

# ALGORITMOS

## LISTA DE EXERCÍCIOS

*Versão em Python*

Instituto Federal do Espírito Santo  
Coordenadoria de Física, *Campus* Cariacica

Wesley Spalenza

# ÍNDICES

**CAPÍTULO 1** - INTRODUÇÃO E DEFINIÇÕES

**CAPÍTULO 2** - COMANDOS DE DECISÃO E REPETIÇÃO: IF, ELSE, FOR e WHILE

**CAPÍTULO 3** - LISTAS E ARRAYS

**CAPÍTULO 4** - ARQUIVOS DE ENTRADA E SAÍDA (I/O), GRÁFICOS, CÁLCULOS SIMBÓLICOS (SYMPY)

**CAPÍTULO 5** - FUNÇÕES

# CAPÍTULO 1

## INTRODUÇÃO E DEFINIÇÕES

**Exercício 1.1** Faça um programa que imprima a palavra “Ola mundo”. Agora o implemente no Python e faça-o rodar como executável. Verifique também o tipo de variável estamos definindo.

**Exercício 1.2** Faça cálculos simples: (a)  $1+3 - 2*4 + 8/2$ , (b)  $2.0e-2 * 3e2$ , (c) multiplicar as letras anteriores. (d)  $2j$ , (e) 3,8 elevado a 5 potência, (f) raiz quadrada de 4, (g) Imprimir o tipo de variável que temos em cada letra.

**Exercício 1.3** Faça cálculos simples com números complexos imprimindo na tela: (a)  $2+4j + 3-5j$ , (b)  $(2+4j) *(2-4j)$ , (c)  $(2+3j)/(2-1j)$ , (d)  $(2+4j)**2$ .

**Exercício 1.4** Imprima as expressões:

(a) valor absoluto de -4 = `abs(-4)`,

(b) divisão inteira de 14 por 3 = `14 // 3`

(c) Valor do resto da divisão de 14 por 3 = `14 % 3`

(d) valor mínimo entre 6 e -2 = `min(6,-2)`

(e) valor máximo entre 6 e -2 = `max(6,-2)`

(f) valor mínimo entre 6, 10, -4 e -2 = `min(6, 10, -4, -2)`

(g) valor máximo entre 1, 10.2, 29 e -2.23 = `max(1, 10.2, 29, -2.23)`

**Exercício 1.5** Imprima as expressões Booleanas de comparações (Operadores Lógicas):

(a) `2 < 3`

(b) `2 > 3`

(c) `5 - 2 > 3 + 4`

(d) `3 == 3`

(e) `3 == 4`

- (f) `3 == 5 - 3`
- (g) `3 <= 4`
- (h) `3 >= 4`
- (i) `3 != 4` # != operador não igual
- (j) `1 == 1, 2 == 1`
- (k) `True == True`
- (l) `True == False`
- (m) `False == False`
- (n) `2 < 3 and 4 > 5`
- (o) `2 < 3 and True`
- (p) `2 < 3 or True`
- (q) `True and True, False and False, True and False`
- (r) Se `var1 = True, var2 = False, print(type(var1),type(var2))`

**Exercício 1.6** A partir de operações matemáticas, imprima as operações com Strings.

Se `s = 'Olá'` e `t = 'pessoal'`, imprima:

- (a) `s == 'Olá'`
- (b) `t == 'pessoal'`
- (c) `s + t`
- (d) `s + " + t`
- (e) `s + ' ' + t`
- (f) `s * t`
- (g) `3*s`
- (h) `2*t + 3*s`
- (i) `5*' '`
- (j) `'l' in s, 'á' in s, 'ss' in t, 'v' in t`. # verifica se é uma substring
- (k) `'g' not in s, 'p' not in t`
- (l) `len(s), len(t), len(s)+len(t)` - representa número de espaços em uma string ou comprimento.

**Exercício 1.7** Defina uma string como `a = '0123456789'` e outra como `b = 'Olá pessoal'` e imprima:

- (a) `a[0], a[4], a[9]`;
- (b) `b[0], b[2], b[5]`;
- (c) `a[-1], b[-3]`.

