

Τελευταία ενημέρωση 30/9/2025

# Python – Μέρος Α

## ΕΚΔΔΑ

1.2 Εισαγωγή στην εκπαιδευτική διαδικασία και στις απαιτήσεις του προγράμματος

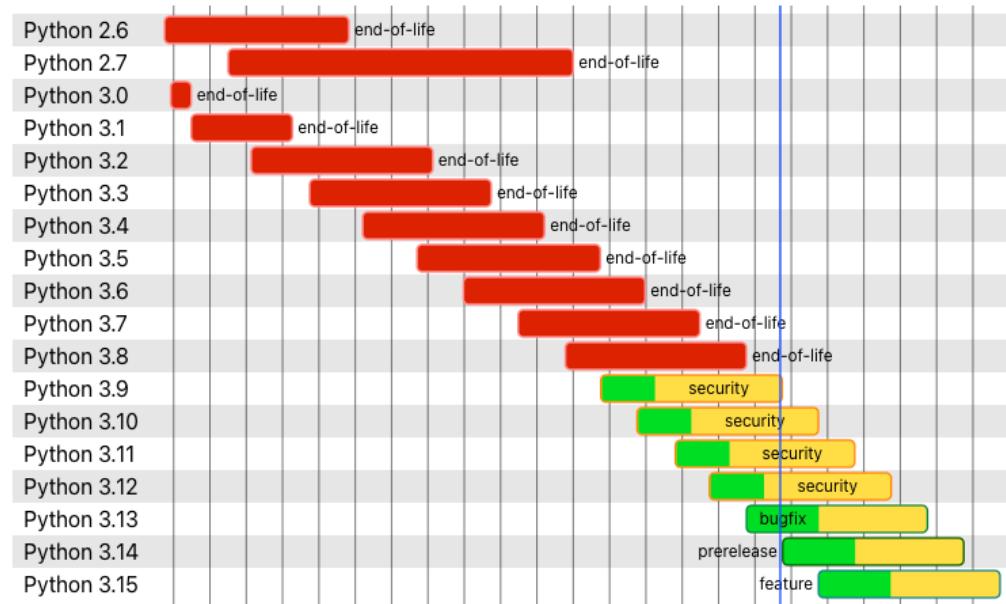
Εβδομάδα 1/7

Σεπτέμβριος 2025

Επιμέλεια παρουσίασης: Χρήστος Γκόγκος

# Τι είναι η Python;

- Η Python είναι μια γλώσσα προγραμματισμού, δηλαδή μπορεί να χρησιμοποιηθεί για τη συγγραφή προγραμμάτων με τρόπο που είναι «κατανοητός» από ανθρώπους και που μπορούν να εκτελεστούν στη συνέχεια από ένα υπολογιστικό σύστημα
- Η Python δημιουργήθηκε το 1990 από τον [Guido Van Rossum](#) ως μια γλώσσα προγραμματισμού υψηλού επιπέδου, γενικού σκοπού δίνοντας έμφαση στη γρήγορη ανάπτυξη κώδικα και στην υψηλή αναγνωσιμότητα



<https://devguide.python.org/versions/>

# Πόσο δημοφιλής είναι η γλώσσα Python;

Αύγουστος 2025

Aug 2025	Aug 2024	Change	Programming Language	Ratings	Change
1	1		 Python	26.14%	+8.10%
2	2		 C++	9.18%	-0.86%
3	3		 C	9.03%	-0.15%
4	4		 Java	8.59%	-0.58%
5	5		 C#	5.52%	-0.87%
6	6		 JavaScript	3.15%	-0.76%
7	8	▲	 Visual Basic	2.33%	+0.15%
8	9	▲	 Go	2.11%	+0.08%
9	25	▲	 Perl	2.08%	+1.17%
10	12	▲	 Delphi/Object Pascal	1.82%	+0.19%

