

ΓΕΛ Βόνιτσας - Πληροφορική Β' Λυκείου – Σχολικό Έτος 2025- 2026.

Σημειώσεις στην γλώσσα προγραμματισμού Python.

Στις επιταγές των καιρών και για την προετοιμασία των μαθητών στον προγραμματισμό με την πιο δημοφιλή και την πιο συχνά χρησιμοποιούμενη για την τεχνητή νοημοσύνη (Α.Ι), ασχολούμαστε με την python. Θα δούμε αρχικά τα κοινά σημεία που έχει με τη ψευδογλώσσα και στη συνέχεια θα επεκταθούμε σε επιπλέον δυνατότητες που μας παρέχει αυτή η γλώσσα.

Σύγκριση με τη ψευδογλώσσα

Μεταβλητές – Εκχώρηση μεταβλητών - Η εντολή print ()

Η Python περιέχει όπως και η ψευδογλώσσα, επίσης τους γνωστούς τύπους μεταβλητών της ψευδογλώσσας χωρίς να χρειάζεται να τους δηλώσει. Ακολουθεί ένα παράδειγμα εκχώρησης **ακέραιας**, **πραγματικής** τιμής και **συμβολοσειράς** σε τρεις μεταβλητές μεταβλητές καθώς και η εκτύπωση των τιμών αυτών.

Η εντολή print χρησιμοποιείται για την εκτύπωση των τιμών των μεταβλητών, έχει μορφή συνάρτησης, για αυτό και χρησιμοποιούνται παρενθέσεις. Η εκχώρηση γίνεται με το σύμβολο του ίσον =

```
1 akeraios = 5
2 pragmatikos = 2.3
3 leksi = "vonitsa"
4 leksi2 = 'vonitsa'
5 print (akeraios , pragmatikos , leksi , leksi2)
6
```

Αντίστοιχα το παράδειγμα σε ψευδογλώσσα. Παρατηρήστε το υποχρεωτικό κομμάτι της δήλωσης του ονόματος του αλγορίθμου, το σύμβολο του βέλους για την εκχώρηση και την εντολή γράψε για την εμφάνιση των τιμών.

```
1 Αλγόριθμος παραδειγμα1
2 ακεραιος ← 5
3 πραγματικος ← 2.3
4 λεξη ← 'Vonitsa'
5 Γράψε ακεραιος, πραγματικος, λεξη
6 Τέλος παραδειγμα1
```

Εισαγωγή τιμών από το πληκτρολόγιο – Η εντολή input () και οι συναρτήσεις μετατροπής τύπου int () και float ()

Στην python χρησιμοποιείται η εντολή input η οποία διαβάζει την είσοδο από το πληκτρολόγιο (πάντα με τη μορφή χαρακτήρων συμβολοσειράς). Η μετατροπή των χαρακτήρων σε αριθμό γίνεται με μετατροπή σε ακέραιο ή σε πραγματικό. Ακολουθεί παράδειγμα εισαγωγής τιμών από το πληκτρολόγιο.

```

1 ακεραιος = int (input ("δώσε μου έναν ακέραιο"))
2 πραγματικος = float ( input ("δώσε μου έναν πραγματικό" ) )
3 λεξη = input ("δώσε μου μια λέξη")
4 print (ακεραιος, πραγματικος, λεξη)

```

```

δώσε μου έναν ακέραιο13
δώσε μου έναν πραγματικό5.7
δώσε μου μια λέξηbonitsa
13 5.7 bonitsa

```

Η εντολή **input ()** ενσωματώνει μήνυμα , επιστρέφει όπως αναφέραμε χαρακτήρες και για αυτό το λόγο χρειάζεται να μετατρέψουμε τους χαρακτήρες σε ακέραιο ή πραγματικό αριθμό με τη συνάρτηση **int ()** ή με τη συνάρτηση **float ()** .

Το αντίστοιχο παράδειγμα σε ψευδογλώσσα απαιτεί περισσότερες γραμμές κώδικα, πρακτικά τις διπλάσιες. Ήδη αρχίζει και αποδεικνύεται η απλότητα της python.

```

Αλγόριθμος παραδειγμα2
Γράψε "δώσε μου έναν ακέραιο"
Διάβασε ακεραιος
Γράψε "δώσε μου έναν πραγματικό "
Διάβασε πραγματικος
Γράψε "δώσε μου μια λέξη"
Διάβασε λεξη
Γράψε ακεραιος, πραγματικος, λεξη
Τέλος παραδειγμα2

