

Προγραμματισμός σε C++ & Python & Εφαρμογές στη Ναυπηγική & Ναυτική Μηχανολογία

ΣΝΜΜ 2019

Μάθημα 1B

Γεώργιος Παπαλάμπρου

Επίκουρος Καθηγητής ΕΜΠ

george.papalambrou@lme.ntua.gr

Εργαστήριο Ναυτικής Μηχανολογίας (Κτίριο Λ)

Σχολή Ναυπηγών Μηχανολόγων Μηχανικών

Εθνικό Μετσόβιο Πολυτεχνείο

February 28, 2019

Περιεχόμενα

Start Python (command line)

Core Python Programming, Wesley Chun, Ch. 2

Τα βασικά στοιχεία Python

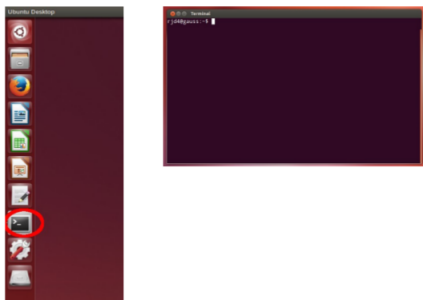
Περιεχόμενο Μαθήματος

- ▶ Εβδομάδα 1. A. Εισαγωγή. Η γλώσσα. Το περιβάλλον Linux. Command line. Python interpreter. Ιστοσελίδα μαθήματος. Βιβλιογραφία. Editors: Sublime, Spyder
B. Εισαγωγή στην γλώσσα Python. Hello World.
- ▶ Εβδομάδα 2. A. Data types. Loops. Control. File I/O
B. Παραδείγματα.
- ▶ Εβδομάδα 3. A. Functions. Modules
B. OOP. Classes
- ▶ Εβδομάδα 4. A. Βιβλιοθήκες Numpy, SciPy.
Errors-Exceptions
B. Παραδείγματα: Γραμμική άλγεβρα, Γραφικά
- ▶ Εβδομάδα 5. Εφαρμογή: Neural Networks. Machine Learning
- ▶ Εβδομάδα 6. Εφαρμογή: Hardware. Πλατφόρμες.
Πρωτόκολλα. Βασικό I/O

Start Python (command line)

- ▶ Στην αρχή ξεκινάμε την Python στο command line του Ubuntu Linux. 6
- ▶ Στα συστήματα *nix (δηλ. UNIX/Linux) ανοίγουμε το τερματικό που βρίσκεται στην αριστερή μπάρα.

Running Python — 1



Start Python (command line)

- ▶ Δίνουμε `python3` και ξεκινάει το περιβάλλον της γλώσσας με τα χαρακτηριστικά `>>>`.

Running Python — 2

```
$ python3
Python 3.2.3 (default, May 3 2012, 15:54:42)
[GCC 4.6.3] on linux2

>>>
```

The diagram illustrates the execution of the `python3` command. It shows the following sequence of events:

- The **Unix prompt** (\$) is followed by the **Unix command** `python3`.
- The command execution results in an **Introductory blurb**:
Python 3.2.3 (default, May 3 2012, 15:54:42)
[GCC 4.6.3] on linux2
- The **Python version** (3.2.3) is indicated by an arrow pointing to the version number in the blurb.
- The **Python prompt** (`>>>`) is shown at the end of the output.

10

Quit Python (command line)

- ▶ Δίνουμε εναλλακτικά `exit()`, `quit()`, CTRL+D και σταματάει η λειτουργία της γλώσσας, επιστρέφοντας στο terminal του Linux.

Quitting Python

```
>>> exit()
>>> quit()
>>> Ctrl + D
```

Any one of these

11

Μία πρώτη εντολή

- ▶ Παραδοσιακά στον προγραμματισμό το πρώτο πρόγραμμα είναι το διάσημο "Hello, world".
- ▶ Δίνουμε έτσι `print('Hello, world')` και enter.
- ▶ Στην επόμενη γραμμή θα εμφανιστεί το Hello, world.

A first Python command

```
>>> print('Hello, world!')  
Hello, world!  
>>>
```

The diagram illustrates the interaction between a user and the Python interpreter. It shows a sequence of three lines of text. The first line is the Python prompt `>>>`, followed by the command `print('Hello, world!')`. The second line is the output `Hello, world!`. The third line is the Python prompt `>>>` again. Red arrows point from labels to the corresponding parts of the code: 'Python prompt' points to the first `>>>`, 'Python command' points to the code line, 'Output' points to the output line, and another 'Python prompt' points to the final `>>>`.

