

Η λίστα του Σουπερμάρκετ



Τι θα μάθουμε;

Θα δούμε πώς μπορούμε να φτιάχνουμε μια μεταβλητή με παραπάνω από μία τιμές. Οι μεταβλητές αυτού του είδους ονομάζονται «λίστες».



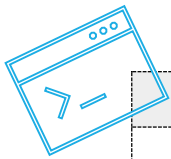
Τι πρέπει να θυμόμαστε;

Θα πρέπει να θυμόμαστε πώς φτιάχνουμε μεταβλητές, πώς συγκρίνουμε μεταβλητές, και πότε τα αποτελέσματα μιας σύγκρισης είναι True και πότε False.



Τι θα μπορούμε να κάνουμε μετά;

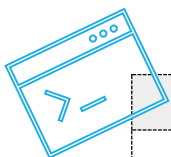
Θα μπορούμε να βάζουμε πολλές τιμές στις «μεταβλητές» μας, έτσι ώστε να μη χρειάζεται να φτιάχνουμε ξεχωριστές μεταβλητές για να αποθηκεύουμε παρόμοιες πληροφορίες.



| Κώδικας | Αποτέλεσμα |
|--|--|
| <pre>pswnia = ["milk", "chips", "juice", "tea"] print(pswnia) print(pswnia[1]) print(pswnia[3])</pre> | <pre>["milk", "chips", "juice", "tea"] chips tea</pre> |

1

1. Παραπάνω φτιάξαμε μια λίστα! Παρατηρήστε τον κώδικα! Μέσα σε τι βάλαμε τις τιμές της λίστας; Πώς διαλέγουμε μόνο μια τιμή από τη λίστα; Πώς θα δείξουμε το «milk»;



| Κώδικας | Αποτέλεσμα |
|--|--|
| <pre>pswnia = ["milk", "chips", "juice", "tea"] print("milk" in pswnia) print(not "milk" in pswnia) print("tea" in pswnia and "chips" in pswnia) print("juice" in pswnia or "coffee" in pswnia) print("coffee" in pswnia) print(not "coffee" in pswnia)</pre> | <pre>True False True True False True</pre> |

2

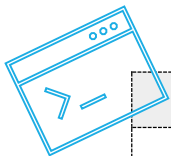
2. Ποιες δύο νέες λέξεις παρατηρείτε στον παραπάνω κώδικα; Τι «έλεγχο» νομίζετε ότι κάνει η καθεμία; Γιατί είναι False τα δύο παραπάνω αποτελέσματα;



Tip: Προσέξτε ότι το **in** και το **not**, εκτός από τις λίστες, μπορούμε να τα χρησιμοποιήσουμε σε οποιοδήποτε κομμάτι του κώδικά μας π.χ. «**if not age >= 18 :**» ή «**if x in pswnia :**».

3

3. Θα συνεχίσουμε μαζί το παρακάτω πρόγραμμα ώστε να ζητάει να γράψουμε ένα προϊόν και να ελέγχει αν αυτό υπάρχει στο Σουπερμάρκετ. Αν υπάρχει θα εμφανίζεται το μήνυμα «Βρέθηκε» αλλιώς το μήνυμα «Δε βρέθηκε».

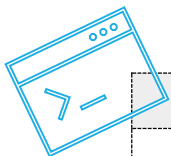


| Κώδικας | Αποτέλεσμα |
|---|------------|
| <pre>proionta = ["milk", "chips", "juice", "tea"]</pre> | |

4

4. Στον παρακάτω κώδικα υπάρχουν δύο λίστες προϊόντων, προϊόντα με έκπτωση και χωρίς. **Συνεργαστείτε με την ομάδα σας** και φτιάξτε ένα πρόγραμμα, χωρίς το else, που:

- να ζητάει να γράψουμε ένα προϊόν
- να ελέγχει αν υπάρχει στις λίστες
- αν έχει έκπτωση να μας εμφανίζει το μήνυμα «Έχει έκπτωση»
- αν δεν έχει έκπτωση να μας εμφανίζει το μήνυμα «Χωρίς έκπτωση»
- αν το προϊόν δεν υπάρχει καθόλου να εμφανίζεται το μήνυμα «Δε βρέθηκε»



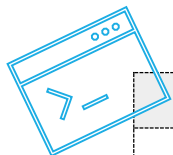
| Κώδικας | Αποτέλεσμα |
|--|------------|
| <pre>ekptwsi = ["milk", "tea"] xwris = ["juice", "coffee"]</pre> | |

5

5. Θα φτιάξετε **μόνοι σας** ένα πρόγραμμα που θα έχει δύο λίστες.

Στη μία λίστα βάλτε τους μονούς αριθμούς από το 0 ως το 10 και στη δεύτερη τους ζυγούς αριθμούς από το 0 ως το 10.

Το πρόγραμμά σας να ζητάει συνέχεια να γράφουμε έναν αριθμό από το 0 ως το 10 και να εμφανίζει το μήνυμα «Είναι μονός» ή «Είναι ζυγός».



| Κώδικας | Αποτέλεσμα |
|---------|------------|
| | |



Tip: Θα χρειαστεί να χρησιμοποιήσετε το **if**, το **while** και το **in**. Ίσως και το **not**. Θυμηθείτε το 0 είναι ζυγός αριθμός!

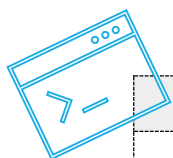
6

6. Το παραπάνω πρόγραμμα είναι εύκολο όταν έχουμε λίγους αριθμούς για τη λίστα, π.χ. δέκα. Τι γίνεται όμως αν θέλουμε να ελέγξουμε αν ένας αριθμός είναι μονός ή ζυγός από το 0 έως το 1.000.000; Δεν μπορούμε να τους γράψουμε όλους έναν-έναν σε λίστες!

Φτιάξετε μόνοι σας ένα πρόγραμμα που να ζητάει συνέχεια να εισάγουμε έναν αριθμό από το 0 ως το 1.000 και να μας δείχνει αν αυτός είναι μονός ή ζυγός!



Tip: Μη χρησιμοποιήσετε λίστες! Θυμηθείτε ότι ζυγοί είναι όλοι οι αριθμοί που διαιρούνται ακριβώς με το 2. Ακριβώς σημαίνει ότι η διαίρεση έχει υπόλοιπο 0. Στην Python βρίσκουμε το πηλίκο μιας διαίρεσης με το / και το υπόλοιπο μιας διαίρεσης με το %



| Κώδικας | Αποτέλεσμα |
|---------|------------|
| | |