```

Συγκριτικός πίνακας ψευδογλώσσας και python

Λειτουργία	Ψευδογλώσσα	Python
Εκχώρηση	<code>a <- 5</code>	<code>a = 5</code>
Έξοδος	Εμφάνισε a	<code>print(a)</code>
Είσοδος (Κείμενο)	Διάβασε όνομα	<code>onoma = input("Δώσε όνομα: ")</code>
Είσοδος (Αριθμός)	Διάβασε ηλικία	<code>hlikia = int(input("Δώσε ηλικία: "))</code>
Πράξεις	<code>ποσό <- τιμή * 1.24</code>	<code>poso = timi * 1.24</code>

Η δομή επιλογής – Πως παίρνω αποφάσεις στην python – Η εντολή if και η δήλωση των block εντολών με την :

Στην python χρησιμοποιείται η εντολή **if** (σε συνδυασμό με την **else** και την **elif** αν χρειάζεται). Επίσης χρησιμοποιείται το σύμβολο της άνω -κάτω τελείας **:** που δηλώνει την αρχή στην ομάδα- block των εντολών που εκτελούνται αν η συνθήκη είναι αληθής.

Η python δεν έχει εντολή τέλους της δομής επιλογής **AN** και **ΤΕΛΟΣ_ΑΝ** όπως η ψευδογλώσσα. Αυτόματα, προσθέτει 4 χαρακτήρες κενού για να δημιουργήσει με οπτικό τρόπο την ομάδα εντολών. Είναι επίσης ευαίσθητη στην αλλαγή των κενών, οπότε δεν

πρέπει να αλλάζουμε αυτή την τακτική της πρόσθεσης κενών στα προγράμματα που δημιουργούμε.

Ακολουθεί ένα παράδειγμα προγράμματος το οποίο μας ρωτάει το όνομα μας, το βαθμό στο μάθημα της πληροφορικής και μας απαντάει ανάλογα αν είμαστε πάνω ή κάτω από τη βάση 9.5 της εικοσαβάθμιας κλίμακας.

```
1 ονομα = input ("δώσε μου το ονομα του μαθητή ")
2 βαθμος = float (input ("δώσε μου το βαθμό στην πληροφορική ") )
3
4 if βαθμος >=9.5:
5     print (" ο μαθητής " , ονομα, ' έχει περάσει το μάθημα')
6 else:
7     print (" ο μαθητής " , ονομα, ' δεν έχει περάσει το μάθημα')
8 print (" τέλος προγράμματος")
```

Το αντίστοιχο πρόγραμμα στη ψευδογλώσσα δεν χρειάζεται να ακολουθήσει τους κανόνες της εμφώλευσης του κώδικα και μπορεί να καταλήξει να είναι λιγότερο κατανοητός οπτικά. Πάλι η Python κάνει την ίδια εργασία με σχεδόν το μισό αριθμό εντολών.

```
1 Αλγόριθμος παραδειγμα3
2 Γράψε "δώσε μου το ονομα του μαθητη"
3 Διάβασε ονομα
4 Γράψε "δώσε μου το βαθμό στην πληροφορικη "
5 Διάβασε βαθμος
6 Αν βαθμος ≥ 9.5 τότε
7   Γράψε "ο μαθητης με ονομα " , ονομα, " περασε το μαθημα"
8 αλλιώς
9   Γράψε "ο μαθητης με ονομα " , ονομα, " δεν περασε το μαθημα"
10 Τέλος_αν
11 Τέλος παραδειγμα3
```

Η δομή επανάληψης στην python – Η εντολή for και η συνάρτηση range()

Για τη δομή επανάληψης στην Python χρησιμοποιείται η εντολή for με μια διαφορά από τη ψευδογλώσσα και την αντίστοιχη εντολή **ΓΙΑ**.

Η εντολή for οποία μετράει με τη βοήθεια μιας μεταβλητής σε πεδία ορισμού όπως και στα μαθηματικά. Το πεδίο ορισμού ορίζεται από τη συνάρτηση **range** (που στα αγγλικά μεταφράζεται και σε πεδίο ορισμού).

Η αρίθμηση ξεκινά από τον αριθμό 0, και το πεδίο ορισμού είναι κλειστό και όχι ανοικτό όπως ονομάζουμε στα μαθηματικά. Χρειάζεται λίγος χρόνος για να το συνηθίσουμε πρακτικά αυτό το γεγονός.

```
for i in range(6):
    print (i)
```

```
0
1
2
3
4
5
```

Το αντίστοιχο παράδειγμα στη ψευδογλώσσα είναι πιο εύκολο στην κατανόηση

```
1 Αλγόριθμος παραδειγμα4
2 Για i από 0 μέχρι 5
3   Γράψε i
4 Τέλος_επανάληψης
5 Τέλος παραδειγμα4
```

Υπάρχει προφανώς και άλλος τρόπος επανάληψης στην python με αρχή και τέλος, όπως το ακόλουθο παράδειγμα που μετράει από το 1 μέχρι το 5.

```
for i in range(1,6):
    print (i)
```

Πίνακες -λίστες στην python.

