Μεταγλωττιστές 2020

Προγραμματιστική Εργασία #2

Ονοματεπώνυμο: ΚΡΙΣΤΙΑΝ ΛΕΚΑ

Α.Μ: Π2017153

1.Συνοπτικη περιγραφή της σειράς βημάτων επεξεργασίας στον κώδικα μου

Αρχικά έκανα import το module re το οποίο παρέχει τακτικές λειτουργίες αντιστοίχισης έκφρασης. Φρόντισα να ανοίγει το αρχείο μέσω της εντολής text = open('testpage.txt','r').read() σε μια γραμμή και να εκτυπώνεται και αφού σιγουρεύτηκα ότι δουλεύει αυτό, άρχισα να εκτελώ τα βήματα επεξεργασίας που ζητούνται στην εργασία με τη σειρά το ένα μετά το άλλο και μετά και για κάθε βήμα τσέκαρα το output μέσω της εντολής print(text) που στην ουσία έκανε print το κείμενο με όσες επεξεργασίες είχαν γίνει μέχρι τότε. Όταν έφτασα στο 5ο ερώτημα χρειάστηκε να φτιάξω μια συνάρτηση function η οποία φυσικά θα καλείτε μέσω της μεθόδου sub και θα αντικαθιστά αυτά που ζητούνται με αυτά που απαιτεί η εργασία. Έφτιαξα και τις υπόλοιπες κανονικές εκφράσεις και τέλος αφού είχαν υλοποιηθεί όλα τα ζητούμενα έκανα print το τελικό κείμενο και σιγουρεύτηκα ότι έκανε αυτά που ζητάει η εργασία. Τέλος έκανα ένα απλό copy το κείμενο και το έκανα paste στο αρχείο output.txt που υπάρχει στο αποθετήριο μου από το fork του αποθετηρίου της εργασίας.

2.Περιγραφη της κανονικής έκφρασης που χρησιμοποίησα σε κάθε βήμα

Α.Η πρώτη κανονική έκφραση που χρησιμοποίησα χρησιμοποιήθηκε για την εξαγωγή και το εκ τύπωμά του τίτλου του κείμενο είναι η Process 1: rexp1 = re.compile(r'<title>(.+?)</title>'), η κανονική έκφραση εδώ έχει ως σκοπό την επιλογή όλων των χαρακτήρων που βρίσκονται μέσα στα title tags <title>…..</title>(τις τέλειες δηλαδή) όσοι και αν είναι αυτοί οι χαρακτήρες. Η τελειά . ανάμεσα στα title tags χρησιμοποιείτε γιατί θέλουμε οποιονδήποτε χαρακτήρα ενώ το + που είναι τελεστής επανάληψης γιατί θέλουμε στην ουσία οποιονδήποτε χαρακτήρα που εμφανίζεται μια η περισσότερες φορές. Ο τελεστής όμως αυτός είναι άπληστος και γιαυτο χρησιμοποιείτε το ? για την αποφυγή πιθανόν σφαλμάτων. Οι παρενθέσεις εδώ(εννοώ οι (.+?) παρενθέσεις) υπάρχουν απλώς για την εύκολη εξαγωγή και για να γίνει ευκολά print οτιδήποτε υπάρχει ανάμεσα στα title tags βλέπε i = rexp1.search(text), print(i.group(1)) κάνουμε print το group 1 στην ουσία ότι υπάρχει στην παρένθεση βλέπε(.+?). Συμπερασματικά στην ουσία έτσι κάνουμε print ότι υπάρχει ανάμεσα στα <title>…</title>(τις τέλειες) αυτό που ζητάει το πρώτο ζητούμενο.

Β.Η δεύτερη κανονική έκφραση που χρησιμοποίησα για την επιλογή και απαλοιφή των σχολίων και του περιεχομένου τους είναι η Process 2: rexp2 = re.compile(r'<!--.*?-->',re.DOTALL), η κανονική έκφραση εδώ έχει ως σκοπό την επιλογή όλων των σχολίων και έπειτα την απαλοιφή τους πιο κάτω μέσω της έκφρασης text = rexp2.sub(' ',text) οπού αντικαθιστά όλα τα σχόλια με το κενό. Καταρχάς το flag re.DOTALL χρησιμοποιήθηκε γιατί τα σχόλια μπορεί να επεκτείνονται σε πολλαπλές γραμμές αντίθετα