**Exercício 1.8** Imprima e discuta a transformação de variáveis:

- (a) `a = 3 + 0.3` como float (flutuante);
- (b) `b = - 3.2` como integer (inteiro);
- (c) `c = 'Palavra'` como string (caractere);

(d) `d = True` como Boolean (Booleana);

(e) `e = '497.93'` como string.

**Exercício 1.9.** Construa um programa que faça uma soma simples e dois números, como caracteres (string), inteiros e real, de modo que haja o pedido de entrada para que o programa "rode", e em seguida imprima cada resultado.

**Exercício 1.10.** Elabore um programa que calcule a área de uma circunferência e apresente a medida calculada.

**Exercício 1.11** - Imprima os modos matemáticos usando a biblioteca `math`.

(a) raiz quadrada de 9;

(b) cosseno de  $\pi$ ;

(c) seno de  $\pi$ ;

(d) tangente de  $2\pi$ ;

(e) logaritmo de 4 na base 2;

(f) logaritmo de 'e' ao quadrado na base 'e', sendo 'e' o número Naperiano;

(g) some as frações  $\frac{1}{2}$  mais  $\frac{2}{5}$  usando `import fractions` e o construtor `Fraction(,)`;

**Exercício 1.12** - Usando a biblioteca '`math`', calcule as funções intrínsecas  $\sinh(x)$ ,  $0.5 * (\exp(x) - \exp(-x))$ ,  $0.5 * (e^{**x} - e^{**(-x)})$ , colocando o valor de  $2\pi$ .

**Exercício 1.13** - Uma partícula é lançada para cima com velocidade inicial de 5m/s. Encontre sua posição quando o tempo for 0.6s, após o lançamento.

**Exercício 1.14** - Faça a conversão de 21 graus Celsius para Fahrenheit.

**Exercício 1.15** - Encontre os tempos, em que um corpo sendo lançado do chão para cima, com velocidade inicial de 5m/s, e esteja a uma altura de 0.2 m. Dica: a princípio teremos dois tempos. Estude o problema fisicamente e computacionalmente.

**Exercício 1.16** – Seja o lançamento oblíquo de uma partícula para as condições abaixo.

`g = 9.81 # m/s^2`

`v0 = 15 # km/h`

```
theta = 60 # graus  
x0 = 0 # m  
y0 = 1 # m
```

Encontre a posição vertical para  $x = 0.5$  m.

**Exercício 1.17** Teste o operador lógico “not” (não) a partir tabela verdade e teste no Python.

- (a) not True
- (b) not False

**Exercício 1.18** Teste o operador lógico “and” (e) a partir tabela verdade e teste no Python.

- (a) True and True
- (b) True and False
- (c) False and False

**Exercício 1.19** Teste o operador lógico “or” (ou) a partir tabela verdade e teste no Python.

- (a) True or True
- (b) True or False
- (c) False or True
- (d) False or False

**Exercício 1.20** Faça um programa que calcule o tempo de viagem de um carro em horas, quando tem-se a distância a se percorrer e a velocidade média esperada na viagem. Imprima o tempo discriminado em horas, minutos e segundos.

**Exercício 1.21** Um objeto viaja a 20 m/s ao longo de uma estrada sem atrito. Em um determinado instante ele entra em uma região onde o atrito cinético é 0,5. Encontre a distância que ele percorre dentro desta região até parar.

**Exercício 1.22** Escreva um programa que pergunte a quantidade de km percorridos por um carro alugado pelo usuário, assim como a quantidade de dias pelos quais o carro foi alugado. Calcule o preço a pagar, sabendo que o carro custa R\$ 60 por dia e R\$ 0,15 por km rodado.

