ΔΙΑΧΕΙΡΙΣΗ ΠΕΡΙΕΧΟΜΕΝΟΥ ΠΑΓΚΟΣΜΙΟΥ ΙΣΤΟΥ ΚΑΙ ΓΛΩΣΣΙΚΑ ΕΡΓΑΛΕΙΑ

Εισαγωγικό Φροντιστήριο

Project του μαθήματος

- Εργασία 2 ατόμων
- Προφορική εξέταση για:
  - Project (80%)
  - Θεωρία (20%)
- Στο φροντιστήριο:
  - Ζητήματα σχεδιασμού
  - Παρουσίαση εργαλείων
- Γλώσσα υλοποίησης της επιλογής σας αλλά:
  - Το φροντιστήριο θα γίνει σε Python 2.x.x

"Although Python 3.0 is now available, NLTK has not yet been ported. For now you should use NLTK with Python 2.5., 2.6.*, or 2.7.* only. NLTK 3.0 will hopefully be ready during 2012."*

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 επεξεργασία φυσικής γλώσσας.

Βαθμολόγηση

- Η παράδοση του project θα γίνει τη μέρα της εξέτασης.
- Βαθμολογούνται:
  - Η ορθότητα της υλοποίησης (σωστά αποτελέσματα)
  - Η πληρότητα της υλοποίησης (όλα τα ζητούμενα)
  - Ο καλός σχεδιασμός
    - Τεκμηριωμένες σχεδιαστικές επιλογές
    - Ολοκληρωμένη εφαρμογή, σχεδιασμός κοντά σε πραγματικές ανάγκες
    - Εφαρμογή της θεωρίας, σωστή χρήση της θεωρίας
  - Καλή απόδοση!!! (μεγάλος όγκος δεδομένων εισόδου)
  - Να έχετε ασχοληθεί και να ξέρετε να απαντήσετε στις ερωτήσεις για το project.

