Μάθημα: Πληροφορική Ι

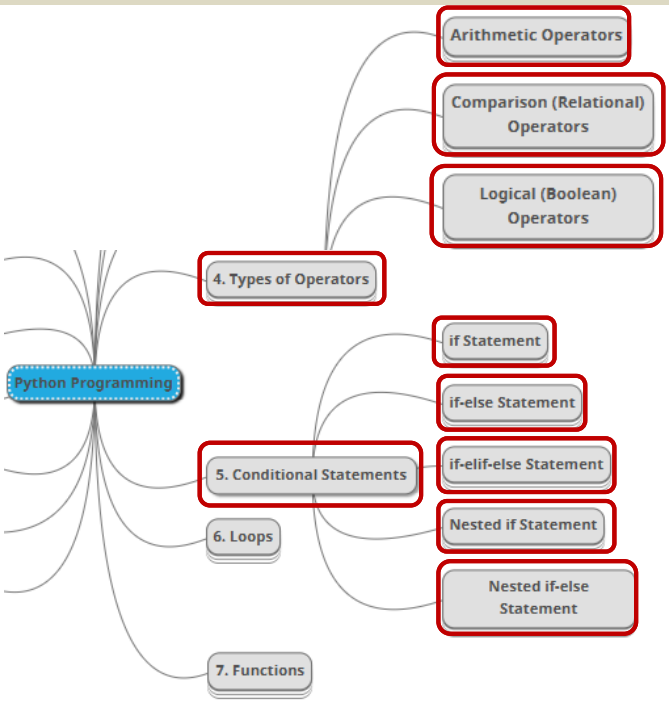
4ο Φύλλο Φροντιστηρίου

Τίτλος: Σύνθετες λογικές εκφράσεις και εμφωλευμένες επιλογές (and, or, not - nested if)

1. Στόχοι μαθήματος

- Κατανόηση της χρήσης πολλαπλών συνθηκών με τους λογικούς τελεστές and, or, not.
- Εξάσκηση σε σύνθετες λογικές εκφράσεις.
- Εισαγωγή στις εμφωλευμένες επιλογές (nested if).
- Εφαρμογή σε ρεαλιστικά επιχειρησιακά σενάρια λήψης αποφάσεων (έκπτωση, bonus, αξιολόγηση υπαλλήλου κλπ).
- Καλλιέργεια δεξιοτήτων ανάγνωσης και ερμηνείας πολύπλοκου κώδικα.

2. Mindmap



Python concept map link: https://360digitmg.com/mindmap/python-programming

3. Βήμα-βήμα στην Python με Colab "Cell by Cell"

Σύντομες Οδηγίες

Για να εκτελέσετε ένα κελί στο Colab: κάντε κλικ μέσα στο κελί και πατήστε Shift + Enter.
Προσοχή στις εσοχές. Ο κώδικας μέσα σε if πρέπει να έχει εσοχή 4 κενών.

Cell 1 (Markdown)

 # Σύνθετες λογικές εκφράσεις

Στην Python μπορούμε να συνδυάσουμε περισσότερες από μία συνθήκες χρησιμοποιώντας:

- `and`: και
- `or`: ή
- `not`: όχι (αντίστροφη συνθήκη)

Cell 2 (Code)

```
# Παραδείγματα συνδυασμού συνθηκών

age = 25

income = 1200

print(age > 18 and income > 1000)   # True

print(age > 30 or income > 1000)    # True

print(not(age > 30))                # True


# Το αποτέλεσμα μιας σύγκρισης είναι `True` ή `False`.
```

Cell 3 (Markdown)

 # Παράδειγμα 1: Έλεγχος επιλεξιμότητας πελάτη για δάνειο

Κριτήρια:

- Ηλικία πάνω από 21 έτη
- Μηνιαίο εισόδημα πάνω από 1000 €
- Καμία εκκρεμότητα

Cell 4 (Code)

```
# Έλεγχος επιλεξιμότητας για δάνειο

age = int(input("Δώσε ηλικία: "))

income = float(input("Δώσε μηνιαίο εισόδημα (€): "))

pending_debt = input("Υπάρχουν εκκρεμότητες; (ν/ο): ")

if age > 21 and income > 1000 and pending_debt == "ο":

    print("Επιλέξιμος για δάνειο")

else:

    print("Μη επιλέξιμος για δάνειο")
```

Cell 5 (Markdown)

 # Παράδειγμα 2: Πολιτική έκπτωσης με πολλαπλά κριτήρια

Έκπτωση εφαρμόζεται όταν:

- το ποσό παραγγελίας > 100 €
- και ο πελάτης είναι “VIP”

Cell 6 (Code)

```
# Υπολογισμός έκπτωσης με δύο κριτήρια


order_value = float(input("Αξία παραγγελίας (€): "))
customer_type = input("Κατηγορία πελάτη (Regular/VIP): ")

if order_value > 100 and customer_type == "VIP":
    print("Δικαιούται έκπτωση 15%.")
else:
    print("Δεν δικαιούται έκπτωση.")
```

Cell 7 (Markdown)

 # Άσκηση 1

Προσθέστε μία ακόμα συνθήκη:

Αν η παραγγελία είναι πάνω από 200 € ή ο πελάτης είναι “VIP”, τότε να εκτυπώνεται “Δικαιούται έκπτωση 20%”.