**Exercício 1.23.** Faça um programa que calcule a hipotenusa de um triangulo Pitagórico dando a entrada dos lados.

**Exercício 1.24** Faça um código que converta um ângulo de graus e radianos, sem usar o pacote math.

**Exercício 1.25.** Elabore um programa que encontre a distância entre dois pontos no espaço.

**Exercício 1.26.** Um carro sai da cidade A até B, que dista 90km, parando durante 30min num posto na metade do caminho à beira da estrada, para refeição e abastecimento. De A até o posto gasta 1h30min, fazendo o percurso do posto a B em mais 30min. Faça um programa que calcule a velocidade escalar média (km/h) na viagem.

**Exercício 1.27.** Calcule a massa convertida a partir da energia de uma estação de energia nuclear, cuja potencia seja de 400MW (mega Watts) em um ano de produção. Use a relação de energia de Einstein,  $E = mc^2$ .

**Exercício 1.28.** Elabore um programa sobre lançamento de uma partícula, dando como entrada o ângulo de lançamento com a horizontal em graus, a posição inicial em x e y e a velocidade inicial em módulo. Dê como resultado o tempo até a altura máxima, a altura máxima, o tempo no alcance máximo no solo e o alcance máximo.

**Exercício 1.29.** Seja um corpo deslizando sob uma região com atrito de coeficiente 0,4. Elabore um programa que se entre com a velocidade inicial de 15m/s e a distância percorrida de 2m, e se obtenha ao final desta velocidade final em (m/s). Agora use o comando `input` para o usuário entrar com os dados.

**Exercício 1.30.** Faça um programa que calcule o tempo de decaimento radioativo do Carbono 14 a partir da equação  $Q(t) = Q_0 \exp(-\lambda t)$ , dado  $\lambda = 0.00012097/\text{ano}$ , que representa a constante de decaimento do Carbono 14.

**Exercício 1.31.** Escreva um programa que calcule a velocidade de escape para corpos em diferentes corpos celestes: (a) Terra, (b) Lua, (c) Planeta anão Ceres e (d) Jupiter.

Body	Mass (kg)	Radius (m)
Earth	$6.0 \times 10^{24}$	$6.4 \times 10^6$
Moon	$7.4 \times 10^{22}$	$1.7 \times 10^6$
Ceres	$8.7 \times 10^{20}$	$4.7 \times 10^5$
Jupiter	$1.9 \times 10^{27}$	$7.1 \times 10^7$

**Exercício 1.32** A partir das listas:

```
pets = [ 'gato', 'cachorro', 'porco'],  
coisas = ['caixa', 2, [3, 4], 3.9],  
vetor = [23.99, 19.99, 34.50, 120.99],
```

imprima, analise e interprete os resultados:

- (a) `pets[0]`, `pets[2]`, `pets[4]`
- (b) `coisas[0]`, `coisas[3]`
- (c) `coisas[-2]`
- (d) tipo da lista `pets`
- (e) `len(pets)`
- (f) somar as listas
- (g) duplicar a lista `pets`
- (h) verificar se `cachorro` está em `pets`
- (i) mínimo da lista `vetor`
- (j) máximo da lista `vetor`
- (k) somar todos os elementos da lista `vetor`.

**Exercício 1.33** Defina as tupla:

```
dias = ('Seg', 'Ter', 'Qua'), final = ('Sab', 'Dom').
```

Obs. Tuplas são listas imutáveis ou de valores constantes.

- (a) Defina uma nova tupla e imprima: `meiosemana = dias + ('Qui', 'Sex')`
- (b) Defina a nova tupla `semana = meiosemana + final`, e imprima.
- (c) Tipo de variável de `semana`.

**Exercício 1.34** Defina as listas:

```
numero = [6, 9, 4, 22]
```

e



```
animais = [ 'gato', 'cachorro', 'porco' ]
```

e imprima a aplicação dos métodos:

- (a) `numero.append(32);`
- (b) `animais.append('vaca');`
- (c) `animais.remove(cachorro);`
- (d) `numero.reverse();`
- (e) `numero.sort()` e `animais.sort();`
- (f) Defina uma nova lista `animais2 = [ 'gato', 'gato', 'cachorro', 'porco' ]`  
e imprima o método `animais2.count('gato')`.