

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

1.1 Ιστορικά στοιχεία

Η γλώσσα προγραμματισμού Python δημιουργήθηκε τον Δεκέμβριο του 1989 από τον Guido Van Rossum στο Ερευνητικό Κέντρο CWI (*Centrum Wiskunde & Informatica*) στην Ολλανδία. Όπως περιγράφει ο ίδιος¹ ξεκίνησε ως ένα πρότζεκτ για τις διακοπές των Χριστουγέννων, για την καλύτερη διαχείριση του συστήματος (*System Administration*) Amoeba. Οι βασικές επιρροές της βρίσκονται σε γλώσσες όχι ιδιαίτερα προσφιλείς, στους περισσότερους από εμάς – στη γλώσσα ABC και στη Modula 3.



Οι δυο κυριότερες εκδόσεις της είναι η 2.x και η 3.x. Η έκδοση 2.x παρουσιάστηκε στις 16 Οκτωβρίου 2000 και πολύ γρήγορα ξεκίνησε να χρησιμοποιείται σε πολλούς τομείς και να αναπτύσσεται μια μεγάλη ποικιλία εφαρμογών σε Python. Η έκδοση αυτή, που δεν είναι συμβατή με την τελευταία έκδοση 3.x, χρησιμοποιείται ακόμη για την ανάπτυξη εφαρμογών.

Η έκδοση 3.x παρουσιάστηκε στις 3 Δεκεμβρίου 2008, με το όνομα Python 3000 ή Py3K. Τη στιγμή που γραφόταν το βιβλίο η τελευταία έκδοση της γλώσσας ήταν η 3.7.3. Οι δοκιμές του κώδικα που περιέχει το βιβλίο έγιναν με διάφορες εκδόσεις της Python (μέχρι και την έκδοση 3.7).

Το όνομα της γλώσσας δεν είναι σχετικό με το γνωστό ερπετό, παρόλο που το λογότυπο που χρησιμοποιείται από το *Ίδρυμα της Python (Python Software Foundation²)*, που συντηρεί την κύρια έκδοση της γλώσσας³, παραπέμπει σε αυτό. Το όνομά της προέρχεται από την ομάδα κωμικών της δεκαετίας του 1970 με το όνομα Monty Python (*Flying Circus*), της βρετανικής τηλεόρασης (BBC).

1. <https://docs.python.org/3/faq/general.html#why-was-python-created-in-the-first-place>

2. <https://www.python.org/psf-landing/>

3. CPython, περισσότερα στοιχεία για τις διαφορετικές υλοποιήσεις της γλώσσας δίνονται στη συνέχεια.

1.2 Γιατί Python;

Η γλώσσα προγραμματισμού Python, παρόλο που χρησιμοποιείται εδώ και δυο δεκαετίες περίπου, έχει αποκτήσει, μόλις τα τελευταία χρόνια στην Ελλάδα, τεράστια απήχηση και αποδοχή, όχι μόνο στον χώρο της *επιστήμης των υπολογιστών* (*Computer Science*), αλλά και σε άλλους εμπορικούς και επιστημονικούς τομείς. Χαρακτηριστική είναι η ταχύτατη διείσδυση της γλώσσας τόσο σε πρακτικές εφαρμογές (σε επιχειρήσεις και οργανισμούς), όσο και στον τομέα της διδασκαλίας και εκμάθησης προγραμματισμού σε διάφορα επίπεδα εκπαίδευσης (από τη μέση μέχρι την ανώτατη εκπαίδευση).

Δεν είναι τυχαίο το ότι χρησιμοποιείται σε πάρα πολλούς και ασυνήθιστα, θα λέγαμε, διαφορετικούς τομείς δραστηριοτήτων και αυτό είναι κάτι που την κάνει μοναδική στο είδος της. Η ιδιαίτερη ευκολία εκμάθησης είναι ένα από τα χαρακτηριστικά της. Για αυτό τον λόγο πλέον αποτελεί, για πολλούς και πολλές, την πρώτη γλώσσα προγραμματισμού που θα μάθουν.

Υπάρχουν όμως και άλλοι ακόμα πολλοί λόγοι που θα οδηγήσουν κάποιον να στραφεί στην εκμάθηση της γλώσσας και να επιλέξει την Python ως τη γλώσσα προγραμματισμού που θα χρησιμοποιεί για τη δημιουργία προγραμμάτων. Στη συνέχεια παρατίθενται οι κυριότεροι από αυτούς.

Δίνοντας όμως πρώτα έναν σύντομο ορισμό για τη γλώσσα, θα λέγαμε ότι:

Η Python είναι μια δυναμική, δωρεάν γλώσσα προγραμματισμού, υψηλού επιπέδου, διερμηνευόμενη, γενικού σκοπού, για διάφορα συστήματα, που υλοποιεί πολλαπλά προγραμματιστικά υποδείγματα

Πιο αναλυτικά, η γλώσσα προγραμματισμού Python είναι:

1. **Δυναμική (Dynamic).** Κάθε *δομή δεδομένων* (*data structure*) (βλ. Εικόνα 1-1) που υποστηρίζει μπορεί να δημιουργηθεί και να αλλάξει δυναμικά κατά την εκτέλεση του προγράμματος. Δεν χρειάζεται να είναι προκαθορισμένη, κατασκευάζεται και τροποποιείται πλήρως σύμφωνα με τις ανάγκες του προγράμματος, την ώρα που αυτό τρέχει. Τα δεδομένα που υπάρχουν στη μνήμη του υπολογιστή υφίστανται δυναμική διαχείριση από την ίδια τη γλώσσα.
2. **Δωρεάν.** Είναι δωρεάν για κάθε χρήση (προσωπική και εμπορική) και διανέμεται ελεύθερα. Είναι *ανοιχτού κώδικα* (*open-source*) και χρησιμοποιεί την άδεια OSI⁴.
3. **Υψηλού επιπέδου (High-level).** Αυτό σημαίνει ότι παρέχει απλούς και εύκολα κατανοητούς κανόνες (συντακτικούς και γραμματικούς) για τη σύνταξη προγραμμάτων που είναι πολύ κοντά στην ανθρώπινη γλώσσα.
4. **Διερμηνευόμενη (Interpreted).** Η εκτέλεση ενός προγράμματος γραμμένου στη γλώσσα Python γίνεται με τη βοήθεια ενός ειδικού προγράμματος, του *διερμηνευτή* (*interpreter*), που αναλαμβάνει να ελέγξει την ορθότητά του και να πραγματοποιήσει την εκτέλεσή του.

