ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
Τμήμα Μηχανικών Ηλεκτρονικών
Υπολογιστών και Πληροφορικής

ΑΡΧΕΣ ΓΛΩΣΣΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΚΑΙ ΜΕΤΑΦΡΑΣΤΩΝ

Εργασια Python

ΟΝΟΜΑΤΕΠΩΝΥΜΟ / ΑΜ:

Βουλδής Άγγελος / ΑΜ:1059624

Ο κώδικας σε python, χωρίζεται σε 5 αρχεία:

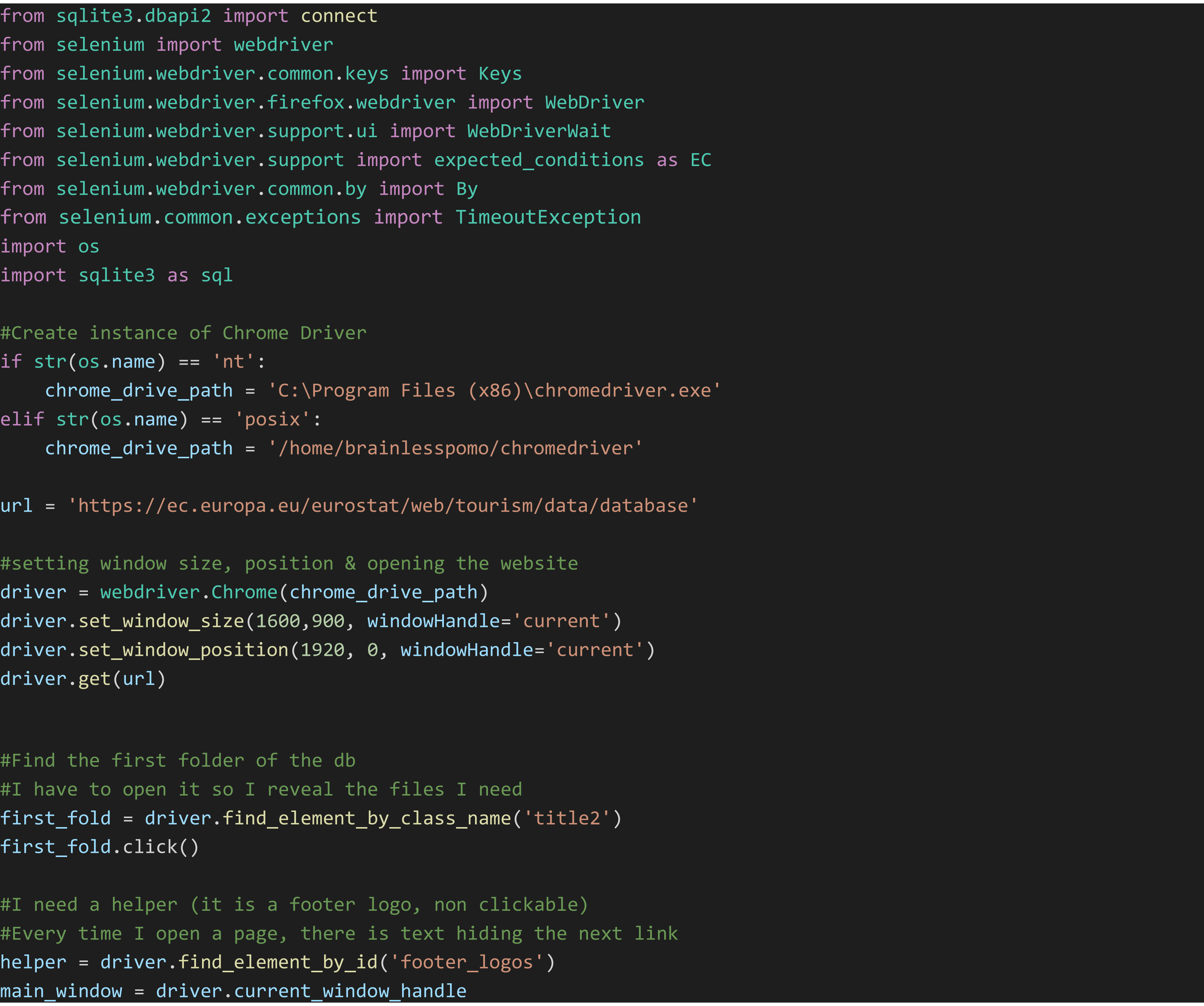
Στο $1^{ο}$ αρχείο χρησιμοποιώ το selenium, για να μπορέσω να μπω στην ιστοσελίδα και να πάρω τα δεδομένα που θα χρειαστώ.

Μετά τα αποθηκεύω, σε 4 arrays, όσες είναι και οι ιστοσελίδες από τις οποίες χρειάζεται να πάρω δεδομένα.

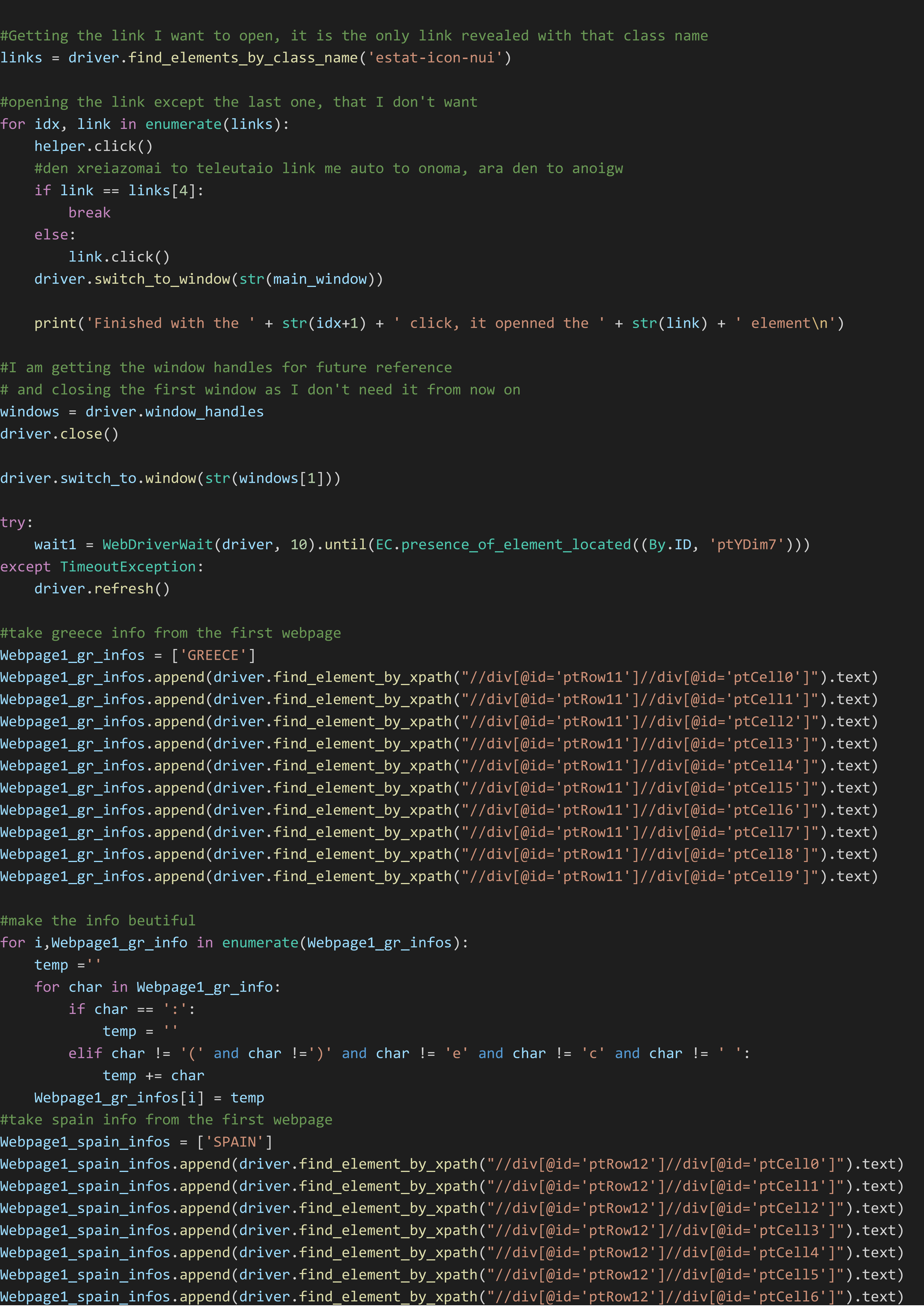
Στην συνέχεια, βγάζω τα άχρηστα δεδομένα ((e), (c), :) που υπάρχουν στην ιστοσελίδα, έτσι ώστε να μπορέσω να τα βάλω σωστά μέσα στο database.

(Επίσης, για έλεγχο, αποθηκεύω και τα στοιχεία σε 4 txt αρχεία, που βρίσκονται στον φάκελο stored_data, το κάθε αρχείο έχει μέσα τα αντίστοιχα δεδομένα με το όνομά του)

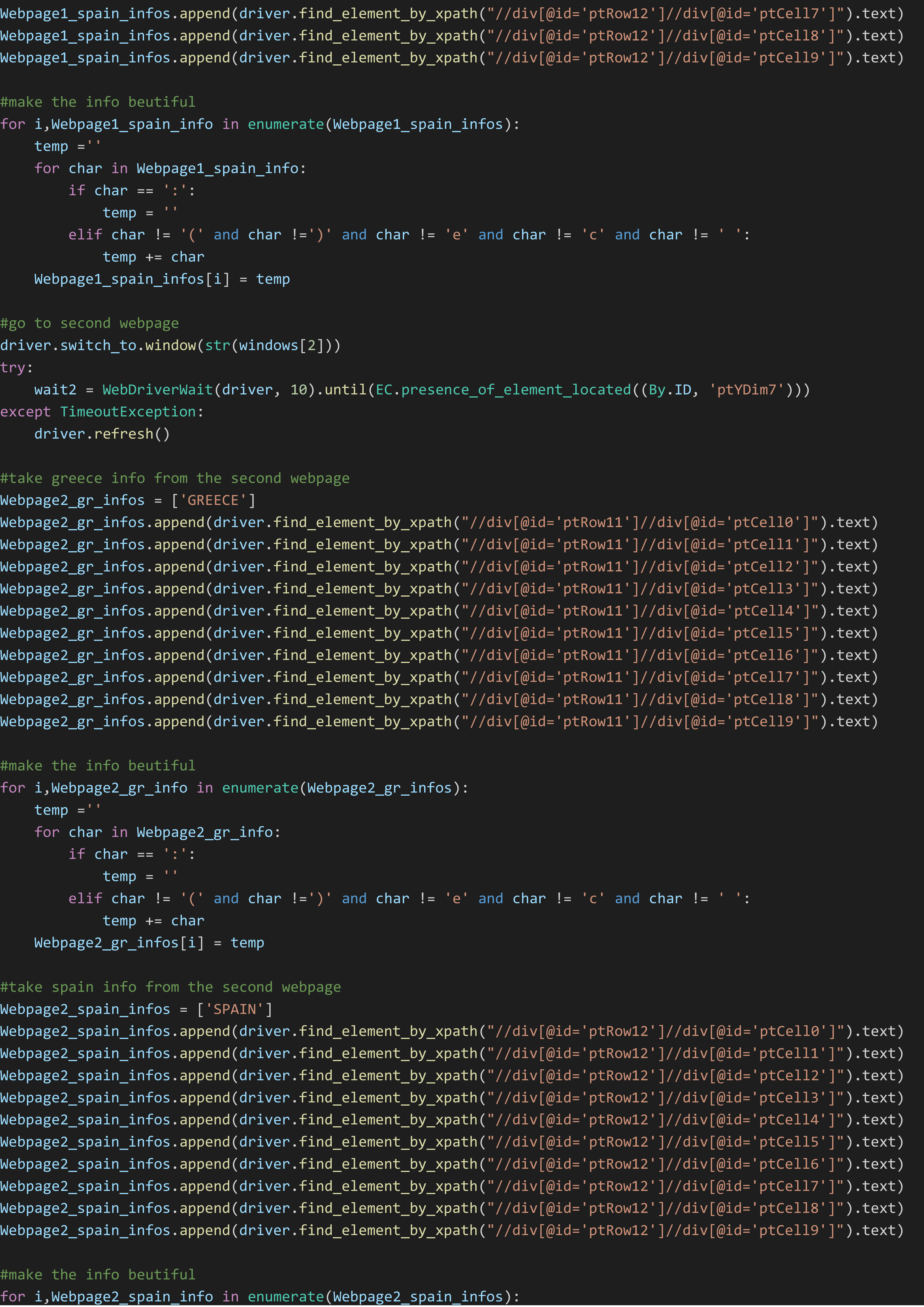
$1^{ο}$ python αρχείο που ονομάζεται get_info_from_webpage.py:



