|  |  |
| --- | --- |
|  | **Πανεπιστήμιο Αιγαίου**  **Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων** |

**321 - 4120 – Προηγμένα Θέματα Γλωσσών Προγραμματισμού**

Διδάσκων: Χαραλαμπίδης Ιωάννης

Εργαστηριακοί Συνεργάτες: Αλέξανδρος Φακής – Μιχάλης Δανούσης

**<Ομαδική Εργαστηριακή Άσκηση>**

|  |  |
| --- | --- |
| Ομάδα:  Κοτσαράπογλου Ιάσων – Ειρηναίος  Χαμακιώτη Ελένη  Κουτσάκης Φίλιππος | Α.Μ:  321/2014092  321/2017212  321/2011077 |
|  |  |

Σάμος, <23/06>, 2022

# Ερώτημα 1

## Δημιουργία κλάσης αναπαράστασης

Δημιουργία κλάσης ταινίας για την αποθήκευση των δεδομένων της.

class **MovieClass**:

    def **\_\_init\_\_**(self, sub\_name, sub\_lang,sub\_creator,times\_dl,created\_at,download\_link):

        self.sub\_name = sub\_name

        self.sub\_lang = sub\_lang

        self.sub\_creator = sub\_creator

        self.times\_dl = times\_dl

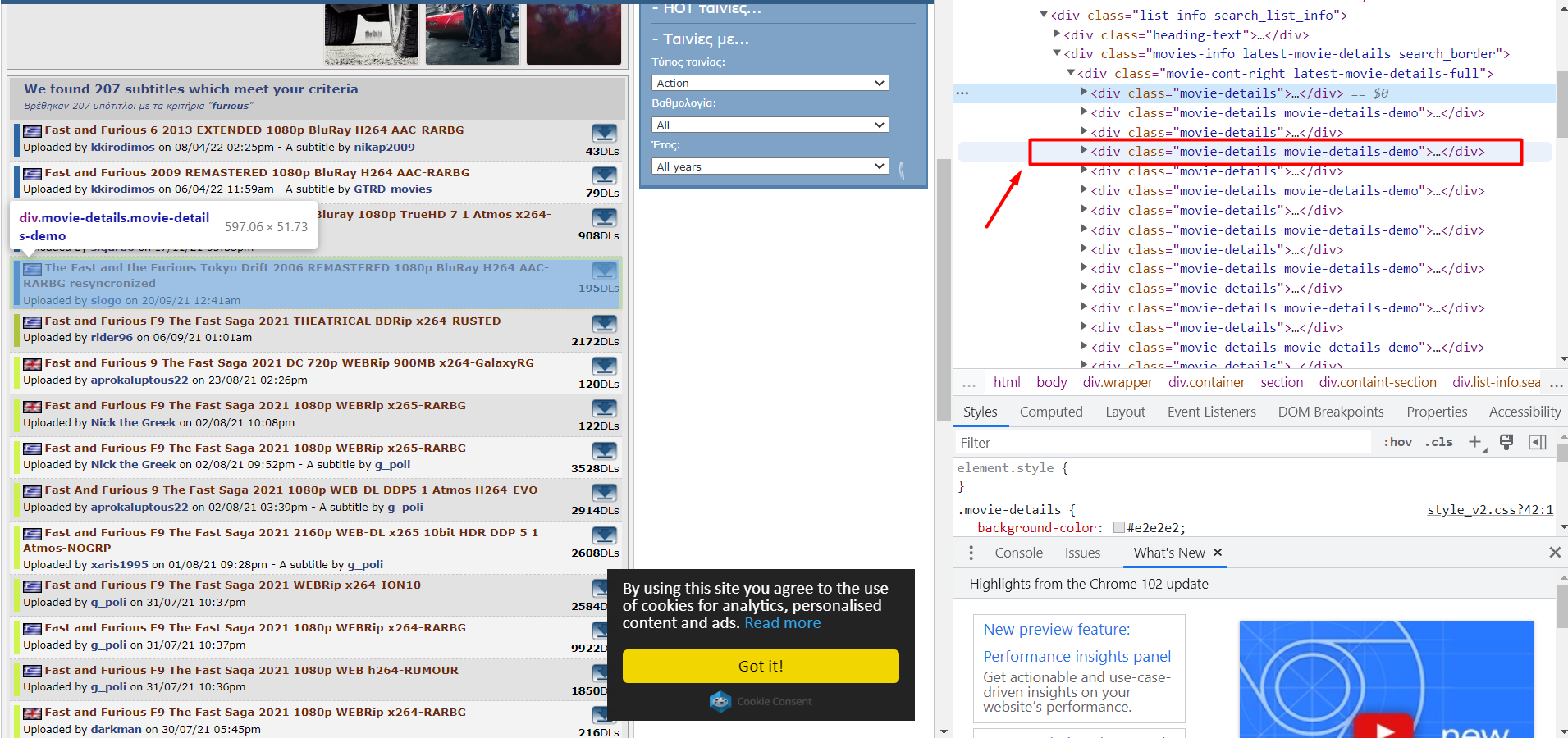
        self.created\_at = created\_at

        self.download\_link = download\_link

## Ανάκτηση των δεδομένων

Για να ανακτήσουμε τα δεδομένα θα πρέπει να ανακαλύψουμε την κλάση η οποία θα πρέπει να μας επιστρέψει την κάθε εγγραφή της λίστας ξεχωριστά. Στη συγκεκριμένη ιστοσελίδα μπορούμε να ανακτήσουμε όλα τα πεδία τα οποία υπάρχουν ανάμεσα στο Movie-details και στο clearfix div το οποίο είναι το τέλος του κάθε div.

