**Curso de Python 3 do básico ao avançado - com projetos reais**

Instalar versão mais atualizada e completa do python no site python.org

Para saber a versão do python: python --version ou py –version ou python -V.

\*Utilizar powershell.

Verificar o PowerShell está restrito:

- Abrir o powershell como administrador;

- Digitar Get-ExecutionPolicy

Caso o PowerShell esteja restrito:

- Executar Set-ExecutionPolicy AllSigned -Force

Comentário no código (Interpretador do python não le): # Comentário

DocString – comentário multilinha (Interpretador do python le):

“””

DocString

“””

‘’’

DocString

‘’’

=============================

#sep - separador

# \r \n - quebra de linha -> CRLF

# \n - quebra de linha -> LF

# end - final do print

print(12, 34, sep=' ', end='\r\n')

print(56, 78, sep='-', end='\n')

print(90, 12, sep='=', end='!!')

print(34, 56, sep='+', end='##\n')

print(78, 90, sep='/', end='\n||')

=============================



=============================

"""

Python = Linguagem de programação

Tipo de tipagem = Dinâmica / Forte

str -> string -> Texto

int -> inteiro -> Número inteiro

float -> número decimal

bool -> booleano/lógico -> Verdadeiro/Falso

Strings são textos que estão dentro de aspas

"""

print(1234)

#Aspas simples

print('1234')

#Aspas duplas

print("1234")

#Escape - Pouco usado

print('12\'34\'') # Por exemplo: Para mostrar aspas

#r - Pouco usado

print(r'12\'34\'') # Para mostrar o escape no compilamento

#Forma melhor para mostrar aspas, por exemplo

print('12"34"')

#Obs: Tudo que estiver dentro de aspas é uma string!

=============================

#Tipo int e float

#int - > Números inteiros

#O tipo int representa qualquer número

#Positivo ou negativo. int sem sinal é considerado

#Positivo.

print(11) #int

print(-11) #int

print(0) #int

#float - > Números com ponto flutuante

#O tipo float representa qualquer número

#Positivo ou negativo com ponto flutuante.

#float sem sinal é considerado positivo.

print(11.0) #float

print(-11.0) #float

print(0.0) #float

#A função type() mostra o tipo que o Python inferiu o valor.

print(type(11)) #int

print(type(11.00)) #float

print(type("Teste")) #str

print(type(True)) #bool

=============================

#Tipo de dado bool (boolean)

#Ao questionar algo em um programa, só existem duas respostas possíveis:

#Sim(True) ou Não(False).

#Existem vários operadores para "questionar"

#Dentre eles o ==, que é um operador lógico que questiona se um valor é igual a outro.

print(10 == 10) #True

print(10 == 5) #False

=============================

#Conversão de tipos, coerção

#Type convertion, typecastasting, coercion

#É o ato de converter um tipo em outro

#Tipos imutáveis e primivos: str, int, float, bool

print(1+1) #Python entendeu que 1 e 1 são números inteiros e fez a soma

print('a'+'b')#Python entendeu que 'a' e 'b' são strings e fez a concatenação

print('1', type('1'))#Python entendeu que '1' é uma string e imprimiu o tipo dela

print(int('1') + 1) #Python converteu a string '1' em um número inteiro e fez a soma

print(float('1') + 1) #Python converteu o número inteiro 1 em um número float e fez a soma

print(bool('1')) #Python converteu a string '1' em um booleano e imprimiu True

print(bool('')) #Python converteu a string vazia em um booleano e imprimiu False

print(str(1 + 1)) #Python converteu o número inteiro 2 em uma string e imprimiu '2'

print(str(1)+'A') #Python converteu o número inteiro 1 em uma string e imprimiu '1A'

Aluno = 'João'

print(str(1)+'-'+Aluno) #Python converteu o número inteiro 1 em uma string e imprimiu '1-João'

import os

os.system('pause')

=============================

#Variáveis são usadas para salvar algho na memória do computador

#PEP8: incie variáveis com letras minúsculas, pode usar números e underline \_.

#O sinal de = é o operador de atribuição. Ele é usado para atribuir um valor a um nome(variável).

#Uso: nome\_variavel=experssão

import os

nome\_completo = 'João Augusto'

soma\_dois = 2 + 2

int\_1 = int('1')

print('\n')

print(nome\_completo,'-', soma\_dois)#Imprime o valor da variável nome\_completo e soma\_dois

print('\n')

print(int\_1,type(int\_1))#Imprime o valor da variável int\_1 e o tipo dela

print('\n')

os.system('pause')

os.system('cls')

#Exemplo

nome = 'João'

idade = 26

maior\_de\_idade = idade >= 18

print('Nome:', nome)

print('Idade:', idade)

if maior\_de\_idade == True:

    print('\n')

    print(nome, 'tem', idade, 'anos e é maior de idade!')

    print('\n')

else:

    print('\n')

    print(nome, 'tem', idade, 'anos e não é maior de idade!')

    print('\n')

os.system('pause')

os.system('cls')

=============================

import os

color = os.system('color 1f')

def menu():

color

print('\n=======================')

print('Solicitação Informações')

print('=======================\n')

os.system('pause')

os.system('cls')

nome = input('\nDigite o nome: ')

sobrenome = input('\nDigite o sobrenome: ')

idade = int(input('\nDigite a idade: '))

dia\_nasc = int(input('\nDigite o dia de nascimento: '))

mes\_nasc = int(input('\nDigite o mês de nascimento: '))

ano\_nasc = int(input('\nDigite o ano de nascimento: '))

altura = float(input('\nDigite a altura: '))

os.system('cls')

resultado(nome, sobrenome, idade, dia\_nasc, mes\_nasc, ano\_nasc, altura)

def resultado(nome, sobrenome, idade, dia\_nasc, mes\_nasc, ano\_nasc, altura):

print(f'\nNome: {nome}')

print(f'\nSobrenome: {sobrenome}')

print(f'\nIdade: ', idade)

print(f'\nDia de Nascimento: {dia\_nasc}')

print(f'\nMês de Nascimento: {mes\_nasc}')

print(f'\nAno de Nascimento: {ano\_nasc}')

print(f'\nAltura: {altura:.2f}')

if idade >= 18:

print(f'\n{nome} {sobrenome}, nasceu em {dia\_nasc}/{mes\_nasc}/{ano\_nasc}, tendo {idade} anos, com {altura:.2f}m de altura, sendo maior de idade!\n')

os.system('pause')

os.system('cls')

elif idade < 18:

print(f'\n{nome} {sobrenome}, nasceu em {dia\_nasc}/{mes\_nasc}/{ano\_nasc}, tendo {idade} anos, com {altura:.2f}m de altura, sendo menor de idade!\n')

os.system('pause')

os.system('cls')

else:

print('\nErro!')

print('\nVerifique os dados inseridos!\n')

os.system('pause')

os.system('cls')

menu()

menu()

=============================

import os

os.system('color 1f')

# Exemplo de uso de operadores lógicos

# Menu

def menu():

    print('\n====================================')

    print('Exemplo de uso de operadores lógicos')

    print('====================================\n')

    print('1 - Adição')

    print('2 - Subtração')

    print('3 - Multiplicação')

    print('4 - Divisão')

    print('5 - Divisão Inteira')

    print('6 - Exponenciação')

    print('7 - Módulo (Resto da Divisão)')

    print('8 - Sair')

    print('\n====================================\n')

    op = int(input('Escolha uma opção: '))

    os.system('cls')

    if op == 1:

        adicao()

    elif op == 2:

        subtracao()

    elif op == 3:

        multiplicacao()

    elif op == 4:

        divisao()

    elif op == 5:

        divisao\_inteira()

    elif op == 6:

        exponenciacao()

    elif op == 7:

        modulo()

    elif op == 8:

        print('\nFinalizando sistema...\n')

        os.system('pause')

        os.system('cls')

    else:

        print('\nErro!!!')

        print('\nFavor verificar os número da opção digitada.\n')

        os.system('pause')

        os.system('cls')

        menu()

# Adição

def adicao():

    a = int(input('Digite o primeiro número: '))

    b = int(input('Digite o segundo número: '))

    os.system('cls')

    soma  = a + b

    if soma > 10:

        print(f'\nA soma de {a} + {b} = {soma} é maior que 10.\n')

        os.system('pause')

        os.system('cls')

        menu()

    elif soma < 10:

        print(f'\nA soma de {a} + {b} = {soma} é menor que 10.\n')

        os.system('pause')

        os.system('cls')

        menu()

    elif soma == 10:

        print(f'\nA soma de {a} + {b} = {soma} é igual a 10.\n')

        os.system('pause')

        os.system('cls')

        menu()

    else:

        print('\nErro no processamento da soma!!!')

        print('\nFavor verificar os números digitados.\n')

        os.system('pause')

        os.system('cls')

        menu()

# Subtração

def subtracao():

    a = int(input('Digite o primeiro número: '))

    b = int(input('Digite o segundo número: '))

    os.system('cls')

    sub = a - b

    if sub > 10:

        print(f'\nA subtração de {a} - {b} = {sub} é maior que 10.\n')

        os.system('pause')

        os.system('cls')

        menu()

    elif sub < 10:

        print(f'\nA subtração de {a} - {b} = {sub} é menor que 10.\n')

        os.system('pause')

        os.system('cls')

        menu()

    elif sub == 10:

        print(f'\nA subtração de {a} - {b} = {sub} é igual a 10.\n')

        os.system('pause')

        os.system('cls')

        menu()

    else:

        print('\nErro no processamento da subtração!!!')

        print('\nFavor verificar os números digitados.\n')

        os.system('pause')

        os.system('cls')

        menu()

# Multiplicação

def multiplicacao():

    a = int(input('Digite o primeiro número: '))

    b = int(input('Digite o segundo número: '))

    os.system('cls')

    mult = a \* b

    if mult > 10:

        print(f'\nA multiplicação de {a} \* {b} = {mult} é maior que 10.\n')

        os.system('pause')

        os.system('cls')

        menu()

    elif mult < 10:

        print(f'\nA multiplicação de {a} \* {b} = {mult} é menor que 10.\n')

        os.system('pause')

        os.system('cls')

        menu()

    elif mult == 10:

        print(f'\nA multiplicação de {a} \* {b} = {mult} é igual a 10.\n')

        os.system('pause')

        os.system('cls')

        menu()

    else:

        print('\nErro no processamento da multiplicação!!!')

        print('\nFavor verificar os números digitados.\n')

        os.system('pause')

        os.system('cls')

        menu()

# Divisão

def divisao():

    a = int(input('Digite o primeiro número: '))

    b = int(input('Digite o segundo número: '))

    os.system('cls')

    div = a / b

    if div > 10:

        print(f'\nA divisão de {a} / {b} = {div:.2f} é maior que 10.\n')

        os.system('pause')

        os.system('cls')

        menu()

    elif div < 10:

        print(f'\nA divisão de {a} / {b} = {div:.2f} é menor que 10.\n')

        os.system('pause')

        os.system('cls')

        menu()

    elif div == 10:

        print(f'\nA divisão de {a} / {b} = {div:.2f} é igual a 10.\n')

        os.system('pause')

        os.system('cls')

        menu()

    else:

        print('\nErro no processamento da divisão!!!')

        print('\nFavor verificar os números digitados.\n')

        os.system('pause')

        os.system('cls')

        menu()

# Divisão Inteira

def divisao\_inteira():

    a = int(input('Digite o primeiro número: '))

    b = int(input('Digite o segundo número: '))

    os.system('cls')

    div\_int = a // b

    if div\_int > 10:

        print(f'\nA divisão inteira de {a} // {b} = {div\_int:.2f} é maior que 10.\n')

        os.system('pause')

        os.system('cls')

        menu()

    elif div\_int < 10:

        print(f'\nA divisão inteira de {a} // {b} = {div\_int:.2f} é menor que 10.\n')

        os.system('pause')

        os.system('cls')

        menu()

    elif div\_int == 10:

        print(f'\nA divisão inteira de {a} // {b} = {div\_int:.2f} é igual a 10.\n')

        os.system('pause')

        os.system('cls')

        menu()

    else:

        print('\nErro no processamento da divisão inteira!!!')

        print('\nFavor verificar os números digitados.\n')

        os.system('pause')

        os.system('cls')

        menu()

# Exponenciação

def exponenciacao():

    a = int(input('Digite o primeiro número: '))

    b = int(input('Digite o segundo número: '))

    os.system('cls')

    exp = a \*\* b

    if exp > 10:

        print(f'\nA exponenciação de {a} \*\* {b} = {exp} é maior que 10.\n')

        os.system('pause')

        os.system('cls')

        menu()

    elif exp < 10:

        print(f'\nA exponenciação de {a} \*\* {b} = {exp} é menor que 10.\n')

        os.system('pause')

        os.system('cls')

        menu()

    elif exp == 10:

        print(f'\nA exponenciação de {a} \*\* {b} = {exp} é igual a 10.\n')

        os.system('pause')

        os.system('cls')

        menu()

    else:

        print('\nErro no processamento da exponenciação!!!')

        print('\nFavor verificar os números digitados.\n')

        os.system('pause')

        os.system('cls')

        menu()

# Módulo (Resto da Divisão)

def modulo():

    a = int(input('Digite o primeiro número: '))

    b = int(input('Digite o segundo número: '))

    os.system('cls')

    mod = a % b

    if mod > 10:

        print(f'\nO restro da divisão (Módulo) de {a} % {b} = {mod} é maior que 10.\n')

        os.system('pause')

        os.system('cls')

        menu()

    elif mod < 10:

        print(f'\nO restro da divisão (Módulo) de {a} % {b} = {mod} é menor que 10.\n')

        os.system('pause')

        os.system('cls')

        menu()

    elif mod == 10:

        print(f'\nO restro da divisão (Módulo) {a} % {b} = {mod} é igual a 10.\n')

        os.system('pause')

        os.system('cls')

        menu()

    else:

        print('\nErro no processamento do restro da divisão (Módulo)!!!')

        print('\nFavor verificar os números digitados.\n')

        os.system('pause')

        os.system('cls')

        menu()

print('\n+++++++++')

print('Exemplos:')

print('+++++++++\n')

ad = 10 + 10

print(f'Adição de 10 + 10: {ad}\n')

subt = 10 - 5

print(f'Subtração de 10 - 5: {subt}\n')

multi = 10 \* 10

print(f'Multiplicação de 10 \* 10: {multi}\n')

divi = 10 / 2.2 # Float - Sempre imprimi com ponto flutuante

print(f'Divisão de 10 / 2.2: {divi:.2f}\n')

divi\_inteira = 10 // 2.2 # Todo número que vem depois do ponto, não virá

print(f'Divisão inteira de 10 // 2.2: {divi\_inteira:.2f}\n')

expo = 2 \*\* 10

print(f'Exponenciação de 2 \*\* 10: {expo}\n')

modu = 55 % 5 # Resto da divisão

print(f'Módulo de 55 % 5: {modu}\n')

os.system('pause')

os.system('cls')

menu()

=============================

import os

concatenacao = 'a' + 'b' + 'c'

concatenacao\_1 = 'a' + 'b' + 'c' + str(1)

print('\n')

print(concatenacao)

print('\n')

print(concatenacao\_1)

print('\n')

a\_dez\_vezes = 'a' \* 10

tres\_vezes\_luiz = 3 \* 'Luiz'

print(a\_dez\_vezes)

print('\n')

print(tres\_vezes\_luiz)

print('\n')

os.system('pause')

os.system('cls')

=============================

import os

# 1. (n + n)

# 2. \*\*

# 3. \* / // %

# 4. + -

conta\_1 = 1 + 1 \*\* 5 + 5 # 7

print('\n')

print(f'Conta\_1: {conta\_1}')

print('\n')

conta\_2 = (1 + 1) \*\* (5 + 5) #1024

print(f'Conta\_2: {conta\_2}')

print('\n')

os.system('pause')

os.system('cls')

=============================

import os

os.system('color 1f')

nome = 'João Augusto'

altura = 1.80

peso = 97.478

imc = ... # Ellipsis - placeholder para adicionar posteriormente o código.

imc = peso / (altura \* altura)

print(f'\n{nome}, tem {altura:.2f} de altura, pesa {peso:.2f}kg e seu IMC é {imc:.2f}\n')

os.system('pause')

os.system('cls')

=============================

import os

os.system('color 1f')

a = 'A'

b = 'B'

c = 1.1

string\_1 = '| a = {} | b = {} | c = {:.2f} |'

string\_2 = '| a = {2:.2f} | b = {0} | c = {1} |' # Índices

string\_3 = '| a = {nome1} | b = {nome2} | c = {nome3:.2f} |'

formato\_1 = string\_1.format(a, b, c)

formato\_2 = string\_2.format(a, b, c)

formato\_3 = string\_3.format(nome1 = a, nome2 = b, nome3 = c) # Parametro nomeado

print(f'\n{formato\_1}')

print(f'{formato\_2}')

print(f'{formato\_3}\n')

os.system('pause')

os.system('cls')

=============================

import os

os.system('color 1f')

nome\_1 = input('Qual o seu nome: ')

print(f'Seu nome é {nome\_1}')

os.system('pause')

os.system('cls')

nome\_2 = input('Qual o seu nome: ')

print(f'Seu nome é {nome\_2 = }')

os.system('pause')

os.system('cls')

n1 = input('Digite um número: ')

n2 = input('Digite outro número: ')

int\_n1 = int(n1)

int\_n2 = int|(n2)

print(f'A soma dos números é: {n1 + n2}')

os.system('pause')

os.system('cls')

=============================

# if | elif | else

# se | Se não se| se não

import os

os.system('color 1f')

ent = input('Você quer "entrar" ou "sair"? ')

if ent == 'entrar':

print('Você entrou no sistema!')

os.system('pause')

elif ent == 'sair':

print('Você saiu do sistema!')

os.system('pause')

else:

print('Você não digitou nem entrar e nem sair!')

os.system('pause')

=============================

# ... ou pass: Não executa nada, funcionando para situações onde não se quer fazer nada no momento naquela parte do código

import os

os.system('color 1f')

condicao1 = True

condicao2 = False

condicao3 = False

condicao4 = False

if condicao1:

    print('A condição 1 é verdadeira')

elif condicao2:

    print('A condição 2 é verdadeira')

elif condicao3:

    print('A condição 3 é verdadeira')

elif condicao4:

    print('A condição 4 é verdadeira')

else:

    print('Nenhuma condição é verdadeira')

print('Fora do if')

os.system('pause')

=============================

"""

Atividade:

Apresentar o valor maior.

