



python: a dynamic
object-oriented
scripting language



Τμήμα Πληροφορικής Πανεπιστημίου Κύπρου
ΕΠΛ 371: Προγραμματισμός Συστημάτων

Βέης Πασχάλης
Ιωακείμ Ευστάθιος



Python: Outline

- **Ιστορική Αναδρομή**
- Πλεονεκτήματα / Μειονεκτήματα
- Εγκατάσταση
- Δυνατότητες:
 - Γενικά
 - Classes + Modules
 - File I/O
 - Error handling
 - Modules for URLs
 - Directories
 - Regular expressions
- BIBEL Bibliography Tool with Python
- Συμπεράσματα
- Βιβλιογραφία



Python: Ιστορική Αναδρομή

- Ύπαρξη ως ιδέα από τα τέλη του 1980
- Εμφάνιση της Python το 1991 (*Guido van Rossum*)
- Ονομασία από τηλεοπτική σειρά “Monty **Python**'s Flying Circus”
- Επηρεασμένη από C, C++, Java, Lisp, Modula-3, Perl, ...
- Επηρέασε Boo, Falcon, JavaScript, Ruby, Combra, D
- PSF License – συμβατό με GNU (GPL)



“Python is an interpreted, general-purpose high-level programming language whose design philosophy emphasizes code readability” - *wikipedia*





Python: Outline

- Ιστορική Αναδρομή
- **Πλεονεκτήματα / Μειονεκτήματα**
- Εγκατάσταση
- Δυνατότητες:
 - Γενικά
 - Classes + Modules
 - File I/O
 - Error handling
 - Modules for URLs
 - Directories
 - Regular expressions
- BIBEL Bibliography Tool with Python
- Συμπεράσματα
- Βιβλιογραφία



Python: Πλεονεκτήματα

- Ευκολία εκμάθησης (προηγούμενη γνώση C, C++, Java, ...)
- Καθαρός – ευανάγνωστος κώδικας
- Ταχύτητα στην ανάπτυξη προγραμμάτων
- Επαναχρησιμοποίηση κώδικα
- Portability
- Αντικειμενοστραφείς έννοιες
- Υποστήριξη από Python community
- Δωρεάν



Python: Μειονεκτήματα

- Interpreted language
- πιο αργή κατά την εκτέλεση (σε σχέση με C, C++, Java)
- Χρονοβόρα διαδικασία η μετάφραση του κώδικα σε άλλη γλώσσα προγραμματισμού
- Μη κατάλληλη για προγράμματα που κάνουν σημαντική χρήση της μνήμης



Python: Outline

- Ιστορική Αναδρομή
- Πλεονεκτήματα / Μειονεκτήματα
- **Εγκατάσταση**
- Δυνατότητες:
 - Γενικά
 - Classes + Modules
 - File I/O
 - Error handling
 - Modules for URLs
 - Directories
 - Regular expressions
- BIBEL Bibliography Tool with Python
- Συμπεράσματα
- Βιβλιογραφία



Python: Εγκατάσταση

Δυνατότητα εγκατάστασης σε πληθώρα πλατφόρμων:

Microsoft Windows, Macintosh, AIX, AROS, AS/400 (OS/400), BeOS, iPod, MorphOS, MS-DOS, OS/2, Palm OS, PlayStation and PSP, QNX, RISC OS (what used to be Acorn), Series 60, Solaris, VMS, Pocket PC, ...

Για Microsoft Windows:

- Κατέβασμα του Windows Msi Installer απο
<http://www.python.org/download/>
- Εκκίνηση του Msi Installer
- Ρύθμιση της μεταβλητής περιβάλλοντος PATH ώστε να συμπεριλαμβάνει τον root folder του αρχείου που εγκαταστάθηκε η Python (default location is `c:\Python##`)

Προσθήκη `;c:\Python##` στην μεταβλητή PATH



Python: Εγκατάσταση

Για Unix-Linux:

(manual installation)

Κατέβασμα του Gzipped source tar ball απο <http://www.python.org/download/releases>

