

3. “Είναι πολλές οι εντολές”

```
10 | print ("Who in the world am I?")  
20 | print ("Ah, that's the great puzzle!")  
30 | #Alice in Wonderland
```

Πρόγραμμα...

Οι πύθωνες είναι αρκετά έξυπνα ζώα, όπως έχουμε αναφέρει και στα προηγούμενα κεφάλαια! Μέχρι τώρα, εργαστήκαμε με μία εντολή κάθε φορά. Οι πύθωνες, όμως, μπορούν να εργαστούν με πολλές εντολές. Με αυτό τον τρόπο μπορούμε να δημιουργήσουμε πολύπλοκα προγράμματα, τα οποία όμως εκτελούν πολλές και διαφορετικές εργασίες.

Στο κεφάλαιο αυτό θα γνωρίσουμε νέες εντολές, τις οποίες και θα συνδυάσουμε για να δημιουργήσουμε το πρώτο μας πρόγραμμα!



Τι θα γνωρίσουμε:

- Στο **Κεφάλαιο 3 “Είναι πολλές οι εντολές”** θα εργαστούμε με κώδικα πολλών γραμμών.
- Θα χρησιμοποιήσουμε την `print()` για να εμφανίσουμε στην οθόνη, ταυτόχρονα, πολλές γραμμές με πληροφορίες.
- Με την εντολή `input()` θα δώσουμε από το πληκτρολόγιο τιμή σε μια μεταβλητή.
- Με την `if...else` θα ελέγχουμε αν ισχύει μία συνθήκη (π.χ. $10 > 5$) ώστε να “αποφασίσει” το πρόγραμμά μας για την πορεία ενός απλού παιχνιδιού.
- Με την `type()` θα προβάλουμε τον τύπο μιας μεταβλητής.
- Με την βοήθεια της `int(input())` θα εισαγάγουμε δεδομένα από το πληκτρολόγιο ως ακέραιο αριθμό.




```
IDLE Shell 3.11.4
Python 3.11.4 (v3.11.4:d2340ef257, Jun 6 2023, 19:15:51) [Clang 13.0.0 (c
lang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Το 5 στη δεύτερη δύναμη ισούται με",5**2)
Το 5 στη δεύτερη δύναμη ισούται με 25
>>> |
```

```
untitled
```

Μέχρι τώρα, γράφαμε και εκτελούσαμε μία εντολή κάθε φορά. Αυτό είναι χρήσιμο, όμως δε μας βοηθά όταν θέλουμε να δημιουργήσουμε μια σύνθετη εφαρμογή.

Στο πιο πάνω παράδειγμα, υπολογίσαμε το τετράγωνο του 5 (ή, το 5 στη δεύτερη δύναμη).

Για τη δημιουργία ενός προγράμματος με αρκετές γραμμές κώδικα, χρησιμοποιούμε λογισμικά επεξεργασίας κώδικα (Editors). Το IDLE περιλαμβάνει τον δικό του Editor. Από το μενού File του IDLE, επιλέγουμε New File (εικόνα πάνω δεξιά).

Παρατηρούμε αμέσως τις διαφορές ανάμεσα στο νέο παράθυρο, στο οποίο θα γράψουμε μια σειρά εντολών (πάνω δεξιά) με το παράθυρο του IDLE με το οποίο εργαστήκαμε στα προηγούμενα κεφάλαια.

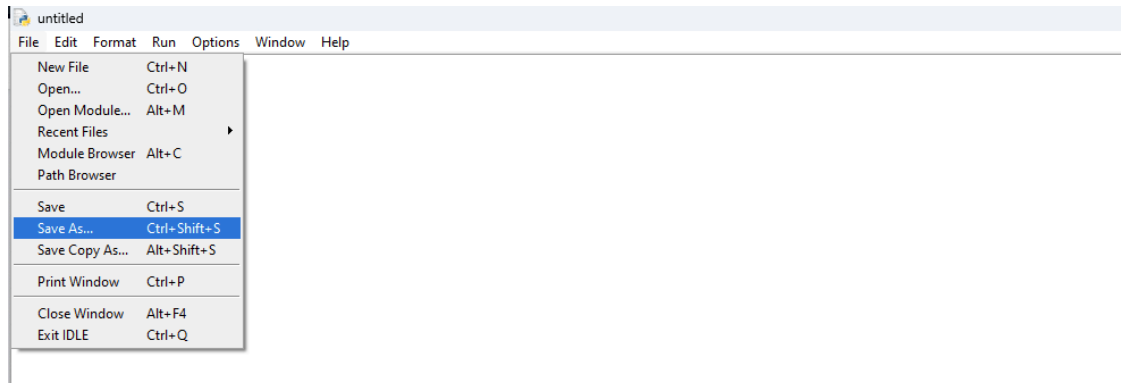
Στην επόμενη σελίδα θα δημιουργήσουμε το πρόγραμμα μας και θα γνωρίσουμε και νέες εντολές.



Για να μπορέσουμε να γράψουμε ένα πρόγραμμα (μία σειρά από εντολές) στην Python, μπορούμε να χρησιμοποιήσουμε ακόμη και απλές εφαρμογές συγγραφής κειμένου! Η εφαρμογή Notepad στα Windows ή το SimpleText σε MacOS είναι αρκετές για να γράψουμε τον κώδικα!

Αρχεία Python

Πριν αρχίσουμε να προσθέτουμε εντολές, είναι καλό να αποθηκεύσουμε το αρχείο μας. Από το μενού File επιλέγουμε Save Us...



Ονομάζουμε το πρόγραμμα μας "myfirstpython.py" και το αποθηκεύουμε. Έχουμε κάνει ήδη ένα τεράστιο βήμα στην Python!

