

# Εργαστήριο Σημασιολογικού Ιστού 2017

## Προγραμματιστική Εργασία #1

---

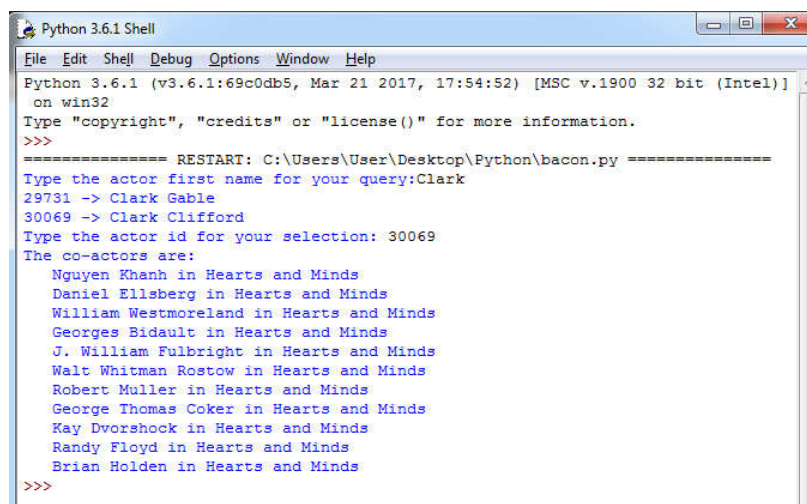
Σαρδέλας Αθανάσιος  
Π201299

### 1. Συνοπτική περιγραφή του κώδικα

Η βασική λειτουργία του κώδικα που παρατίθεται περιλαμβάνει τα εξής μέρη:

- Υλοποίηση της σύνδεσης με το *SPARQL endpoint*.
- Δίνεται το όνομα του ηθοποιού/ών προς αναζήτηση.
- Συντάσσεται με τον ορθό τρόπο το SPARQL ερώτημα περιλαμβάνοντας το δηλωθέν όνομα ηθοποιού.
- Σε περίπτωση επιτυχημένης αναζήτησης, τα αποτελέσματα αποθηκεύονται σε μορφή πίνακα και παρουσιάζονται στην οθόνη. Αλλιώς γίνεται έξοδος με εμφάνιση σχετικού μηνύματος λάθους.
- Ζητείται από τον χρήστη να πληκτρολογήσει τον διακριτικό ακέραιο *aid* του αντίστοιχου ηθοποιού.
- Συντάσσεται με τον ορθό τρόπο το δεύτερο SPARQL ερώτημα περιλαμβάνοντας το *aid* ηθοποιού.
- Σε περίπτωση επιτυχημένης αναζήτησης, τα αποτελέσματα αποθηκεύονται σε μορφή πίνακα και παρουσιάζονται στην οθόνη με την μορφή «ηθοποιός συνεργάτης» στην «ταινία που συμπρωταγωνίστησαν». Αλλιώς γίνεται έξοδος με εμφάνιση σχετικού μηνύματος λάθους.

Τα αποτελέσματα από μία τυχαία εκτέλεση προβάλλονται στην παρακάτω εικόνα.



```
Python 3.6.1 Shell
File Edit Shell Debug Options Window Help
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\User\Desktop\Python\bacon.py =====
Type the actor first name for your query:Clark
29731 -> Clark Gable
30069 -> Clark Clifford
Type the actor id for your selection: 30069
The co-actors are:
  Nguyen Khanh in Hearts and Minds
  Daniel Ellsberg in Hearts and Minds
  William Westmoreland in Hearts and Minds
  Georges Bidault in Hearts and Minds
  J. William Fulbright in Hearts and Minds
  Walt Whitman Rostow in Hearts and Minds
  Robert Muller in Hearts and Minds
  George Thomas Coker in Hearts and Minds
  Kay Dvorshock in Hearts and Minds
  Randy Floyd in Hearts and Minds
  Brian Holden in Hearts and Minds
>>>
```

## 2. Ανάλυση των ερωτημάτων SPARQL

Τα ερωτήματα SPARQL που αναζητούν την πληροφορία στο *linkedmdb* αρχικά δόθηκαν στο online εργαλείο SPARQL Explorer όπως παρουσιάζονται στις οθόνες παρακάτω για την εκτίμηση της απάντησης που θα προέκυπτε από το εκτελέσιμο της Python.

