Aluno: Pedro dos Santos Bittencourt Cyrne

Curso: Analise desenvolvimento de sistemas Turno: Manhã

Feito no PYCHARM

PDF do codigo requisitado:

**COMENTADO:**

import re #Importa a biblioteca "Regular expression"

email\_pattern = r'^[a-zA-Z0-64.\_%+-]+@[a-zA-Z0-255.-]+\.[a-zA-Z]{2,}$' #Informa o padrão de email nesse caso: Um limite de 64 digitos para a primeira parte do email aceitando letras numeros e simbulos,um @, um limite de 255 digitos para a segunda parte do email aceitando apenas letras, numeros, hifens e pontos e um limite de 3 para a extensao do email

telefone\_pattern = r'^\(\d{2}\) \d{4,5}-?\d{4}$' #Informa o padrão de telefone nesse caso: Um limite de 2 digitos para a primeira parte (DDD) aceitando numeros, um espaço, um limite de 4 digitos para a segunda parte)

telefone\_patternB = r'^\d{8,9}$' #Um numero de 8 ou 9 digitos

DDD\_pattern = r'^\d{2}$' #Um limite de 2 digitos

resposta = str(input("Olá! para começarmos precisaremos de algumas informações suas!\nDeseja continuar? "))

if resposta[:3].lower() == "sim" : # a variavel é cortada para apenas 3 caracteres e em seguida convertida para minusculo, permitindo assim verificar se a resposta é "sim"

email = input("Digite seu E-mail: ")

while re.match(email\_pattern, email) == None : #Isso permite que o programa esta no padrao informado continuamente, apenas "parando" quando o usuario digitar corretamente

email = input("E-mail inválido!\nDigite novamente o E-mail: ")

print("E-mail validado!")

telefone = input("Digite seu telefone: ")

while re.match(telefone\_pattern, telefone) == None : # semelhante a anterior, porem aqui existe uma assistencia ao usuario. Diferentes sites e plataformas possuem padroes diferentes quanto a numero de telefones, para facilitar a vida do usuario que nao consegue informar o padrao corretamente, o programa ira auxiliar ele no processo

if re.match(telefone\_patternB, telefone) : #Utiliza o patternB para verificar se o usuario inseriu "corretamente" o numero de telefone, mas fora do padrao que sera armazenado

DDD = input("Digite seu DDD: ")

while re.match(DDD\_pattern, DDD) == None : #Enquanto o usuario nao inserir o DDD corretemante o programa continuara requisitando um

DDD = input("DDD inválido!\nDigite novamente seu DDD: ")

if len(telefone) == 9 :

telefone = f"({DDD}) {telefone[:5]}-{telefone[5:]}" #transforma todos os dados informados pelo usuario para montar o numero de telefone no padrao desejado pelo programa

else :

telefone = f"({DDD}) {telefone[:4]}-{telefone[4:]}"

else : #caso ele apenas tenha informado incorretamente o numero de telefone

telefone = input("Telefone inválido!\nDigite novamente o Telefone: ")

print("Telefone validado!")

print("Programa encerrado!")

#Feito por Pedro Cyrne

**NÃO COMENTADO:**

import re

email\_pattern = r'^[a-zA-Z0-64.\_%+-]+@[a-zA-Z0-255.-]+\.[a-zA-Z]{2,}$'

telefone\_pattern = r'^\(\d{2}\) \d{4,5}-?\d{4}$'

telefone\_patternB = r'^\d{8,9}$'

DDD\_pattern = r'^\d{2}$'

resposta = str(input("Olá! para começarmos precisaremos de algumas informações suas!\nDeseja continuar? "))

if resposta[:3].lower() == "sim" :

email = input("Digite seu E-mail: ")

while re.match(email\_pattern, email) == None :

email = input("E-mail inválido!\nDigite novamente o E-mail: ")

print("E-mail validado!")

telefone = input("Digite seu telefone: ")

while re.match(telefone\_pattern, telefone) == None :

if re.match(telefone\_patternB, telefone) :

DDD = input("Digite seu DDD: ")

while re.match(DDD\_pattern, DDD) == None :

DDD = input("DDD inválido!\nDigite novamente seu DDD: ")

if len(telefone) == 9 :

telefone = f"({DDD}) {telefone[:5]}-{telefone[5:]}"

else :

telefone = f"({DDD}) {telefone[:4]}-{telefone[4:]}"

else :

telefone = input("Telefone inválido!\nDigite novamente o Telefone: ")

print("Telefone validado!")

print("Programa encerrado!")