

Μεταγλωτιστές 2020

Προγραμματιστική Εργασία #2

Ονοματεπώνυμο: Φουτσιτζή Σοφρονία

ΑΜ: Π2017063

Η υλοποίηση της συγκεκριμένης εργασίας έγινε σε γλώσσα προγραμματισμού Python και χρησιμοποιήθηκε αποκλειστικά η βιβλιοθήκη `re` των κανονικών εκφράσεων.

Αρχικά, στοχεύοντας στην υλοποίηση της εργασίας μελετήθηκε ιδιαίτερα η θεωρία αλλά και όλα τα εργαστήρια τα οποία έλαβαν χώρα κατά την διάρκεια του εξαμήνου και αφορούσαν την συγκεκριμένη βιβλιοθήκη και το σύνολο των κανονικών εκφράσεων. Έπειτα, ξεκίνησε κατά σειρά η δημιουργία κανονικών εκφράσεων για την υλοποίηση του προγράμματος.

Αρχικά, έγινε άνοιγμα του αρχείου στο οποίο περιέχεται το κείμενο σε `html`, το οποίο χρησιμοποιήθηκε, συγκεκριμένα, για ανάγνωση. Ένω έγινε άνοιγμα και ενός δευτέρου αρχείου (`output.txt`), που χρησιμοποιήθηκε για την εξαγωγή/ εκτύπωση των ζητούμενων.

Για την εξαγωγή και εκτύπωση του τίτλου της ιστοσελίδας χρησιμοποιήθηκε η ακόλουθη κανονική έκφραση.

```
>rexp = re.compile(r'(<title>)(.+?)</title>')
```

Συγκεκριμένα, η κανονική έκφραση εδώ βρίσκει ότι ξεκινάει με `<title>`, συνεχίζει με οποιοδήποτε χαρακτήρα, λόγω της τελείας `“.”`, τον οποίο αναζητά μία ή περισσότερες φορές λόγω του συν `“+”`. Ο χαρακτήρας αυτός ωστόσο είναι άπλειστος οπότε για την αποφυγή των πιθανών σφαλμάτων χρησιμοποιείται και ο τελεστής `“?”`, η λειτουργία του οποίου είναι η αναζήτηση του οποιουδήποτε χαρακτήρα, στην προκειμένη περίπτωση, μηδέν ή παραπάνω φορές. Η αναζήτηση σταματάει όταν γίνει η εύρεση του `</title>`. Τα μέρη της κανονικής έκφρασης είναι διαχωρισμένα με παρενθέσεις, καθώς επιθυμούμε να εκτυπώσουμε ένα συγκεκριμένο μέρος της (τον τίτλο), κάτι που θα υλοποιηθεί με την χρήση `group()`. Για την εύρεση και εκτύπωση του τίτλου χρησιμοποιήθηκε μία δομή επανάληψης και η `finditer()`.

Για το επόμενο ζητούμενο δηλαδή την απαλοιφή των σχολίων χρησιμοποιήθηκε, αρχικά η κανονική έκφραση:

```
>rexp1 = re.compile(r'<!--(.+?)-->',re.DOTALL)
```

η οποία αναζητά τα σχόλια (το `re.DOTALL` χρησιμοποιείται καθώς θέλουμε να γίνει αναζήτηση ακόμα και αν έχει `newline`) και με την χρήση της `sub` γίνεται η απαλοιφή, δηλαδή η αντικατάσταση του ό,τι βρίσκει η `rexp1` με κενό. Αυτό που προκύπτει το “αποθηκεύουμε ως `newtext`, και στην συνέχεια χρησιμοποιούμε αυτό για τις επόμενες αλλαγές.

```
>newtext = rexp1.sub("",text)
```

Η απαλοιφή των `script` και `style` tags υλοποιήθηκε κατά παρόμοιο τρόπο καθώς αυτό που θέλουμε να σβήσουμε από το κείμενο είναι οτιδήποτε περιέχεται μεταξύ των `<style>` και `</style>` αλλά και `<script>` και `</script>`, αντίστοιχα. Για την αναζήτηση χρησιμοποιήθηκε μία κανονική έκφραση που περιέχει δύο μέλη για κάθε ένα `<script>` και `<style>`, αντίστοιχα, τα οποία διαχωρίζονται με `or (|)`. Επίσης, χρησιμοποιείται η `re.DOTALL` (το οποίο χρησιμοποιώ προληπτικά σε όλες μου τις κανονικές εκφράσεις), καθώς δεν γνωρίζουμε πόσες σειρές καταλαμβάνει το κάθε tag.

```
>rexp2 = re.compile(r'(<script>(.+?)</script>)|(<style>(.+?)</style>)',re.DOTALL)
```