"""

import os

os.system('color 1f')

def menu(valor1, valor2):

    print('\n============')

    print('VALOR MAIOR')

    print('============\n')

    valor1 = input('Digite o primeiro valor: ')

    valor2 = input('Digite o segundo valor: ')

    os.system('cls')

    valores(valor1, valor2)

def valores(valor1, valor2):

    if valor1 > valor2:

        print(f'\nO primeiro valor ({valor1}) é maior que o segundo valor ({valor2}).\n')

        os.system('pause')

        os.system('cls')

        menu(valor1 = 0, valor2 = 0)

    elif valor1 < valor2:

        print(f'\nO segundo valor ({valor2}) é maior que o primeiro valor ({valor1}).\n')

        os.system('pause')

        os.system('cls')

        menu(valor1 = 0, valor2 = 0)

    elif valor1 == valor2:

        print(f'\nOs valores 1 ({valor1}) e 2 ({valor2}) são iguais.\n')

        os.system('pause')

        os.system('cls')

        menu(valor1 = 0, valor2 = 0)

    else:

        print('\nValores inválidos.\n')

        os.system('pause')

        os.system('cls')

        menu(valor1 = 0, valor2 = 0)

menu(valor1 = 0, valor2 = 0)

=============================

'''

Operadores lógicos

\* and(e) / or(ou) / not(não)

\* and - Todas as condições precisam ser verdadeiras - Se qualquer valor for considerado falso, a expressão inteira será avalisada naquele valor.

\* or - Qualquer condição verdadeira avalia toda a expressão como verdadeira - Se qualquer valor for considerado verdadeiro, a expressão inteira será avalisada naquele valor.

\* São considerados falsy: 0, 0.0. '', False

\* Também existe o tipo None que é usado para representar um não valor.

\* not - Inverte o valor lógico da expressão - Se a expressão for verdadeira, o not a tornará falsa e vice-versa. - not False = True | not True = False

'''

# Exemplo

import os, getpass

# Mudança de cor do terminal.

os.system('color 1f')

# Menu do app.

def menu():

    print('\n==========================')

    print('EXEMPLO DE ACESSO A SISTEMA')

    print('==========================\n')

    print('[E] - Entrar\n')

    print('[S] - Sair\n')

    # Opção de escolha para entrar no sistema ou sair.

    op = input('Escolha uma opção: ')

    os.system('cls')

    # Caso o usuário escolha entrar no sistema, chama a função acesso().

    if op == 'E' or op == 'e':

        acesso()

    # Caso o usuário queira sair do sistema, o programa é encerrado.

    elif op == 'S' or op == 's':

        print('\nSaindo...\n')

        os.system('pause')

        os.system('cls')

    # Caso o usuário não digite nada, o mesmo será notificado pelo sistema e o menu é chamado novamente.

    elif not op:

        print('\nOpção não digitada!\n')

        print('Favor preencher campo solicitado!\n')

        os.system('pause')

        os.system('cls')

        menu()

    # Caso o usuário digite uma opção inválida, o mesmo será notificado pelo sistema e o menu é chamado novamente.

    else:

        print('\nOpção inválida!\n')

        print('Favor verificar dados repassados!\n')

        os.system('pause')

        os.system('cls')

        menu()

# Função de acesso ao sistema.

def acesso():

    print('\n======')

    print('ACESSO')

    print('======\n')

    print('[A] - Acessar o sistema\n')

    print('[R] - Retornar ao Menu\n')

    # Opção de escolha para acesso no sistema ou retornar ao menu.

    opcao = input('Escolha uma opção: ')

    os.system('cls')

    # Caso o usuario escolha acessar o sistema, solicita usuário e senha.

    if opcao == 'A' or opcao == 'a':

        # Aplicativo solicita usuário e senha para acesso.

        usuario = input('Usuário: ')

        senha = getpass.getpass('Senha: ')

        val\_acesso =  usuario == 'Admin' and senha == 'Admin@123'

        os.system('cls')

        # Caso o usuário e senha estejam corretos, o acesso é permitido.

        if val\_acesso == True:

            print('\nAcesso Permitido!\n')

            os.system('pause')

            os.system('cls')

            acesso()

        # Caso o usuário e senha estejam incorretos ou não digitados, o acesso é negado.

        elif val\_acesso != True:

            print('\nAcesso Negado!\n')

            print('Usuário ou Senha incorreto ou não digitado!\n')

            os.system('pause')

            os.system('cls')

            acesso()

        # Caso o usuário digite uma opção inválida, o mesmo será notificado pelo sistema.

        else:

            print('\nAcesso Negado!\n')

            print('Favor verificar dados repassados!\n')

            os.system('pause')

            os.system('cls')

            acesso()

    # Caso o usuário escolha retornar ao menu, o mesmo é chamado novamente.

    elif opcao == 'R' or opcao == 'r':

        menu()

    # Caso o usuário não digite nada, o mesmo será notificado pelo sistema.

    elif not opcao:

        print('\nOpção não digitada!\n')

        print('Favor preencher campo solicitado!\n')

        os.system('pause')

        os.system('cls')

        acesso()

    # Caso o usuário digite uma opção inválida, o mesmo será notificado pelo sistema.

    else:

        print('\nOpção inválida!\n')

        os.system('pause')

        os.system('cls')

        acesso()

# Chamada da função menu() para iniciar o sistema.

menu()

=============================

'''

Operadores in e not in

\* Strings são iteráveis (Pode navegar em cada item. Ex: letra por letra)

\* 0 1 2 3 4 5

\* P y t h o n

\* -6-5-4-3-2-1

'''

# Exemplo de iteráveis

import os

os.system('color 1f')

nome = 'Python'

print('\n')

print(nome[0]) # P

print(nome[1]) # y

print(nome[2]) # t

print(nome[3]) # h

print(nome[4]) # o

print(nome[5]) # n

print('\n')

print(nome[-6]) # P

print(nome[-5]) # y

print(nome[-4]) # t

print(nome[-3]) # h

print(nome[-2]) # o

print(nome[-1]) # n

print('\n')

os.system('pause')

os.system('cls')

# Exemplo de in e not in

palavra = input('Digite a palavra que formou: ')

os.system('cls')

if palavra in nome:

    print(f'\n"{palavra}" é a palavra formada\n')

    os.system('pause')

    os.system('cls')

elif palavra not in nome:

    print(f'\n"{palavra}" não é a palavra formada\n')

    os.system('pause')

    os.system('cls')

else:

    print('\nInformação inválida!\n')

    os.system('pause')

    os.system('cls')

=============================

'''

Interpolação básica de strings

s - string

d e i - int

f - float

x e X - hexadecimal (ABCDEF0123456789)

'''

import os

os.system('color 1f')

nome = 'João'

preco = 1000.95897643

variavel = '%s, o preço é R$ %.2f' % (nome, preco)

print('\n' + variavel + '\n')

print(10 \* '=' + '\n')

print('O hexadecimal de %d é %06X' % (255, 255) + '\n')

os.system('pause')

os.system('cls')

=============================

'''

Formatação básica de strings

\* s - string

\* d - int

\* f - float

\* .<número de dígitos>f - float com n dígitos após a vírgula

\* x ou X - hexadecimal

\* (Caractere)(><^)(quantidade)

\* > - Esquerda

\* < - Direita

\* ^ - Centro

\* Sinal: - ou +

\* Ex: 0 >- 100,.1f

\* Conversion flags: !r, !s, !a

'''

import os

os.system('color 1f')

variavel = 'Python'

print('\n' + 10 \* '=')

print(f'\n{variavel}.\n')

print(10 \* '=')

print(f'\n{variavel: >10}.\n')

print(10 \* '=')

print(f'\n{variavel: <10}.\n')

print(10 \* '=')

print(f'\n{variavel: ^10}.\n')

print(10 \* '=')

print(f'\nR${1000.4873648: .2f}\n')

print(10 \* '=' + '\n')

print(f'O hexadecimal de 255 é {255:06X}')

print(10 \* '=' + '\n')

os.system('pause')

os.system('cls')

=============================

'''

Fatiamento de strings

\* Olá Mundo

\* 012345678

\*-987654321

\* Fatiamento [i:f:p] [::] - i: indice inicial, f: indice final, p: passo

\* No final do fatiamento, sempre indicar um número de indice a mais.

\* Obs: a função len() retorna a quantidade de caracteres da string.

'''

# Exemplo de iteráveis

import os

os.system('color 1f')

variavel = 'Olá Mundo'

print('\n' + '=' \* 10 + '\n')

print(variavel[0])

print(variavel[1])

print(variavel[2])

print(variavel[3])

print(variavel[4])

print(variavel[5])

print(variavel[6])

print(variavel[7])

print(variavel[8])

print('\n' + '=' \* 10 + '\n')

print(variavel[-9])

print(variavel[-8])

print(variavel[-7])

print(variavel[-6])

print(variavel[-5])

print(variavel[-4])

print(variavel[-3])

print(variavel[-2])

print(variavel[-1])

print('\n' + '=' \* 10 + '\n')

os.system('pause')

os.system('cls')

# Exemplo de fatiamento de strings

print('\n' + '=' \* 10 + '\n')

print(variavel[0:3] + '\n')  # Olá

print(variavel[4:9] + '\n')  # Mundo

print(variavel[0:len(variavel):2] + '\n') # Ol Mnd

print(variavel[::-1] + '\n') # odnuM alO

print(variavel[-1:-10:-1]) # odnuM alO

print('\n' + '=' \* 10 + '\n')

os.system('pause')

os.system('cls')

# Exemplo len()

print('\n' + '=' \* 10 + '\n')

print(len(variavel))  # 9

print(f'\nO tamanho da string "{variavel}" é: {len(variavel)}')

print('\n' + '=' \* 10 + '\n')

os.system('pause')

os.system('cls')

=============================

'''

Exercício

\* Peça ao usuário para digitar seu nome e idade.

\* Se nome e idade forem digitados:

    Exiba:

        Seu nome é <nome>

        Seu nome invertido é <nome invertido>

        Seu nome contém (ou não) espaços

        Seu nome tem <n> letras

        A primeira letra do seu nome é <letra>

        A última letra do seu nome é <letra>

    Se nada for digitado, exiba:

        'Desculpe, você deixou os campos vazios.'

'''

import os

os.system('color 1f')

def menu():

    print('\n' + '=' \* 10 + '\n')

    print('EXERCÍCIO')

    print('\n' + '=' \* 10 + '\n')

    print('Bem-vindo ao programa de análise de nome!\n')

    print('[E] - Entrar')

    print('[S] - Sair')

    print('\n' + '=' \* 10 + '\n')

    opcao = input('Escolha uma opção: ')

    os.system('cls')

    if opcao == 'E' or opcao == 'e':

        info()

    elif opcao == 'S' or opcao == 's':

        print(' \nSaindo do programa...\n')

        os.system('pause')

        os.system('cls')

    else:

        print('\nOpção inválida. Tente novamente.\n')

        os.system('pause')

        os.system('cls')

        menu()

def info():

    print('\n' + '=' \* 10 + '\n')

    print('[I] - Informar dados')

    print('[R] - Retornar ao menu')

    print('\n' + '=' \* 10 + '\n')

    op = input('Escolha uma opção: ')

    os.system('cls')

    if op == 'I' or op == 'i':

        nome = input('\nDigite seu nome: ')

        idade = input('\nDigite sua idade: ')

        os.system('cls')

        if bool(nome) and bool(idade) == True:

            print('\n' + '=' \* 10 + '\n')

            print(f'Seu nome é {nome} e tem {idade} anos!\n')

            print(f'Seu nome invertido é {nome[::-1]}!\n')

            if ' ' in nome:

                print(f'Seu nome contém espaços!\n')

            elif ' ' not in nome:

                print(f'Seu nome não contém espaços!\n')

            else:

                print(f'Erro ao identificar se seu nome contém espaços ou não!\n')

            print(f'Seu nome tem {len(nome)} letras!\n')

            print(f'A primeira letra do seu nome é {nome[0]}!\n')

            print(f'A última letra do seu nome é {nome[-1]}!')

            print('\n' + '=' \* 10 + '\n')

            os.system('pause')

            os.system('cls')

            info()

        elif nome == '' or idade == '':

            print('\nDesculpe, você deixou algum dos campos vazio.')

            print('\nFavor preencher os campos corretamente!\n')

            os.system('pause')

            os.system('cls')

            info()

        else:

            print('\nErro ao identificar os campos!\n')

            os.system('pause')

            os.system('cls')

            info()

    elif op == 'R' or op == 'r':

        print('\nRetornando ao menu...\n')

        os.system('pause')

        os.system('cls')

        menu()

    else:

        print('\nOpção inválida. Tente novamente.\n')

        os.system('pause')

        os.system('cls')

        info()

menu()

=============================

'''

Introdução ao try/except

\* try - tentar executar o código

\* except - se ocorrer um erro, execute o código dentro do except

'''

import os

os.system('color 1f')

def sol\_num():

    print('\n'+ '=' \* 10)

    print('[I] - IF/ELSE')

    print('[T] - TRY/EXCEPT')

    print('[S] - SAIR')

    print('=' \* 10 + '\n')

    op = input('\nEscolha uma opção: ')

    os.system('cls')

    if op == 'I' or op == 'i':

        numero = input('\nDigite um número: ')

        os.system('cls')

        if\_num(numero)

    elif op == 'T' or op == 't':

        numero = input('\nDigite um número: ')

        os.system('cls')

        try\_except(numero)

    elif op == 'S' or op == 's':

        print('\nSaindo do sistema...\n')

        os.system('pause')

        os.system('cls')

    else:

        print('\nOpção inválida\n')

        os.system('pause')

        os.system('cls')

        sol\_num()

# Exemplo de execução do sistema sem utilizar o try/except, onde ao inserir um número futuante, apresenta que não é dígito.

def if\_num(numero):

    if numero.isdigit():

        print(f'\nO dobro de {numero} é {int(numero) \* 2}\n')

        os.system('pause')

        os.system('cls')

        sol\_num()

    elif numero != numero.isdigit():

        print(f'\nO número {numero} não é dígito\n')

        os.system('pause')

        os.system('cls')

        sol\_num()

    else:

        print('\nFalha na execução do programa')

        print('\nVerifique se o número digitado\n')

        os.system('pause')

        os.system('cls')

        sol\_num()

# Exemplo try/except

def try\_except(numero):

    try:

        try:

            print(f'\nO dobro de {numero} é {int(numero) \* 2}\n')

            os.system('pause')

            os.system('cls')

            sol\_num()

        except ValueError:

            print(f'\nO dobro de {float(numero):.2f} é {float(numero) \* 2:.2f}\n')

            os.system('pause')

            os.system('cls')

            sol\_num()

    except ValueError:

        print('\nFalha na execução do programa')

        print('\nVerifique se o número digitado\n')

        os.system('pause')

        os.system('cls')

        sol\_num()

sol\_num()