<https://www.tiobe.com/tiobe-index/>

# Γιατί η Python είναι τόσο δημοφιλής;

- Είναι δωρεάν και εύκολα διαθέσιμη σε όλους (π.χ., Windows, MacOS, Linux, κ.λπ.)
- Διαθέτει διερμηνευτή εντολών και σημειωματάρια (jupyter notebooks) για άμεση εκτέλεση τμημάτων κώδικα
- Αρκετή λειτουργικότητα περιλαμβάνεται στην ίδια την γλώσσα - batteries included!
- Είναι ανοικτού κώδικα (open source)
- Διαθέτει δωρεάν μεγάλο πλήθος βιβλιοθηκών ( π.χ., numpy, pandas, scikit-learn, pytorch κ.α.)
- Το προγράμμα σε Python είναι σύντομα και εύκολα αναγνώσιμα, «εκτελέσιμος ψευδοκώδικας»

# Ο κώδικας Python ως ψευδοκώδικας

## Ψευδοκώδικας

```
PROMPT for ph_value
GET the ph_value and make it a number
IF ph_value is greater than or equal to 7 THEN
    DISPLAY "Neutral or alkaline"
ELSE
    DISPLAY "Acidic"
ENDIF
```

## Python

```
ph_value = float(input("Enter pH value:"))
if ph_value >= 7:
    print("Neutral or alkaline")
else:
    print("Acidic")
```

<https://interactivetextbooks.tudelft.nl/programming-foundations/content/chapter6/pseudocode.html>

# To Zen της Python (1/2)

- Το Zen της Python είναι ένα σύνολο 19 βασικών αρχών για τη συγγραφή κομψού, αναγνώσιμου και λειτουργικού κώδικα Python
- Δημιουργήθηκε από τον [Tim Peters](#) και έχει αποτελέσει το λεγόμενο PEP 20
- PEP (Python Enhancement Proposal) είναι ένα έγγραφο που είτε παρέχει κάποια χρήσιμη πληροφορία για την κοινότητα της Python είτε περιγράφει ένα νέο χαρακτηριστικό της Python και του περιβάλλοντός της
- Δείτε σχετικά και το PEP 8 - <https://peps.python.org/pep-0008/>

# To Zen της Python (2/2)

```
Python 3.13.7 | packaged by conda-forge | (main, Sep 3 2025, 14:24:41)
darwin
Type "help", "copyright", "credits" or "license" for more information
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```

- 1.Το όμορφο είναι καλύτερο από το άσχημο.
- 2.Το ρητό είναι καλύτερο από το υπονοούμενο.
- 3.Το απλό είναι καλύτερο από το σύνθετο.
- 4.Το σύνθετο είναι καλύτερο από το περίπλοκο.
- 5.Το επίπεδο είναι καλύτερο από το ένθετο.
- 6.Το αραιό είναι καλύτερο από το πυκνό.
- 7.Η αναγνωσιμότητα μετράει.
- 8.Οι ειδικές περιπτώσεις δεν είναι αρκετά ειδικές για να παραβιάσουν τους κανόνες.
- 9.Ωστόσο η πρακτικότητα νικά την απόλυτη καθαρότητα.
- 10.Τα λάθη δεν πρέπει ποτέ να περνούν σιωπηλά.
- 11.Εκτός αν σιωπηθούν ρητά.
- 12.Μπροστά στην αμφιβολία, αρνηθείτε τον πειρασμό να μαντέψετε.
- 13.Θα πρέπει να υπάρχει ένας – και κατά προτίμηση μόνο ένας – προφανής τρόπος για να το κάνετε.
- 14.Αν και αυτός ο τρόπος μπορεί να μην είναι προφανής στην αρχή, εκτός αν είστε Ολλανδός.
- 15.Το τώρα είναι καλύτερο από το ποτέ.
- 16.Αν και το ποτέ είναι συχνά καλύτερο από το **αμέσως** τώρα.
- 17.Αν η υλοποίηση είναι δύσκολο να εξηγηθεί, είναι κακή ιδέα.
- 18.Αν η υλοποίηση είναι εύκολο να εξηγηθεί, μπορεί να είναι καλή ιδέα.
- 19.Οι χώροι ονομάτων είναι μια ηχηρά καλή ιδέα – ας κάνουμε και άλλες τέτοιες!

# Pythonic (πυθωνικός) κώδικας

- Ο πυθωνικός κώδικας χρησιμοποιεί ιδιώματα της γλώσσας Python και είναι εύκολα αναγνώσιμος
- Ιδίωμα μιας γλώσσας είναι ένας κοινά αποδεκτός και ενδεδειγμένος τρόπος επίλυσης ενός προβλήματος που χρησιμοποιούν οι έμπειροι προγραμματιστές

Παράδειγμα 2 τρόπων αντιμετάθεσης 2 μεταβλητών