entries\_list\_regex = '<div class="movie-details(.\*?)<div class="clearfix">'



Αντίστοιχα δημιουργούμε άλλα regular expressions για τα υπόλοιπα στοιχεία:

sub\_name\_regex = '<span>(.\*?)</span>'

    sub\_title\_lang\_regex = 'class="sprite (\w+)gif"><\/div><span>'

    sub\_user\_creator\_regex = 'title="Subtitles uploaded by .\*?">([\w\_-]\*?)<\/a>'

    times\_dl\_regex = '<\/a><p><b>(\d+)<\/b>DLs'

    created\_at\_regex = 'on (\d+\/\d+\/\d+ \d+:\d+[apm]{2})(<\/p><\/div>)?'

    download\_link\_regex = '<div class="movie-download"><a href="(\/.\*?)" title="'

Τα στοιχεία που επιλέγουμε είναι:

* + **Όνομα υπότιτλου**
  + **Γλώσσα υπότιτλου**
  + **Δημιουργος**
  + **Ημερομηνία**
  + **Αριθμός download**
  + **URL του υπότιτλου**

Δημιουργία του τελικού dictionary για την αποθήκευση των δεδομένων σε αντίστοιχο Dataframe στον κώδικα μας.

final\_dictionary = {'created\_at':[],'download\_link':[], 'genre':[], 'no\_reviews':[], 'rating':[], 'sub\_creator':[], 'sub\_lang':[],'sub\_name':[], 'times\_dl':[]}

Δημιουργία αντίστοιχου DataFrame από το παραπάνω dictionary.

global final\_dictionary

df1 = **pd**.**DataFrame**(final\_dictionary)

Μετά την δημιουργία του αντικειμένου Ταινίας με όλα τα στοιχεία από το 1ο και το 2ο επίπεδο θα πρέπει να μετατρέψουμε το αντικείμενο σε dictionary και να το προσθέσουμε στο αρχικό dataframe που δημιουργήσαμε παραπάνω.

*#metatropi se dictionary*

movie = **jsons**.**dump**(movie)

df2 = **pd**.**DataFrame**([movie], index = [0])

*#prosthiki tou neou DataFrame στο*

df1 = df1.**append**(df2, ignore\_index = True)

**print** (df1)

Δημιουργία του τελικού DataFrame και εγγραφή του στο Excel.

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

# Ερώτημα 2

Ποσοστά κελιών ανά στήλη τα οποία εμπεριέχουν κενές τιμές.

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Εμφάνιση των κελιών τα οποία έχουν συμπληρωθεί με τιμές οι οποίες έχουνε μέσο όρο τις μη μηδενικές τιμές της ίδιας στήλης.

Εικόνα που περιέχει πίνακας

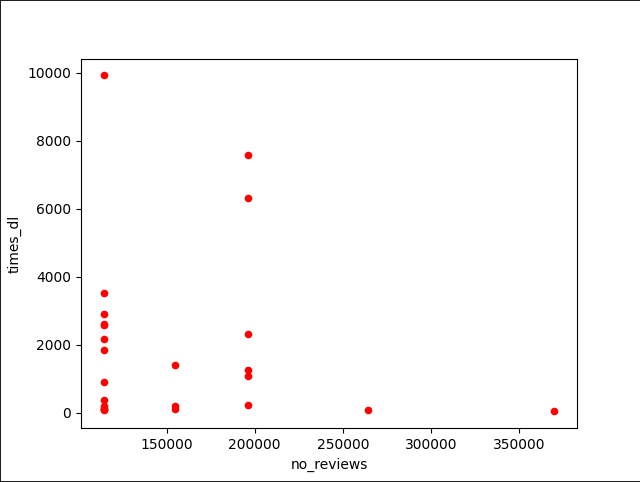
Περιγραφή που δημιουργήθηκε αυτόματα

Εμφάνιση του τελικού excel όπως προέκυψε από την Άσκηση 1.

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Εμφάνιση του γραφήματος scatterplot με τον αριθμό των reviews σε σχέση με τον αριθμό download που έχει ο κάθε υπότιτλος.



# Ερώτημα 3

Στο ερώτημα 3 επιλέξαμε το selenium για την απόκτηση των στοιχείων κάθε ιστοσελίδας. Με την χρήση των CSS Selectors μπορούμε να επιλέξουμε την κλάση που μας ενδιαφέρει και να πάρουμε με την χρήση του child element (>) το αντίστοιχο παιδί.

 sub\_title\_name = entry.find\_element(**By**.CSS\_SELECTOR, ".movie-heading > span")

**print** (sub\_title\_name.text)

        sub\_title\_lang = entry.find\_element(**By**.CSS\_SELECTOR, '.movie-heading > div')

        lang = sub\_title\_lang.get\_attribute('class')[:-3]

**print** (lang[-2:])

        lang = lang[-2:]

        sub\_user\_creator = entry.find\_element(**By**.CSS\_SELECTOR, '.movie-info > p > a')

**print** (sub\_user\_creator.text)

        times\_dl = entry.find\_element(**By**.CSS\_SELECTOR, '.movie-download > p > b')

**print** (times\_dl.text)

        created\_at = entry.find\_element(**By**.CSS\_SELECTOR, '.movie-info > p')

**print** (created\_at.text)

        download\_link = entry.find\_element(**By**.CSS\_SELECTOR, '.movie-heading')

        download\_link = download\_link.get\_attribute('href')

Εικόνα που περιέχει κείμενο, πίνακας

Περιγραφή που δημιουργήθηκε αυτόματα