



# Curso Python – Nível 1: Bitwalker



"Aquele que dá os primeiros passos entre zeros e uns."



## Aula 1 – Primeiro Contato com o Python



"O primeiro passo entre zeros e uns."

---



### Objetivo da Aula

- Entender o que é um algoritmo
  - Conhecer o Python como linguagem de programação
  - Escrever e executar o primeiro programa
- 



### Parte 1 – O que é um Algoritmo?

Um algoritmo é uma sequência de passos para resolver um problema.

Exemplo (em português):

Algoritmo para fazer café

1. Esquentar água
2. Colocar pó de café no filtro
3. Jogar a água quente no filtro
4. Aguardar
5. Servir o café



Na programação, fazemos a mesma coisa: descrevemos passos, mas usando uma linguagem que o computador entende.

---



### Parte 2 – O que é o Python?

- É uma linguagem de programação simples e muito usada no mundo todo.
- Criada em 1991 por Guido van Rossum.
- Usada em: inteligência artificial, ciência de dados, web, jogos, automação, robótica.
- Tem como filosofia ser fácil de ler e escrever.

•



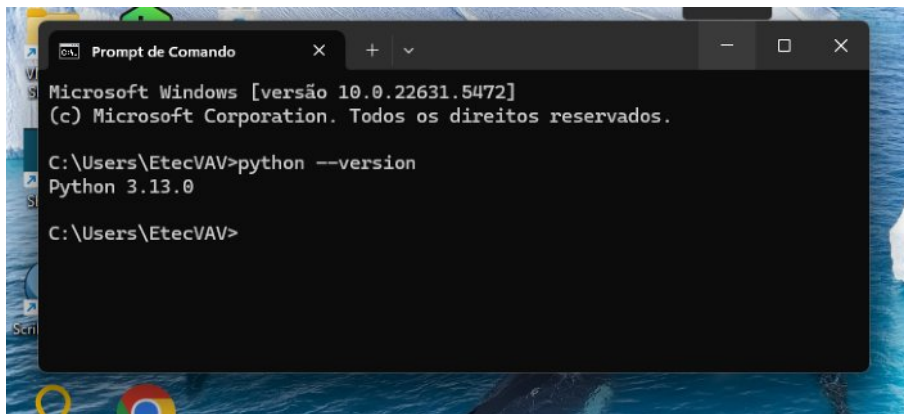
## Parte 3 – Instalando e Usando Python

Site oficial: <https://www.python.org>

Neste site, você encontra link para fazer o download.

Depois de instalado, você pode rodar programas Python de 3 formas:

1.(CMD) Terminal/Prompt de comando→ digitando python ou python3

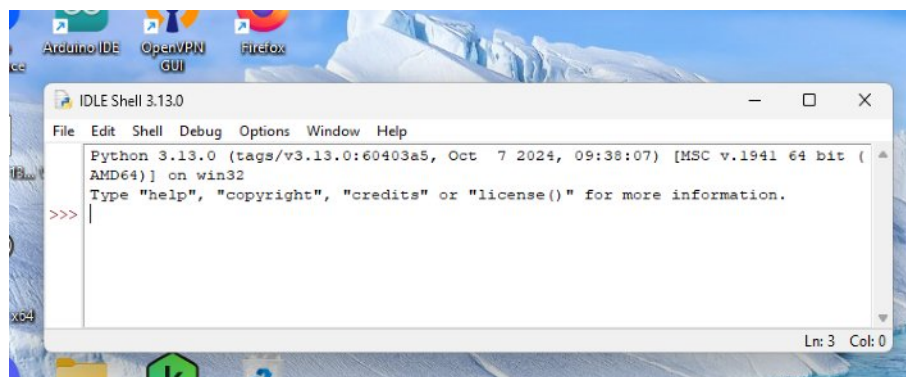
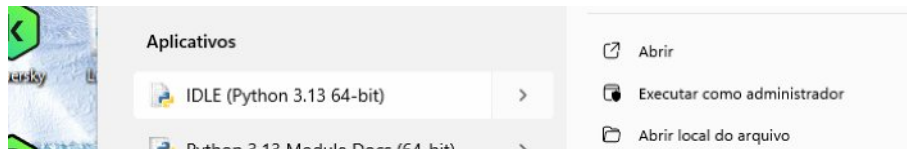


```
Microsoft Windows [versão 10.0.22631.5472]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\EtecVAV>python --version
Python 3.13.0

C:\Users\EtecVAV>
```