=============================

'''

CONSTANTE = "Variáveis" que não vão mudar

Muitas condições no mesmo if (ruim)

    <- Contagem de complexidade (ruim)

\* Obs: As variáveis que não mudam, são escritas em maiúsculo

'''

import os

from colorama import init, Fore, Style

init()

def main(vel\_carro\_passou\_radar\_1, range\_carro\_radar\_1, velocidade, local\_carro, RADAR\_1, LOCAL\_1, RADAR\_RANGE):

    RADAR\_1 = 60 # Velocidade máxima do radar 1

    LOCAL\_1 = 100 # Local onde o radar 1 está

    RADAR\_RANGE = 1 # A distância onde o radar pega

    print(Style.BRIGHT, Fore.BLUE + f'\n{"=" \* 10} Sistema Radar {"=" \* 10}\n' + Style.RESET\_ALL)

    print('[A] - Abrir sistema radar')

    print('[S] - Sair')

    print(Style.BRIGHT, Fore.BLUE + f'\n{'=' \* 35}\n'+ Style.RESET\_ALL)

    opcao = input('Escolha uma opção: ')

    os.system('cls')

    if opcao == 'A' or opcao == 'a':

        velocidade = int(input(Style.BRIGHT + f'\nQual a velocidade do automóvel: ')) # Velocidade atual do carro

        local\_carro = int(input(Style.BRIGHT + f'\nQual o local do automóvel no momento: ')) # Local em que o carro está na estrada

        vel\_carro\_passou\_radar\_1 = velocidade > RADAR\_1 # Verifica se o carro está acima da velocidade do radar 1

        range\_carro\_radar\_1 = local\_carro >= (LOCAL\_1 - RADAR\_RANGE) and local\_carro <= (LOCAL\_1 + RADAR\_RANGE) # Verifica se o carro está na área de alcance do radar 1

        os.system('cls')

        verifica\_radar(vel\_carro\_passou\_radar\_1, range\_carro\_radar\_1, velocidade, RADAR\_1)

    elif opcao == 'S' or opcao == 's':

        print(Style.BRIGHT, Fore.BLUE + f'\nSaindo do sistema radar...\n' + Style.RESET\_ALL)

        os.system('pause')

        os.system('cls')

    else:

        print(Fore.RED, Style.BRIGHT + f'\nOpção inválida!\n' + Style.RESET\_ALL)

        os.system('pause')

        os.system('cls')

        main(0, 0, 0, 0, 0, 0, 0)

# Verifica se o carro está na área de alcance do radar e a velocidade é maior que a permitida

def verifica\_radar(vel\_carro\_passou\_radar\_1, range\_carro\_radar\_1, velocidade, RADAR\_1):

    if vel\_carro\_passou\_radar\_1 == True:

        if range\_carro\_radar\_1 == True:

            print(Style.BRIGHT, Fore.RED + f'\nVocê está acima da velocidade permitida!' + Style.RESET\_ALL)

            print(f'\nVelocidade máxima permitida: {RADAR\_1} km/h')

            print('\nVelocidade atual do veículo:' + Style.BRIGHT, Fore.RED + f'{velocidade} km/h\n' + Style.RESET\_ALL)

            os.system('pause')

            os.system('cls')

            main(0, 0, 0, 0, 0, 0, 0)

        elif range\_carro\_radar\_1 == False:

            print(Style.BRIGHT, Fore.YELLOW + f'\nVocê não se encontra na área de alcance do radar!' + Style.RESET\_ALL)

            print(f'\nVelocidade atual do veículo:' + Style.BRIGHT, Fore.YELLOW + f'{velocidade}km/h\n' + Style.RESET\_ALL)

            os.system('pause')

            os.system('cls')

            main(0, 0, 0, 0, 0, 0, 0)

        else:

            print(Style.BRIGHT, Fore.RED + f'\nErro no processamento dos dados!\n' + Style.RESET\_ALL)

            os.system('pause')

            os.system('cls')

            main(0, 0, 0, 0, 0, 0, 0)

    elif vel\_carro\_passou\_radar\_1 == False:

        if range\_carro\_radar\_1 == True:

            print(Style.BRIGHT, Fore.GREEN + f'\nVocê está dentro da velocidade permitida!' + Style.RESET\_ALL)

            print(f'\nVelocidade máxima permitida: {RADAR\_1} km/h')

            print(f'\nVelocidade atual do veículo:' +  Style.BRIGHT, Fore.GREEN + f'{velocidade}km/h\n' + Style.RESET\_ALL)

            os.system('pause')

            os.system('cls')

            main(0, 0, 0, 0, 0, 0, 0)

        elif range\_carro\_radar\_1 == False:

            print(Style.BRIGHT, Fore.YELLOW + f'\nVocê não se encontra na área de alcance do radar!' + Style.RESET\_ALL)

            print(f'\nVelocidade atual do veículo:' + Style.BRIGHT, Fore.YELLOW + f'{velocidade}km/h\n' + Style.RESET\_ALL)

            os.system('pause')

            os.system('cls')

            main(0, 0, 0, 0, 0, 0, 0)

        else:

            print(Style.BRIGHT, Fore.RED + f'\nErro no processamento dos dados!\n' + Style.RESET\_ALL)

            os.system('pause')

            os.system('cls')

            main(0, 0, 0, 0, 0, 0, 0)

    else:

        print(Style.BRIGHT, Fore.RED + f'\nErro no processamento dos dados!\n' + Style.RESET\_ALL)

        os.system('pause')

        os.system('cls')

        main(0, 0, 0, 0, 0, 0, 0)

main(0, 0, 0, 0, 0, 0, 0)

=============================

'''

Flag (Bandeira) - Marcar um local

None = Não Valor

is e is not = É ou não é (tipo, valor, identidade)

id = Identidade (local na memória)

'''

import os

os.system('color 1f')

v1 = 'a'

print(id(v1))

os.system('pause')

os.system('cls')

condicao = False

passou\_no\_if = None

if condicao == True:

    passou\_no\_if = True

    print('\nVerdadeiro\n')

elif condicao == False:

    print('\nFalso\n')

else:

    print('\nErro no processamento!\n')

if passou\_no\_if is not None:

    print('Passou no if\n')

elif passou\_no\_if is None:

    print('Não passou no if\n')

else:

    print('Erro no processamento!\n')

os.system('pause')

os.system('cls')

=============================

# EXERCÍCIOS

import os

os.system('color 1f')

'''

\* Faça um programa que peça ao usuário para digitar um número inteiro;

\* Informe se este número é par ou Ímpar;

\* Caso o usuário não digite um número inteiro, informe que não é um número inteiro;

'''

num\_inteiro = input('\nDigite um número inteiro: ')

os.system('cls')

if num\_inteiro.isdigit() == True:

    if int(num\_inteiro) % 2 == 0:

        print(f'\nO número {num\_inteiro} é par.\n')

    elif int(num\_inteiro) % 2 == 1:

        print(f'\nO número {num\_inteiro} é ímpar.\n')

    else:

        print(f'\nErro no processamento do dado repassado!\n')

elif num\_inteiro.isdigit() == False:

    print(f'\nO número {num\_inteiro} não é um número inteiro!\n')

else:

    print(f'\nErro no processamento do dado repassado!\n')

os.system('pause')

os.system('cls')

'''

\* Faça um programa que pergunte a hora ao usuário;

\* Baseando-se no horário informado, exiba a saudação apropriada;

\* Ex: Bom dia 0-11, Boa tarde 12-17, Boa noite 18-23.

'''

hrs = input('\nHora: ')

min = input('\nMinuto: ')

os.system('cls')

if hrs.isdigit() and min.isdigit() == True:

    if int(hrs) >= 0 and int(hrs) <= 11:

        if int(min) >= 0 and int(min) <= 59:

            print(f'\nBom dia, são {hrs}:{min}\n')

        elif int(min) < 0 or int(min) > 59:

            print('\nFavor inserir os minutos entre 0 a 59!\n')

        else:

            print('\nErro de processamento dos dados repassados\n')

    elif int(hrs) >= 12 and int(hrs) <= 17:

        if int(min) >= 0 and int(min) <= 59:

            print(f'\nBoa tarde, são {hrs}:{min}\n')

        elif int(min) < 0 or int(min) > 59:

            print('\nFavor inserir os minutos entre 0 a 59!\n')

        else:

            print('\nErro de processamento dos dados repassados\n')

    elif int(hrs) >= 18 and int(hrs) <= 23:

        if int(min) >= 0 and int(min) <= 59:

            print(f'\nBoa noite, são {hrs}:{min}\n')

        elif int(min) < 0 or int(min) > 59:

            print('\nFavor inserir os minutos entre 0 a 59!\n')

        else:

            print('\nErro de processamento dos dados repassados\n')

    elif int(hrs) < 0 or int(hrs) > 23:

        print('\nFavor inserir as horas entre 0 a 23!\n')

    else:

        print('\nErro de processamento dos dados repassados\n')

elif hrs.isdigit() == False or min.isdigit() == False:

    print('\nFavor inserir números inteiros!\n')

else:

    print('\nErro de processamento dos dados repassados\n')

os.system('pause')

os.system('cls')

'''

\* Faça um programa que peça o primeiro nome do usuário;

\* Se o nome tiver 4 letras ou menos, escreva 'Seu nome é curto';

\* Se tiver entre 5 e 6 letras, escreva 'Seu nome 'normal';

\* Maior que 6 letras escreva 'Seu nome é muito grande'.

'''

nome = input('\nQual o seu primeiro nome: ')

os.system('cls')

if bool(nome) == True:

    if len(nome) <= 1:

        print('\nFavor inserir seu nome correto!\n')

    elif len(nome) <= 4:

        print('\nSeu nome é curto!\n')

    elif len(nome) >= 5 and len(nome) <= 6:

        print('\nSeu nome é normal!\n')

    elif len(nome) > 6:

        print('\nSeu nome é muito grande!\n')

elif nome == '':

    print('\nDesculpe, você deixou algum dos campos vazio.')

    print('\nFavor preencher os campos corretamente!\n')

else:

    print('\nErro ao identificar os campos!\n')

os.system('pause')

os.system('cls')

=============================

=============================

'''

https://docs.python.org/pt-br/3/libary/stdtypes.html

Imutáveis que vimos: str, int, float, bool

'''

import os

os.system('color 1f')

string = 'joão augusto'

string\_2 = string

string\_3 = f'{string[:3]}0{string[4:]}'

print(f'\n{string}')

print(f'\n{string\_2}')

print(f'\n{string\_3}\n')

os.system('pause')

os.system('cls')

print(f'\n{string.capitalize()}\n')

os.system('pause')

os.system('cls')

=============================

'''

Repetições

while (enquanto)

Executa uma ação enquanto uma condição for verdadeira.

'''

import os

os.system('color 1f')

cond = True

while cond:

    name = input('\nQual o seu nome: ')

    os.system('cls')

    if name == 'Joao':

        print(f'\nSeu nome é {name} e está certo\n')

        os.system('pause')

        os.system('cls')

        break

    elif name != 'Joao':

        if name == 'exit' or 'sair':

            print('\nSaindo...\n')

            os.system('pause')

            os.system('cls')

            break

        else:

            print(f'\nSeu nome é {name} e está errado\n')

            os.system('pause')

            os.system('cls')

    else:

        print('\nErro no processamento do dado repassado\n')

        os.system('pause')

        os.system('cls')

=============================

'''

Repetições

while (enquanto)

Executa uma ação enquanto uma condição for verdadeira

Loop infinito -> Quando um código não tem fim

'''

import os

os.system('color 1f')

cont = 0

while cont < 10:

    cont = cont + 1

    print(f'\n\_{cont}\_')

print('\nAcabou\n')

os.system('pause')

os.system('cls')

=============================

'''

Operadores de atribuição

= += -= \*= /= //= \*\*= %=

'''

import os

os.system('color 1f')

contador = 0

while contador < 10:

    contador += 1 # Atribuindo o operador +=

    print(f'\n{contador}')

print('\nAcabou\n')

# Exemplo de concatenação com operador de atribuição +=

cont = 'a'

print(f'{cont}\n')

cont += 'bc'

print(f'{cont}\n')

# Exemplo de multiplicar str com operador de atribuição \*=

mult = '='

mult \*= 30

print(f'{mult}\n')

os.system('pause')

os.system('cls')

=============================

'''

Repetições

while (enquanto)

Executa uma ação enquanto uma condição for verdadeira

Loop infinito -> Quando um código não tem fim

'''

import os

os.system('color 1f')

contador = 0

while True:

    contador += 1

    if contador % 2 != 0:

        continue

    print(f'\n{contador}')

    if contador >= 10:

        break

print('\nAcabou\n')

os.system('pause')

os.system('cls')

=============================

'''

Repetições

while (enquanto)

Executa uma ação enquanto uma condição for verdadeira

Loop infinito -> Quando um código não tem fim

'''

import os

os.system('color 1f')

qtd\_lines = 5

qtd\_coluns = 5

line = 1

while line <= qtd\_lines:

    colun = 1

    while colun <= qtd\_coluns:

        print(f'{line} - {colun}')

        colun += 1

    print('\n')

    line += 1

print('Acabou\n')

os.system('pause')

os.system('cls')

=============================

'''

Iterando strings com while

'''

import os

os.system('color 1f')

nome = input('Nome: ')

tamanho\_nome = len(nome)

x = 0

nome\_atualizado = ''

while x < tamanho\_nome:

    letra = nome[x]

    nome\_atualizado += f'\*{letra}'

    x += 1

nome\_atualizado += '\*'

print(nome\_atualizado)

os.system('pause')

os.system('cls')

=============================

'''

Calculadora com while

.lower() - Converti maiusculo para minusculo.

.startwith - Começa com...

.endswith - Termina com...

except Exception as error  - apresenta o erro ocorrido. Ex:

try:

    ...

except Exception as error:

    print(error)

'''

import os

os.system('color 1f')

try:

    while True:

        print('\n' + '=' \* 10 + 'Calculadora' + '=' \* 10)

        print('\n[A] - Acessar')

        print('\n[S] - Sair')

        print('\n' + '=' \* 31)

        op = input('\nEscolha uma opção: ').lower()

        os.system('cls')

        try:

            if op == 'a':

                print('\n' + '=' \* 10 + 'Operadores' + '=' \* 10)

                print('\n[1] - Soma')

                print('\n[2] - Subtração')

                print('\n[3] - Multiplicação')

                print('\n[4] - Divisão')

                print('\n' + '=' \* 30)

                oper = int(input('\nEcolha um operador: '))

                os.system('cls')

                if oper == 1 or oper == 2 or oper == 3 or oper == 4:

                    n1 = float(input('\nDigite o primeiro número: '))

                    n2 = float(input('\nDigite o segundo número:'))

                    os.system('cls')

                    if oper == 1:

                        res = n1 + n2

                        print(f'\nO resultado de {n1:.2f} + {n2:.2f} = {res:.2f}\n')

                        os.system('pause')

                        os.system('cls')

                    elif oper == 2:

                        res = n1 - n2

                        print(f'\nO resultado de {n1:.2f} - {n2:.2f} = {res:.2f}\n')

                        os.system('pause')

                        os.system('cls')

                    elif oper == 3:

                        res = n1 \* n2

                        print(f'\nO resultado de {n1:.2f} X {n2:.2f} = {res:.2f}\n')

                        os.system('pause')

                        os.system('cls')

                    elif oper == 4:

                        res = n1 / n2

                        print(f'\nO resultado de {n1:.2f} / {n2:.2f} = {res:.2f}\n')

                        os.system('pause')

                        os.system('cls')

                    else:

                        os.system('cls')

                        print('\nErro na opção repassada!!!\n')

                        print('Favor verificar solicitação ou entrar em contato com o suporte técnico!\n')

                        os.system('pause')

                        os.system('cls')

                else:

                    os.system('cls')

                    print('\nErro na opção repassada!!!\n')

                    print('Favor verificar solicitação ou entrar em contato com o suporte técnico!\n')

                    os.system('pause')

                    os.system('cls')

            elif op == 's':

                print('\nSaindo...\n')

                os.system('pause')

                os.system('cls')

                break

            else:

                os.system('cls')

                print('\nErro na opção repassada!!!\n')

                print('Favor verificar solicitação ou entrar em contato com o suporte técnico!\n')

                os.system('pause')

                os.system('cls')

        except Exception as error:

            os.system('cls')

            print(f'\nErro: {error}\n')

            print('Favor verificar solicitação ou entrar em contato com o suporte técnico!\n')

            os.system('pause')

            os.system('cls')

except Exception as error:

    os.system('cls')

    print(f'\nErro: {error}\n')

    print('Favor entrar em contato com o suporte técnico!\n')

    os.system('pause')

    os.system('cls')

=============================

'''

while / else

\* Quando o laço while é executado completamente, o else é executado.