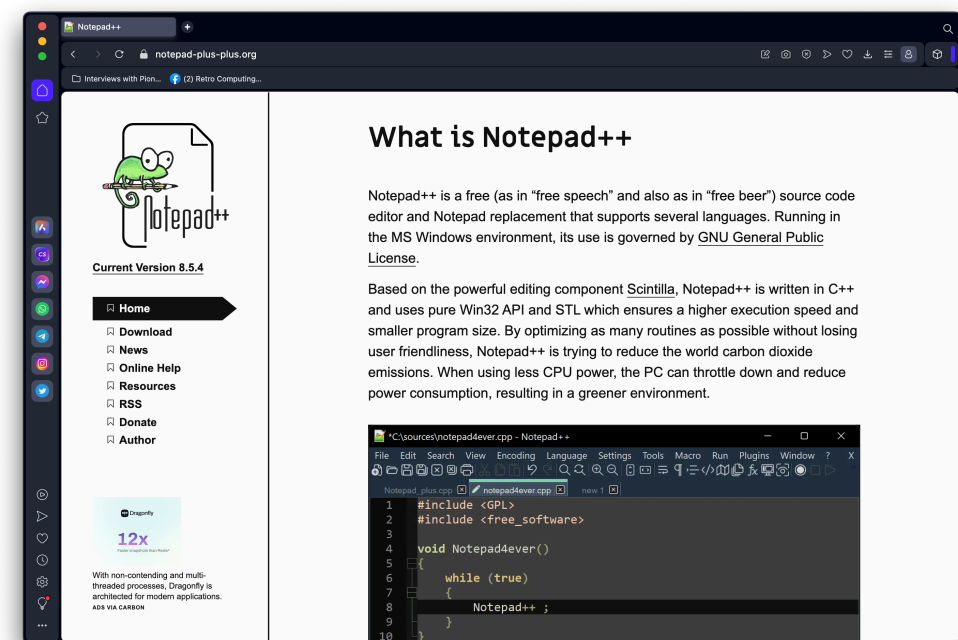


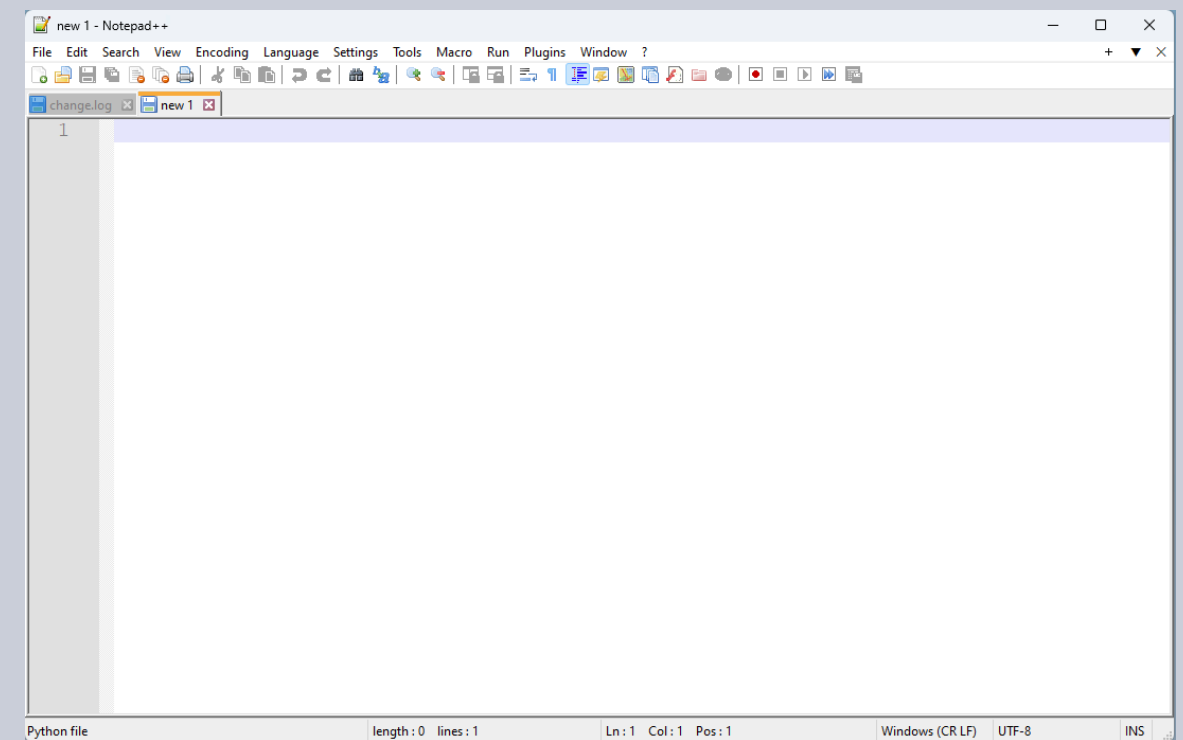
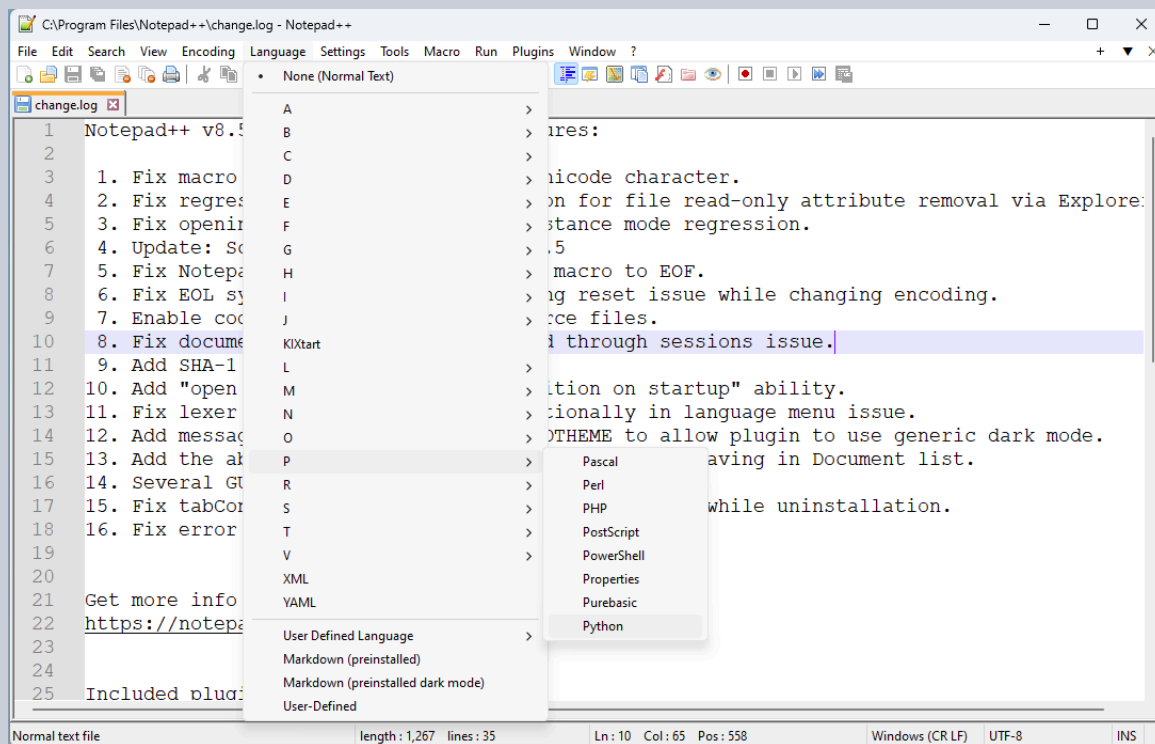
Τα αρχεία της Python έχουν κατάληξη .py, Το πρώτο μας αρχείο έχει ονομασία myfirstpython.py. Είναι καλό να δίνουμε στα αρχεία μας ονόματα που μας δείχνουν τι ακριβώς κάνει το πρόγραμμα.

Εναλλακτικά...

Εκτός από το IDLE της Python, μπορούμε να χρησιμοποιήσουμε και άλλες εφαρμογές, οι οποίες δίνουν και άλλες δυνατότητες (για παράδειγμα, υποστήριξη πολλών γλωσσών προγραμματισμών).

Υπάρχουν πάρα πολλά λογισμικά που μπορούν να μας εξυπηρετήσουν. Σε MacOS, μια καλή επιλογή είναι το BBEDIT. Σε Windows, μια καλή επιλογή είναι το Notepad++ (εικόνα κάτω). Και τα δύο είναι δωρεάν και μπορούμε να τα κατεβάσουμε από το App Store του υπολογιστή μας ή από τις ιστοσελίδες τους.





Σχετικά με το Notepad++

Αν θέλετε να δοκιμάσετε κάτι διαφορετικό, μπορείτε να γράψετε κώδικα στο Notepad++, μια εξαιρετική δωρεάν εφαρμογή για Windows.