SPARQL Explorer for <http://www.linkedmdb.org/sparql>

SPARQL:  

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX oddlinker: <http://data.linkedmdb.org/resource/oddlinker/>
PREFIX map: <file:/C:/d2r-server-0.4/mapping.n3#>
PREFIX db: <http://data.linkedmdb.org/resource/>
PREFIX dbpedia: <http://dbpedia.org/property/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX dc: <http://purl.org/dc/terms/>
PREFIX movie: <http://data.linkedmdb.org/resource/movie/>

SELECT ?actor_name ?actor_id WHERE {
  ?actor movie:actor_actorid ?actor_id .
  ?actor movie:actor_name ?actor_name .
  filter(regex(?actor_name, 'Clark','i'))
}
```

Results:

SPARQL results:

actor_name	actor_id
"Caitlin Clarke"	29410
"Clark Gable"	29731
"Clark Clifford"	30069
"Lana Clarkson"	30802
"Jim Clark"	30844
"Petula Clark"	30942
"Vince Clarke"	31009
"Lenny Clarke"	31291
"Jeremy Clarkson"	31363

Powered by [D2R Server](#)

Εικόνα 1 Πρώτο ερώτημα επιλογής ονόματος ηθοποιού.

SPARQL Explorer for <http://www.linkedmdb.org/sparql>

SPARQL:  

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX oddlinker: <http://data.linkedmdb.org/resource/oddlinker/>
PREFIX map: <file:/C:/d2r-server-0.4/mapping.n3#>
PREFIX db: <http://data.linkedmdb.org/resource/>
PREFIX dbpedia: <http://dbpedia.org/property/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX dc: <http://purl.org/dc/terms/>
PREFIX movie: <http://data.linkedmdb.org/resource/movie/>

SELECT ?actname ?mtitle WHERE {
  ?film movie:actor ?mactor .
  ?film rdfs:label ?mtitle .
  ?film movie:actor <http://data.linkedmdb.org/resource/actor/30069> .
  ?mactor movie:actor_name ?actname .
  ?mactor movie:actor_actorid ?aid .
  FILTER(?aid!=30069)
}
```

Results:

SPARQL results:

actname	mtitle
"Nguyen Khanh"	"Hearts and Minds"
"Daniel Ellsberg"	"Hearts and Minds"
"William Westmoreland"	"Hearts and Minds"
"Georges Bidault"	"Hearts and Minds"
"J. William Fulbright"	"Hearts and Minds"
"Walt Whitman Rostow"	"Hearts and Minds"
"Robert Muller"	"Hearts and Minds"
"George Thomas Coker"	"Hearts and Minds"
"Kay Dvorshock"	"Hearts and Minds"
"Randy Floyd"	"Hearts and Minds"
"Brian Holden"	"Hearts and Minds"

Powered by [D2R Server](#)

Εικόνα 2 Δεύτερο ερώτημα εμφάνισης συμπρωταγωνιστών

Το πρώτο ερώτημα στις γραμμές 10 – 18 του αρχείου `bacon.py`, αναζητά στην βάση του *linkedmdb*, η οποία αναφέρεται μέσω των κατάλληλων `prefixes`, τους ηθοποιούς (μεταβλητή `?actor_id` διακριτικός ακέραιος ηθοποιού) με τα ονόματά τους (μεταβλητή `?actor_name`), με βάση την είσοδο του χρήστη. Η αναζήτηση γίνεται με περαιτέρω επιλογή από το σύνολο όλων των ηθοποιών χρησιμοποιώντας την εντολή *FILTER* και *regex* κατάλληλα, σύμφωνα με την είσοδο του χρήστη για να επιστρέψει τα αντίστοιχα ονόματα. Τελικά επιστρέφονται τα `?actor_id`, `?actor_name`.

Στο δεύτερο ερώτημα στις γραμμές 60 – 71 του αρχείου `bacon.py`, αναζητούνται πάλι στην βάση του *linkedmdb*, η οποία αναφέρεται μέσω των κατάλληλων `prefixes`, οι ηθοποιοί (μεταβλητή `?actname`) που συμπρωταγωνιστούν με τον επιλεγμένο ηθοποιό ο οποίος υποδεικνύεται μέσω του `(?film movie:actor <http://data.linkedmdb.org/resource/actor/"""+str(aid)+""">)` και οι τίτλοι των ταινιών (μεταβλητή `?mtitle`) που συμπρωταγωνίστησαν. Τέλος γίνεται φιλτράρισμα των ηθοποιών (με χρήση *FILTER*) ώστε να διαφέρουν από τον ηθοποιό που έδωσε ο χρήστης. Τελικά επιστρέφονται τα `?actname`, `?mtitle`.