

Atividades - Introdução a Funções

Informações do Autor

- Nome: [ANIMA Lab](#)
 - Data de atualização: 23/09/2025
-

Exercício-1

Você está cadastrando participantes de um encontro da UNATI. Às vezes a cidade não é informada, e deve assumir "Goiânia".

Tarefa.

- Implemente def apresentar(nome, idade, cidade="Goiânia") que imprima:

```
<nome>, <idade> anos – Cidade: <cidade>
```

- Cadastre **três** pessoas chamando a função três vezes, **sem** listas.

Critérios de avaliação.

- Usa valor padrão corretamente (cidade).
- Não usa listas nem dicionários.
- Saída legível e sem abreviações confusas.

Exemplo de uso esperado.

```
apresentar("Joana", 70)
apresentar("Pedro", 66, "Anápolis")
apresentar("Lúcia", 73)
```

Exercício-2

Um ambulatório precisa de uma função para calcular a dose total (em mg) de um medicamento: $dose_total = peso_kg * mg_por_kg$. O padrão do protocolo local é 0.75 mg/kg.

Tarefa.

- Implemente `def dose(peso_kg, mg_por_kg=0.75)`: que **retorne** a dose total (float).
- Implemente `def prescricao(nome, peso_kg, mg_por_kg=0.75)`: que **imprima**:

Paciente: <nome> – Peso: <peso_kg> kg – Dose: <dose_total> mg

- Faça duas prescrições (duas chamadas separadas), sem listas.

Exemplo de uso esperado.

```
prescricao("Carlos", 80)
prescricao("Rita", 60, 0.8)
```

Exercício-3

Um projeto social cobra uma tarifa base por deslocamento. Essa tarifa pode ser atualizada por um coordenador.

Tarefa.

- Defina uma variável global `tarifa_base = 4.50`.
- Implemente:
 - `def calcular_passagem(distancia_km)`: retorna `distancia_km * tarifa_base`.
 - `def alterar_tarifa(nova_tarifa)`: usa global `tarifa_base` para atualizar a tarifa.
- Demonstre:
 1. Calcule o preço de 10 km (tarifa inicial).
 2. Altere a tarifa para 5.00.
 3. Recalcule o preço de 10 km e mostre a diferença.

Exemplo de sequência.

```
# tarifa_base definida fora das funções
print(calcular_passagem(10))
alterar_tarifa(5.00)
print(calcular_passagem(10))
```

Exercício-4

Você precisa de uma função que indique rapidamente se uma medida é OK ou ALTA para triagem:

- **ALTA** se sistólica ≥ 140 ou diastólica ≥ 90 , senão **OK**.

Tarefa.

- Implemente def `classifica_pressao(sistolica, diastolica)`: que retorna "ALTA" ou "OK".
- Implemente def `relatorio_pressao(nome, sistolica, diastolica, cidade="Goiânia")` que imprima:

```
<nome> - <cidade> - Medida: <sistolica>/<diastolica> - Classi
```

- Faça duas chamadas com valores diferentes (sem listas).

Exemplo de uso esperado.

```
relatorio_pressao("Ana", 135, 85)
relatorio_pressao("Miguel", 142, 91, "Trindade")
```

Exercício-5

Na recepção, você precisa montar rapidamente o texto do crachá no formato:

```
<NOME COMPLETO> - Idade: <idade>
```

Sem guardar dados em coleções, apenas gerando o texto na hora.

Tarefa.

- Crie `formata = lambda nome, sobrenome, idade: nome + " " + sobrenome + " - Idade: " + str(idade)`
- Imprima o crachá de duas pessoas (duas chamadas).

Exemplo de uso esperado.

```
print(formata("Lucília", "Ribeiro", 68))
print(formata("Paulo", "Santos", 72))
```