\* Quando é executado um break no while, não será executado o else também.

'''

import os

os.system('color 1f')

valor1 = 'Repassando um valor'

x = 0

y = 0

print('\n')

while x < len(valor1):

    letra = valor1[x]

    if letra == ' ':

        print('\nEncontrei um espaço vazio!\n')

        break

    print(letra)

    x += 1

else:

    print('\nNão encontrei espaço vazio!\n')

os.system('pause')

os.system('cls')

####################################

valor2 = 'Repassandoumvalor'

print('\n')

while y < len(valor2):

    letra = valor2[y]

    if letra == ' ':

        print('\nEncontrei um espaço vazio!\n')

        break

    print(letra)

    y += 1

else:

    print('\nNão encontrei espaço vazio!\n')

os.system('pause')

os.system('cls')

=============================

'''

.count(palavra) -  Método para buscar quantas vezes a palavra ou letra foi repassada na frase ou texto.

.upper() - Insere o texto ou palavra todo MAIÚSCULO.

.lower() - Insere o texto ou palavra todo minúsculo.

'''

import os

os.system('color 1f')

# \ - quebra de linha no código

frase = 'O python é uma linguagem de programação'\

        'multiparadigma. '\

        'Python foi criado por Guido Van Rossum.'.upper()

print(f'\nQuantas vezes apareceu a palavra "Python": {frase.count('PYTHON')}')

print(f'\nQuantas vezes apareceu a letra "A": {frase.count('A')}\n')

os.system('pause')

os.system('cls')

i = 0

qtd\_mais\_vzs\_letra = 0

mais\_vzs\_letra = ''

while i < len(frase):

    letra\_atual = frase[i]

    if letra\_atual == ' ':

        i += 1

        continue

    n\_vez\_letra = frase.count(letra\_atual)

    if qtd\_mais\_vzs\_letra < n\_vez\_letra:

        qtd\_mais\_vzs\_letra = n\_vez\_letra

        mais\_vzs\_letra = letra\_atual

    i += 1

print(f'\nA letra que mais apareceu foi "{mais\_vzs\_letra}", sendo {qtd\_mais\_vzs\_letra} vezes.\n')

os.system('pause')

os.system('cls')

=============================

'''

Laço de repetição FOR/IN

\* Utilizar o while quando não saber quantas repetições serão necessárias

'''

import os

os.system('color 1f')

texto = 'Python'

i = 0

texto\_modificado = ''

print('\n')

for letra in texto:

    i += 1

    texto\_modificado += f'\*{letra}'

    print(f'{letra} - {i}')

print(f'\n{texto\_modificado}\*\n')

os.system('pause')

os.system('cls')

=============================

'''

For + Range

range -> range(start - Início, stop - Final, step - Pular de quantos em quantos)

'''

import os

os.system('color 1f')

numeros1 = range(11)

numeros2 = range (5, 11)

numeros3 = range(0, 11, 2)

numeros4 = range(0, -11, -2)

print('\n')

for valor1 in numeros1:

    print(valor1)

print('\n')

os.system('pause')

os.system('cls')

print('\n')

for valor2 in numeros2:

    print(valor2)

print('\n')

os.system('pause')

os.system('cls')

print('\n')

for valor3 in numeros3:

    print(valor3)

print('\n')

os.system('pause')

os.system('cls')

print('\n')

for valor4 in numeros4:

    print(valor4)

print('\n')

os.system('pause')

os.system('cls')

=============================

'''

\* Iterável -> str, range, entre outros (\_\_iter\_\_)

\* Iterador -> Quem sabe entregar um valor por vez

\* next -> Me entregue o próximo valor

\* iter -> Me entregue seu iterador

'''

import os

os.system('color 1f')

numeros = range(0, 101, 8)

tst = iter('Joao')

texto\_espaco = ' ' + 'Joao'

texto = iter(texto\_espaco)

txt = next(texto)

print('\n')

for numero in numeros:

    print(numero)

print('\n')

os.system('pause')

os.system('cls')

# iter - Indica o endereço onde o objeto do interador está na memória.

print(f'\n{texto}\n')

os.system('pause')

os.system('cls')

print('\n')

# Quando finaliza as informações repassadas, mas o next continua buscando, apresenta erro de 'StopIteration', realizando o cancelamento da próxima busca.

for txt in texto:

    print(txt)

print('\n')

os.system('pause')

os.system('cls')

print('\n')

while True:

    try:

        txt\_rep = next(tst)

        print(txt\_rep)

    except StopIteration:

        print('\nErro StopIteration tratado!')

        break

print('\n')

os.system('pause')

os.system('cls')

=============================

'''

RANGE NO FOR

'''

import os

os.system('color 1f')

print('\n')

for i in range(10):

    if i == 2:

        print('i é 2, pulando...')

        continue

    if i == 8:

        print('i é 8, seu else não executará!')

        break

    for j in range(1, 3):

        print(i, j)

else:

    print('For completo com sucesso!')

print('\n')

os.system('pause')

os.system('cls')

=============================

'''

Faça um jogo para o usuário adivinhar qual a palavra secreta.

\* Você vai propor uma palavra secreta qualquer e vai dar a possibilidade para o usuário digitar apenas uma letra.

\* Quando o usuário digitar uma letra, você vai conferir se a letra digitada está na palavra secreta.

\* Se a letra digitada estiver na palavra secreta, exiba a letra

\* Se a letra digitada não estar na palavra secreta, exiba "\*"

Faça a contagem de tentativas do seu usuário.

'''

import os

import random

os.system('Color 1f')

def main(op):

    print('\n' + '=' \* 10 + 'PALAVRA SECRETA' + '=' \* 10 + '\n')

    print('[C] - Começar o Jogo')

    print('[S] - Sair')

    print('\n' + '=' \* 35 + '\n')

    op = input('Opção: ')

    os.system('cls')

    if op == '' or op == ' ':

        print('\nFavor repassar a opção desejada!!!\n')

        os.system('pause')

        os.system('cls')

        main(op)

    elif op.upper() == 'C':

        jogo()

    elif op.upper() == 'S':

        print('\nSaindo...\n')

        os.system('pause')

        os.system('cls')

    elif op.upper() != 'C' or op.upper() != 'S':

        print('\nFavor digitar uma das opções repassadas!!!\n')

        os.system('pause')

        os.system('cls')

        main(op = '')

    else:

        print('\nErro no processamento dos dados!!!\n')

        os.system('pause')

        os.system('cls')

        main(op = '')

os.system('exit')

def jogo():

    palavras\_secretas = ['teste', 'banana', 'dia', 'noite', 'tarde', 'sol', 'lua', 'cafe', 'amor', 'pao', 'pai', 'mae', 'filho', 'filha', 'neto', 'neta', 'avo', 'tio', 'tia']

    palavra\_secreta\_sorteada = random.choice(palavras\_secretas)

    esconde\_palavra\_secreta = ''

    letras\_acertadas = ''

    i = 0

    for letra in palavra\_secreta\_sorteada:

        esconde\_palavra\_secreta += '\*'

        i += 1

    print(f'\nA palavra sortiada é: {esconde\_palavra\_secreta}\n')

    tentativas = 0

    erros = 0

    while True:

        if erros == 10:

            print('Que Pena! Você perdeu!\n')

            os.system('pause')

            os.system('cls')

            main(op = '')

        letra\_secreta = input('Digite a letra que faz parte da palavra secreta (Para retornar, escreva "menu"): ')

        tentativas += 1

        os.system('cls')

        if letra\_secreta.upper() == 'MENU':

            main(op = '')

        elif len(letra\_secreta) > 1:

            print('\nFavor repassar somente uma letra!!!\n')

            os.system('pause')

            os.system('cls')

            jogo()

        elif letra\_secreta == '' or letra\_secreta == ' ':

            print('\nFavor digitar uma letra!!!\n')

        elif letra\_secreta in palavra\_secreta\_sorteada:

            print('\nLetra certa!')

            letras\_acertadas += letra\_secreta

        elif letra\_secreta not in palavra\_secreta\_sorteada:

            erros += 1

            print('\nLetra errada!')

        else:

            print('\nErro no processamento dos dados!!!')

        palavra\_formada = ''

        for letra\_secreta in palavra\_secreta\_sorteada:

            if letra\_secreta in letras\_acertadas:

                palavra\_formada += letra\_secreta

            else:

                palavra\_formada += '\*'

        print(f'\nPalavra Secreta: {palavra\_formada}\n')

        print(f'Erros: {erros} vezes\n')

        print(f'Tentativas: {tentativas} vezes\n')

        if palavra\_formada == palavra\_secreta\_sorteada:

            print('PARABÉNS!!! VOCÊ GANHOU!!!\n')

            os.system('pause')

            os.system('cls')

            main(op = '')

main(op = '')

=============================

'''

Listas em Python

Tipo list - Mutável

Suporta vários valores de qualquer tipo

Conhecimentos reutilizáveis - índices e fatiamento

Métodos úteis: append, insert, pop, del, clear, extend, +

'''

import os

os.system('color 1f')

string = 'ABCDE'

lista = [123, True, 'Joao', 1.2, []]

print(f'\n{lista[0]}')

print(f'\n{lista[1]}')

print(f'\n{lista[2]}')

print(f'\n{lista[3]}')

print(f'\n{lista[4]}')

print('\n' + '=' \* 30 + '\n')

lista[2] = 'Maria'

print(f'{lista[2]}\n')

os.system('pause')

os.system('cls')

=============================

'''

Listas em Python

Tipo list - Mutável

Suporta vários valores de qualquer tipo

Conhecimentos reutilizáveis - índices e fatiamento

Métodos úteis: append, insert, pop, del, clear, extend, +

CRUD:

Create Read Update Delete

Criar, ler, alterar , apagar = lista[i] (CRUD)

\* Para apagar um item da lista, opte em transferir esse item para o final da lista para apagar.

'''

import os

os.system('color 1f')

lista = [10, 20, 30, 40] # Lista.

# Visualização da lista inteira e do item que está no índice 2.

print(f'\n{lista}\n')

print(f'{lista[2]}\n')

print('=' \* 30 + '\n')

# Modificando o índice 2 da lista e imprimindo na tela a lista modificada.

lista[2] = 300

print(f'{lista}\n')

print('=' \* 30 + '\n')

# Deletando o item do índice 2 da lista e imprimindo a lista modificada.

del lista[2]

print(f'{lista}\n')

print('=' \* 30 + '\n')

# Modificando o índice 2 e adicionando um item na lista, onde esse adicionamento é posicionado no final da mesma, e imprimindo a lista modificada na tela.

lista[2] = 30

lista.append(40)

print(f'{lista}\n')

print('=' \* 30 + '\n')

# Removendo o último item da lista e imprimindo na tela a lista modificada.

lista.pop()

print(f'{lista}\n')

print('=' \* 30 + '\n')

# Removendo o item do índice 0 da lista e imprimindo na tela a lista modificada.

lista.pop(0)

print(f'{lista}\n')

print('=' \* 30 + '\n')

os.system('pause')

os.system('cls')

=============================

'''

Conhecimentos reutilizáveis - índices e fatiamento

Métodos úteis:

    append - Adiciona um item ao final

    insert - Adiciona um item no índice escolhido

    pop - Remove do final ou do índice escolhido

    del - Apaga um índice

    clear - Limpa a lista

    extend - Estende a lista

    + - Concatena listas

CRUD:

Create | Read | Update  | Delete

Criar  | Ler  | Alterar | Apagar

lista[i] (CRUD)

lista.insert(índice, item)

'''

import os

os.system('color 1f')

lista = [10, 20, 30, 40] # Lista

print('\n' + '=' \* 30 + '\n')

# Adicionando um item na lista, onde esse adicionamento é posicionado no final da mesma, e imprimindo a lista modificada na tela.

lista.append(50)

print(f'{lista}\n')

print('=' \* 30 + '\n')

# Removendo o último item da lista e imprimindo na tela a lista modificada.

lista.pop()

print(f'{lista}\n')

print('=' \* 30 + '\n')

# Adicionando novamente um item na lista, removendo esse item de uma forma diferente, e imprimindo a lista modificada na tela.

lista.append(50)

print(f'{lista}\n')

print('=' \* 30 + '\n')

del lista[-1]

print(f'{lista}\n')

print('=' \* 30 + '\n')

# Limpando a lista completa e exibindo a mesma vazia

lista.clear()

print(f'{lista}\n')

print('=' \* 30 + '\n')

#  Adicionando novamente os itens na lista

lista.append(10)

lista.append(20)

lista.append(30)

lista.append(40)

print(f'{lista}\n')

print('=' \* 30 + '\n')

# Adicionando um item na lista, onde esse adicionamento é possicionado a partir do índice da mesma, repassado no insert.

lista.insert(0, 0)

print(f'{lista}\n')

print('=' \* 30 + '\n')

os.system('pause')

os.system('cls')

=============================

'''

Conhecimentos reutilizáveis - índices e fatiamento

Métodos úteis:

    append - Adiciona um item ao final

    insert - Adiciona um item no índice escolhido

    pop - Remove do final ou do índice escolhido

    del - Apaga um índice

    clear - Limpa a lista

    extend - Estende a lista

    + - Concatena listas

CRUD:

Create | Read | Update  | Delete

Criar  | Ler  | Alterar | Apagar

lista[i] (CRUD)

lista.insert(índice, item)

'''

import os

os.system('color 1f')

# Repassando as listas A e B, concatenando em uma lista C.

lista\_a = [1, 2, 3]

lista\_b = [4, 5, 6]

lista\_c = lista\_a + lista\_b

print('\n' + '=' \* 30 + '\n')

print(f'{lista\_c}\n')

print('=' \* 30 + '\n')

# Concatenando as listas A e B na própria lista A.

lista\_a.extend(lista\_b)

print(f'{lista\_a}\n')

print('=' \* 30 + '\n')

os.system('pause')

os.system('cls')

=============================

'''

Cuidados com dados mutáveis

= - Copiado o valor (imutáveis)

= - Aponta para o mesmo valor na memória (mutável)

lista.copy() - Copia a lista para outra lista

'''

import os

os.system('color 1f')

lista\_a = ['Joao', 1, True, 1.2]

lista\_b = lista\_a.copy()

lista\_a[0] = 'Maria'

print(f'\n{lista\_a}')

print(f'\n{lista\_b}\n')

os.system('pause')

os.system('cls')

=============================

'''

for in com listas

'''

import os

os.system('color 1f')

lista = ['Maria', 'Helena', 'Joao']

print('\n' + '=' \* 30)

for nome in lista:

    print(f'\n= {nome} =')

    print('\n' + '=' \* 30)

print('\n')

os.system('pause')

os.system('cls')

=============================

'''

EXERCÍCIO

Exiba os índices da lista

'''

import os

os.system('color 1f')

lista = ['Maria', 'Helena', 'Joao']

lista.append('Luiz')

indices = range(len(lista))

print('\n' + '=' \* 30)

for indice in indices:

    print(f'\n{indice} - {lista[indice]}')

    print('\n' + '=' \* 30)

print('\n')

os.system('pause')

os.system('cls')

=============================

'''

Introdução ao desempacotamento + tuples (tuplas)

( \_ ) - É uma variável que não será utilizada

'''

import os

os.system('color 1f')

nomes = ['Maria', 'Helena', 'Joao']

nome1, nome2, nome3 = nomes

print('\n' + '\*' \* 30)

print(f'\n{nome1} - {nome2} - {nome3}')

nome1, \*resto = nomes

print('\n' + '\*' \* 30)

print(f'\n{nome1} - {resto}')

nome1, \*\_ = nomes

print('\n' + '\*' \* 30)

print(f'\n{nome1} - {\_}')

\_, nome2, \*\_ = nomes

print('\n' + '\*' \* 30)

print(f'\n{\_} - {nome2} - {\_}')

\_, \_, nome3, \*\_ = nomes

print('\n' + '\*' \* 30)

print(f'\n{\_} - {\_} - {nome3}')

print('\n' + '\*' \* 30 + '\n')

os.system('pause')

os.system('cls')

=============================

'''

Tipo tupla - Uma lista imutável

\* Não é possível realizar a mudança de algum dado da lista

"nomes[1] = mudar" - Apresentará erro TypeError: 'tuple' object does not support item assignment.

tuple() - Função para transformar uma lista mutável em tupla.

list() - Função para transformar uma tupla em lista mutável.