Cell 8 (Code)

```
# Γράψτε εδώ τον κώδικά σας
```

Cell 9 (Markdown)

 # Η εμφωλευμένη δομή if (nested if)

Μπορούμε να έχουμε μια δομή if μέσα σε άλλη if για πιο σύνθετες αποφάσεις.

Cell 10 (Code)

 # Παράδειγμα εμφωλευμένης δομής

```
age = int(input("Δώσε ηλικία: "))
experience = int(input("Έτη προϋπηρεσίας: "))

if age >= 25:
    if experience >= 5:
        print("Επιλέξιμος για προαγωγή.")
    else:
        print("Χρειάζεται περισσότερη εμπειρία.")
else:
    print("Μη επιλέξιμος λόγω ηλικίας.")
```

Cell 11 (Markdown)

 # Επιχειρησιακό παράδειγμα 3: Αξιολόγηση υπαλλήλου

Κανόνες:

- Αν η επίδοση είναι ≥ 90 και η εμπειρία ≥ 5 έτη: “Εξαιρετικός”
- Αν η επίδοση ≥ 75 και εμπειρία ≥ 3 έτη: “Καλός”
- Διαφορετικά: “Χρειάζεται βελτίωση”

Cell 12 (Code)

```
# Αξιολόγηση υπαλλήλου με εμφωλευμένα if

performance = float(input("Δώσε ποσοστό επίδοσης (%): "))

experience = int(input("Δώσε έτη εμπειρίας: "))



if performance >= 90:
    if experience >= 5:
        status = "Εξαιρετικός"
    else:
        status = "Καλός"
elif performance >= 75:
    if experience >= 3:
        status = "Καλός"
    else:
        status = "Χρειάζεται βελτίωση"
else:
    status = "Χρειάζεται βελτίωση"
print("Αξιολόγηση υπαλλήλου:", status)
```

Cell 13 (Markdown)

 # Άσκηση 2

Επεκτείνετε το παραπάνω πρόγραμμα ώστε:

- Αν ο υπάλληλος είναι “Εξαιρετικός”, να λαμβάνει bonus 500 €.
- Αν είναι “Καλός”, bonus 200 €.
- Αν “Χρειάζεται βελτίωση”, κανένα bonus.

Cell 14 (Code)

```
# Γράψτε εδώ τον κώδικά σας
```

Cell 15 (Markdown)

 # Επιχειρησιακό παράδειγμα 4: Έλεγχος παροχών προσωπικού

Κανόνες:

- Αν ο εργαζόμενος είναι πλήρους απασχόλησης και έχει εμπειρία ≥ 2 έτη: **δικαιούται επίδομα.**
- Αν εργάζεται μερικώς αλλά έχει επίδοση ≥ 85%: **δικαιούται μερικό επίδομα.**
- Διαφορετικά: **δεν δικαιούται επίδομα.**

Cell 16 (Code)

```
# Έλεγχος επιλεξιμότητας για επίδομα

employment_type = input("Πλήρους ή Μερικής απασχόλησης; (Π/Μ): ")
experience = int(input("Έτη εμπειρίας: "))
performance = float(input("Επίδοση (%): "))


if employment_type == "Π" and experience >= 2:
    print("Δικαιούται πλήρες επίδομα.")
elif employment_type == "Μ" and performance >= 85:
    print("Δικαιούται μερικό επίδομα.")
else:
    print("Δεν δικαιούται επίδομα.")
```

Cell 17 (Markdown)

 # Συνήθη λάθη

- Λάθος χρήση του `and` και `or` (οι συνθήκες πρέπει να έχουν νόημα ξεχωριστά).
- Παραλείψεις εσοχών σε εμφωλευμένα `if`.
- Μη σωστή σειρά ελέγχου (οι πιο περιοριστικές συνθήκες πρέπει να ελέγχονται πρώτες).

4. Ανακεφαλαίωση μαθήματος

- Συνδυασμός συνθηκών με and, or, not.
- Εμφωλευμένες δομές if για πιο σύνθετες αποφάσεις.
- Εφαρμογή σε επιχειρησιακές περιπτώσεις (έκπτωση, δάνειο, bonus, αξιολόγηση).
- Ερμηνεία και ανάλυση της ροής εκτέλεσης σύνθετων συνθηκών.

5. Πηγές

- ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ ΜΕ ΡΥΤΗΟΝ - ΑΓΓΕΛΙΔΑΚΗΣ

- ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ ΜΕ ΑΡΩΓΗ ΤΗΣ ΓΛΩΣΣΑΣ ΡΥΤΗΟΝ - ΜΑΝΗΣ

- Google Colab Tutorial For Beginners

- Google Colab Tutorial for Beginners | Get Started with Google Colab

Οι Διδάσκοντες

Δρ. Πάνος Φιτσιλής, Καθηγητής (fitsilis@uth.gr)
Δρ. Χαράλαμπος Α. Δερβένης, ΕΤΕΠ (dervenisch@uth.gr)