

# Introdução à Programação com Python

## Sintaxe Básica

Claudio Scheer

9 de agosto de 2025

# Por que Python?

- Simples e legível: código fácil de entender e escrever.
- Sintaxe direta: menos símbolos, mais significado.
- Muito usado: do iniciante ao profissional.
- Comunidade grande e muitos exemplos.

## Ideia

Uma instrução faz algo. Podemos guardar o resultado em uma variável.

```
1 mensagem = "Olá"           # guarda um texto
2 numero = 10                 # guarda um número
3 nome = input("Seu nome: ") # pede um texto
4
```

**Padrão:** `nome_variavel = valor` ou `nome_variavel = instrução()`

# Funções: Definição

## O que é uma função?

Um “pacotinho” de passos para fazer uma tarefa.

Indentação (espaços no começo da linha) mostra o que faz parte da função.

```
1 def faz_algo():  
2     nome = input("Seu nome: ") # dentro da função (indentado)  
3     return nome                # retorna o nome  
4  
5 def soma(a, b):  
6     resultado = a + b # dentro da função  
7     return resultado  
8
```

# Funções: Exemplos do Python

- `print`: mostra um valor na tela.
- `input`: pede para digitar um valor.

```
1 print("Oi!") # mostra um texto
2 texto = input("Digite algo: ") # lê um texto
3 print("Você digitou:", texto)
4
```

# Tipos de Dados (exemplos simples)

```
1 idade = 5                # número inteiro
2 altura = 1.70            # número decimal
3 nome = "Ana"             # texto (string)
4 ligado = True            # booleano (True ou False)
5 numeros = [1, 2, 3]      # lista
6 pessoa = {"nome": "Ana", "idade": 5} # dicionário
7
```

# Contas (Operadores Aritméticos)

Operador	Descrição	Exemplo
+	Somar	<code>x + y</code>
-	Subtrair	<code>x - y</code>
*	Multiplicar	<code>x * y</code>
/	Dividir	<code>x / y</code>
%	Resto da divisão	<code>x % y</code>
**	Exponenciação	<code>x ** y</code>

```
1 a = 7
2 b = 3
3 soma = a + b      # 10
4 subtracao = a - b # 4
5 produto = a * b   # 21
6 divisao = a / b   # 2.333...
7 resto = a % b     # 1
8 potencia = a ** b # 343
```

# Comparações (Operadores Lógicos)

Operador	Descrição	Exemplo
>	maior que	<code>x &gt; y</code>
<	menor que	<code>x &lt; y</code>
==	igual a	<code>x == y</code>
!=	diferente de	<code>x != y</code>
>=	maior ou igual	<code>x &gt;= y</code>
<=	menor ou igual	<code>x &lt;= y</code>

```
1 x = 10
2 y = 20
3 eh_igual = (x == y)      # False
4 eh_maior = (x > y)       # False
5 eh_menor_ou_igual = (x <= y) # True
6
```



# Controle de Fluxo: if/else

## Ideia

Escolhe o que fazer com base em uma condição.

```
1 idade = 18
2 if idade >= 18:
3     mensagem = "maior de idade"
4 else:
5     mensagem = "menor de idade"
6 print(mensagem)
7
```

# Somar dois números (passos)

- 1 Pedir ao usuário um número entre 1 e 20.
- 2 Pedir ao usuário um segundo número entre 1 e 20.
- 3 Somar os dois números.
- 4 Mostrar o resultado.

# Somar dois números (exemplo)

```
1 primeiro = int(input("Digite um número entre 1 e 20: "))
2 segundo = int(input("Digite outro número entre 1 e 20: "))
3
4 total = primeiro + segundo
5 print("A soma é:", total)
6
```

Resposta: @exercicios/soma-inteiros.py

# Atividade (Placar do Jogo)

## Desafio

O placar foi **3 x 2** para o time A, contra o time B.

- O jogo empatou?
- Quem ganhou?
- Resposta: @exercicios/verificar-empate.py

*Dica:* compare os placares usando os operadores de comparação.

# Atividade: Operações Básicas

- Descrição: ler dois números e mostrar soma, subtração, multiplicação e divisão.
- Objetivo: praticar operadores aritméticos e exibir resultados.
- Resposta: @exercicios/operacoes-matematicas.py

# Atividade: Calculadora de Idade

- Descrição: pedir ano de nascimento e calcular idade aproximada.
- Objetivo: ler entrada, converter e calcular.
- Resposta: @exercicios/calculadora-idade.py

# Atividade: Bhaskara

- Descrição: ler `a`, `b`, `c` e calcular as raízes da equação do 2º grau.
- Objetivo: usar potência, raiz quadrada e condições para o delta.
- Resposta: `@exercicios/bhaskara.py`

# Atividade: Tabuada

- Descrição: mostrar a tabuada de um número.
- Objetivo: praticar repetição e multiplicação.
- Resposta: @exercicios/tabuada.py



# Atividade: Validar Caracteres

- Descrição: pedir um texto e dizer se tem pelo menos N caracteres.
- Objetivo: trabalhar com strings e `len`.
- Resposta: @exercicios/validar-caracteres.py

# Atividade: Somar Números Pares

- Descrição: somar todos os números pares em um intervalo (ex.: 0 a 100).
- Objetivo: praticar laços, condição de par e acumulação.
- Resposta: @exercicios/somar-pares.py

# Atividade: Verificar Empate

- Descrição: ler placar de A e B; dizer se empatou e quem ganhou.
- Objetivo: comparar valores e imprimir mensagens.
- Resposta: `@exercicios/verificar-empate.py`

# Atividade: Soma de Dois Números

- Descrição: pedir dois inteiros e mostrar a soma.
- Objetivo: ler, converter e somar.
- Resposta: @exercicios/soma-inteiros.py

- Pratique: escreva e execute pequenos programas.
- Leia mensagens de erro com calma.
- Indentação correta é essencial.
- Comece simples e evolua aos poucos.

Obrigado!

Perguntas?