'''

import os

os.system('color 1f')

nomes = ('Maria', 'Helena', 'Luiz')

print(f'\n{nomes}\n')

print(f'{nomes[1]}\n')

nome = ['Maria', 'Helena', 'Luiz']

print(f'{tuple(nome)}\n')

os.system('pause')

os.system('cls')

=============================

'''

Enumerate - Enumera iteráveis (índices)

enumerate()

\t - TAB

'''

import os

os.system('color 1f')

lista = ['Maria', 'Helena', 'Luiz']

lista.append('João')

print('\n')

for indice, nome in enumerate(lista):

    print(indice, nome)

print('\n')

# Outra forma de fazer

lista\_enumerada = list(enumerate(lista))

print(f'{lista\_enumerada}\n')

os.system('pause')

os.system('cls')

=============================

'''

EXERCÍCIO

Faça uma lista de compras com listas

O usuário deve ter a possibilidade de inserir, apagar e listar valores da sua lista

Não permita que o programa quebre com erros de índices inexistentes na lista.

'''

import os, getpass

os.system('color 1f')

try:

    def main():

        print('\n' + '-' \* 5 + 'LISTA DE COMPRA' + '-' \* 5 + '\n')

        print('[A] - Acessar o sistema')

        print('[S] - Sair')

        print('\n' + '-' \* 25 + '\n')

        op = input('Escolha uma opção: ')

        os.system('cls')

        if op == '' or op == ' ':

            print('\nFavor repassar a opção desejada!!!\n')

            os.system('pause')

            os.system('cls')

            main()

        elif op.upper() == 'A':

            senha = getpass.getpass('\nFavor repassar a senha de acesso: ')

            os.system('cls')

            while True:

                if senha == 'AcessoLista':

                    lista\_compra()

                    break

                elif senha == '' or senha == ' ':

                    print('\nFavor repassar a senha solicitada!!!\n')

                    os.system('pause')

                    os.system('cls')

                    main()

                    break

                elif senha != 'AcessoLista':

                    print('\nSENHA ERRADA!!!')

                    print('\nFavor inserir senha correta!\n')

                    os.system('pause')

                    os.system('cls')

                    main()

                    break

                else:

                    print('\nERRO!!!')

                    print('\nFavor entrar em contato com o suporte do sistema.')

                    os.system('pause')

                    os.system('cls')

                    main()

                    break

        elif op.upper() == 'S':

            print('\nSaindo...\n')

            os.system('pause')

            os.system('cls')

        elif op.upper() != 'A' or op.upper() != 'S':

            print('\nOPÇÃO ERRADA!!!')

            print('\nFavor inserir opção certa.\n')

            os.system('pause')

            os.system('cls')

            main()

        else:

            print('\nERRO!!!')

            print('\nFavor entrar em contato com o suporte do sistema.')

            os.system('pause')

            os.system('cls')

    def lista\_compra():

        lista = []

        while True:

            print('\n' + '-' \* 25 + '\n')

            print('[1] - Adicionar produto')

            print('[2] - Mostrar lista')

            print('[3] - Apagar produto')

            print('[4] - Retornar ao menu')

            print('\n' + '-' \* 25 + '\n')

            opcao = input('Escolha uma opção: ')

            os.system('cls')

            if opcao == '' or opcao == ' ':

                print('\nFavor repassar a opção desejada!!!\n')

                os.system('pause')

                os.system('cls')

            elif opcao == '1':

                ad\_prod = input('\nProduto: ')

                lista.append(ad\_prod)

                os.system('cls')

            elif opcao == '2':

                print('\nÍndice | Produto')

                print('----------------')

                for indice, prod in enumerate(lista):

                    print(f'   {indice}   | {prod}')

                print('----------------\n')

                os.system('pause')

                os.system('cls')

            elif opcao == '3':

                ap\_prod = int(input('\nÍndice do produto: '))

                del lista[ap\_prod]

                os.system('cls')

            elif opcao == '4':

                main()

                break

            elif opcao != '1' or opcao != '2' or opcao != '3' or opcao != '4':

                print('\nOPÇÃO ERRADA!!!')

                print('\nFavor inserir opção certa.\n')

                os.system('pause')

                os.system('cls')

            else:

                print('\nERRO!!!')

                print('\nFavor entrar em contato com o suporte do sistema.')

                os.system('pause')

                os.system('cls')

                os.system('exit')

                break

except:

    print('\nERRO!!!')

    print('\nFavor entrar em contato com o suporte do sistema.')

    os.system('pause')

    os.system('cls')

main()

=============================

'''

EXERCÍCIO

Faça uma lista de compras com listas

O usuário deve ter a possibilidade de inserir, apagar e listar valores da sua lista

Não permita que o programa quebre com erros de índices inexistentes na lista.

'''

import os, getpass

os.system('color 1f')

try:

def main():

print('\n' + '-' \* 5 + 'LISTA DE COMPRA' + '-' \* 5 + '\n')

print('[A] - Acessar o sistema')

print('[S] - Sair')

print('\n' + '-' \* 25 + '\n')

op = input('Escolha uma opção: ')

os.system('cls')

if op == '' or op == ' ':

print('\nFavor repassar a opção desejada!!!\n')

os.system('pause')

os.system('cls')

main()

elif op.upper() == 'A':

senha = getpass.getpass('\nFavor repassar a senha de acesso: ')

os.system('cls')

while True:

if senha == 'AcessoLista':

lista\_compra()

break

elif senha == '' or senha == ' ':

print('\nFavor repassar a senha solicitada!!!\n')

os.system('pause')

os.system('cls')

main()

break

elif senha != 'AcessoLista':

print('\nSENHA ERRADA!!!')

print('\nFavor inserir senha correta!\n')

os.system('pause')

os.system('cls')

main()

break

else:

print('\nERRO!!!')

print('\nFavor entrar em contato com o suporte do sistema.')

os.system('pause')

os.system('cls')

main()

break

elif op.upper() == 'S':

print('\nSaindo...\n')

os.system('pause')

os.system('cls')

elif op.upper() != 'A' or op.upper() != 'S':

print('\nOPÇÃO ERRADA!!!')

print('\nFavor inserir opção certa.\n')

os.system('pause')

os.system('cls')

main()

else:

print('\nERRO!!!')

print('\nFavor entrar em contato com o suporte do sistema.')

os.system('pause')

os.system('cls')

def lista\_compra():

lista = []

while True:

print('\n' + '-' \* 25 + '\n')

print('[1] - Adicionar produto')

print('[2] - Mostrar lista')

print('[3] - Apagar produto')

print('[4] - Retornar ao menu')

print('\n' + '-' \* 25 + '\n')

opcao = input('Escolha uma opção: ')

os.system('cls')

if opcao == '' or opcao == ' ':

print('\nFavor repassar a opção desejada!!!\n')

os.system('pause')

os.system('cls')

elif opcao == '1':

ad\_prod = input('\nProduto: ')

lista.append(ad\_prod)

os.system('cls')

elif opcao == '2':

print('\nÍndice | Produto')

print('----------------')

for indice, prod in enumerate(lista):

print(f' {indice} | {prod}')

print('----------------\n')

os.system('pause')

os.system('cls')

elif opcao == '3':

ap\_prod = input('\nÍndice do produto: ')

os.system('cls')

try:

ap\_prod\_int = int(ap\_prod)

del lista[ap\_prod\_int]

os.system('cls')

except IndexError:

print('\nÍNDICE ERRADO!!!')

print('\nFavor inserir índice certo.\n')

os.system('pause')

os.system('cls')

except ValueError:

                    print('\nÍNDICE ERRADO!!!')

                    print('\nFavor inserir número para índice.\n')

                    os.system('pause')

                    os.system('cls')

                except Exception:

                    print('\nErro Desconhecido.')

                    print('\nFavor entrar em contato com o suporte do sistema.\n')

                    os.system('pause')

                    os.system('cls')

except:

print('\nERRO!!!')

print('\nNão foi possível repassar os dados para o sistema.\n')

os.system('pause')

os.system('cls')

elif opcao == '4':

main()

break

elif opcao != '1' or opcao != '2' or opcao != '3' or opcao != '4':

print('\nOPÇÃO ERRADA!!!')

print('\nFavor inserir opção certa.\n')

os.system('pause')

os.system('cls')

else:

print('\nERRO!!!')

print('\nFavor entrar em contato com o suporte do sistema.\n')

os.system('pause')

os.system('cls')

os.system('exit')

break

except:

print('\nERRO!!!')

print('\nFavor entrar em contato com o suporte do sistema.\n')

os.system('pause')

os.system('cls')

main()

=============================

'''

Imprecisão de ponto flutuante

Double-precision floating-point format IEEE 754

'''

import os, decimal

os.system('color 1f')

numero\_1 = 0.1

numero\_2 = 0.7

numero\_3 = numero\_1 + numero\_2

# Apresenta o problema de imprecisão, imprimindo o valor 0.7999999999...

print('\n' + '+' \* 10 + '\n')

print(f'Sem Formatação: {numero\_3}')

print('\n' + '+' \* 10 + '\n')

# Primeira forma de contornar, imprimindo 0.80

print(f'Utilizando o Format: {numero\_3:.2f}')

print('\n' + '+' \* 10 + '\n')

# Segunda forma de contornar, imprimindo 0.8

print(f'Utilizando o Round: {round(numero\_3, 1)}')

print('\n' + '+' \* 10 + '\n')

# Terceira forma de contornar, imprimindo

numero\_1 = decimal.Decimal('0.1')

numero\_2 = decimal.Decimal('0.7')

numero\_3 = numero\_1 + numero\_2

print(f'Utilizando a biblioteca decimal: {numero\_3}')

print('\n' + '+' \* 10 + '\n')

os.system('pause')

os.system('cls')

=============================

'''

split e join com list e str

split - divide uma string

join - une uma string

strip() - Corta os espaços do início e final da informação imprimida

'''

import os

os.system('color 1f')

frase = 'Olha só que, coisa interessante'

# split

#######################

lista\_palavras\_1 = frase.split()

print(f'\n{lista\_palavras\_1}\n')

#######################

lista\_palavras\_2 = frase.split(',')

print(f'{lista\_palavras\_2}\n')

#######################

lista\_palavras\_3 = frase.split(', ')

for i, frase in enumerate(lista\_palavras\_3):

    print(lista\_palavras\_3[i])

print('\n')

#######################

# join

#######################

frases\_unidas = '-'.join('ABC')

print(f'{frases\_unidas}\n')

#######################

frases\_unidas\_1 = ' - '.join(lista\_palavras\_3)

print(f'{frases\_unidas\_1}\n')

os.system('pause')

os.system('cls')

=============================

'''

Lista de listas e seus índices

'''

import os

os.system('color 1f')

salas = [

    #   0        1

    ['Maria', 'Helena', ], # 0

    #   0

    ['Elaine', ], # 1

    #  0       1         2                3

    ['Luiz', 'João', 'Eduarda', (0, 10, 20, 30, 40)], # 2

]

print(f'\n{salas}\n')

print(f'{salas[2][1]}\n')

print(f'{salas[2][3][3]}\n')

for sala in salas:

    for aluno in sala:

        print(f'{aluno}')

print('\n')

os.system('pause')

os.system('cls')

=============================

'''

Interpretador do Python

python mod.py (executa o mod)

python -u (unbuffered)

python -m mod (lib mod como script)

python -c 'cmd' (comando)

python -i mod.py (interativo com mod)

The Zen of Python, por Tim Peters

Bonito é melhor que feio.

Explícito é melhor que implícito.

Simples é melhor que complexo.

Complexo é melhor que complicado.

Plano é melhor que aglomerado.

Esparso é melhor que denso.

Legibilidade conta.

Casos especiais não são especiais o bastante para quebrar as regras.

Embora a praticidade vença a pureza.

Erros nunca devem passar silenciosamente.

A menos que sejam explicitamente silenciados.

Diante da ambiguidade, recuse a tentação de adivinhar.

Deve haver um -- e só um -- modo óbvio para fazer algo.

Embora esse modo possa não ser óbvio à primeira vista a menos que você seja holandês.

Agora é melhor que nunca.

Embora nunca frequentemente seja melhor que \*exatamente\* agora.

Se a implementação é difícil de explicar, é uma má ideia.

Se a implementação é fácil de explicar, pode ser uma boa ideia.

Namespaces são uma grande ideia -- vamos fazer mais dessas!

'''

=============================

'''

Desempacotamento em chamadas de métodos e funções

O parâmetro "end" do Python é usado para adicionar qualquer string ao final da saída de print().

Por outro lado, para separar a saída por caractere ou string no lugar do valor de espaço padrão, usamos o parâmetro "sep" do Python.

'''

import os

os.system('color 1f')

string = 'ABCD'

lista = ['Maria', 'Helena', 'Eduarda']

tupla = 'Python', 'é', 'legal'

a, b, c = lista

print(f'\n{a, c}\n')

#########################

for nome in lista:

print(nome, end = ' ')

print('\n')

#########################

print(\*lista + '\n')

#########################

print(\*string + '\n')

#########################

print(\*tupla + '\n')

os.system('pause')

os.system('cls')

=============================

'''

Operação ternária (condicional de uma linha)

<valor> if <condição> else <outro valor>

'''

import os

os.system('color 1f')

##########################

print('\nValor\n' if True else '\nOutro valor\n')

##########################

condicao = 10 == 10

variavel = 'Igual a 10' if condicao else 'Não é igual a 10'

print(f'{variavel}\n')

##########################

digito\_1 = 5

digito\_2 = 12

novo\_digito\_1 = digito\_1 if digito\_1 <= 9 else 0

print(f'{novo\_digito\_1}\n')

novo\_digito\_2 = digito\_2 if digito\_2 <= 9 else 0

print(f'{novo\_digito\_2}\n')

os.system('pause')

os.system('cls')

=============================

'''

Calculo do primeiro dígito do CPF

CPF: 746.824.890-70

Colete a soma dos 9 primeiros dígitos do CPF

multiplicando cada um dos valores por uma

contagem regressiva começando de 10

Ex.:  746.824.890-70 (746824890)

   10  9  8  7  6  5  4  3  2

\*  7   4  6  8  2  4  8  9  0

   70  36 48 56 12 20 32 27 0

Somar todos os resultados:

70+36+48+56+12+20+32+27+0 = 301

Multiplicar o resultado anterior por 10

301 \* 10 = 3010

Obter o resto da divisão da conta anterior por 11

3010 % 11 = 7

Se o resultado anterior for maior que 9:

    resultado é 0

contrário disso:

    resultado é o valor da conta

O primeiro dígito do CPF é 7

'''

import os

os.system('color 1f')

while True:

    cpf = input('\nDigite o CPF (somente números): ')

    os.system('cls')

    if len(cpf) == 11:

        try:

            cpf\_list = list(cpf)

            cpf\_list\_num = list(cpf\_list)

            cpf\_int = [int(i) for i in cpf\_list\_num]

            cpf\_soma = (cpf\_int[0] \* 10) + (cpf\_int[1] \* 9) + (cpf\_int[2] \* 8) + (cpf\_int[3] \* 7) + (cpf\_int[4] \* 6) + (cpf\_int[5] \* 5) + (cpf\_int[6] \* 4) + (cpf\_int[7] \* 3) + (cpf\_int[8] \* 2)

            cpf\_mult = cpf\_soma \* 10

            cpf\_rest = cpf\_mult % 11

            if cpf\_rest > 9:

                calc\_primeiro\_dig = 0

            elif cpf\_rest <= 9 or cpf\_rest >= 0:

                calc\_primeiro\_dig = cpf\_rest

            else:

                print('\nErro!!! Favor entrar em contato com o suporte!\n')

                os.system('pause')

                os.system('cls')

                break

            if calc\_primeiro\_dig == cpf\_int[9]:

                print('\nO primeiro dígito do CPF é verdadeiro!\n')

                os.system('pause')

                os.system('cls')

                break

            elif calc\_primeiro\_dig != cpf\_int[9]:

                print('\nO primeiro dígito do CPF é falso!\n')

                os.system('pause')

                os.system('cls')

            else:

                print('\nErro!!! Favor entrar em contato com o suporte!\n')

                os.system('pause')

                os.system('cls')

                break

        except ValueError:

            print('\nCPF inválido!\n')

            os.system('pause')

            os.system('cls')

    elif cpf == '' or cpf == ' ' or len(cpf) >= 12 or len(cpf) < 11:

        print('\nCPF inválido!\n')

        os.system('pause')

        os.system('cls')

    else:

        print('\nErro!!! Favor entrar em contato com o suporte!\n')

        os.system('pause')

        os.system('cls')

        break

=============================

'''