Η python δεν έχει πίνακες αλλά λίστες. Η λίστα μοιάζει σε λειτουργία όμως με τον πίνακα, με κάποιες διαφορές:

Ξεκινάμε συνήθως με μία κενή λίστα

Τα στοιχεία της λίστας τα προσθέτουμε στο τέλος της

Στην οποία προσθέτουμε βήμα- βήμα όσα στοιχεία θέλουμε, παρά να δηλώνουμε ένα συγκεκριμένο μέγεθος πίνακα.

Δήλωση μίας λίστας

Η λίστα έχει και αυτή ένα όνομα, όπως οι μεταβλητές και χρησιμοποιούμε τις τετραγωνισμένες αγκύλες για να ορίσουμε τα στοιχεία της χωρισμένα με το κόμμα.

```
αριθμοι = [ 1,2,3,4,5,6]
ονοματα = [ "γιωργος" , "μαρια", "ελενη", "αλεξανδρος"]
```

Εισαγωγή στοιχείων σε λίστα

Η εισαγωγή των στοιχείων σε μία ήδη υπάρχουσα λίστα γίνεται με την εφαρμογή της συνάρτησης append (που σημαίνει επέκταση-πρόσθεση) στο όνομα της λίστας.

```

1 αριθμοι = [ 1,2,3,4,5,6]
2 ονοματα = [ "γιωργος" , "μαρια", "ελενη", "αλεξανδρος"]
3 print (αριθμοι)
4 print (ονοματα)
5 αριθμοι.append (7)
6 ονοματα.append("νικος")
7 print (αριθμοι)
8 print (ονοματα)

```

```

[1, 2, 3, 4, 5, 6]
['γιωργος', 'μαρια', 'ελενη', 'αλεξανδρος']
[1, 2, 3, 4, 5, 6, 7]
['γιωργος', 'μαρια', 'ελενη', 'αλεξανδρος', 'νικος']

```

Εμφάνιση στοιχείων λίστας

Η εκτύπωση των επιμέρους στοιχείων μιας λίστας γίνεται με τη χρήση των αγκυλών (η αρίθμηση ξεκινά από το 0).

```

1 ονοματα = [ "γιωργος" , "μαρια", "ελενη", "αλεξανδρος"]
2 print (ονοματα[0])
3 print (ονοματα[1])

```

```

γιωργος
μαρια

```

Μήκος της λίστας

Το μήκος της λίστας δίνεται από τη συνάρτηση len () που σημαίνει length (Μήκος)

```

ονοματα = [ "γιωργος" , "μαρια", "ελενη", "αλεξανδρος"]
a = len(ονοματα)
print ("μηκος", a)

```

```

μηκος 4

```

Εμφάνιση όλων των στοιχείων της λίστας

```

ονοματα = [ "γιωργος" , "μαρια", "ελενη", "αλεξανδρος"]
for i in ονοματα :
    print (i)

```

```

γιωργος
μαρια
ελενη
αλεξανδρος

```

Αφαίρεση στοιχείων από τη λίστα γίνεται με την εντολή pop() που εφαρμόζεται πάνω στη λίστα και αφαιρεί το τελευταίο στοιχείο.

```
1 ονοματα = [ "γιωργος", "μαρια", "ελενη", "αλεξανδρος" ]
2 ονοματα.pop()
3 ονοματα.pop()
4 ονοματα.pop()
5 for i in ονοματα:
6     print (i)
```

```
γιωργος
```

Αφαίρεση γίνεται και σε ενδιάμεσα στοιχεία της λίστα με την εντολή remove() η οποία όπως χρειάζεται να πάρει όρισμα το συγκεκριμένο στοιχείο της λίστας. Αν το στοιχείο της λίστας δεν υπάρχει, τότε πρόγραμμα σταματά την εκτέλεση του αντικανονικά.

```
1 ονοματα = [ "γιωργος", "μαρια", "ελενη", "αλεξανδρος" ]
2 ονοματα.remove("μαρια")
3
4 for i in ονοματα:
5     print (i)
```

```
γιωργος
ελενη
αλεξανδρος
```