Aula 10
Funções

Profa. Simone de Abreu

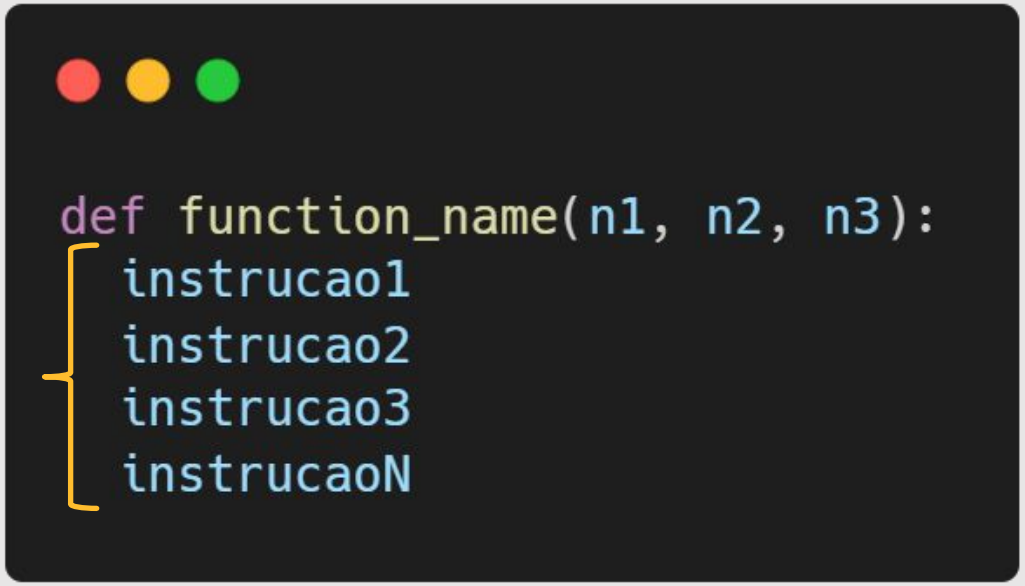
Funções

- Servem para **organização e reutilização** de código dentro de um sistema
- Uma **função** é um bloco de código **nomeado** que tem a responsabilidade de executar uma determinada tarefa
- Quais usamos algumas funções até aqui?

Funções - elementos

Header da função

corpo da função



- **def** – define a função
- **function_name** – nome da função
- () – lista de parâmetros opcional
- : – indica o início do bloco da função

Exemplo 1

```
def hello():
    print("Hello everyone!!")

hello() # invoca a função hello()
hello() # invoca hello() novamente
```

Saída:

```
Hello everyone!!
Hello everyone!!
```

Como posso personalizar a função hello() ?

Funções – Parâmetros e argumentos

- **Parâmetros**
  - Uma função pode ou não receber uma lista dados
- **Argumentos posicionais**
  - São os valores passados para dentro de uma função no momento que ela é invocada e na posição exata dos parâmetros

Exemplo 2

```
def hello(name, message):
    print(f"{message}, {name}")

hello("Paul", "Hello") # invoca a função hello passando 2 argumentos
```

Saída:

```
Hello, Paul
```

Funções – com retorno

- Uma função pode ou não receber parâmetros e também pode ou não retornar dado ao chamador
- Para retornar dado, por exemplo, um resultado, o programa deve utilizar a palavra **return**

Exemplo 3

```python
def somar(num1, num2):
    soma = num1 + num2
    return soma


resultado = somar(10, 40)
print(f"A soma é {resultado}")
```

Saída:

```
A soma é 50
```

Funções – com parâmetros padrão (default)

- É possível definir um valor **default** para um parâmetro de função, assim, caso o argumento não seja passado, assume o valor **default**

Exemplo 4

```
def hello(name, message="Hi"):
    print(f"{message}, {name}")

hello("Kelly", "Hello")
hello("Paul") # invoca passando 1 argumento
```

Saída:

```
Hello, Kelly
Hi, Paul
```

Funções – com argumentos nomeados

- É possível nomear cada argumento ao invocar uma função
- Com isso, não é necessário manter a ordem posicional

Exemplo 5

```
def hello(name, message="Hi"):
    print(f"{message}, {name}")

hello("Kelly", "Hello") # argumento posicional (ordem importa)
hello("Paul") # invoca passando 1 argumento

hello(message="Bye", name="Kelly") # argumentos nomeados
```

Saída:

```
Hello, Kelly
Hi, Paul
Bye, Kelly
```

KEEP
CALM
AND
VAMOS
PRATICAR

Pense, Pareie, Compartilhe

Em DUPLA

Exercícios

1. Construa a função verificar() que retorna True se o número digitado for positivo, False se o número digitado for negativo ou Neutro se for zero.

```
def verificar(num):
```

Exercícios

2. Criar a função converter() que recebe um valor representando os segundos. A função deve converter o valor para horas, minutos e segundos e imprimir.

```
def converter(segundos):
```

Exercícios

3. Criar uma função para receber 3 notas, calcular e retornar a média aritmética.

- Fazer a validação de cada uma das 3 notas. Cada nota deve estar entre 0 e 10. Criar uma função para realizar essa validação

Exercícios

4 - Criar um programa para calcular o IMC (índice de massa corporal). O programa deve implementar duas funções: 1 para realizar o processamento do IMC e outra para verificar a classificação.

Exercícios

5 - Faça uma função que retorne o reverso de um número inteiro informado.

Por exemplo: 271 -> 172.

Exercícios

6 - Faça um programa que solicite a data de nascimento (dd/mm/aaaa) do usuário e imprima a data com o nome do mês por extenso. O programa deve chamar uma função que retorna o mês convertido.

- Exemplo: – Entrada - Data de Nascimento: 29/10/1973 – Saída - Você nasceu em 29 de Outubro de 1973.

Exercícios

7 - Faça uma função chamada calcular_imposto. A função possui dois parâmetros :

a) taxa_imposto, que é a porcentagem de imposto sobre vendas

b) b) custo, que é o custo de um item antes do imposto.

A função retorna o valor de custo alterado para incluir o imposto sobre vendas.