4. <https://opensource.org/licenses/Python-2.0>

Σε κάθε πρόγραμμα, οποιασδήποτε γλώσσας προγραμματισμού (από την Assembly και τη C, μέχρι τη C#, την Python ή την Java), πρέπει να γραφτεί ο κώδικας – οι εντολές ενός προγράμματος, σε κάποιο (συνήθως απλό, χωρίς μορφοποιήσεις) αρχείο κειμένου. Αυτό το αρχείο ονομάζεται αρχείο *πηγαίου κώδικα* (*source code*) και περιέχει όλες τις οδηγίες προς τον υπολογιστή για την επίλυση ενός προβλήματος. Για να μπορέσει όμως να εκτελέσει ο υπολογιστής αυτές τις εντολές, θα πρέπει να μετατραπούν, να μεταφραστούν δηλαδή, σε γλώσσα κατάλληλη για τον υπολογιστή, σε *γλώσσα μηχανής* (*machine language, code*). Αυτή τη μετάφραση αναλαμβάνει να την κάνει ένα ειδικό πρόγραμμα που καλείται *μεταγλωττιστής* (*compiler*) ή *διερμηνευτής* (*interpreter*). Κάθε γλώσσα προγραμματισμού συνοδεύεται από ένα τέτοιο πρόγραμμα, διαθέσιμο είτε μέσα από κάποιο Ολοκληρωμένο Περιβάλλον Ανάπτυξης Λογισμικού (IDE, Integrated Development Environment)⁵, είτε μέσα από τη *γραμμή εντολών* (*command line*)⁶ του λειτουργικού συστήματος (Windows, Linux, MacOSx).

Ο μεταγλωττιστής αναλαμβάνει να ελέγξει όλες τις εντολές του πηγαιού κώδικα για την ύπαρξη *συντακτικών σφαλμάτων* (*syntax errors*). Αν εντοπίσει λάθη, θα σταματήσει και θα εμφανίσει, συνήθως κωδικοποιημένα, μια περιγραφή του σφάλματος και τη γραμμή όπου εντοπίστηκε αυτό. Αν το πρόγραμμα δεν έχει λάθη, η διαδικασία της μεταγλώττισης θα ολοκληρωθεί. Το αποτέλεσμα της μεταγλώττισης είναι η δημιουργία ενός αρχείου με *δυναμικό κώδικα* (*binary code*), *εκτελέσιμου* (*executable*) από τον υπολογιστή. Σε αρκετές περιπτώσεις η διαδικασία αυτή περιλαμβάνει και τη *σύνδεση* (*link*) του πηγαιού κώδικα και τη *φόρτωση* (*load*) άλλων προγραμμάτων και βιβλιοθηκών της γλώσσας, που είναι απαραίτητα για τη λειτουργία του προγράμματος.

Ο διερμηνευτής, από την άλλη, λειτουργεί αρκετά διαφορετικά. Τις περισσότερες φορές αναλαμβάνει να ελέγξει μια εντολή, να τη μεταφράσει σε κώδικα μηχανής και να την εκτελέσει. Αμέσως μετά θα πάει στην επόμενη εντολή για να επαναλάβει την ίδια διαδικασία. Αν εντοπίσει κάποιο σφάλμα σε μια από τις εντολές του προγράμματος, θα σταματήσει εκεί, εμφανίζοντας ένα κατάλληλο μήνυμα.

Η εκτέλεση ενός προγράμματος γραμμένου σε Python γίνεται από τον διερμηνευτή της γλώσσας. Κατά την εκτέλεση του προγράμματος δημιουργείται, στον ίδιο φάκελο όπου βρίσκεται το αρχείο πηγαιού κώδικα, ένα αρχείο με *δυναμικό κώδικα* (*binary code*), *εκτελέσιμου* (*executable*) από τον υπολογιστή.

5. Ένα IDE, όπως το περιβάλλον Spyder που περιγράφεται αμέσως μετά, προσφέρει έναν χώρο για τη συγγραφή προγραμμάτων, όπου ενσωματώνεται ένας *συντάκτης* (*editor*) για τη συγγραφή των εντολών αλλά και ο διερμηνευτής για την εκτέλεσή τους. Παρέχει συνήθως διάφορα εργαλεία και βοηθητικές λειτουργίες. Οι λειτουργίες αυτές περιλαμβάνουν τη χρωματική απόδοση του κώδικα, την αρίθμηση των εντολών-γραμμών του προγράμματος, τη βηματική εκτέλεση, την παροχή βοήθειας για τη σύνταξη των στοιχείων της γλώσσας, την αυτόματη συμπλήρωση εντολών, την παρακολούθηση των τιμών των στοιχείων της γλώσσας κ.λπ.

6. Η γραμμή εντολών είναι γνωστή και ως *τερματικό* (*terminal*) ή *κέλυφος* (*shell*), ανάλογα με το *λειτουργικό σύστημα* (*Operating System*) που χρησιμοποιείται.

