

# ALGORITMOS E PROGRAMAÇÃO COM PYTHON

## 1. INTRODUÇÃO À PROGRAMAÇÃO COM PYTHON

### Programas de Computador

Os **programas de computadores** são algoritmos escritos em linguagens de programação como Python, que são interpretados e executados por uma máquina. Diferente dos algoritmos da vida real, os programas devem ser precisos e sem ambiguidades.

### Linguagem Python

Python é uma linguagem de programação de alto nível, interpretada, com sintaxe clara e intuitiva. É excelente para iniciantes por sua legibilidade e versatilidade.

## 2. VARIÁVEIS E CONSTANTES EM PYTHON

### Variáveis

Em Python, as variáveis são criadas simplesmente atribuindo valores:

```
python - variaveis.py
```

```
1  # Exemplos de declaração de variáveis
2  nome_variavel = "texto"
3  numero_inteiro = 42
4  numero_real = 3.14
5  valor_logico = True
```

### Constantes

Por convenção, constantes são escritas em MAIÚSCULAS:

```
python - constantes.py
```

```
1  # Constantes (convenção)
2  NOME_DA_CONSTANTE = 3
3  PI = 3.14159
4  TAXA_JUROS = 0.15
```

### 3. TIPOS DE DADOS EM PYTHON

python - tipos.py

```
1  # String (cadeia de caracteres)
2  nome = "Gallo"
3  mensagem = 'Computação é coisa linda!'
4
5  # Caractere (em Python é string de tamanho 1)
6  letra = 'a'
7
8  # Inteiro
9  numero = 5
10 idade = 25
11
12 # Float (real)
13 preco = 1.99
14 altura = 1.75
15
16 # Boolean (lógico)
17 fim_da_aula = False
18 ativo = True
19
20 # None (vazio/nulo)
21 resultado = None
```

### 4. ENTRADA E SAÍDA DE DADOS

#### Saída - print()

python - saida.py

```
1  # Escreve no console
2  print("Oi mundo!")
3
4  # Escreve variáveis e textos
5  mensagem = "Computação é coisa linda!!!!"
6  print(mensagem)
7
8  # Múltiplos valores
9  nome = "João"
10 idade = 20
11 print("Nome:", nome, "Idade:", idade)
12
13 # Formatação moderna (f-strings)
14 print(f"Olá {nome}, você tem {idade} anos")
```

## Entrada - input()

python - entrada.py

```
1  # Lendo dados do usuário
2  nome = input("Digite seu nome: ")
3  idade = int(input("Digite sua idade: "))
4  altura = float(input("Digite sua altura: "))
5
6  print(f"Olá {nome}, você tem {idade} anos e {altura}m de altura")
```

## Limpar console

python - limpar-console.py

```
1  import os
2
3  def limpar_console():
4      os.system('cls' if os.name == 'nt' else 'clear')
5
6  # Uso
7  limpar_console()
```

## 5. OPERADORES EM PYTHON

### Operadores Aritméticos

python - operadores-aritmeticos.py

```
1  # Operações básicas
2  a = 10
3  b = 3
4
5  soma = a + b          # 13
6  subtracao = a - b     # 7
7  multiplicacao = a * b # 30
8  divisao = a / b       # 3.333...
9  divisao_inteira = a // b # 3
10 resto = a % b         # 1
11 potencia = a ** b     # 1000
12
13 # Precedência: ** → * / // % → + -
14 resultado = 6 + 7 * 9 # 69 (multiplicação primeiro)
```

## Operadores Relacionais

python - operadores-relacionais.py

```
1  # Comparações (retornam True ou False)
2  print(3 > 4)      # False
3  print(7 != 7)     # False
4  print(9 == 10 - 1) # True
5  print(33 <= 100)  # True
6  print(6 >= 5 + 1) # True
```

## Operadores Lógicos

python - operadores-logicos.py

```
1  # and (e), or (ou), not (não)
2  print(5 > 3 and 2 < 1)    # False
3  print(1 > 3 or 1 <= 1)    # True
4  print(not (8 < 4))        # True
5
6  # Precedência: not → and → or
```

## 6. ESTRUTURAS CONDICIONAIS

### if simples

python - if-simples.py

```
1  numero = int(input("Digite um número: "))
2
3  if numero == 0:
4      print("O número digitado é 0")
```

### if-else

python - if-else.py

```
1  numero = int(input("Digite um número: "))
2
3  if numero == 0:
4      print("Impossível dividir por zero")
5  else:
6      print(f"20 dividido por {numero} = {20/numero}")
```

## if-elif-else

python - if-elif.py

```
1  nota = float(input("Digite a nota: "))
2
3  if nota >= 7:
4      print("Aluno aprovado")
5  elif nota >= 4:
6      print("Aluno em recuperação")
7  else:
8      print("Aluno reprovado")
```

## 7. ESTRUTURAS DE REPETIÇÃO

### while (pré-testado)

python - while.py

```
1  # Executa enquanto a condição for True
2  parar = 'N'
3  while parar != 'S':
4      print("Olá Mundo!")
5      parar = input("Deseja parar o laço? (S/N): ").upper()
```

### do-while (simulado)

python - do-while.py

```
1  # Python não tem do-while nativo, mas podemos simular
2  while True:
3      lado = float(input("Informe o valor do lado do quadrado: "))
4      if lado > 0:
5          break
6      print("Valor inválido! Tente novamente.")
7
8  print(f"A área do quadrado é {lado * lado}")
```

### for (com variável de controle)

python - for.py

```
1  # Range: range(início, fim, passo)
2  numero = int(input("Digite um número: "))
3
4  for i in range(11): # 0 a 10
5      print(f"{numero} x {i} = {numero * i}")
```

