

# Algoritmos e Estruturas de Dados

**Expressões**

**C**

# 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.

soma é uma variável.

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

</TabItem> <TabItem value="java" label="Java">

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*".

</TabItem> <TabItem value="python" label="Python">

Um exemplo de expressão é

# 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.

```
<Tabs groupId='language'> <TabItem value="pseudocodigo" label="Pseudocódigo"
default>
```

## |Operador|Função|

|

|

|

| ← |atribuição|

## Exemplo

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

</TabItem> <TabItem value="java" label="Java">

## |Operador|Função|

|

|

| = |atribuição|

## Exemplo

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

</TabItem> <TabItem value="python" label="Python">

## |Operador|Função|

|

|

| `=` |atribuição|



## Exemplo

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

`</TabItem> </Tabs>`

-----|

| + | adição |  $5+2 = 7$  |

| - | subtração |  $5-2 = 3$  |

| \* | multiplicação |  $3*6 = 18$  |

| / | divisão |  $10/2 = 5.0$  |

| pot(x,y) | potência.  $x^y$  |  $\text{pot}(5,2) = 25$  |

| rad(x) | raiz quadrada.  $\sqrt{x}$  |  $\text{rad}(16) = 4$  |

| mod(x,y) | resto da divisão inteira. |  $\text{mod}(20,6) = 2$  |

| div(x,y) | quociente da divisão inteira. |  $\text{div}(20,6) = 3$  |

## Exemplos

```
int valor_i;  
float valor_f;  
  
valor_i = 5+10;  
System.out.println(valor_i); //15  
valor_i = 8-4;  
System.out.println(valor_i); //4  
valor_i = 6*4;  
System.out.println(valor_i); //24
```

## Exercícios

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

```
escreva(6>7); //falso
escreva(6<7); //verdadeiro
escreva(6=7); //falso
escreva(4=4); //verdadeiro
escreva(12>=12); //verdadeiro
escreva(15>=12); //verdadeiro
escreva(12>=12); //verdadeiro
escreva(12<=12); //verdadeiro
escreva(15<=12); //falso
escreva(8<=12); //verdadeiro
escreva(3<>11); //verdadeiro
escreva(12<>12); //falso
```

</TabItem> <TabItem value="java" label="Java">

## Exemplos

```
System.out.println(6>7); //false
System.out.println(6<7); //true
System.out.println(6==7); //false
System.out.println(4==4); //true
System.out.println(12>=12); //true
System.out.println(15>=12); //true
```

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:

```
<Tabs groupId='language'> <TabItem value="pseudocodigo" label="Pseudocódigo"
default>
```

	Operador	Função
--	----------	--------

--	--	--

--	--	--

e	e	lógico (conjunção)
---	---	--------------------

ou	ou	lógico (disjunção)
----	----	--------------------

não	não	lógico (negação)
-----	-----	------------------

```
</TabItem> <TabItem value="java" label="Java">
```

	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.

```
<Tabs groupId='language'> <TabItem value="pseudocodigo" label="Pseudocódigo"
default>
```

Operador	Função
----------	--------

```
</TabItem> <TabItem value="java" label="Java">
```

Operador	Função
----------	--------

```
</TabItem> <TabItem value="python" label="Python">
```

Operador	Função
----------	--------

```
</TabItem> </Tabs>
```

|Tipo|Símbolo|

-|

|Operadores pós-fixados

operadores pré-fixados

*cast*| x++ | x-- |

++x

+x

-x

~x

!x

(type) x |

multiplicação, divisão| \* | / | % |

|adição, subtração| + | - |

|comparação| , | <= | > | >= |

|igualdade| == | != |

|e lógico| && |

|ou lógico| <code> || </code> |

|atribuição| = |

Adaptado de [GOODRICH 2013, p.24].

