Εισαγωγή στον Προγραμματισμό Η/Υ

Διδάσκοντες: Ι. Πολενάκης, Κ. Σκιαδόπουλος, Δ. Μουρατίδου

Τμήμα Πληροφορικής, Ιόνιο Πανεπιστήμιο

Περιεχόμενο (Syllabus)

- 1η Εβδομάδα: Εισαγωγή Στην Ανάλυση και Μοντελοποίηση Προβλημάτων
- 2η Εβδομάδα: Σχεδίαση Δομής Αλγοριθμικής Προσέγγισης
- 3η Εβδομάδα: Συντακτικά Στοιχεία και Μεταβλητές.
- 4η Εβδομάδα: Εντολές Ελέγχου και Διακλάδωσης.
- 5η Εβδομάδα: Συνθήκες και Βρόχοι Επανάληψης.
- 6η Εβδομάδα: Συναρτήσεις και Αρθρωτός Προγραμματισμός.
- 7η Εβδομάδα: Αναδρομή.
- 8η Εβδομάδα: Δομές Δεδομένων (Λίστες, Πλειάδες, Λεξικά, Σύνολα).
- 9η Εβδομάδα: Είσοδος/Έξοδος και Αρχεία.
- 10η Εβδομάδα: Διαχείριση Εξαιρέσεων και Αποσφαλμάτωση.
- 11η Εβδομάδα: Βασικές Έννοιες Αντικειμενοστραφούς Προγραμματισμού.
- 12η Εβδομάδα: Ανάπτυξη Εφαρμογών με τη Python.
- 13η Εβδομάδα: Ανάπτυξη Διεπαφής Χρήστη με τη Python.

Συνιστώμενη βιβλιογραφία προς μελέτη

- **Μαθαίνοντας προγραμματισμό με την Python** — Νίκος Μαμουλής, ΠΕΔΙΟ, 2023, ISBN: 122084640
- **Το βιβλίο της Python - Γράφοντας κώδικα** — Σαμαράς Νικόλαος, Τσιπλίδης Κωνσταντίνος, Κριτική, 2019, ISBN: 86055492
- **Εισαγωγή στον Προγραμματισμό με την Python** — Schneider David, Χ. ΓΚΙΟΥΡΔΑ ΣΙΑ ΕΕ, 2016, ISBN: 59357236

Αξιολόγηση (Τελικός Βαθμός)

Ο Τελικός Βαθμός (ΤΒ) ορίζεται επι των βαθμών του εργαστηρίου (ΒΕ) και βαθμού γραπτής εξέταση (ΒΓ) ως:

$$\mathbf{ΤΒ} = [\mathbf{ΒΕ} \cdot (\mathbf{0,3})] + [\mathbf{ΒΓ} \cdot (\mathbf{0,7})]$$

ΔΕΔΟΜΕΝΟΥ ΌΤΙ ... $\mathbf{ΒΓ \geq 5}$

Αξιολόγηση (Τελικός Βαθμός)

Ο Τελικός Βαθμός (ΤΒ) ορίζεται επι των βαθμών του εργαστηρίου (ΒΕ) και βαθμού γραπτής εξέταση (ΒΓ) ως:

```
BE = float(input("Δώσε βαθμό Εργαστηρίου (0-10): "))
BG = float(input("Δώσε βαθμό Γραπτής Εξέτασης (0-10): "))

# Έλεγχος προϋπόθεσης
if BG >= 5:
    TB = BE * 0.3 + BG * 0.7
    print(f"Ο Τελικός Βαθμός είναι: {TB:.2f}")
else:
    print("Αποτυχία: Ο βαθμός Γραπτής Εξέτασης (BG) πρέπει να είναι τουλάχιστον 5.")
```

Στόχος Μαθήματος

- Εισαγωγή σε αλγοριθμική σκέψη και θεμελιώδη στοιχεία Python για επίλυση προβλημάτων.
- Μέθοδος: Σύντομες θεωρητικές ενότητες + εργαστήρια (hands-on).
- Με το τέλος του μαθήματος ο/η φοιτητής/τρια θα μπορεί να:
  - Κατανοεί βασικές έννοιες αλγορίθμων (διαδοχή, επιλογή, επανάληψη).
  - Μεταφράζει προβλήματα σε βήματα (αλγόριθμος).
  - Υλοποιεί αλγορίθμους σε Python.
  - Δοκιμάζει και αποσφαλματώνει απλά προγράμματα.
  - Χρησιμοποιεί βασικές δομές δεδομένων (λίστες, πλειάδες, λεξικά).

Αλγοριθμική Σκέψη

- **Κύρια στοιχεία**:
  - Διάσπαση προβλήματος σε μικρά βήματα (decomposition).
  - Αφαίρεση/εξατομίκευση (abstraction).
  - Αναγνώριση προτύπων (pattern recognition).
  - Σχεδιασμός σωστής και αποδοτικής λύσης.