12

Ένα πρώτο πρόγραμμα script

- ▶ Συνήθως τα προγράμματα διατηρούνται σε αρχεία (scripts) τα οποία καλούνται στο command line.
- ▶ Στην Python τα αρχεία αυτά είναι τύπου *.py.
- ▶ Το αρχείο εδώ μπορεί να ονομαστεί hello1.py, οπότε στο command line θα δώσουμε python3 hello1.py.
- ▶ Προσοχή στην έκδοση της γλώσσας: δίνοντας python3 καλκαλείται η έκδοση 3 (τρέχουσα). Δίνοντας python καλείται η έκδοση 2, που εγκαταλείπεται πια.

Python scripts

File in home directory

```
print('Hello, world!')
```

hello1.py

Run from Unix prompt

```
$ python3 hello1.py
```

Hello, world!

```
$
```

Unix prompt

Unix command to run Python

Python script

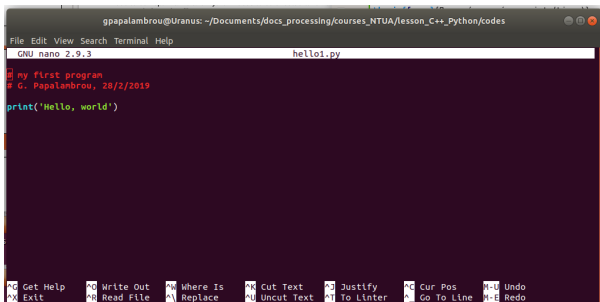
Python script's output

Unix prompt

16

Πως γράφεται ένα script (Linux)

- ▶ Στο terminal (\$) δίνουμε `pico hello1.py`. Καλούμε έτσι τον editor `pico`, και δημιουργείται το νέο πρόγραμμα. Εναλλακτικά καλούμε τον `nano`.
- ▶ Για τέλος δίνουμε `CTRL + X` για έξοδο, αφού αποθηκεύσουμε το αρχείο.
- ▶ Για σχόλια 1 γραμμής θέτουμε το `#`. Για σχόλια σε κείμενο με περισσότερες γραμμές, θέτουμε στην αρχή και στο τέλος 3 μονά εισαγωγικά `'`.

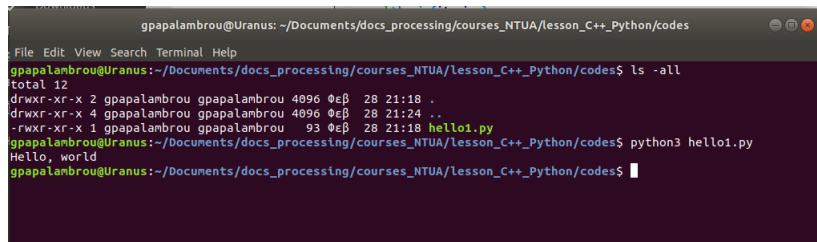


```
gppalambrou@Uranus: ~/Documents/docs_processing/courses_NTUA/lesson_C++_Python/codes
File Edit View Search Terminal Help
GNU nano 2.9.3 hello1.py
# my first program
# G. Papalambrou, 28/2/2019
print('Hello, world')
```

Get Help Write Out Where Is Cut Text Justify Cur Pos Undo
Exit Read File Replace Uncut Text To Linter Go To Line Redo

Πως καλείται ένα script (Linux)

- ▶ Πρώτα ελέγχουμε αν το script είναι εκτελέσιμο. Στο terminal (\$) δίνουμε `ls -all`. Στην πρώτη στήλη πρέπει να είναι το αρχείο σημειωμένο με x (δηλ. executable). Αν δεν είναι (πχ `-r-`), δίνουμε `chmod +x *.py`. Ελέγχουμε πάλι με `ls -all`, ώστε πχ. `-xr`.
- ▶ Για να καλέσουμε το script στην γλώσσα Python 3 δίνουμε `python3 hello1.py`. Στην ουσία καλούμε τον interpreter να τρέξει το πρόγραμμα.



```
gpalambrou@Uranus: ~/Documents/docs_processing/courses_NTUA/lesson_C++_Python/codes
File Edit View Search Terminal Help
gpalambrou@Uranus:~/Documents/docs_processing/courses_NTUA/lesson_C++_Python/codes$ ls -all
total 12
drwxr-xr-x 2 gpalambrou gpalambrou 4096 Φεβ  28 21:18 .
drwxr-xr-x 4 gpalambrou gpalambrou 4096 Φεβ  28 21:24 ..
-rwxr-xr-x 1 gpalambrou gpalambrou   93 Φεβ  28 21:18 hello1.py
gpalambrou@Uranus:~/Documents/docs_processing/courses_NTUA/lesson_C++_Python/codes$ python3 hello1.py
Hello, world
gpalambrou@Uranus:~/Documents/docs_processing/courses_NTUA/lesson_C++_Python/codes$
```