Python

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

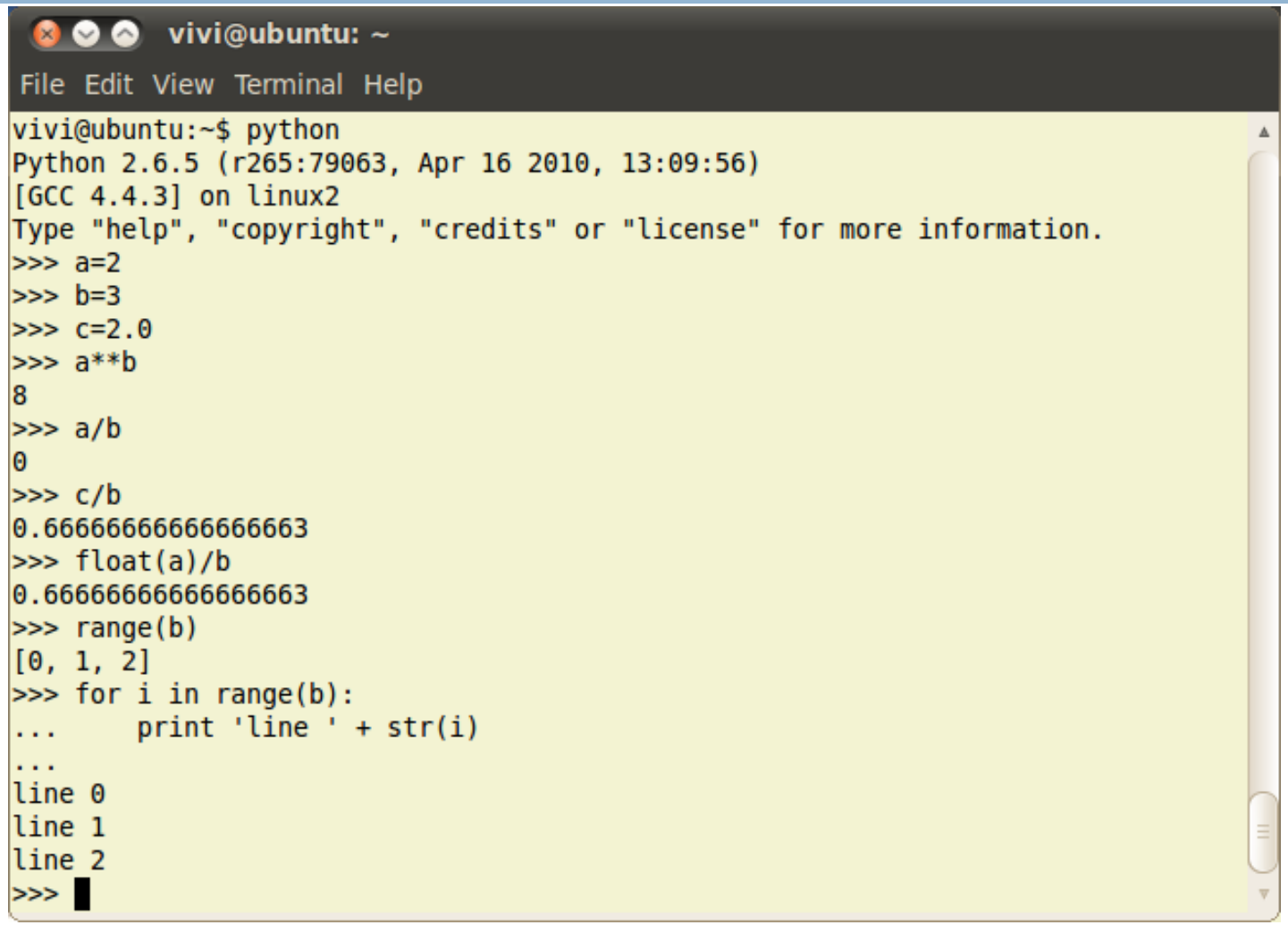
My First Program

print "Hello World!"

Αντί για:

```
#include <stdio.h>
int main(int argc, char** argv)
{
        printf("HelloWorld!\n");
}
```

Interactive Mode



Πηγές για Python

- Python Documentation
  - http://docs.python.org/tutorial/index.html
- Dive into Python
  - http://diveintopython.org/
- Ελληνική κοινότητα προγραμματιστών Python
  - http://python.org.gr
- effbot.org
  - http://effbot.org/
- Google
  - http://www.google.com

Editors

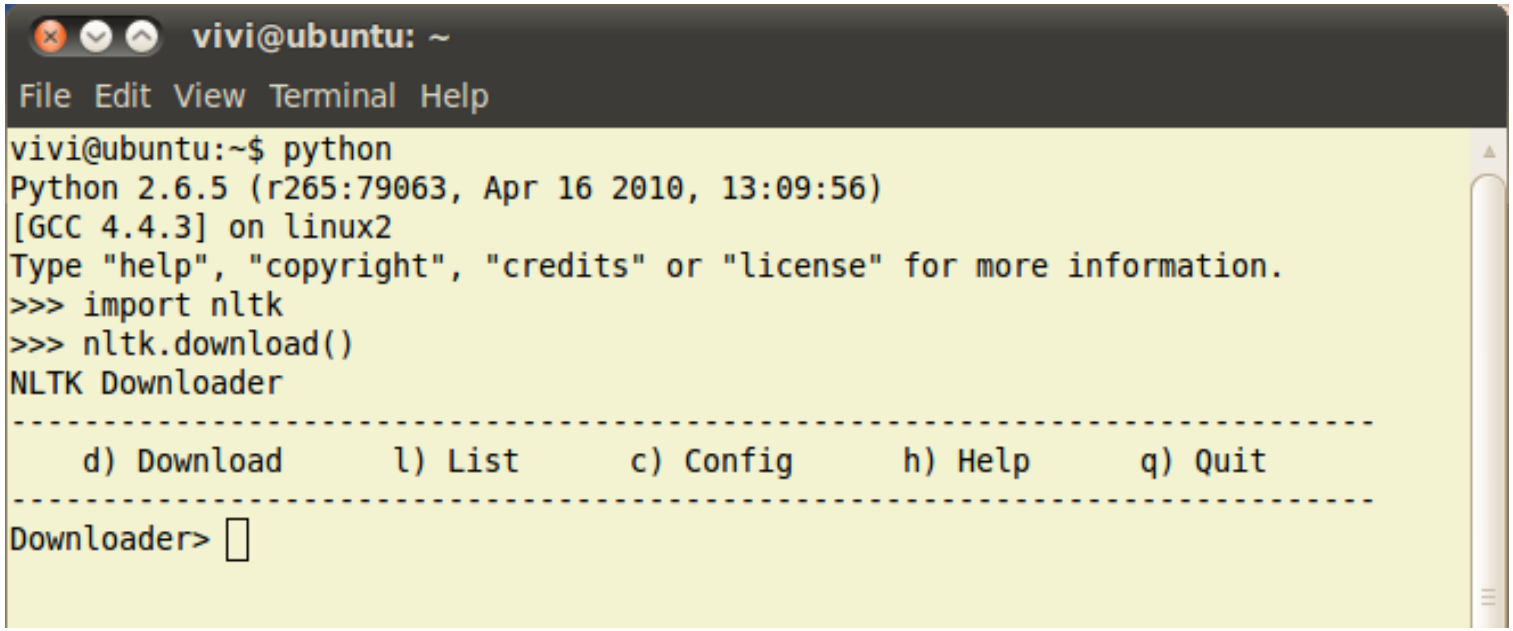
- Editors
  - Windows
    - Notepad etc.
  - Linux
    - Gedit etc.
- IDEs
  - Windows
    - Active Python
    - Netbeans
    - …
  - Linux
    - Eclipse
    - Netbeans
    - …