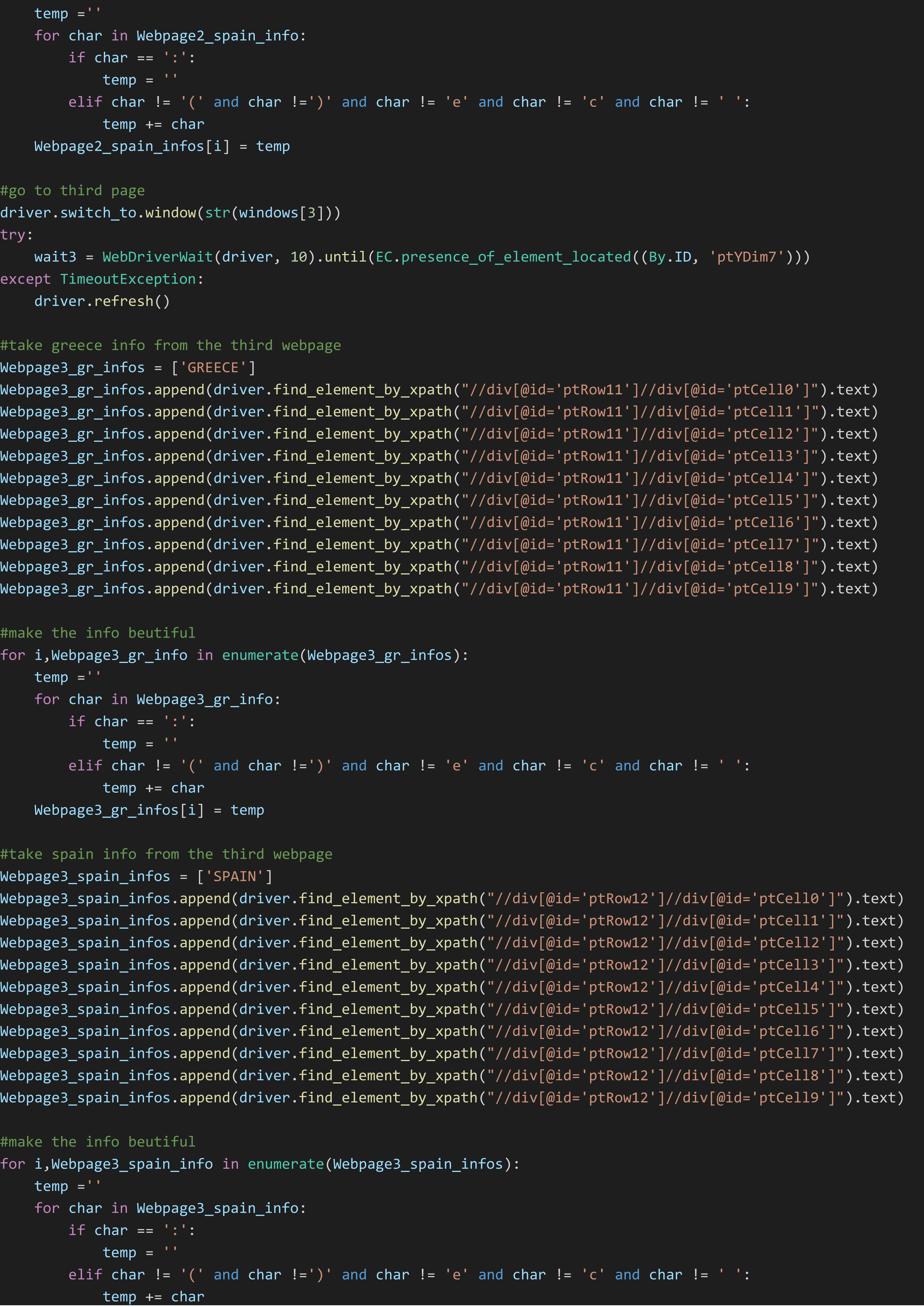
```
from sqlite3.dbapi2 import connect
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.firefox.webdriver import WebDriver
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By
from selenium.common.exceptions import TimeoutException
import os
import sqlite3 as sql

#Create instance of Chrome Driver
if str(os.name) == 'nt':
    chrome_drive_path = 'C:\Program Files (x86)\chromedriver.exe'
elif str(os.name) == 'posix':
    chrome_drive_path = '/home/brainlesspomo/chromedriver'

url = 'https://ec.europa.eu/eurostat/web/tourism/data/database'

#setting window size, position & opening the website
driver = webdriver.Chrome(chrome_drive_path)
driver.set_window_size(1600,900, windowHandle='current')
driver.set_window_position(1920, 0, windowHandle='current')
driver.get(url)


#Find the first folder of the db
#I have to open it so I reveal the files I need
first_fold = driver.find_element_by_class_name('title2')
first_fold.click()

#I need a helper (it is a footer logo, non clickable)
#Every time I open a page, there is text hiding the next link
helper = driver.find_element_by_id('footer_logos')
main_window = driver.current_window_handle
```



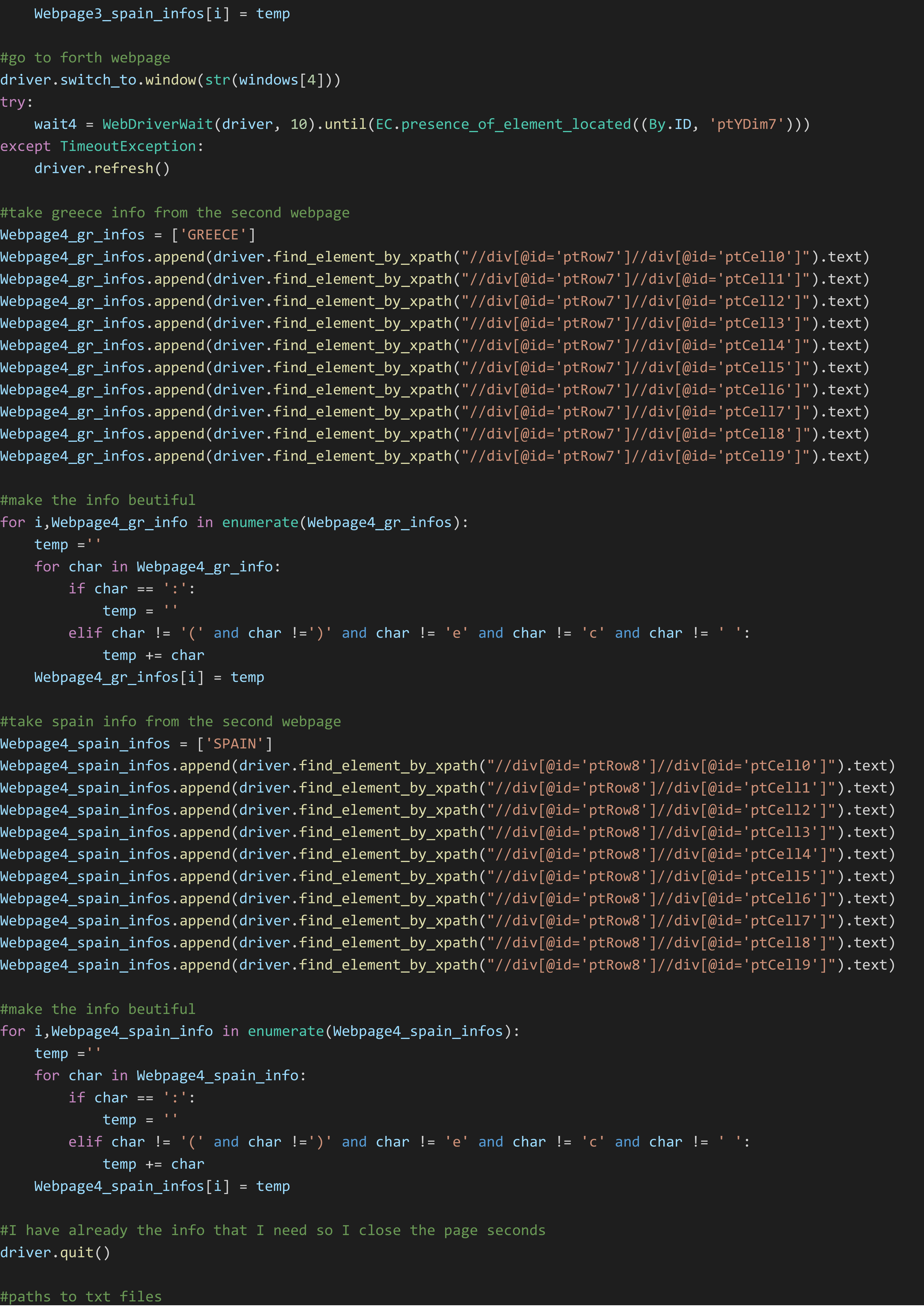
```python
#Getting the link I want to open, it is the only link revealed with that class name
links = driver.find_elements_by_class_name('estat-icon-nui')

#opening the link except the last one, that I don't want
for idx, link in enumerate(links):
    helper.click()
    #den xreiazomai to teleutaio link me auto to onoma, ara den to anoigw
    if link == links[4]:
        break
    else:
        link.click()
    driver.switch_to_window(str(main_window))

    print('Finished with the ' + str(idx+1) + ' click, it openned the ' + str(link) + ' element\n')

#I am getting the window handles for future reference
# and closing the first window as I don't need it from now on
windows = driver.window_handles
driver.close()

driver.switch_to.window(str(windows[1]))

try:
    wait1 = WebDriverWait(driver, 10).until(EC.presence_of_element_located((By.ID, 'ptYDim7')))
except TimeoutException:
    driver.refresh()

#take greece info from the first webpage
Webpage1_gr_infos = ['GREECE']
Webpage1_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow11']//div[@id='ptCell0']").text)
Webpage1_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow11']//div[@id='ptCell1']").text)
Webpage1_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow11']//div[@id='ptCell2']").text)
Webpage1_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow11']//div[@id='ptCell3']").text)
Webpage1_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow11']//div[@id='ptCell4']").text)
Webpage1_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow11']//div[@id='ptCell5']").text)
Webpage1_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow11']//div[@id='ptCell6']").text)
Webpage1_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow11']//div[@id='ptCell7']").text)
Webpage1_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow11']//div[@id='ptCell8']").text)
Webpage1_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow11']//div[@id='ptCell9']").text)

#make the info beutiful
for i,Webpage1_gr_info in enumerate(Webpage1_gr_infos):
    temp =''
    for char in Webpage1_gr_info:
        if char == ':':
            temp = ''
        elif char != '(' and char !=')' and char != 'e' and char != 'c' and char != ' ':
            temp += char
    Webpage1_gr_infos[i] = temp
#take spain info from the first webpage
Webpage1_spain_infos = ['SPAIN']
Webpage1_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow12']//div[@id='ptCell0']").text)
Webpage1_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow12']//div[@id='ptCell1']").text)
Webpage1_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow12']//div[@id='ptCell2']").text)
Webpage1_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow12']//div[@id='ptCell3']").text)
Webpage1_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow12']//div[@id='ptCell4']").text)
Webpage1_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow12']//div[@id='ptCell5']").text)
Webpage1_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow12']//div[@id='ptCell6']").text)
```