- δικό κώδικα (*byte code*)⁷, με την επέκταση `.pyc`⁸ στο όνομα του αρχείου. Αυτό το αρχείο χρησιμοποιεί ο διερμηνευτής για την εκτέλεση του προγράμματος.
5. **Αντικειμενοστραφής (Object-oriented)**. Η Python είναι μια γλώσσα προσανατολισμένη στα αντικείμενα (*objects*). Έχει τα χαρακτηριστικά μιας αντικειμενοστραφούς γλώσσας, ωστόσο, υπάρχουν αρκετές διαφωνίες⁹ για το αν υλοποιεί πλήρως αυτό το μοντέλο προγραμματισμού. Αυτό οφείλεται στο γεγονός ότι εφαρμόζει μια χαλαρή πολιτική σε ένα από τα χαρακτηριστικά του συγκεκριμένου μοντέλου, το χαρακτηριστικό της *ενθυλάκωσης (encapsulation)*. Παρέχει έναν υποτυπώδη και έμμεσο τρόπο προστασίας της ιδιωτικότητας των αντικειμένων. Αυτό όμως είναι μια λειτουργία που περιγράφεται αναλυτικά στο Κεφάλαιο 11.
 6. **Γενικού σκοπού (General-purpose)**. Αυτό σημαίνει ότι μπορεί να χρησιμοποιηθεί σχεδόν για κάθε πρόβλημα και να καλύψει τις απαιτήσεις και τις ανάγκες για μεγάλη ποικιλία εφαρμογών. Αυτός άλλωστε είναι και ένας από τους βασικούς λόγους που χρησιμοποιείται σε τόσο πολλές και διαφορετικές περιπτώσεις.
 7. **Διαθέσιμη για διάφορα συστήματα**. Ένα πρόγραμμα σε Python, που έχει γραφτεί σε περιβάλλον Windows, μπορεί, τις περισσότερες φορές, να χρησιμοποιηθεί χωρίς αλλαγές και σε άλλα λειτουργικά συστήματα. Εξαιρούνται ορισμένες περιπτώσεις όπου κάποιες βιβλιοθήκες κώδικα λειτουργούν με διαφορετικό τρόπο από το ένα σύστημα στο άλλο.
 8. **Κατάλληλη για πολλαπλά προγραμματιστικά υποδείγματα (Multi-Paradigm)**. Αυτό το χαρακτηριστικό περιγράφεται στην επόμενη ενότητα. Εν συντομία θα αναφέρουμε μόνο ότι ενσωματώνει διαφορετικές λειτουργίες και δίνει απόλυτη ελευθερία στον δημιουργό ενός προγράμματος για το ποια χαρακτηριστικά θα χρησιμοποιήσει κάθε φορά.
 9. **Ικανή να ενσωματώνει υπηρεσίες (Frameworks)**. Αρκετά διαδεδομένη είναι η χρήση έτοιμων συστημάτων υπηρεσιών, που υλοποιούνται σε Python. Κυρίως αφορούν την εγκατάσταση και λειτουργία διαδικτυακών υπηρεσιών, όπως *διακομιστές ιστοσελίδων (web servers)*, *υπηρεσίες νέφους (cloud services)* κ.λπ. Τα πιο γνωστά από αυτά είναι το Flask¹⁰ και το Django¹¹.

7. <https://docs.python.org/3/tutorial/modules.html#compiled-python-files>

8. Κάθε υλοποίηση (βλ. επόμενη παράγραφο) της γλώσσας χρησιμοποιεί διαφορετικό τρόπο παραγωγής δυαδικού κώδικα. Τα συγκεκριμένα αρχεία χρησιμοποιούνται στην έκδοση CPython που χρησιμοποιείται στο βιβλίο. Αντίθετα, η Jython, για παράδειγμα, παράγει κώδικα που τρέχει με την *εικονική μηχανή της Java (JVM, Java Virtual Machine)*.

9. Πρόκειται για διαφορά στη φιλοσοφία της γλώσσας σε σχέση με άλλες (όπως η C++ και η Java). Στον αντίποδα των διαφωνιών, στη Mailing List της Python γίνεται αναφορά για την ομοιότητά της με τη γλώσσα Perl, όπως διατυπώνεται στο βιβλίο *Programming Perl* (γνωστό και ως *Camel Book*): "a Perl module would prefer that you stayed out of its living room because you weren't invited, not because it has a shotgun."

10. <http://flask.pocoo.org>

11. <https://www.djangoproject.com>

10. **Υλοποιήσιμη σε πολλές εκδοχές.** Υπάρχουν αρκετές διανομές της Python, με διαφορετικές υλοποιήσεις (*implementations*). Η επίσημη έκδοση¹² είναι γνωστή με το όνομα CPython, αφού χρησιμοποιεί τη γλώσσα C σε διάφορες λειτουργίες και σε ενσωματωμένες βιβλιοθήκες της. Εκτός όμως από αυτήν υπάρχουν και άλλες εκδόσεις της γλώσσας, η καθεμία με τα δικά της χαρακτηριστικά και ιδιότητες, που κυρίως αφορούν τη συνεργασία και την επικοινωνία της με άλλα συστήματα γλωσσών προγραμματισμού. Ακολουθούν οι πιο γνωστές από αυτές:

- ⊕ **Cython.** Είναι ένας μεταγλωττιστής τόσο για κώδικα γραμμένο σε απλή Python, όσο και σε ένα υπερσύνολο της γλώσσας που ονομάζεται (επίσης) Cython. Χρησιμοποιείται για τη συγγραφή *επεκτάσεων κώδικα* (*extensions of code*) σε C και τη σύνδεσή τους με προγράμματα σε Python. Είναι δυνατή η δημιουργία εφαρμογών για την κλήση προγραμμάτων σε C/C++ ή την ενσωμάτωσή τους σε αυτά.
- ⊕ **Jython.** Κάνει δυναμική μεταγλώττιση σε *δυναμικό κώδικα* (*byte code*) της Java και μπορεί να *επεκτείνει* (*extend*) τις κλάσεις της.
- ⊕ **Iron Python.** Είναι μια υλοποίηση της γλώσσας στενά συνδεδεμένη με το Net Framework, το οποίο μπορεί και χρησιμοποιεί. Άλλες γλώσσες του Net Framework μπορούν να χρησιμοποιούν προγράμματα φτιαγμένα σε Iron Python.
- ⊕ **Brython.** Είναι έκδοση της γλώσσας που σκοπό έχει να αντικαταστήσει την Javascript στην κατασκευή *κώδικα που εκτελείται σε ιστοσελίδες* (*web programming*). Χρησιμοποιεί και αξιοποιεί πλήρως το περιβάλλον της HTML 5.
- ⊕ **Pyjs.** Κάνει εύκολη τη δημιουργία εφαρμογών για ιστοσελίδες γραμμένες εξ ολοκλήρου σε Python.