```
a, b = 7, 42
print(a, b) # Έξοδος: 7 42
```

```
# Μη πυθωνικός κώδικας αντιμετάθεσης 2 μεταβλητών
temp = a
a = b
b = temp
print(a, b) # Έξοδος: 42 7
```

```
# Πυθωνικός κώδικας αντιμετάθεσης 2 μεταβλητών
a, b = b, a
print(a, b) # Έξοδος: 7 42
```

# Εγκατάσταση της Python

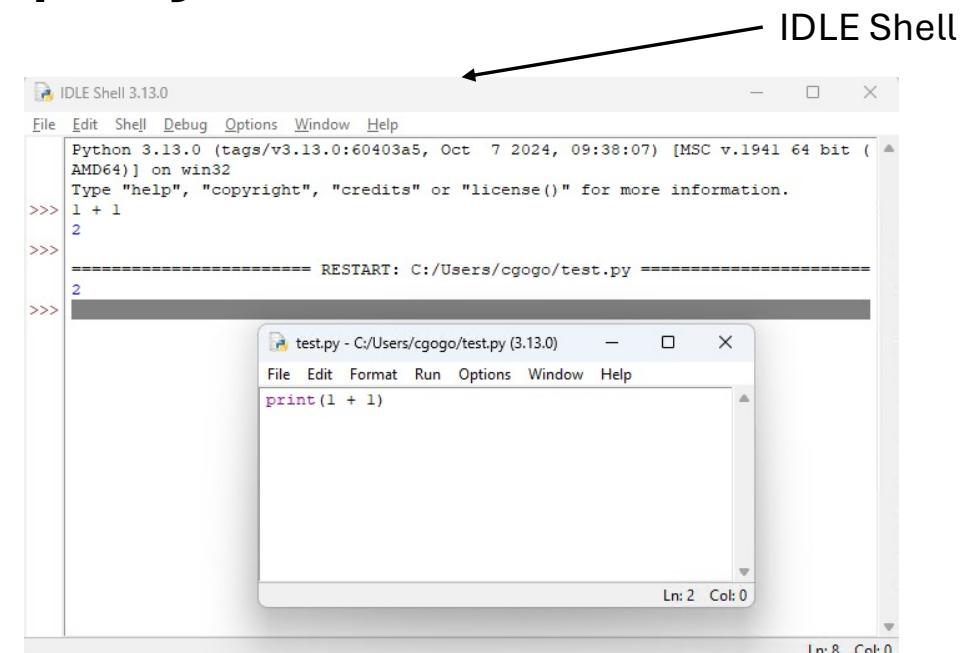
- Επίσκεψη στο <https://www.python.org/>, μεταφόρτωση και εγκατάσταση μιας πρόσφατης έκδοσης της Python 3.major\_version.minor\_version που είναι διαθέσιμη, π.χ. Python 3.13.7 (για Σεπτέμβριο 2025)
- Στα Windows η εγκατάσταση γίνεται εύκολα κατεβάζοντας και εκτελώντας έναν “Windows installer” ανάλογα με τον τύπο υπολογιστή που διαθέτετε (π.χ., 64-bit για επεξεργαστές Intel ή ARM64 για επεξεργαστές ARM)
- Υπάρχουν και άλλοι περισσότερο προχωρημένοι τρόποι εγκατάστασης της Python (π.χ., un, conda, poetry)

Δείτε για την εγκατάσταση της Python στα Windows  