## 8. VETORES (LISTAS) EM PYTHON

### Declaração e uso

python - vetores.py

```
1  # Listas (vetores)
2  notas = [7, 8, 5, 8, 10]
3  nomes = ["Ana", "João", "Maria"]
4
5  # Acesso por índice (começa em 0)
6  print(notas[0])    # 7 (primeiro elemento)
7  print(notas[4])    # 10 (último elemento)
8
9  # Modificação
10 notas[0] = 9
11 print(notas)        # [9, 8, 5, 8, 10]
12
13 # Lista vazia
14 vetor = []
15
16 # Lista com diferentes tipos (Python permite)
17 mista = [1, "texto", 3.14, True]
```

### Operações com listas

python - lista.py

```
1  # Adicionar elementos
2  notas = [1, 2, 3]
3  notas.append(4)      # [1, 2, 3, 4]
4  notas.insert(1, 10)  # [1, 10, 2, 3, 4]
5
6  # Remover elementos
7  notas.remove(10)     # [1, 2, 3, 4]
8  ultimo = notas.pop() # ultimo = 4, notas = [1, 2, 3]
9
10 # Tamanho da lista
11 tamanho = len(notas) # 3
12
13 # Percorrer lista
14 for nota in notas:
15     print(nota)
16
17 for indice, nota in enumerate(notas):
18     print(f"Nota {indice}: {nota}")
```

## 9. MATRIZES (LISTAS 2D) EM PYTHON

### Declaração e uso

python - matrizes.py

```
1  # Matriz 4x3 (4 alunos, 3 notas cada)
2  notas_alunos = [
3      [10, 9, 7],      # Aluno 0
4      [6, 8, 10],      # Aluno 1
5      [8, 7, 5],      # Aluno 2
6      [5, 3, 3]       # Aluno 3
7  ]
8
9  # Acesso: matriz[linha][coluna]
10 print(notas_alunos[0][1]) # 9 (primeira linha, segunda coluna)
11
12 # Modificação
13 notas_alunos[0][1] = 8.5
14
15 # Matriz vazia
16 matriz = []
```

### Percorrendo matrizes

python - matrizes2.py

```
1  notas_alunos = [
2      [10, 9, 7],      # Aluno 0
3      [6, 8, 10],      # Aluno 1
4      [8, 7, 5],      # Aluno 2
5      [5, 3, 3]       # Aluno 3
6  ]
7
8  # Percorrer todas as notas
9  for i in range(len(notas_alunos)):
10     for j in range(len(notas_alunos[i])):
11         print(f"Aluno {i}, Nota {j}: {notas_alunos[i][j]}")
12
13 # Forma mais pythonica
14 for linha in notas_alunos:
15     for nota in linha:
16         print(nota, end=" ")
17     print() # nova linha
```

## 10. FUNÇÕES EM PYTHON

### Funções com retorno

python - funcao.py

```
1 def calcular_media(nota1, nota2, nota3):
2     """Calcula a média de três notas"""
3     media = (nota1 + nota2 + nota3) / 3
4     return media
5
6 # Uso
7 media_aluno = calcular_media(7, 8, 9)
8 print(f"Média: {media_aluno:.2f}")
```

### Funções sem retorno (procedimentos)

python - procedimento.py

```
1 def mostrar_mensagem(nome):
2     """Exibe uma mensagem de boas-vindas"""
3     print(f"Olá {nome}, seja bem-vindo!")
4
5 # Uso
6 mostrar_mensagem("Maria")
```

### Função de exemplo: sistema de perguntas

python - funcao-exemplo.py

```
1 def fazer_pergunta(texto):
2     """Faz uma pergunta Sim/Não e retorna True/False"""
3     resposta = input(f"{texto} (S/N): ").upper()
4     return resposta == "S"
5
6 # Uso
7 if fazer_pergunta("Acabou a aula?"):
8     print("Vamos embora")
9 else:
10    print("Trabalhemmmm")
```



## Funções com parâmetros opcionais

python - funcao-param-opc.py

```
1 def saudacao(nome, mensagem="Olá"):
2     """Saudação personalizada"""
3     return f"{mensagem}, {nome}!"
4
5 print(saudacao("João"))           # Olá, João!
6 print(saudacao("Maria", "Bem-vindo")) # Bem-vindo, Maria!
```

## 11. EXEMPLOS COMPLETOS

### Calculadora de médias

python - calc-media.py

```
1 def calcular_media_turma():
2     """Calcula a média de uma turma"""
3     quantidade = int(input("Quantos alunos na turma? "))
4     notas_turma = []
5
6     for i in range(quantidade):
7         nota = float(input(f"Nota do aluno {i+1}: "))
8         notas_turma.append(nota)
9
10    media = sum(notas_turma) / len(notas_turma)
11    print(f"Média da turma: {media:.2f}")
12
13    # Mostrar situação de cada aluno
14    for i, nota in enumerate(notas_turma):
15        situacao = "Aprovado" if nota >= 7 else "Reprovado"
16        print(f"Aluno {i+1}: {nota} - {situacao}")
17
18    calcular_media_turma()
```

## Jogo da velha simples

python - jogo-velha.py

```
1  def criar_tabuleiro():
2      """Cria um tabuleiro 3x3 vazio"""
3      return [[' ' for _ in range(3)] for _ in range(3)]
4
5  def mostrar_tabuleiro(tabuleiro):
6      """Exibe o tabuleiro"""
7      for linha in tabuleiro:
8          print('|'.join(linha))
9          print('-' * 5)
10
11 # Exemplo de uso
12 jogo = criar_tabuleiro()
13 jogo[0][0] = 'X'
14 jogo[1][1] = 'O'
15 mostrar_tabuleiro(jogo)
```