Εκτέλεση των πιο κάτω εντολών:

```
bash$ tar -xzf Python-#.#.tgz
```

```
bash$ cd Python-#.#
```

```
bash$ sudo ./configure
```

```
bash$ sudo make
```

```
bash$ sudo make install
```

(automatic installation)

Απλούστερη εγκατάσταση μέσω της εντολής:

```
bash$ sudo apt-get install python
```

(automatic installation)

Πιο απλά μέσω του εργαλείου Synaptic Package manager (γραφικό περιβάλλον)

System > Administration > Synaptic Package manager



Python: Εγκατάσταση

Εκτέλεση του Hello world! : (Linux)

```
etagrats@Teldrasil: ~  
File Edit View Search Terminal Help  
etagrats@Teldrasil:~$ cat example.py  
print "Hello World!"  
  
etagrats@Teldrasil:~$ python example.py  
Hello World!  
etagrats@Teldrasil:~$  
  
etagrats@Teldrasil: ~  
File Edit View Search Terminal Help  
etagrats@Teldrasil:~$ python  
Python 2.6.6 (r266:84292, Sep 15 2010, 16:22:56)  
[GCC 4.4.5] on linux2  
Type "help", "copyright", "credits" or "quit()">  
>>> print "Hello World!"  
Hello World!  
>>>  
  
etagrats@Teldrasil: ~  
File Edit View Search Terminal Help  
etagrats@Teldrasil:~$ cat ex2.py  
#!/usr/bin/python  
print "Hello world!"  
  
etagrats@Teldrasil:~$ ./ex2.py  
Hello world!  
etagrats@Teldrasil:~$
```

[Ctrl + D]
to exit python

(windows)

```
Administrator: C:\Windows\system32\cmd.exe  
C:\Users\Etagrats>python example02.py  
Hello World!  
  
C:\Users\Etagrats>  
  
Administrator: C:\Windows\system32\cmd.exe -  
C:\Users\Etagrats>python  
Python 2.2.3 (#42, May 30 2003, 18:  
Type "help", "copyright", "credits"  
>>> print "Hello World!"  
Hello World!  
>>>
```

Σημ: Απο την έκδοση Python3.# και μετά η εντολή `print "Hello World"` δεν είναι συμβατή
Συμβατή εντολή είναι `print("Hello World")`



Python: Outline

- Ιστορική Αναδρομή
- Πλεονεκτήματα / Μειονεκτήματα
- Εγκατάσταση
- **Δυνατότητες:**
 - Γενικά
 - Classes + Modules
 - File I/O
 - Error handling
 - Modules for URLs
 - Directories
 - Regular expressions
- BIBEL Bibliography Tool with Python
- Συμπεράσματα
- Βιβλιογραφία



Python: Δυνατότητες

Γενικά περί Python:

Ομοιότητες με C, C++, Java, ...

C

για την εκτέλεση πλήθους
εντολών αν κάποια συνθήκη
είναι αληθής προϋποθέτει ότι
οι εντολές **εσωκλείονται**
σε { } και **δεν απαιτείται**
στοίχιση του κώδικα

≠

Python

για την εκτέλεση πλήθους
εντολών αν κάποια συνθήκη
είναι αληθής προϋποθέτει ότι
οι εντολές **δεν εσωκλείονται**
σε { } και **απαιτείται**
στοίχιση του κώδικα

Πιο ευανάγνωστος ο κώδικας σε Python



Python: Outline

- Ιστορική Αναδρομή
- Πλεονεκτήματα / Μειονεκτήματα
- Εγκατάσταση
- **Δυνατότητες:**
 - Γενικά
 - **Classes + Modules**
 - File I/O
 - Error handling
 - Modules for URLs
 - Directories
 - Regular expressions
- BIBEL Bibliography Tool with Python
- Συμπεράσματα
- Βιβλιογραφία