Διαδικασία Επίλυσης Προβλήματος

- **Βήματα:**

1. Κατανόηση εισροών/εκροών & περιορισμών.
2. Σχεδιασμός αλγορίθμου (ψευδοκώδικας / διάγραμμα ροής).
3. Υλοποίηση σε Python.
4. Δοκιμή με κανονικά & ακραία παραδείγματα.
5. Αποσφαλμάτωση και βελτιστοποίηση.

Σύντομη Εισαγωγή στους Αλγορίθμους

- Ως **αλγόριθμος** ορίζεται μια πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, που στοχεύουν στην επίλυση ενός προβλήματος.

- Πιο απλά (**αλγόριθμο**) ονομάζουμε μία σειρά από εντολές που έχουν αρχή και τέλος, είναι σαφείς και έχουν ως σκοπό την επίλυση κάποιου προβλήματος.

- **Εντολή** του αλγορίθμου είναι καθένα από τα διακριτά βήματά του, το οποίο προσδιορίζει μια σαφή ενέργεια.

Σύντομη Εισαγωγή στους Αλγορίθμους

Δημιουργία αλγορίθμου

Τα βήματα δημιουργίας αλγόριθμου είναι:

1. Διατύπωση του προβλήματος
2. Κατανόηση του προβλήματος
3. Λύση του προβλήματος
4. Διατύπωση του αλγόριθμου
5. Έλεγχος της λύσης

Σύντομη Εισαγωγή στους Αλγορίθμους

Κριτήρια αλγορίθμου

Οι αλγόριθμοι θα πρέπει να πληρούν κάποια πρότυπα και να διατυπώνονται με συγκεκριμένο τρόπο.

Έτσι ένας αλγόριθμος πρέπει να ικανοποιεί τα επόμενα κριτήρια:

- Καθοριστικότητα
- Περατότητα
- Αποτελεσματικότητα
- Επεκτασιμότητα
- Να έχει είσοδο δεδομένων, επεξεργασία και έξοδο αποτελεσμάτων

Σύντομη Εισαγωγή στους Αλγορίθμους

Κριτήρια αλγορίθμου

Οι αλγόριθμοι θα πρέπει να πληρούν κάποια πρότυπα και να διατυπώνονται με συγκεκριμένο τρόπο.

- **Καθοριστικότητα**
  - Κάθε κανόνας του ορίζεται επακριβώς και η αντίστοιχη διεργασία είναι συγκεκριμένη. Κάθε εντολή πρέπει να καθορίζεται χωρίς καμία αμφιβολία για τον τρόπο εκτέλεσής της. Π.χ. Σε μία διαίρεση να λαμβάνεται υπόψη και η περίπτωση όπου ο διαιρετέος λαμβάνει μηδενική τιμή.
  - Τυπικές περιπτώσεις η διαίρεση με το μηδέν, υπόριζος ποσότητα αρνητική, κλπ.
  - Προβλήματα καθοριστικότητας αντιμετωπίζονται συχνά με τη λογική της επιλογής, δηλ. Αν $\alpha>0$ τότε ..... αλλιώς .....

Σύντομη Εισαγωγή στους Αλγορίθμους

Κριτήρια αλγορίθμου

Οι αλγόριθμοι θα πρέπει να πληρούν κάποια πρότυπα και να διατυπώνονται με συγκεκριμένο τρόπο.

- **Περατότητα**
  - Κάθε εκτέλεση είναι πεπερασμένη, δηλαδή τελειώνει ύστερα από έναν πεπερασμένο αριθμό διεργασιών ή βημάτων.
  - Μία διαδικασία που δεν τελειώνει μετά από συγκεκριμένο/πεπερασμένο αριθμό βημάτων λέγεται απλά υπολογιστική διαδικασία.

Σύντομη Εισαγωγή στους Αλγορίθμους

Κριτήρια αλγορίθμου

Οι αλγόριθμοι θα πρέπει να πληρούν κάποια πρότυπα και να διατυπώνονται με συγκεκριμένο τρόπο.

- **Αποτελεσματικότητα**
  - Είναι μηχανιστικά αποτελεσματικός, δηλαδή όλες οι διαδικασίες που περιλαμβάνει μπορούν να πραγματοποιηθούν με ακρίβεια και σε πεπερασμένο χρόνο "με μολύβι και χαρτί".
  - Κάθε μεμονωμένη εντολή του αλγορίθμου να είναι απλή (και όχι σύνθετη). Δηλαδή μία εντολή δεν αρκεί να έχει ορισθεί αλλά πρέπει να είναι και εκτελέσιμη.

Σύντομη Εισαγωγή στους Αλγορίθμους

Κριτήρια αλγορίθμου

Οι αλγόριθμοι θα πρέπει να πληρούν κάποια πρότυπα και να διατυπώνονται με συγκεκριμένο τρόπο.

- **Είσοδος δεδομένων**
  - Κατά την εκκίνηση εκτέλεσης του αλγορίθμου καμία, μία ή περισσότερες τιμές δεδομένων πρέπει να δίνονται ως είσοδοι στον αλγόριθμο.
  - Η περίπτωση που δε δίνονται τιμές δεδομένων εμφανίζεται όταν ο αλγόριθμος δημιουργεί και επεξεργάζεται κάποιες πρωτογενείς τιμές με τη βοήθεια συναρτήσεων παραγωγής τυχαίων αριθμών ή με τη βοήθεια άλλων απλών εντολών.