11. **Επεκτάσιμη (Extendable).** Μπορεί εύκολα να συνεργαστεί με άλλες γλώσσες προγραμματισμού. Σε αυτή τη λειτουργία βοηθούν και οι διάφορες υλοποιήσεις της Python.

Δεν είναι τυχαίο ότι χρησιμοποιείται (μαζί με άλλες γλώσσες προγραμματισμού) από εταιρείες¹³ και οργανισμούς όπως οι Dropbox, Facebook, Spotify, Netflix, Quora, Reddit, IBM, Nokia, Disney, Yahoo!, NASA, JP Morgan, Instagram, PayPal, eBay κ.ά. Χαρακτηριστικά, ένας από τους μηχανικούς λογισμικού της Google αναφέρει σε άρθρο¹⁴ του, για την απόφαση της εταιρείας να χρησιμοποιήσει την Python, ότι ήταν μια καλή επιλογή σχεδιασμού η λογική:

Python where we can, C++ where we must

12. <https://www.python.org>

13. <https://www.fullstackpython.com/companies-using-python.html>

14. <https://stackoverflow.com/questions/2560310/heavy-usage-of-python-at-google>

1.3 Προγραμματιστικά υποδείγματα

Η έννοια του *προγραμματιστικού υποδείγματος* (*programming paradigm*) συνδέεται με τις τεχνικές, τις λειτουργίες και τους μηχανισμούς που χρησιμοποιεί μια γλώσσα προγραμματισμού. Η γλώσσα Python διαθέτει χαρακτηριστικά και λειτουργίες *πολλαπλών προγραμματιστικών υποδειγμάτων* (*multi-programming paradigms*). Ο τρόπος χρήσης της και το μοντέλο που θα χρησιμοποιηθεί σε κάθε πρόγραμμα εξαρτάται αποκλειστικά από τον χρήστη και τον σχεδιασμό που αυτός θα κάνει σε κάθε περίπτωση, καθώς και ανάλογα με το πρόβλημα που πρέπει να αντιμετωπίσει. Σε κάθε υλοποίηση ενός προγράμματος, υπάρχουν τα κατάλληλα εργαλεία, με τις αντίστοιχες τεχνικές, που μπορεί κάποιος να επιλέξει, ώστε να επιτύχει με τον καλύτερο τρόπο τον στόχο του. Τα υποδείγματα που μπορεί να ακολουθεί ένα πρόγραμμα γραμμένο σε Python είναι τα παρακάτω:

1. **Διαδικαστικός προγραμματισμός (Procedural Programming).** Η υλοποίηση περιλαμβάνει την ιεραρχική σχεδίαση του προγράμματος και τη δημιουργία μιας σειράς από αυτόνομα τμήματα κώδικα, με τη μορφή υποπρογραμμάτων. Για αυτό τον λόγο, συνήθως ακολουθείται ο σχεδιασμός *από πάνω προς τα κάτω* (*top to bottom*), από το γενικό προς το ειδικό. Ένα πρόβλημα, επομένως, σπάει, ή χωρίζεται, σε ανεξάρτητα τμήματα, τα οποία στη συνέχεια θα αποτελέσουν αντίστοιχες ενότητες υποπρογραμμάτων, με τη μορφή συναρτήσεων ή βιβλιοθηκών κώδικα. Τα χαρακτηριστικά του *Δομημένου Προγραμματισμού* (*Structured Programming*) είναι διαθέσιμα και χρησιμοποιούνται για αυτό τον σκοπό. Είναι οι δομές της ακολουθίας (Κεφάλαιο 2), της επιλογής (Κεφάλαιο 3) και της επανάληψης (Κεφάλαιο 4).
2. **Προστακτικός προγραμματισμός (Imperative Programming).** Δίνεται έμφαση στο πώς θα λυθεί ένα πρόβλημα. Το πρόγραμμα δημιουργείται με σκοπό να αλλάζει την κατάσταση (*state*) σε στοιχεία του υπολογιστή (μνήμη, επεξεργαστή κ.λπ.), ώστε τελικά να μπορέσει να λύσει το δοσμένο πρόβλημα.
3. **Αντικειμενοστραφής προγραμματισμός (Object-oriented Programming).** Κύριο συστατικό αυτού του μοντέλου είναι οι κλάσεις αντικειμένων που περιγράφονται στο Κεφάλαιο 11. Κάθε οντότητα μέσα σε ένα πρόγραμμα παίρνει τη μορφή ενός *αντικειμένου* (*object*), που ενσωματώνει μέσα του κώδικα και δεδομένα. Έτσι, κάθε αντικείμενο διαθέτει όχι μόνο γνωρίσματα και ιδιότητες, αλλά και κώδικα που προσδιορίζει τη συμπεριφορά του και χειρίζεται τα δεδομένα του.
4. **Συναρτησιακός προγραμματισμός (Functional Programming).** Αν και η Python δεν ακολουθεί πιστά αυτό το μοντέλο (όπως άλλες γλώσσες σαν τη γλώσσα Haskell), ωστόσο διαθέτει αρκετά από τα χαρακτηριστικά του. Ένα από αυτά είναι η δυνατότητα μιας εντολής να παίρνει τη μορφή μαθηματικής συνάρτησης, όπου το αποτέλεσμα παράγεται μέσα από δηλώσεις και εκφράσεις, και όχι από εντολές ή οδηγίες. Τέτοια χαρακτηριστικά ενσωματώνουν οι λειτουργίες των ανώνυμων συναρτήσεων και του «κλεισίματος» που περιγράφονται στο Κεφάλαιο 10 με τις συναρτήσεις.