Python: Δυνατότητες

Classes + Modules

```
1 #####
2 ### module01.py ###
3 #####
4
5 #a class that defines a person
6 #a person has name and age
7 #a person can speak
8 class Person:
9
10     # this function acts as a constructor
11     def __init__(self, name, age):
12         self.name = name
13         self.age = age
14
15     # function that allows a person to
16     # tell us his name and age
17     def speak(self):
18         print "My name is ", self.name,
19         print " and i am ", self.age, " years old."
20
```

```
1 #####
2 ### example01.py ###
3 #####
4
5 # import all classes located in module01.py
6 from module01 import *
7
8 # a class that defines a Student
9 # this class inherits from class Person
10 class Student(Person):
11
12     # this function acts like a constructor
13     def __init__(self, name, age, studies):
14         Person.__init__(self, name, age)
15         self.studies = studies
16
17     # function that allows a student
18     # to tell us his name, age, studies
19     def speak(self):
20         Person.speak(self)
21         print "I study ", self.studies, "."
22
23 #####
24 ### main body #####
25 #####
26 guy = Student("Giannakis", 21, "Computer Science")
27 guy.speak()
28
```



Python: Δυνατότητες

Classes + Modules

Εκτέλεση κώδικα:

```
etagrats@Teldrasil: ~/python classes
File Edit View Search Terminal Help
etagrats@Teldrasil:~/python classes$ ls
example01.py module01.py
etagrats@Teldrasil:~/python classes$ python example01.py
My name is Giannakis and i am 21 years old.
I study Computer Science .
etagrats@Teldrasil:~/python classes$
```



Python: Outline

- Ιστορική Αναδρομή
- Πλεονεκτήματα / Μειονεκτήματα
- Εγκατάσταση
- **Δυνατότητες:**
 - Γενικά
 - Classes + Modules
 - **File I/O**
 - Error handling
 - Modules for URLs
 - Directories
 - Regular expressions
- BIBEL Bibliography Tool with Python
- Συμπεράσματα
- Βιβλιογραφία



Python: Δυνατότητες

File I/O

```
1 #####
2 ### FileIOexample.py ###
3 #####
4
5 # define the name of the output file
6 file_name = "dataFile.dat"
7
8 # create a list to temporary store words
9 word_list = []
10
11 # inform user how to terminate the loop
12 print "\nTo terminate and store words, type \"exit\""
13
14 # get words from user and store them to a list
15 while 1 > 0:
16     word = raw_input("Give a word: ")
17     if word == "exit":
18         break
19     else:
20         word_list.append(word)
21
22 # create a file object with write permission
23 # "open the file to write"
24 outp = open(file_name, "w")
25
26 # write the words to file
27 for word in word_list:
28     outp.write(word + "\n")
29
30 # close the file
31 outp.close()
32 |
```

```
33 # open file to read
34 inp = open(file_name, "r")
35
36 # read all lines and store them to a list
37 lines = inp.readlines()
38
39 # appear contents of list
40 print "\nThe words you entered where:"
41 for line in lines:
42     print line,
43
44 # close the file
45 inp.close()
46
```



Python: Δυνατότητες

File I/O

Εκτέλεση κώδικα:

```
etagrats@Teldrasil: ~/python fileIO
File Edit View Search Terminal Help
etagrats@Teldrasil:~/python fileIO$ ls
FileIOexample.py
etagrats@Teldrasil:~/python fileIO$ python FileIOexample.py

To terminate and store words, type "exit"
Give a word: one
Give a word: two
Give a word: three
Give a word: exit

The words you entered where:
one
two
three
etagrats@Teldrasil:~/python fileIO$ ls
dataFile.dat FileIOexample.py
etagrats@Teldrasil:~/python fileIO$ cat dataFile.dat
one
two
three
etagrats@Teldrasil:~/python fileIO$
```



Python: Outline

- Ιστορική Αναδρομή
- Πλεονεκτήματα / Μειονεκτήματα
- Εγκατάσταση
- **Δυνατότητες:**
 - Γενικά
 - Classes + Modules
 - File I/O
 - **Error handling**
 - Modules for URLs
 - Directories
 - Regular expressions
- BIBEL Bibliography Tool with Python
- Συμπεράσματα
- Βιβλιογραφία



