Μεταγλωτιστές 2020

Προγραμματιστική Εργασία #2

Ονοματεπώνυμο: Μαθιουδάκης Εμμανουήλ

ΑΜ: Π2017027

- Τα βήματα επεξεργασίας στον κώδικα μου είναι αυστηρά βασισμένα στην αρίθμηση των ζητουμένων της εργασίας, όπως μπορούμε να δούμε και από τα σχόλια του. Συνοπτικά, έχουμε αρχικά την εισαγωγή της βιβλιοθήκης που περιέχει τα «εργαλεία» που θα χρειαστούμε, έπειτα την δήλωση μιας "callback"συνάρτησης που θα μας χρειαστεί για την επίλυση του 6^{ου} ζητούμενου και μετά αριθμημένα τα βήματα επεξεργασίας (τα ονόματα των μεταβλητών καθώς και τα σχόλια δηλώνουν τον αριθμό και τις λεπτομέρειες του ζητουμένου αντίστοιχα). Εκτυπώσεις του επεξεργασμένου κειμένου έγιναν μόνο σε σημεία που καθόριζε η εργασία και με τον τρόπο που αυτή ζητούσε.

- Οι περιγραφές των κανονικών εκφράσεων που χρησιμοποιήθηκαν είναι οι εξής:

    1. (r'<title>(.+?)</title>') --- Εφόσον μας ζητείται να εξάγουμε και να εκτυπώσουμε τον τίτλο ο οποίος βρίσκεται μεταξύ των tags <title> και </title>, χρησιμοποιούμε τα ονόματα των tags ώστε να καταλάβει το πρόγραμμα την αρχή και το τέλος της έκφρασης και ενδιάμεσα χρησιμοποιούμε τον συμβολισμό ενός οποιουδήποτε χαρακτήρα εκτός από το newline (.) μαζί με τον συμβολισμό της μη άπληστης μορφής του (+) το οποίο επιτρέπει το «διάβασμα» του χαρακτήρα που προηγείται (δηλαδή του οποιουδήποτε πέρα απο newline) μια ή περισσότερες φορές. Αυτό είναι κλεισμένο σε παρένθεση διότι η εργασία μας ζητάει την εκτύπωση του τίτλου, κάτι που επιτυγχάνεται με την χρήση του m.group(1) το οποίο είναι αυτή η παρένθεση.

    2. (r'<!--.*?-->',re.DOTALL) --- Στο συγκεκριμένο βήμα μας ζητείται η απαλοιφή των σχολίων τα οποία είναι δηλωμένα ανάμεσα στα <!-

-και -->. Για αυτό το λόγο χρησιμοποιούνται αυτά τα σύμβολα ώστε το πρόγραμμα να καταλάβει την αρχή και το τέλος της έκφρασης που αναζητάμε. Ενδιάμεσα έχουμε ξανά τον συμβολισμό οποιουδήποτε χαρακτήρα (.) μαζί με την μη άπληστη μορφή του (*) το οποίο επιτρέπει το διάβασμα του προηγούμενου χαρακτήρα 0 ή περισσότερες φορές. Αυτό όπως και πριν χρησιμοποιείται για να «διαβάσει» το πρόγραμμα τις λέξεις που αποτελούν τα σχόλια. Επειδή όμως, όπως δηλώσαμε πριν, ο συμβολισμός οποιουδήποτε χαρακτήρα διαβάζει τα πάντα πέρα από το newline, χρησιμοποιούμε το re.DOTALL που επιτρέπει την παράβλεψη αυτής της ιδιαιτερότητας ώστε το πρόγραμμα να μπορεί να διαβάσει σχόλια που πιάνουν περισσότερες από μια γραμμές κειμένου. Στη συνέχεια, με την χρήση της έκφρασης αυτής χρησιμοποιούμε την μέθοδο sub() με την οποία αντικαθιστούμε ολόκληρα τα σχόλια με το κενό.

3. (r'<(script|style).*?>.*?</(script|style)>',re.DOTALL) --- Στο ζητούμενο αυτό πρέπει να απαλοίψουμε τα <script> και <style> tags μαζί με όλο τους το περιεχόμενο. Αρχικά δηλώνουμε το πρώτο αναγνωριστικό tag το περιεχόμενο του οποίου μπορεί να είναι είτε script είτε style, για αυτό και γίνεται χρήση του (|) το οποίο σημαίνει είτε το ένα είτε το άλλο. Επίσης, επειδή πριν το κλείσιμο του πρώτου tag μπορούν να πάρουν τα συγκεκριμένα διάφορα attributes χρησιμοποιούμε τον γνωστό (από τα προηγούμενα ζητούμενα) συνδυασμό αναγνώρισης λέξεων (.*?) μια φορά πριν το κλείσιμο του tag και μια μετά ώστε να συμπεριλάβουμε και τα περιεχόμενά του. Έπειτα, δηλώνουμε το κλείσιμο του tag με τον ίδιο τρόπο όπως πριν ώστε να καταλαβαίνει το πρόγραμμα που να σταματήσει. Γίνεται πάλι χρήση του re.DOTALL διότι μπορεί να είναι περισσότερα από ενα lines το κάθε tag. Τέλος, ολόκληρη η έκφραση χρησιμοποιείται με την μέθοδο sub() ώστε να τα αντικαταστήσουμε με το κενό (δηλαδή να τα απαλοίψουμε).

