



**UDESC**  
UNIVERSIDADE  
DO ESTADO DE  
SANTA CATARINA

Centro de Educação  
do Planalto Norte

Curso: Bacharelado em Sistemas de Informação

Disciplina: Estrutura de Dados I - (3DAD103)

Professor: Alexandre Mendonça Fava

## Exercícios de Revisão

Exercícios adaptados do livro do Nilo Ney Coutinho Menezes:

Introdução à Programação com Python

Algumas operações aritméticas não são definidas em Python da maneira clássica. Por exemplo, a multiplicação em Python é representada por um asterisco (\*), ao invés do tradicional símbolo de multiplicação ( $\times$ ).

**Exercício 1** Converta as seguintes expressões matemáticas para que possam ser calculadas usando o interpretador Python:

$$10 + 20 \times 30$$

$$4^2 + 30$$

$$(9^4 + 2) \times 6 - 1$$

**Exercício 2** Transforme a seguinte expressão na sua forma tradicional:

$$10 \% 3 * 10 ** 2 + 1 - 10 * 4 / 2$$

**Exercício 3**      Considerando o seguinte algoritmo:

```
início
    ler x
    ler y
     $z \leftarrow (x * y) + 5$ 
    se  $z \leq 0$  então
        resposta  $\leftarrow$  'A'
    senão
        se  $z \leq 100$  então
            resposta  $\leftarrow$  'B'
        senão
            resposta  $\leftarrow$  'C'
        fim_se
    fim_se
    escrever z, resposta
fim
```

Faça um teste de mesa e complete o quadro a seguir para os seguintes valores:

Variáveis			
X	Y	Z	Resposta
3	2		
150	3		
7	-1		
-2	5		
50	3		

Algumas operações relacionais não são definidos em Python da maneira clássica. Por exemplo, a ‘diferença’ em Python é representada por uma exclamação seguida de um igual (!=), ao invés do tradicional símbolo matemático ( $\neq$ ).

**Exercício 4** Complete a tabela abaixo, respondendo True ou False. Considere  $a = 4$ ,  $b = 10$ ,  $c = 5.0$ ,  $d = 1$  e  $f = 5$ .

Expressão	Resultado
$a == c$	<input type="radio"/> True <input type="radio"/> False
$a < b$	<input type="radio"/> True <input type="radio"/> False
$d > b$	<input type="radio"/> True <input type="radio"/> False
$c != f$	<input type="radio"/> True <input type="radio"/> False
$a == b$	<input type="radio"/> True <input type="radio"/> False
$c < d$	<input type="radio"/> True <input type="radio"/> False
$b > a$	<input type="radio"/> True <input type="radio"/> False
$c >= f$	<input type="radio"/> True <input type="radio"/> False
$f >= c$	<input type="radio"/> True <input type="radio"/> False
$c <= c$	<input type="radio"/> True <input type="radio"/> False
$c <= f$	<input type="radio"/> True <input type="radio"/> False

**Exercício 5** Em Python, nomes de variáveis devem ser declarados obedecendo um conjunto de regras. Analisando a tabela ao lado, responda ‘SIM’ para as declarações válidas em Python e responda ‘NÃO’ para as inválidas.

Nome	Válido
a1	
velocidade	
velocidade90	
salário_médio	
salário médio	
_b	
1a	

Python suporta três operadores básicos: not (não), and (e), or (ou). Esses operadores são traduzidos como não ( $\neg$  negação), e ( $\wedge$  conjunção) e ou ( $\vee$  disjunção).

**Exercício 6** Complete a tabela a seguir utilizando  $a = \text{True}$ ,  $b = \text{False}$  e  $c = \text{True}$ .

Expressão	Resultado
$a \text{ and } a$	<input type="radio"/> True <input type="radio"/> False
$b \text{ and } b$	<input type="radio"/> True <input type="radio"/> False
$\text{not } c$	<input type="radio"/> True <input type="radio"/> False
$\text{not } b$	<input type="radio"/> True <input type="radio"/> False
$\text{not } a$	<input type="radio"/> True <input type="radio"/> False
$a \text{ and } b$	<input type="radio"/> True <input type="radio"/> False
$b \text{ and } c$	<input type="radio"/> True <input type="radio"/> False
$a \text{ or } c$	<input type="radio"/> True <input type="radio"/> False
$b \text{ or } c$	<input type="radio"/> True <input type="radio"/> False
$c \text{ or } a$	<input type="radio"/> True <input type="radio"/> False
$c \text{ or } b$	<input type="radio"/> True <input type="radio"/> False
$c \text{ or } c$	<input type="radio"/> True <input type="radio"/> False
$b \text{ or } b$	<input type="radio"/> True <input type="radio"/> False

**Exercício 7** Repetições representam a base de vários programas. São utilizadas para executar a mesma parte de um programa várias vezes, normalmente dependendo de uma condição. Cite ao menos uma estrutura de repetição em Python e seu funcionamento.