Python: Δυνατότητες

Error handling

```
1 #####
2 ### noErrorHandl.py ###
3 #####
4
5 # ask user to give two numbers
6 # in order to subtract them
7 a = input('Enter a number to subtract from > ')
8 b = input('Enter the number to subtract > ')
9 print a, " - ", b, " = ",
10 print a - b
11
```

```
1 #####
2 ### withErrorHandl.py ###
3 #####
4
5 # ask user to give two numbers
6 # in order to subtract them
7 flag = 1
8 while flag == 1:
9
10     # input error handling with try-except
11     try:
12         a = input('Enter a number to subtract from > ')
13         b = input('Enter the number to subtract > ')
14     except NameError:
15         print "\nYou cannot subtract a letter"
16         continue
17
18     print a, " - ", b, " = ",
19     print a - b
20
21     # input error handling with try-except
22     try:
23         flag = input('\nPress 1 to try again > ')
24     except NameError:
25         flag = 0
26
```



Python: Δυνατότητες

Error handling

```
etagrats@Telrasil: ~/ErrHandl
File Edit View Search Terminal Help
etagrats@Telrasil:~/ErrHandl$ python noErrorHandl.py
Enter a number to subtract from > 2
Enter the number to subtract > 3
2 - 3 = -1
etagrats@Telrasil:~/ErrHandl$ python noErrorHandl.py
Enter a number to subtract from > 1
Enter the number to subtract > e
Traceback (most recent call last):
  File "noErrorHandl.py", line 8, in <module>
    b = input('Enter the number to subtract > ')
  File "<string>", line 1, in <module>
NameError: name 'e' is not defined
etagrats@Telrasil:~/ErrHandl$ python noErrorHandl.py
Enter a number to subtract from > e
Traceback (most recent call last):
  File "noErrorHandl.py", line 7, in <module>
    a = input('Enter a number to subtract from > ')
  File "<string>", line 1, in <module>
NameError: name 'e' is not defined
etagrats@Telrasil:~/ErrHandl$
```

```
etagrats@Telrasil: ~/ErrHandl
File Edit View Search Terminal Help
etagrats@Telrasil:~/ErrHandl$ python withErrorHandl.py
Enter a number to subtract from > 2
Enter the number to subtract > 3
2 - 3 = -1

Press 1 to try again > 1
Enter a number to subtract from > 1
Enter the number to subtract > e

You cannot subtract a letter
Enter a number to subtract from > e

You cannot subtract a letter
Enter a number to subtract from > 4
Enter the number to subtract > 5
4 - 5 = -1

Press 1 to try again > w
etagrats@Telrasil:~/ErrHandl$
```



Python: Outline

- Ιστορική Αναδρομή
- Πλεονεκτήματα / Μειονεκτήματα
- Εγκατάσταση
- **Δυνατότητες:**
 - Γενικά
 - Classes + Modules
 - File I/O
 - Error handling
 - **Modules for URLs**
 - Directories
 - Regular expressions
- BIBEL Bibliography Tool with Python
- Συμπεράσματα
- Βιβλιογραφία



Python: Δυνατότητες

Module for URLs

```
1 #####
2 ### examp01.py ###
3 #####
4
5 # extensible library for opening URLs
6 import urllib
7
8 # create a file handler for the specific url
9 filehandle = urllib.urlopen('http://www.informatik.uni-
   trier.de/~ley/db/conf/indexa.html')
10
11 # open a file to store the url
12 myFile = open('test.html','w')
13
14 # read url content from url-handler
15 for lines in filehandle.readlines():
16     #store content to local file
17     myFile.write(lines)
18
19 #close local file
20 myFile.close()
21
22 #close url handler
23 filehandle.close()
24
```