1.4 Σε ποιους απευθύνεται το βιβλίο

Η γλώσσα προγραμματισμού Python αποτελεί, τη στιγμή που γράφεται το βιβλίο, μια από τις πρώτες επιλογές ως γλώσσα εισαγωγής στον προγραμματισμό. Το βιβλίο δεν απευθύνεται μόνο σε φοιτητές τμημάτων σχετικών με την πληροφορική, που έχουν ήδη κάποια εμπειρία με μια ή περισσότερες γλώσσες προγραμματισμού υψηλού επιπέδου, όπως η C/C++, η Fortran, η Java κ.ά., ή έχουν ασχοληθεί με περιβάλλοντα επιστημονικών υπολογισμών, όπως το Matlab, το Octave, το Sage, το Maple ή το Mathematica· απευθύνεται και σε όσους θέλουν να χρησιμοποιήσουν την Python ως εργαλείο για την πραγματοποίηση έρευνας και την επεξεργασία δεδομένων με σκοπό την κατασκευή προγραμμάτων σε τομείς όπως η *επιστήμη δεδομένων (Data Science)*, η *στατιστική* (όπου η Python ανταγωνίζεται, επί ίσοις όροις, και με μεγάλη επιτυχία τη γλώσσα R), σε θέματα *τεχνητής νοημοσύνης (Artificial Intelligence)*, όπως η *μηχανική μάθηση (Machine Learning)*, τα *νευρωνικά δίκτυα (Neural Networks)*, στην οικονομία, στα μαθηματικά, σε τομείς των φυσικών επιστημών και ειδικότερα στη φυσική και τις βιοεπιστήμες και, γενικά, σε κάθε χώρο που απαιτείται η επεξεργασία δεδομένων μεγάλου όγκου για την πραγματοποίηση υπολογισμών.

Σκοπός μας ήταν να καταστήσουμε τα στοιχεία της Python κατανοητά και προσιτά και σε μαθητές Γυμνασίου και Λυκείου, που θέλουν να ξεκινήσουν τη γνωριμία τους με θέματα προγραμματισμού υπολογιστών. Στη δομή του βιβλίου ελήφθη υπόψη το γεγονός ότι οι μαθητές που μαθαίνουν προγραμματισμό στα σχολικά εργαστήρια πληροφορικής έχουν ήδη μια πρώτη επαφή με περιβάλλοντα όπως η Logo ή η Pascal (παλιότερα), η Visual Basic σε κάποιες περιπτώσεις, κυρίως το Scratch και το AppInventor, αλλά και σε αρκετές περιπτώσεις η C (είτε ως εργαλείο για την ενασχόληση σε θέματα IoT – Internet of Things, είτε ως μια γλώσσα για τη συμμετοχή στον Πανελλήνιο Διαγωνισμό Πληροφορικής), ή ακόμα, και η Java (ειδικά στα σχολεία που χρησιμοποιούν το Greenfoot ή το Alice και είναι μέλη του Oracle Academy).

Ένα μεγάλο άλλωστε ποσοστό της γενιάς επιστημόνων και αποφοίτων μέσης εκπαίδευσης, της τελευταίας 20ετίας, έχουν κοινό χαρακτηριστικό την εξοικείωσή τους με βασικές έννοιες του προγραμματισμού μέσα από την εκμάθηση της ψευδογλώσσας για τη δημιουργία *αλγορίθμων (algorithms)*. Εξάλλου, η Python ήδη αποτελεί τη γλώσσα στην οποία εξετάζονται σε πανελλαδικό επίπεδο οι υποψήφιοι για εισαγωγή στην τριτοβάθμια εκπαίδευση μαθητές των επαγγελματικών λυκείων, όπου μαθαίνουν τα βασικά χαρακτηριστικά και τις δομές της.

Φροντίσαμε λοιπόν η εισαγωγή σε έννοιες και δομές, που χρησιμοποιούνται σε ήδη γνώριμα περιβάλλοντα προγραμματισμού τα οποία αναφέρθηκαν πιο πριν, να γίνει με τρόπο κατανοητό, ώστε να φανεί η διαφορά τους με την Python, και οικείο για τους περισσότερους αναγνώστες. Για αυτό τον λόγο, σε ορισμένα σημεία του βιβλίου γίνεται αναφορά στις τεχνικές που χρησιμοποιούνται σε άλλες διαδικαστικές γλώσσες. Για παράδειγμα, η εύρεση του αθροίσματος των στοιχείων ενός πίνακα με τη χρήση αθροιστή (περισσότερα για αυτό το θέμα βλ. Κεφάλαιο 4 και Κεφάλαιο 5), ενώ αποτελεί βασική λειτουργία των πινάκων και είναι από τα βασικά που μαθαίνουμε στους μαθητές και φοιτητές μας για την ψευ-

δογλώσσα και τη C, είναι κάτι που σπάνια θα χρησιμοποιήσουμε στην Python, δουλεύοντας με μια (αντίστοιχη με τους πίνακες) δομή όπως η λίστα.

Το βιβλίο θα καλύψει τις ανάγκες τόσο του αρχάριου όσο και του προχωρημένου σε θέματα προγραμματισμού χρήστη. Δεν απαιτούνται, ωστόσο, προηγούμενες γνώσεις σε κάποια άλλη γλώσσα προγραμματισμού. Για κάθε βασικό αντικείμενο της Python, ξεκινά η περιγραφή του από τα βασικά και απολύτως απαραίτητα και προχωρεί σε εμβάθυνση στις τεχνικές και στις λειτουργίες του. Σχεδόν σε κάθε κεφάλαιο περιγράφεται μια από τις ενσωματωμένες βιβλιοθήκες της γλώσσας, όπως τα `frozensets`, η `math`, η `random`, αλλά και άλλες, λιγότερο γνωστές, όπως η `pickle`, η `array`, η `fraction` κ.ά.