[https://www.youtube.com/watch?v=Pi\\_SMAktdXk](https://www.youtube.com/watch?v=Pi_SMAktdXk)  
από το 11:33 μέχρι το 16:00

# Μετά την εγκατάσταση της Python

- Μετά την εγκατάσταση της Python μπορούν να χρησιμοποιηθούν:
  - το **Python IDLE shell** που επιτρέπει τη αλληλοεπιδραστική εκτέλεση εντολών Python, την εμφάνιση αποτελεσμάτων, αλλά και τη συγγραφή εκτέλεσης και αποσφαλμάτωση προγραμμάτων σε Python (στα Windows δημιουργείται με την εγκατάσταση της Python εικονίδιο σχετικής εφαρμογής)
  - Το **REPL (Read Evaluate Print Loop)** της Python που μπορεί να κληθεί από τη γραμμή εντολών πληκτρολογώντας την εντολή `python` (ή `python3` ή `py` ανάλογα με την εγκατάσταση)
  - Ωστόσο, υπάρχουν και άλλοι τρόποι συγγραφής και εκτέλεσης κώδικα σε Python που μπορεί να είναι βολικότεροι για συγγραφή και εκτέλεση αποσπασμάτων κώδικα ή μεγαλύτερων εφαρμογών

REPL



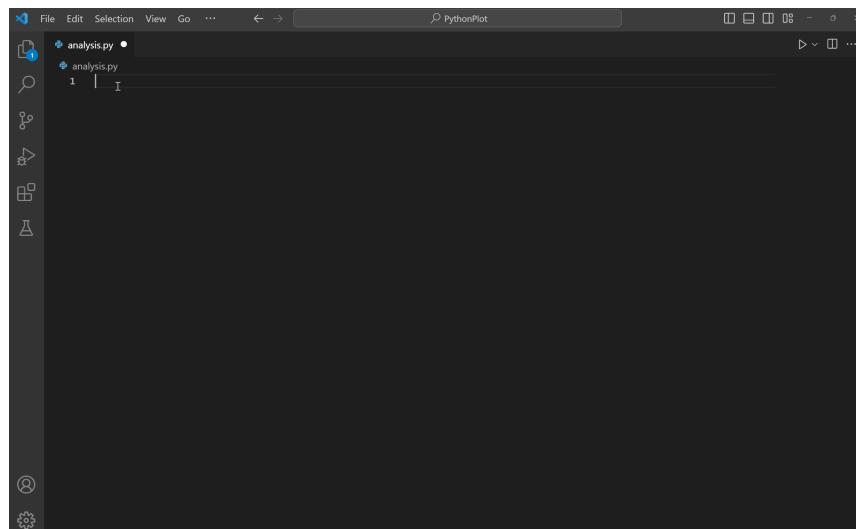
The image shows a terminal window with the following content:

```
C:\Users\cgogo>python
Python 3.13.0 (tags/v3.13.0:60403a5, Oct  7 2024, 09:38:07)
[MSC v.1941 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 1 + 1
2
>>>
```

# Περιβάλλοντα ανάπτυξης λογισμικού με την Python

- **Google Colab**
- **Visual Studio Code IDE**
- PyCharm IDE
- Anaconda + Spyder IDE
- PyDev IDE
- Python IDLE
- Jupyter Notebooks

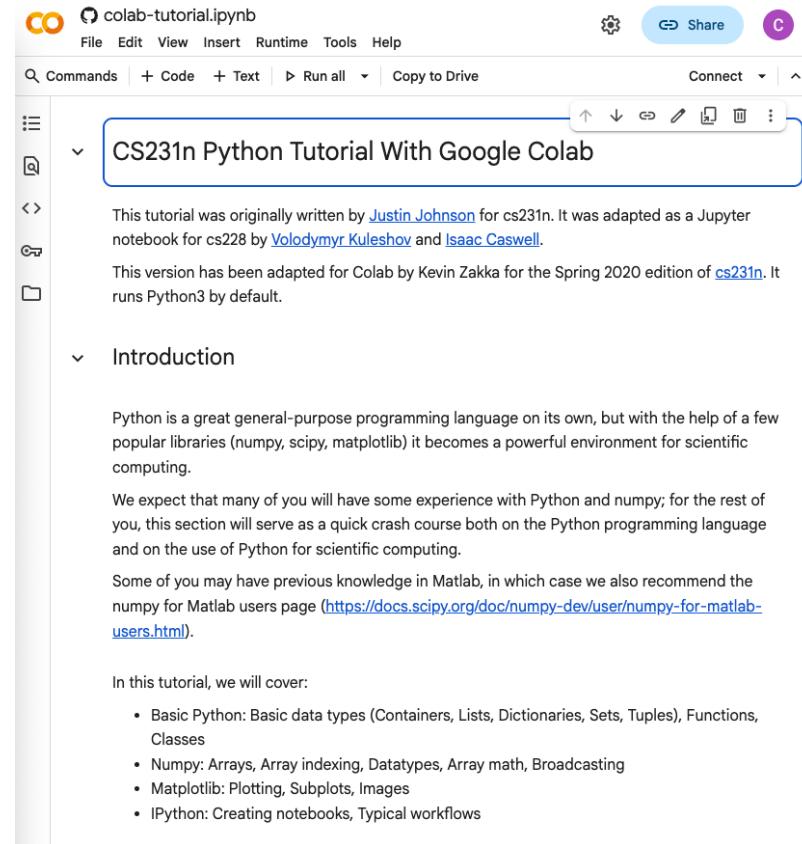
Ένα IDE (Integrated Development Environment) τυπικά αυξάνει την παραγωγικότητα του προγραμματιστή καθώς μέσα από ένα κοινό περιβάλλον παρέχονται πολλές ευκολίες όπως η οργάνωση κώδικα σε projects, η εύκολη συγγραφή και εκτέλεση κώδικα, ο εντοπισμός και διόρθωση σφαλμάτων, η σύνδεση με λογισμικά όπως το git για διατήρηση εκδόσεων του κώδικα, η φόρτωση και η ενημέρωση βιβλιοθηκών κ.α.



<https://code.visualstudio.com/docs/python/editing>

# Google Colab

- Το Google Colab (Colaboratory) είναι ένα δωρεάν, cloud-based Jupyter notebook περιβάλλον που επιτρέπει τη συγγραφή κώδικα Python απευθείας στον φυλλομετρητή χωρίς να χρειάζεται οποιαδήποτε εγκατάσταση
- Ο χρήστης θα πρέπει να έχει λογαριασμό στην Google για να μπορεί να χρησιμοποιήσει το Colab
- Αποκτήστε πρόσβαση στο Colab μέσω του <https://colab.research.google.com/>



The screenshot shows the Google Colab interface with the following details:

- Header:** colab-tutorial.ipynb, Share, C
- Toolbar:** File, Edit, View, Insert, Runtime, Tools, Help; Commands, Code, Text, Run all, Copy to Drive, Connect
- Section:** CS231n Python Tutorial With Google Colab
- Text:** This tutorial was originally written by [Justin Johnson](#) for cs231n. It was adapted as a Jupyter notebook for cs228 by [Volodymyr Kuleshov](#) and [Isaac Caswell](#). This version has been adapted for Colab by Kevin Zakka for the Spring 2020 edition of [cs231n](#). It runs Python3 by default.
- Section:** Introduction
- Text:** Python is a great general-purpose programming language on its own, but with the help of a few popular libraries (numpy, scipy, matplotlib) it becomes a powerful environment for scientific computing. We expect that many of you will have some experience with Python and numpy; for the rest of you, this section will serve as a quick crash course both on the Python programming language and on the use of Python for scientific computing. Some of you may have previous knowledge in Matlab, in which case we also recommend the numpy for Matlab users page (<https://docs.scipy.org/doc/numpy-dev/user/numpy-for-matlab-users.html>).
- Text:** In this tutorial, we will cover:
  - Basic Python: Basic data types (Containers, Lists, Dictionaries, Sets, Tuples), Functions, Classes