2.IDLE (editor que já vem com o Python)



### 3.VS Code (instalando a extensão Python)

---

## Parte 4 – Primeiro Programa

Para criar os programas, você pode usar um editor simples, como o **bloco de notas (Notepad)** ou o **VisualCode**.

Crie um arquivo chamado **primeiro.py** com o seguinte conteúdo:


```
print("Olá, mundo!")
```

Agora, execute:

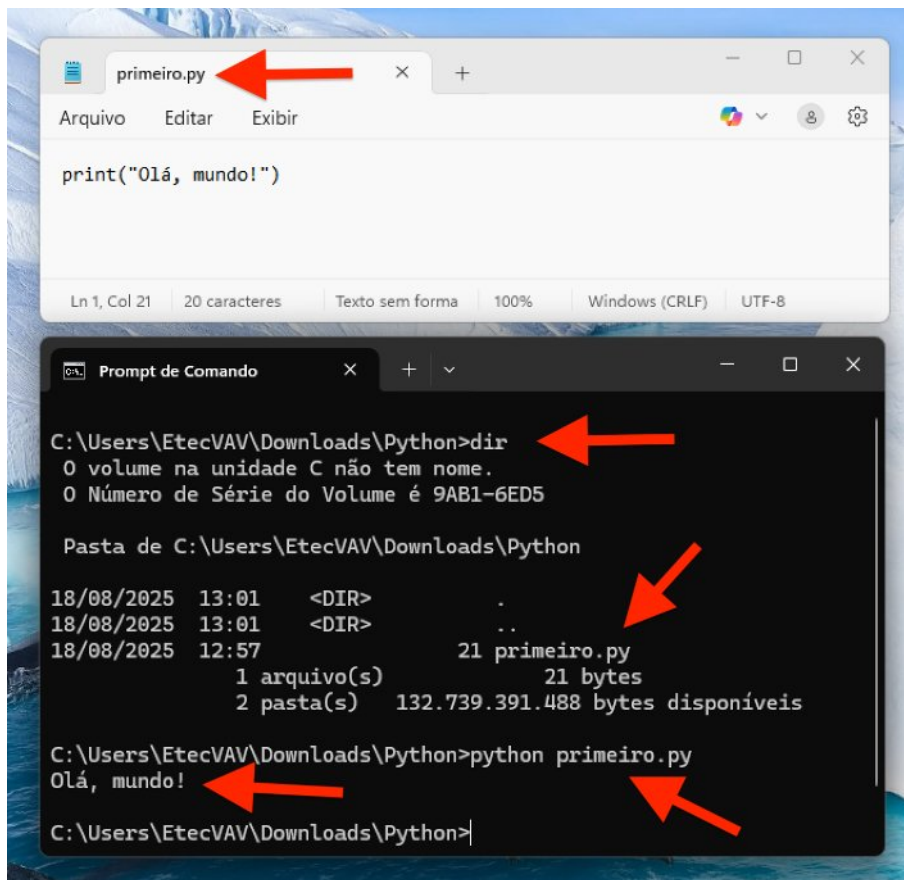
- No terminal:

```
python primeiro.py
```

- Ou abra o arquivo no IDLE e clique em Run → Run Module (F5).

 Saída esperada:

Olá, mundo!



```
C:\Users\EtecVAV\Downloads\Python>dir
O volume na unidade C não tem nome.
O Número de Série do Volume é 9AB1-6ED5

Pasta de C:\Users\EtecVAV\Downloads\Python

18/08/2025 13:01 <DIR> .
18/08/2025 13:01 <DIR> ..
18/08/2025 12:57           21 primeiro.py
                   1 arquivo(s)          21 bytes
                   2 pasta(s) 132.739.391.488 bytes disponíveis

C:\Users\EtecVAV\Downloads\Python>python primeiro.py
Olá, mundo!
```



## Exercício Resolvido

Escreva um programa que exiba uma mensagem de apresentação:

```
print("Meu nome é Zahroniel e estou começando a aprender Python!")
```

Saída:

Meu nome é Zahroniel e estou começando a aprender Python!