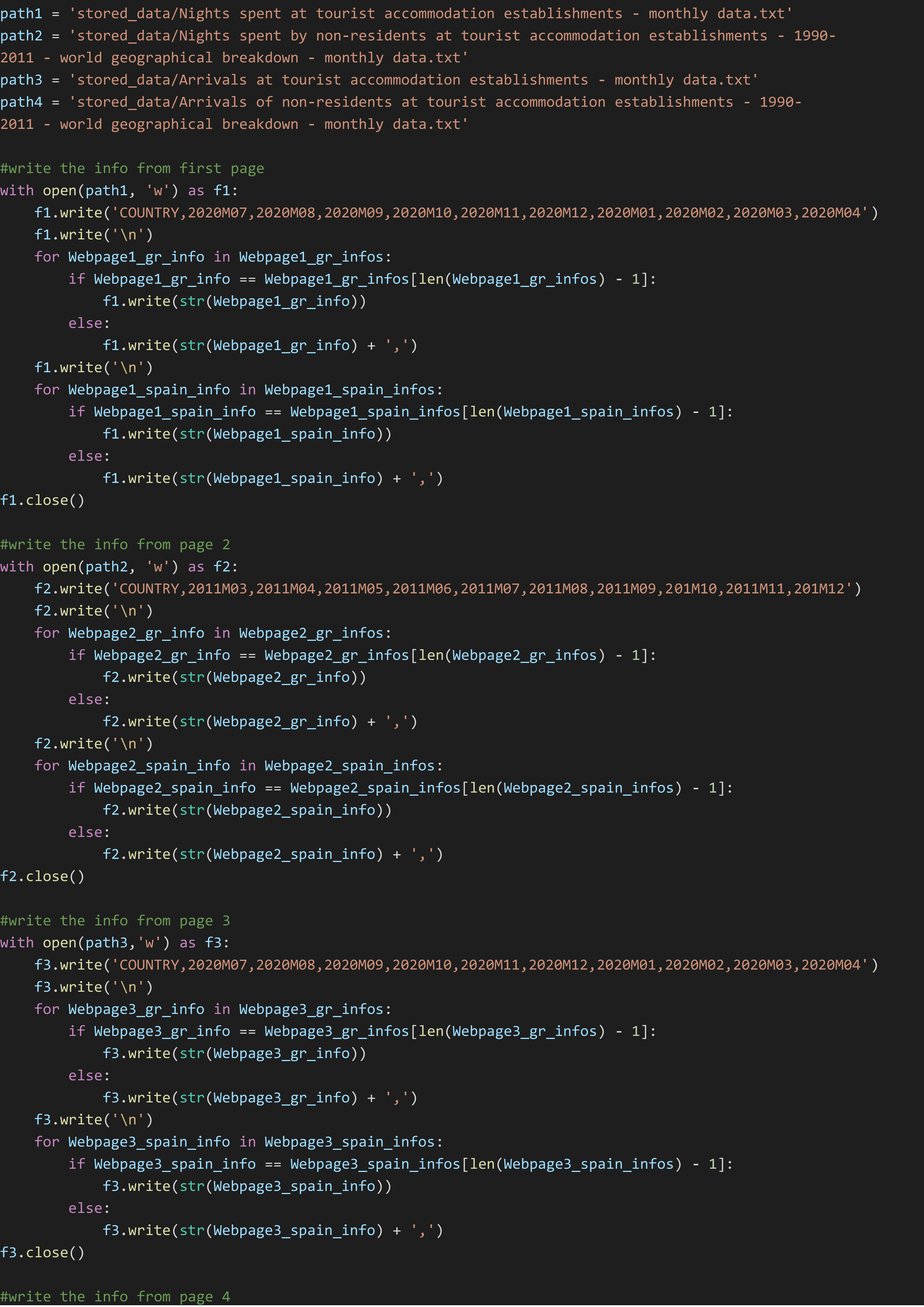
```python
Webpage1_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow12']//div[@id='ptCell7']").text)
Webpage1_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow12']//div[@id='ptCell8']").text)
Webpage1_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow12']//div[@id='ptCell9']").text)

#make the info beutiful
for i,Webpage1_spain_info in enumerate(Webpage1_spain_infos):
    temp =''
    for char in Webpage1_spain_info:
        if char == ':':
            temp = ''
        elif char != '(' and char !=')' and char != 'e' and char != 'c' and char != ' ':
            temp += char
    Webpage1_spain_infos[i] = temp

#go to second webpage
driver.switch_to.window(str(windows[2]))
try:
    wait2 = WebDriverWait(driver, 10).until(EC.presence_of_element_located((By.ID, 'ptYDim7')))
except TimeoutException:
    driver.refresh()

#take greece info from the second webpage
Webpage2_gr_infos = ['GREECE']
Webpage2_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow11']//div[@id='ptCell0']").text)
Webpage2_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow11']//div[@id='ptCell1']").text)
Webpage2_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow11']//div[@id='ptCell2']").text)
Webpage2_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow11']//div[@id='ptCell3']").text)
Webpage2_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow11']//div[@id='ptCell4']").text)
Webpage2_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow11']//div[@id='ptCell5']").text)
Webpage2_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow11']//div[@id='ptCell6']").text)
Webpage2_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow11']//div[@id='ptCell7']").text)
Webpage2_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow11']//div[@id='ptCell8']").text)
Webpage2_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow11']//div[@id='ptCell9']").text)

#make the info beutiful
for i,Webpage2_gr_info in enumerate(Webpage2_gr_infos):
    temp =''
    for char in Webpage2_gr_info:
        if char == ':':
            temp = ''
        elif char != '(' and char !=')' and char != 'e' and char != 'c' and char != ' ':
            temp += char
    Webpage2_gr_infos[i] = temp

#take spain info from the second webpage
Webpage2_spain_infos = ['SPAIN']
Webpage2_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow12']//div[@id='ptCell0']").text)
Webpage2_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow12']//div[@id='ptCell1']").text)
Webpage2_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow12']//div[@id='ptCell2']").text)
Webpage2_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow12']//div[@id='ptCell3']").text)
Webpage2_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow12']//div[@id='ptCell4']").text)
Webpage2_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow12']//div[@id='ptCell5']").text)
Webpage2_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow12']//div[@id='ptCell6']").text)
Webpage2_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow12']//div[@id='ptCell7']").text)
Webpage2_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow12']//div[@id='ptCell8']").text)
Webpage2_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow12']//div[@id='ptCell9']").text)

#make the info beutiful
for i,Webpage2_spain_info in enumerate(Webpage2_spain_infos):
```



