

Γλωσσική Τεχνολογία

Εισαγωγικό Φροντιστήριο

Project του μαθήματος

- ▶ Εργασία 2 ατόμων
- ▶ Προφορική εξέταση για:
 - ▶ Project (80%)
 - ▶ Θεωρία (20%)
- ▶ Στο φροντιστήριο:
 - ▶ Συζητάμε σχεδιαστικές επιλογές
 - ▶ Λύνουμε ζητήματα υλοποίησης
 - ▶ Παρουσιάζουμε διαθέσιμα εργαλεία
 - ▶ Οι αλγόριθμοι και οι έννοιες παρουσιάζονται στο μάθημα
- ▶ Γλώσσα υλοποίησης της επιλογής σας αλλά:
 - ▶ Το φροντιστήριο θα γίνει σε Python 2.6.5
 - ▶ Θα σας διευκολύνουν εργαλεία που υπάρχουν σε Python 2.6.5



Γιατί Python;

- ▶ Εύκολη! Θα τη μάθετε αμέσως.
- ▶ Χρειάζεται να γράψετε πολύ λιγότερο κώδικα. (Ο χρόνος development είναι 10 φορές μικρότερος)
- ▶ Είναι scripting, παρόλα αυτά αρκετά γρήγορη. (Implemented in C)
- ▶ Ο κώδικας σε Python είναι μικρότερος και πιο «καθαρός», εύκολος να διαβαστεί και να κατανοηθεί. (Τα blocks κώδικα ορίζονται από κενά)
- ▶ Cross-Platform: Μπορείτε να προγραμματίσετε σε Windows ή Linux
- ▶ Υπάρχουν πολλά και δωρεάν διαθέσιμα resources στο δίκτυο για να διαβάσετε.
- ▶ Υπάρχει σε Python το NLTK (Natural Language Toolkit), το οποίο περιλαμβάνει ήδη υλοποιημένα εργαλεία για Γλωσσική Τεχνολογία!!!



Θέματα Φροντιστηρίου

- ▶ Εισαγωγή στην Python
- ▶ Διαχείριση strings
- ▶ Δομές Δεδομένων
- ▶ File I/O
- ▶ Κλήση διεργασιών
- ▶ Regular Expressions
- ▶ XML Handling
- ▶ Παρουσίαση NLTK και διαθέσιμων εργαλείων
- ▶ HTML Handling
- ▶ Επεξηγήσεις στην άσκηση



Python

- ▶ High Level
- ▶ Scripting
- ▶ Elegant Syntax
- ▶ Interpreted
- ▶ Object Oriented
- ▶ Functional
- ▶ Dynamic Typing
- ▶ Automatic Memory Management



My First Program

print “Hello World!”

Αντί για:

```
#include <stdio.h>
```

```
int main(int argc, char** argv)
```

```
{
```

```
    printf(“HelloWorld!\n”);
```

```
}
```



Python – Features I

▶ Data Types

- ▶ boolean, integer, float, long, complex
- ▶ Δεν ορίζονται ρητά. Καθορίζεται ο τύπος στον χρόνο εκτέλεσης.

▶ Sequence Types

- ▶ string, list, tuple

▶ Mapping Types

- ▶ dictionary



Python – Features II

- ▶ **Control Flow**

- ▶ if, for, while, break, continue

- ▶ **Functions**

- ▶ `def foo():`

- ▶ **Classes**

- ▶ `class foo():`

- ▶ **Modules**

- ▶ `from module import something`
- ▶ `import module`



Python – Features III

- ▶ Error/Exception Handling
- ▶ File I/O
- ▶ HTML/XML Parsing
- ▶ και πολλά περισσότερα..



Interactive Mode

File Edit View Terminal Help

```
nick@opportunity:~$ python
Python 2.6.4 (r264:75706, Nov  2 2009, 14:38:03)
[GCC 4.4.1] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> a = 2.0
>>> b = 3
>>> a - b
-1.0
>>> a**b
8.0
>>> a/b
0.66666666666666663
>>> for i in range(b):
...     print "τα σpy!!"
...
τα σpy!!
τα σpy!!
τα σpy!!
>>> □
```



Από που να διαβάσω;

- ▶ Python Documentation

- ▶ <http://docs.python.org/tutorial/index.html>

- ▶ Dive into Python

- ▶ <http://diveintopython.org/>

- ▶ TasPython

- ▶ <http://taspython.eu/>

- ▶ effbot.org

- ▶ <http://effbot.org/>

- ▶ Google

- ▶ <http://www.google.com>



Που να γράψω κώδικα;

▶ Editors

- ▶ Windows
 - ▶ Notepad etc.
- ▶ Linux
 - ▶ Gedit etc.