---



## Exercício Proposto

Escreva um programa que exiba:

Olá, **[seu nome]**! Seja bem-vindo ao mundo do Python.



Dica: use o `print()` e substitua **[seu nome]** pelo seu nome verdadeiro.

---



## Conclusão da Aula

- Aprendemos que programar = escrever algoritmos para o computador
- Descobrimos que o Python é uma linguagem simples e poderosa
- Fizemos nosso primeiro programa usando `print()`



# Aula 2 – Variáveis e Dados Básicos



"Guardando pequenas caixas de informação no computador."

---



## Objetivo da Aula

- Entender o que são variáveis
  - Trabalhar com diferentes tipos de dados
  - Usar entrada (input) e saída (print)
- 



## Parte 1 – O que são Variáveis?

Uma variável é como uma caixa com nome, onde podemos guardar informações.

- A caixa tem nome (identificador)
- A caixa guarda um valor

Exemplo:

```
nome = "Zaion"  
idade = 17
```



Aqui criamos duas variáveis:

- nome → guarda o texto "Zaion"
  - idade → guarda o número 17
- 



## Parte 2 – Tipos de Dados Básicos no Python

1. Texto (string) → str

```
mensagem = "Olá, mundo!"
```

2. Número inteiro → int

```
ano = 2025
```

3. Número decimal → float

```
pi = 3.14
```



O Python descobre o tipo automaticamente, não precisamos declarar.

---



## Parte 3 – Entrada de Dados (input)

O comando `input()` permite perguntar algo ao usuário.

```
nome = input("Digite seu nome: ")  
print("Olá,", nome)
```



Se digitarmos Kael, a saída será:

```
Digite seu nome: Kael  
Olá, Kael
```

The screenshot shows a Windows desktop with a blue background. In the foreground, there is a black command prompt window and a white Notepad window. The Notepad window, titled 'segundo.py', contains the following Python code:

```
nome = input("Digite seu nome: ")  
print("Olá,", nome)
```

The command prompt window, titled 'Prompt de Comando', shows the following output:

```
C:\Users\EtecVAV\Downloads\Python>dir  
O volume na unidade C não tem nome.  
O Número de Série do Volume é 9AB1-6ED5  
  
Pasta de C:\Users\EtecVAV\Downloads\Python  
  
18/08/2025 13:31 <DIR> .  
18/08/2025 13:01 <DIR> ..  
18/08/2025 12:57 21 primeiro.py  
18/08/2025 13:31 55 segundo.py  
                76 bytes  
2 arquivo(s) 132.724.649.984 bytes disponíveis  
2 pasta(s)  
  
C:\Users\EtecVAV\Downloads\Python>python segundo.py  
Digite seu nome: Kael  
Olá, Kael  
  
C:\Users\EtecVAV\Downloads\Python>
```

Red arrows point to the file name 'segundo.py' in the Notepad window, the 'dir' command in the command prompt, the file 'segundo.py' in the directory listing, the 'python segundo.py' command, and the output 'Olá, Kael'.



## Parte 4 – Saída de Dados (print)

O `print()` exibe informações na tela. Podemos juntar textos e variáveis.

```
nome = "Alin"  
idade = 12  
print("Meu nome é ", nome, " e eu tenho ", idade, " anos.")
```



Saída esperada:

Meu nome é Alin e eu tenho 12 anos.

**Outra forma é usar f-strings (mais moderna e limpa):**

```
print(f'Meu nome é {nome} e eu tenho {idade} anos.')
```



Saída esperada:

Meu nome é Alin e eu tenho 12 anos.

---



## Exercício Resolvido

Programa que pergunta o nome e a idade:

```
nome = input("Qual é o seu nome? ")
idade = input("Quantos anos você tem? ")

print(f'Olá, {nome}! Você tem {idade} anos.')
```

---



## Exercícios Propostos

1. Escreva um programa que peça o nome do usuário e mostre uma mensagem:

Seja bem-vindo, [nome]!