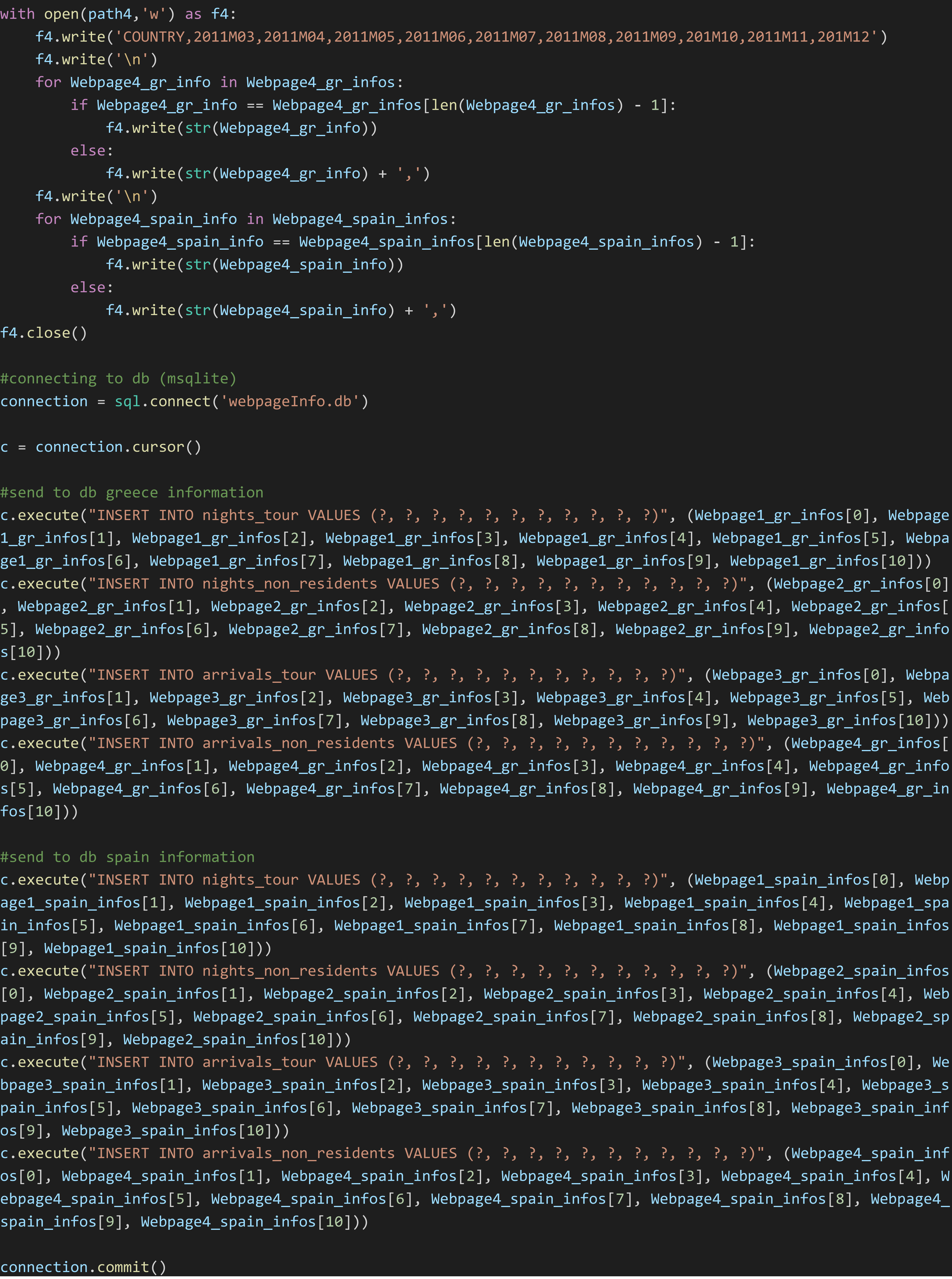
```
    temp =''
    for char in Webpage2_spain_info:
        if char == ':':
            temp = ''
        elif char != '(' and char !=')' and char != 'e' and char != 'c' and char != ' ':
            temp += char
    Webpage2_spain_infos[i] = temp

#go to third page
driver.switch_to.window(str(windows[3]))
try:
    wait3 = WebDriverWait(driver, 10).until(EC.presence_of_element_located((By.ID, 'ptYDim7')))
except TimeoutException:
    driver.refresh()

#take greece info from the third webpage
Webpage3_gr_infos = ['GREECE']
Webpage3_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow11']//div[@id='ptCell0']").text)
Webpage3_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow11']//div[@id='ptCell1']").text)
Webpage3_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow11']//div[@id='ptCell2']").text)
Webpage3_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow11']//div[@id='ptCell3']").text)
Webpage3_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow11']//div[@id='ptCell4']").text)
Webpage3_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow11']//div[@id='ptCell5']").text)
Webpage3_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow11']//div[@id='ptCell6']").text)
Webpage3_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow11']//div[@id='ptCell7']").text)
Webpage3_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow11']//div[@id='ptCell8']").text)
Webpage3_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow11']//div[@id='ptCell9']").text)

#make the info beutiful
for i,Webpage3_gr_info in enumerate(Webpage3_gr_infos):
    temp =''
    for char in Webpage3_gr_info:
        if char == ':':
            temp = ''
        elif char != '(' and char !=')' and char != 'e' and char != 'c' and char != ' ':
            temp += char
    Webpage3_gr_infos[i] = temp

#take spain info from the third webpage
Webpage3_spain_infos = ['SPAIN']
Webpage3_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow12']//div[@id='ptCell0']").text)
Webpage3_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow12']//div[@id='ptCell1']").text)
Webpage3_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow12']//div[@id='ptCell2']").text)
Webpage3_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow12']//div[@id='ptCell3']").text)
Webpage3_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow12']//div[@id='ptCell4']").text)
Webpage3_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow12']//div[@id='ptCell5']").text)
Webpage3_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow12']//div[@id='ptCell6']").text)
Webpage3_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow12']//div[@id='ptCell7']").text)
Webpage3_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow12']//div[@id='ptCell8']").text)
Webpage3_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow12']//div[@id='ptCell9']").text)

#make the info beutiful
for i,Webpage3_spain_info in enumerate(Webpage3_spain_infos):
    temp =''
    for char in Webpage3_spain_info:
        if char == ':':
            temp = ''
        elif char != '(' and char !=')' and char != 'e' and char != 'c' and char != ' ':
            temp += char
```



```python
    Webpage3_spain_infos[i] = temp

#go to forth webpage
driver.switch_to.window(str(windows[4]))
try:
    wait4 = WebDriverWait(driver, 10).until(EC.presence_of_element_located((By.ID, 'ptYDim7')))
except TimeoutException:
    driver.refresh()

#take greece info from the second webpage
Webpage4_gr_infos = ['GREECE']
Webpage4_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow7']//div[@id='ptCell0']").text)
Webpage4_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow7']//div[@id='ptCell1']").text)
Webpage4_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow7']//div[@id='ptCell2']").text)
Webpage4_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow7']//div[@id='ptCell3']").text)
Webpage4_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow7']//div[@id='ptCell4']").text)
Webpage4_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow7']//div[@id='ptCell5']").text)
Webpage4_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow7']//div[@id='ptCell6']").text)
Webpage4_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow7']//div[@id='ptCell7']").text)
Webpage4_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow7']//div[@id='ptCell8']").text)
Webpage4_gr_infos.append(driver.find_element_by_xpath("//div[@id='ptRow7']//div[@id='ptCell9']").text)

#make the info beutiful
for i,Webpage4_gr_info in enumerate(Webpage4_gr_infos):
    temp =''
    for char in Webpage4_gr_info:
        if char == ':':
            temp = ''
        elif char != '(' and char !=')' and char != 'e' and char != 'c' and char != ' ':
            temp += char
    Webpage4_gr_infos[i] = temp

#take spain info from the second webpage
Webpage4_spain_infos = ['SPAIN']
Webpage4_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow8']//div[@id='ptCell0']").text)
Webpage4_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow8']//div[@id='ptCell1']").text)
Webpage4_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow8']//div[@id='ptCell2']").text)
Webpage4_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow8']//div[@id='ptCell3']").text)
Webpage4_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow8']//div[@id='ptCell4']").text)
Webpage4_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow8']//div[@id='ptCell5']").text)
Webpage4_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow8']//div[@id='ptCell6']").text)
Webpage4_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow8']//div[@id='ptCell7']").text)
Webpage4_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow8']//div[@id='ptCell8']").text)
Webpage4_spain_infos.append(driver.find_element_by_xpath("//div[@id='ptRow8']//div[@id='ptCell9']").text)

#make the info beutiful
for i,Webpage4_spain_info in enumerate(Webpage4_spain_infos):
    temp =''
    for char in Webpage4_spain_info:
        if char == ':':
            temp = ''
        elif char != '(' and char !=')' and char != 'e' and char != 'c' and char != ' ':
            temp += char
    Webpage4_spain_infos[i] = temp

#I have already the info that I need so I close the page seconds
driver.quit()

#paths to txt files
```



```
path1 = 'stored_data/Nights spent at tourist accommodation establishments - monthly data.txt'
path2 = 'stored_data/Nights spent by non-residents at tourist accommodation establishments - 1990-
2011 - world geographical breakdown - monthly data.txt'
path3 = 'stored_data/Arrivals at tourist accommodation establishments - monthly data.txt'
path4 = 'stored_data/Arrivals of non-residents at tourist accommodation establishments - 1990-
2011 - world geographical breakdown - monthly data.txt'

#write the info from first page
with open(path1, 'w') as f1:
    f1.write('COUNTRY,2020M07,2020M08,2020M09,2020M10,2020M11,2020M12,2020M01,2020M02,2020M03,2020M04')
    f1.write('\n')
    for Webpage1_gr_info in Webpage1_gr_infos:
        if Webpage1_gr_info == Webpage1_gr_infos[len(Webpage1_gr_infos) - 1]:
            f1.write(str(Webpage1_gr_info))
        else:
            f1.write(str(Webpage1_gr_info) + ',')
    f1.write('\n')
    for Webpage1_spain_info in Webpage1_spain_infos:
        if Webpage1_spain_info == Webpage1_spain_infos[len(Webpage1_spain_infos) - 1]:
            f1.write(str(Webpage1_spain_info))
        else:
            f1.write(str(Webpage1_spain_info) + ',')
f1.close()

#write the info from page 2
with open(path2, 'w') as f2:
    f2.write('COUNTRY,2011M03,2011M04,2011M05,2011M06,2011M07,2011M08,2011M09,201M10,2011M11,201M12')
    f2.write('\n')
    for Webpage2_gr_info in Webpage2_gr_infos:
        if Webpage2_gr_info == Webpage2_gr_infos[len(Webpage2_gr_infos) - 1]:
            f2.write(str(Webpage2_gr_info))
        else:
            f2.write(str(Webpage2_gr_info) + ',')
    f2.write('\n')
    for Webpage2_spain_info in Webpage2_spain_infos:
        if Webpage2_spain_info == Webpage2_spain_infos[len(Webpage2_spain_infos) - 1]:
            f2.write(str(Webpage2_spain_info))
        else:
            f2.write(str(Webpage2_spain_info) + ',')
f2.close()

#write the info from page 3
with open(path3,'w') as f3:
    f3.write('COUNTRY,2020M07,2020M08,2020M09,2020M10,2020M11,2020M12,2020M01,2020M02,2020M03,2020M04')
    f3.write('\n')
    for Webpage3_gr_info in Webpage3_gr_infos:
        if Webpage3_gr_info == Webpage3_gr_infos[len(Webpage3_gr_infos) - 1]:
            f3.write(str(Webpage3_gr_info))
        else:
            f3.write(str(Webpage3_gr_info) + ',')
    f3.write('\n')
    for Webpage3_spain_info in Webpage3_spain_infos:
        if Webpage3_spain_info == Webpage3_spain_infos[len(Webpage3_spain_infos) - 1]:
            f3.write(str(Webpage3_spain_info))
        else:
            f3.write(str(Webpage3_spain_info) + ',')
f3.close()

#write the info from page 4
```

