Μεταγλωτιστές 2020
Προγραμματιστική Εργασία #2

Ονοματεπώνυμο: Βασιλείου Σπυριδούλα και Κρυελέση Χριστίνα
ΑΜ: Π2017195 και Π2017129

• Συνοπτική περιγραφή της σειράς βημάτων επεξεργασίας στον κώδικά σας.

Αρχικά, έγινε εισαγωγή της βιβλιοθήκης re των κανονικών εκφράσεων για να επεξεργαστεί κείμενο HTML ιστοσελίδας (import re).

Στη συνέχεια, γράφτηκαν για κάθε βήμα / ζητούμενο οι αντίστοιχες κανονικές εκφράσεις.(rexp = re.compile('.....')).

Διαβάστηκε το αρχείο (testpage.txt), και τέλος γράφτηκαν κανονικές εκφράσεις με τη μέθοδο sub() και τυπώθηκε το κείμενο, όπως έχει διαμορφωθεί μετά τις μετατροπές.

• Περιγραφή της κανονικής έκφρασης που χρησιμοποιήθηκε σε κάθε βήμα.

1. Εξαγωγή και εκτύπωση του τίτλου (οτιδήποτε βρίσκεται μεταξύ <title> και </title>).

- rexp1 = re.compile('<title>(.+?)</title>',re.DOTALL)

Με τη μεταβλητή rexp1 ψάχνουμε για ταιριάσματα σύμφωνα με την κανονική έκφραση <title>(.+?)</title>. θα ταιριάξει string που αρχίζουν από το tag <title> μετά θα ακολουθεί ένας **οποιοσδήποτε** χαρακτήρας(.)(ταιριάζει και το newline με το flag re.DOTALL), 1 ή περισσότερες φορές(+)μετά θα ακολουθεί </title>.

- for m in rexp1.finditer(text):
  print(m.group(1))

Για την εκτύπωση του τίτλου χρησιμοποιήσαμε την finditer και group για να εκτυπωθεί υποσύνολο του ταιριάσματος (οτιδήποτε βρίσκεται μεταξύ <title> και </title>) με το group(1) εκτυπώνουμε αυτό που ταιριάζει από την αυτή την παρένθεση (.+?).

2. Απαλοιφή των σχολίων (οτιδήποτε βρίσκεται μεταξύ <!-- και -->).

- rexp2 = re.compile('<!--(.+?)-->',re.DOTALL)

Η ίδια λογική με το παραπάνω ερώτημα.

- text = rexp2.sub(' ',text)

Η απαλοιφή έγινε αντικαθιστωντας με έναν κενό χαρακτήρα (space).
Η sub() παίρνει ως ορίσματα αυτό που θα αντικαταστήσει και το κείμενο εισόδου και σε κάθε ταίριασμα, μέσω της rexp2, αντικαθιστά κάθε σχόλιο με ' '.

3. Απαλοιφή των <script> και <style> tags με όλο τους το περιεχόμενο, μέχρι δηλαδή να συναντήσετε το αντίστοιχο </script> ή </style> (και τα τελευταία).

- rexp3a = re.compile('<script.*?>(.+?)</script>',re.DOTALL)
- rexp3b = re.compile('<style.*?>(.+?)</style>',re.DOTALL)

  ή (απαλοιφή των <script> και <style> tags με μία μόνο κανονική έκφραση)

- rexp3 = re.compile('<(style|script).*?>(.+?)</(style|script)>')

Σε αυτή την κανονική έκφραση χρησιμοποιήθηκε και το σύμβολο της εναλλαγής για να ταιριάξει είτε το style είτε το script (style|script).

- text = rexp3a.sub(' ',text)
- text = rexp3b.sub(' ',text)
  ή
- text = rexp3.sub(' ',text)

Χρήση της sub() για την απαλοιφή και αντικατάσταση με κενό χαρακτήρα.

4. Εξαγωγή και εκτύπωση του συνδέσμου (ιδιότητα href) από <a> tags και του κειμένου τους(ό,τι βρίσκεται δηλαδή μεταξύ των <a> και)

- rexp4 = re.compile(r'<a(.*?href="(.+?)">.+?)',re.DOTALL)

Η rexp4 ταιριάζει ό,τι ξεκινάει από τους χαρακτήρες <a μετά θα ακολουθεί ένας οποιοσδήποτε χαρακτήρας(.)(ταιριάζει και το newline με το flag re.DOTALL) 1 ή περισσότερες φορές (+), μετά θα ακολουθεί το href=",μετά θα ακολουθεί ένας οποιοσδήποτε χαρακτήρας(.) 1 ή περισσότερες φορές (+) αυτός θα είναι ο σύνδεσμος ,στη συνέχεια το > μέτα με το .+? ότι εξηγήσαμε και παραπάνω (αυτό είναι το κείμενο) και τέλος ακολουθεί το .

- for m2 in rexp4.finditer(text):
  print('{} {}'.format(m2.group(2),m2.group(1)))

  ή (για απλή εκτύπωση ,χωριστά)

  print(m2.group(2))
  print(m2.group(1))

Με το .format και το m2.group(2) παίρνουμε τον σύνδεσμο και με το m2.group(1) παίρνουμε το κείμενο (ό,τι βρίσκεται δηλαδή μεταξύ των <a> και).

5. Απαλοιφή όλων των tags από το κείμενο.
- rexp5 = re.compile('<.+?>',re.DOTALL)

Η rexp5 ταιριάζει ό,τι ξεκινάει από τον χαρακτήρα < και μετά ακολουθεί ένας οποιοσδήποτε χαρακτήρας(.) 1 ή περισσότερες φορές (+) και τέλος ακολουθεί > έτσι ταιριάζει όλα τα tags.

- text = rexp5.sub(' ',text)

Χρήση της sub() για την απαλοιφή και αντικατάσταση με κενό χαρακτήρα.

6. Μετατροπή των ειδικών HTML entities που υπάρχουν στο κείμενο σύμφωνα με τον παρακάτω πίνακα:

HTML entities	Αντικαταστήστε με
&amp;	&
&gt;	>
&lt;	<
	χαρακτήρα space

- def cb(m):
```
    if m.group(1) == 'amp':
        return  '&'
    elif m.group(1) == 'gt':
       return  '>'
    elif m.group(1) == 'lt':
       return  '<'
    elif m.group(1) == 'nbsp':
       return  ' '
```

Με την συνάρτηση cb χρησιμοποιείται το ταίριασμα από το group(1) για να αντικατασταθεί με τα κατάλληλα σύμβολα μέσω των συνθηκών.

- rexp6 = re.compile('&(.+?);')

Εντοπίζει την πρώτη στήλη του πίνακα.

- text = rexp6.sub(cb,text)

7. Μετατροπή ακολουθιών συνεχόμενων χαρακτήρων whitespace σε ένα ακριβώς κενό.

- rexp7 = re.compile(r'\s+')

Χρησιμοποιούμε raw string (r'...') επειδή υπάρχει \ μέσα στην κανονική έκφραση.
Η rexp7 ταιριάζει οποιοσδήποτε whitespace 1 ή περισσότερες φορές.

- text = rexp7.sub(' ',text)

• Η εργασία υλοποιήθηκε με βάση την θεωρία του εργαστηρίου:
http://mixstef.github.io/courses/compilers/lecturedoc/unit2/module1.html