NLTK – Natural Language Toolkit

- Πακέτο βιβλιοθηκών και προγραμμάτων της Python για εφαρμογές Επεξεργασίας Φυσικής Γλώσσας.
- Χρησιμοποιείται ευρύτατα ως ερευνητικό εργαλείο στο πεδίο της υπολογιστικής γλωσσολογίας
- Περιλαμβάνει πολλά γνωστά corpora
- Πρέπει να το εγκαταστήσετε χωριστά
- http://www.nltk.org/
  - Download του NLTK & οδηγίες για εγκατάσταση
- http://www.nltk.org/book
  - το βιβλίο “Natural Language Processing with Python”
  - Περιλαμβάνει περιγραφή όλων των διαθέσιμων εργαλείων

Πρόσβαση στα resources

- Το NLTK με την εντολή download δίνει τη δυνατότητα εγκατάστασης διάφορων resources



Διαθέσιμα resources

- Μέρος της λίστας των διαθέσιμων:

```
Hit Enter to continue:
  [*] toolbox.............. Toolbox Sample Files
  [*] treebank............. Penn Treebank Sample
  [*] udhr................. Universal Declaration of Human Rights Corpus
  [*] unicode_samples..... Unicode Samples
  [*] webtext.............. Web Text Corpus
  [*] wordnet.............. WordNet
  [*] wordnet_ic........... WordNet-InfoContent
  [*] words................ Word Lists
  [*] book_grammars....... Grammars from NLTK Book
  [*] ycoe................. York-Toronto-Helsinki Parsed Corpus of Old
                            English Prose
  [*] basque_grammars..... Grammars for Basque
  [-] large_grammars...... Large context-free and feature-based grammars
                            for parser comparison
  [*] sample_grammars..... Sample Grammars
  [*] spanish_grammars.... Grammars for Spanish
  [*] tagsets.............. Help on Tagsets
  [*] maxent_treebank_pos_tagger Treebank Part of Speech Tagger (Maximum entropy
)
  [*] rslp................. RSLP Stemmer (Removedor de Sufixos da Lingua
                            Portuguesa)
  [*] hmm_treebank_pos_tagger Treebank Part of Speech Tagger (HMM)
Hit Enter to continue:
```

Βιβλίο

- “Natural Language Processing with Python”
- Μπορείτε να εγκαταστήσετε τις πηγές του βιβλίου:

```
Collections:
  [-] all-corpora..... All the corpora
  [-] all............. All packages
  [-] book............ Everything used in the NLTK Book

([*] marks installed packages; [-] marks out-of-date or corrupt packages)

---------------------------------------------------------------------------
    d) Download      l) List      c) Config      h) Help      q) Quit
---------------------------------------------------------------------------
Downloader> d

Download which package (l=list; x=cancel)?
  Identifier> book
```

