

הורדת נתונים מהרשת ורשתות חברתיות

מטרת מעבדה זו היא להכיר כלים שונים להורדת נתונים מהרשת - scraping
עליכם לבנות אפליקציה עם ממשק פשוט (עדיף גרפי) ולמטרת POC בלבד
עליכם להתנסות בסוגים שונים של חבילות להורדה, להשוות ביניהם ולממש שמירה לבסיס נתונים מומלץ
MongoDB.
מומלץ להפריד את הקוד לשכבות (תצוגה, לוגיקה ובסיס הנתונים)
בעבור כל כלי יש להגדיר אפשרות חיפוש כלשהי מתוך ה API של הכלי ואפשרות שמירה לבסיס הנתונים את
הרשומות החוזרות
מומלץ שתהיה דרך לאחזר את הנתונים מבסיס הנתונים

דוגמאות קוד
שמירה למונגו

```
import pymongo

myclient = pymongo.MongoClient("mongodb://localhost:27017/")

mydb = myclient["mydatabase"]

print(myclient.list_database_names())
dblist = myclient.list_database_names()
if "mydatabase" in dblist:
    print("The database exists.")

mycol = mydb["customers"]
print(mydb.list_collection_names())
collist = mydb.list_collection_names()
if "customers" in collist:
    print("The collection exists.")
mydict = { "name": "AAA", "address": "BBBB" }

x = mycol.insert_one(mydict)
```

יצירת GUI

```
import tkinter
r = tk.Tk()
```

```

r.title('Counting Seconds')
button = tk.Button(r, text='Stop', width=25,height=25, command=r.destroy)
button.pack()
r.mainloop()

```

הורדת פוסטים מפייסבוק

```

from selenium import webdriver
import time
def download_facebook_post(page):
    driver = webdriver.Chrome()
    driver.get('https://www.facebook.com/' + page + '/')
    for scroll in range(5):
        driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
        time.sleep(2)
    posts = driver.find_elements_by_class_name("userContentWrapper")
    result = []
    for post in posts:
        row = {}
        row.update({"scrap_type": "facebook"})
        row.update({"page": page})
        time_element = post.find_element_by_css_selector("abbr")
        utime = time_element.get_attribute("data-utime")
        row.update({"utime" :utime})

        text = ""
        text_elements = post.find_elements_by_css_selector("p")
        for elm in text_elements:
            text += elm.text
        row.update({"post": text})
        result.append(row)
    driver.close()
    return result

```

מקורות והתקנות

```

pip install pymongo
pip install twint

```

```
pip install selenium
pip install instalooter
pip install beautifulsoup4
```

חבילה המאפשרת הורדת טוויטים

<https://github.com/twintproject/twint>

מימוש בעזרת סלניום להורדה מפייסבוק

<https://github.com/LeviBorodenko/non-api-fb-scraper>

הורדה מאינסטגרם

<https://github.com/althonos/InstaLooter>

הורדה מאינסטגרם (דורש הזדהות לחלק מהמתודות)

<https://pythonawesome.com/a-minimalistic-instagram-scraper-written-in-python/>

הורדת דפים מהרשת

<https://2.python-requests.org/en/latest/>

חבילה לעיבוד וניתוח דפי HTML

<https://pypi.org/project/beautifulsoup4/>

מדריך לסקרפינג מאינסטגרם עם סלניום ו BeautifulSoup

<https://medium.com/@srujana.rao2/scraping-instagram-with-python-using-selenium-and-beautiful-soup-8b72c186a058>

מדריך לסקרפינג אתרים עם BeautifulSoup

<https://www.dataquest.io/blog/web-scraping-tutorial-python/>

תרגילים שונים בסקרפינג - כל תרגיל עובד על סט יכולות שונה.

<https://blog.michaelyin.info/scrapy-exercises-make-you-prepared-for-web-scraping-challenge/>

מדריך לסקרפינג אתרים עם BeautifulSoup

<https://towardsdatascience.com/how-to-web-scrape-with-python-in-4-minutes-bc49186a8460>

מדריך בסיסי לסקרפינג אתרים מהרשת סלניום ו BeautifulSoup

<https://www.edureka.co/blog/web-scraping-with-python/>