Σύντομη Εισαγωγή στους Αλγορίθμους

Κριτήρια αλγορίθμου

Οι αλγόριθμοι θα πρέπει να πληρούν κάποια πρότυπα και να διατυπώνονται με συγκεκριμένο τρόπο.

- **Έξοδος αποτελεσμάτων**
  - Δίδει τουλάχιστον ένα μέγεθος ως αποτέλεσμα που εξαρτάται κατά κάποιο τρόπο από τις αρχικές εισόδους.
  - Ο αλγόριθμος πρέπει να δημιουργεί τουλάχιστον μία τιμή (δεδομένων) ως αποτέλεσμα προς το χρήστη ή προς ένα άλλο αλγόριθμο.

Σύντομη Εισαγωγή στους Αλγορίθμους

Περιγραφή και αναπαράσταση

Τέσσερις είναι οι βασικοί τρόποι αναπαράστασης ενός αλγορίθμου:

1. **Ελεύθερο κείμενο**, που αποτελεί τον πιο αδόμητο τρόπο παρουσίασης αλγορίθμου. Ελλοχεύει η δημιουργία μιας μη εκτελέσιμης κατάστασης παραβιάζοντας έτσι το κριτήριο της αποτελεσματικότητας.
2. **Διάγραμμα ροής**, που συνιστά έναν πιο γραφικό τρόπο παρουσίασης του αλγορίθμου. Η χρήση διαγραμμάτων ροής δεν είναι η πλέον ενδεδειγμένη λύση για ένα πρόβλημα και για αυτό χρησιμοποιούνται σπάνια στη βιβλιογραφία.
3. **Φυσική γλώσσα που εκτελείται κατά βήματα**. Σε αυτή την περίπτωση μπορεί να παραβιαστεί το κριτήριο του καθορισμού μεταξύ των βημάτων.
4. **Κωδικοποίηση του αλγορίθμου σε ψευδογλώσσα** ή **γλώσσα προγραμματισμού**. Έτσι ο αλγόριθμος παρουσιάζεται πιο συνοπτικός, συμπαγής ενώ πληροί και τις προϋποθέσεις του δομημένου προγραμματισμού.

Σύντομη Εισαγωγή στους Αλγορίθμους

Βασικές εντολές

- **Δομή ακολουθίας**
  - Η δόμηση των διαδικασιών σε τέτοια μορφή, έτσι ώστε οι διαδικασίες να εκτελούνται με τη σειρά από τον υπολογιστή.
- **Δομή επιλογής**
  - Η προγραμματιστική δομή που περικλείει τον έλεγχο μιας συνθήκης και μία ή δύο ομάδες εντολών. Από τις ομάδες των εντολών εκτελείται η πρώτη, αν ισχύει η συνθήκη, ή αν υπάρχει και δεύτερη ομάδα εντολών εκτελείται η δεύτερη αν δεν ισχύει η συνθήκη.
  - Με τη συνθήκη εννοούμε δυο όρους ίδιου τύπου και ανάμεσα τους ένας τελεστής σύγκρισης.
  - Με τον τελεστή σύγκρισης εννοούμε ένα από τα σύμβολα < , > , <> , <= , >= , =.
  - Το αποτέλεσμα της σύγκρισης των όρων (μεταβλητές ή σταθερές) είναι η αλγεβρική τιμή Αληθής (**True**) ή Ψευδής (**False**).

Σύντομη Εισαγωγή στους Αλγορίθμους

Βασικές εντολές

- **Δομή επανάληψης**
  - Η προγραμματιστική δομή που περικλείει τον συνεχή έλεγχο μίας συνθήκης και μία ομάδα εντολών. Οι εντολές εκτελούνται ανάλογα με την δομή της επανάληψης. Υπάρχουν τριών ειδών επαναλήψεις.
  - **Όσο ... επανάλαβε**. Σε αυτή την δομή επανάληψης ελέγχεται πρώτα η συνθήκη και εφόσον ισχύει (δίνει τιμή αληθή) εκτελείται η ομάδα εντολών.
  - **Επανάλαβε ... μέχρις ότου**. Σε αυτή την δομή επανάληψης εκτελείται η ομάδα εντολών, στη συνέχεια ελέγχεται αν ισχύει η συνθήκη και εφόσον **ΔΕΝ ισχύει** (δίνει τιμή ψευδής) εκτελείται ξανά η ομάδα εντολών.
  - **Για Ν φορές επανάλαβε.** Σε αυτή την δομή επανάληψης εκτελείται η ομάδα εντολών Ν φορές όπου Ν είναι αριθμός θετικός ακέραιος.

Σύντομη Εισαγωγή στους Αλγορίθμους

Βασικές εντολές

- **Δομή επανάληψης**
  - **Για Ν φορές επανάλαβε.** Σε αυτή την δομή επανάληψης εκτελείται η ομάδα εντολών Ν φορές όπου Ν είναι αριθμός θετικός ακέραιος.

