

## Ενότητα 3: Τα δεδομένα στο Web

(και η σημασιολογία τους –“semantics”)

# Σημασιολογία –“semantics”

- ▶ Τι **σημαίνουν** τα δεδομένα; Ποια η **έννοιά** τους;
- ▶ **Μετάδοση** έννοιας και ενσωμάτωση στη **γνώση**..
- ▶ ..έχοντας ως αποτέλεσμα την κατάλληλη **δράση**!
- ▶ Πώς θα γίνει αντιληπτή από τις “**μηχανές**”;

# Έμμεση Σημασιολογία

**Παράδειγμα:** ονόματα συναρτήσεων ή μεταβλητών

```
quicksort :: (Ord a) => [a] -> [a]
quicksort [] = []
quicksort (x:xs) =
    let smallerOrEqual = [t|t<=xs,t<=x]
        greater = [t|t<=xs,t>x]
    in quicksort smallerOrEqual ++ [x] ++ quicksort greater
```

- ▶ Τι καταλαβαίνετε ως προγραμματιστές;
  - ▶ Ανεξάρτητα από το αν γνωρίζετε τη γλώσσα!
  - ▶ Τα σχόλια θα βοηθούσαν περαιτέρω στη μεταφορά σημασιολογίας..
- ▶ Τι μπορεί όμως να καταλάβει η “μηχανή”;

# Ιστοσελίδα και μηχανή

Τι “αντιλαμβάνεται” η μηχανή:

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>????????????????</title>
  </head>