2. Peça a idade do usuário e mostre quantos anos ele terá daqui a 5 anos.

(Dica: use `int()` para transformar o texto em número)

Exemplo 1:

```
idade = input("Digite sua idade: ")
print("Daqui a 5 anos você terá", int(idade) + 5, "anos.")
```

Exemplo 2:

```
idade = int(input("Digite sua idade: "))
print("Daqui a 5 anos você terá", idade + 5, "anos.")
```

No Exemplo 1, a idade é um texto (string) que é convertida em inteiro antes de fazer a soma. No Exemplo 2, o texto lido é convertido e armazenado na variável idade.

3. Peça a idade do usuário e informe que ano ele nasceu. (`AnoAtal - idade`)



## Conclusão da Aula

- Variáveis guardam informações que podem mudar
- Tipos básicos: texto (str), inteiro (int) e decimal (float)
- `input()` recebe dados, `print()` mostra na tela



Na próxima aula, vamos dar um passo a mais e trabalhar com operadores e expressões (somar, comparar e usar lógica com as variáveis).





# Aula 3 – Operadores e Expressões



"Fazendo o computador calcular e pensar um pouco."

---



## Objetivo da Aula

- Usar **operadores matemáticos**
  - Comparar valores com **operadores relacionais**
  - Combinar condições com **operadores lógicos**
  - Resolver pequenos problemas usando expressões
- 



## Parte 1 – Operadores Aritméticos

Servem para fazer contas.

Operador	Exemplo	Resultado
+	5 + 2	7
-	5 - 2	3
*	5 * 2	10
/	5 / 2	2.5
//	5 // 2	2 (divisão inteira)
%	5 % 2	1 (resto)
**	2 ** 3	8 (potência)

Exemplo:

```
a = 10
b = 3
print(a + b)
print(a / b)
print(a // b)
print(a % b)
```



Saída esperada:

13

3.333...

3

1



## Parte 2 – Operadores de Comparação (Relacionais)

Usados para verificar condições.

Operador	Exemplo	Resultado
==	5 == 5	True
!=	5 != 3	True
<	3 < 5	True
>	5 > 3	True
<=	5 <= 5	True
>=	5 >= 6	False

True -> Verdadeiro  
False -> Falso

```
idade = 18  
print(idade >= 18)
```



Saída esperada:

True

```
idade = 15  
print(idade >= 18)
```



Saída esperada:

False



## Parte 3 – Operadores Lógicos

Permitem combinar condições.

Operador	Exemplo	Resultado
and	(5 > 2) and (5 < 10)	True
or	(5 > 10) or (5 < 2)	False
not	not(5 > 2)	False

```
idade = 20  
tem_carteira = True
```

```
print(idade >= 18 and tem_carteira)
```



Saída esperada:

True



## Parte 4 – Expressões Completas

Use # para fazer comentarios no programas.

Podemos misturar operações:

```
# Este programa calcula a média de duas notas
```

```
nota1 = 7
```

```
nota2 = 8
```

```
media = (nota1 + nota2) / 2
```

```
print(media > 6) # Saída esperada: True
```

---



## Exercício Resolvido

Faça um programa que peça dois números e mostre:

- A soma
- A multiplicação
- A divisão inteira
- O resto da divisão

```
# Programa: calcula.py
```

```
# Este programa pergunta ao usuário dois números e informa:
```

```
# O resultado da soma, multiplicação, divisão inteira e o resto da divisão
```

```
a = int(input("Digite o primeiro número: "))
```

```
b = int(input("Digite o segundo número: "))
```

```
print("Soma:", a + b)
```

```
print("Multiplicação:", a * b)
```

```
print("Divisão inteira:", a // b)
```

```
print("Resto da divisão:", a % b)
```

