

CUPRINS

1	Documentarea mediului	4
1.1.1	Python interpretor	4
1.1.2	Installed packages	4

1 Documentarea mediului

1.1 Python interpretor

Versiunile de Python folosite pentru această tema sunt :

- Python version: 3.9.7
- Pip version: 23.2.1

1.2 Installed packages

Pachetele Python instalate pe calculatorul meu sunt urmatoarele cu versiunile specifice

	Fisiere importate python	Pachete instalate
Library for audio	import soundcard as sc	SoundCard==0.4.2
	import soundfile as sf	soundfile==0.12.1
Library for video	import cv2	opencv-contrib-python==4.8.0.76 opencv-contrib-python-headless==4.8.0.76
	import pyautogui	PyAutoGUI==0.9.54
	import numpy as np	numpy==1.22.4
Library for opening a page on youtube	#from selenium import webdriver	webdriver-manager==4.0.1
	import pywhatkit	pywhatkit==5.4
	import time	
include threading	import threading	
Audio Analysis	import matplotlib.pyplot as plt	matplotlib==3.4.1
	from scipy.io import wavfile	scipy==1.6.2
	from scipy import signal	
Library to calculate SPL	from pydub import AudioSegment	
	import scipy.io.wavfile as wavfile	scipy==1.6.2
	import math	
	import csv	

Documentarea pachetelor instalate

Fisiere importate python	Descriere
import soundcard as sc	<p>Rol de a gestiona microfoane și difuzoare, în aplicațiile Python. Se folosește la înregistrarea audio sau redarea audio.</p> <p>Cu soundcard, puteți lista și selecta dispozitivele audio disponibile pe sistemul dvs. Acest lucru vă permite să alegeți dispozitivul audio pe care doriți să-l utilizați pentru înregistrare sau redare.</p> <ol style="list-style-type: none"> 1. Gestionarea Dispozitivelor Audio: Cu soundcard, puteți lista și selecta dispozitivele audio disponibile pe sistemul dvs. Acest lucru vă permite să alegeți dispozitivul audio pe care doriți să-l

	<p>utilizați pentru înregistrare sau redare.</p> <ol style="list-style-type: none"> 2. Înregistrarea Audio: Biblioteca vă permite să înregistrați sunet de la microfoane sau alte surse audio conectate la sistem. Puteți specifica parametri precum samplerate (rată de eșantionare), numărul de cadre și altele pentru a controla înregistrarea. 3. Redarea Audio: Puteți utiliza soundcard pentru a reda sunet pe difuzoarele sau dispozitivele audio conectate la sistem. Acest lucru vă permite să redați fișiere audio sau să generați sunet dinamic în aplicațiile dvs. 4. Manipularea Datelor Audio: Puteți utiliza biblioteca pentru a manipula date audio, cum ar fi convertirea între diferite formate audio sau procesarea semnalului audio. 5. Suport pentru Dispozitive Diferite: soundcard oferă suport pentru diferite platforme și sisteme de operare, permițându-vă să lucrați cu dispozitive audio indiferent de platforma pe care o utilizați.
import soundfile as sf	<p>oferă funcționalități pentru citirea și scrierea fișierelor de sunet în diferite formate audio. Această bibliotecă este utilă atunci când doriți să manipulați sau să procesați date audio din fișiere audio sau să salvați date audio în formate specifice.</p> <p>ată câteva dintre rolurile principale ale bibliotecii soundfile în Python:</p> <ol style="list-style-type: none"> 1. Citirea Fișierelor Audio: soundfile permite citirea fișierelor audio din diverse formate, cum ar fi WAV, FLAC, AIFF, MP3 și multe altele. Puteți utiliza funcții precum soundfile.read() pentru a citi datele audio din fișiere și pentru a le stoca într-o structură de date Python (de obicei, un array NumPy). 2. Scrierea Fișierelor Audio: Puteți utiliza soundfile pentru a crea sau a scrie date audio în fișiere audio în diferite formate. Funcții precum soundfile.write() vă permit să specificați formatul fișierului și ratele de eșantionare pentru a salva datele audio în formatul dorit. 3. Manipularea Datelor Audio: Biblioteca oferă funcționalități pentru manipularea datelor audio, cum ar fi schimbarea ratei de eșantionare, convertirea între formate audio și efectuarea de operații de procesare a semnalului audio. 4. Suport pentru Diverse Formate Audio: soundfile este capabilă să lucreze cu o gamă largă de formate audio, ceea ce o face foarte versatilă pentru manipularea fișierelor audio.
import cv2	<p>Prelucrarea Imaginilor: OpenCV oferă numeroase funcții și metode pentru prelucrarea imaginilor, inclusiv redimensionarea, tăierea, rotirea, filtre pentru corecția culorii și contrastului, operații aritmetice cu imagini, și multe altele.</p>
import pyautogui	<p>Biblioteca pyautogui în Python este folosită pentru a automatiza interacțiunile cu interfața utilizatorului pe un ecran de calculator. Ea furnizează instrumente pentru controlul mouse-ului, tastaturii și pentru efectuarea de capturi de ecran. Iată câteva dintre rolurile și funcționalitățile principale ale pyautogui:</p> <ol style="list-style-type: none"> 1. Controlul Mouse-ului: pyautogui permite simularea mișcărilor și clicurilor mouse-ului. Puteți poziționa cursorul la anumite coordonate, face clic-uri pe butoanele mouse-ului sau efectua clicuri multiple, precum clic-dreapta și clic-stânga.