με τον τίτλο ο οποίος ήταν σε μια γραμμή. Η κανονική έκφραση είναι ιδιά σχεδόν με το την κανονική έκφραση του τίτλου με κάποιες μικρές διάφορες. Αντί για τον τελεστή + χρησιμοποίησα εδώ * γιατί μπορεί να υπάρχουν κενά σχόλια. Παρενθέσεις δεν χρειάζονται καθώς θέλουμε να απαλείψουμε όλα τα σχόλια με το κενό οπότε δεν θα τα κάνουμε εκτύπωση η τίποτα. Και φυσικά αλλάζει και η σύνταξη <title>…</title> με <!--…-->. Το πως τα κάνουμε απαλοιφή το είπαμε πιο πανό μέσω της sub με το κενό(δηλαδή αντικαθιστουνται τα σχόλια με το κενό).Στην ουσία μέσω όλων των εντολών που περιεγράφηκαν έχω απαλείψει τα σχόλια από το κείμενο το οποίο είναι το 2ο ζητούμενο της εργασίας.

Γ.Η τρίτη κανονική έκφραση που χρησιμοποιήθηκε για την επιλογή και απαλοιφή των (script και style tags μαζί με το περιεχόμενο τους) είναι η Process 3: rexp3 = re.compile (r'<(script|style).*?>.*?</(script|style)>',re.DOTALL), αυτή εδώ η κανονική έκφραση έχει ως σκοπό την επιλογή όλων των style και script tags με το περιεχόμενο τους ώστε αργότερα να γίνουν απαλοιφή με το κενό ‘ ‘ μέσω text = rexp3.sub(' ',text). Ας δούμε την κανονική έκφραση αναλυτικότερα. Εδώ έχω χρησιμοποιήσει τον τελεστή | ο οποίος στην ουσία σημαίνει είτε το ένα μέρος είτε το άλλο ώστε να επιλέγονται και τα script και τα style tags καθώς θέλουμε και τα <style>..</style> και <script>…</script> εδώ εννοώ και το περιεχόμενο τους. Με τους τελεστές .*? επιλέγονται και όλα τα περιεχόμενα των tags καθώς θέλουμε και αυτά. Έχω εξηγήσει πως δουλεύουν ας το εξηγήσω και άλλη μια τελευταία φορά με τον τελεστή . επιλέγουμε οποιονδήποτε χαρακτήρα αναμεσά στα tags επιπλέον βάζοντας το * επιλέγουμε οποιονδήποτε χαρακτήρα εμφανίζεται καμία η περισσότερες φορές και βάζοντας τον τελεστή ? αποφεύγουμε τα σφάλματα καθώς η λειτουργία του είναι η αναζήτηση οπουδήποτε χαρακτήρα, στην προκείμενη περίπτωση καμία η περισσότερες φορές. Παρενθέσεις δεν χρειαζόμαστε γιατί κάνουμε απαλοιφή και το flag re.Do tall χρησιμοποιείτε καθώς μπορεί τα tags να επεκτείνονται σε περισσότερες από γραμμές. Στην ουσία μέσω των εντολών που εξήγησα πιο πάνω και των 2(και sub δηλαδή) βρίσκομαι στο σημείο στο οποίο έχω απαλείψει και τα (style kai script) tags και το περιεχόμενο τους από το κείμενο όπως ήτανε και το 3ο ζητούμενο της εργασίας.

Δ.Η τέταρτη κανονική έκφραση που χρησιμοποιήθηκε για την επιλογή και την εκτύπωση των links είναι η Process 4: rexp4 = re.compile(r'<a.+?href="(.*?)".*?>(.*?)',re.DOTALL), ας δούμε πως δουλεύει αναλυτικότερα. Εδώ είχατε πει πως είχαμε το ελεύθερο να αυτοσχεδιάσουμε οπότε επέλεξα να επιλέξω και εκτυπώσω και τα περιεχόμενα των hrefs=”…” αλλά και ότι βρίσκεται αναμεσά στα <a>.. tags.Θα μπορούσα εδώ να εμφανίζω μόνο τα περιεχόμενα των href αλλά επέλεξα να εμφανίζονται όλα. Όπως και φαίνεται στην κανονική έκφραση υπάρχουν 2 σετ παρενθέσεων έτσι ώστε να μπορούμε ευκολά να εκτυπώσουμε και τα links και ότι βρίσκεται αναμεσά σε <a>.. ευκολά μέσω των εντολών for i in rexp4.finditer(text):, print('{} {}'.format(i.group(1),i.group(2)))(από κάτω) έχουμε 2 groups 1 και 2 για κάθε σετ παρενθέσεων. Όσο για την κανονική έκφραση δουλεύει έτσι, ταιριάζετέ αρχικά ότι βρίσκετε αναμεσά σε <a.. href έπειτα ότι βρίσκεται σε href=”..”(θα είναι το https:// k.t.l) , έπειτα ότι βρίσκεται από “ μέχρι > και τέλος φυσικά ότι βρίσκεται αναμεσά <a>…. Έχω εξηγήσει στις προηγούμενες κανονικές εκφράσεις πως δουλεύουν οι συνδυασμοί .*?,.+? και το flag re.DOTALL η σύνταξη ενός link είναι συνήθως …. Μέσω αυτών των εντολών θα εκτυπώσουμε στην ουσία όλα τα links όπως ήτανε και το 4ο ζητούμενο της εργασίας.