Με την εκκίνηση του Notepad++, εμφανίζονται γενικές πληροφορίες για το πρόγραμμα που κατεβάσαμε. Μας ενδιαφέρει να "πούμε" στο Notepad++ ότι θέλουμε να προγραμματίσουμε σε Python.

Όταν ορίσουμε τη Python ως την κύρια γλώσσα, τότε εμφανίζεται και ο αντίστοιχος χρωματισμός στις εντολές.

Από το μενού Language, επιλέγουμε Python (από το γράμμα "P"), όπως φαίνεται και στην εικόνα πάνω.

Για να δημιουργήσουμε ένα νέο αρχείο, από το μενού File επιλέγουμε New (εικόνα πάνω δεξιά).

Το Notepad++ χρησιμοποιείται (και) από μαθητές/μαθήτριες και φοιτητές/φοιτήτριες, καθώς είναι δωρεάν και προσφέρει μεγάλη ευελιξία με υποστήριξη πολλών γλωσσών προγραμματισμού.



Let's Run!

```
myfirstpython.py - /Users/akoftero/Documents/myfirstpython.py (3.11.4)
print("Καλωσορίσατε στο πρόγραμμα 'Βρες την ηλικία μου'")
Ln: 1 Col: 57
```

Έχουμε γράψει την πρώτη μας εντολή! Σε αντίθεση με προηγούμενα παραδείγματα, με την αλλαγή γραμμής, δεν εκτελείται! Θα πρέπει, από το μενού Run, να επιλέξουμε "Run Module".

```
IDLE Shell 3.11.4
Python 3.11.4 (v3.11.4:d2340ef257, Jun 6 2023, 19:15:51) [Clang 13.0.0 (clang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: /Users/akoftero/Documents/myfirstpython.py =====
Καλωσορίσατε στο πρόγραμμα 'Βρες την ηλικία μου'
>>>
Ln: 6 Col: 0
```



Το πρόγραμμά μας "τρέχει" και εμφανίζεται το αποτέλεσμα του (το μήνυμα **Καλωσορίσατε στο πρόγραμμα 'Βρες την ηλικία μου'**).

Με την εντολή 'Run', το πρόγραμμά μας από Python (που καταλαβαίνουμε εμείς) "μεταφράζεται" στη γλώσσα που καταλαβαίνουν οι πύθωνες (οκ... οι υπολογιστές). Αυτό συμβαίνει κάθε φορά που επιλέγουμε το 'Run'



Στις επόμενες σελίδες θα εμπλουτίσουμε το πρόγραμμά μας με νέες εντολές!


```
*myfirstpython.py - /Users/akoftero/Documents/myfirstpython.py (3.11.4)*
print("*****")
print("Καλωσορίσατε στο πρόγραμμα 'Βρες την ηλικία μου'")
print("Μαντέψτε την ηλικία του φιλικού μας πύθωνα")
print("*****")|
```

Ln: 4 Col: 55

```
IDLE Shell 3.11.4
Python 3.11.4 (v3.11.4:d2340ef257, Jun 6 2023, 19:15:
51) [Clang 13.0.0 (clang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license()" for
more information.
>>>
===== RESTART: /Users/akoftero/Documents/myfir
stpython.py =====
*****
Καλωσορίσατε στο πρόγραμμα 'Βρες την ηλικία μου'
Μαντέψτε την ηλικία του φιλικού μας πύθωνα
*****
>>>
```

Ln: 9 Col: 0

Έχουμε προσθέσει 4 γραμμές εντολών print. Στην πρώτη και στην τέταρτη γραμμή, χρησιμοποιούμε το σύμβολο "*" για να εμφανιστεί πλαίσιο πάνω και κάτω από το κείμενο μας (εικόνα πάνω δεξιά).

Η γραμμή 2 και 3 του κώδικα μας (εικόνα πάνω) μας δίνει πληροφορίες για το πρόγραμμα. Στη γραμμή 2 μας δίνει το όνομα του παιχνιδιού, ενώ στη γραμμή 3 μας εξηγεί ποιος είναι ο σκοπός μας.



Το σύμβολο "*" μπορούμε να το χρησιμοποιήσουμε σε συνδυασμό με την εντολή print() για να δημιουργήσουμε σχήματα στην οθόνη. Θα δημιουργήσουμε σχήματα σε επόμενο κεφάλαιο αυτού του βιβλίου

Στη συνέχεια, θα προσθέσουμε και άλλο κώδικα.

```
print()
print()
print("Πόσα χρόνια ζει ένας πύθωνας;")
```

Οι δύο γραμμές print() που έχουμε προσθέσει, αφήνουν δύο γραμμές κενές ανάμεσα στο εισαγωγικό μέρος του προγράμματος, και στο ερώτημα που θέσαμε.

Στη συνέχεια, θα δείξουμε πώς δίνουμε πληροφορίες με το πληκτρολόγιο.



Εισαγωγή δεδομένων

Μέχρι τώρα, καταφέραμε να γράψουμε κώδικα 7 γραμμών. Όμως, το πρόγραμμά μας δεν μας επιτρέπει να δώσουμε την ηλικία που μπορεί να έχει ένας πύθωνας. Για να δώσουμε πληροφορίες από το πληκτρολόγιο, θα πρέπει να χρησιμοποιήσουμε μια νέα εντολή, την `input`.

```
age=input()
```

Ας δούμε την πιο πάνω εντολή: το `"age"` είναι μια μεταβλητή. Στη μεταβλητή αυτή θα αποθηκεύσουμε την ηλικία (αριθμό) που θα δώσουμε από το πληκτρολόγιο.

Η `"input"` είναι η εντολή με την οποία η Python περιμένει να δώσουμε μια πληροφορία με το πληκτρολόγιο. Μόλις πατήσουμε το πλήκτρο ENTER, η πληροφορία που πληκτρολογήσαμε (αριθμός, κείμενο) αποθηκεύεται στη μεταβλητή `"age"`.

```
print("Πόσα χρόνια ζει ένας πύθωνας;")
age=input()
```



Σημείωση: η μεταβλητή πιο πάνω δεν έχει τη σωστή μορφή! Αυτό θα το εξηγήσουμε αργότερα, όταν θα μιλήσουμε για αλφαριθμητικές τιμές και για ακέραιους!

Όταν εκτελέσουμε το πιο πάνω πρόγραμμα, εμφανίζεται το μήνυμα της `print` ("Πόσα χρόνια ζει ένας πύθωνας;") και αναμένει ο υπολογιστής να πληκτρολογήσουμε μια πληροφορία. Πατάμε ENTER για να ολοκληρωθεί η διαδικασία.

Για να κάνουμε πιο ενδιαφέρον το πρόγραμμά μας, προσθέτουμε ακόμη μια εντολή:

```
print("Ζει μέχρι τα",age)
```

Η πιο πάνω εντολή εμφανίζει στην οθόνη το μήνυμα "Ζει μέχρι τα" και ακολουθεί ο αριθμός που δώσαμε με το πληκτρολόγιο (και αποθηκεύθηκε στη μεταβλητή `"age"`).

