Intensivão de Python - 09 a 12/05/2022

Aula 1 - Automação de processos

Desafio da aula 1: todos os dias atualiza dados de venda de uma empresa, após isso, enviar e-mail para a diretoria. Python automatizado com análise dos dados e geração de relatórios.
Automação de comandos do mouse e teclado. No link disponibilizado passo a passo para fazer o desafio.
Será utilizado o Jupyter.

Quando construir qualquer projeto, por onde começar: primeiro escrever em português a solução do problema a ser resolvido, qual a lógica do seu desafio em português.
Passos para resolver o desafio:
Passo 1: entrar no sistema da empresa, o caso um link disponibilizado;
Passo 2: navegar no sistema e encontrar a base de dados, entrar na pasta
Passo 3: fazer o download
Passo 4: importar a base de dados para o python;
Passo 5: calcular os indicadores;
Passo 6: enviar um e-mail para o diretor com o relatório;

pyautogui é uma biblioteca com soluções diversas que serve para automatizar tarefas. Permite automatizar mouse, teclado e a tela do computador
pyperclip é uma biblioteca para copiar links com caracteres especiais, cuja finalidade é evitar erros nos códigos ao executar

É necessário instalar a biblioteca do pyautogui, basta digitar no Jupyter !pip install pyautogui e o !pip install pyperclip. Digitar sem o ! no terminal quando não utilizar o Jupyter

Sempre que usar alguma biblioteca deve ser chamada nas linhas de código, bastando digitar: import pyautogui

import pyperclip

Todos os comandos serão sempre pyautogui.click, pyautogui.white, pyautogui.press

pyautogui.hotkey() - esse comando é um conjunto de teclas, por exemplo pyautogui.(“ctrl”, “t”) abre uma nova janela do Chrome
pyautogui.press() - esse comando pressiona alguma tecla, a qual especificar entre os parênteses

pyautogui.white() - esse comando escreve o que for inserido entre parênteses e se for formato string, deve estar entre “” ou ‘’, aspas simples ou duplas
pyautogui.hotkey(“ctrl”, “c”, “ctrl”, “v”), sempre entre aspas, pois esse comando recebe caracteres como string
pyautogui.write() para escrever com o teclado
pyautogui.press() para apertar uma tecla
Com esses comandos podemos abrir programas e aplicativos

Como a execução é muito rápida, sempre inserir um delay, com o comando pyautogui.PAUSE = 1
O pyperclip permite utilizarmos caracteres especiais, usando o comando pyperclip.copy(“link do site ou sistema da empresa”) e depois usa o comando pyautogui.hotkey(“ctrl”, “v”)
Ao abrir sites e caixas de e-mail, utilizar o comando delay

Após entrar no site, depende da velocidade da Internet e isto é necessário tratar isso

Chamar o import time
Comando: time.sleep(5) esse comando vai pausar pontualmente
Esse comando time.sleep() - entre parênteses inserir o tempo necessário para conclusão da tarefa

Para navegar, usando automação com click usando o mouse
pyautogui.position() - devemos passar a posição
pyautogi.click() - inserindo a posição do ícone na tela para a automação clicar
Quando comandar pyautogui.position()
Usar a função time para encontrar a posição dos ícones na tela, sempre haverá uma posição x e y, por exemplo: pyautogui.click(x=1070 , y=725 , clicks=2)
Sempre ater para os tempos que o navegador carrega arquivos e usar time.sleep();
Sempre testar os comandos por etapas

Como importar uma base de dados para o python, qual comando lê um arquivo em excel. A biblioteca Pandas permite trabalharmos com base de dados em Python, este já vem instalado no Jupyter, bastas dar o comando import pandas
pandas
numpy
openpyxl
Quando não utilizar o Jupyter

Sempre usar import pandas as pd para análise de dados. No caso, foi utilizado uma planilha em excel
Armazenar o arquivo em uma variável, como por exemplo: tabela =

pd.read_excel(r“caminho completo do arquivo\nome do arquivo com extensão”) - sempre colocar um “r”antes do caminho do arquivo para não dar bug no python
Se tiver várias abas no excel, deve-se usar (nome do arquivo, sheets=1) aba 1
display(tabela) - vem mais organizado
print(tabela) - pode utilizar esse comando, porém vem em formato complicado de analisar

Na análise de dados da tabela, deve-se usar os comandos e armazenar em variável. Por exemplo: faturamento = tabela[“Valor Final”].sum(). Ater para o nome da coluna
Só depois que finalizar o código, aí sim se preocupar com a formatação dos dados privados

Tornar esse código como executável. Assistir um vídeo no YouTube uma aula como tornar um Python arquivo executável.