RESOLUÇÃO DO EXERCÍCIO DE OUTRA FORMA

Calculo do primeiro dígito do CPF

CPF: 746.824.890-70

Colete a soma dos 9 primeiros dígitos do CPF

multiplicando cada um dos valores por uma

contagem regressiva começando de 10

Ex.:  746.824.890-70 (746824890)

   10  9  8  7  6  5  4  3  2

\*  7   4  6  8  2  4  8  9  0

   70  36 48 56 12 20 32 27 0

Somar todos os resultados:

70+36+48+56+12+20+32+27+0 = 301

Multiplicar o resultado anterior por 10

301 \* 10 = 3010

Obter o resto da divisão da conta anterior por 11

3010 % 11 = 7

Se o resultado anterior for maior que 9:

    resultado é 0

contrário disso:

    resultado é o valor da conta

O primeiro dígito do CPF é 7

'''

import os

os.system('color 1f')

while True:

    cpf = input('\nDigite o CPF (somente números): ')

    os.system('cls')

    if len(cpf) == 11:

        try:

            nove\_digitos = cpf[:9]

            contador\_regressivo = 10

            resultado = 0

            for digito in nove\_digitos:

                resultado += int(digito) \* contador\_regressivo

                contador\_regressivo -= 1

            digito = (resultado \* 10) % 11

            digito = digito if digito <= 9 else 0

            if str(digito) == cpf[9]:

                print('\nO primeiro dígito do CPF é verdadeiro!\n')

                os.system('pause')

                os.system('cls')

                break

            elif str(digito) != cpf[9]:

                print('\nO primeiro dígito do CPF é falso!\n')

                os.system('pause')

                os.system('cls')

            else:

                print('\nErro!!! Favor entrar em contato com o suporte!\n')

                os.system('pause')

                os.system('cls')

                break

        except ValueError:

            print('\nCPF inválido!\n')

            os.system('pause')

            os.system('cls')

    elif cpf == '' or cpf == ' ' or len(cpf) >= 12 or len(cpf) < 11:

        print('\nCPF inválido!\n')

        os.system('pause')

        os.system('cls')

    else:

        print('\nErro!!! Favor entrar em contato com o suporte!\n')

        os.system('pause')

        os.system('cls')

        break

=============================

'''

Calculo do segundo dígito do CPF

CPF: 746.824.890-70

Colete a soma dos 9 primeiros dígitos do CPF,

MAIS O PRIMEIRO DIGITO,

multiplicando cada um dos valores por uma

contagem regressiva começando de 11

Ex.:  746.824.890-70 (7468248907)

   11 10  9  8  7  6  5  4  3  2

\*  7   4  6  8  2  4  8  9  0  7 <-- PRIMEIRO DIGITO

   77 40 54 64 14 24 40 36  0 14

Somar todos os resultados:

77+40+54+64+14+24+40+36+0+14 = 363

Multiplicar o resultado anterior por 10

363 \* 10 = 3630

Obter o resto da divisão da conta anterior por 11

3630 % 11 = 0

Se o resultado anterior for maior que 9:

    resultado é 0

contrário disso:

    resultado é o valor da conta

O segundo dígito do CPF é 0

'''

import os

os.system('color 1f')

while True:

    cpf = input('\nDigite o CPF (somente números): ')

    os.system('cls')

    if len(cpf) == 11:

        try:

            cpf\_list = list(cpf)

            cpf\_list\_num = list(cpf\_list)

            cpf\_int = [int(i) for i in cpf\_list\_num]

            cpf\_soma\_1 = (cpf\_int[0] \* 10) + (cpf\_int[1] \* 9) + (cpf\_int[2] \* 8) + (cpf\_int[3] \* 7) + (cpf\_int[4] \* 6) + (cpf\_int[5] \* 5) + (cpf\_int[6] \* 4) + (cpf\_int[7] \* 3) + (cpf\_int[8] \* 2)

            cpf\_rest\_1 = (cpf\_soma\_1 \* 10) % 11

            if cpf\_rest\_1 > 9:

                calc\_primeiro\_dig = 0

            elif cpf\_rest\_1 <= 9 or cpf\_rest\_1 >= 0:

                calc\_primeiro\_dig = cpf\_rest\_1

            else:

                print('\nErro!!! Favor entrar em contato com o suporte!\n')

                os.system('pause')

                os.system('cls')

                break

            cpf\_soma\_2 = (cpf\_int[0] \* 11) + (cpf\_int[1] \* 10) + (cpf\_int[2] \* 9) + (cpf\_int[3] \* 8) + (cpf\_int[4] \* 7) + (cpf\_int[5] \* 6) + (cpf\_int[6] \* 5) + (cpf\_int[7] \* 4) + (cpf\_int[8] \* 3) + (cpf\_int[9] \* 2)

            cpf\_rest\_2 = (cpf\_soma\_2 \* 10) % 11

            if cpf\_rest\_2 > 9:

                calc\_segundo\_dig = 0

            elif cpf\_rest\_2 <= 9 or cpf\_rest\_2 >= 0:

                calc\_segundo\_dig = cpf\_rest\_2

            else:

                print('\nErro!!! Favor entrar em contato com o suporte!\n')

                os.system('pause')

                os.system('cls')

                break

            if calc\_primeiro\_dig == cpf\_int[9] and calc\_segundo\_dig == cpf\_int[10]:

                print('\nO CPF é verdadeiro!\n')

                os.system('pause')

                os.system('cls')

                break

            elif calc\_primeiro\_dig != cpf\_int[9] or calc\_segundo\_dig != cpf\_int[10]:

                print('\nO CPF é falso!\n')

                os.system('pause')

                os.system('cls')

            else:

                print('\nErro!!! Favor entrar em contato com o suporte!\n')

                os.system('pause')

                os.system('cls')

                break

        except ValueError:

            print('\nCPF inválido!\n')

            os.system('pause')

            os.system('cls')

    elif cpf == '' or cpf == ' ' or len(cpf) >= 12 or len(cpf) < 11:

        print('\nCPF inválido!\n')

        os.system('pause')

        os.system('cls')

    else:

        print('\nErro!!! Favor entrar em contato com o suporte!\n')

        os.system('pause')

        os.system('cls')

        break

=============================

'''

RESOLUÇÃO DO EXERCÍCIO DE OUTRA FORMA

Calculo do segundo dígito do CPF

CPF: 746.824.890-70

Colete a soma dos 9 primeiros dígitos do CPF,

MAIS O PRIMEIRO DIGITO,

multiplicando cada um dos valores por uma

contagem regressiva começando de 11

Ex.:  746.824.890-70 (7468248907)

   11 10  9  8  7  6  5  4  3  2

\*  7   4  6  8  2  4  8  9  0  7 <-- PRIMEIRO DIGITO

   77 40 54 64 14 24 40 36  0 14

Somar todos os resultados:

77+40+54+64+14+24+40+36+0+14 = 363

Multiplicar o resultado anterior por 10

363 \* 10 = 3630

Obter o resto da divisão da conta anterior por 11

3630 % 11 = 0

Se o resultado anterior for maior que 9:

    resultado é 0

contrário disso:

    resultado é o valor da conta

O segundo dígito do CPF é 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Outra forma para verificar se o CPF é válido:

cpf = 74682489070

cpf\_gerado\_calculo = {nove\_digitos}{digito\_1}{digito\_2}

if cpf == cpf\_gerado\_calculo:

    print(f'{cpf} válido')

elif cpf != cpf\_gerado\_calculo:

    print(f'{cpf} inválido')

else:

    print('CPF inválido')

'''

import os

os.system('color 1f')

while True:

    cpf = input('\nDigite o CPF (somente números): ')

    os.system('cls')

    if len(cpf) == 11:

        try:

            nove\_digitos = cpf[:9]

            contador\_regressivo\_1 = 10

            resultado\_1 = 0

            for digito in nove\_digitos:

                resultado\_1 += int(digito) \* contador\_regressivo\_1

                contador\_regressivo\_1 -= 1

            digito\_1 = (resultado\_1 \* 10) % 11

            digito\_1 = digito\_1 if digito\_1 <= 9 else 0

            dez\_digitos = cpf[:10]

            contador\_regressivo\_2 = 11

            resultado\_2 = 0

            for digito in dez\_digitos:

                resultado\_2 += int(digito) \* contador\_regressivo\_2

                contador\_regressivo\_2 -= 1

            digito\_2 = (resultado\_2 \* 10) % 11

            digito\_2 = digito\_2 if digito\_2 <= 9 else 0

            if str(digito\_1) == cpf[9] and str(digito\_2) == cpf[10]:

                print('\nO primeiro dígito do CPF é verdadeiro!\n')

                os.system('pause')

                os.system('cls')

                break

            elif str(digito\_1) != cpf[9] or str(digito\_2) != cpf[10]:

                print('\nO primeiro dígito do CPF é falso!\n')

                os.system('pause')

                os.system('cls')

            else:

                print('\nErro!!! Favor entrar em contato com o suporte!\n')

                os.system('pause')

                os.system('cls')

                break

        except ValueError:

            print('\nCPF inválido!\n')

            os.system('pause')

            os.system('cls')

    elif cpf == '' or cpf == ' ' or len(cpf) >= 12 or len(cpf) < 11:

        print('\nCPF inválido!\n')

        os.system('pause')

        os.system('cls')

    else:

        print('\nErro!!! Favor entrar em contato com o suporte!\n')

        os.system('pause')

        os.system('cls')

        break

=============================

'''

Possíveis problemas e soluções para o código de algoritmo do CPF.

.replace('o que quer substituir', 'para qual caracter quer substituir') - pode ser feito de forma encadeada.

'''

import os

os.system('color 1f')

while True:

    cpf = input('\nDigite o CPF (somente números): ')

    os.system('cls')

    corrigir\_cpf = cpf.replace('.', '').replace('-', '').replace(' ', '')

    if len(corrigir\_cpf) == 11:

        try:

            nove\_digitos = corrigir\_cpf[:9]

            contador\_regressivo\_1 = 10

            resultado\_1 = 0

            for digito in nove\_digitos:

                resultado\_1 += int(digito) \* contador\_regressivo\_1

                contador\_regressivo\_1 -= 1

            digito\_1 = (resultado\_1 \* 10) % 11

            digito\_1 = digito\_1 if digito\_1 <= 9 else 0

            dez\_digitos = corrigir\_cpf[:10]

            contador\_regressivo\_2 = 11

            resultado\_2 = 0

            for digito in dez\_digitos:

                resultado\_2 += int(digito) \* contador\_regressivo\_2

                contador\_regressivo\_2 -= 1

            digito\_2 = (resultado\_2 \* 10) % 11

            digito\_2 = digito\_2 if digito\_2 <= 9 else 0

            if str(digito\_1) == corrigir\_cpf[9] and str(digito\_2) == corrigir\_cpf[10]:

                print('\nO CPF é verdadeiro!\n')

                os.system('pause')

                os.system('cls')

                break

            elif str(digito\_1) != corrigir\_cpf[9] or str(digito\_2) != corrigir\_cpf[10]:

                print('\nO CPF é falso!\n')

                os.system('pause')

                os.system('cls')

            else:

                print('\nErro!!! Favor entrar em contato com o suporte!\n')

                os.system('pause')

                os.system('cls')

                break

        except ValueError:

            print('\nCPF inválido!\n')

            os.system('pause')

            os.system('cls')

    elif cpf == '' or cpf == ' ' or len(cpf) >= 12 or len(cpf) < 11:

        print('\nCPF inválido!\n')

        os.system('pause')

        os.system('cls')

    else:

        print('\nErro!!! Favor entrar em contato com o suporte!\n')

        os.system('pause')

        os.system('cls')

        break

=============================

while True:

    cpf\_2 = input('\nDigite o CPF: ')

    os.system('cls')

    corrigir\_cpf = re.sub(r'[^0-9]', '', cpf\_2)

    if len(corrigir\_cpf) == 11:

        try:

            nove\_digitos = corrigir\_cpf[:9]

            contador\_regressivo\_1 = 10

            resultado\_1 = 0

            for digito in nove\_digitos:

                resultado\_1 += int(digito) \* contador\_regressivo\_1

                contador\_regressivo\_1 -= 1

            digito\_1 = (resultado\_1 \* 10) % 11

            digito\_1 = digito\_1 if digito\_1 <= 9 else 0

            dez\_digitos = corrigir\_cpf[:10]

            contador\_regressivo\_2 = 11

            resultado\_2 = 0

            for digito in dez\_digitos:

                resultado\_2 += int(digito) \* contador\_regressivo\_2

                contador\_regressivo\_2 -= 1

            digito\_2 = (resultado\_2 \* 10) % 11

            digito\_2 = digito\_2 if digito\_2 <= 9 else 0

            if str(digito\_1) == corrigir\_cpf[9] and str(digito\_2) == corrigir\_cpf[10]:

                print('\nO CPF é verdadeiro!\n')

                os.system('pause')

                os.system('cls')

                break

            elif str(digito\_1) != corrigir\_cpf[9] or str(digito\_2) != corrigir\_cpf[10]:

                print('\nO CPF é falso!\n')

                os.system('pause')

                os.system('cls')

            else:

                print('\nErro!!! Favor entrar em contato com o suporte!\n')

                os.system('pause')

                os.system('cls')

                break

        except ValueError:

            print('\nCPF inválido!\n')

            os.system('pause')

            os.system('cls')

    elif cpf\_2 == '' or cpf\_2 == ' ' or len(cpf\_2) >= 12 or len(cpf\_2) < 11:

        print('\nCPF inválido!\n')

        os.system('pause')

        os.system('cls')

    else:

        print('\nErro!!! Favor entrar em contato com o suporte!\n')

        os.system('pause')

        os.system('cls')

        break

=============================

while True:

    cpf\_3 = input('\nDigite o CPF: ')

    os.system('cls')

    corrigir\_cpf = re.sub(r'[^0-9]', '', cpf\_3)

    prim\_char\_ent\_cpf\_rep = corrigir\_cpf == corrigir\_cpf[0] \* len(corrigir\_cpf)

    if len(corrigir\_cpf) == 11 and not prim\_char\_ent\_cpf\_rep:

        try:

            nove\_digitos = corrigir\_cpf[:9]

            contador\_regressivo\_1 = 10

            resultado\_1 = 0

            for digito in nove\_digitos:

                resultado\_1 += int(digito) \* contador\_regressivo\_1

                contador\_regressivo\_1 -= 1

            digito\_1 = (resultado\_1 \* 10) % 11

            digito\_1 = digito\_1 if digito\_1 <= 9 else 0

            dez\_digitos = corrigir\_cpf[:10]

            contador\_regressivo\_2 = 11

            resultado\_2 = 0

            for digito in dez\_digitos:

                resultado\_2 += int(digito) \* contador\_regressivo\_2

                contador\_regressivo\_2 -= 1

            digito\_2 = (resultado\_2 \* 10) % 11

            digito\_2 = digito\_2 if digito\_2 <= 9 else 0

            if str(digito\_1) == corrigir\_cpf[9] and str(digito\_2) == corrigir\_cpf[10]:

                print('\nO CPF é verdadeiro!\n')

                os.system('pause')

                os.system('cls')

                break

            elif str(digito\_1) != corrigir\_cpf[9] or str(digito\_2) != corrigir\_cpf[10]:

                print('\nO CPF é falso!\n')

                os.system('pause')

                os.system('cls')

            else:

                print('\nErro!!! Favor entrar em contato com o suporte!\n')

                os.system('pause')

                os.system('cls')

                break

        except ValueError:

            print('\nCPF inválido!\n')

            os.system('pause')

            os.system('cls')

    elif cpf\_3 == '' or cpf\_3 == ' ' or len(cpf\_3) >= 12 or len(cpf\_3) < 11 or prim\_char\_ent\_cpf\_rep:

        print('\nCPF inválido!\n')

        os.system('pause')

        os.system('cls')

    else:

        print('\nErro!!! Favor entrar em contato com o suporte!\n')

        os.system('pause')

        os.system('cls')

        break

os.system('pause')

os.system('cls')

=============================

'''

Gerador de CPFs

'''

import os, re, random

os.system('color 1f')

qtd\_cpf = int(input('\nQuantidade de CPFs que deseja gerar: '))

os.system('cls')

print('\n')

for a in range(qtd\_cpf):

    while True:

        num\_nove\_cpf = ''

        cpf\_form = ''

        for i in range(9):

            num\_nove\_cpf += str(random.randint(0, 9))

        corrigir\_cpf = re.sub(r'[^0-9]', '', num\_nove\_cpf)

        prim\_char\_ent\_cpf\_rep = corrigir\_cpf == corrigir\_cpf[0] \* len(cpf\_form)

        try:

            nove\_digitos = corrigir\_cpf[:9]

            contador\_regressivo\_1 = 10

            resultado\_1 = 0

            for digito in nove\_digitos:

                resultado\_1 += int(digito) \* contador\_regressivo\_1

                contador\_regressivo\_1 -= 1

            digito\_1 = (resultado\_1 \* 10) % 11

            digito\_1 = digito\_1 if digito\_1 <= 9 else 0

            dez\_digitos = corrigir\_cpf[:10]

            contador\_regressivo\_2 = 11

            resultado\_2 = 0

            for digito in dez\_digitos:

                resultado\_2 += int(digito) \* contador\_regressivo\_2

                contador\_regressivo\_2 -= 1

            digito\_2 = (resultado\_2 \* 10) % 11

            digito\_2 = digito\_2 if digito\_2 <= 9 else 0

            cpf\_form = num\_nove\_cpf + str(digito\_1) + str(digito\_2)

            if str(digito\_1) == cpf\_form[9] and str(digito\_2) == cpf\_form[10]:

                print(f'O CPF {cpf\_form} é verdadeiro!')

                break

            elif str(digito\_1) != cpf\_form[9] or str(digito\_2)!= cpf\_form[10]:

                print(f'O CPF {cpf\_form} é falso!')

                break

            else:

                print('Erro!!! Favor entrar em contato com o suporte!')

                break

        except ValueError:

            print('CPF inválido!')

            os.system('pause')

            os.system('cls')

print('\n')

os.system('pause')

os.system('cls')

=============================

'''

Introdução ás funções (def) em Python

Funções são trechos de código usados para replicar determinada ação ao longo do seu código.