Στη συνέχεια θα βελτιώσουμε ακόμη περισσότερο τον κώδικα, με χρήση και άλλων εντολών.

```
Python 3.11.4 (v3.11.4:d2340ef257, Jun 6 2023, 19:15:51) [Clang
13.0.0 (clang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license()" for more infor
mation.
>>>
===== RESTART: /Users/akoftero/Documents/myfirstpython.py
=====
*****
Καλωσορίσατε στο πρόγραμμα 'Βρες την ηλικία μου'
*****

Πόσα χρόνια ζει ένας πύθωνας;
20
Ο πύθωνας ζει μέχρι τα 20
>>>
```

Ας δούμε ξανά την εντολή input():

```
print("Πόσα χρόνια ζει ένας πύθωνας;")  
age=input()
```

Στο πιο πάνω παράδειγμα χρησιμοποιήσαμε δύο εντολές. Η print() μας δίνει το μήνυμα που εξηγεί τι πρέπει να πληκτρολογήσουμε. Στη δεύτερη εντολή, με την input() ζητάμε να πληκτρολογήσει κάποιος/α έναν αριθμό. Ο αριθμός αποθηκεύεται στη μεταβλητή age.

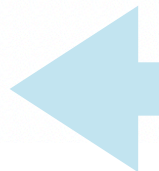
Θα μπορούσαμε τις δύο πιο πάνω να τις αντικαταστήσουμε με μία εντολή, που κάνει ακριβώς το ίδιο:

```
age=input("Πόσα χρόνια ζει ένας πύθωνας;")
```

Μέσα στις παρενθέσεις της input() έχουμε πληκτρολογήσει το μήνυμα (κείμενο) που είχαμε στην print().

```
*****  
Καλωσορίσατε στο πρόγραμμα 'Βρες την ηλικία μου'  
*****
```

Πόσα χρόνια ζει ένας πύθωνας;



Στο πιο πάνω παράδειγμα, το σημείο εισαγωγής του αριθμού είναι στα δεξιά του κειμένου (εικόνα κάτω αριστερά). Μπορούμε να τροποποιήσουμε την εντολή input() ώστε τον αριθμό να τον πληκτρολογήσουμε στην επόμενη γραμμή:

```
age=input("Πόσα χρόνια ζει ένας πύθωνας;\n")
```

Η προσθήκη του \n μεταφέρει την πληροφορία που θα δώσουμε από το πληκτρολόγιο στην αμέσως επόμενη γραμμή (εικόνα κάτω).

```
*****  
Καλωσορίσατε στο πρόγραμμα 'Βρες την ηλικία μου'  
*****
```

Πόσα χρόνια ζει ένας πύθωνας;



Στην επόμενη σελίδα θα δούμε και άλλες χρήσεις της εντολής input().

Ο αγαπημένος μας πύθωνας μπορεί να πληκτρολογήσει και κείμενο εκτός από αριθμούς!

```
name=input("Όνομα παίκτη;")  
print("Καλωσόρισες",name)
```

Στην πρώτη εντολή, ζητάμε να γράψει ο παίκτης ή η παίκτρια το όνομα του/της. Το όνομα θα αποθηκευθεί στη μεταβλητή name.

Με την εντολή print, εκτυπώνεται το μήνυμα "Καλωσόρισες" και στη συνέχεια, το περιεχόμενο της μεταβλητής name (εικόνα κάτω).

```
*****  
Καλωσορίσατε στο πρόγραμμα 'Βρες την ηλικία μου'  
*****
```

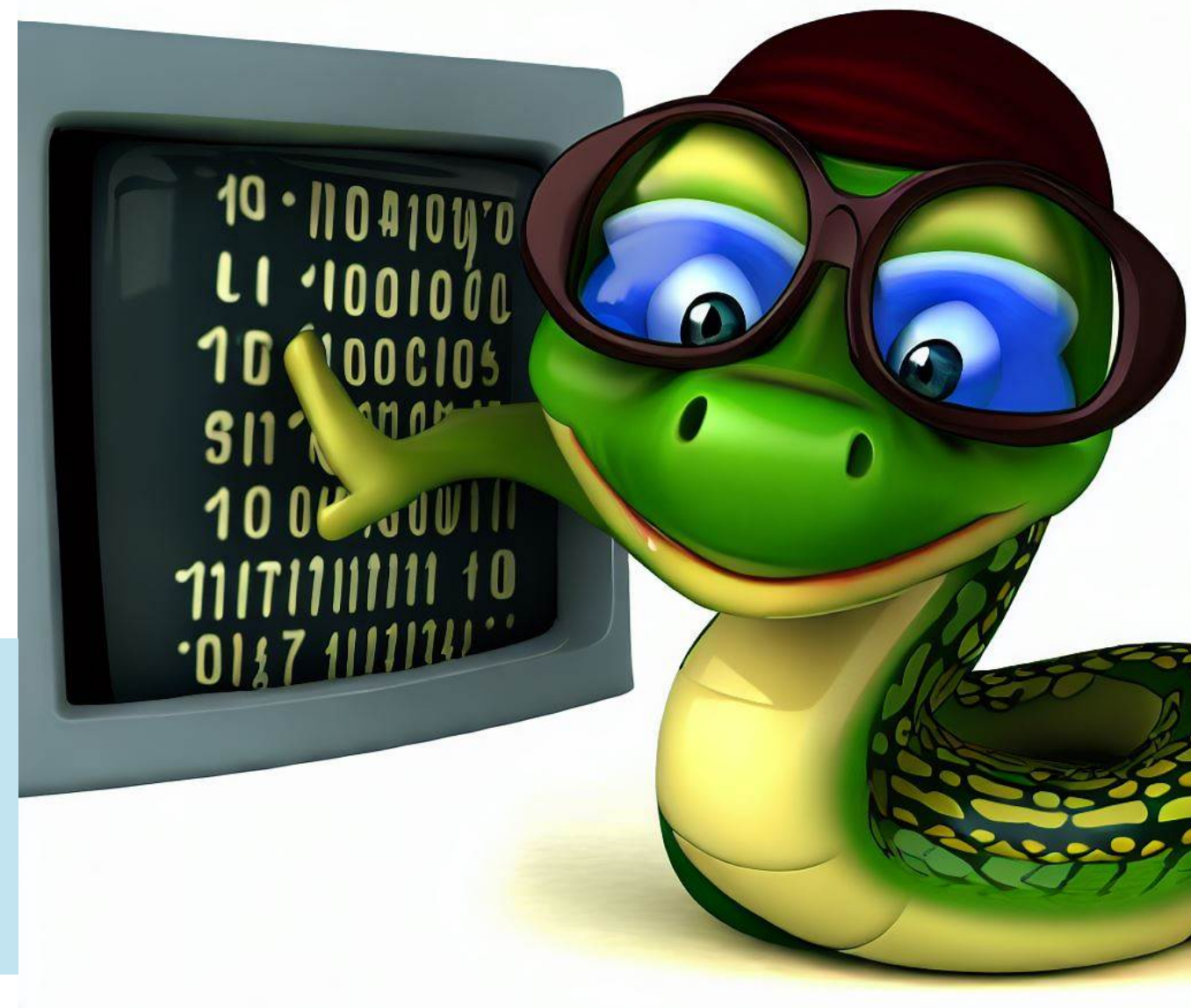
```
Όνομα παίκτη; Αλέξανδρος  
Καλωσόρισες Αλέξανδρος
```



Λίγη γραμματική: όταν πληκτρολογούμε το όνομά μας, αυτό είναι στην ονομαστική. Όταν το προβάλλει ο υπολογιστής, αυτό συνεχίζει να παραμένει στην ονομαστική, αν και θα έπρεπε να είναι στην κλητική.