Βιβλίο – Χρήση των πηγών

- Σε άγνωστα αντικείμενα θυμηθείτε το dir για να δουλέψετε

```
Type "help", "copyright", "credits" or "license" for more information.
>>> from nltk.book import *
*** Introductory Examples for the NLTK Book ***
Loading text1, ..., text9 and sent1, ..., sent9
Type the name of the text or sentence to view it.
Type: 'texts()' or 'sents()' to list the materials.
text1: Moby Dick by Herman Melville 1851
text2: Sense and Sensibility by Jane Austen 1811
text3: The Book of Genesis
text4: Inaugural Address Corpus
text5: Chat Corpus
text6: Monty Python and the Holy Grail
text7: Wall Street Journal
text8: Personals Corpus
text9: The Man Who Was Thursday by G . K . Chesterton 1908
>>> dir(text6)
['_CONTEXT_RE', '_COPY_TOKENS', '__class__', '__delattr__', '__dict__', '__doc__
', '__format__', '__getattribute__', '__getitem__', '__hash__', '__init__', '__l
en__', '__module__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__se
tattr__', '__sizeof__', '__str__', '__subclasshook__', '__weakref__', '_context'
, 'collocations', 'common_contexts', 'concordance', 'count', 'dispersion_plot',
'findall', 'generate', 'index', 'name', 'plot', 'readability', 'search', 'simila
r', 'tokens', 'vocab']
```

Βιβλίο – Μέτρηση συχνοτήτων

- Απόδοση?

```
>>> text6
<Text: Monty Python and the Holy Grail>
>>> text6.tokens[100:150]
['search', 'of', 'knights', 'who', 'will', 'join', 'me', 'in', 'my', 'court', 'at
', 'Camelot', '.', 'I', 'must', 'speak', 'with', 'your', 'lord', 'and', 'master',
 '.', 'SOLDIER', '#', '1', ':', 'What', '?', 'Ridden', 'on', 'a', 'horse', '?', '
ARTHUR', ':', 'Yes', '!', 'SOLDIER', '#', '1', ':', 'You', "'", 're', 'using', 'c
oconuts', '!', 'ARTHUR', ':', 'What']
>>> len(text6.tokens)
16967
>>> sorted(set(text6.tokens))[100:150]
['Are', 'Arimathea', 'Armaments', 'Arthur', 'As', 'Ask', 'Assyria', 'At', 'Attila
', 'Augh', 'Autumn', 'Auuuuuuuugh', 'Away', 'Ay', 'Ayy', 'B', 'BEDEVERE', 'BLACK'
, 'BORS', 'BRIDE', 'BRIDGEKEEPER', 'BROTHER', 'Back', 'Bad', 'Badon', 'Battle', '
Be', 'Beast', 'Bedevere', 'Bedwere', 'Behold', 'Between', 'Beyond', 'Black', 'Blo
ody', 'Blue', 'Bon', 'Bones', 'Book', 'Bors', 'Brave', 'Bravely', 'Bravest', 'Bre
ad', 'Bridge', 'Bring', 'Bristol', 'Britain', 'Britons', 'Brother']
>>> len(set(text6.tokens))
2166
>>> len(text6.vocab())
2166
>>> text6.tokens.count('coconut')
6
>>>
```

Βιβλίο – Frequency Distribution

- FreqDist
  - Δέχεται ως είσοδο τη λίστα από tokens
  - Δίνει ένα dictionary με key το token και value τη συχνότητα εμφάνισής του

```
>>> freq=FreqDist(text6)
>>> freq
<FreqDist with 16967 outcomes>
>>> sorted(freq.keys())[100:150]
['Are', 'Arimathea', 'Armaments', 'Arthur', 'As', 'Ask', 'Assyria', 'At', 'Attila
', 'Augh', 'Autumn', 'Auuuuuuuugh', 'Away', 'Ay', 'Ayy', 'B', 'BEDEVERE', 'BLACK'
, 'BORS', 'BRIDE', 'BRIDGEKEEPER', 'BROTHER', 'Back', 'Bad', 'Badon', 'Battle', '
Be', 'Beast', 'Bedevere', 'Bedwere', 'Behold', 'Between', 'Beyond', 'Black', 'Blo
ody', 'Blue', 'Bon', 'Bones', 'Book', 'Bors', 'Brave', 'Bravely', 'Bravest', 'Bre
ad', 'Bridge', 'Bring', 'Bristol', 'Britain', 'Britons', 'Brother']
>>> len(freq.keys())
2166
>>> freq['coconut']
6
```