    Δεσμεύσεις της εντολής *Για*:

    - αν η τιμή έναρξης της επανάληψης είναι μικρότερη ή το πολύ ίση με την τιμή τέλους τότε αναγκαστικά το βήμα μεταβολής πρέπει να είναι θετικό.
    - αν η τιμή έναρξης είναι μεγαλύτερη η το πολύ ίση με την τιμή τέλους τότε αναγκαστικά το βήμα μεταβολής πρέπει να είναι αρνητικό.
    - σε καμία περίπτωση δεν πρέπει το βήμα να είναι όσο με το μηδέν.

Σύντομη Εισαγωγή στους Αλγορίθμους

Εφαρμογή των αλγορίθμων

- Οι αλγόριθμοι μπορούν να υλοποιηθούν από προγράμματα ηλεκτρονικών υπολογιστών, μολονότι συχνά σε περιορισμένες μορφές. Ένα λάθος στον σχεδιασμό ενός αλγόριθμου για την λύση ενός προβλήματος μπορεί να οδηγήσει σε αποτυχίες/βλάβες στο εφαρμοσμένο πρόγραμμα.

- Οι αλγόριθμοι δεν υλοποιούνται μόνο ως προγράμματα υπολογιστών, αλλά συχνά επίσης και με άλλα μέσα, όπως π.χ. σε ένα βιολογικό νευρικό δίκτυο, ή σε ένα ηλεκτρονικό κύκλωμα, ή σε μια μηχανική συσκευή.

Σύντομη Εισαγωγή στους Αλγορίθμους

Εφαρμογή των αλγορίθμων

- Η ανάλυση και η μελέτη των αλγορίθμων είναι ένας τομέας τής επιστήμης της πληροφορικής, και ασκείται συχνά αφαιρετικά (χωρίς τη χρήση μιας συγκεκριμένης γλώσσας προγραμματισμού ή άλλη εφαρμογή).
  - Από αυτή την άποψη, μοιάζει με άλλους μαθηματικούς τομείς, συγκεκριμένα στο ότι η εστίαση της ανάλυσης είναι πάνω στις βασικές αρχές του αλγορίθμου, και όχι σε οποιαδήποτε ιδιαίτερη εφαρμογή του.
  - Ένας τρόπος απεικόνισης ένας αλγόριθμου είναι το γράψιμο του ψευδοκώδικα.
  - Άλλοι τρόποι είναι: με ελεύθερο κείμενο, με φυσική γλώσσα περιγράφοντας τα βήματα και με λογικό διάγραμμα

Σύντομη Εισαγωγή στους Αλγορίθμους

Εφαρμογή των αλγορίθμων

- Η ανάλυση και η μελέτη των αλγορίθμων είναι ένας τομέας τής επιστήμης της πληροφορικής, και ασκείται συχνά αφαιρετικά (χωρίς τη χρήση μιας συγκεκριμένης γλώσσας προγραμματισμού ή άλλη εφαρμογή).
  - Από αυτή την άποψη, μοιάζει με άλλους μαθηματικούς τομείς, συγκεκριμένα στο ότι η εστίαση της ανάλυσης είναι πάνω στις βασικές αρχές του αλγορίθμου, και όχι σε οποιαδήποτε ιδιαίτερη εφαρμογή του.
  - Ένας τρόπος απεικόνισης ένας αλγόριθμου είναι το γράψιμο του ψευδοκώδικα.
  - Άλλοι τρόποι είναι: με ελεύθερο κείμενο, με φυσική γλώσσα περιγράφοντας τα βήματα και με λογικό διάγραμμα

Μετάβαση προς την Κωδικοποίηση

Θεμελιώδεις ενότητες:

- Μεταβλητές & τύποι (int, float, str, bool).
- Είσοδος/Έξοδος (input, print).
- Έλεγχος ροής (if, for, while).
- Συναρτήσεις (def) & modularity.
- Εισαγωγή βιβλιοθηκών (import).

Μετάβαση προς την Κωδικοποίηση

Παραδείγματα:

- Άθροισμα/Μέσος όρος λίστας.
- Έλεγχος άρτιων/περιττών σε συλλογή.
- Γραμμική αναζήτηση (linear search).
- Υπολογισμός factorial (αναδρομή vs επανάληψη).

Μετάβαση προς την Κωδικοποίηση

Δομές Δεδομένων:

- Λίστες (lists): δημιουργία, επανάληψη, append/pop.
- Πλειάδες (tuples): αμεταβλητότητα.
- Λεξικά (dict): κλειδί–τιμή.
- Σύνολα (set): μοναδικά στοιχεία.

Μετάβαση προς την Κωδικοποίηση

Πρακτικές Αποσφαλμάτωσης και Δοκιμών:

- Χρήση print για γρήγορη διερεύνηση.
- Ανάγνωση stack traces & exceptions.
- assert και απλά unit tests.
- Καλή επιλογή test cases.

Τι είναι η Python

Η Python είναι…… μια γλώσσα προγραμματισμού υψηλού επιπέδου και γενικού σκοπού.