Elas podem receber valores para parâmetros (argumentos) e retornar um valor específico.

Por padrão, funções Python retornam None (nada).

'''

import os, gc

os.system('color 1f')

def imprimir (a, b, c):

    print(a, b, c)

print('\n')

imprimir(1, 2, 3)

imprimir(4, 5, 6)

print('\n')

os.system('pause')

os.system('cls')

gc.collect()

def saudacao(nome = 'Sem nome'):

    print(f'Olá, {nome}')

print('\n')

saudacao('João Augusto')

saudacao()

print('\n')

os.system('pause')

os.system('cls')

gc.collect()

=============================

'''

Argumentos nomeados e não nomeados em funções Pyhton

Argumento nomeado tem nome com sinal de igual

Argumento não nomeado recebe apenas o argumento (valor)

Parâmetro - É a váriavel do código, que armazena as informações repassadas.

Argumento - É a informação repassada para a váriavel.

'''

###############################

import os, gc

os.system('color 1f')

def soma(x, y):

    print(f'{x = } e {y = } | x + y = ', x + y)

print('\n')

soma(1, 2)

soma(y = 2, x = 1)

print('\n')

os.system('pause')

os.system('cls')

gc.collect()

###############################

def subtracao(x, y, z):

    print(f'{x = }, {y = } e {z = } | x - y - z = ', x - y - z)

print('\n')

subtracao(1, 2 , 3)

subtracao(y = 2, z = 3, x = 1)

print('\n')

os.system('pause')

os.system('cls')

gc.collect()

###############################

=============================

'''

Valores padrão para parâmetros

Ao definir uma função, os parâmetros podem ter valores padrão.

Caso o valor não seja enviado para o parâmetro, o valor padrão será usado.

Refatorar - Editar o seu código.

'''

import os

os.system('color 1f')

def soma(x, y, z = None):

if z is not None:

print(f'\n{x = } + {y = } + {z = } = ', x + y + z)

elif z is None:

print(f'\n{x = } + {y = } = ', x + y)

else:

print('\nERRO! Favor contatar suporte!')

soma(1, 2)

soma(3, 5)

soma(100, 200)

soma(7, 9, 0)

print('\n')

os.system('pause')

os.system('cls')

=============================

'''

Escopo de funções em Python

Escopo significa o local onde aquele código pode atingir.

Existe o escopo global e local.

O escopo global é onde todo o código é alcançavel.

O escopo local é onde apenas nomes do mesmo local podem ser alcançados.

'''

import os

os.system('color 1f')

def escopo():

x = 1

print(x)

print('\n')

escopo()

print('\n')

os.system('pause')

os.system('cls')

##################

x = 1

def escopo\_2():

x = 10

def outra\_funcao():

x = 11

y = 2

print(x, y)

outra\_funcao()

print(x)

print('\n')

print(x)

escopo\_2()

print(x)

print('\n')

os.system('pause')

os.system('cls')

=============================

'''

Retorno de valores das funções (return).

'''

import os

os.system('color 1f')

def soma(x, y):

if x > 10:

return '\nNão foi possível somar, pois "X" é maior que 10.\n'

if y > 10:

return '\nNão foi possível somar, pois "Y" é maior que 10.\n'

else:

return f'\n{x + y}\n'

soma1 = soma(2, 2)

soma2 = soma(11, 3)

soma3 = soma(4, 11)

print(soma1)

print(soma2)

print(soma3)

os.system('pause')

os.system('cls')

=============================

'''

args - Argumentos não nomeados

\* - \*args (empacotamento e desempacotamento)

'''

import os

os.system('color 1f')

# Lembre-te de desempacotamento

x, y, \*resto = 1, 2, 3, 4

print(f'\n{x, y, resto}\n')

def soma\_1(x, y):

return x + y

os.system('pause')

os.system('cls')

###################

def soma\_2(\*args):

args = list(args)

print(args, type(args))

print('\n')

soma\_2(1, 2, 3, 4, 5)

print('\n')

os.system('pause')

os.system('cls')

###################

def soma\_3(\*args):

total = 0

for numero in args:

print(f'{total} + {numero}')

total += numero

print(f'Total: {total}')

print('\n')

print('\n')

soma\_3(1, 2, 3, 4, 5)

os.system('pause')

os.system('cls')

=============================

'''

'''

import os

os.system('color 1f')

# Empacotamento do que é enviado para função dentro de uma tupla.

def soma(\*args):

total = 0

for numero in args:

total += numero

return total

soma\_12345 = soma(1, 2, 3, 4, 5)

print(f'\nSoma de 1, 2, 3, 4 e 5 fica {soma\_12345}.\n')

os.system('pause')

os.system('cls')

#########################

# Desempacotamento uma tupla para função.

numeros = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

outra\_soma = soma(\*numeros)

print(f'\n{outra\_soma}\n')

print(f'{sum(numeros)}\n')

os.system('pause')

os.system('cls')

=============================

'''

Exercícios com função

Crie uma função que multiplica todos os argumentos não nomeados recebidos.

Retorne o total para uma variável e mostre o valor da variável.

Crie uma função fala se um número é par ou ímpar.

Retorne se o número é par ou ímpar.

'''

import os

os.system('color 1f')

def multi(\*args):

total = 1

for numero in args:

total \*= numero

return total

def par\_impar(multi\_dados):

if multi\_dados % 2 == 0:

return '"Par"'

elif multi\_dados % 2 != 0:

return '"Impar"'

else:

return '(Erro ao processar!)'

multi\_dados = multi(1, 2, 3, 4, 5)

par\_ou\_impar = par\_impar(multi\_dados)

print(f'\nA multiplicação de 1 X 2 X 3 X 4 X 5 = {multi\_dados}, sendo {par\_ou\_impar}.\n')

os.system('pause')

os.system('cls')

=============================

'''

Exercícios com função - RESOLUÇÂO

Crie uma função que multiplica todos os argumentos não nomeados recebidos.

Retorne o total para uma variável e mostre o valor da variável.

Crie uma função fala se um número é par ou ímpar.

Retorne se o número é par ou ímpar.

'''

import os

os.system('color 1f')

def multiplicar(\*args):

total = 1

for numero in args:

total \*= numero

return total

multiplicacao = multiplicar(1, 2, 3, 4, 5)

print(f'\n{multiplicacao}\n')

os.system('pause')

os.system('cls')

######################

def par\_impar(numero):

multiplo\_de\_dois = numero % 2 == 0

if multiplo\_de\_dois:

return f'{numero} é par'

return f'{numero} é impar'

print(f'\n{par\_impar(2)}\n')

print(f'{par\_impar(3)}\n')

print(f'{par\_impar(15)}\n')

print(f'{par\_impar(16)}\n')

os.system('pause')

os.system('cls')

=============================

'''

Higher Order Functions

Funções de primeira classe

'''

import os

os.system('color 1f')

def saudacao(msg, nome):

return f'{msg}, {nome}!'

# O \*args está recebendo o restante dos dados e compactando

def executa(funcao, \*args):

# Aqui o \*args está descompactando os dados resebidos

return funcao(\*args)

print(f'\n{executa(saudacao, 'Bom dia', 'João')}\n')

os.system('pause')

os.system('cls')

=============================

Termos técnicos: Higher Order Functions e First-Class Functions

Academicamente, os termos Higher Order Functions e First-Class Functions têm significados diferentes.

Higher Order Functions - Funções que podem receber e/ou retornar outras funções

First-Class Functions - Funções que são tratadas como outros tipos de dados comuns (strings, inteiros, etc...)

Não faria muita diferença no seu código, mas penso que deveria lhe informar isso.

Observação: esses termos podem ser diferentes e ainda refletir o mesmo significado.

=============================

'''

Closure e funções que retornam outras funções

'''

import os

os.system('color 1f')

def criar\_saudacao(saudacao):

def saudar(nome):

return f'{saudacao}, {nome}!'

return saudar

falar\_bom\_dia = criar\_saudacao('Bom dia')

falar\_boa\_tarde = criar\_saudacao('Boa tarde')

falar\_boa\_noite = criar\_saudacao('Boa noite')

print('\n')

for nome in ['Maria', 'João', 'Davi']:

print('---')

print(falar\_bom\_dia(nome))

print(falar\_boa\_tarde(nome))

print(falar\_boa\_noite(nome))

print('---')

print('\n')

os.system('pause')

os.system('cls')

=============================

'''

EXERCÍCIOS

Crie funções que duplicam, triplicam e quadrupicam o número recebido como parâmetro.

'''

import os

os.system('color 1f')

def multiplica\_numero(multiplicador):

def numero\_repassado(numero):

return multiplicador \* numero

return numero\_repassado

numero = int(input('\nNúmero: '))

os.system('cls')

duplicar = multiplica\_numero(2)

triplicar = multiplica\_numero(3)

quadruplicar = multiplica\_numero(4)

print(f'\nO dobro de {numero} é {duplicar(numero)}')

print(f'O triplo de {numero} é {triplicar(numero)}')

print(f'O quadruplo de {numero} é {quadruplicar(numero)}\n')

os.system('pause')

os.system('cls')

=============================

'''

EXERCÍCIOS

Crie funções que duplicam, triplicam e quadrupicam o número recebido como parâmetro.

'''

import os

os.system('color 1f')

def duplicar(numero):

return numero \* 2

def triplicar(numero):

return numero \* 3

def quadruplicar(numero):

return numero \* 4

print('\n')

print(duplicar(2))

print(triplicar(2))

print(quadruplicar(2))

print('\n')

os.system('pause')

os.system('cls')

############################

def criar\_multiplicador(multiplicador):

def multiplicar(numero):

return numero \* multiplicador

return multiplicar

duplicar\_2 = criar\_multiplicador(2)

triplicar\_3 = criar\_multiplicador(3)

quadruplicar\_4 = criar\_multiplicador(4)

print('\n')

print(duplicar\_2(2))

print(triplicar\_3(2))

print(quadruplicar\_4(2))

print('\n')

os.system('pause')

os.system('cls')

=============================

'''

Manipulando chaves e valores em dicionários.

'''

import os

os.system('color 1f')

pessoa = {

'nome': 'João Augusto',

'sobrenome': 'Colasso Handocha',

'idade': 27,

'altura': 1.80,

'endereços': [

{'rua': 'tal tal', 'número': 123},

{'rua': 'outra rua', 'número': 345},

],

}

print(f'\nNome e Sobrenome: {pessoa['nome']} {pessoa['sobrenome']}')

print(f'Idade: {pessoa['idade']}')

print(f'Altura: {pessoa['altura']}')

print(f'Endereços: {pessoa['endereços']}\n')

os.system('pause')

os.system('cls')

##############################

dados\_pessoa = {}

dados\_pessoa['nome'] = input('\nQual seu nome: ')

os.system('cls')

dados\_pessoa['idade'] = int(input('\nQuantos anos você tem (Insira somente números): '))

os.system('cls')

dados\_pessoa['altura'] = float(input('\nQual sua altura: '))

os.system('cls')

dados\_pessoa['peso'] = float(input('\nQual seu peso: '))

os.system('cls')

print(f'\nInformações salvas no banco de dados: {dados\_pessoa}\n')

os.system('pause')

os.system('cls')

print(f"\n{dados\_pessoa['nome']}, tem {dados\_pessoa['idade']} anos, {dados\_pessoa['altura']:.2f} metros de altura, pesando {dados\_pessoa['peso']:.3f}Kg.\n")

os.system('pause')

os.system('cls')

##############################

dados = {}

dados['nome'] = 'João'

dados['sobrenome'] = 'Augusto'

print(f'\n{dados}\n')

del dados['sobrenome']

print(f'{dados}\n')

if dados.get('sobrenome') is None:

print('Não existe\n')

else:

print(f'{dados['sobrenome']}\n')

os.system('pause')

os.system('cls')

=============================

'''

Métodos úteis dos dicionários em Python

len - quantas chaves

keys - iterável com as chaves

values - iteráveç com os valores

items - iterável com chaves e valores

setdefault - adiciona valor se a chave não existe

copy - retorna uma cópia rasa (shallow copy)

get - obtém uma chave

pop - apaga um item com a chave especificada (del)

popitem - apaga o último item adicionado

update - atualiza um dicionário com outro

'''

import os, copy

os.system('color 1f')

pessoa = {

'nome': 'João Augusto',

'idade': 27,

}

print('\n')

# Quantidade de chaves

print(len(pessoa))

# Lista os nomes das chaves (Pode ser transformada como list ou tuple) (Pode utilizar também o for de duas formas para conseguiur o nome das chaves)

print(pessoa.keys())

print(tuple(pessoa.keys()))

print(list(pessoa.keys()))

print('\n')

for chave in pessoa.keys():

print(chave)

print('\n')

for chave in pessoa:

print(chave)

print('\n')

# Lista os valores(resultados) das chaves (Pode ser transformada como list ou tuple) (Pode utilizar também o for para conseguiur os dados das chaves)

print(pessoa.values())

print(list(pessoa.values()))

print(tuple(pessoa.values()))

print('\n')

for valor in pessoa.values():

print(valor)

print('\n')

# Lista as chaves e os valores (Pode ser transformada como list ou tuple) (Pode utilizar também o for para conseguiur as chaves com os valores)

print(pessoa.items())

print(list(pessoa.items()))

print(tuple(pessoa.items()))

print('\n')

for chave, valor in pessoa.items():

print(f'{chave}: {valor}')

print('\n')

# Repassa a informação de uma chave não criada onde utilizará outro valor a partir do momento que é criada

pessoa.setdefault('peso', 'Erro na informação peso')

print(pessoa['peso'])

# Realiza uma cópia rasa do dicionário, onde qualquer mudança nessa cópia (caso não seja feito o .copy()), afeta também o dicionários original (Como no exemplo, o d2 está apontando que é o mesmo dicionário do d1) (Para valores mutáveis, como exemplo uma lista, ele não realiza a cópia, e sim fiquem sinconizados entre eles)

# Para realizar uma cópia mais profunda, visando os valores mutáveis não serem sincronizados, insira import copy, e posteriormente insira o comando como copy.deepcopy()

d1 = {

'c1': 1,

'c2': 2,

'l1': [0, 1, 2],

}

d2 = d1

d2['c1'] = 1000

print(d1)

d2 = d1.copy()

d2['c1'] = 100

d2['l1'][1] = 999

print(d1)

print(d2)

d2['l1'][1] = 1

print('-------------')

d2 = copy.deepcopy(d1)

d2['c1'] = 100

d2['l1'][1] = 999

print(d1)

print(d2)

# Retorna um valor None ou o valor da chave, sendo possível repassar um valor padrão

p1 = {

'nome': 'João',

'sobrenome': 'Augusto',

}

print(p1.get('nome', 'Nome não foi encontrado'))

print(p1.get('sobrenome', 'Sobrenome não encontrado'))

# Apaga um item de uma chave específica no dicionário

nome = p1.pop('nome')

print(nome)

print(p1)

p1['nome'] = 'João'

# Apaga a última chave adcionada

ultima\_chave = p1.popitem()

print(ultima\_chave)

print(p1)

p1['sobrenome'] = 'Augusto'