Βιβλίο – Concordances

```
>>> text1.concordance("monstrous")
Building index...
Displaying 11 of 11 matches:
ong the former , one was of a most monstrous size . ... This came towards us ,
ON OF THE PSALMS . " Touching that monstrous bulk of the whale or ork we have r
ll over with a heathenish array of monstrous clubs and spears . Some were thick
d as you gazed , and wondered what monstrous cannibal and savage could ever hav
that has survived the flood ; most monstrous and most mountainous ! That Himmal
they might scout at Moby Dick as a monstrous fable , or still worse and more de
th of Radney .'" CHAPTER 55 Of the monstrous Pictures of Whales . I shall ere l
ing Scenes . In connexion with the monstrous pictures of whales , I am strongly
ere to enter upon those still more monstrous stories of them which are to be fo
ght have been rummaged out of this monstrous cabinet there is no telling . But
of Whale - Bones ; for Whales of a monstrous size are oftentimes cast up dead u
>>> text2.concordance("monstrous")
Building index...
Displaying 11 of 11 matches:
. " Now , Palmer , you shall see a monstrous pretty girl ." He immediately went
your sister is to marry him . I am monstrous glad of it , for then I shall have
ou may tell your sister . She is a monstrous lucky girl to get him , upon my ho
k how you will like them . Lucy is monstrous pretty , and so good humoured and
 Jennings , " I am sure I shall be monstrous glad of Miss Marianne ' s company
 usual noisy cheerfulness , " I am monstrous glad to see you -- sorry I could n
t however , as it turns out , I am monstrous glad there was never any thing in
so scornfully ! for they say he is monstrous fond of her , as well he may . I s
possible that she should ." " I am monstrous glad of it . Good gracious ! I hav
thing of the kind . So then he was monstrous happy , and talked on some time ab
e very genteel people . He makes a monstrous deal of money , and they keep thei
```

Βιβλίο – Concordances

- Από το περιβάλλον συνεμφάνισης μπορούμε να αντλήσουμε στοιχεία για τη σημασιολογία των λέξεων:

```
>>> text1.similar('monstrous')
Building word-context index...
abundant candid careful christian contemptible curious delightfully
determined doleful domineering exasperate fearless few gamesome
horrible impalpable imperial lamentable lazy loving
>>> text2.similar('monstrous')
Building word-context index...
very exceedingly heartily so a amazingly as extremely good great
remarkably sweet vast
>>> text2.common_contexts(["monstrous","very"])
a_lucky a_pretty am_glad be_glad is_pretty
>>>
```

Webtext

- Συλλογή κειμένων από το διαδίκτυο
- Τι διαφορές έχουν από τα υπόλοιπα κείμενα?

```
>>> dir(webtext)
['CorpusView', '__class__', '__delattr__', '__dict__', '__doc__', '__format__'
, '__getattribute__', '__hash__', '__init__', '__module__', '__new__', '__redu
ce__', '__reduce_ex__', '__repr__', '__setattr__', '__sizeof__', '__str__', '_
_subclasshook__', '__weakref__', '_encoding', '_fileids', '_get_items', '_get_
root', '_para_block_reader', '_read_para_block', '_read_sent_block', '_read_wo
rd_block', '_root', '_sent_tokenizer', '_tag_mapping_function', '_word_tokeniz
er', 'abspath', 'abspaths', 'encoding', 'fileids', 'files', 'items', 'open', '
paras', 'raw', 'read', 'readme', 'root', 'sents', 'tokenized', 'words']
>>> webtext.files()
['firefox.txt', 'grail.txt', 'overheard.txt', 'pirates.txt', 'singles.txt', 'w
ine.txt']
>>> len(webtext.sents())
25776
>>> len(webtext.words())
396736
>>> webtext.words().count('site')
176
```