- **Υψηλού επιπέδου (high-level)**
  - Η Python είναι πιο αφηρημένη γλώσσα σε σχέση με τη C.
  - Αυτό προσθέτει εκφραστικότητα και δίνει δύναμη στους προγραμματιστές να γράφουν πιο ευέλικτο και αποτελεσματικό κώδικα.
  - Μειονέκτημα: σε ορισμένες περιπτώσεις μπορεί να είναι πιο αργή, αλλά υπάρχουν πολλές μέθοδοι βελτιστοποίησης.
- **Γενικού σκοπού (general-purpose)**
  - Δεν περιορίζεται σε συγκεκριμένο αντικείμενο (π.χ. η R επικεντρώνεται στατιστικά).
  - Δεν περιορίζεται σε συγκεκριμένο προγραμματιστικό παράδειγμα (π.χ. η Java επιβάλλει αντικειμενοστρεφή προσέγγιση), καθιστώντας τη ένα πανίσχυρο εργαλείο για πολλές εφαρμογές: σε επιστήμες, στην εκπαίδευση, σε ανάπτυξη εφαρμογών.
  - Ο προγραμματιστής βρίσκει άπειρα πακέτα για ειδικές ανάγκες.
  - Προωθεί τον «υπολογιστικό τρόπο σκέψης» των μαθητών.

Τι είναι η Python

Η Python προωθεί…… την απλότητα και την αναγνωσιμότητα του κώδικαα)

- **Δε χρειάζεται δήλωση τύπου μεταβλητών (δυναμική τυποποίηση).**
- **Δεν χρησιμοποιεί άγκιστρα {} για blocks, αλλά απλή εσοχή.**
- **Οφέλη:**
  - Εστίαση στην ουσία → ανάπτυξη αλγοριθμικής σκέψης χωρίς να μπερδεύουν συντακτικές λεπτομέρειες.
  - Αυξημένη παραγωγικότητα → απλός κώδικας = λιγότερος χρόνος, λιγότερο κόστος.

Τι είναι η Python

Παράδειγμα: Hello World

- **C**

```c
int main()
{
  printf("Hello World\n");
  return 0;
}
```

- **Java**

```java
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
     }
}
```

- **Python**

```python
print('Hello World!')
```

Αναγνωσιμότητα: Ο απλός κώδικας διαβάζεται εύκολα από πολλούς. Στην ανάπτυξη είναι σημαντικό, γιατί ο κώδικας γράφεται μία φορά αλλά διαβάζεται πολλές.

Τι είναι η Python

- Εγκαταστήστε τη γλώσσα και εξοικειωθείτε με το Integrated DeveLopment Environment (IDLE):
  - Κατεβάστε και εγκαταστήστε την πιο πρόσφατη έκδοση της Python για Windows (https://www.python.org/downloads/)
  - Διαβάστε τις οδηγίες του wiki της Python για διάφορα λειτουργικά συστήματα (https://wiki.python.org/moin/BeginnersGuide/Download)

- **Μοντέλο εκτέλεσης γλώσσας (Διερμηνέας Python)**: Η Python υλοποιείται κυρίως ως ερμηνευμένη γλώσσα, αλλά μέρος της διαδικασίας εκτέλεσης είναι στην πραγματικότητα η μεταγλώττιση κώδικα .
- Η ερμηνεία σημαίνει ότι ο κώδικας Python που γράφετε εισάγεται σε ένα πρόγραμμα διερμηνείας που εκτελεί τον κώδικα καθώς τον διαβάζει γραμμή προς γραμμή (συνήθως ο πηγαίος κώδικας Python αποθηκεύεται σε αρχεία .py)
- Αυτή η διαδικασία ερμηνείας διαφέρει σημαντικά από τη μεταγλώττιση (σε γλώσσες όπως η C ή η Java) όπου ο μεταγλωττιστής της γλώσσας μεταφράζει αρχικά τον κώδικα σε μεταγλωττισμένη μορφή (μια μορφή κώδικα πιο κοντά σε αυτό που «καταλαβαίνει» η μηχανή).

Τι είναι η Python

- Εγκαταστήστε τη γλώσσα και εξοικειωθείτε με το Integrated DeveLopment Environment (IDLE):
  - Κατεβάστε και εγκαταστήστε την πιο πρόσφατη έκδοση της Python για Windows (https://www.python.org/downloads/)
  - Διαβάστε τις οδηγίες του wiki της Python για διάφορα λειτουργικά συστήματα (https://wiki.python.org/moin/BeginnersGuide/Download)

- **Μοντέλο εκτέλεσης γλώσσας (Διερμηνέας Python)**: Η Python υλοποιείται κυρίως ως ερμηνευμένη γλώσσα, αλλά μέρος της διαδικασίας εκτέλεσης είναι στην πραγματικότητα η μεταγλώττιση κώδικα .