Αρχική (βασική) Βιβλιογραφία

- ▶ Το υλικό των αρχικών παραδόσεων προέρχεται από τα εξής: **Core Python Programming (2nd Edition), Wesley Chun**
- ▶ Ξεκινάμε με το κεφ. 2: Getting started

Δομή προγράμματος σε Python: modules, sec. 3.4

```
#!/usr/bin/env python
```

(1) Startup line (Unix)

```
"this is a test module"
```

(2) Module documentation

```
import sys  
import os
```

(3) Module imports

```
debug = True
```

(4) (Global) Variable declarations

```
class FooClass (object):  
    "Foo class"  
    pass
```

(5) Class declarations (if any)

```
def test():  
    "test function"  
    foo = FooClass()  
    if debug:  
        print 'ran test()'
```

(6) Function declarations (if any)

```
if __name__ == '__main__':  
    test()
```

(7) "main" body

1. Startup line

Generally used only in Unix environments, the startup line allows for script execution by name only (invoking the interpreter is not required).

2. Module documentation

Summary of a module's functionality and significant global variables; accessible externally as `module.__doc__`.

3. Module imports

Import all the modules necessary for all the code in current module; modules are imported once (when this module is loaded); imports within functions are not invoked until those functions are called.

4. Variable declarations

Declare here (global) variables that are used by multiple functions in this module. We favor the use of local variables over globals, for good programming style mostly, and to a lesser extent, for improved performance and less memory usage.

5. Class declarations

Any classes should be declared here. A class is defined when this module is imported and the `class` statement executed. Documentation variable is `class.__doc__`.

6. Function declarations

Functions that are declared here are accessible externally as `module.function()`; function is defined when this module is imported and the `def` statement executed. Documentation variable is `function.__doc__`.

7. "main" body

All code at this level is executed, whether this module is imported or started as a script; generally does not include much functional code, but rather gives direction depending on mode of execution.

Ένα πλήρες πρόγραμμα Python

Example 3.1. File Create (`makeTextFile.py`)

This application prompts the user for a (nonexistent) filename, then has the user enter each line of that file (one at a time). Finally, it writes the entire text file to disk.

```
1 #!/usr/bin/env python
2
3 'makeTextFile.py -- create text file'
4
5 import os
6 ls = os.linesep
7
8 # get filename
9 while True:
10
11     if os.path.exists(fname):
12         print "ERROR: '%s' already exists" % fname
13     else:
14         break
15
16 # get file content (text) lines
17 all = []
18 print "\nEnter lines ('.' by itself to quit).\n"
19
20 # loop until user terminates input
21 while True:
22     entry = raw_input('> ')
23     if entry == '.':
24         break
25     else:
26         all.append(entry)
27
28 # write lines to file with proper line-ending
29 fobj = open(fname, 'w')
30 fobj.writelines(['%s%s' % (x, ls) for x in all])
31 fobj.close()
32 print 'DONE!'
```

Τα βασικά στοιχεία Python

▼ Chapter 2. Getting Started	57
Section 2.1. Program Output, the print Statement,...	59
Section 2.2. Program Input and the raw_input() Bui...	61
Section 2.3. Comments	63
Section 2.4. Operators	64
Section 2.5. Variables and Assignment	66
Section 2.6. Numbers	67
Section 2.7. Strings	69
Section 2.8. Lists and Tuples	70
Section 2.9. Dictionaries	71
Section 2.10. Code Blocks Use Indentation	72
Section 2.11. if Statement	73
Section 2.12. while Loop	74
Section 2.13. for Loop and the range() Built-in Fun...	75
Section 2.14. List Comprehensions	78
Section 2.15. Files and the open() and file() Built-in...	79
Section 2.16. Errors and Exceptions	81
Section 2.17. Functions	82
Section 2.18. Classes	84
Section 2.19. Modules	87
Section 2.20. Useful Functions	89