

פרויקט בתכנות מונחה אובייקטיבים א':

מה הפרויקט עושה:

המשתמש מכניס שם של שיר, התוכנה מחזירה לו רשימה של חמישה שירים מהם הוא בוחר את השיר שהוא רוצה. לאחר בחירת השיר נפתח השיר והקליפ בחולון חדש באתר יוטיוב.

חבילות שהודרנו:

בשביל להוריד את כל החבילות הרלוונטיות עשינו את השלבים הבאים:
 fill → setting → name of the package
 רשימה של החבילות ומה כל חבילה עשוה -

a. **import webbrowser** – פתיחה מיידע מדף אינטרנט.

b. **import requests** – ספרייה לעבודה עם HTTP. מפשט את התהליך של קבלת וקריאה הנתונים מהאינטרנט מול כתיבת הקוד ב python .

c. **from bs4 import BeautifulSoup** - ספרייה שמקלה על העבודה בין האינטרנט לפיתון, אפשר להבין בעזרתה איך בניית האתר אינטרנט ולגש למידע הרלוונטי בפייטון לפי כוורות וכו'.

d. **from selenium import webdriver** - מייצג את העבודה מול הדפדפן (פתיחה דפדפן, גלישה לאתרם וכו'). חבילת selenium של פייטון משמשת לביצוע בדיקות אוטומטיות של אתרים כי היא שולטת בדף ומאפשרת לכוון את ההתקנות שלו. בעזרת הספרייה זו הганנו על הקושי לגשת שירותים לאתר אינטרנט של גוגל. בעודנו פותחים דפדפן אינטרנט חדש "שעוקף" את הקשיים של הגישה ישירה לאתר גוגל.

e. **import time** – חבילה שבעזרתה מגדרים את הזמן עד שהחלון האינטרנט יפתח "זמן הדילוי".
 f. **import tkinter as tk** – הורדת GUI בעזרו אנו מעצבות את החלון שנפתח למשתמש, ככלומר העיצוב של הממשק בין התוכנה למשתמש.

g. **-from tkinter import Tk, Button, Label, Entry, StringVar, Listbox, END, SINGLE** – פירוט של הפונקציות ב GUI שבהם השתמשנו.

הסבר כללי על הקוד:

- **שורות 8-11:** החבילות בהן השתמשנו. (מפורט לעיל)
- **שורות 37-40:** שמות המשתנים המופיעים בקוד (הסביר על כל אחד בהמשך)
- **שורות 125-40:** מחלוקת שיצרנו ובה מספר פונקציות האחראיות על כל התהליכי של חיפוש סרטונים ביוטיוב והציג התוצאות למשתמש באמצעות ממשק גרפי. (פירוט על כל פונקציה בהמשך).
- המחלקה YouTubeDisplayProgram נועדה ליצור ממשק גרפי (GUI) שבו המשתמש יכול להזין שם של שיר, לבצע חיפוש ביוטיוב, ולהציג רשימה של תוצאות חיפוש לרלוונטיות. התוצאה המוצגת היא רשימה של עד 5 שירים, כאשר משתמש בוחר אחד מהם, יפתח הקליפ ביוטיוב בחולון חדש.
- **שורות 128-130:** הפעלת פעולות המחלוקת בעת הרצת הקוד.

פידוט של שמות המשתנים:

SLEEP_TIME	= הזמן שמחכים עד שנפתחחלון לאחר בחירת השיר הרצוי
YOUTUBE_URL	= כתובת המיצגת מיקום יוטיוב באינטרנט
SEARCH_URL	= הכתובת של יוטיוב באינטרנט בה נחפש את השיר
TITLE_REQUEST_FROM_USER	= שם השיר אותו המשתמש מכניס
WINDOW_OPEN_TITLE	= שם (כותרת) החלונית של GUI שנפתחה כשמריצים את התוכנית
WINDOW_FRAME_SIZE	= גודל החלונית של GUI שנפתחה
WINDOW_BACKGROUND_COLOR	= צבע החלונית של GUI שנפתחה
INPUT_FRAME_COLOR	= צבע מסגרת החלונית בה המשתמש מכניס את שם השיר
TITLE_FONT	= גופן הכותרת בחלוןית
TITLE_SIZE	= גודל הכותרת בחלוןית
INPUT_FONT	= גופן הקלט שהמשתמש מכניס (שם השיר)
INPUT_FONT_SIZE	= גודל הגוף של הקלט
INPUT_BOX_SIZE	= גודל הלשונית בה המשתמש כותב את הקלט
BUTTON_LABEL	= הכפתור עליו המשתמש לוחץ לאחר כתיבת הקלט
BUTTON_FONT	= גופן הכפתור
BUTTON_FONT_SIZE	= גודל גופן הכפתור
BUTTON_BACKGROUND	= צבע רקע של הכפתור
BUTTON_FONT_COLOR	= צבע הגוף של הכפתור
OPTIONS_FONT	= הגוף של 5 האופציות של השירים לאחר הכנסת הקלט ולחיצה על כפתור "ENTER"
OPTIONS_FONT_SIZE	= גודל הגוף של האופציות המוצעות
OPTIONS_WINDOW_SIZE	= גודל החלונית שבה רואים את האופציות המוצעות
OPTION_WINDOW_HIGH_SIZE	= הגובה המקורי של החלונית האופציות המוצעות

פידוט של הפונקציות בהן השתמשנו:א. `:def get_html_content(search_url)`

המשתנה `search_url` מכיל את הכתובת URL של החיפש ביוטיוב עליו עושים את החיפוש של העדשה המשמש הכנס.