▶ IDEs

- ▶ Windows
 - ▶ Active Python
 - ▶ Netbeans
 - ▶ ...
- ▶ Linux
 - ▶ Eclipse
 - ▶ Netbeans
 - ▶ ...



Native Datatypes – Lists #1

- ▶ Η λίστες είναι ο τύπος δεδομένων που χρησιμοποιείται περισσότερο στην Python
- ▶ Είναι arrays, των οποίων το μέγεθος αλλάζει δυναμικά όσο προστίθενται στοιχεία.
- ▶ Τα στοιχεία δεν είναι απαραίτητο να είναι του ίδιου τύπου δεδομένων!

Ορισμός Λίστας

```
>>>a = ['spam', 'eggs', 100, 1234]
>>>a
['spam', 'eggs', 100, 1234]
```



Native Datatypes – Lists #2

Αναφορά στα στοιχεία της λίστας

- Index

`a[0], a[1], ..., a[len-2], a[len-1]`

- Negative index

`a[-len], a[-(len-1)], ..., a[-1]`

Προσθήκη στοιχείων

```
>>>a.append(333)
```

```
>>>a
```

```
['spam', 'eggs', 100, 1234, 333]
```

```
>>>a.insert(2,-1)
```

```
>>>a
```

```
['spam', 'eggs', -1, 100, 1234, 333]
```

```
>>>a.extend([1,2])
```

```
>>>a
```

```
['spam', 'eggs', -1, 100, 1234, 333, 1, 2]
```

Native Datatypes – Lists #3

Διαγραφή στοιχείων

```
>>>a.pop()
>>>a
['spam', 'eggs', -1, 100, 1234]
>>>a.remove(-1)
>>>a
['spam', 'eggs', 100, 1234]
>>>del a[1]
>>a
['spam', 100, 1234]
```

Αναζήτηση

```
>>>'spam' in a:
True
>>>'cat' in a:
False
```



Native Datatypes – Lists #4

Απαρίθμηση

```
>>>for i in a:  
...     print i  
...  
'spam'  
100  
1234  
>>>for i in range(len(a)):  
...     print i, a[i]  
...  
0,'spam'  
1,100  
2,1234
```



Native Datatypes - Dictionaries #1

- ▶ Τα dictionaries ορίζουν σχέσεις μεταξύ κλειδιών και τιμών.
- ▶ Πρόκειται για συλλογές εγγραφών που αποτελούνται από ένα κλειδί και την τιμή που αντιστοιχεί σε αυτό.

Ορισμός Dictionary

```
>>>tel = {'jack': 4098, 'sape': 4139}
>>>tel
{'jack': 4098, 'sape': 4139}
```

Αναφορά στα στοιχεία

```
>>>tel['jack']
4098
```



Native Datatypes - Dictionaries #2

Προσθήκη στοιχείων

```
>>> tel['guido'] = 4127
>>> tel
{'sape': 4139, 'guido': 4127, 'jack': 4098}
```

Διαγραφή στοιχείων

```
>>> del tel['sape']
>>> tel
{'guido': 4127, 'jack': 4098}
```



Native Datatypes - Dictionaries #3

Αναζήτηση

```
>>>'guido' in tel
```

```
True
```

```
>>>'sape' in tel
```

```
False
```

Απαρίθμηση

```
>>>for k,v in tel.items():
```

```
...     print k,v
```

```
...
```

```
'guido', 4127
```

```
'jack', 4098
```



Native Datatypes – Tuples #1

- ▶ Μια tuple είναι μια λίστα που δεν αλλάζει.
- ▶ Η τιμές που περιέχει μια tuple δεν μπορούν να αλλάξουν μετά τη δημιουργία της.

Ορισμός Tuple

```
>>> t = ("Mon", "Tue", "Thu", "Wed", "Fri", "Sat", "Sun")
>>> t
('Mon', 'Tue', 'Thu', 'Wed', 'Fri', 'Sat', 'Sun')
```

Αναφορά στα στοιχεία της tuple

- Index

`t[0],t[1],...,t[len-2], t[len-1]`

- Negative index

`t[-len],t[-(len-1)],..., t[-1]`