Για την εξαγωγή και εκτύπωση του συνδέσμου (ιδιότητα href) από τα <a> tags του κειμένου χρησιμοποιήθηκαν δύο εκφράσεις. Αρχικά, η πρώτη κανονική έκφραση υλοποιεί αναζήτηση για την εύρεση των <a> tags.

```
>rexp3 = re.compile(r'(<a)(.+?)/a>',re.DOTALL)
```

Η δεύτερη είναι αυτή που θα αναζητήσει την ιδιότητα href.

```
>rexp4 = re.compile(r'href="(.(+?)">([^\<].+?)<',re.DOTALL)
```

Για την ορθή αναζήτηση τους υλοποιήθηκαν δύο δομές επανάληψης (for m2 in rexp3.finditer(newtext1) και for m3 in rexp4.finditer(m2.group(2))) , όπου η δεύτερη εμπεριέχεται στην πρώτη, με αποτέλεσμα να αναζητά την ιδιότητα που επιθυμούμε μέσα στα συγκεκριμένα tags. Όπως είναι ορατό και στις δύο περιπτώσεις (κανονικές εκφράσεις) χρησιμοποιήθηκαν παρενθέσεις καθώς η αναζήτηση του href θα υλοποιηθεί στο ενδιάμεσο των <a> tags, δηλαδή στο group(2) και αυτό που επιθυμούμε έπειτα να εκτυπώσουμε βρίσκεται σε συγκεκριμένα σημεία της κανονικής έκφρασης.

Η αναζήτηση όλων των tags του κειμένου έγινε με μία κανονική έκφραση της μορφής:

```
>rexp5 = re.compile(r'<(.(+?))>',re.DOTALL)
```

καθώς παρατηρήθηκε ότι τα tags είναι της μορφής πχ <meta charset="utf-8" > και αντικαταστάθηκαν με την χρήση της συνάρτησης sub(), ενώ το κείμενο που προέκυψε απόθηκεύτηκε σε μία νέα μεταβλητή newtext2, όπως γίνεται σε κάθε αλλαγή/αντικατάσταση/απαλοιφή που υλοποιήθηκε στο πρόγραμμα.

```
>newtext2 = rexp5.sub("",newtext1)
```

Με παρόμοιο τρόπο με το παραπάνω ζητούμενο υλοποιήθηκε και η αντικατάσταση των HTML entities (&, >, <,), με τον αντίστοιχο επιθυμητό χαρακτήρα (&, >,<,space). Ωστόσο, καθώς, η δομή στο κείμενο html των όσων έπρεπε να αντικατασταθούν ήταν παρόμοια χρησιμοποιήθηκε μία κανονική έκφραση η οποία αναζητούσε τα κοινά στοιχεία, δηλαδή το & και το ;, χωρίς να ενδιαφέρεται για το περιεχόμενο.

```
>rexp6 = re.compile(r'&(.(+?));',re.DOTALL)
```

Έπειτα, για την σωστή αντικατάσταση στην sub() χρησιμοποιήθηκε μία συνάρτηση repl, η οποία ανάλογα με το περιεχόμενο (group(2) σύμφωνα με τις παρενθέσεις που τοποθετήθηκαν) επέστρεφε το αντίστοιχο σύμβολο.

```
>newtext3 = rexp6.sub(repl,newtext2)
```

Τέλος, γθα την απαλοιφή των κενών αλλά και των επιπλέον γραμμών χρησιμοποιήθηκε μία κανονική έκφραση της μορφής:

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

και η αντικατάσταση έγινε με την χρήση της sub(). Το τελικό κείμενο που προέκυψε εκτυπώθηκε σε ένα αρχείο.

```
File Edit Search View Document Help
Warning, you are using the root account, you may harm your system.

import re

def repl(m): #antikatastash twm html entities me ta antistixa sunbola (xrhsh synarthshs)
    if m.group(1)=='amp': #sugkrisi toy group(1)
        return '&'
    if m.group(1)=='gt':
        return '>'
    if m.group(1)=='lt':
        return '<'
    if m.group(1)=='nbsp':
        return ' '

rexp = re.compile(r'<title>(.*?)</title>') #prwto zhtoumeno eksagwgh titlou
rex1 = re.compile(r'<!--(.+?)-->',re.DOTALL) #apalifh twm sxoliwn
rex2 = re.compile(r'<script>(.*?)</script>(<style>(.*?)</style>)' #apalifh scrip kai style
rex3 = re.compile(r'<a>(.*?)</a>') # ebreash <a> kai </a>
rex4 = re.compile(r'href="(.*?)>(.*?)<' # href
rex5 = re.compile(r'<(.+?)>',re.DOTALL) #evresh olwn twm tags
rex6 = re.compile(r'<(.+?)>',re.DOTALL) #evresh twm eisodwn poy ksekinan me & kai teleiwoun se ;
rex7 = re.compile(r'<(.+?)>' #evresh kemwn kai new lines
with open('testpage.txt','r') as fp: #anagwnsh testpage.html
    text = fp.read()
    f = open("output.txt","w") #anoigma neou arxeiou gia write
    for m in rexp.finditer(text):
        print(m.group(1),file=f) #ektipwsh perioxomenou titlou
    newtext = rex1.sub('',text) # me thn rex1 antikathistoume ta sxolia me keno
    newtext1 = rex2.sub('',newtext) #apalifh twm oswn tha brei h rex2 dld scrip kai style
    for m2 in rex3.finditer(newtext1): #gia kathe <a> </a>
        for m3 in rex4.finditer(m2.group(1)): #briskw to href pou perilambanetai sto <a> </a>
            print(m3.group(1),m2.group(1),file=f) #ektipwsh (sto arxeio output.txt) me thn xrhsh
    newtext2 = rex5.sub('',newtext1) # antikatatash oswn briskei h rex5 me keno(apalifh twm sxoliwn)
    newtext3 = rex6.sub(repl,newtext2) #allagh twm oswn briskei h rex6 ($amp, %gt..)me bash thn sunar
    newtext4 = rex7.sub(' ',newtext3) #apalifh twm kemwn kai new lines me mono ena keno
    print(newtext4,file=f) #ektipwsh sto arxeio

root@kali:~# python3 html-processor.py
root@kali:~#
```

```
File Edit Search View Document Help
Warning, you are using the root account, you may harm your system.

IONIO ΠΑΝΕΠΙΣΤΗΜΙΟ
https://webmail.ionio.gr/ WebMail
https://ionio.gr/gr/community/directory/ Τηλεφωνικός Κατάλογος
https://opencourses.ionio.gr Open eClass (OpenCourses) - Ασύγχρονη Τηλεκπαίδευση
http://e-class.ionio.gr/ e-Class - Ασύγχρονη Τηλεκπαίδευση
http://gram-web.ionio.gr/unistudent/?lang=el-gr Online Υψηλές Γραμματείες (gram-web)
http://gram-web.ionio.gr/classweb/?lang=el-gr Σύστημα Υποβοήθησης Διδασκαλίας (classweb)
https://studentsupport.ionio.gr/gr/ Υποστήριξη Παρεμβάσεων Κοινωνικής Μέριμνας
http://iup.ionio.gr/index.php?newlang=greek Βιβλιοθήκη
http://sites.ionio.gr/international/gr/ Διεθνείς & Δημόσιες Σχέσεις
http://dasta.ionio.gr/internship/ Γραφείο Πρακτικής Άσκησης
http://kedivim.ionio.gr/ Κέντρο Επιμόρφωσης και Δια Βίου Μάθησης
http://modip.ionio.gr/gr/ Μονάδα Διασφάλισης Ποιότητας - ΜΟ.ΔΙ.Π
http://museum.ionio.gr/gr/ Μουσείο Ιονίου Πανεπιστημίου
http://noc.ionio.gr/ Δίκτυα & Τηλεπικοινωνίες
http://history.ionio.gr/gr/ Ιστορία
http://dflti.ionio.gr/el/ Ξένων Γλωσσών, Μετάφρασης & Δερμηνείας
http://tab.ionio.gr/?q=el Αρχαιονομία, Βιβλιοθηκονομία & Μουσειολογία
http://di.ionio.gr/ Πληροφορική
http://avarts.ionio.gr/gr/ Τεχνών Ήχου και Εικόνας
http://music.ionio.gr/gr/ Μουσικών Σπουδών
http://dmc.ionio.gr/ Ψηφιακών Μέσων και Επικοινωνίας
http://ethnomus.ionio.gr/gr/ Εθνομουσικολογία
http://envi.ionio.gr/gr/ Περιβάλλοντος
http://fst.ionio.gr/gr/ Επιστήμης και Τεχνολογίας Τροφίμων
http://rd.ionio.gr/ Περιφερειακής Ανάπτυξης
http://tourism.ionio.gr/ Τουρισμού
https://ionio.gr/gr/about/statistics/ Στατιστικά κίνησης
https://ionio.gr/gr/about/links/ Σύνδεσμοι
https://ionio.gr/gr/about/sitemap/ Χάρτης δικτυακού τόπου
https://ionio.gr/gr/about/webteam/ Ομάδα Ανάπτυξης & Διαχείρισης
https://www.facebook.com/ionio.gr" title="facebook" 
https://twitter.com/myionio" title="twitter" 
http://www.youtube.com/user/myionio" title="youtube" 
https://www.instagram.com/ionianuniversity/" title="instagram" 
https://ionio.gr/feeds/rss_gr.xml" title="rss feed" 
https://ionio.gr/en/" class="tooltip-bottom-left" title="Change language to English En
```

Πηγές

- Κανονικές εκφράσεις και Python
“<http://mixstef.github.io/courses/compilers/lecturedoc/unit2/module1.html#sub>”