	<ol style="list-style-type: none"> 2. Controlul Tastaturii: Puteți simula introducerea de text sau taste folosind pyautogui. Acesta poate fi util pentru automatizarea introducerii de date în formulare sau pentru efectuarea de comenzi de la tastatură. 3. Timp și Pauze: Biblioteca permite controlul timpului și adăugarea de pauze între acțiuni pentru a simula interacțiunea umană. Aceasta este importantă pentru a evita probleme de performanță sau erori în timpul automatizării. 4. Capturi de Ecran: pyautogui poate efectua capturi de ecran ale întregului ecran sau ale unei regiuni specifice. Acest lucru este util pentru capturarea de imagini din aplicații sau pentru generarea de documentație automată. 5. Citirea Informațiilor din Ecran: Biblioteca poate citi informații de pe ecran, cum ar fi culorile pixelilor sau valorile de pixel, pentru a identifica elemente sau caracteristici ale interfeței utilizatorului. 6. Automatizarea Proceselor: pyautogui este utilizată pentru automatizarea sarcinilor repetitive pe computer, cum ar fi interacțiunile cu aplicații sau site-uri web. 7. Testare Automată: Este folosită pentru a efectua testare automată a aplicațiilor sau site-urilor web, simulând interacțiunile utilizatorului pentru a verifica funcționalitatea corectă. 8. Crearea de Scripturi de Automatizare: Cu pyautogui, puteți crea scripturi de automatizare personalizate pentru sarcinile sau fluxurile de lucru specifice.
import numpy as np	rol central în calculul științific și în prelucrarea datelor.
#from selenium import webdriver	<p>utilizată pentru automatizarea interacțiunilor cu browser-ele web. Ea oferă instrumente puternice pentru deschiderea de pagini web, interacțiunea cu elementele paginii, executarea de acțiuni ca și un utilizator real și extragerea de date din paginile web. Iată câteva dintre rolurile și funcționalitățile cheie ale selenium în Python:</p> <ol style="list-style-type: none"> 1. Automatizarea Navigării Web: selenium permite programatorilor să deschidă browser-e web (cum ar fi Chrome, Firefox, sau altele) și să navigheze pe site-urile web automate. Aceasta poate fi utilă pentru teste automate, extragerea de date sau pentru efectuarea de operațiuni repetitive pe web. 2. Interacțiunea cu Elementele Paginii Web: selenium furnizează funcții pentru identificarea și interacțiunea cu elementele paginii web, cum ar fi butoane, casete de text, meniuri dropdown, casete de selectare, și altele. Puteți să faceți clic pe elemente, să completați formulare, să extrageți text sau să selectați opțiuni. 3. Așteptarea Explicită: Biblioteca oferă suport pentru așteptarea explicită, ceea ce înseamnă că puteți specifica anumite condiții sau evenimente pe care trebuie să le așteptați înainte de a continua. Acest lucru este util pentru gestionarea încărcării dinamice a paginilor web sau pentru așteptarea apariției unui element. 4. Manipularea Ferestrelor și Tab-urilor: selenium permite manipularea ferestrelor browser-ului și a tab-urilor. Puteți