# Modifica e atualiza o dicionário (Pode ser programado de 4 formas)

p1.update({

'nome': 'JOAO',

'sobrenome': 'AUGUSTO',

})

################

p1.update(idade = 27)

################

tupla = (('peso', 97), ('altura', 1.80))

p1.update(tupla)

################

lista = [['cep', 11111111], ['genero', 'MASCULINO']]

p1.update(lista)

################

print(p1)

print('\n')

os.system('pause')

os.system('cls')

=============================

'''

EXERCíCIO

Sistema de perguntas e respostas

'''

import os

os.system('color 1f')

# Declaração da função do menu do sistema

def main(opcao\_menu):

# Apresentação das opções do menu

print('\n' + '-' \* 20)

print('[C] - Começar o Jogo')

print('[S] - Sair')

print('-' \* 20 + '\n')

# Solicitação de escolha de uma das opções solicitadas anteriormente

opcao\_menu = input('Escolha sua opção: ')

os.system('cls')

# Caso seja repassado uma opção vazia, solicita que seja inserido a opção desejada, retornando novamente ao menu inicial

if opcao\_menu == '' or opcao\_menu == ' ':

print('\nFavor inserir a opção desejada!!!\n')

os.system('pause')

os.system('cls')

main(opcao\_menu)

# Caso a opção "C" seja escolhida, é executado o jogo

elif opcao\_menu.upper() == 'C':

jogo\_perguntas(perguntas = '')

# Caso a opção "S" seja escolhida, é encerrado o programa

elif opcao\_menu.upper() == 'S':

print('\nSaindo...\n')

os.system('pause')

os.system('cls')

os.system('exit')

# Caso seja repassado outra opção que não seja "C" ou "S", é solicitado digitar a opção correta, sendo retornado ao menu

elif opcao\_menu.upper() != 'C' or opcao\_menu.upper() != 'S':

print('\nFavor digitar uma das opções repassadas!!!\n')

os.system('pause')

os.system('cls')

main(opcao\_menu)

# Caso apresente qualquer informação incorreta, repassa mensagem de erro e retorna ao menu

else:

print('\nErro no processamento da aplicação!!!\n')

os.system('pause')

os.system('cls')

main(opcao\_menu)

# Declaração da função do jogo

def jogo\_perguntas(perguntas):

# Lista dos dicionários das perguntas do jogo

perguntas = [

{

'pergunta': 'Quanto é 2 + 2?',

'opcoes': ['1', '2', '3', '4'],

'resposta': '4',

},

{

'pergunta': 'Quanto é 5 X 5?',

'opcoes': ['25', '55', '10', '51'],

'resposta': '25',

},

{

'pergunta': 'Quanto é 10 / 2?',

'opcoes': ['4', '5', '2', '1'],

'resposta':'5',

},

]

# Número de acertos e erros

acertos = 0

erros = 0

# Pergunta 1

print(f'\n[1] - {perguntas[0]['pergunta']}\n')

alternativas = 1

for opcao in perguntas[0]['opcoes']:

print(f'{alternativas}) {opcao}')

alternativas += 1

resposta\_1 = input('\nQual é a resposta: ')

os.system('cls')

if resposta\_1 == '4':

print(f'\nParabéns!!! Resposta certa é {perguntas[0]['resposta']}\n')

acertos += 1

os.system('pause')

os.system('cls')

elif resposta\_1 != '4':

print(f'\nErrou!!! Que pena! Resposta certa é {perguntas[0]['resposta']}\n')

erros += 1

os.system('pause')

os.system('cls')

else:

print('\nErro no processamento da aplicação!!!\n')

os.system('pause')

os.system('cls')

main(opcao\_menu = '')

# Pergunta 2

print(f'\n[2] - {perguntas[1]['pergunta']}\n')

alternativas = 1

for opcao in perguntas[1]['opcoes']:

print(f'{alternativas}) {opcao}')

alternativas += 1

resposta\_2 = input('\nQual é a resposta: ')

os.system('cls')

if resposta\_2 == '1':

print(f'\nParabéns!!! Resposta certa é {perguntas[1]['resposta']}\n')

acertos += 1

os.system('pause')

os.system('cls')

elif resposta\_2 != '1':

print(f'\nErrou!!! Que pena! Resposta certa é {perguntas[1]['resposta']}\n')

erros += 1

os.system('pause')

os.system('cls')

else:

print('\nErro no processamento da aplicação!!!\n')

os.system('pause')

os.system('cls')

main(opcao\_menu = '')

# Pergunta 3

print(f'\n[3] - {perguntas[2]['pergunta']}\n')

alternativas = 1

for opcao in perguntas[2]['opcoes']:

print(f'{alternativas}) {opcao}')

alternativas += 1

resposta\_3 = input('\nQual é a resposta: ')

os.system('cls')

if resposta\_3 == '2':

print(f'\nParabéns!!! Resposta certa é {perguntas[2]['resposta']}\n')

acertos += 1

os.system('pause')

os.system('cls')

elif resposta\_3 != '2':

print(f'\nErrou!!! Que pena! Resposta certa é {perguntas[2]['resposta']}\n')

erros += 1

os.system('pause')

os.system('cls')

else:

print('\nErro no processamento da aplicação!!!\n')

os.system('pause')

os.system('cls')

main(opcao\_menu = '')

# Resultado

print('\nQuantidade de perguntas: 3')

print(f'Acertos: {acertos}')

print(f'Erros:{erros}\n')

os.system('pause')

os.system('cls')

main(opcao\_menu = '')

# Mensagem de boas vindas

print('\n' + '=' \* 31 + '\nBem vindo ao jogo de perguntas!\n' + '=' \* 31 + '\n')

os.system('pause')

os.system('cls')

# Acesso a declaração da função do menu

main(opcao\_menu = '')

=============================

'''

EXERCíCIO

Sistema de perguntas e respostas

'''

import os

os.system('color 1f')

perguntas = [

{

'pergunta': 'Quanto é 2 + 2?',

'opcoes': ['1', '2', '3', '4'],

'resposta': '4',

},

{

'pergunta': 'Quanto é 5 X 5?',

'opcoes': ['25', '55', '10', '51'],

'resposta': '25',

},

{

'pergunta': 'Quanto é 10 / 2?',

'opcoes': ['4', '5', '2', '1'],

'resposta':'5',

},

]

qtd\_acertos = 0

for pergunta in perguntas:

print('Pergunta:', pergunta['pergunta'])

print('\n')

opcoes = pergunta['opcoes']

for i, opcao in enumerate(opcoes):

print(f'{i})', opcao)

print('\n')

escolha = input('Escolha uma opção: ')

acertou = False

escolha\_int = None

qtd\_opcoes = len(opcoes)

if escolha.isdigit():

escolha\_int = int(escolha)

if escolha\_int is not None:

if escolha\_int >= 0 and escolha\_int < qtd\_opcoes:

if opcoes[escolha\_int] == pergunta['resposta']:

acertou = True

print('\n')

if acertou:

qtd\_acertos += 1

print('Acertou')

else:

print('Errou')

print('\n')

print('\nVocê acertou', qtd\_acertos)

print('de', len(perguntas), 'perguntas.\n')

os.system('pause')

os.system('cls')

=============================

'''

Sets - Conjuntos em Python (tipo set)

Conjuntos são ensinados na matemática

https://brasilescola.uol.com.br/matematica/conjunto.htm

Representados graficamente pelo diagrama de Venn

Sets em Python são mutáveis, porém aceitam apenas tipos imutáveis como valor interno.

Criando um set

set(interável) ou {1, 2, 3}

Sets são eficientes para remover valores duplicados de iteráveis.

- Não aceitam valores mutáveis;

- Seus valores serão sempre únicos;

- Não tem índexes;

- Não garantem ordem;

- São iteráveis (for, in, not in)

Métodos úteis: add, update, clear, discard

Operadores úteis:

união | união (union) - Une

intersecção & (intersection) - Itens presentes em ambos

diferença - Itens presentes apenas no set da esquerda

diferença simétrica ^ - Itens que não estão em ambos

'''

import os

os.system('color 1f')

s1 = set() # set vazio

s2 = set('Teste') # com dados

s3 = {'João', 1, 2, 3} # com dados em outro formato

print(f'\n{s1}')

print(f'{s2}')

print(f'{s3}\n')

os.system('pause')

os.system('cls')

=============================

'''

Sets - Conjuntos em Python (tipo set)

Conjuntos são ensinados na matemática

https://brasilescola.uol.com.br/matematica/conjunto.htm

Representados graficamente pelo diagrama de Venn

Sets em Python são mutáveis, porém aceitam apenas tipos imutáveis como valor interno.

Criando um set

set(interável) ou {1, 2, 3}

Sets são eficientes para remover valores duplicados de iteráveis.

- Não aceitam valores mutáveis;

- Seus valores serão sempre únicos;

- Não tem índexes;

- Não garantem ordem;

- São iteráveis (for, in, not in)

Métodos úteis: add, update, clear, discard

Operadores úteis:

união | união (union) - Une

intersecção & (intersection) - Itens presentes em ambos

diferença - Itens presentes apenas no set da esquerda

diferença simétrica ^ - Itens que não estão em ambos

'''

import os

os.system('color 1f')

s1 = {1, 2, 3, 3, 3, 1} # set consegue eliminar valores duplicados (Ex: 1, 2, 3)

print(f'\n{s1}\n')

# Forma longa de eliminar valores duplicados em uma lista

l1 = [1, 2, 3, 3, 3, 1]

s2 = set(l1)

l2 = list(s1)

print(f'\n{l2}\n')

os.system('pause')

os.system('cls')

=============================

'''

Sets - Conjuntos em Python (tipo set)

Conjuntos são ensinados na matemática

https://brasilescola.uol.com.br/matematica/conjunto.htm

Representados graficamente pelo diagrama de Venn

Sets em Python são mutáveis, porém aceitam apenas tipos imutáveis como valor interno.

Criando um set

set(interável) ou {1, 2, 3}

Sets são eficientes para remover valores duplicados de iteráveis.

- Não aceitam valores mutáveis;

- Seus valores serão sempre únicos;

- Não tem índexes;

- Não garantem ordem;

- São iteráveis (for, in, not in)

Métodos úteis: add, update, clear, discard

Operadores úteis:

união | união (union) - Une

intersecção & (intersection) - Itens presentes em ambos

diferença - Itens presentes apenas no set da esquerda

diferença simétrica ^ - Itens que não estão em ambos

'''

import os

os.system('color 1f')

s1 = set()

s1.add('João') # Adiciona um valor no set (Não aceita mais de um valor para adicionar)

s1.update('Olá mundo') # Muito parecido com o add, mas deforma valores interados

s1.update(('Augusto', 1, 2, 3)) # Para não deformar e inserir mais informações, pode ser utilizado uma tupla

print(f'\n{s1}\n')

s1.discard('Augusto') # Descarta um valor expecífico que está no set

print(f'{s1}\n')

s1.clear() # Realiza a limpa do set

print(f'{s1}\n')

os.system('pause')

os.system('cls')

=============================

'''

Sets - Conjuntos em Python (tipo set)

Conjuntos são ensinados na matemática

https://brasilescola.uol.com.br/matematica/conjunto.htm

Representados graficamente pelo diagrama de Venn

Sets em Python são mutáveis, porém aceitam apenas tipos imutáveis como valor interno.

Criando um set

set(interável) ou {1, 2, 3}

Sets são eficientes para remover valores duplicados de iteráveis.

- Não aceitam valores mutáveis;

- Seus valores serão sempre únicos;

- Não tem índexes;

- Não garantem ordem;

- São iteráveis (for, in, not in)

Métodos úteis: add, update, clear, discard

Operadores úteis:

união | união (union) - Une

intersecção & (intersection) - Itens presentes em ambos

diferença - Itens presentes apenas no set da esquerda

diferença simétrica ^ - Itens que não estão em ambos

'''

import os

os.system('color 1f')

s1 = {1, 2, 3}

s2 = {2, 3, 4}

s3 = s1 | s2 # Para unir dois sets

s4 = s1 & s2 # Para retornar os itens presentes nos dois sets

s5 = s1 - s2 # Para retornar os itens que estão apenas na esquerda

s6 = s1 ^ s2 # Para retornar os itens que não estão em ambos os sets

print(f'\n{s3} - {s4} - {s5} - {s6}\n')

os.system('pause')

os.system('cls')

=============================

'''

Exemplo de uso dos sets

'''

import os

os.system('color 1f')

letras = set()

while True:

letra = input('Digite: ')

letras.add(letra.lower())

print(f'\n{letras}\n')

if 'l' in letras:

print('Parabéns!\n')

break

os.system('pause')

os.system('cls')

=============================

'''

EXERCÍCIO

Crie uma função que encontra o primeiro duplicado considerando o segundo número como a duplicação.

Retorne a duplicação considerada.

Requisitos:

A ordem do número duplicado é considerada a partir da segunda ocorrência do número, ou seja, o número duplicado em si.

Exemplo:

[1, 2, 3, ->3<-, 2, 1] -> 1, 2 e 3 são duplicados (retorne 3)

[1, 2, 3, 4, 5, 6] -> Retorne -1 (não tem duplicados)

[1, 4, 9, 8, ->9<-, 4, 8] (retorne 9)

Se não encontrar duplicados na lista, retorne -1

'''

import os

os.system('color 1f')

def primeiro\_duplicado(lista):

ja\_visto = set()

for num in lista:

if num in ja\_visto:

return num

ja\_visto.add(num)

return -1

lista\_de\_listas\_de\_inteiros = [

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10],

[9, 1, 8, 9, 9, 7, 2, 1, 6, 8],

[1, 3, 2, 2, 8, 6, 5, 9, 6, 7],

[3, 8, 2, 8, 6, 7, 7, 3, 1, 9],

[4, 8, 8, 8, 5, 1, 10, 3, 1, 7],

[1, 3, 7, 2, 2, 1, 5, 1, 9, 9],

[10, 2, 2, 1, 3, 5, 10, 5, 10, 1],

[1, 6, 1, 5, 1, 1, 1, 4, 7, 3],

[1, 3, 7, 1, 10, 5, 9, 2, 5, 7],

[4, 7, 6, 5, 2, 9, 2, 1, 2, 1],

[5, 3, 1, 8, 5, 7, 1, 8, 8, 7],

[10, 9, 8, 7, 6, 5, 4, 3, 2, 1],

]

for idx, lista in enumerate(lista\_de\_listas\_de\_inteiros):

resultado = primeiro\_duplicado(lista)

print(f'Lista {idx + 1}: Primeiro duplicado é {resultado}')

os.system('pause')

os.system('cls')

=============================

'''

SOLUÇÂO

Crie uma função que encontra o primeiro duplicado considerando o segundo número como a duplicação.

Retorne a duplicação considerada.

Requisitos:

A ordem do número duplicado é considerada a partir da segunda ocorrência do número, ou seja, o número duplicado em si.

Exemplo:

[1, 2, 3, ->3<-, 2, 1] -> 1, 2 e 3 são duplicados (retorne 3)

[1, 2, 3, 4, 5, 6] -> Retorne -1 (não tem duplicados)

[1, 4, 9, 8, ->9<-, 4, 8] (retorne 9)

Se não encontrar duplicados na lista, retorne -1

'''

import os

os.system('color 1f')

lista\_de\_listas\_de\_inteiros = [

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10],

[9, 1, 8, 9, 9, 7, 2, 1, 6, 8],

[1, 3, 2, 2, 8, 6, 5, 9, 6, 7],

[3, 8, 2, 8, 6, 7, 7, 3, 1, 9],

[4, 8, 8, 8, 5, 1, 10, 3, 1, 7],

[1, 3, 7, 2, 2, 1, 5, 1, 9, 9],

[10, 2, 2, 1, 3, 5, 10, 5, 10, 1],

[1, 6, 1, 5, 1, 1, 1, 4, 7, 3],

[1, 3, 7, 1, 10, 5, 9, 2, 5, 7],

[4, 7, 6, 5, 2, 9, 2, 1, 2, 1],

[5, 3, 1, 8, 5, 7, 1, 8, 8, 7],

[10, 9, 8, 7, 6, 5, 4, 3, 2, 1],

]

def encontra\_primeiro\_duplicado(lista\_de\_inteiros):

numero\_checados = set()

primeiro\_duplicado = -1

for numero in lista\_de\_inteiros:

if numero in numero\_checados:

primeiro\_duplicado = numero

break

numero\_checados.add(numero)

return primeiro\_duplicado

print('\n')

for lista in lista\_de\_listas\_de\_inteiros:

print(f'{lista}: {encontra\_primeiro\_duplicado(lista)}')

print('\n')

os.system('pause')

os.system('cls')

=============================

=============================

=============================