Python: Δυνατότητες

Module for URLs

```
etagrats@Teldrasil: ~/python urllib
File Edit View Search Terminal Help
etagrats@Teldrasil:~/python urllib$ ls
examp01.py
etagrats@Teldrasil:~/python urllib$ python examp01.py
etagrats@Teldrasil:~/python urllib$ ls
examp01.py test.html
etagrats@Teldrasil:~/python urllib$ cat test.html
<html><head><title>DB&LP: Conferences & Workshops</title><link href="../../dblp.css" r
el="stylesheet" type="text/css" /><link href="../../favicon.ico" rel="shortcut icon" type="ima
ge/x-icon" /></head>
<body>
<table width="100%"><tr><td align="left"><a href="../index.html"></a></td>
<td align="right"><a href="http://www.uni-trier.de"></a></td></tr></table>

<h1>Computer Science Conferences & Workshops</h1>

<h2>DBLP:</h2>
<h3><a href="index-a.html">A</a> -
<a href="index-b.html">B</a> -
<a href="index-c.html">C</a> -
<a href="index-d.html">D</a> -
<a href="index-e.html">E</a> -
```




Python: Outline

- Ιστορική Αναδρομή
- Πλεονεκτήματα / Μειονεκτήματα
- Εγκατάσταση
- **Δυνατότητες:**
 - Γενικά
 - Classes + Modules
 - File I/O
 - Error handling
 - Modules for URLs
 - **Directories**
 - Regular expressions
- BIBEL Bibliography Tool with Python
- Συμπεράσματα
- Βιβλιογραφία

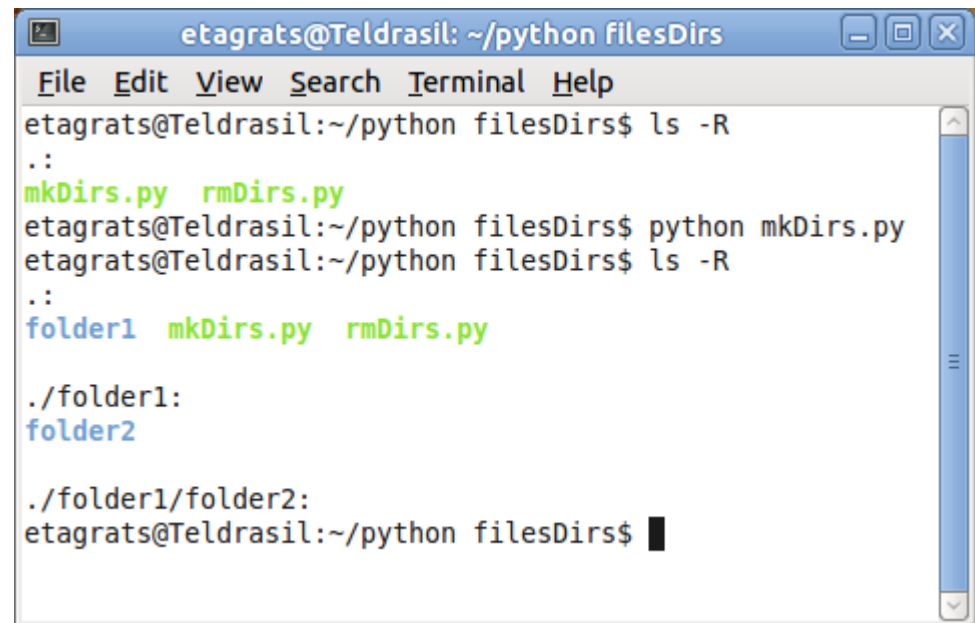


Python: Δυνατότητες

Directories

Creating directories:

```
1 #####
2 ### mkDirs.py ###
3 #####
4
5 # include os module in order
6 # to access system functions
7 import os
8
9 # like executing $mkdir folder1
10 os.mkdir("folder1")
11
12 # like executing $cd folder1
13 os.chdir("folder1")
14
15 # like executing $mkdir folder2
16 os.mkdir("folder2")
```



```
etagrats@Teldrasil: ~/python filesDirs
File Edit View Search Terminal Help
etagrats@Teldrasil:~/python filesDirs$ ls -R
.:
mkDirs.py rmDirs.py
etagrats@Teldrasil:~/python filesDirs$ python mkDirs.py
etagrats@Teldrasil:~/python filesDirs$ ls -R
.:
folder1 mkDirs.py rmDirs.py

./folder1:
folder2

./folder1/folder2:
etagrats@Teldrasil:~/python filesDirs$
```