Την εντολή input() μπορούμε να τη χρησιμοποιήσουμε για να εισαγάγουμε αριθμούς με τους οποίους θα εκτελέσουμε μαθηματικές πράξεις.

Μπορούμε επίσης να χρησιμοποιήσουμε την input() για να επιλέξουμε (πολλαπλή επιλογή). Και αυτό θα το δούμε στη συνέχεια του βιβλίου μας!

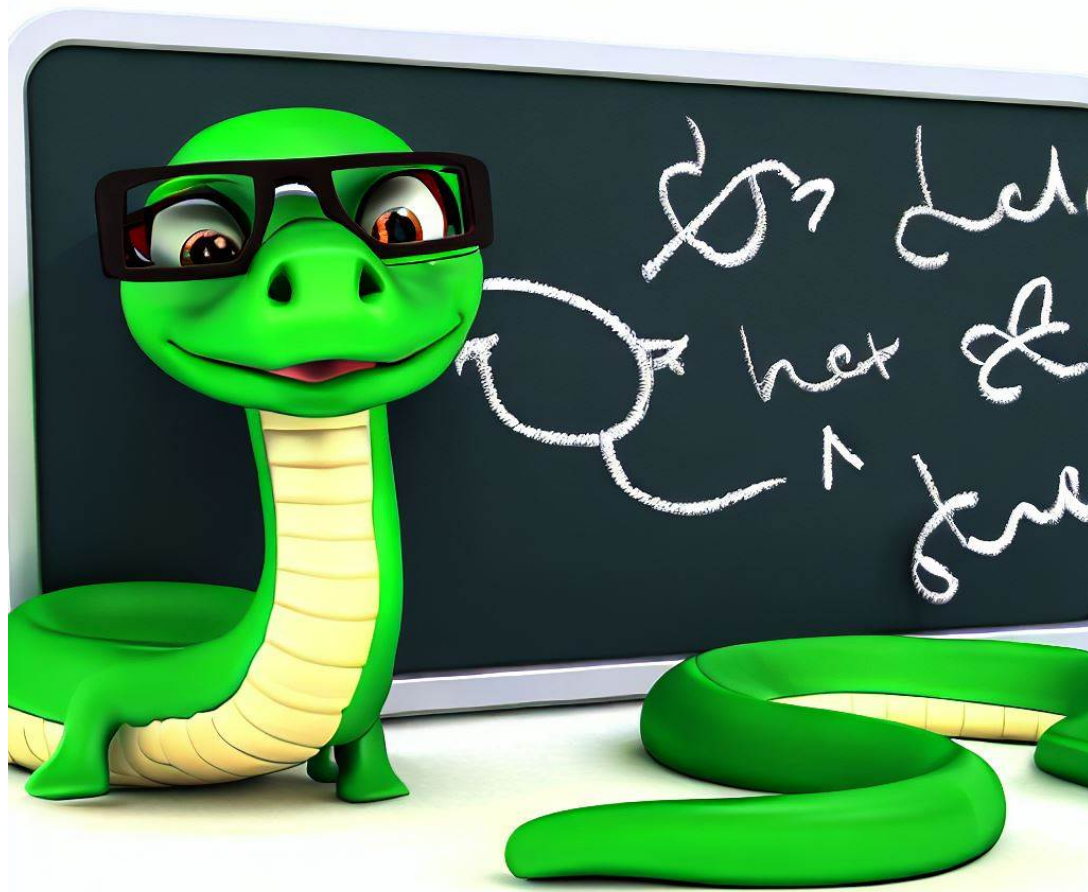


Ας σχολιάσουμε...

Οι πύθωνες είναι σχολαστικά και "σχολιαστικά" ζώα. Τους αρέσει να βάζουν σημειώσεις στα κείμενά τους. Αυτό τους βοηθά να καταλαβαίνουν τι γράφει κάθε παράγραφος ή τμήμα κειμένου. Το ίδιο και όταν γράφουν προγράμματα στον υπολογιστή!

Τα σχόλια (comments), βοηθούν στο να καταλαβαίνουμε καλύτερα τι κάνει ένα πρόγραμμα ή/και μέρη ενός προγράμματος.

Για να γράψουμε σχόλιο, θα πρέπει -με κάποιο τρόπο- να πούμε στην Python πως δεν είναι εντολή για εκτέλεση, και απλά να την αγνοήσει.



```
*myfirstpython.py - /Users/akoftero/Documents/myfirstpython.py (3.11.4)*
print("*****")
print("Καλωσορίσατε στο πρόγραμμα 'Βρες την ηλικία μου'")
print("*****")
print()
print()
#print("Πόσα χρόνια ζει ένας πύθωνας;")
name=input("Όνομα παίκτη;")
print("Καλωσόρισες",name)
age=input("Πόσα χρόνια ζει ένας πύθωνας;\n")
|
```

Ln: 11 Col: 0

Στο πιο πάνω παράδειγμα, η εντολή `print()` εμφανίζεται με κόκκινο χρώμα. Μπροστά της έχουμε βάλει το σύμβολο `"#"`. Το συγκεκριμένο σύμβολο, στην αρχή της εντολής, δηλώνει στην Python ότι πρέπει να την αγνοήσει στην εκτέλεση γιατί πρόκειται για σχόλιο.

Στην πιο κάτω εικόνα βλέπουμε τα σχόλια στην αρχή μιας γραμμής (εξηγούν τι κάνει ο κώδικας) αλλά και δεξιά από μια εντολή (εξηγούν τι κάνει η εντολή).

```
*myfirstpython.py - /Users/akoftero/Documents/myfirstpython.py (3.11.4)*
print("*****")
print("Καλωσορίσατε στο πρόγραμμα 'Βρες την ηλικία μου'")
print("*****")
print()
print()
#Με τις εντολές που ακολουθούν, ζητάμε έναν αριθμό
#για να μαντέψουμε πόσα χρόνια ζει ένας πύθωνας
name=input("Όνομα παίκτη;")
print("Καλωσόρισες",name)
age=input("Πόσα χρόνια ζει ένας πύθωνας;\n") #ζητάμε έναν αριθμό
|
```

Ln: 10 Col: 64

Ο αποφασιστικός πύθωνας!

Καιρός να αποφασίσει ο φιλικός πράσινος πύθωνας αν βρήκαμε σωστά την ηλικία του! Θα πρέπει -σωστά μαντέψατε- να χρησιμοποιήσουμε την εντολή "if". Ένας πύθωνας ζει περίπου 10 χρόνια.

```
if age==10:  
    print("Βρήκατε τη σωστή ηλικία!")
```

Στην πρώτη γραμμή, συγκρίνουμε την τιμή της μεταβλητής age (που δώσαμε από το πληκτρολόγιο) με τον αριθμό 10 (τα χρόνια που ζει ένας πύθωνας).

Στη σύγκριση, έχουμε το διπλό σύμβολο == που δηλώνει το "ίσο με". Επίσης χρησιμοποιούμε το "μεγαλύτερο ή ίσο" (>=), "μικρότερο ή ίσο" (<=) και "μικρότερο/μεγαλύτερο".

ΑΝ ισχύει η συνθήκη (ο αριθμός που δώσαμε είναι το 10, ΤΟΤΕ εμφανίζεται το μήνυμα "Βρήκατε τη σωστή ηλικία".