Κείμενα από τον Παγκόσμιο Ιστό

- Προκλήσεις στην επεξεργασία κειμένων από τον Παγκόσμιο Ιστό:
  - Τεράστιος όγκος δεδομένων
  - Συνεχής αύξηση των δεδομένων
  - Πολλές γλώσσες
  - Κείμενα χαμηλής ποιότητας (πχ ασύντακτα και ανορθόγραφα)
  - Html μορφή και προβλήματα στην επεξεργασία της (not well-formed)
  - Ιδιαιτερότητες στην επικοινωνία:
    - Transliteration (greeklish, romanization etc.)
    - Internet Acronyms(afk, lol, btw, twot etc.)
    - Internet slang(noob, troll, fail etc.)

Penn Treebank Corpus Sample

```
>>> dir(treebank)
['__class__', '__delattr__', '__dict__', '__doc__', '__format__', '__getattrib
ute__', '__hash__', '__init__', '__module__', '__new__', '__reduce__', '__redu
ce_ex__', '__repr__', '__setattr__', '__sizeof__', '__str__', '__subclasshook_
_', '__weakref__', '_comment_char', '_detect_blocks', '_encoding', '_fileids',
 '_get_items', '_get_root', '_normalize', '_parse', '_read_block', '_read_pars
ed_sent_block', '_read_sent_block', '_read_tagged_sent_block', '_read_tagged_w
ord_block', '_read_word_block', '_root', '_tag', '_tag_mapping_function', '_wo
rd', 'abspath', 'abspaths', 'encoding', 'fileids', 'files', 'items', 'open', '
parsed', 'parsed_sents', 'raw', 'read', 'readme', 'root', 'sents', 'tagged', '
tagged_sents', 'tagged_words', 'tokenized', 'words']
>>> treebank.sents()
[['Pierre', 'Vinken', ',', '61', 'years', 'old', ',', 'will', 'join', 'the', '
board', 'as', 'a', 'nonexecutive', 'director', 'Nov.', '29', '.'], ['Mr.', 'Vi
nken', 'is', 'chairman', 'of', 'Elsevier', 'N.V.', ',', 'the', 'Dutch', 'publi
shing', 'group', '.'], ...]
>>> treebank.tagged_sents()
[[('Pierre', 'NNP'), ('Vinken', 'NNP'), (',', ','), ('61', 'CD'), ('years', 'N
NS'), ('old', 'JJ'), (',', ','), ('will', 'MD'), ('join', 'VB'), ('the', 'DT')
, ('board', 'NN'), ('as', 'IN'), ('a', 'DT'), ('nonexecutive', 'JJ'), ('direct
or', 'NN'), ('Nov.', 'NNP'), ('29', 'CD'), ('.', '.')], [('Mr.', 'NNP'), ('Vin
ken', 'NNP'), ('is', 'VBZ'), ('chairman', 'NN'), ('of', 'IN'), ('Elsevier', 'N
NP'), ('N.V.', 'NNP'), (',', ','), ('the', 'DT'), ('Dutch', 'NNP'), ('publishi
ng', 'VBG'), ('group', 'NN'), ('.', '.')], ...]
```

Penn Treebank Corpus Sample

- Penn Treebang Tagset: το tagset του Treetagger

```
>>> treebank.tagged_words()
[('Pierre', 'NNP'), ('Vinken', 'NNP'), (',', ','), ...]
>>> tags=[t for w,t in treebank.tagged_words()]
>>> len(tags)
100676
>>> sorted(set(tags))
['#', '$', "''", ',', '-LRB-', '-NONE-', '-RRB-', '.', ':', 'CC', 'CD', 'DT',
'EX', 'FW', 'IN', 'JJ', 'JJR', 'JJS', 'LS', 'MD', 'NN', 'NNP', 'NNPS', 'NNS',
'PDT', 'POS', 'PRP', 'PRP$', 'RB', 'RBR', 'RBS', 'RP', 'SYM', 'TO', 'UH', 'VB'
, 'VBD', 'VBG', 'VBN', 'VBP', 'VBZ', 'WDT', 'WP', 'WP$', 'WRB', '``']
>>> len(set(tags))
46
>>>
```

