Algoritmos e Estruturas de Dados

Expressões

Python

Expressões

Uma expressão é uma combinação de elementos, que podem ser valores, variáveis, operadores e chamadas a funções. Com o uso das expressões é possível realizar cálculos que produzem novos valores, ou seja, fazem a transformação das informações.

Variáveis

Como visto anteriormente, variáveis são localizações na memória que armazenam dados.

Para associar um valor a uma variável utiliza-se o operador ← ao lado direito da variável.

Exemplo

Um exemplo de expressão é

```
soma = 5 + 4
```

em que:

- `5` e `4` são valores,
- `+` é um operador aritmético,
- `=` é o operador de atribuição, e
- `soma` é uma variável.

Podemos ler a expressão como *"soma **recebe** cinco mais quatro"*.

A execução da expressão `5 + 4` expressão resulta no valor `9`, o qual será armazenado na variável `soma`.

Operadores

Os operadores são utilizados para construir **expressões**, que podem conter diferentes quantidades de operandos.

Atribuição

Como dito anteriormente, a atribuição é o operador que determina a passagem de valor para uma variável. Por definição toda variável pode ser seu valor alterado, e a modificação deste valor é realizada com o operador de atribuição.

|Operador|Função|
|

`--`

|
| `=` |atribuição|

Exemplo

- `lado = 8`
- `distancia = 49.6`
- `nome = "Adalberto"`
- `custo = 5.50`

`+`	adição	`5+2 = 7`
`-`	subtração	`5-2 = 3`
`*`	multiplicação	`3*6 = 18`
`/`	divisão	`11/2 = 5.5`
`**`	potência. x^y	`5**2 = 25`
`sqrt(x)`	raiz quadrada. $\sqrt{x}$ (`from math import sqrt`)	`sqrt(16) = 4.0`
`%`	resto da divisão inteira.	`20%6 = 2`
`//`	quociente da divisão inteira.	`20//6 = 3`

Exemplos

```
valor_i = 5+10
print(valor_i) #15
valor_i = 8-4
print(valor_i) #4
valor_i = 6*4
print(valor_i) #24
valor_i = 6*4.5
print(valor_i) #27.0
valor_i = 10/2
print(valor_i) #5.0
```

Exercícios

- Calcule a área de um quadrado de lado L. Utilize duas variáveis.
- No dia de seu aniversário, Tomás deseja saber qual e a sua idade, em dias. Considere que Tomás está completando X anos, e cada ano possui 365 dias. Utilize duas variáveis.

Relacionais

São operadores de comparação entre valores. As expressões realizadas com estes operadores retornam um resultado do tipo lógico, `verdadeiro` ou `falso` (FORBELLONE 2022, p.27).

```
print("Olá Mundo")
```

Exemplos

```
print(6>7) #False
print(6<7) #True
print(6==7) #False
print(4==4) #True
print(12>=12) #True
print(15>=12) #True
print(12>=12) #True
print(12<=12) #True
print(15<=12) #False
print(8<=12) #True
print(3!=11) #True
print(12!=12) #False
```

A conjunção corresponde ao **e** lógico. Possui resultado verdadeiro apenas quando ambas entradas forem verdadeiras, e falso para os demais casos.

Disjunção

A disjunção corresponde ao **ou** lógico. É falso apenas quando ambas as entradas são falsas. Basta apenas um dos dos operandos serem verdadeiros para resultar em verdadeiro.

Negação

A negação corresponde ao **não** lógico. Consiste na inversão lógica do valor de entrada. A negação é um operador unário, ou seja, atua sobre um único operando.

Em resumo, para os operadores lógicos temos:

|Operador|Função|

Operadores compostos

É possível combinar alguns operadores, que realizam a operação utilizando os parâmetros passados ao operador, e em seguida realiza uma atribuição utilizando a variável à esquerda dos operadores.

Operador	Função
`+=`	adição e atribuição
`-=`	subtração e atribuição
`*=`	multiplicação e atribuição
`/=`	divisão e atribuição
`//=`	divisão inteira e atribuição

mesma prioridade, a expressão será executada da esquerda para a direita.

|Tipo|Símbolo|

-|
parênteses, colchetes e chaves	`()` , `[]` , `{}`
potência	`**`
positivo, negativo	`+x` `-x`
multiplicação e divisão	`*` `/` `//` `%`
adição e subtração	`+` `-`
comparações	`in` `not in` `is` `is not` `<` `<=` `>=` `!=` `==`
não lógico	`not`
e lógico	`and`
ou lógico	`or`
atribuição	`=`

Adaptado de PYTHON SOFTWARE FOUNDATION. Expressions - Python 3 documentation.