שימוש ב `selenium` :

- מפעילים דףון כרום במצב headless – מצב שבו אין ממשק גרפי פתוח למשתמש ובכך אנו ניגשים לתוצאות.
- המשתנה `driver` : שם משתמש אליו טוענים את כתובת URL שקלטו מהמשתמש.
- `time.sleep` : מגדר כמה זמן מחכים עד שהתוכן יטען.
- `element = driver.find_element(By.CSS_SELECTOR, "html")`
- `html = element.get_attribute("outerHTML")`
- שתי השורות האלו מוצאות את מה שאנו מ Chapman לфи הקלט של המשתמש ולוקחת את URL המלא.
- לאחר מכן סגורים את הדף.

ב. : def update_list(html, listbox)

בעזרת הפונקציה זו אנחנו מנהים את כתובות URL ושואים ממנה את המידע הרלוונטי לנו. בנוסח הפונקציה מעדכנת את listbox - משתנה שמציג את רשימת השירים שיוצגו למשתמש.

שימוש ב BeautifulSoup :

- משתמשים בחבילה זו בשיל "לפרק" את הכתובת URL ולקחים ממנה את התגיות עם המידע הרלוונטי לנו - שמות השיר וקישורים לקליפ.
- list_of_options : מתחלים רשימה ריקה שתכיל את האפשרויות של השירים והכתובות URL שלהם.
- לפני שטוענים את האופציות למשתנה listbox מוחקים את הנתונים הקודמים.
- בעזרת הולאה רצים על כל תוכאות החיפוש. בכל פעם שהולאה רצתה אנחנו לוקחים את המידע הבא :

 1. שם השיר מתוך הכותרת : txt = link.text.strip()
 2. קישור לקליפ : a = link.select("[href][0]a")
 3. בניית רשימה של השירים עם שם השיר מהcotורתה וה קישור URL : lst_of_options.append({'name': txt, 'url': a})
 4. הוספת שם השיר ל listbox.insert(END, txt) : listbox.insert(END, txt)

אחרנו בחרנו שהרשימה שתוצג למשתמש תהיה 5 תוכאות בלבד (אפשר לשנות את זה).

5. לבסוף הפונקציה מחזירה את רשימת השירים עם כתובות URL.

ג. :def user_select(listbox, lst_of_options)

הfonקציה אחראית להפעיל פיזית את הבחירה של המשתמש. ראשית בודקים האם הבחירה בבחירה ע"י המשתמש, במידה ולא התקבלה התוכנית מפסיקת לרוֹא. לאחר מכן אנו שומרים את האינדקס שלבחירה המשתמש - [0](). index = listbox.curselection() 이후 בונים את הכתובת URL המלאה לפיבחירה המשתמש - [url][url] song_url = YOUTUBE_URL + lst_of_options[index] בשלב האחרון הפונקציה פותחת את הקישור של השיר - (webbrowser.open_new(song_url))

ד. : def search_and_open(entry_song_name, listbox)

הfonקציה מבצעת חיפוש ביוטיוב על פי שם השיר שהמשתמש הזין, מעדכנת את Listbox עם תוכאות החיפוש, ומאפשרת למשתמש לבחור שיר מתוך הרשימה ולפתח אותו בדף.

- קבלת הקלט : search = entry_song_name.get()
 - בניית כתובות URL מלאה : search_url = SEARCH_URL + str(search)
 - שליחת הבקשה וקבלת ה html = get_html_content(search_url) : HTML
 - עדכן תיבת הרשימה עם האופציות לבחירה : lst_of_options = update_list(html, listbox)
 - בעזרת הפונקציה lambda אנו מבטיחים שהמשתמש יבחר באופציה אחת מכל האופציות שיש ומהזירה את הקlip.
- listbox.bind('<<ListboxSelect>>', lambda choose: user_select(listbox, lst_of_options))

ה. : def create_gui()

הfonקציה אחראית על הממשק בין המשתמש למחשב היא יוצרת חלון אינטראקטיבי מול המשתמש. בחלון קולטים את השיר אותו רוצים לשמע - מחזירים רשימה של חמץ אופציות לבחירה, המשתמש בוחר אופציה אחת מתוכן ונפתח לו חלון חדש עם הקlip לאתר יוטיוב.