**ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ**

Τμήμα Πληροφορικής

# Μεταγλωττιστές

Ακαδ. Έτος: 2016-2017

# Προγραμματιστική εργασία #1

Χαριλάου Γιώργος

Π2014151

**Περιγραφή του κώδικα:**

Η περιγραφή έχει γίνει στον κώδικα και είναι επίσης στο Github. Μια σύντομη περιγραφή θα έλεγε ότι αρχικά αφού ανοίξει το αρχείο το αρχείο .srt , η πρώτη δουλειά που γίνεται διαχωρίζονται η γραμμές του αρχείο βάση των whitelines. Στη συνέχεια γίνεται ένας έλεγχος κατά τον οποίο ταιριάζει με μια λογική έκφραση ('^(\d+)$') αν οι χρονικές περιόδοι είναι όπως αυτήν. Στη συνέχεια βάζει στο out τα strings του αρχείου και επίσης διαχωρίζει με τη μέθοδο split τις χρονικές περιόδους για κάθε string ξεχωριστά (από πού αρχίζει μέχρι που τελειώνει). Μετά, χωρίζουμε τον χρόνο σε ώρες, λεπτά, δευτερόλεπτα και κλάσματα δευτερολέπτου και τα εισάγει στην αρχή και το τέλος στο map. Για το offset, δημιουργούνται 2 πίνακες όπου προσθέτεται κάθε φορά στον πρώτο πίνακα ο δεύτερος μαζί με το Offset, έτσι έχουμε ως αποτέλεσμα στο map να βρίσκεται τώρα η αρχή, το τέλος καθώς και το offset. Τέλος γίνεται η εγγραφή στο καινούργιο αρχείο βάση κάποια μορφοποίηση.

```
#βιλιοθήκες που χρησιμοποιούνται.
import sys
import re
import argparse


parser = argparse.ArgumentParser()
# add mandatory (positional) arguments
parser.add_argument("fname",help="input srt file name") #για το όνομα του αρχείου.
parser.add_argument("offset",type=float,help="subtitle offset in seconds to apply (can be fractional)")  # πόση χρονική περίοδο θέλει ο χρήστης.

# parse arguments
args = parser.parse_args()

with open(args.fname,newline='') as ifp: #άνοιγμα του αρχείου.
        i=0 #αρχικοποιά την μεταβλητή i με την τιμή 0.
        for line in ifp: #για κάθε γραμμή του αρχείου .srt που εισάγεται..
                line = line.strip()          #κάθε γραμμή χωρίζεται με τα whitespace
```

```
            if not line: #αν δεν είναι γραμμή..
                out += '\n' #το αποτέλεσμα να είναι το '\n'.
                continue #συνέχεισε

            i+=1 #αύξησε την τιμή i κατά ένα.

            if re.compile('^(\d+)$').match(line): #αν η λογική έκφραση διαχωρισμού των τιμών στο .srt αρχείο ταιριάζει με την γραμμή
                i = 1 #η μεταβλητή i παίρνει την τιμή 1.

            if i == 1: #αν τότε είναι ίση με 1..
                out += '%s\n' % line # το αποτέλεσμα να είναι + το string και την γραμμή.

            elif i == 2: #αν όμως είναι 2..
                start, end = line.split(' --> ') #η αρχή και το τέλος χωρίζεται βάζει του συμβόλου ' --> ' στο αρχείο .srt που βρίσκει.

        def parse_time(time): #function που χωρίζει τον χρόνο.
            hour, minute, second = time.split(':') #χωρίζει την ώρα, τα λεπτά και τα δευτερόλεπτα όπου βρίσκει τον χαρακτήρα ':'
            hour, minute = int(hour), int(minute)        #παίρνει σαν μεταβλητή integer τις ώρες και τα λεπτά
            second_parts = second.split(',')      #χωρίζει τα δευτερόλεπτα από τα κλάσματα δευτερολέπτου όπου βρεί τον χαρακτήρα ','
            second = int(second_parts[0])        #χωρίζει τα δευτερόλεπτα σε μέρη σε ένα πίνακα
            milisecond = int(second_parts[1])  # και στη συνέχεια τα κλάσματα του δευτερολέπτου σε ένα άλλο.

            return [hour, minute, second, milisecond] #και η συνάρτηση επιστρέφει σαν λίστα τις ώρες, τα λεπτά, τα δευτερόλεπτα και τα κλάσματα του δευτερολέπτου.

        start, end = map(parse_time, (start, end)) #η αρχή και το τέλος που διαχωρίστηκε πιο πάνω ισούται με το map που πέρνει ως ορίσματα την κλάση διαχωρισμού της ώρας με ορίσματα την αρχή και το τέλος που δημιουργήθηκαν.
```

```
    def offset(time): #η κλάση offset που επιστρέφει τον χρόνο με το offset.
        offset #δημιουργία της μεταβλητής
        time[1] += (time[2] + offset) / 60 #έτσι ο χρόνος με τιμή 1 προσθέτει κάθε φορά + τον χρόνο με τιμή 2 και το offset και το διαιρεί με το 60 για τον χρόνο
        time[2] = (time[2] + offset) % 60 # ο χρόνος με τιμή 2 είναι ο χρόνος με τιμή 2 και το offset % 60.
        return time #επιστροφή του χρόνου.

    start, end = map(offset, (start, end)) #τώρα η αρχή και το τέλος της χρονικής περιόδου του .srt αρχείου είναι ίση με το map όπου οι τιμές του είναι το offset και η αρχή με το τέλος της χρονικής περιόδου.

    out += '%s:%s:%s,%s --> %s:%s:%s,%s\n' % ( #σε αυτό το σημείο γίνεται η μορφοποίηση του αρχείου εξόδου και η έξοδος του πίνακα.
        str(start[0]).rjust(2, '0'),
        str(start[1]).rjust(2, '0'),
        str(start[2]).rjust(2, '0'),
        str(start[3]).rjust(3, '0'),

        str(end[0]).rjust(2, '0'),
        str(end[1]).rjust(2, '0'),
        str(end[2]).rjust(2, '0'),
        str(end[3]).rjust(3, '0'))

                elif i >= 3: #αν το i είναι μεγαλύτερο του 3..
                        out += '%s\n' % line# η έξοδος είναι απλά η έξοδος των string όπως έχουν.

                sys.stdout.write(newline , out)# γράφει κάθε νές γραμμή χρονικής πληροφορίας στο νέο αρχείο.

f.close(ifp)
```

## Βιβλιογραφία


- https://docs.python.org/2/library/re.html
- https://www.tutorialspoint.com/python/string_split.htm
- http://stackoverflow.com/questions/13013734/string-strip-in-python
- https://docs.python.org/3/library/stdtypes.html#str.format
- https://docs.python.org/3/library/string.html#format-specification-mini-language
- https://gist.github.com/chanux/2042676
- https://github.com/ericremoreynolds/fix-srt/blob/master/fix-srt.py
- https://github.com/drm/srtfix.py/blob/master/srtfix.py