Το βιβλίο που κρατάς στα χέρια σου ελπίζουμε ότι θα σε βοηθήσει στη γνωριμία με τον ιδιαίτερο χαρακτήρα και τα γνωρίσματα της Python, είτε αυτή αποτελεί την πρώτη είτε την επόμενη γλώσσα προγραμματισμού που θα μάθεις.

1.5 Περίληψη κεφαλαίων

Το βιβλίο περιλαμβάνει τα επόμενα κεφάλαια. Δίνεται μια σύντομη περιγραφή για το καθένα. Η σειρά ανάγνωσης δεν είναι απαραίτητο να είναι γραμμική.

1. **Εισαγωγή.** Δίνονται κάποιες βασικές πληροφορίες για την Python και για τα χαρακτηριστικά της, κάποια ιστορικά στοιχεία και λεπτομέρειες σχετικά με διάφορες σημάνσεις που υπάρχουν στο βιβλίο. Περιγράφονται οι βασικοί τρόποι εργασίας με τον διερμηνευτή και η διαδικασία εγκατάστασης της γλώσσας.
2. **Μεταβλητές και αριθμοί.** Αναφέρονται πληροφορίες για τους τύπους αριθμών που υποστηρίζει η Python, καθώς επίσης και κάποια βασικά στοιχεία για τη διαχείριση μεταβλητών. Γίνεται αναφορά στη διαχείριση πακέτων βιβλιοθηκών και τη χρησιμοποίησή τους μέσα σε κάποιο πρόγραμμα. Παρουσιάζονται επίσης κάποιες ενσωματωμένες βιβλιοθήκες διαχείρισης αριθμών.
3. **Δομές επιλογής.** Γίνεται αναλυτική περιγραφή των δομών επιλογής της Python και των απαραίτητων εργαλείων για τη δημιουργία λογικών ελέγχων και την κατασκευή λογικών συνθηκών και εκφράσεων. Παρουσιάζονται οι λογικοί και σχεσιακοί τελεστές, καθώς και οι δυνατότητες της γλώσσας για τη διαχείριση σφαλμάτων κατά την εκτέλεση του προγράμματος.
4. **Δομές επανάληψης.** Παρουσιάζονται οι επαναληπτικές δομές της Python και οι τρόποι που μπορούν αυτές να χρησιμοποιηθούν στην πράξη. Περιγράφεται η λειτουργία και η ενσωμάτωση ένθετων δομών.
5. **Λίστες.** Περιγράφεται μια από τις δομές ακολουθίας που διαθέτει η Python, η δομή της λίστας. Μαζί με τις σχετικές μεθόδους διαχείρισης των δεδομένων και των ιδιοτήτων μιας λίστας, δίνονται παραδείγματα για τη δημιουργία και τη χρήση της.
6. **Συμβολοσειρές.** Παρουσιάζονται τα χαρακτηριστικά των συμβολοσειρών και οι μέθοδοι χειρισμού τους, μαζί με τις διαφορετικές μορφές που μπορεί να έχουν μέσα στον κώδικα. Δίνονται πληροφορίες σχετικά με τους τρόπους κω-

