

Expressões

Operadores de atribuição, funções integradas e entrada/saída de dados

Prof. Me. Lucio Nunes

Prof. MSc. Rafael Maximo

Tópicos

Nesta aula iremos falar sobre:

- Expressões aritméticas;
- Operadores de atribuição;
- Precedência e associatividade;
- Avaliação de expressões;
- Funções integradas;
- Entrada e saída de dados; e
- Tipos de erros.

Objetivos

Acompanhe, a seguir, os objetivos de aprendizagem para esta aula:

- Relembrar o que é uma expressão;
- Conhecer os operadores de atribuição composta;
- Compreender o conceito de precedência e associatividade de operadores;
- Entender como o Python avalia expressões;
- Utilizar funções integradas da linguagem Python;
- Obter e exibir dados de/para o usuário;
- Conhecer os tipos comuns de erros em programação.

Operadores de atribuição

Tabela 1: Relação parcial dos operadores de atribuição composta do Python (continua).

Operador	Descrição	Exemplos
=	Atribui o resultado da expressão à variável.	x = expressão
+=	Atribui o resultado da adição à variável.	x += expressão x = x + (expressão)
-=	Atribui o resultado da subtração à variável.	x -= expressão x = x - (expressão)
*=	Atribui o resultado da multiplicação à variável.	x *= expressão x = x * (expressão)
=	Atribui o resultado da expressão à variável.	x = expressão

Fonte: Elaborado pelo autor.

Operadores de atribuição

Tabela 1: Relação parcial dos operadores de atribuição composta do Python (conclusão).

Operador	Descrição	Exemplos
/=	Atribui o quociente da divisão real à variável.	x /= expressão x = x / (expressão)
//=	Atribui o quociente da divisão inteira à variável.	x //= expressão x = x // (expressão)
%=	Atribui o resto da divisão inteira à variável.	x %= expressão x = x % (expressão)
**=	Atribui o resultado da exponenciação à variável.	x **= expressão x = x ** (expressão)
/=	Atribui o quociente da divisão real à variável.	x /= expressão x = x / (expressão)

Fonte: Elaborado pelo autor.

Expressões

Uma expressão é uma combinação de operandos com um ou mais operadores e que resulta em um valor.

- Os operandos podem ser, por exemplo, constantes ou variáveis;
- A combinação de operandos e operadores deve ser válida;
- Podemos usar parênteses, como na matemática.

Exemplo: **(2 + 5)**

Precedência e associatividade de operadores

A **precedência dos operadores** indica a prioridade que um operador possui em relação aos outros em uma mesma expressão, assim como na matemática.

- $2 + 3 * 5 \rightarrow 17 \text{ ou } 25?$
- $(2+3) * 5 \rightarrow 17 \text{ ou } 25?$

Precedência e associatividade de operadores

A **associatividade** é uma propriedade que define como operadores de mesma precedência devem ser agrupados, caso estejam na mesma expressão e sem parênteses que determinem a ordem da avaliação.

- $8 / 4 * 2 \rightarrow 4.0 \text{ ou } 1.0?$
- $2 ** 1 ** 2 \rightarrow 2 \text{ ou } 4?$

Precedência e associatividade de operadores

Tabela 2: Precedência e associatividade dos operadores vistos até agora.

Operador	Descrição	Associatividade
**	Exponenciação	à direita
+, - (unários)	Identidade e negação (inversão de sinal)	à esquerda
*, /, //, %	Multiplicação, divisão real, divisão inteira e resto da divisão	
+, - (binários)	Adição e subtração	
=, +=, -=, *=, /=, //=, %=, **=	Atribuição (simples e composta)	Não associativos

Fonte: Elaborado pelo autor.

Avaliação de expressões

```
>>> a = 5
>>> b = 4
>>> c = 9
>>> d = 7
>>> e = 1
>>> f = 2
>>> s = 10
>>> s += a + b**(c-d)/e * f
>>> s
```

Agora conseguimos entender a forma como o Python **avalia** nossas expressões.

Qual o valor de s?

Funções

Uma função é uma sequência de instruções que executa alguma tarefa específica e que tem um nome.

- *Integradas*: na própria linguagem de programação e disponíveis para uso a qualquer momento;
- *Importadas*: criadas por outros programadores e disponibilizadas para serem incluídas no ambiente de programação;
- *Definidas*: construídas pelo próprio programador no código-fonte.

Saída de dados

A saída de dados na tela é feita com a função `print`.

```
>>> print('Hello World!')    # 1 argumento.  
Hello World!  
>>> print('2 + 2 =', 2+2)    # 2 argumentos.  
2 + 2 = 4  
>>> print('oi', 2*5, 'tchau') # 3 argumentos.  
oi 10 tchau
```

Entrada de dados

A entrada de dados do teclado é feita com a função **input**.

```
nome = input('Seu nome: ')      # "Megan"
valor = input('Valor: ')        # "2000.0"
parcelas = input('Parcelas: ')  # "5"
```

A função **input** sempre irá retornar uma *string*, então se queremos trabalhar com números é necessário fazer a conversão explicitamente:

```
valor = float(input('Valor: '))  # 2000.0
parcelas = int(input('Parcelas: ')) # 5
```

Tipos de erros

- **Erros de sintaxe:** erros na escrita das instruções da linguagem, violando regras e estruturas;
- **Erros em tempo de execução:** erros que só aparecem quando o programa é executado, pois a sintaxe está correta;
- **Erros de lógica/semântica:** erros relacionados ao algoritmo. Nesses casos, o código-fonte pode ser executado e o programa não gerará nenhuma mensagem de erro, mas o resultado não resolve o problema proposto.

```
>>> n = 2 + 3) * 5
SyntaxError: unmatched ')'
```

```
>>> n = int(input('int: '))
int: ABC
ValueError: invalid literal
for int() with base 10: 'ABC'
```

```
>>> print('2 + 5 =', 2 * 5)
2 + 5 = 10
```

Vamos praticar!

Crie um programa para um site! O programa deverá solicitar o valor de um item e a quantidade de unidades compradas deste item, ao final deve exibir o total com desconto de 10%. Considere que a quantidade será um número natural positivo.



**Livro “Pense em Python”
2º Edição**

Referências

- PYTHON SOFTWARE FOUNDATION. **Expressions**. 2020. Disponível em:
<<https://docs.python.org/3/reference/expressions.html>>. Acesso em: 21 jan. 2021.
- STURTZ, J. Operators and Expressions in Python. **Real Python**, 2018. Disponível em:
<<https://realpython.com/python-operators-expressions/>>. Acesso em: 21 jan. 2021.
- STURTZ, J. Basic Input, Output, and String Formatting in Python. **Real Python**, 2019. Disponível em:
<<https://realpython.com/python-input-output/>>. Acesso em: 21 jan. 2021.