Πλεονεκτήματα/μειονεκτήματα της ερμηνευμένης υλοποίησης:

- Πλεονεκτήματα : η ερμηνεία προσφέρει μια πιο διαδραστική μαθησιακή εμπειρία στον αρχάριο, καθώς εκτελεί αμέσως τον κώδικα και παρέχει ανατροφοδότηση.
- Μειονεκτήματα : η ερμηνεία τείνει να είναι πιο αργή στην εκτέλεση κώδικα, καθώς ένας διερμηνέας διαβάζει (αναλύει), ερμηνεύει και εκτελεί τον πηγαίο κώδικα «από την αρχή» κάθε φορά (δεν δημιουργείται μεταγλωττισμένη μορφή κώδικα).

Τι είναι η Python

- Εγκαταστήστε τη γλώσσα και εξοικειωθείτε με το Integrated DeveLopment Environment (IDLE):
  - Κατεβάστε και εγκαταστήστε την πιο πρόσφατη έκδοση της Python για Windows (https://www.python.org/downloads/)
  - Διαβάστε τις οδηγίες του wiki της Python για διάφορα λειτουργικά συστήματα (https://wiki.python.org/moin/BeginnersGuide/Download)

- **Bytecode και Εικονική Μηχανή Python (PVM)**
  - Η Python, ωστόσο, δεν είναι μια γλώσσα που ερμηνεύεται απλά. Η εκτέλεση κώδικα είναι κάπως πιο περίπλοκη.
  - Στην ουσία, η Python δημιουργεί μια ενδιάμεση μεταφρασμένη μορφή κώδικα που ονομάζεται ' bytecode ' . Αυτός είναι κώδικας ανεξάρτητος από την πλατφόρμα (συνήθως σε αρχεία .pyc) ο οποίος εκτελείται από μια ειδική μορφή προγράμματος Python που ονομάζεται "Python Virtual Machine" (PVM).
  - Δεδομένου ότι η εκτέλεση bytecode είναι ταχύτερη από την εκτέλεση πηγαίου κώδικα από την αρχή, η Python τείνει να είναι ταχύτερη από την καθαρά ερμηνευμένη υλοποίηση.

Τι είναι η Python

- Εγκαταστήστε τη γλώσσα και εξοικειωθείτε με το Integrated DeveLopment Environment (IDLE):
  - Κατεβάστε και εγκαταστήστε την πιο πρόσφατη έκδοση της Python για Windows (https://www.python.org/downloads/)
  - Διαβάστε τις οδηγίες του wiki της Python για διάφορα λειτουργικά συστήματα (https://wiki.python.org/moin/BeginnersGuide/Download)

- **Bytecode και Εικονική Μηχανή Python (PVM)**

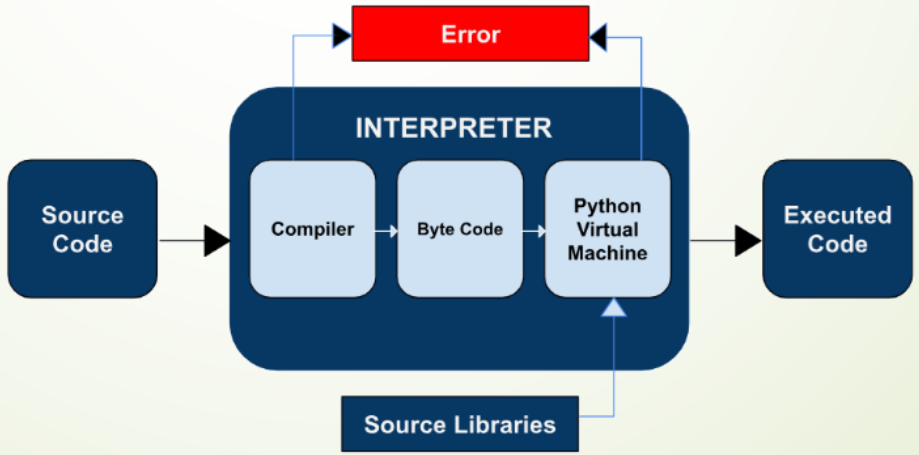
  Όταν εργάζεστε με Python, το πρόβλημα της πιο αργής εκτέλεσης του ερμηνευμένου κώδικα μετριάζεται από τουλάχιστον δύο παράγοντες:

  1. Το σύγχρονο υλικό είναι αρκετά γρήγορο για να εκτελέσει αποτελεσματικά bytecode Python σε πολλές περιπτώσεις.
  2. Σε κρίσιμες χρονικά καταστάσεις, υπάρχει πάντα η πιθανότητα «ανάμειξης» της Python με ταχύτερα εκτελούμενο μεταγλωττισμένο κώδικα C.