  - Numpy: Arrays, Array indexing, Datatypes, Array math, Broadcasting
  - Matplotlib: Plotting, Subplots, Images
  - IPython: Creating notebooks, Typical workflows

<https://colab.research.google.com/github/cs231n/cs231n.github.io/blob/master/python-colab.ipynb>

# Εγκατάσταση του Visual Studio Code και των επεκτάσεων του για Python

- Εγκατάσταση του Visual Studio Code από το <https://code.visualstudio.com/download>
- Εγκατάσταση του VS Code Python extension, ακολουθήστε τις οδηγίες στο <https://code.visualstudio.com/docs/python/python-tutorial>

# Εκπαιδευτικό υλικό επιμόρφωσης

- Μια παρουσίαση ανά εβδομάδα μελέτης
- Παραπομπές μελέτης ανά εβδομάδα μελέτης
- Λυμένες ασκήσεις
- Ασκήσεις προς λύση
- Εργασίες
- Σύνδεσμοι προς χρήσιμο υλικό (βιβλία, παρουσιάσεις, μαθήματα, tutorials, βιντεομαθήματα κ.α.) που μπορείτε να βρείτε στο <https://hackmd.io/@cgogos/SkdEDp9qgg>

# Μορφή ασκήσεων επιμόρφωσης

- Οι ασκήσεις είναι με τη μορφή notebook που:
  - μπορεί χωρίς καμία επιπλέον εγκατάσταση λογισμικού να φορτωθεί και να εκτελείται στο Google Colab,
  - είτε να φορτωθεί και να εκτελείται τοπικά στο Jupyter,
  - είτε να φορτωθεί και να εκτελείται τοπικά στο VS Code,
  - είτε να φορτωθεί και να εκτελείται χρησιμοποιώντας κάποιο άλλο λογισμικό (π.χ., PyCharm, Anaconda+Spyder, jupyterlab)
- Ένα notebook με ασκήσεις προς επίλυση ανά εβδομάδα μελέτης
- Κάθε άσκηση έχει την εκφώνηση της και έναν χώρο στον οποίο οι επιμορφούμενοι συμπληρώνουν την απάντηση
- Οι επιμορφούμενοι ανεβάζουν (προαιρετικά) το συμπληρωμένο notebook (αρχείο .ipynb) στην πλατφόρμα του προγράμματος «Python-Α» και λαμβάνουν ανατροφοδότηση

# Εργασίες (υποχρεωτικές)

- Ανακοινώνονται 3 εργασίες στην 3<sup>η</sup>, 5<sup>η</sup> και 7<sup>η</sup> εβδομάδα
- Οι επιμορφούμενοι ανεβάζουν τις λύσεις τους στην πλατφόρμα του προγράμματος και λαμβάνουν ανατροφοδότηση
- Οι λύσεις που υποβάλλονται μπορεί να είναι αρχεία python κώδικα (.py) ή αρχεία notebook (.ipynb)

# Αξιολόγηση γνώσεων και δεξιοτήτων

- Η τελική αξιολόγηση των επιμορφούμενων γίνεται μέσω ενός κουίζ ερωτήσεων κλειστού τύπου, σε όλη την ύλη, στο τέλος του προγράμματος επιμόρφωσης

# Περί χρήσης AI

Στη επιμόρφωση αυτή η χρήση μεγάλων γλωσσικών μοντέλων όπως το ChatGPT, DeepSeek, Claude κ.α.:

- **επιτρέπεται** για ανάκτηση πληροφοριών, επεξήγηση κώδικα και δημιουργία συνόψεων θεμάτων (summarizations)
- **επιτρέπεται** για την αναγνώριση και την επεξήγηση λαθών σε κώδικα που εσείς έχετε γράψει
- **απαγορεύεται** για τη δημιουργία του ζητούμενου κώδικα των ασκήσεων
- **απαγορεύεται** κατά τη διάρκεια της τελικής αξιολόγησης γνώσεων και δεξιοτήτων