Exemplo: números 2 e 4



Saída esperada:

```
C:\Users\EtecVAV\Downloads\Python>python segundo.py
```

Digite o primeiro número: 4

Digite o segundo número: 3

Soma: 7

Multiplicação: 12

Divisão inteira: 1

Resto da divisão: 1



## Exercícios Propostos

1. Peça a idade de uma pessoa e mostre se ela é maior de idade (18 anos ou mais).



Saída esperada:

```
C:\Users\EtecVAV\Downloads\Python>python MaiorIdade.py
```

Digite sua idade: 15

Maior de idade? False

2. Peça dois números e mostre qual é o maior.



Saída esperada:

```
C:\Users\EtecVAV\Downloads\Python>python MaiorNumero.py
```

Digite o primeiro número: 3

Digite o segundo número: 5

a > b? False

b > a? True

a == b? False

3. Pergunte se a pessoa tem **mais de 18 anos** e se **possui carteira de motorista**. Mostre **Pode dirigir** ou **Não pode dirigir**.



Saída esperada:

```
C:\Users\EtecVAV\Downloads\Python>python CarteiraV1.py
```

Digite sua idade: 19

Possui CNH? (s/n): s

Pode dirigir? True

```
C:\Users\EtecVAV\Downloads\Python>python CarteiraV1.py
```

Digite sua idade: 19

Possui CNH? (s/n): n

Pode dirigir? False

```
C:\Users\EtecVAV\Downloads\Python>python CarteiraV1.py
```

Digite sua idade: 15

Possui CNH? (s/n): n

Pode dirigir? False

4. Peça duas notas de um aluno, calcule a média e diga se ele foi aprovado (média  $\geq 6$ ).



Saída esperada:

```
C:\Users\EtecVAV\Downloads\Python>python MediaNota.py
```

Digite a primeira nota: 7

Digite a segunda nota: 2

Média: 4.5

Aprovado? False

```
C:\Users\EtecVAV\Downloads\Python>python MediaNota.py
```

Digite a primeira nota: 7

Digite a segunda nota: 9

Média: 8.0

Aprovado? True



## Conclusão da Aula

- Aprendemos a usar operadores matemáticos, relacionais e lógicos
  - Criamos expressões para cálculos e condições
  - Já conseguimos resolver pequenos problemas do dia a dia
-

📖 Na **Aula 4**, o Bitwalker vai aprender sobre **Estrutura Sequencial**: a ordem das instruções e como transformar isso em programas simples como calcular médias e conversões.



# Aula 4 – Estrutura Sequencial



"Linha após linha, o programa segue o caminho."

---



## Objetivo da Aula

- Entender que o programa segue uma ordem de cima para baixo
  - Criar programas sequenciais (sem condições ou repetições ainda)
  - Usar cálculos e textos na mesma lógica
  - Aprender a comentar o código
- 



## Parte 1 – O que é Estrutura Sequencial?

- É a forma mais simples de programa: as instruções são executadas uma após a outra, na ordem escrita.
- Cada linha só roda depois da anterior.

Exemplo:

```
print("Início do programa")
a = 10
b = 20
soma = a + b
print("A soma é:", soma)
print("Fim do programa")
```



Saída:

```
Início do programa
A soma é: 30
Fim do programa
```

---



## Parte 2 – Comentários no Código

Servem para explicar o que você está fazendo. O Python ignora tudo que está depois do #.

```
# Este programa calcula a soma de dois números
a = 5
b = 3
print(a + b) # Mostra o resultado
```

---



## Parte 3 – Exemplos Simples

### Exemplo 1 – Cálculo de Média

```
nota1 = float(input("Digite a primeira nota: "))
nota2 = float(input("Digite a segunda nota: "))
nota3 = float(input("Digite a terceira nota: "))
```

```
media = (nota1 + nota2 + nota3) / 3
```

```
print(f"A média do aluno é {media:.2f}")
```

---

### Exemplo 2 – Conversor de Temperatura

```
celsius = float(input("Digite a temperatura em Celsius: "))
fahrenheit = (celsius * 9/5) + 32
```

```
print(f"{celsius}°C equivalem a {fahrenheit}°F")
```