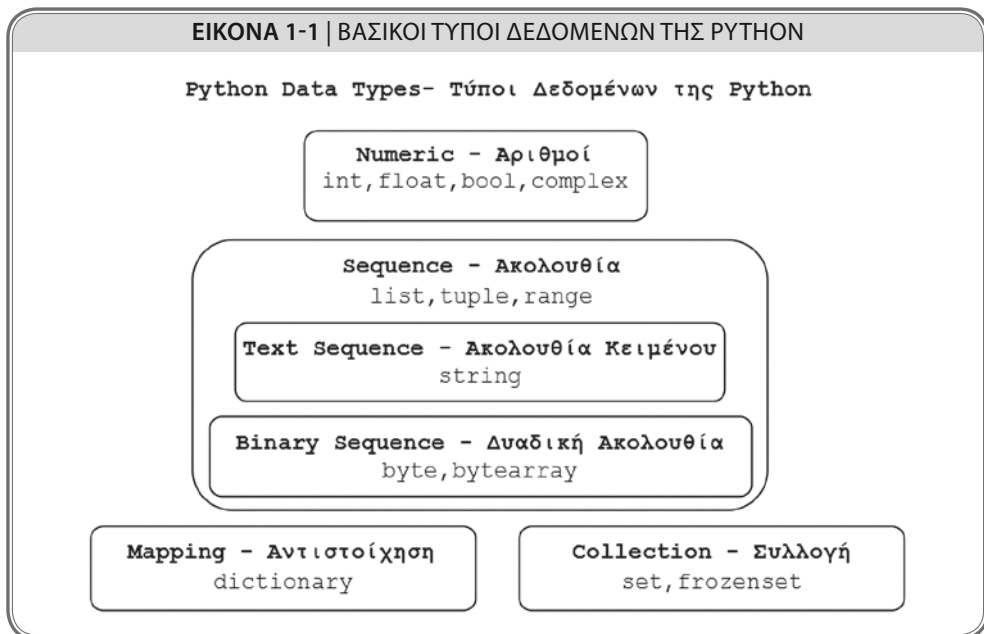
- δικοποίησης χαρακτήρων και τις μεθόδους μετατροπής κειμένου από μια μορφή σε μια άλλη.
7. **Πλειάδες.** Δίνονται τα χαρακτηριστικά, οι ιδιότητες και οι λειτουργίες της πλειάδας. Γίνεται εισαγωγή στην έννοια της γεννήτριας τιμών και πώς αυτή μπορεί να χρησιμοποιηθεί, στην πράξη, μαζί με άλλες δομές, καθώς και στη λειτουργία του πακεταρίσματος τιμών.
 8. **Σύνολα.** Περιγράφεται η δομή του συνόλου, η σύνδεσή του με τη θεωρία συνόλων, οι ιδιότητές του και οι μέθοδοι χειρισμού του περιεχομένου του. Μέσα από παραδείγματα, δίνεται έμφαση στην πρακτική αξιοποίηση των ιδιοτήτων των πράξεων συνόλων, για την επίλυση σύνθετων προβλημάτων.
 9. **Λεξικά.** Γίνεται περιγραφή των χαρακτηριστικών του λεξικού, της μορφής που μπορεί να έχει αυτό και των λειτουργιών του. Δίνονται παραδείγματα για την εισαγωγή, τη διαγραφή ή την ενημέρωση των στοιχείων που περιέχει, καθώς επίσης και για τους τρόπους που μπορεί να χρησιμοποιηθεί στην πράξη.
 10. **Συναρτήσεις & βιβλιοθήκες.** Το Κεφάλαιο 10 περιλαμβάνει την περιγραφή των τρόπων κατασκευής συναρτήσεων, γεννητριών, συναρτήσεων λ , βιβλιοθηκών και πακέτων. Παρουσιάζονται οι διαφορετικές μορφές συναρτήσεων που υποστηρίζει η Python, καθώς και οι σύνθετες λειτουργίες όπως το «κλείσιμο» και η διακόσμηση.
 11. **Αντικείμενα.** Το κεφάλαιο 11 αφορά τον αντικειμενοστραφή τρόπο λειτουργίας της Python, τη δημιουργία σύνθετων δομών, όπως οι κλάσεις αντικειμένων, καθώς και τις ειδικές λειτουργίες που μπορούν αυτές να ενσωματώνουν. Μέσα από παραδείγματα, αναλύονται όλα τα χαρακτηριστικά, οι ιδιότητες και τα ιδιαίτερα γνωρίσματα του προσανατολισμένου στα αντικείμενα προγραμματιστικού μοντέλου που υλοποιείται στην Python.
 12. **Αρχεία.** Στο τελευταίο κεφάλαιο του βιβλίου γίνεται αναφορά και περιγραφή των βασικών τύπων αρχείων που διαχειρίζεται η Python, των αρχείων κειμένου και των δυαδικών αρχείων. Παρουσιάζονται, μέσα από μικρές εφαρμογές και παραδείγματα, οι τρόποι επεξεργασίας του περιεχομένου αρχείων και διαχείρισής του από τις δομές της γλώσσας. Γίνεται, τέλος, περιγραφή δυο βιβλιοθηκών διαχείρισης περιεχομένου συχνά χρησιμοποιούμενων αρχείων.
 13. **Παράρτημα Α.** Μαθηματικοί τύποι. Αναφέρονται χρήσιμοι βασικοί μαθηματικοί τύποι που χρησιμοποιούνται συχνά.
 14. **Παράρτημα Β.** Πίνακες χαρακτήρων. Δίνονται οι πίνακες των χαρακτήρων ASCII και των κωδίκων Unicode για τα ελληνικά. Στο τέλος παρατίθεται και ο αντίστοιχος κώδικας σε Python, για τη δημιουργία αυτών των πινάκων.
 15. **Παράρτημα Γ.** Μια σελίδα π . Παρατίθενται μερικά (τα πρώτα 3,141) από τα ψηφία του αριθμού π , καθώς και ο αντίστοιχος κώδικας σε Python που τα δημιουργεί.
 16. **Βιβλιογραφία.** Πηγές και αξιόλογα βιβλία για την εκμάθηση της γλώσσας Python. Αφορούν διάφορα θέματα της Python.
 17. **Γλωσσάρι.** Αντιστοιχίες μεταφρασμένων όρων (από τα αγγλικά στα ελληνικά) που χρησιμοποιούνται μέσα στο βιβλίο. Περιέχει επίσης τον ορισμό και την

ερμηνεία κάποιων βασικών εννοιών σχετικών με λειτουργίες της Python.

18. **Ευρετήριο όρων.** Δίνονται βασικοί όροι και στοιχεία που χρησιμοποιούνται στο βιβλίο, μαζί με τις αντίστοιχες παραπομπές τους στο κείμενο.

1.6 Τύποι δεδομένων

Στο βιβλίο περιγράφονται οι τρόποι λειτουργίας, οι ιδιότητες και οι μέθοδοι χειρισμού των τύπων δεδομένων της Python που φαίνονται συνοπτικά και σε κατηγορίες στην Εικόνα 1-1.



1.7 Εγκατάσταση της Python

Για να μπορέσει κάποιος να φτιάξει ένα πρόγραμμα σε Python, θα πρέπει να διαθέτει στον υπολογιστή του τη γλώσσα. Αυτό μπορεί να γίνει με διάφορους τρόπους και ανάλογα με το λειτουργικό σύστημα (*Operating System*) που διαθέτει. Στις επόμενες παραγράφους περιγράφεται η εγκατάσταση (*installation - setup*) της Python σε συστήματα Windows και Linux.

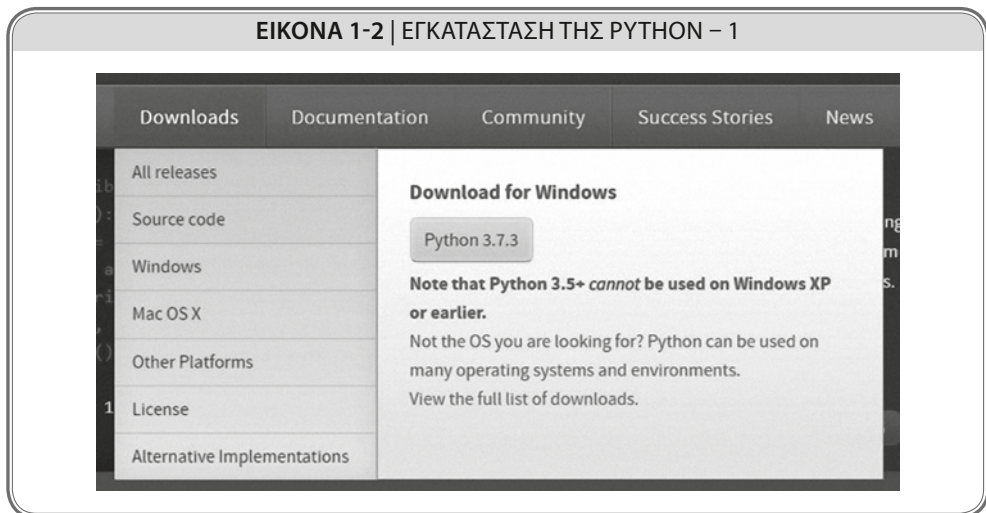
1.7.1 Windows

Τα Windows δεν έχουν εγκατεστημένη τη γλώσσα Python. Αυτό όμως μπορεί να γίνει πολύ εύκολα και με πολλούς τρόπους. Για να εργαστεί κάποιος με την Python σε περιβάλλον Windows, μπορεί να ακολουθήσει έναν από τους δυο παρακάτω τρόπους.