  <body>
    <h1>????????????????</h1>
    ?????????????????????
    ?????????????????????
  </body>
</html>
```

- ▶ Προσοχή όμως: αν και δεν “αντιλαμβάνεται” το νόημα του περιεχομένου του τίτλου, η μηχανή ξέρει την έννοια “τίτλος”!
  - ▶ π.χ. έχει ιδιαίτερο βάρος στις μηχανές αναζήτησης

# Κλασσικά Web Δεδομένα

- ▶ Στο κλασσικό Web, τα δεδομένα είναι “εσωτερική υπόθεση” του παραγωγού τους
  - ▶ Αποθηκευμένα σε βάσεις δεδομένων απροσπέλαστες από τρίτους
  - ▶ Μετασχηματίζονται από τον ίδιο τον παραγωγό σε ιστοσελίδες
  - ▶ Ή χρησιμοποιούνται από web εφαρμογές που έχει γράψει ο ίδιος ο παραγωγός
  - ▶ Δεν χρειάζεται μετάδοση της σημασιολογίας των δεδομένων (είναι γνωστή εκ των προτέρων!)
- ▶ Μπορούν όμως τα δεδομένα αυτά να χρησιμοποιηθούν από τρίτους;
  - ▶ Σε εφαρμογές που δεν γνωρίζει ο παραγωγός των δεδομένων;
  - ▶ Από εφαρμογές που δεν σχεδιάστηκαν ειδικά για τα δεδομένα αυτά;

# Ανοικτά Σημασιολογικά Δεδομένα

- ▶ Πριν από κάθε άλλο, θα πρέπει φυσικά
  - ▶ τα δεδομένα να είναι απευθείας προσπελάσιμα στο Web
  - ▶ σε μια μορφή εύκολα επεξεργάσιμη από τις μηχανές
- ▶ Το κυριότερο όμως είναι να είναι ρητά δηλωμένη και προσπελάσιμη και η σημασιολογία των δεδομένων
  - ▶ Μόνο τότε θα μπορούν να χρησιμοποιηθούν από εφαρμογές που προσαρμόζουν τη συμπεριφορά τους ανάλογα με το τι σημαίνουν τα δεδομένα
  - ▶ εκτελώντας ενέργειες πάνω σε σετ δεδομένων για τα οποία δεν σχεδιάστηκαν εξ'αρχής

# Σημασιολογία και Μεταδεδομένα

- ▶ Μεταδεδομένα:
  - ▶ “δεδομένα για/σχετικά με/που αφορούν τα δεδομένα”
  - ▶ Είναι ακριβώς εκείνα που μεταδίδουν τη σημασιολογία των κανονικών δεδομένων
- ▶ Στις εφαρμογές του Σημασιολογικού Ιστού
  - ▶ Τα μεταδεδομένα μεταδίδονται ρητά μαζί με τα δεδομένα
  - ▶ Έχουν την ίδια αξία και είναι το ίδιο απαραίτητα με τα δεδομένα

# Δεδομένα σε Μορφή Πίνακα

- ▶ Η σημασιολογία των δεδομένων προκύπτει από τη θέση τους (γραμμή-στήλη)

Τύπος	Αριθμός
Λεωφορεία	58.519
ΙΧ	2.062.880
Οδοποιητικά	7.642
Εκχιονιστικά	6
Ποδήλατα	8.540.192

- ▶ Απλή μορφή στη δημιουργία και ενημέρωση
- ▶ Με πεπερασμένες όπως δυνατότητες στην εισαγωγή πληροφορίας που δεν “συμμορφώνεται” με τις προϋπάρχουσες γραμμές και στήλες
  - ▶ Πώς θα εκφράζατε π.χ. θέσεις επιβατών στα λεωφορεία;
  - ▶ Και τι θα βάζατε εκεί για τα άλλα οχήματα;



# Δοκιμάστε και εσείς!

- ▶ Διαλέξτε ένα ωρολόγιο πρόγραμμα στο τρέχον εξάμηνο
  - ▶ π.χ. του Η' εξαμήνου
- ▶ Προσπαθήστε να εκφράσετε την περιεχόμενη πληροφορία σε μορφή πίνακα
- ▶ **Προσοχή!** το ζητούμενο **δεν είναι** να αναπαράγετε τον πίνακα του ωρολογίου προγράμματος ως έχει!
  - ▶ Βρείτε την **κομβική οντότητα πληροφορίας** του ωρολογίου προγράμματος
  - ▶ Κάθε γραμμή του πίνακα θα είναι εκφράζει μια τέτοια οντότητα
  - ▶ Και η υπόλοιπη πληροφορία θα τοποθετηθεί στις στήλες, ως ιδιότητες κάθε κομβικής οντότητας
  - ▶ Το εξάμηνο σπουδών θα είναι και αυτό μια ιδιότητα

# Μοντέλο και Μορφή Αποθήκευσης

- ▶ Η μορφή πίνακα είναι ένα μοντέλο οργάνωσης δεδομένων (**data model**)
  - ▶ Προσδιορίζει τον τρόπο δόμησης της πληροφορίας
- ▶ Η δομημένη πληροφορία όμως
  - ▶ Πρέπει να **αποθηκευτεί** ως ακολουθία bytes
  - ▶ Και να **μεταδοθεί** μεταξύ παραγωγού και καταναλωτή
- ▶ Συνεπώς, πέρα από το μοντέλο των δεδομένων, είναι απαραίτητο ένα μορφότυπο σειριοποίησης (**serialization format**) των δεδομένων

# Η μορφή αποθήκευσης CSV

- ▶ Comma Separated Values

- ▶ Ένα ..μη πρότυπο πρότυπο (τουλάχιστον μέχρι πρόσφατα)
- ▶ Με πολλές “διαλέκτους” (σχεδόν κάθε εφαρμογή έχει τη δική της!)
- ▶ Για εξαγωγή-εισαγωγή μεγάλων σετ δεδομένων σε μορφή πίνακα από-σε βάσεις δεδομένων

- ▶ Έλεγχος Ιδιοτήτων:

- ▶ Ανοικτό πρότυπο: **NAI**
- ▶ Χρήση στο Web: **NAI** (αν και όχι τόσο συχνά)
- ▶ Ευκολία προγραμματισμού: **NAI** (βιβλιοθήκες για πολλές γλώσσες)
- ▶ Ρητά μεταδεδομένα: **OXI** (προαιρετικά, ονόματα στηλών στην πρώτη γραμμή)

# Δοκιμάστε και εσείς!

- ▶ **Βήμα 1<sup>ο</sup>:** Αποθηκεύστε τον πίνακα που φτιάξατε προηγουμένως σε μορφή csv
  - ▶ μέσω π.χ. εφαρμογής spreadsheet
- ▶ **Βήμα 2<sup>ο</sup>:** Ανταλλάξτε csv με έναν συμφοιτητή σας
  - ▶ Μπορείτε να κατανοήσετε την σημασιολογία των ξένων δεδομένων;
  - ▶ Τι χρειάζεται να μάθετε από τον παραγωγό των δεδομένων για να τα χρησιμοποιήσετε;

# Δοκιμάστε και εσείς!

- ▶ **Βήμα 3<sup>ο</sup>:** Γράψτε πρόγραμμα Python που διαβάζει το αρχείο csv και μπορεί να απαντήσει σε ερωτήματα όπως:
  - ▶ Τι διδάσκεται την ώρα/μέρα X στην αίθουσα Y;
  - ▶ Ποιες μέρες διδάσκει ο X το μάθημα Y;
  - ▶ Τι μαθήματα έχει σήμερα το εξάμηνο X;
  - ▶ Ποιος διδάσκει την ημέρα X στην αίθουσα Y;
  - ▶ κ.ο.κ
- ▶ Προσπαθήστε να σχεδιάσετε ένα γενικό interface ερωτημάτων!

# Python και CSV

Παράδειγμα ανάγνωσης αρχείου csv:

```
import csv

# open csv file for reading
ifp = open('test.csv', 'rb')
# create csv reader object
ir = csv.reader(ifp) # defaults to excel 'dialect'

# iterate over table rows in csv file
for row in ir:

    # each row is a list of strings
    # (table column values for this row)

    # do something with each row here...

ifp.close()
```

# Python και CSV

Παράδειγμα εγγραφής αρχείου csv:

```
import csv
```

```
# open csv file for writing
```

```
ofp = open('newfile.csv','wb')
```

```
# create csv writer object
```

```
ow = csv.writer(ofp)
```

```
# example: write a row in csv file
```

```
# row is a sequence of strings or numbers (or None)
```

```
ow.writerow(["abc","test","", "more"])
```

```
# after finishing, close output file
```

```
ofp.close()
```