal	Unário	1 (maior)
=	Atribuição	Binário	—

Precedência de Operadores

Operador	Descrição	Tipo	Precedência
+	Adição	Binário	4
-	Subtração	Binário	4
*	Multiplicação	Binário	3
/	Divisão	Binário	3
//	Divisão inteira	Binário	3
%	Módulo	Binário	3
^	Exponenciação	Binário	2 (maior binário)
√	Radiciação	Binário	2 (maior binário)

Precedência de Operadores

Sabendo da precedência dos operadores, é relativamente simples identificar a ordem de execução das operações na expressão a seguir:

$$10 + 2 * 0.5 ^ 2$$

a b c

1º) $10 + 2 * 0.5 ^ 2$

2º) $10 + 2 * 0.25$

3º) $10 + 0.5$

4º) 10.5

Precedência de Operadores

Mas e quando existem **múltiplas operações com a mesma precedência** na expressão aritmética?

ASSOCIATIVIDADE

A associatividade determina a ordem de execução de operadores com a mesma precedência

Precedência de Operadores

A associatividade pode ser...

- à esquerda: operações processadas da esquerda para a direita
- à direita: operações processadas da direita para a esquerda

A associatividade padrão em linguagens de programação é a **associatividade à esquerda**

Note que a escolha por uma associatividade ou outra pode **alterar o resultado** da expressão aritmética

Precedência de Operadores

Associatividade **à esquerda**

$$\begin{array}{c} a \quad b \quad c \\ 8 / 1 * 4 * 2 \end{array}$$

1º) **8 / 1** * 4 * 2

2º) **8 * 4** * 2

3º) **32 * 2**

4º) **64**

Associatividade **à direita**

$$\begin{array}{c} a \quad b \quad c \\ 8 / 1 * 4 * 2 \end{array}$$

1º) 8 / 1 * **4 * 2**

2º) 8 / **1 * 8**

3º) **8 / 8**

4º) **1**

Precedência de Operadores

Porém, a precedência de operadores pode ser alterada utilizando **parênteses** nas expressões aritméticas

Os parênteses mais internos de uma expressão aritmética são sempre resolvidos primeiro!

Dentro dos parênteses, entretanto, valem as mesmas regras de precedência e associatividade

Precedência de Operadores

$$10 * (2 + 5 - 1) ^ 2$$

1º) $10 * (2 + 5 - 1) ^ 2$

2º) $10 * (7 - 1) ^ 2$

3º) $10 * 6 ^ 2$

4º) $10 * 36$

5º) 360

Tipo do Resultado

Qual é o **tipo de dados retornado** pelas expressões aritméticas em Python?

Depende! Existem dois fatores: o tipo de dados na expressão e as operações realizadas na mesma

Se existir um número real, o resultado será do tipo real (**float**)

Se existir uma divisão, o resultado será do tipo real (**float**)

Se existir uma radiciação, o resultado será do tipo real (**float**)

Em qualquer outro caso, o resultado será do tipo inteiro (**int**)

Python

A maioria dos operadores aritméticos vistos até aqui estão nativamente implementados em Python

Estão presentes na linguagem todos os operadores unários e os binários de...

- Atribuição (=)
- Adição (+) e subtração (-)
- Multiplicação (*) e divisão (/)
- Divisão inteira em piso (//) e resto da divisão inteira (%)
- Exponenciação (**)
- Unários (+ e -)

Python

A **radiciação não está disponível nativamente** na linguagem Python; porém, existem duas maneiras comuns de executar a radiciação em Python:

- Através de exponenciação
 - Por exemplo, para obter a raiz quadrada basta definir o expoente em 0.5
- Através da biblioteca matemática *math*
 - Importação da biblioteca e utilização da função *sqrt* para executar a raiz quadrada
 - **Ainda não estudamos!!!!**

Python

Nossos exemplos anteriores...

Notação **tradicional**

$$a = 3.1415 * r ^ 2$$

$$10 + 2 * 0.5 ^ 2$$

$$8 / 1 * 4 * 2$$

$$10 * (2 + 5 - 1) ^ 2$$

Notação **Python**

$$a = 3.1415 * r ** 2$$

$$10 + 2 * 0.5 ** 2$$

$$8 / 1 * 4 * 2$$

$$10 * (2 + 5 - 1) ** 2$$

Exercício #06

Considere a seguinte expressão aritmética:

$$10 + (25 - 5) * 2 + (10 // 3) ** 2$$

Faça um algoritmo em Python que mostre na tela todos os **resultados parciais de cada operação realizada**. Por exemplo, o primeiro *print* deve ser:

```
print("(25 - 5)", 25-5) ou print("25 - 5", 20) ou print("25-5", parc)
```

Mostre o resultado da expressão calculada em Python para conferir o resultado do seu cálculo



Fundamentos de Programação
Aula 06

Obrigado e
ATÉ A
PRÓXIMA!