Η εντολή print() εμφανίζεται κάτω και δεξιά της if. Όλες οι εντολές που εκτελούνται ΑΝ ισχύει η συνθήκη, πρέπει να είναι κάτω από την if και με εσοχή (indentation).

Θέλουμε, όμως, να εμφανίζει και ένα μήνυμα όταν δώσουμε διαφορετικό αριθμό. Θα πρέπει να προσθέσουμε και το 'else'.

```
if age==10:  
    print("Βρήκατε τη σωστή ηλικία!")  
else:  
    print("Δυστυχώς κάνατε λάθος")
```

ΑΝ η τιμή της μεταβλητής age είναι το 10, ΤΟΤΕ θα εμφανίσει το πρώτο μήνυμα, ΔΙΑΦΟΡΕΤΙΚΑ (else) θα εμφανίσει το δεύτερο μήνυμα.

Μπορούμε να γράψουμε τις εντολές αυτές σε μία γραμμή:

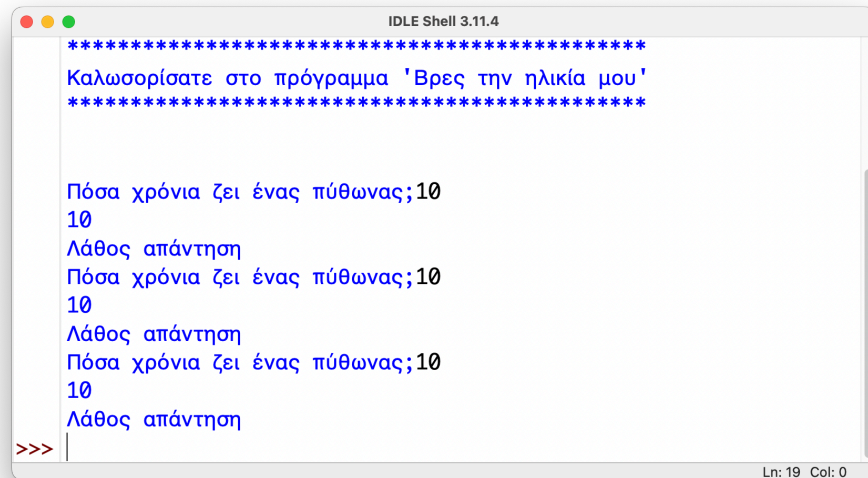
```
print("Σωστά!") if age==10 else print("Λάθος")
```

Στην εικόνα πιο κάτω, βλέπουμε το αποτέλεσμα του προγράμματος, όταν δώσουμε λάθος απάντηση. Στην επόμενη σελίδα θα μάθουμε πώς μπορούμε να συνεχίσουμε το παιχνίδι και να δώσουμε ξανά απάντηση.

```
IDLE Shell 3.11.4  
*****  
Καλωσορίσατε στο πρόγραμμα 'Βρες την ηλικία μου'  
*****  
  
'Όνομα παίκτη; Αλέξανδρος  
Καλωσόρισες Αλέξανδρος  
Πόσα χρόνια ζει ένας πύθωνας; 12  
Λάθος απάντηση  
>>>
```


Σφάλμα στον κώδικα!

Στην προηγούμενη σελίδα, είδαμε το μήνυμα που βγάζει το πρόγραμμά μας, αν δώσουμε λάθος αριθμό για την ηλικία του πύθωνα. Τι συμβαίνει όμως όταν δώσουμε τον σωστό αριθμό;



```
*****
Καλωσόρισατε στο πρόγραμμα 'Βρες την ηλικία μου'
*****

Πόσα χρόνια ζει ένας πύθωνας;10
10
Λάθος απάντηση
Πόσα χρόνια ζει ένας πύθωνας;10
10
Λάθος απάντηση
Πόσα χρόνια ζει ένας πύθωνας;10
10
Λάθος απάντηση
>>>
```

Όπως βλέπουμε και στην πιο πάνω εικόνα, παρόλο που πληκτρολογούμε τη σωστή απάντηση (που είναι το 10), το θεωρεί λάθος! Μα γιατί; αφού ο κώδικάς μας είναι σωστός. Ή κάτι πάει λάθος...;

Ακέραιοι και πορτοκάλια...

Σε προηγούμενη σελίδα, είχαμε μιλήσει για είδη μεταβλητών. Κάποιες μεταβλητές, οι τιμές που παίρνουν είναι ακέραιοι αριθμοί (integers ή int). Άλλες είναι κείμενο

(που μπορεί να περιέχει ή να αποτελείται και από αριθμούς) και ονομάζονται string. Άλλες δέχονται δεκαδικούς (floating point ή float).

Αν θέλουμε να μάθουμε μια μεταβλητή τι τύπος είναι, χρησιμοποιούμε το type().

Στον κώδικά μας έχουμε τη μεταβλητή age, στην οποία δίνουμε τιμή από το πληκτρολόγιο.

```
age=input("Πόσα χρόνια ζει ένας πύθωνας;\n")
```

Στη συνέχεια, συγκρίνουμε αυτή τη μεταβλητή για να δούμε αν ισούται με τον αριθμό 10.

```
if age==10:
```

Όμως, υπάρχει κάτι που δεν προσέξαμε: όταν δίνουμε πληροφορίες στην input, αυτόματα τις τοποθετεί στη μεταβλητή όχι ως ακέραιο (int) αλλά ως κείμενο (str)!

Έτσι, ό,τι αριθμό και αν δώσουμε στην input από το πληκτρολόγιο, θα θεωρήσει πως είναι κείμενο και όχι αριθμός και άρα η σύγκριση θα είναι πάντα λάθος (σαν να του λέμε να συγκρίνει αν τα πορτοκάλια είναι ο αριθμός 10)! Στη συνέχεια θα δούμε πώς θα το διορθώσουμε!

Τύπος μεταβλητής

Θέλουμε να δώσουμε αριθμό από το πληκτρολόγιο, ώστε να μπορεί να τον συγκρίνει με την ηλικία του πύθωνα και να διαπιστώσει αν βρήκαμε τη σωστή απάντηση ή όχι.

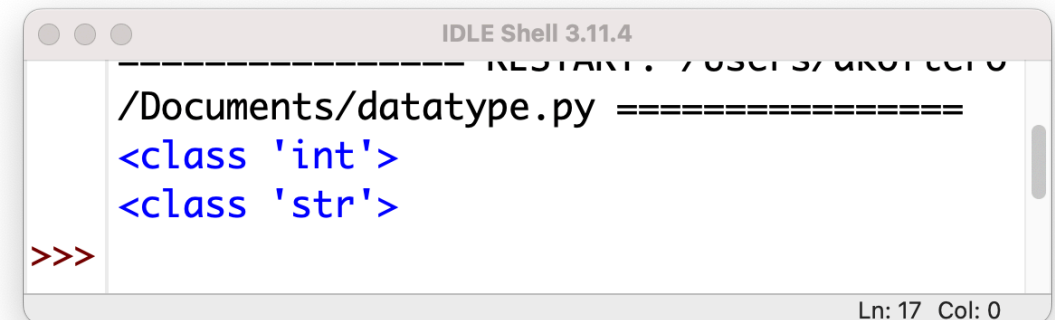
Το ερώτημα είναι: πώς λέμε στην Python ότι ο αριθμός που θέλουμε να δώσουμε είναι ακέραιος και όχι κείμενο; Πρώτα από όλα, ας δούμε πώς εντοπίζουμε τον τύπο μιας μεταβλητής...