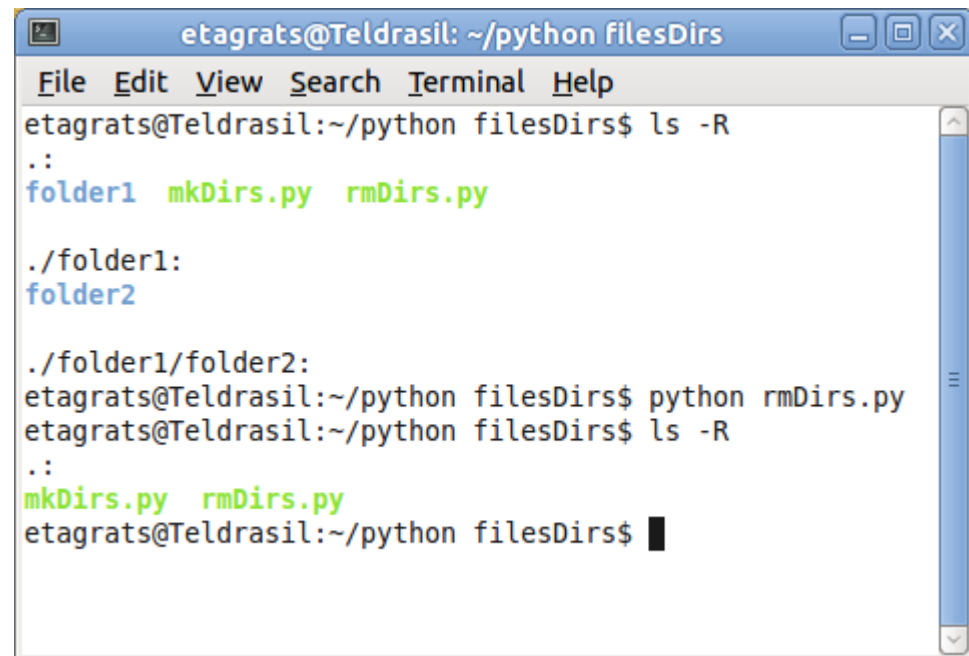


Python: Δυνατότητες

Directories

Removing directories:

```
1 #####
2 ### rmDirs.py ###
3 #####
4
5 # include os module in order
6 # to access system functions
7 import os
8
9 # removes all directories listed
10 # on the path. first removes
11 # folder2 then folder1
12 os.removedirs("folder1/folder2")
13
```



```
etagrats@Teldrasil: ~/python filesDirs
File Edit View Search Terminal Help
etagrats@Teldrasil:~/python filesDirs$ ls -R
.:
folder1 mkDirs.py rmDirs.py

./folder1:
folder2

./folder1/folder2:
etagrats@Teldrasil:~/python filesDirs$ python rmDirs.py
etagrats@Teldrasil:~/python filesDirs$ ls -R
.:
mkDirs.py rmDirs.py
etagrats@Teldrasil:~/python filesDirs$
```



Python: Outline

- Ιστορική Αναδρομή
- Πλεονεκτήματα / Μειονεκτήματα
- Εγκατάσταση
- **Δυνατότητες:**
 - Γενικά
 - Classes + Modules
 - File I/O
 - Error handling
 - Modules for URLs
 - Directories
 - **Regular expressions**
- BIBEL Bibliography Tool with Python
- Συμπεράσματα
- Βιβλιογραφία