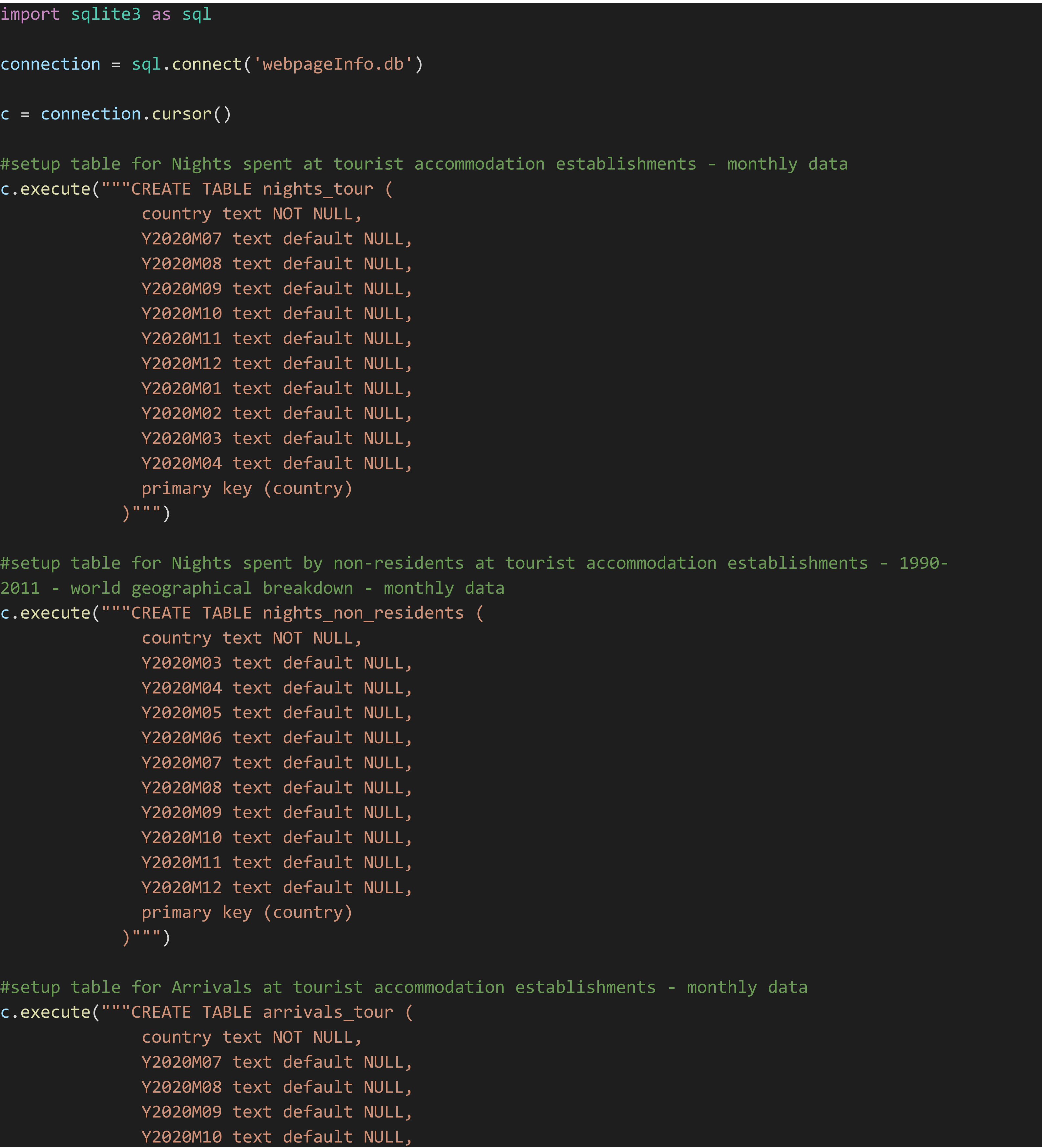


```
with open(path4,'w') as f4:
    f4.write('COUNTRY,2011M03,2011M04,2011M05,2011M06,2011M07,2011M08,2011M09,201M10,2011M11,201M12')
    f4.write('\n')
    for Webpage4_gr_info in Webpage4_gr_infos:
        if Webpage4_gr_info == Webpage4_gr_infos[len(Webpage4_gr_infos) - 1]:
            f4.write(str(Webpage4_gr_info))
        else:
            f4.write(str(Webpage4_gr_info) + ',')
    f4.write('\n')
    for Webpage4_spain_info in Webpage4_spain_infos:
        if Webpage4_spain_info == Webpage4_spain_infos[len(Webpage4_spain_infos) - 1]:
            f4.write(str(Webpage4_spain_info))
        else:
            f4.write(str(Webpage4_spain_info) + ',')
f4.close()

#connecting to db (msqlite)
connection = sql.connect('webpageInfo.db')

c = connection.cursor()

#send to db greece information
c.execute("INSERT INTO nights_tour VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)", (Webpage1_gr_infos[0], Webpage1_gr_infos[1], Webpage1_gr_infos[2], Webpage1_gr_infos[3], Webpage1_gr_infos[4], Webpage1_gr_infos[5], Webpage1_gr_infos[6], Webpage1_gr_infos[7], Webpage1_gr_infos[8], Webpage1_gr_infos[9], Webpage1_gr_infos[10]))
c.execute("INSERT INTO nights_non_residents VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)", (Webpage2_gr_infos[0], Webpage2_gr_infos[1], Webpage2_gr_infos[2], Webpage2_gr_infos[3], Webpage2_gr_infos[4], Webpage2_gr_infos[5], Webpage2_gr_infos[6], Webpage2_gr_infos[7], Webpage2_gr_infos[8], Webpage2_gr_infos[9], Webpage2_gr_infos[10]))
c.execute("INSERT INTO arrivals_tour VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)", (Webpage3_gr_infos[0], Webpage3_gr_infos[1], Webpage3_gr_infos[2], Webpage3_gr_infos[3], Webpage3_gr_infos[4], Webpage3_gr_infos[5], Webpage3_gr_infos[6], Webpage3_gr_infos[7], Webpage3_gr_infos[8], Webpage3_gr_infos[9], Webpage3_gr_infos[10]))
c.execute("INSERT INTO arrivals_non_residents VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)", (Webpage4_gr_infos[0], Webpage4_gr_infos[1], Webpage4_gr_infos[2], Webpage4_gr_infos[3], Webpage4_gr_infos[4], Webpage4_gr_infos[5], Webpage4_gr_infos[6], Webpage4_gr_infos[7], Webpage4_gr_infos[8], Webpage4_gr_infos[9], Webpage4_gr_infos[10]))

#send to db spain information
c.execute("INSERT INTO nights_tour VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)", (Webpage1_spain_infos[0], Webpage1_spain_infos[1], Webpage1_spain_infos[2], Webpage1_spain_infos[3], Webpage1_spain_infos[4], Webpage1_spain_infos[5], Webpage1_spain_infos[6], Webpage1_spain_infos[7], Webpage1_spain_infos[8], Webpage1_spain_infos[9], Webpage1_spain_infos[10]))
c.execute("INSERT INTO nights_non_residents VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)", (Webpage2_spain_infos[0], Webpage2_spain_infos[1], Webpage2_spain_infos[2], Webpage2_spain_infos[3], Webpage2_spain_infos[4], Webpage2_spain_infos[5], Webpage2_spain_infos[6], Webpage2_spain_infos[7], Webpage2_spain_infos[8], Webpage2_spain_infos[9], Webpage2_spain_infos[10]))
c.execute("INSERT INTO arrivals_tour VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)", (Webpage3_spain_infos[0], Webpage3_spain_infos[1], Webpage3_spain_infos[2], Webpage3_spain_infos[3], Webpage3_spain_infos[4], Webpage3_spain_infos[5], Webpage3_spain_infos[6], Webpage3_spain_infos[7], Webpage3_spain_infos[8], Webpage3_spain_infos[9], Webpage3_spain_infos[10]))
c.execute("INSERT INTO arrivals_non_residents VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)", (Webpage4_spain_infos[0], Webpage4_spain_infos[1], Webpage4_spain_infos[2], Webpage4_spain_infos[3], Webpage4_spain_infos[4], Webpage4_spain_infos[5], Webpage4_spain_infos[6], Webpage4_spain_infos[7], Webpage4_spain_infos[8], Webpage4_spain_infos[9], Webpage4_spain_infos[10]))

connection.commit()
```

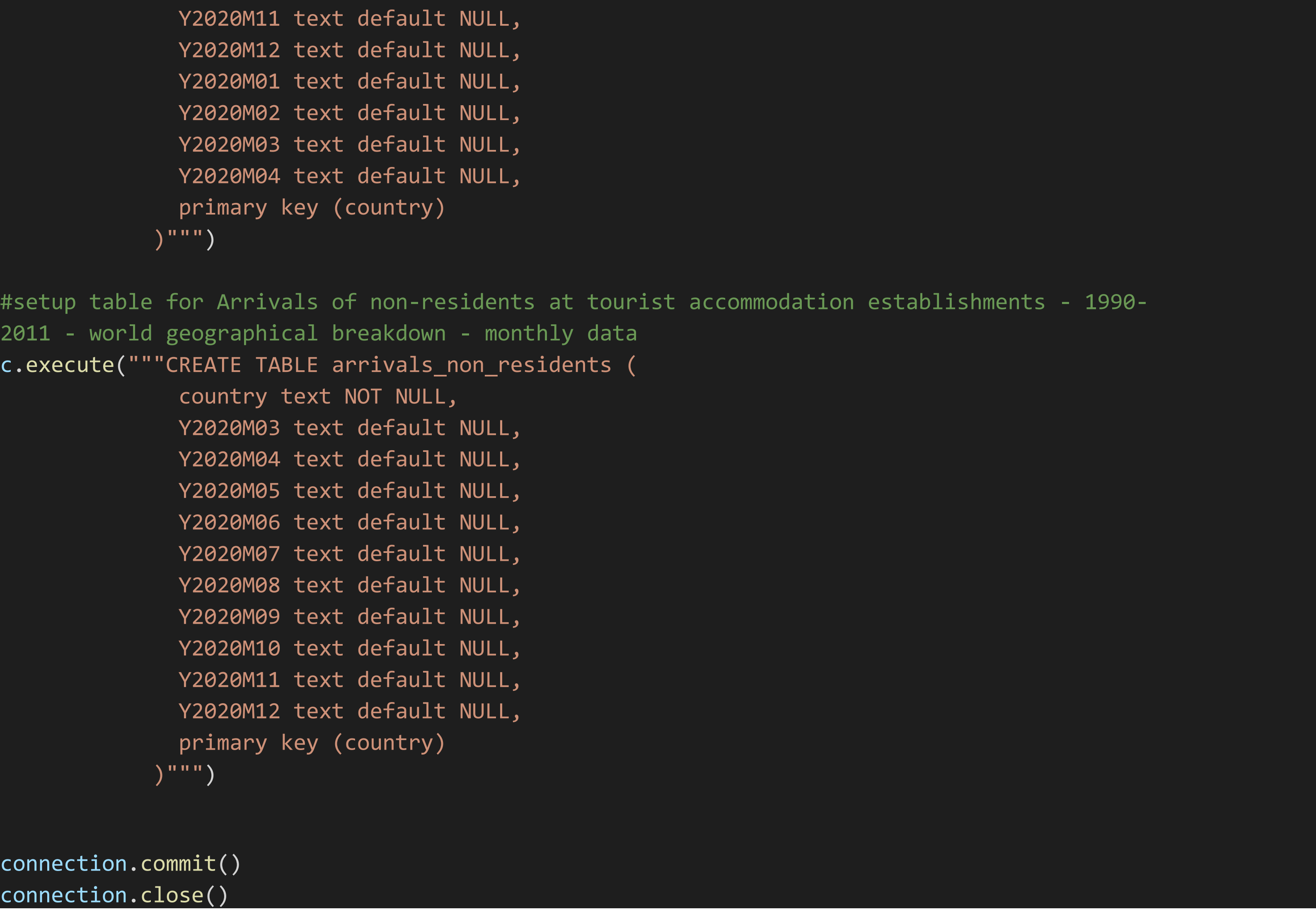
Μετά στο project μου υπάρχει το 2ο python αρχείο που σετάρει το database που χρησιμοποιώ, το οποίο είναι φτιαγμένο σε sqlite και εξ ολοκλήρου σε python, μέσω του sqlite3

Το database μου, ονομάζεται webpageInfo.db

2ο αρχείο που ονομάζεται setup_db.py:



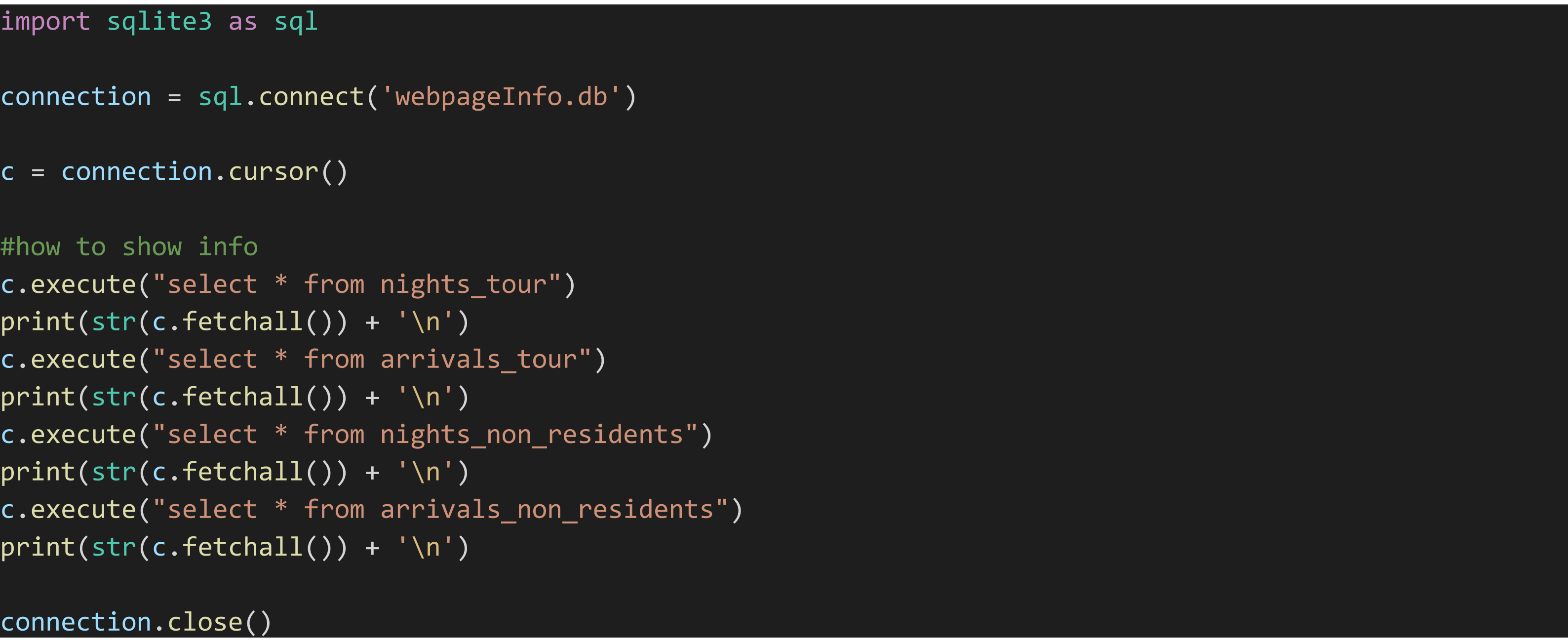
```
import sqlite3 as sql

connection = sql.connect('webpageInfo.db')

c = connection.cursor()

#setup table for Nights spent at tourist accommodation establishments - monthly data
c.execute("""CREATE TABLE nights_tour (
              country text NOT NULL,
              Y2020M07 text default NULL,
              Y2020M08 text default NULL,
              Y2020M09 text default NULL,
              Y2020M10 text default NULL,
              Y2020M11 text default NULL,
              Y2020M12 text default NULL,
              Y2020M01 text default NULL,
              Y2020M02 text default NULL,
              Y2020M03 text default NULL,
              Y2020M04 text default NULL,
              primary key (country)
            )""")

#setup table for Nights spent by non-residents at tourist accommodation establishments - 1990-
2011 - world geographical breakdown - monthly data
c.execute("""CREATE TABLE nights_non_residents (
              country text NOT NULL,
              Y2020M03 text default NULL,
              Y2020M04 text default NULL,
              Y2020M05 text default NULL,
              Y2020M06 text default NULL,
              Y2020M07 text default NULL,
              Y2020M08 text default NULL,
              Y2020M09 text default NULL,
              Y2020M10 text default NULL,
              Y2020M11 text default NULL,
              Y2020M12 text default NULL,
              primary key (country)
            )""")

#setup table for Arrivals at tourist accommodation establishments - monthly data
c.execute("""CREATE TABLE arrivals_tour (
              country text NOT NULL,
              Y2020M07 text default NULL,
              Y2020M08 text default NULL,
              Y2020M09 text default NULL,
              Y2020M10 text default NULL,
```



```
              Y2020M11 text default NULL,
              Y2020M12 text default NULL,
              Y2020M01 text default NULL,
              Y2020M02 text default NULL,
              Y2020M03 text default NULL,
              Y2020M04 text default NULL,
              primary key (country)
            )""")

#setup table for Arrivals of non-residents at tourist accommodation establishments - 1990-2011 - world geographical breakdown - monthly data
c.execute("""CREATE TABLE arrivals_non_residents (
              country text NOT NULL,
              Y2020M03 text default NULL,
              Y2020M04 text default NULL,
              Y2020M05 text default NULL,
              Y2020M06 text default NULL,
              Y2020M07 text default NULL,
              Y2020M08 text default NULL,
              Y2020M09 text default NULL,
              Y2020M10 text default NULL,
              Y2020M11 text default NULL,
              Y2020M12 text default NULL,
              primary key (country)
            )""")


connection.commit()
connection.close()
```

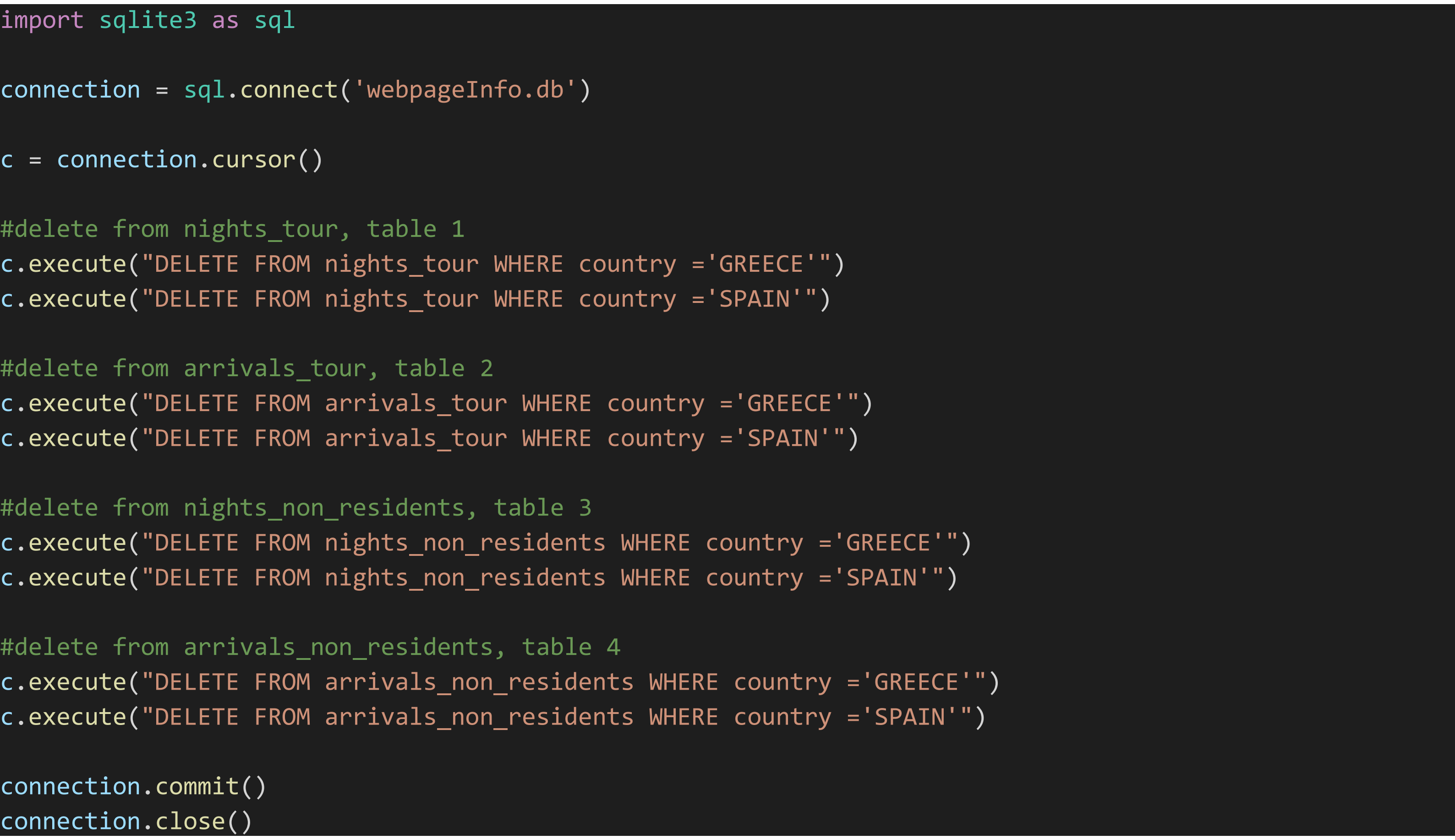
Μετά υπάρχουν άλλα 2 αρχεία των οποίων ο σκοπός τους είναι ο έλεγχος.

Το 3ο μας δίνει τα δεδομένα όλων των tables του database, με όνομα check_db_info.py:



```
import sqlite3 as sql

connection = sql.connect('webpageInfo.db')

c = connection.cursor()

#how to show info
c.execute("select * from nights_tour")
print(str(c.fetchall()) + '\n')
c.execute("select * from arrivals_tour")
print(str(c.fetchall()) + '\n')
c.execute("select * from nights_non_residents")
print(str(c.fetchall()) + '\n')
c.execute("select * from arrivals_non_residents")
print(str(c.fetchall()) + '\n')

connection.close()
```