Θα χρησιμοποιήσουμε μια νέα εντολή, την `type()` για να ελέγξουμε τον τύπο μιας μεταβλητής. Για να εμφανίσουμε τον τύπο της μεταβλητής στην οθόνη, θα την πληκτρολογήσουμε μέσα στην `print()`. Για να δούμε το πιο κάτω παράδειγμα:

```
x=10      #στη μεταβλητή x δίνουμε το 10 ως αριθμό
y="10"    #στη μεταβλητή y δίνουμε το 10 ως κείμενο
print(type(x))
print(type(y))
```

Η `type(x)` εντοπίζει το είδος της μεταβλητής `x`. Όμως, για να εμφανιστεί το περιεχόμενο της στην οθόνη, πρέπει να την βάλουμε μέσα στην `print()`. Το ίδιο και για τη

μεταβλητή `y`. Εκτελούμε το πρόγραμμα για να δούμε το αποτέλεσμα:



```
----- IDLE Shell 3.11.4 -----
/Documents/datatype.py =====
<class 'int'>
<class 'str'>
>>>
```

Η πρώτη μεταβλητή (`x=10`) αναγνωρίζεται ως ακέραιος (`<class 'int'>`). Η δεύτερη μεταβλητή, αν και περιέχει τον αριθμό 10, επειδή είναι σε εισαγωγικά (`y="10"`), αναγνωρίζεται ως κείμενο (αλφαριθμητικό - `str`), γι'αυτό και στη δεύτερη γραμμή (εικόνα πάνω) εμφανίζεται το μήνυμα `<class 'str'>`.

Οι μεταβλητές τύπου `str` (string) χρησιμοποιούνται για την εισαγωγή κειμένου, αλλά μπορεί να περιλαμβάνουν και αριθμούς (ή και μόνο αριθμούς), αλλά και σύμβολα. Γι'αυτό και ονομάζεται "αλφαριθμητική" (string).



Στη συνέχεια θα λύσουμε το πρόβλημα της εισαγωγής αριθμού ως ακεραίου από το πληκτρολόγιο!

Μεταβλητή ως Ακέραιος

Αν δοκιμάσουμε να δώσουμε έναν αριθμό από το πληκτρολόγιο, με την `input()`, θα αναγνωρίζεται ως κείμενο (αλφαριθμητικό) και όχι ως ακέραιος αριθμός (σημείωση: το ίδιο ισχύει και για δεκαδικούς).

Ας δούμε την εντολή με την οποία δίνουμε πληροφορία με το πληκτρολόγιο, στη μεταβλητή `age`.

```
age=input("Παρακαλώ δώστε έναν αριθμό")
```

Θα κάνουμε μια αλλαγή στον πιο πάνω κώδικα, ώστε η τιμή της μεταβλητής `age` να είναι ακέραιος:

```
age=int(input("Παρακαλώ δώστε ένα αριθμό"))
```

Η πιο πάνω αλλαγή της εντολής `input()` λύνει το πρόβλημα, καθώς ενημερώνουμε την Python ότι θα δώσουμε αριθμό (ακέραιο) από το πληκτρολόγιο.



Προσέξτε τις διπλές παρενθέσεις στο τέλος της εντολής πιο πάνω. Αυτό οφείλεται στο ότι η `input()` θα πρέπει να μπει μέσα στην `int()`.

```
int( input("Enter number") )
```

Ο σωστός κώδικας με την αλλαγή είναι:

```
*myfirstpython.py - /Users/akoftero/Documents/myfirstpython.py (3.11.4)*
print("*****")
print("Καλωσορίσατε στο πρόγραμμα 'Βρες την ηλικία μου'")
print("*****")
print()
print()
#Με τις εντολές που ακολουθούν, ζητάμε έναν αριθμό
#για να μαντέψουμε πόσα χρόνια ζει ένας πύθωνας

count=0 #μετρά τις προσπάθειες που κάνουμε
age=1 #αρχική τιμή του age
while count<3:
    | age=int(input("Πόσα χρόνια ζει ένας πύθωνας;")) #ζητάμε έναν αριθμό
    | print(age)
    | if age==10:
    |     | print("Σωστή απάντηση")
    |     | break
    | else:
    |     | print("Λάθος απάντηση")

    count=count+1 #αυξάνει η τιμή του count κατά 1

Ln: 12 Col: 1
```

Αν δοκιμάσουμε τώρα να εκτελέσουμε το πρόγραμμα, θα δούμε πως αναγνωρίζει το 10 ως τη σωστή απάντηση!

```
IDLE Shell 3.11.4

Πόσα χρόνια ζει ένας πύθωνας;10
10
Σωστή απάντηση
>>>

Ln: 27 Col: 0
```


"Έντομα" στον κώδικα!

Οι πύθωνες, όταν μεταφράζουν από μια γλώσσα σε μια άλλη, προτιμούν να το κάνουν πρόταση με πρόταση. Άλλοι, προτιμούν να παίρνουν το κείμενο (ή ό,τι έχει να πει κάποιος ομιλητής) και να το μεταφράζουν ολόκληρο.

Αυτό έχει και πλεονεκτήματα και μειονεκτήματα (που δε θα αναλύσουμε σε αυτό το βιβλίο). Όμως, καθώς "τρέχει" ένα πρόγραμμα η Python, εκτελεί τις εντολές με τη σειρά.

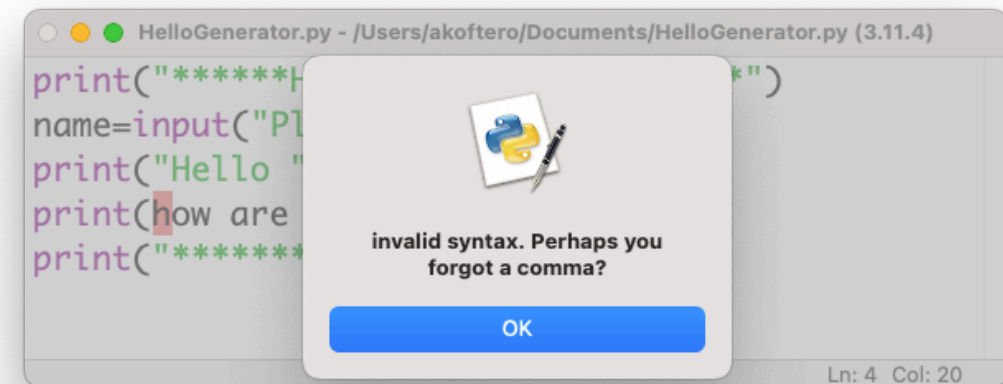
```
HelloGenerator.py - /Users/akoftero/Documents/HelloGenerator.py (3.11.4)
print("*****Hello Generator *****")
name=input("Please enter name: ")
print("Hello ",name)
print(How are you today)
print("*****Goodbye*****")
Ln: 1 Col: 0
```



Ο όρος "bug" (έντομο) χρησιμοποιείται από το 1843 και λέγεται ότι τον δημιούργησε ο Θωμάς Έντισον, ο διάσημος εφευρέτης. Το 1947, χρησιμοποιήθηκε από την Γκρέης Χόπερ, εξαιτίας ενός σκώρου που προκάλεσε βραχυκύκλωμα στον υπολογιστή Mark 2!