Python: Δυνατότητες

Regular Expressions

Εύρεση ταιριάσματος σε μια γραμμή:

```
1 #####
2 ### regExprSimp.py ###
3 #####
4
5 # import regular expression module
6 import re
7
8 # a string with no meaning
9 data_str = "<h1>hello<h2>world<p1>22-123123<p5>apple"
10
11 # compile regular expression
12 p = re.compile('>[a-z]+<')
13
14 # find all patterns that match the regular
15 # and store them to a list expression
16 results_list = p.findall(data_str)
17
18 # appear results
19 i=0
20 while i < len(results_list):
21     print "match(", i+1, ") : ", results_list[i]
22     i+=1
23
```

```
etagrats@Teldrasil: ~/python re
File Edit View Search Terminal Help
etagrats@Teldrasil:~/python re$ python regExprSimp.py
match( 1 ) : >hello<
match( 2 ) : >world<
etagrats@Teldrasil:~/python re$
```



Python: Δυνατότητες

Regular Expressions

Εύρεση ταιριασμάτων
σε πολλαπλές γραμμές:

```
etagrats@Teldrasil: ~/python re
File Edit View Search Terminal Help
etagrats@Teldrasil:~/python re$ cat datafile.txt
<h3><a href="index-a.html">A</a> -
<a href="index-b.html">B</a> -
<a href="index-c.html">C</a> -
<a href="index-d.html">D</a> -
<a href="index-e.html">E</a> -

etagrats@Teldrasil:~/python re$ python regExprFile.py
match( 1 ) : index-a.html
match( 2 ) : index-b.html
match( 3 ) : index-c.html
match( 4 ) : index-d.html
match( 5 ) : index-e.html
etagrats@Teldrasil:~/python re$
```

```
1 #####
2 ### regExprFile.py ###
3 #####
4
5 # import regular expression module
6 import re
7
8 # compile regular expression
9 p = re.compile('index-[a-z].html')
10
11 # open file containing data to be searched
12 openfile = open("datafile.txt", 'r')
13
14 # read file contents
15 filedata = openfile.read()
16
17 # close file
18 openfile.close()
19
20 # find all patterns that match the regular
21 # and store them to a list expression
22 results_list = p.findall(filedata)
23
24 # appear results
25 i=0
26 while i < len(results_list):
27     print "match(", i+1, ") : ", results_list[i],
28     i+=1
29
```



Python: Outline

- Ιστορική Αναδρομή
- Πλεονεκτήματα / Μειονεκτήματα
- Εγκατάσταση
- Δυνατότητες:
 - Γενικά
 - Classes + Modules
 - File I/O
 - Error handling
 - Modules for URLs
 - Directories
 - Regular expressions
- **BIBEL Bibliography Tool with Python**
- Συμπεράσματα
- Βιβλιογραφία



Python: BIBEL tool



Python: Outline

- Ιστορική Αναδρομή
- Πλεονεκτήματα / Μειονεκτήματα
- Εγκατάσταση
- Δυνατότητες:
 - Γενικά
 - Classes + Modules
 - File I/O
 - Error handling
 - Modules for URLs
 - Directories
 - Regular expressions
- BIBEL Bibliography Tool with Python
- **Συμπεράσματα**
- Βιβλιογραφία



Python: Συμπεράσματα

- Μπορεί να αντικαταστήσει οποιαδήποτε γλώσσα προγραμματισμού
- Ο κώδικας είναι ευανάγνωστος και απλός στην ανάγνωσή του (απαιτείται κάποιος βαθμός εξοικείωσης)
- Δεν υπάρχει ένας και μοναδικός τρόπος για την εκτέλεση μιας λειτουργίας



Python: Outline

- Ιστορική Αναδρομή
- Πλεονεκτήματα / Μειονεκτήματα
- Εγκατάσταση
- Δυνατότητες:
 - Γενικά
 - Classes + Modules
 - File I/O
 - Error handling
 - Modules for URLs
 - Directories
 - Regular expressions
- BIBEL Bibliography Tool with Python
- Συμπεράσματα
- **Βιβλιογραφία**



Python: Βιβλιογραφία

- [http://en.wikipedia.org/wiki/Python_\(programming_language\)](http://en.wikipedia.org/wiki/Python_(programming_language))
- <http://wiki.python.org/moin/>
- <http://www.tutorialspoint.com/python/>
- <http://www.sthurlow.com/python/>
- <http://docs.python.org>