Τι είναι η Python

- Εγκαταστήστε τη γλώσσα και εξοικειωθείτε με το Integrated DeveLopment Environment (IDLE):
  - Κατεβάστε και εγκαταστήστε την πιο πρόσφατη έκδοση της Python για Windows (https://www.python.org/downloads/)
  - Διαβάστε τις οδηγίες του wiki της Python για διάφορα λειτουργικά συστήματα (https://wiki.python.org/moin/BeginnersGuide/Download)

- **Μοντέλο εκτέλεσης πυρήνα γλώσσας Python**



1. Εισαγωγικές Έννοιες

- **Μεταβλητές και δηλώσεις ανάθεσης**
  - **Στη γραμμή εντολών**

```
>>> 45 + 72
>>> 117
```

  - **Το "=" δεν είναι τελεστής**

```
>>> x = 7
>>> y = 3
>>> x + y
>>> 10
```

  - **Το "=" δεν είναι τελεστής - Είναι σύμβολο εκχώρησης (assignment symbol)**
  - **Τέλος γραμμής = τέλος εντολής (statement) /Multiline statements βλ. παρακάτω**

1. Εισαγωγικές Έννοιες

- **Ονόματα μεταβλητών**

>>> x = 7

“x”: όνομα μεταβλητής

- **Μπορεί να αποτελείται από**
  1. **Γράμματα**
  2. **Αριθμούς**
  3. **Κάτω παύλα (underscore: “_”)**
- **Δεν επιτρέπεται να ξεκινά με αριθμό**
- **Διάκριση πεζών-κεφαλαίων (case sensitive)**
- **number ≠ Number**

1. Εισαγωγικές Έννοιες

- **Τύποι δεδομένων / μεταβλητών**

Κάθε μεταβλητή έχει ένα τύπο (data type)

- >>> x = 7
  >>> type(x)
  <class 'int'>
- >>> type(7.2)
  <class 'float'>
- Text Type: str; Numeric Types: int, float, complex; Sequence Types: list, tuple, range; Mapping Type: dict; Set Types: set, frozenset; Boolean Type: bool; Binary Types: bytes, bytearray, memoryview

Δεν απαιτείται να

1. Προκαθορίσουμε τον τύπο μιας μεταβλητής
2. Δεσμεύουμε χώρο για μια μεταβλητή
3. Απλά γράφουμε το όνομα και κάνουμε εκχώρηση

1. Εισαγωγικές Έννοιες

- **Αριθμητικοί τελεστές (Arithmetic operators)**

Πράξη Python	Αριθμητικός τελεστής	Αλγεβρική παράσταση	Παράσταση Python	Παράδειγμα
Πρόσθεση	+	f + 7	f + 7	8 ← 3 + 5
Αφαίρεση	-	p – c	p - c	-2 ← 3 – 5
Πολλαπλασιασμός	*	b · m	b * m	15 ← 3 * 5
Ύψωση σε δύναμη	**	x^y	x ** y	243 ← 3^5
Πραγματική διαίρεση	/	$\frac{x}{y}$ ή x/y ή $x \div y$	x / y	1,75 ← 7 / 4
Ακέραια διαίρεση (floor)	//	$\left\lfloor \frac{x}{y} \right\rfloor$ ή $\lfloor x/y \rfloor$ ή $\lfloor x \div y \rfloor$	x // y	1 ← 7 // 4; -4 ← -13 // 4
Υπόλοιπο (modulo)	%	r mod s	r % s	0 ← 0%3 ; 1 ← 1%3 2 ← 2%3 ; 0 ← 3%3 1 ← 4%3

1. Εισαγωγικές Έννοιες

- **Εξαιρέσεις και ίχνη**

>>> 12/0

traceback (most recent call last):
file "<pyshell#14>", line 1, in <module> 12/0
zeroDivisionError: division by zero

- Διαίρεση με το μηδέν -> δεν ορίζεται -> εξαίρεση
- Αναφορά εξαίρεσης με ίχνος
  - Πού έγινε το σφάλμα file "<pyshell#14>", line 1, in <module> 12/0
  - Κατηγοριοποίηση σφάλματος zeroDivisionError
  - Περιγραφή division by zero

1. Εισαγωγικές Έννοιες

- **Ομαδοποίηση παραστάσεων με παρενθέσεις**
- Ομαδοποίηση από αριστερά προς δεξιά
  - a + b + c = (a + b) + c
  - Όλοι οι τελεστές με ίση προτεραιότητα
  - Εξαίρεση: τελεστής δύναμης
- Τύποι τελεστών
  - Τελεστέοι ακέραιοι -> αποτέλεσμα ακέραιο (εκτός της διαίρεσης)
  - Τελεστέοι δεκαδικοί -> αποτέλεσμα δεκαδικό
  - Μικτοί τελεστέοι -> αποτέλεσμα δεκαδικό
- Πλεονάζουσες παρενθέσεις
  - Δεν επηρεάζουν το αποτέλεσμα
  - Για ομαδοποίηση & ευκρίνεια & αναγνωσιμότητα

1. Εισαγωγικές Έννοιες

- **Η συνάρτηση print**
- Ενσωματωμένη συνάρτηση που προβάλει το/τα ορίσματά της ως γραμμή κειμένου στη συνήθη έξοδο (μονές ή διπλές απόστροφοι)

```
>>> print("Hello world")
Hello world
```

- Για περισσότερα του ενός, βάζει αυτόματα κενό χαρακτήρα ανάμεσά τους

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

Hello world
```

- Χαρακτήρας διαφυγής (escape character): \ (backslash)
  - \ + *αμέσως επόμενος χαρακτήρας: ακολουθία διαφυγής (escape sequence) , λ.χ.* \”