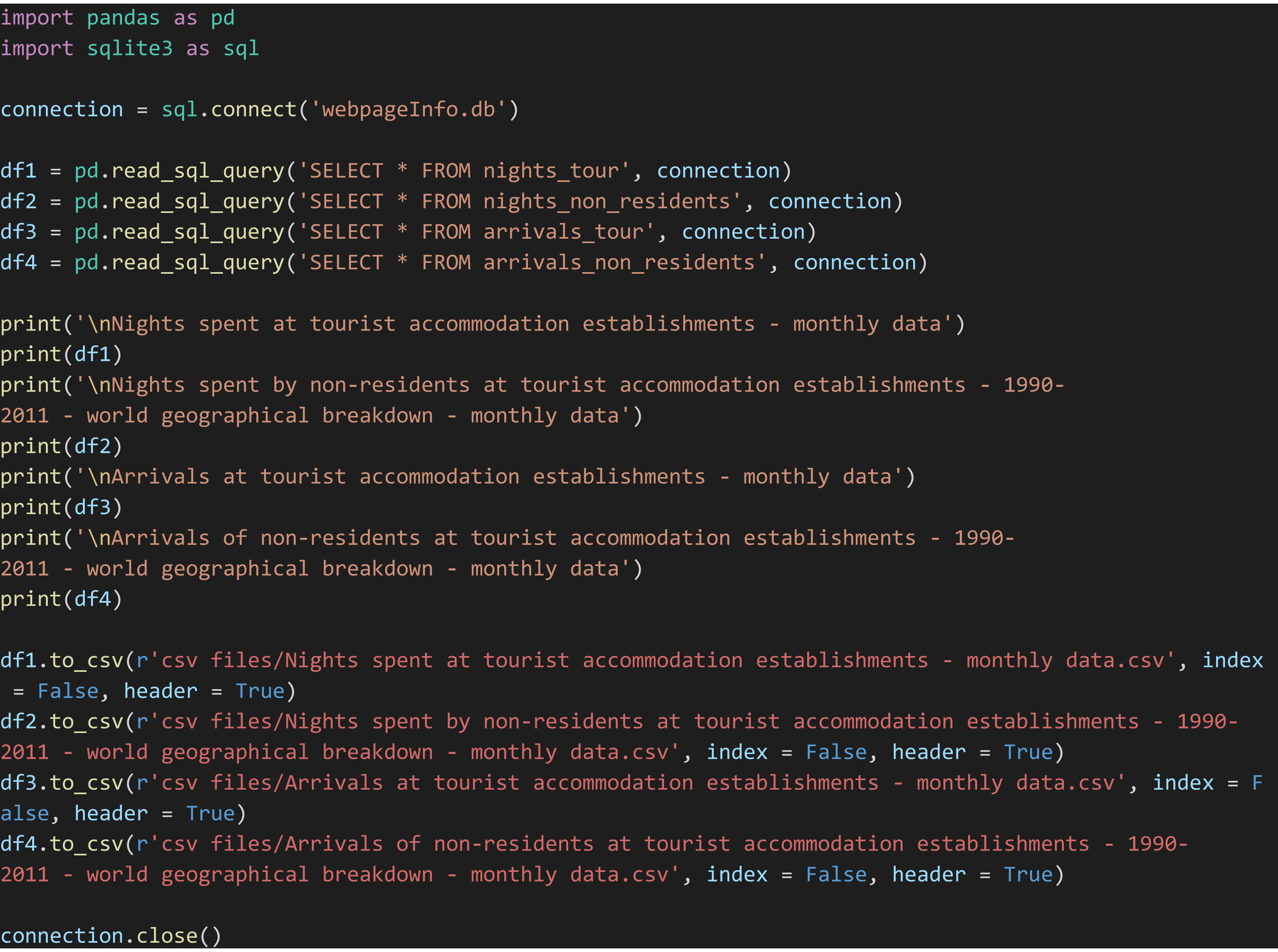
Το 4ο διαγράφει όλα τα δεδομένα από το database, για ελέγξουμε αν μπορεί το πρόγραμμα να τα ξαναπάρει από τα site:



```
import sqlite3 as sql

connection = sql.connect('webpageInfo.db')

c = connection.cursor()

#delete from nights_tour, table 1
c.execute("DELETE FROM nights_tour WHERE country ='GREECE'")
c.execute("DELETE FROM nights_tour WHERE country ='SPAIN'")

#delete from arrivals_tour, table 2
c.execute("DELETE FROM arrivals_tour WHERE country ='GREECE'")
c.execute("DELETE FROM arrivals_tour WHERE country ='SPAIN'")

#delete from nights_non_residents, table 3
c.execute("DELETE FROM nights_non_residents WHERE country ='GREECE'")
c.execute("DELETE FROM nights_non_residents WHERE country ='SPAIN'")

#delete from arrivals_non_residents, table 4
c.execute("DELETE FROM arrivals_non_residents WHERE country ='GREECE'")
c.execute("DELETE FROM arrivals_non_residents WHERE country ='SPAIN'")

connection.commit()
connection.close()
```

Το 5ο και τελευταίο αρχείο, Έχει τίτλο check_info_create_csv.py

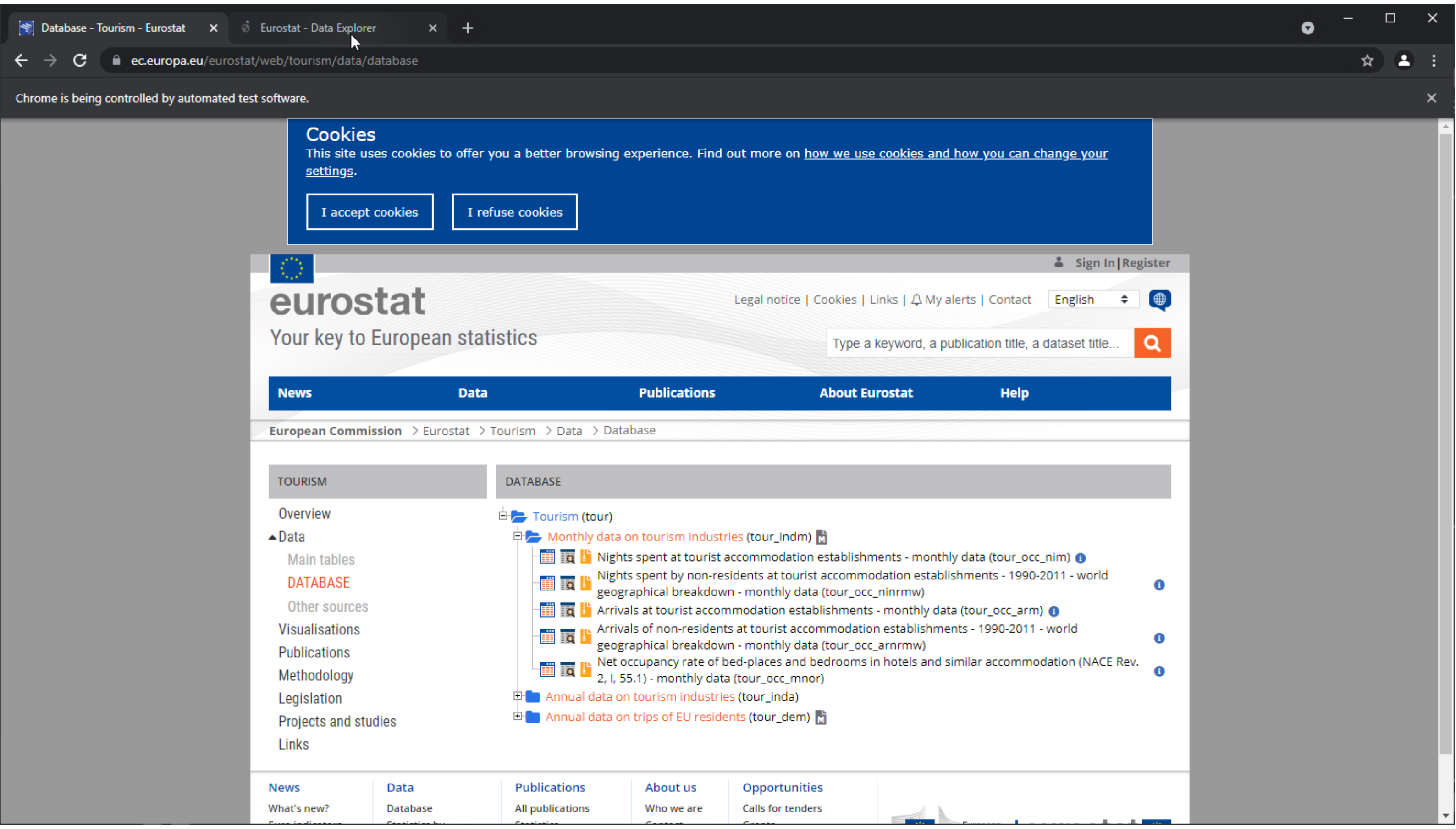
Ο σκοπός του είναι, μέσω του pandas, να παίρνει τα δεδομένα από το database και να τα δείχνει σε ένα dataframe και μετά να φτιάχνει 4 csv αρχεία, με τα δεδομένα του database, τα οποία βρίσκονται στον φάκελο csv files (ανάλογα και τα site από τα οποία πήραμε τα δεδομένα)



```
import pandas as pd
import sqlite3 as sql

connection = sql.connect('webpageInfo.db')

df1 = pd.read_sql_query('SELECT * FROM nights_tour', connection)
df2 = pd.read_sql_query('SELECT * FROM nights_non_residents', connection)
df3 = pd.read_sql_query('SELECT * FROM arrivals_tour', connection)
df4 = pd.read_sql_query('SELECT * FROM arrivals_non_residents', connection)

print('\nNights spent at tourist accommodation establishments - monthly data')
print(df1)
print('\nNights spent by non-residents at tourist accommodation establishments - 1990-2011 - world geographical breakdown - monthly data')
print(df2)
print('\nArrivals at tourist accommodation establishments - monthly data')
print(df3)
print('\nArrivals of non-residents at tourist accommodation establishments - 1990-2011 - world geographical breakdown - monthly data')
print(df4)

df1.to_csv(r'csv files/Nights spent at tourist accommodation establishments - monthly data.csv', index = False, header = True)
df2.to_csv(r'csv files/Nights spent by non-residents at tourist accommodation establishments - 1990-2011 - world geographical breakdown - monthly data.csv', index = False, header = True)
df3.to_csv(r'csv files/Arrivals at tourist accommodation establishments - monthly data.csv', index = False, header = True)
df4.to_csv(r'csv files/Arrivals of non-residents at tourist accommodation establishments - 1990-2011 - world geographical breakdown - monthly data.csv', index = False, header = True)

connection.close()
```

Μπορούμε να παρατηρήσουμε καλύτερα πώς δουλεύει αυτή η εφαρμογή μέσω των παρακάτω screenshots:

Αρχικά, πάει στην σελίδα “ec.europa.eu/eurostat/web/tourism/data/database” και «ανοίγει» τον φάκελο που περιέχει link για τις σελίδες που χρειαζόμαστε



Chrome is being controlled by automated test software.

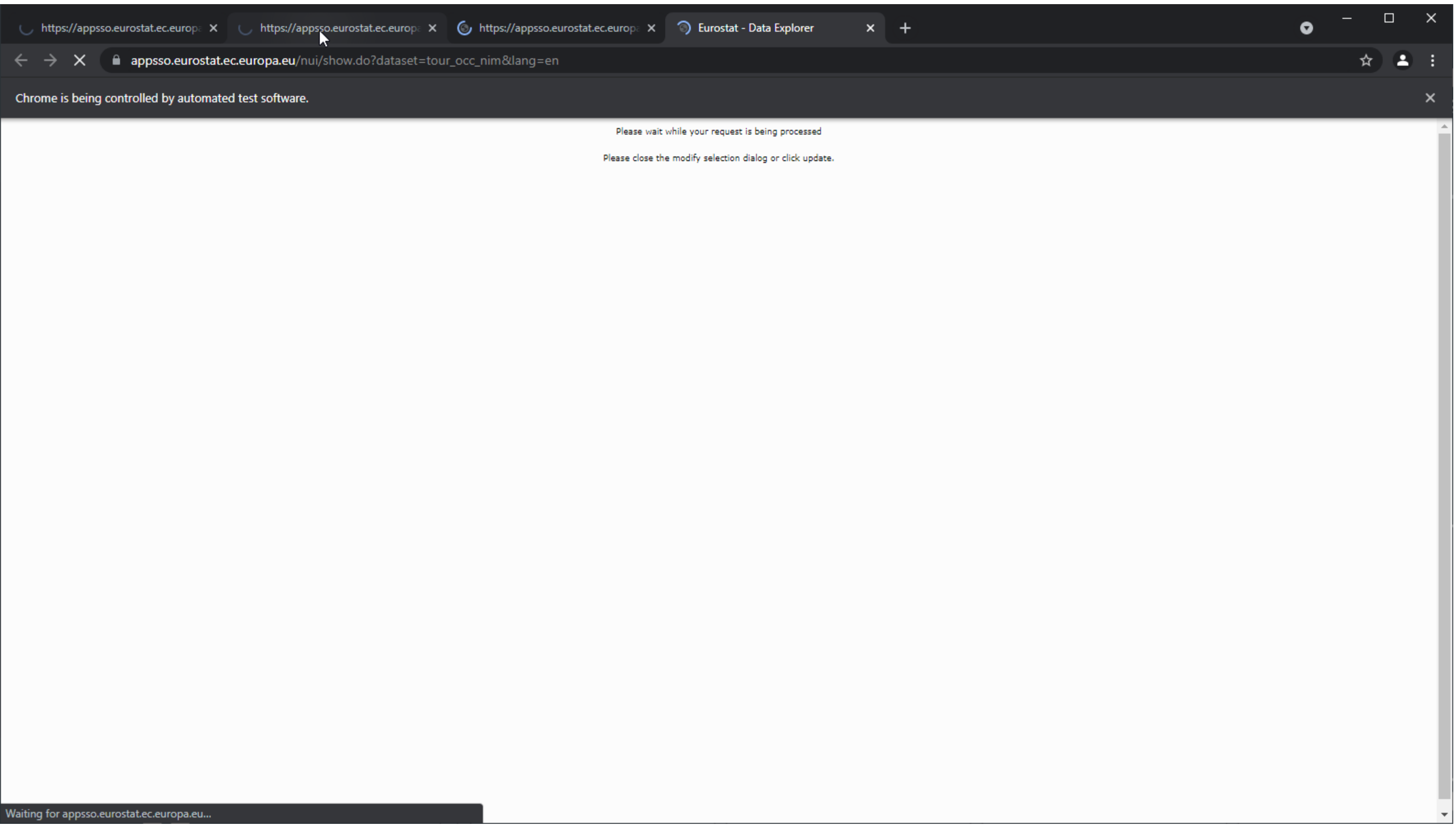
Cookies

This site uses cookies to offer you a better browsing experience. Find out more on how we use cookies and how you can change your settings.

