**מטלת סוף מודול פייתון**

אריאל מלינקר

תשובה לתרגיל 3:

import requests **#using requests module**   
from bs4 import BeautifulSoup as bs **#using BeautifulSoup from bs4 module and using "bs" as reference**   
import os **#using os module**

wantet\_path = input("enter the path you want to save the results in\n>") **#asking the user for a path for the function to use**

def imagedown(url,path, folder): **#create a function called imagedown**  
 try:  
 os.chdir(path+f"\{folder}") **#gose to the chosen path (user choses the path and {folder} is a variable that I get from the main function**  
 os.mkdir("images") **#create a folder in the path I chose last command the folder called images**  
 except:  
 pass  
 r = requests.get(url) **#get a link from the variable "url" the I get from the main function**  
 soup = bs(r.text, 'html.parser') **#use parser on the link I got last command**  
 images = soup.find\_all("img") **#search for tags called img**  
 for image in images: **#put the content of images in image until it's out of content**  
 try:  
 link = image.get('src') **#search for everything under src**  
 img\_name = link.split("/")[-1] **#creating a name for the image by taking the last part of the link**  
 except:  
 pass  
 try:  
 with open(path+"\{folder}\images\{img\_name}",'wb') as f: **#open a file in the chosen path under f as reference**   
 try:  
 im = requests.get(link) **#take all the links from variable link**  
 f.write(im.content) **#write everything "im" variable took from the links to** the file opened   
 except:  
 pass  
 except:  
 pass  
 print("all images downloaded") **#print on screen "all images downloaded"**  
  
def pdfdown(url,path, folder): **#create a function called pdfdown**  
 link\_list = [] **# empty list called link\_test**  
 unique = [] **# empty list called unique**   
 try:  
 os.chdir(path+"\{folder}") **#gose to the path the user chose ({folder} is a variable that I get from the main function**  
 os.mkdir("files") **#create a folder in the path I chose last command the folder called files**  
 except:  
 pass  
 r = requests.get(url) **#get a link from the variable "url" the I get from the main function**  
 soup = bs(r.text, "html.parser") **#use parser on the link I got last command**  
 for a in soup.find\_all("a"): **#search for tags called a and put them in a variable called 'a'**  
 links = a.get("href") **#search for everything under "href" and put it in variable "links"**  
 if ".pdf" in str(links) or ".txt" in str(links): **#if there is .pdf or .txt in any link in variable links do:**   
 link\_list.append(str(links)) **#put that links in an empty list called link\_list**  
 for y in link\_list: **#gose through every link and put it in variable y**  
 if y not in unique: **#if y content doesn’t match unique content do:**  
 unique.append(y) **#put the content of y in unique empty list**  
 for z in unique: **#gose through everything in unique and put it in z**  
 pdf\_name = z.split('/')[-1] **#create a file name it's the name of the last part of the link**  
 try:  
 with open(path+f"\{folder}\files\{pdf\_name}",'wb') as f: **#create a file in the chosen path**  
 try:  
 im = requests.get(z) **#get the links from variable z into im**  
 f.write(im.content) **#write all information from the links into the new file**  
 except:  
 pass  
 except:  
 pass  
 print("all files downloaded") **#print on screen "all files downloaded"**  
  
while 1: **# loop that goes forever**  
 url = input("enter the url you want to scrape\n>") **#ask the user for al link to search**   
 folder = input("enter a name for a new folder\n>") **#ask the user for a name for the new folder**  
 try:  
 os.chdir(wanted\_path) **#gose to the path the user chose**  
 os.mkdir(folder) **#create a folder in the chosen path with the name the user chose**  
 except:  
 pass  
 r = requests.get(url) **#get the link the user gave**   
 soup = bs(r.text, "html.parser") **#use parser on the link the user gave**  
 for a in soup.find\_all('a'): **#everything under the tag "a" goes in variable "a"**  
 links = a.get('href') **#eveything under "href" goes in variable links**  
 if "https" in str(links): **#if there is "https" in variable links do:**  
 print(links) **#print everything in variable links**  
 txt\_file = folder + " links.txt" **#create a name for a txt file**   
 with open(wanted\_path+f"\{folder}\{txt\_file}", 'a') as f: **#create a file in the new folder**  
 f.write(str(links)+"\n") **#write the content of "links" in the file**   
 pdfdown(url,wanted\_path, folder) **#calling the pdfdown function given it three variables**  
 imagedown(url,wanted\_path folder) **#calling the imagedown function given it three variables**  
 print("a file containing all links including links without https//: created in your folder ") **#talling the user there is a file created with all links**  
 again = input("""you want to scan another url  
y= yse  
n =no  
>""") **#asking the user if he want to scan another link**  
 if again == "y": **#if the content of variable again equal "y" do:**  
 continue **#go back to the start of the loop**  
 elif again == "n": **#if the content of the variable again equal "n" do:**  
 break **#break from the loop**  
 else: **#if the content of again do not equal "y" or "n" do:**  
 print("error") **#print on screen "error"**  
 break **#break from loop**