Ε.Η πέμπτη κανονική έκφραση που χρησιμοποιήθηκε ήταν η Process 5: rexp5 = re.compile(r'(<.+?/>)|(<.+?>|</.+?>)',re.DOTALL) σκοπός της οποίας είναι η επιλογή και απαλοιφή όλων των tags με το κενό text = rexp5.sub(' ',text). Ας την δούμε αναλυτικότερα. Εδώ έχουμε 2 περιπτώσεις.

Tags να είναι τύπου <…../> και φυσικά tags με διπλή μορφή όπως τα λέω εγώ τα οποία είναι έτσι <..><…/>. Οπότε πρέπει να λάβουμε υπόψη και τις 2 περιπτώσεις. Αυτό το έχω κάνει υλοποιώντας μια | και χωρίζοντας τις κανονικές εκφράσεις με παρενθέσεις έτσι ώστε στην ουσία να επιλέγονται είτε tags με μονή μορφή είτε με διπλή μορφή. Γενικά οι υπόλοιποι χαρακτήρες έχουν εξηγηθεί πιο πάνω οπότε δεν έχει νόημα να τους εξηγήσω και εδώ. Μέσω αυτής της κανονικής έκφρασης έχω απαλείψει όλα τα tags το οποίο είναι το 5ο ζητούμενο της εργασίας.

Ζ.Η έκτη κανονική έκφραση που χρησιμοποιήθηκε είναι η Process 6: rexp6 = re.compile(r'&(amp|gt|lt|nbsp)'). Η ουσία εδώ είναι να επιλεχτούν τα html entities που ζητάει η εργασία στο πινακάκι(όλα έχουν &μπροστά και μέσω μιας συνάρτησης που έχει υλοποιηθεί η οποία είναι πολύ απλή και κατά τη γνώμη μου δεν χρειάζεται εξήγηση και μέσω της εντολής text = rexp6.sub(function,text) καλείτε αυτή η συνάρτηση και εάν κάτι από αυτά που ζητάει το πινακάκι γίνει match(δηλαδή υπάρχει στο κείμενο) θα αντικαταστεί από τα returns που έβαλα βάση το πινακάκι φυσικά. Οπότε ότι από αυτά που ζητάει το πινακάκι υπάρχουν στο κείμενο μου θα αντικαθιστούν από αυτά που ζητάει πάλι το πινακάκι. Αυτό ήταν το 6ο ζητούμενο της εργασίας.

Η.Η έβδομη κανονική έκφραση που χρησιμοποίησα είναι η Process 7: rexp7 = re.compile(r'\s+') η οποία έχει ως σκοπό την επιλογή των whitespaces που αυτό γίνεται με τον χαρακτήρα \s και έπειτα χρησιμοποιείται και ο + γιατί θέλουμε τα whitespaces 1 η περισσότερες φορές. Έπειτα με την έκφραση text = rexp7.sub(' ',text) απαλείφονται τα συνεχόμενα whitespaces που υπάρχουν στο κείμενο με ένα whitespace μόνο.

Τέλος εκτυπώνεται το κείμενο μετά από όλα τα Processes(μερικά process κομμάτια τους όπως ο τίτλος και λινκς έχουν εκτυπωθεί ήδη)

3.Αναφορα σε πήγες που χρησιμοποίησα

Α. Χρησιμοποίησα κάποιες από της υπόδειξης που είχε η εργασία όπως η υπόδειξη για το διάβασμα του κείμενο και για την συνάρτηση

Β. Έπειτα χρησιμοποίησα και σημειώσεις από το site του μαθήματος ένα link είναι http://mixstef.github.io/courses/compilers/lecturedoc/unit2/module1.html το οποίο είναι μια εισαγωγή στις κανονικές εκφράσεις στην python

Και ένα άλλο http://mixstef.github.io/courses/compilers/lecture.html χρησιμοποίησα ότι αφορά τα εργαστήρια από εδώ