---

### Exemplo 3 – Conversão de Tempo

```
minutos = int(input("Digite a quantidade de minutos: "))
segundos = minutos * 60
```

```
print(f"{minutos} minutos equivalem a {segundos} segundos")
```

---



## Exercício Resolvido

Problema: Escreva um programa que peça dois números, some-os e mostre o resultado.

```
a = float(input("Digite o primeiro número: "))
b = float(input("Digite o segundo número: "))
soma = a + b
```

```
print(f"A soma de {a} e {b} é {soma}")
```

---



## Exercícios Propostos

1. Peça 3 notas de um aluno, calcule a média e exiba o resultado com 2 casas decimais.
2. Faça um programa que converta um valor em reais (R\$) para dólares (US\$).  
(Dica: peça o valor em reais e a cotação do dólar)
3. Escreva um programa que peça a idade de uma pessoa em anos e mostre quantos dias ela já viveu (aproximadamente, sem considerar anos bissextos).
4. Crie um programa que leia o preço de um produto e mostre o preço com 10% de desconto.
5. Faça um programa que leia a altura e o peso de uma pessoa e calcule o IMC:



$$\text{IMC} = \text{peso} / (\text{altura} * \text{altura})$$

---



## Conclusão da Aula

- O Python executa as instruções em sequência
  - Aprendemos a usar comentários para explicar o código
  - Criamos programas simples e úteis com cálculos diretos
- 



Na Aula 5, o Bitwalker vai descobrir como lidar com os primeiros erros e acertos — entendendo mensagens de erro e aprendendo a corrigi-las.



# Aula 5 – Lidando com Erros



"Cada erro é um professor disfarçado."

---



## Objetivo da Aula

- Reconhecer que errar é natural ao programar
  - Identificar os tipos mais comuns de erros no Python
  - Aprender a ler mensagens de erro
  - Corrigir pequenos problemas no código
- 



## Parte 1 – Por que os erros acontecem?

- O computador segue as instruções ao pé da letra.
- Se houver algo escrito errado ou uma ordem incorreta, o Python avisa.
- Esses avisos aparecem como mensagens de erro.



Importante: erros não são inimigos, mas guias.

---



## Parte 2 – Tipos de Erros



### 1. Erro de Sintaxe (SyntaxError)

Acontece quando escrevemos algo fora da gramática do Python.

Exemplo errado:

```
print("Olá, mundo!" # faltou fechar o parêntese
```

Mensagem do Python:

```
SyntaxError: unexpected EOF while parsing
```

---



### 2. Erro de Nome (NameError)

Acontece quando usamos uma variável que não foi criada.

Exemplo errado:

```
print(nome)
```

Mensagem do Python:

NameError: name 'nome' is not defined

---

### 3. Erro de Tipo (TypeError)

Acontece quando tentamos misturar tipos de dados incompatíveis.

Exemplo errado:

```
idade = 15  
print("Idade: " + idade) # não pode juntar texto + número diretamente
```

Mensagem do Python:

TypeError: can only concatenate str (not "int") to str

---

### 4. Erro de Valor (ValueError)

Acontece quando damos um valor inválido para uma conversão.

Exemplo errado:

```
numero = int("abc") # não dá para converter "abc" em número
```

Mensagem do Python:


ValueError: invalid literal for int() with base 10: 'abc'

---

## Parte 3 – Como interpretar erros?

Quando aparece um erro, o Python mostra:

1. Linha do código onde o problema aconteceu
2. Tipo de erro (SyntaxError, NameError, etc.)
3. Explicação curta do que deu errado

 Ler a mensagem com calma já ajuda a corrigir.

---

## Exercício Resolvido

Provoque um erro e depois corrija.

Código errado:

```
print("Olá, meu nome é" nome)
```

Erro:

SyntaxError: invalid syntax

Código corrigido:

```
print("Olá, meu nome é", "nome")
```

---



## Exercícios Propostos

1. Provoque um SyntaxError e corrija.  
(Exemplo: esquecer o ) no print)
  2. Provoque um NameError usando uma variável inexistente.
  3. Provoque um TypeError tentando somar texto e número.
  4. Provoque um ValueError tentando converter "abc" em número.
  5. Escreva um pequeno texto explicando com suas palavras:  
“O que aprendi sobre erros no Python?”
- 



## Conclusão da Aula

- Erros são normais e fazem parte do aprendizado
  - Tipos principais: SyntaxError, NameError, TypeError, ValueError
  - Cada mensagem de erro é uma pista para corrigir o código
- 



Na Aula 6, vamos encerrar o nível Bitwalker com mini projetos práticos, juntando tudo que aprendemos até agora.