4. (r'<a.*?href="(.+?)">(.*?)',re.DOTALL) --- Εδώ επειδή μας ζητείται η εξαγωγή και εκτύπωση των συνδέσμων που βρίσκονται μέσα στο <a> tag (ως attribute) και του κειμένου τους δηλώνουμε

την αρχή της έκφρασης (<a), μετά χρησιμοποιούμε τον συνδυασμό αναγνώρισης λέξεων καθώς μπορεί να υπάρχουν διαφορετικά attributes πρίν από το href που μας ενδιαφέρει με την χρήση του (.*?), έπειτα δηλώνουμε το href=”(.+?)” ώστε να δηλώσουμε συγκεκριμένα το μέρος που βρίσκεται αυτό που ψάχνουμε και μετά το κλείσιμο του tag (>) έχουμε ακόμη μια φορά τον συνδιασμό για την αναγνώριση λέξεων ώστε να κρατήσουμε το κείμενο ανάμεσα από τα tags. Τέλος τοποθετούμε το κλείσιμο του tag και το re.DOTALL για την αναγνώριση πολλαπλών lines κώδικα. Το τελικό αποτέλεσμα επιτυγχάνεται με την χρήση των παρενθέσεων, οι οποίες σε συνδυασμό με τα m.group(1) και m.group(2) εμφανίζουμε μόνο τον σύνδεσμο και το κείμενο που βρίσκεται ενδιάμεσα στα tags.

5. (r'<[^>]+>',re.DOTALL) --- Το πέμπτο ζητούμενο ήταν η απαλοιφή όλων των tags. Αυτό επιτεύχθηκε με την χρήση μιας έκφρασης που υπήρχε και στο υλικό του εργαστηρίου που υπάρχει ανεβασμένο. Η έκφραση αυτή χρησιμοποιεί τον χαρακτήρα ανοίγματος του tag (<), στην συνέχεια μια κλάση χαρακτήρων μέσα στην οποία δηλώνει πως ακολουθεί οποιοσδήποτε χαρακτήρας εκτός του (>) μια ή περισσότερες φορές ([^>]+) ώστε να εξασφαλίσει πως συμπεριλαμβάνονται όλα τα περιεχόμενα του tag (attributes) εως ότου φτάσει στον τελικό χαρακτήρα κλεισίματος του επόμενου tag (>). Χρησιμοποιείται επίσης το re.DOTALL καθώς κάποια tags επεκτείνονται σε περισσότερες από μια γραμμές. Τέλος, με την χρήση της μεθόδου sub() γίνεται η απαλοιφή, αντικαθιστώντας όλα τα tags με κενά.

6. (r'&(amp|gt|lt|nbsp);') --- Το έκτο ζητούμενο απαιτούσε την μετατροπή συγκεκριμένων HTML entities στα αντίστοιχα σύμβολά τους με την χρήση μιας callback συνάρτησης. Η έκφραση περιέχει το κοινό για όλα τα entities “&”, έπειτα τα διαφορετικά ονόματα αυτών σε μια παρένθεση με την χρήση του συμβόλου «είτε το ένα είτε το άλλο» (|) ώστε να αναγνωρίζει καθένα από τα 4 διαφορετικά ονόματα των entities που μπορεί να πετύχει στο κείμενο και στην συνέχεια τον κοινό για όλα χαρακτήρα (;). Έπειτα χρησιμοποιείται η μέθοδος sub() με

γνωρίσματα την προαναφερθείσα συνάρτηση και το κείμενο. Η συνάρτηση ελέγχει τι αναγνωρίστηκε από το m.group(1), δηλαδή την παρένθεση με το όνομα που θα προκύψει και ανάλογα επιστρέφει το κατάλληλο σύμβολο.

7. (r'\s+') --- Το συγκεκριμένο ζητούμενο απαιτούσε την μετατροπή συνεχόμενων χαρακτήρων whitespace σε ένα ακριβώς κενό. Η παραπάνω έκφραση που χρησιμοποιήθηκε υπάρχει και στο υλικό του μαθήματος και ουσιαστικά αναγνωρίζει οποιοδήποτε whitespace (\s) μια ή περισσότερες φορές (+). Έπειτα χρησιμοποιείται η μέθοδος sub() για την μετατροπή τους σε ένα ακριβώς κενό.

8. Το συγκεκριμένο ζητούμενο δεν χρειαζόταν κανονική έκφραση.

- Οι μόνες αναφορές που χρησιμοποιήθηκαν ήταν οι προτεινόμενες, δηλαδή το ανεβασμένο υλικό του μαθήματος και θεώρησα πως δεν χρειάζεται να το συμπεριλάβω.