	<p>deschide noi ferestre, comuta între ele și efectua acțiuni în fiecare fereastră sau tab separat.</p> <ol style="list-style-type: none"> 5. Manipularea Cookie-urilor și Sesizărilor: Puteți gestiona cookie-urile, să ștergeți, să adăugați sau să modificați cookie-uri în sesiunea browser-ului. De asemenea, puteți manipula sesiuni de autentificare și să completați formulare de autentificare. 6. Captura de Ecran și Salvarea de Fișiere: Puteți efectua capturi de ecran ale paginilor web și să salvați pagini web complete sau elemente specifice ale paginii ca fișiere. 7. Testare Automată a Aplicațiilor Web: Selenium este folosită pe scară largă pentru testarea automată a aplicațiilor web. Puteți crea și rula scripturi de test automat pentru a verifica funcționalitățile aplicației web. 8. Web Scraping (Extragerea de Date): Selenium poate fi folosită pentru a extrage date din pagini web, inclusiv text, imagini, sau orice alt conținut disponibil public pe internet. 9. Interoperabilitate cu Alte Biblioteci: selenium poate fi combinată cu alte biblioteci pentru a efectua acțiuni avansate, precum manipularea datelor extrase din pagini web cu biblioteci de procesare a datelor sau pentru a automatiza testarea interfețelor de utilizator în cadrul aplicațiilor web. 10. Monitorizarea Navigării și Rularea de Scripturi JavaScript: selenium permite monitorizarea și înregistrarea acțiunilor de navigare web și poate rula scripturi JavaScript pe paginile web. <p>În ansamblu, selenium este o bibliotecă puternică pentru automatizarea interacțiunilor cu browser-e web, facilitând dezvoltarea de aplicații de testare, automatizarea sarcinilor pe web și extragerea de date din paginile web.</p>
import pywhatkit	<p>Biblioteca pywhatkit este o bibliotecă Python care oferă funcționalități pentru automatizarea unor sarcini comune, cum ar fi trimiterea de mesaje WhatsApp, redarea de videoclipuri YouTube, căutarea pe Google, și multe altele. Iată câteva dintre rolurile și funcționalitățile cheie ale pywhatkit:</p> <ol style="list-style-type: none"> 1. Trimiterea de Mesaje WhatsApp: pywhatkit permite trimiterea automată de mesaje pe WhatsApp. Aceasta poate fi folosită pentru a trimite mesaje text sau chiar imagini către contactele WhatsApp. 2. Redarea de Videoclipuri YouTube: Puteți utiliza pywhatkit pentru a căuta și reda videoclipuri de pe YouTube direct dintr-un script Python. De asemenea, puteți reda videoclipuri în mod aleatoriu sau să specificați anumite videoclipuri. 3. Căutarea pe Google: Biblioteca oferă funcții pentru a căuta automat pe Google folosind un termen de căutare specificat. Rezultatele căutării pot fi deschise într-o fereastră de browser. 4. Căutarea pe Wikipedia: Puteți căuta informații în Wikipedia folosind pywhatkit și puteți extrage informații utile pentru a le utiliza în scripturile dvs. 5. Generarea de Text în Manuscris: pywhatkit poate fi utilizată pentru a transforma textul obișnuit în text în manuscris. Acest lucru poate fi folosit pentru a crea efecte vizuale în documente