I accept cookies | I refuse cookies

- Tourism (tour)
  - Monthly data on tourism industries (tour_indm)
    - Nights spent at tourist accommodation establishments - monthly data (tour_occ_nim)
    - Nights spent by non-residents at tourist accommodation establishments - 1990-2011 - world geographical breakdown - monthly data (tour_occ_ninrmw)
    - Arrivals at tourist accommodation establishments - monthly data (tour_occ_arm)
    - Arrivals of non-residents at tourist accommodation establishments - 1990-2011 - world geographical breakdown - monthly data (tour_occ_arnrmw)
    - Net occupancy rate of bed-places and bedrooms in hotels and similar accommodation (NACE Rev. 2, I, 55.1) - monthly data (tour_occ_mnor)
  - Annual data on tourism industries (tour_inda)
  - Annual data on trips of EU residents (tour_dem)

Στην συνέχεια, πατάει ένα-ένα τα link και ανοίγει τις σελίδες από πάνω προς τα κάτω εκτός της τελευταίας που δεν θέλουμε καθώς και κλείνει την αρχική αφού δεν την χρειαζόμαστε πλέον:


https://appsso.eurostat.ec.europ
https://appsso.eurostat.ec.europ
https://appsso.eurostat.ec.europ
Eurostat - Data Explorer
appsso.eurostat.ec.europa.eu/nui/show.do?dataset=tour_occ_nim&lang=en

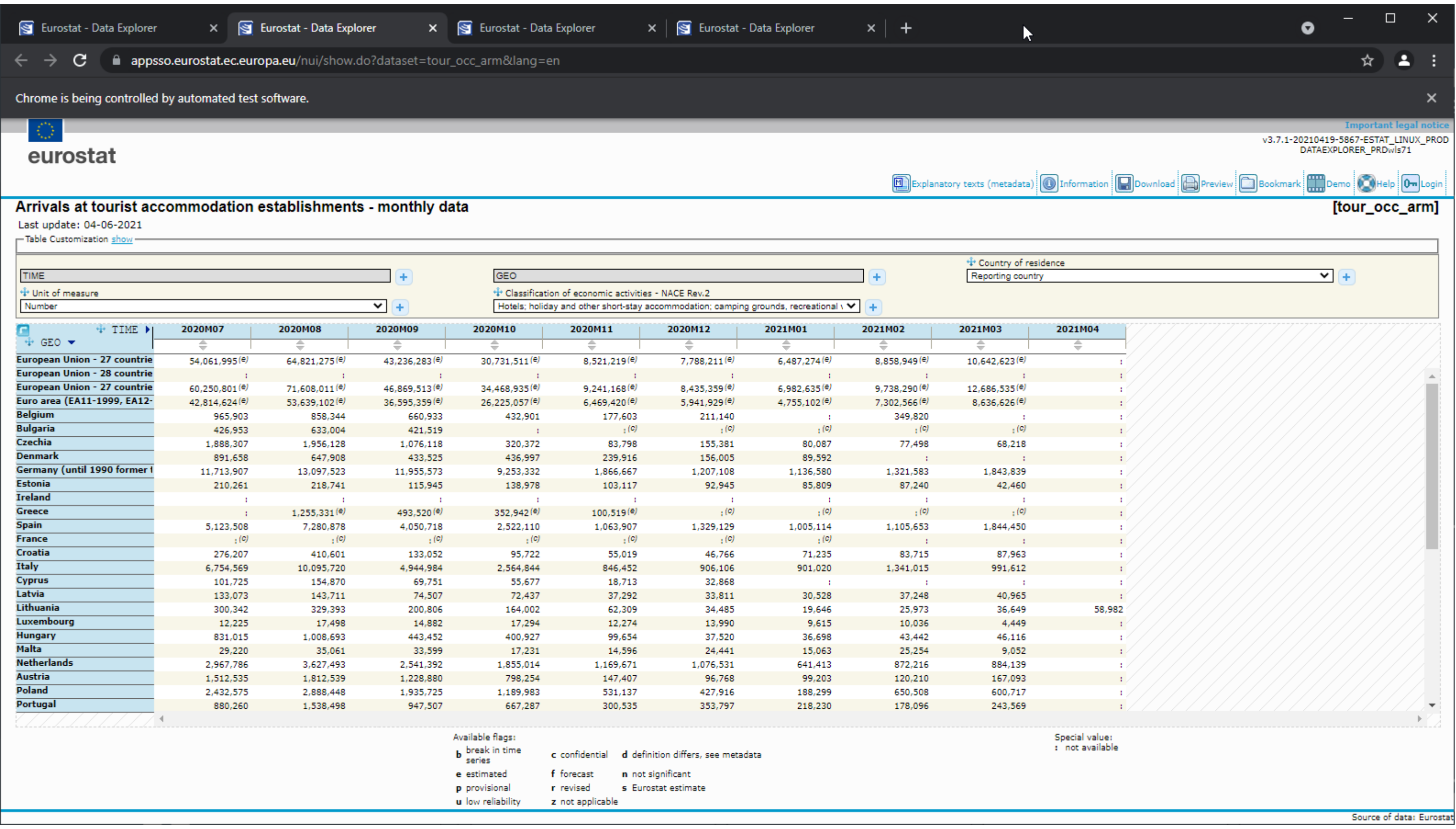
Chrome is being controlled by automated test software.

Please wait while your request is being processed

Please close the modify selection dialog or click update.

Waiting for appsso.eurostat.ec.europa.eu...

Μετά, πάει στις σελίδες από δεξιά προς τα αριστερά (έτσι όπως και ανοίχτηκαν), για να πάρει τα δεδομένα τις Ελλάδας και της Ισπανίας, τα οποία αποθηκεύει σε 4 διαφορετικές λίστες



Arrivals at tourist accommodation establishments - monthly data [tour_occ_arm]

Last update: 04-06-2021

Table Customization show

TIME

GEO

Unit of measure: Number

Classification of economic activities - NACE Rev.2: Hotels; holiday and other short-stay accommodation; camping grounds, recreational v

Country of residence: Reporting country

GEO \ TIME	2020M07	2020M08	2020M09	2020M10	2020M11	2020M12	2021M01	2021M02	2021M03	2021M04
European Union - 27 countrie	54,061,995 (e)	64,821,275 (e)	43,236,283 (e)	30,731,511 (e)	8,521,219 (e)	7,788,211 (e)	6,487,274 (e)	8,858,949 (e)	10,642,623 (e)	:
European Union - 28 countrie	:	:	:	:	:	:	:	:	:	:
European Union - 27 countrie	60,250,801 (e)	71,608,011 (e)	46,869,513 (e)	34,468,935 (e)	9,241,168 (e)	8,435,359 (e)	6,982,635 (e)	9,738,290 (e)	12,686,535 (e)	:
Euro area (EA11-1999, EA12-	42,814,624 (e)	53,639,102 (e)	36,595,359 (e)	26,225,057 (e)	6,469,420 (e)	5,941,929 (e)	4,755,102 (e)	7,302,566 (e)	8,636,626 (e)	:
Belgium	965,903	858,344	660,933	432,901	177,603	211,140	:	349,820	:	:
Bulgaria	426,953	633,004	421,519	:	: (c)	: (c)	: (c)	: (c)	: (c)	:
Czechia	1,888,307	1,956,128	1,076,118	320,372	83,798	155,381	80,087	77,498	68,218	:
Denmark	891,658	647,908	433,525	436,997	239,916	156,005	89,592	:	:	:
Germany (until 1990 former t	11,713,907	13,097,523	11,955,573	9,253,332	1,866,667	1,207,108	1,136,580	1,321,583	1,843,839	:
Estonia	210,261	218,741	115,945	138,978	103,117	92,945	85,809	87,240	42,460	:
Ireland	:	:	:	:	:	:	:	:	:	:
Greece	:	1,255,331 (e)	493,520 (e)	352,942 (e)	100,519 (e)	: (c)	: (c)	: (c)	: (c)	:
Spain	5,123,508	7,280,878	4,050,718	2,522,110	1,063,907	1,329,129	1,005,114	1,105,653	1,844,450	:
France	: (c)	: (c)	: (c)	: (c)	: (c)	: (c)	: (c)	:	:	:
Croatia	276,207	410,601	133,052	95,722	55,019	46,766	71,235	83,715	87,963	:
Italy	6,754,569	10,095,720	4,944,984	2,564,844	846,452	906,106	901,020	1,341,015	991,612	:
Cyprus	101,725	154,870	69,751	55,677	18,713	32,868	:	:	:	:
Latvia	133,073	143,711	74,507	72,437	37,292	33,811	30,528	37,248	40,965	:
Lithuania	300,342	329,393	200,806	164,002	62,309	34,485	19,646	25,973	36,649	58,982
Luxembourg	12,225	17,498	14,882	17,294	12,274	13,990	9,615	10,036	4,449	:
Hungary	831,015	1,008,693	443,452	400,927	99,654	37,520	36,698	43,442	46,116	:
Malta	29,220	35,061	33,599	17,231	14,596	24,441	15,063	25,254	9,052	:
Netherlands	2,967,786	3,627,493	2,541,392	1,855,014	1,169,671	1,076,531	641,413	872,216	884,139	:
Austria	1,512,535	1,812,539	1,228,880	798,254	147,407	96,768	99,203	120,210	167,093	:
Poland	2,432,575	2,888,448	1,935,725	1,189,983	531,137	427,916	188,299	650,508	600,717	:
Portugal	880,260	1,538,498	947,507	667,287	300,535	353,797	218,230	178,096	243,569	:

Available flags:

b break in time series	**c** confidential	**d** definition differs, see metadata
e estimated	**f** forecast	**n** not significant
p provisional	**r** revised	**s** Eurostat estimate
u low reliability	**z** not applicable	

Special value:
: not available

Source of data: Eurostat

Στην συνέχεια, κλείνει τον browser, αφού έχει πάρει τα δεδομένα που χρειάζεται και τα μετατρέπει για να φαίνονται πιο ωραία, τα αποθηκεύει στα txt αρχεία που βρίσκονται στον φάκελο stored_data αλλά και στο database (που είναι το ζητούμενο).