

CURSO

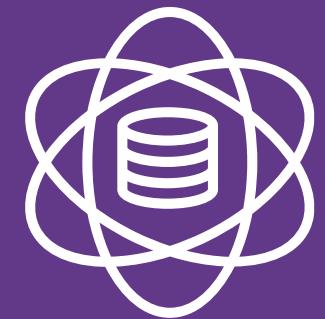
# **BIG DATA SCIENCE**

## **\_ANALISTA DE DADOS**

**Senac**

Todas as formas  
de aprender

**tecnologia**



# Vamos lembrar?

**Lista**  
**Tuplas**  
**Set**  
**Dicionário**



Dessa vez é muita lembrança.  
Será que vou lembrar de tudo?

Faça um algoritmos em python que mostre os números inteiros de 0 a 100 com 4 versões: lista, tupla, set e dicionário:

```
# Usando LISTA
numeros_lista = [i for i in range(101)]
print("Lista de números de 0 a 100:")
print(numeros_lista)

# Usando TUPLA
numeros_tupla = tuple(i for i in range(101))
print("Tupla de números de 0 a 100:")
print(numeros_tupla)

# Usando SET
numeros_set = {i for i in range(101)}
print("Set de números de 0 a 100:")
print(numeros_set)

# Usando DICT
numeros_dict = {i: i for i in range(101)}
print("Dict de números de 0 a 100:")
print(numeros_dict)
```



# Funções

Uma função em um algoritmo é um bloco de código nomeado que realiza uma tarefa específica e retorna um valor após sua execução. Elas servem para modularizar o programa, reutilizar código e torná-lo mais legível, evitando repetição. A principal diferença entre uma função e um procedimento é que a função obrigatoriamente usa o comando `return` para devolver um resultado, enquanto um procedimento não retorna valor, mas pode modificar variáveis.

Hum... ficou  
bem melhor!



## **Características principais de uma função:**

**Retorno de valor:** A função retorna um valor para quem a chamou, que pode ser usado em expressões ou atribuído a uma variável.

**Obrigação de retornar:** O uso do comando `retorne` é obrigatório para indicar qual valor a função devolverá ao programa principal.

**Parâmetros:** Uma função pode aceitar parâmetros, que são valores de entrada para processamento interno.

**Reutilização de código:** Permite que um mesmo trecho de código seja executado várias vezes sem a necessidade de reescrevê-lo.

**Modularização:** Divide o programa em partes lógicas menores e mais fáceis de entender.

**Escopo local:** Variáveis declaradas dentro da função têm escopo local, o que significa que só existem e são acessíveis dentro do bloco da função.



# Funções em Python

Uma função em Python é um bloco de código reutilizável que executa uma tarefa específica. Elas são definidas usando a palavra-chave `def`, seguidas por um nome, parênteses e dois pontos. Funções podem receber dados (parâmetros), processá-los e retornar um resultado usando a palavra-chave `return`. Elas ajudam a organizar o código, torná-lo mais legível e a evitar repetições.



Salve  
galera!

Faça um algoritmo que leia a entrada de dois números inteiros e faça a soma entre eles:

```
def somar(a, b):
    return a + b

#Aplicação principal
num1 = int(input("Digite o primeiro número: "))
num2 = int(input("Digite o segundo número: "))

soma = somar(num1, num2)

print(f"Soma: {soma}")
```



Você  
consegue!

Com base no exemplo, implemente as funções de subtração, multiplicação, divisão e a média entre eles:



Eu falei que você  
pode e pode  
mesmo! Veja como  
é descomplicado

```
def somar(a, b):  
    return a + b  
  
def subtrair(a, b):  
    return a - b  
  
def multiplicar(a, b):  
    return a * b  
  
def dividir(a, b):  
    if b != 0:  
        return a / b  
    else:  
        return "Erro: divisão por zero!"  
  
def media(a, b):  
    return (a + b) / 2
```

```
# Programa principal  
print("==== Calculadora de Operações com Dois Números ===")  
  
# Entrada de dados  
num1 = int(input("Digite o primeiro número: "))  
num2 = int(input("Digite o segundo número: "))  
  
# Cálculos  
soma = somar(num1, num2)  
subtracao = subtrair(num1, num2)  
multiplicacao = multiplicar(num1, num2)  
divisao = dividir(num1, num2)  
media_valor = media(num1, num2)  
  
# Saída de resultados  
print("--- Resultados ---")  
print(f"Soma: {soma}")  
print(f"Subtração: {subtracao}")  
print(f"Multiplicação: {multiplicacao}")  
print(f"Divisão: {divisao}")  
print(f"Média: {media_valor}")
```

# Desafio:

Criar um programa que ajude um pescador a controlar sua produtividade. Toda vez que ele traz um peso de peixes maior que o estabelecido pelo regulamento de pesca do estado de Santa Catarina (100 quilos), deve pagar uma multa de R\$ 4,00 por quilo excedente. O pescador precisa que você faça um programa que leia o peso de peixes pescados no dia e verifique se há excesso. Se houver peso excedente, mostrar o valor que ele pagará de multa; caso contrário, mostrar uma mensagem dizendo que ele não deve pagar nada (resolver a questão utilizando uma função para calcular a multa caso seja necessário).