```
>>> print('World\'s Hello')
World's Hello
```

1. Εισαγωγικές Έννοιες

- **Μερικές ακολουθίες διαφυγής**

Ακολουθία διαφυγής	Περιγραφή
\n	Εισάγει χαρακτήρα νέας γραμμής. Κατά την προβολή οι χαρακτήρες μετά την ακολουθία ξεκινούν σε νέα γραμμή. (newline)
\t	Εισάγει οριζόντιο στηλοθέτη. Κατά την προβολή οι χαρακτήρες μετά την ακολουθία μετατοπίζονται στην επόμενη θέση στηλοθέτη (tab)
\\	Εισάγει ανάστροφη κάθετο σε συμβολοσειρά
\"	Εισάγει χαρακτήρα διπλής αποστρόφου σε συμβολοσειρά
\'	Εισάγει χαρακτήρα μονήςαποστρόφου σε συμβολοσειρά

1. Εισαγωγικές Έννοιες

- **Αγνόηση αλλαγής γραμμής μακράς συμβολοσειράς**

>>> print('Αυτή είναι μια μακρά συμβολοσειρά \ που χωρίζεται σε 2 γραμμές κατά την ανάθεσή της')
Αυτή είναι μια μακρά συμβολοσειρά που χωρίζεται σε 2 γραμμές κατά την ανάθεσή της

- **Concatenation & casting, όλα σε μια γραμμή**

>>> print('Sum is', 7 + 3)
Sum is 10

- **Τριπλές απόστροφοι**

3 συνεχόμενοι μονοί ή διπλοί απόστροφοι,
>>> '''a_string''' → 'a_string' και >>> """a_string""" → 'a_string'
Κατάλληλες για

- Συμβολοσειρές με πολλαπλές γραμμές κατά την απεικόνιση,
- Συμβολοσειρές που περιέχουν μονούς ή διπλούς αποστρόφους ως μέρος του string
- Συμβολοσειρές τεκμηρίωσης (docstrings)

>>> print("""Display "hi" and 'bye' in quotes""")
Display "hi" and 'bye' in quotes

1. Εισαγωγικές Έννοιες

- **Συμβολοσειρές τεκμηρίωσης**
- Προτείνεται στην αρχή κάθε συνάρτησης, αρχείου, ομάδα εντολών
- Όμοια με τα σχόλια
  """Αυτή είναι μια συνάρτηση ομοιότητας ακεραίων"""

Κενός χώρος

- Πλήρως κενές γραμμές ή/και διαστήματα (spaces) ή/και στηλοθέτες
- Κάνουν τον κώδικα ευανάγνωστο
- Εκτός περιπτώσεων η Python τις αγνοεί

1. Εισαγωγικές Έννοιες

- **Λήψη δεδομένων από τον χρήστη**

  - **Με τη συνάρτηση input**

```
>>> name = input("What's your name? ")
What's your name? Paul
>>> name
'Paul'
>>> print(name)
Paul
```

  - **Η input επιστρέφει πάντα συμβολοσειρά**

```
>>> v1 = input('Enter 1st number: ')
Enter 1st number: 2

>>> v2 = input('Enter 2nd number: ')
Enter 2nd number: 7

>>> v1 + v2
'27'

>>> int(v1) + int(v2)
9
```

1. Εισαγωγικές Έννοιες

Λήψη αποφάσεων

- Συνθήκη (condition): λογική (Boolean) παράσταση με τιμή TRUE / FALSE

```
>>> 7 < 4
False
>>> 7 > 4
True
```

- Τελεστές σύγκρισης
- Προτεραιότητες: Βάλτε παρενθέσεις!
- Αλυσιδωτές συγκρίσεις (μεταβλητή εντός εύρους;)

```
>>> 1 <= 2 <= 3
True
```

```
>>> x = 5
>>> 6 <= x <= 30
False
```

```
>>> 1 <= x <= 30
True
```

Αλγεβρικός τελεστής	Τελεστής Python	Παράδειγμα συνθήκης	Σημασία
>	>	x > y	x μεγαλύτερο του y
<	<	x < y	x μικρότερο του y
≥	>=	x >= y	x μεγαλύτερο ίσο του y
≤	<=	x <= y	x μικρότερο ή ίσο του y
=	==	x == y	x ίσο του y
≠	!=	x != y	x δεν είναι ίσο του y

1. Εισαγωγικές Έννοιες

- **Προτεραιότητα τελεστών**

Τελεστές	Ομαδοποίηση	Τύπος
()	Από αριστερά προς δεξιά	Παρενθέσεις
**	Από δεξιά προς αριστερά	Ύψωση σε δύναμη
* / // %	Από αριστερά προς δεξιά	Πολλαπλασιασμός, πραγματική διαίρεση, ακέραια διαίρεση, υπόλοιπο
+ –	Από αριστερά προς δεξιά	Πρόσθεση, αφαίρεση
> <= < >=	Από αριστερά προς δεξιά	Μικρότερο από, μικρότερο από ή ίσο μεγαλύτερο από, μεγαλύτερο από ή ίσο
== !=	Από αριστερά προς δεξιά	Ίσο, όχι ίσο