# Aula 6 – Mini Projetos do Bitwalker



"Transformando pequenos passos em criações completas."

---



## Objetivo da Aula

- Consolidar o aprendizado dos níveis anteriores
  - Criar programas completos usando apenas estrutura sequencial
  - Praticar entrada, processamento e saída de dados
  - Ganhar confiança resolvendo problemas do dia a dia
- 



## Mini Projeto 1 – Calculadora Simples

Descrição: programa que lê dois números e mostra:

- soma
- subtração
- multiplicação
- divisão

Código:

```
print("=== Calculadora Simples ===")
a = float(input("Digite o primeiro número: "))
b = float(input("Digite o segundo número: "))

print("Soma:", a + b)
print("Subtração:", a - b)
print("Multiplicação:", a * b)
print("Divisão:", a / b)
```

---



## Mini Projeto 2 – Apresentação Personalizada

Descrição: pergunta o nome e idade, depois mostra uma frase divertida.

Código:

```
print("=== Apresentação ===")
nome = input("Qual é o seu nome? ")
idade = int(input("Quantos anos você tem? "))

print(f"Olá, {nome}! Você tem {idade} anos e está começando sua jornada no Python!")
```

---



## Mini Projeto 3 – Conversor de Temperatura

Descrição: lê uma temperatura em graus Celsius e converte para Fahrenheit.

Fórmula:

$$F = (C * 9/5) + 32$$

Código:

```
print("=== Conversor de Temperatura ===")
celsius = float(input("Digite a temperatura em °C: "))
fahrenheit = (celsius * 9/5) + 32

print(f"{celsius:.1f} °C equivalem a {fahrenheit:.1f} °F")
```

---



## Mini Projeto 4 – Loja com Desconto

Descrição: programa que pede o preço de um produto e mostra com 10% de desconto.

Código:

```
print("=== Loja Pythonica ===")
preco = float(input("Digite o preço do produto (R$): "))
desconto = preco * 0.10
final = preco - desconto

print(f"Preço original: R$ {preco:.2f}")
print(f"Desconto: R$ {desconto:.2f}")
print(f"Preço final: R$ {final:.2f}")
```

---



## Mini Projeto 5 – Calculadora de IMC

Descrição: lê peso e altura, calcula o IMC (Índice de Massa Corporal).

Fórmula:

$$IMC = peso / (altura * altura)$$

Código:

```
print("=== Calculadora de IMC ===")
peso = float(input("Digite seu peso (kg): "))
altura = float(input("Digite sua altura (m): "))

imc = peso / (altura ** 2)

print(f"Seu IMC é {imc:.2f}")
```

---



## Exercícios Finais – Desafios do Bitwalker

1. Crie um programa que peça dois números e mostre:

- O dobro do primeiro
- A metade do segundo
- A soma dos dois

2.Faça um programa que leia o ano de nascimento e calcule a idade atual.

3.Escreva um programa que peça o salário de uma pessoa e mostre quanto seria com 15% de aumento.

4.Crie um programa que peça o nome do aluno e duas notas, depois exiba:

O aluno João teve média 7.5

---



## Conclusão do Nível Bitwalker

- Agora você já sabe:
    - ✓ O que é algoritmo
    - ✓ Como usar variáveis e tipos básicos
    - ✓ Fazer operações matemáticas e lógicas
    - ✓ Executar programas sequenciais
    - ✓ Interpretar erros e corrigi-los
  - Conseguiu criar programas completos do zero!
- 



A partir daqui, o próximo passo do curso será evoluir para o nível seguinte, onde entraremos em estruturas de decisão e repetição.

---

**Assinado por:**

Zahroniel Syrran & Kael'Aran