Estou  
acreditando  
que vão  
conseguir!



```
def calcular_multa(peso):
    limite = 100 # limite estabelecido pelo regulamento
    valor_multa = 4.00 # valor da multa por quilo excedente

    if peso > limite:
        excesso = peso - limite
        multa = excesso * valor_multa
        return excesso, multa
    else:
        return 0, 0 # sem excesso nem multa

# Programa principal
print("== Controle de Produtividade do Pescador ==")

# Entrada de dados
peso_peixes = float(input("Digite o peso total de peixes pescados (em kg): "))

# Cálculo
excesso, multa = calcular_multa(peso_peixes)

# Saída de resultados
print("--- Resultado ---")
if excesso > 0:
    print(f"Peso excedente: {excesso:.2f} kg")
    print(f"Valor da multa: R$ {multa:.2f}")
else:
    print("Dentro do limite permitido. Nenhuma multa a pagar.")
```

# Mais um pouquinho:

Criar um programa que leia a altura e o peso de N pessoas e mostre seu IMC com a respectiva classificação. O cálculo do IMC deverá ser realizado através de uma função que receberá os valores de entrada necessários para o cálculo.

$$\text{Fórmula} \rightarrow \text{IMC} = \text{PESO} / (\text{ALTURA} * \text{ALTURA})$$

IMC (kg/m <sup>2</sup> )	Classificação
Menor que 16,9	Muito abaixo do peso
17 a 18,4	Abaixo do peso
18,5 a 24,9	Peso normal
25 a 29,9	Acima do peso
30 a 34,9	Obesidade grau I
35 a 40	Obesidade grau II
Maior que 40	Obesidade grau III



```
def calcular_imc(peso, altura):
    imc = peso / (altura * altura)
    return imc

# Função para classificar o IMC
def classificar_imc(imc):
    if imc < 16.9:
        return "Muito abaixo do peso"
    elif 17 <= imc <= 18.4:
        return "Abaixo do peso"
    elif 18.5 <= imc <= 24.9:
        return "Peso normal"
    elif 25 <= imc <= 29.9:
        return "Acima do peso"
    elif 30 <= imc <= 34.9:
        return "Obesidade grau I"
    elif 35 <= imc <= 40:
        return "Obesidade grau II"
    else:
        return "Obesidade grau III"
```

# Solução:

```
# Programa principal
print("== Cálculo de IMC ==")

n = int(input("Digite o número de pessoas: "))

for i in range(1, n + 1):
    print(f"--- Pessoa {i} ---")
    peso = float(input("Digite o peso (kg): ").replace(",","."))
    altura = float(input("Digite a altura (m): ").replace(",","."))

    imc = calcular_imc(peso, altura)
    classificacao = classificar_imc(imc)

    print(f"IMC: {imc:.2f}")
    print(f"Classificação: {classificacao}")

print("== Fim do programa ==")
```

Criar um algoritmo em Python que tenha a entrada de nome e salário bruto.

Com função, resolve as situações abaixo:

### **INSS:**

Se o salário bruto  $\geq 1800$  desconte 11%

Se o salário bruto  $< 1800$  desconte 9%

### **Vale transporte**

Se o salário bruto  $\geq 1500$  desconte 6%

Se o salário bruto  $< 1500$  desconte 5%

### **Bônus:**

Se o salário bruto  $\geq 1240$  some com mais 700 reais

Se o salário bruto  $< 1240$  some com mais 500 reais

### **Cargo:**

Se o salário bruto  $\geq 3000$  = “Acionista”

Se o salário bruto  $\geq 2000$  = “Gerente”

Se o salário bruto  $< 2000$  = “Vendedor”

Salário Líquido:

Salário Líquido = salário bruto – (inss + vale) + bônus



Deixa eu  
esconder o  
rosto !

Mostre todas as informações no final:

# Solução:

```
def calcularInss(inss):
    if salarioBruto >= 1800:
        inss = salarioBruto * 0.11
    else:
        inss = salarioBruto * 0.09

    return inss

# Função para calcular o desconto do Vale-Transporte

def calcularVale(vale):
    if salarioBruto >= 1500:
        vale = salarioBruto * 0.06
    else:
        vale = salarioBruto * 0.05

    return vale
```

```
# Função para calcular o salário líquido

def salarioLiquido(salarioLiquido):
    salarioLiquido = salarioBruto - (calcularInss(inss) + calcularVale(vale)) + calcularBonus(bonus)
    return salarioLiquido
```

```
def calcularBonus(bonus):
    if salarioBruto >= 1240:
        bonus = 700
    else:
        bonus = 500

    return bonus

# Função para classificar o Cargo

def classificarCargo(cargo):
    if salarioBruto >= 3000:
        cargo = "Acionista"
    elif salarioBruto >= 2000:
        cargo = "Gerente"
    else:
        cargo = "Vendedor"

    return cargo
```



Acreditei em você e  
você conseguiu!

```
# Programa principal
print("==== Cálculo de Salário ===")

nome = input("Digite o nome do funcionário: ")
salarioBruto = float(input("Digite o salário bruto: R$ "))

# Cálculos
inss = calcularInss(salarioBruto)
vale = calcularVale(salarioBruto)
bonus = calcularBonus(salarioBruto)
cargo = classificarCargo(salarioBruto)
salarioLiquido = salarioLiquido(salarioBruto)

# Exibição dos resultados
print("--- Folha de Pagamento ---")
print(f"Nome: {nome}")
print(f"Cargo: {cargo}")
print(f"Salário Bruto: R$ {salarioBruto:.2f}")
print(f"Desconto INSS: R$ {inss:.2f}")
print(f"Desconto Vale-Transporte: R$ {vale:.2f}")
print(f"Bônus: R$ {bonus:.2f}")
print(f"Salário Líquido: R$ {salarioLiquido:.2f}")

print("==== Fim do Programa ===")
```