	<p>sau pentru alte scopuri creative.</p> <ol style="list-style-type: none"> Traducerea de Text: Biblioteca oferă funcționalități pentru a traduce textul în diverse limbi străine, folosind servicii precum Google Translate. Generarea de QR Codes: Puteți crea coduri QR pentru diverse tipuri de informații, cum ar fi adrese URL, texte sau altele, utilizând pywhatkit. Transformarea de Text în Voce: Biblioteca permite transformarea textului în vorbire și redarea acestuia folosind sintetizarea vocală. Identificarea Codurilor QR din Imagini: pywhatkit poate fi utilizată pentru a identifica și a decoda codurile QR din imagini. Manipularea Imaginilor: Biblioteca oferă funcții pentru manipularea imaginilor, cum ar fi redimensionarea, rotirea sau adăugarea de efecte. <p>pywhatkit este o bibliotecă utilă pentru automatizarea unor sarcini comune și pentru adăugarea de funcționalități interesante în proiectele Python. Ea facilitează interacțiunea cu servicii populare precum WhatsApp, YouTube și Google, precum și manipularea de date text și imagine.</p>
import time	<p>Crearea de Pauze: <code>time.sleep()</code> permite programatorilor să introducă pauze în execuția programului pentru o anumită perioadă de timp. Acest lucru poate fi util pentru a aștepta resurse sau pentru a efectua temporizări.</p>
import threading	<p>Biblioteca <code>threading</code> în Python este folosită pentru a crea și controla thread-uri (fire de execuție) într-un program Python. Thread-urile sunt unități de execuție independente care permit programului să efectueze mai multe sarcini în paralel sau concurent, ceea ce poate îmbunătăți performanța și eficiența în aplicații care au sarcini intensive sau care trebuie să răspundă la evenimente în timp real. Iată câteva dintre rolurile și funcționalitățile cheie ale bibliotecii <code>threading</code>:</p> <ol style="list-style-type: none"> Crearea de Thread-uri: Biblioteca <code>threading</code> permite programatorilor să creeze thread-uri noi într-un program Python. Un thread reprezintă un flux de control separat, care poate executa cod independent de celelalte thread-uri. Concurența: Utilizând thread-uri, puteți efectua mai multe sarcini în paralel, ceea ce poate îmbunătăți performanța în aplicații care au sarcini paralele sau care așteaptă resurse externe, cum ar fi date de la rețea sau de la dispozitive hardware. Coordonarea Thread-urilor: <code>threading</code> furnizează mecanisme pentru sincronizarea și coordonarea activităților thread-urilor. De exemplu, puteți utiliza semafoare, bariere sau mutex-uri pentru a asigura accesul exclusiv la resurse partajate sau pentru a sincroniza activitățile thread-urilor. Controlul Ciclurilor de Viață ale Thread-urilor: Puteți inițializa, porni, suspenda, relua și opri thread-urile folosind funcțiile și metodele din <code>threading</code>. Gestionarea Excepțiilor: <code>threading</code> oferă facilități pentru gestionarea excepțiilor în thread-uri. Puteți prinde și gestiona excepții într-un mod specific pentru fiecare thread.

	<ol style="list-style-type: none"> 6. Thread-uri Daemon: Puteți crea thread-uri daemon, care se termină automat atunci când programul principal se încheie. Acestea sunt utile pentru sarcini de fundal care nu necesită să fie finalizate înainte de încheierea programului. 7. Limitarea Numărului de Thread-uri Active: Puteți controla numărul de thread-uri active pentru a evita supraîncărcarea sistemului sau pentru a gestiona resursele în mod eficient. 8. Comunicarea între Thread-uri: threading permite thread-urilor să comunice între ele folosind obiecte precum cozi (queues) sau pipe-uri, permițând partajarea datelor sau comunicarea asincronă între thread-uri. 9. Programare Asincronă: threading poate fi utilizată pentru a implementa programare asincronă în Python, permițând gestionarea evenimentelor și sarcinilor concurente în aplicații. 10. Debugging și Profilare: threading oferă instrumente pentru debugare și profilare, pentru a identifica și remedia problemele legate de thread-uri într-o aplicație.
--	---