



Introduction to Python Programming

Hellenic Mediterranean University

Lecture 1

Dr. Alina Eqtami

Γιατί Python;

- Clean syntax, fast start for beginners
- Widely used in robotics, data, and AI
- We will focus on **how programs work**, not on memorizing commands
- Στόχος: να χτίσουμε “mental model” για το τι κάνει ένα πρόγραμμα

Today's Promise – Υπόσχεση 1ου μαθήματος

Μετά το μάθημα

Θα μπορείτε να προβλέπετε την έξοδο ενός μικρού κώδικα πριν πατήσετε Run.

Τι σημαίνει Run;

Run = ο υπολογιστής εκτελεί ακριβώς τις οδηγίες που γράψαμε.

Practical information

- Python runs on any computer
- We will use ready lab computers
- Everything we use is **free**

Wrong mindset

Python = a list of commands to memorize

Right mindset

Python = a way to describe a **process** that changes **state (memory)** over time

Agenda

- 1 What is a program? (τι είναι πρόγραμμα;)
- 2 Memory as boxes (μνήμη = κουτάκια)
- 3 Execution trace (εκτέλεση γραμμή-γραμμή)
- 4 Prediction exercises (τι θα τυπώσει;)
- 5 Why loops exist (γιατί χρειαζόμαστε loops)
- 6 Lab workflow (αμφιθέατρο vs εργαστήριο)

Key Idea #1

A computer is literal

- It does not understand what we *mean*
- It executes what we *write*
- It is consistent and predictable

Με απλά λόγια

Ο υπολογιστής δεν έχει “κοινή λογική”. Εκτελεί οδηγίες.

A computer does NOT do what we want.
It does exactly what we write.

Ο υπολογιστής δεν κάνει αυτό που θέλω.
Κάνει αυτό που του είπα ακριβώς.

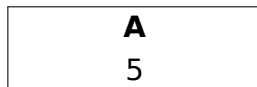
What is a Program? – Τι είναι πρόγραμμα;

Ορισμός

Πρόγραμμα = ακολουθία εντολών που αλλάζει την κατάσταση της μνήμης.

Memory Model – Μοντέλο μνήμης

- Think of memory as **boxes**
- Each box has a **name** and a **value**
- Instructions update values over time



Σκεφτείτε “κουτάκια”: όνομα + τιμή.

Without Python

- put 5 in box A
- put 3 in box B
- put $A + B$ in box C

Without Python

- put 5 in box A
- put 3 in box B
- put $A + B$ in box C

Ερώτηση

Τι έχει μέσα το C;

Without Python

- put 5 in A
- put $A + 1$ in A
- put $A + 1$ in A
- put $A + 1$ in A

Without Python

- put 5 in A
- put $A + 1$ in A
- put $A + 1$ in A
- put $A + 1$ in A

Ερώτηση

Τι έχει το A στο τέλος;

Πάμε γραμμή-γραμμή και ενημερώνουμε τη μνήμη.

Step	A
start	5
1	6
2	7
3	8

Now in Python – Τώρα σε Python

```
A = 5  
A = A + 1  
A = A + 1  
A = A + 1
```

Το σημαντικό δεν είναι η σύνταξη. Είναι το “state update”.

Key Idea #2 – Ιδέα #2

Important

The symbol = does NOT mean equality.

Meaning

It means: store a new value in memory.

Δηλαδή: “βάλει νέα τιμή στο κουτί”.

print() – Το print() είναι παράθυρο

- print() shows a value **at a specific moment**
- It helps us debug and understand state

Το print δείχνει “τι έχει τώρα το κουτί”.

Example 1 - Παράδειγμα

```
x = 5
print(x)
x = x + 2
print(x)
```

Ερώτηση: τι θα τυπώσει; Κάντε trace.

Key message

Code is not text. Code is a process that evolves step by step.

Ο υπολογιστής δεν “βλέπει” το τέλος. Εκτελεί γραμμή-γραμμή.

In this course

We train the ability to predict program behavior.

Στόχος

Να μπορείτε να κοιτάτε κώδικα και να λέτε τι θα κάνει πριν τρέξει.

Before running a program:

- 1 Read
- 2 Trace (update memory)
- 3 Predict output
- 4 Run to verify

Prediction 1 - Πρόβλεψη 1

```
x = 2  
y = 3  
x = y  
y = 5  
print(x)
```

Τι θα τυπώσει; Γιατί;

Trace for Prediction 1 - Trace

Line	x	y
start	-	-
1	2	-
2	2	3
3	3	3
4	3	5

Prediction 2 – Πρόβλεψη 2

```
a = 10  
b = a  
a = 0  
print(b)
```

Μην μαντέψετε: trace.

Prediction 3 - Πρόβλεψη 3

```
x = 1  
x = x + 1  
x = x + 1  
print(x)
```

Τι θα τυπώσει; (state update)

Expressions vs Statements – Εκφράσεις vs Εντολές

Expression

Computes a value (e.g., $x + 1$)

Statement

Performs an action (e.g., $x = x + 1$)

Σήμερα μας νοιάζει κυρίως το “statement” ως αλλαγή μνήμης.

- `int` : integers
- `float` : real numbers
- `str` : text

Οι τύποι θα έρθουν σταδιακά. Πρώτα χτίζουμε mental model.

- `int` : counting - πόσα
- `float` : measurement - πόσο
- `str` : information/text - τι

Important

The variable does not have a type. The value stored in it has a type.

Errors are Information – Τα λάθη είναι πληροφορία

- Errors are normal
- Messages help us understand what the computer did
- Debugging is a core skill

Δεν είναι “αποτυχία”. Είναι feedback.

A Problem – Ένα πρόβλημα

```
x = 0
x = x + 5
x = x + 5
x = x + 5
x = x + 5
x = x + 5
```

Πολύ επανάληψη. Πώς το κάνουμε καλύτερα;

Motivation

We want repetition without repeating code.

Πρώτα η ανάγκη, μετά η εντολή.

Loop Idea – Ιδέα βρόχου (χωρίς σύνταξη)

- A loop repeats a state change
- It stops when a condition becomes false (or true)

Loop = εξέλιξη κατάστασης στο χρόνο.

A loop stops only if something changes.

Ένα loop τελειώνει μόνο αν κάτι μέσα του
αλλάζει.

Key point

If nothing changes, nothing ends.

Ο υπολογιστής δεν “καταλαβαίνει” ότι πρέπει να σταματήσει.

Lecture vs Lab – Αμφιθέατρο vs Εργαστήριο

Lecture

Lab

Prediction

Verification

Στο εργαστήριο “ελέγχουμε” τις προβλέψεις μας.

In the Lab (today) – Στο εργαστήριο (σήμερα)

- Run the examples from the lecture
- Fill trace tables
- Discuss mismatches between prediction and output

Πριν πατήσετε Run: γράφετε τι περιμένετε να δείτε.

- We will not ask: “Do you understand?”
- We will ask: **“What will it print?”**
- Disagreement is good: it means thinking is happening

Takeaways – Τι κρατάμε

- 1 Programs change memory (state)
- 2 Execution is step-by-step (time)
- 3 Predict before running
- 4 Loops are about state change

Task

Write prediction + trace for 5 short snippets (provided in the lab).

Στόχος: “τρέξιμο στο μυαλό” πριν το Run.

- Input and basic operators
- Conditions (if) as control of flow
- First small programs

Why we will start with `while`

It exposes the mechanism of state change.

To `for` κρύβει μηχανισμό. Το `while` τον αποκαλύπτει.

Thank you! – Ευχαριστώ!

Questions?
Απορίες;