Brown Corpus

- Ξεκίνησε τη δεκαετία του ’60 στο Brown University
- Συλλογή αντιπροσωπευτικών κειμένων της αγγλικής
- Αποτέλεσε τη βάση για πολλά μορφοσυντακτικά σχολιασμένα corpora
- Το Brown Corpus αποτελεί ιστορικό κομμάτι της υπολογιστικής γλωσσολογίας

```
>>> from nltk.corpus import brown
>>> dir(brown)
['__class__', '__delattr__', '__dict__', '__doc__', '__format__', '__getattribute
__', '__hash__', '__init__', '__module__', '__new__', '__reduce__', '__reduce_ex_
_', '__repr__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', '__we
akref__', '_add', '_f2f', '_delimiter', '_encoding', '_f2c', '_file', '_fileids',
 '_get_items', '_get_root', '_init', '_map', '_para_block_reader', '_pattern', '_
resolve', '_root', '_sent_tokenizer', '_sep', '_tag_mapping_function', '_word_tok
enizer', 'abspath', 'abspaths', 'categories', 'encoding', 'fileids', 'files', 'it
ems', 'open', 'paras', 'raw', 'readme', 'root', 'sents', 'tagged_paras', 'tagged_
sents', 'tagged_words', 'words']
>>> len(brown.words())
1161192
>>> len(brown.sents())
57340
```

Brown Corpus

- Περιέχει κείμενα ταξινομημένα σε κατηγορίες
- Επιτρέπει ανάκτηση λέξεων και προτάσεων ανά κατηγορία

```
>>> brown.categories()
['adventure', 'belles_lettres', 'editorial', 'fiction', 'government', 'hobbies',
'humor', 'learned', 'lore', 'mystery', 'news', 'religion', 'reviews', 'romance',
'science_fiction']
>>> brown.words(categories='science_fiction')
['Now', 'that', 'he', 'knew', 'himself', 'to', 'be', ...]
>>> brown.sents(categories='science_fiction')
[['Now', 'that', 'he', 'knew', 'himself', 'to', 'be', 'self', 'he', 'was', 'free'
, 'to', 'grok', 'ever', 'closer', 'to', 'his', 'brothers', ',', 'merge', 'without
', 'let', '.'], ["Self's", 'integrity', 'was', 'and', 'is', 'and', 'ever', 'had',
 'been', '.'], ...]
>>>
```

Brown Corpus – Παράδειγμα χρήσης

- Εύρεση του λεξιλογίου που συναντάται στα κείμενα επιστημονικής φαντασίας

```
>>> swords = brown.tagged_words(categories='science_fiction')
>>> nouns=[w for w,t in swords if t.startswith('N')]
>>> nouns[100:150]
['Mike', 'summer', 'floods', 'Mike', 'tray', 'darkness', 'Jubal', 'night-sight',
'conditions', 'Mike', 'parents', 'night', 'Mount', 'Everest', 'water', 'brothers'
, 'tolerance', 'changes', 'temperature', 'pressure', 'weakness', 'snow', 'crystal
', 'water', 'life', 'individual', 'meantime', 'night', 'company', 'water', 'broth
er', 'tray', 'lights', 'Mike', 'light', 'ripples', 'goodness', 'beauty', 'pool',
'grass', 'stars', 'Mike', 'Mars', 'Mars', 'Antares', 'Mars', 'Mike', 'Mars', 'que
stion', 'language']
>>> freq=FreqDist(nouns)
>>> sfreq=sorted(freq, key=freq.get, reverse=True)
>>> sfreq[0:50]
['time', 'Ekstrohm', 'Helva', 'Mercer', 'Hal', "B'dikkat", 'Mike', 'ship', 'peopl
e', 'Jack', 'man', 'years', 'Earth', 'course', 'head', 'Ryan', 'mind', 'way', 'He
sperus', 'light', 'Gabriel', 'bodies', 'night', 'planet', 'Jubal', 'Mars', 'eyes'
, 'half-man', 'thing', 'voice', 'Angels', "Helva's", 'Nogol', 'brain', 'face', 'k
ind', 'place', 'words', 'Da', 'Digby', 'Lady', 'Macneff', 'Siddo', 'captain', 'ga
pt', 'hands', 'pain', 'power', 'problem', 'space']
>>>
```