1. Να εγκαταστήσει το περιβάλλον διερμηνευτή της επίσημης έκδοσης της γλώσσας¹⁵. Αυτό του δίνει τη δυνατότητα να έχει πρόσβαση στην κονσόλα του διερμηνευτή (με το σήμα >>>) αλλά και στο (πολύ απλό, που δεν απαιτεί πολλούς υπολογιστικούς πόρους) IDE που διαθέτει. Μπορεί να εγκατασταθεί παράλληλα και κάποιο πιο σύνθετο και με περισσότερες δυνατότητες IDE, όπως το Pycharm, που δίνει τη δυνατότητα για εγκατάσταση πακέτων βιβλιοθηκών και τη βηματική εκσφαλμάτωση (*debugging*) των προγραμμάτων.
2. Να εγκαταστήσει ένα ολοκληρωμένο περιβάλλον (IDE), όπως το Spyder, μέσα από το σύστημα Anaconda ή το Canopy Enthought. Αν γίνει αυτό, δεν χρειάζεται να εγκατασταθεί και ο διερμηνευτής της Python, αφού και τα δυο αυτά περιβάλλοντα περιλαμβάνουν τον δικό τους διερμηνευτή, με τη δική τους (IPython) κονσόλα εργασίας (σήμα προτροπής [xx]:). Επίσης, και τα δυο ενσωματώνουν έναν πολύ μεγάλο αριθμό προεγκατεστημένων (*preinstalled*) βιβλιοθηκών και διαθέτουν δικό τους σύστημα αναζήτησης και διαχείρισης βιβλιοθηκών.

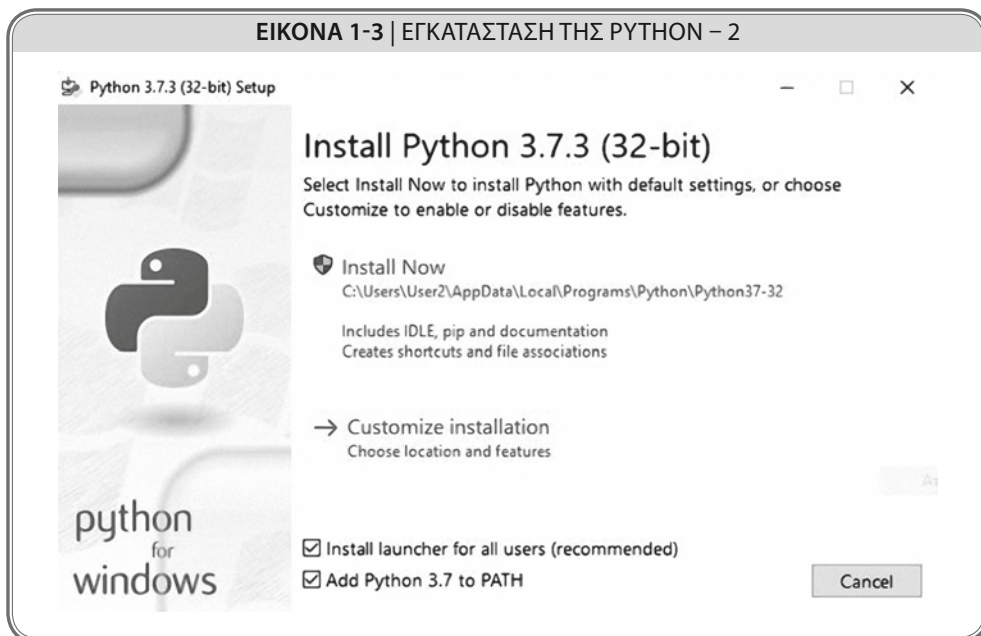
1.7.1.1 Εγκατάσταση της επίσημης (official) έκδοσης της Python

Η γλώσσα είναι διαθέσιμη από τον επίσημο ιστότοπό¹⁶ της. Μπορεί να ληφθεί ως πακέτο εγκατάστασης. Από την κεντρική σελίδα και την επιλογή Downloads (Εικόνα 1-2), επιλέγουμε το λειτουργικό σύστημα Windows και κατεβάζουμε το αρχείο εγκατάστασης.



Μόλις ολοκληρωθεί η λήψη, με διπλό κλικ στο πρόγραμμα εγκατάστασης ξεκινά η διαδικασία εγκατάστασης της γλώσσας. Στο πρώτο παράθυρο διαλόγου που εμφανίζεται (Εικόνα 1-3) επιλέγουμε INSTALL NOW:

15, 16. <https://www.python.org>



Φροντίζουμε να είναι επιλεγμένο το πλαίσιο “Add Python to PATH”, ώστε ο διεργμηνευτής της γλώσσας να μπορεί να εκτελεί τα προγράμματα από οποιονδήποτε φάκελο του σκληρού δίσκου. Αν δεν γίνει αυτό κατά την εγκατάσταση, στο Κεφάλαιο 10 περιγράφεται η διαδικασία που πρέπει να ακολουθηθεί.

Η εγκατάσταση θα ξεκινήσει εμφανίζοντας το παράθυρο της Εικόνας 1-4.