Στην εικόνα (αριστερά) βλέπουμε πως υπάρχει λάθος στην τέταρτη γραμμή: το κείμενο δεν έχει μπει σε εισαγωγικά.

Κατά την εκτέλεση του προγράμματος, η Python θα μας ενημερώσει άμεσα για το συγκεκριμένο λάθος και θα υποδείξει τη γραμμή στην οποία βρίσκεται.



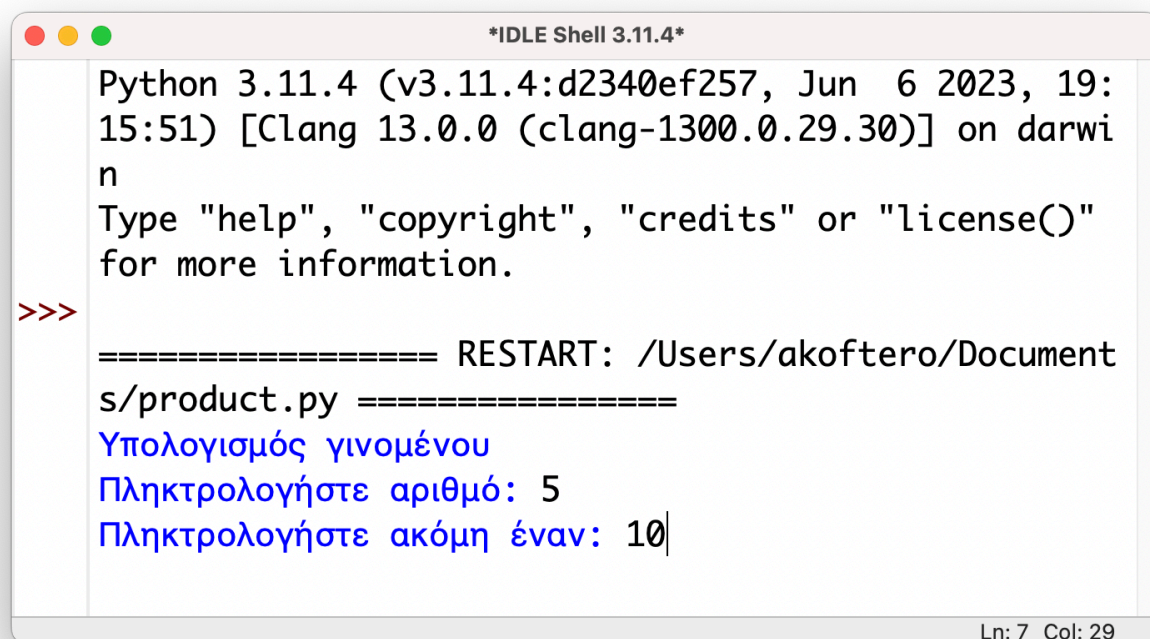
Το λάθος πιο πάνω είναι συντακτικό, επειδή λείπουν ή βάλαμε λάθος κάποιους χαρακτήρες. Στη συνέχεια θα δούμε και άλλα λάθη.



Ας δούμε ένα πιο “σοβαρό” λάθος. Θα δημιουργήσουμε ένα απλό πρόγραμμα με το οποίο υπολογίζουμε το γινόμενο δύο αριθμών:

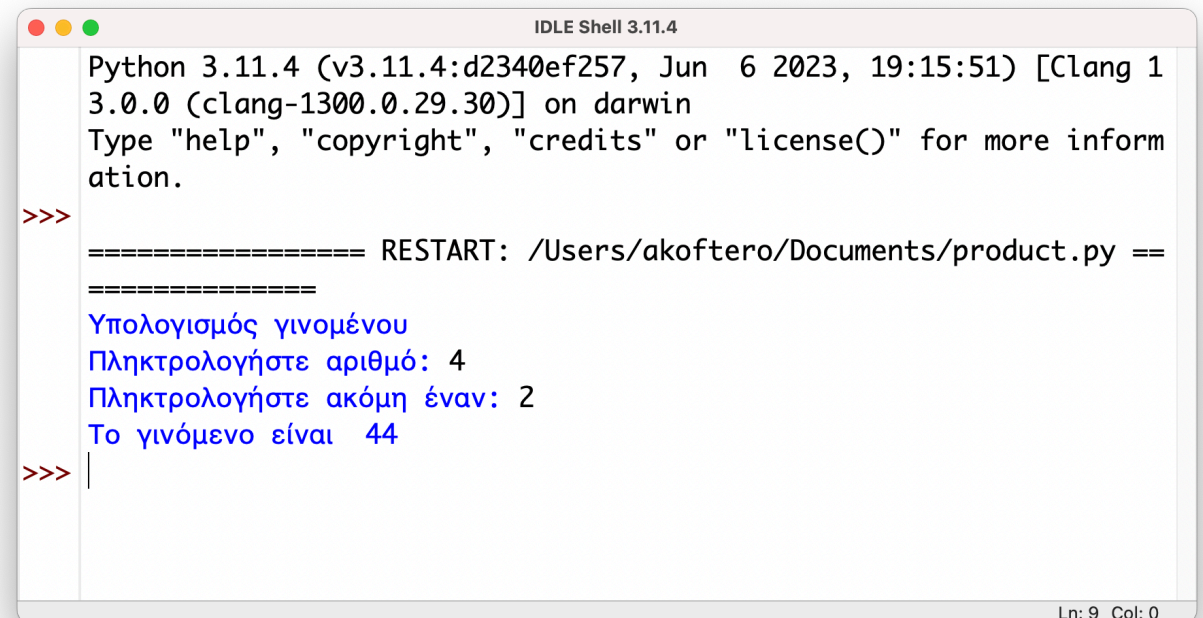
```
print("Υπολογισμός γινομένου")
number1=input("Πληκτρολογήστε έναν αριθμό: ")
number2=int(input("Πληκτρολογήστε ακόμη έναν: "))
print("Το γινόμενο είναι ", number1*number2)
```

Δεν υπάρχει κανένα “λάθος” στον πιο πάνω κώδικα. Κατά την εκτέλεση του, όντως θα μας ζητήσει να δώσουμε 2 αριθμούς από το πληκτρολόγιο. Κάθε αριθμός θα μπει σε μια μεταβλητή (number1 και number2). Όταν τρέξουμε το πρόγραμμα, θα εκτελεστούν κανονικά οι πρώτες 3 γραμμές:



```
Python 3.11.4 (v3.11.4:d2340ef257, Jun 6 2023, 19:15:51) [Clang 13.0.0 (clang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: /Users/akoftero/Documents/product.py =====
Υπολογισμός γινομένου
Πληκτρολογήστε αριθμό: 5
Πληκτρολογήστε ακόμη έναν: 10
```

Έχουμε δώσει 2 αριθμούς από το πληκτρολόγιο. Όταν πατήσουμε το πλήκτρο ENTER για να προχωρήσει στην εκτέλεση και της τελευταίας εντολής, εμφανίζεται το ακόλουθο αποτέλεσμα:



```
Python 3.11.4 (v3.11.4:d2340ef257, Jun 6 2023, 19:15:51) [Clang 13.0.0 (clang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: /Users/akoftero/Documents/product.py =====
Υπολογισμός γινομένου
Πληκτρολογήστε αριθμό: 4
Πληκτρολογήστε ακόμη έναν: 2
Το γινόμενο είναι 44
>>> |
```

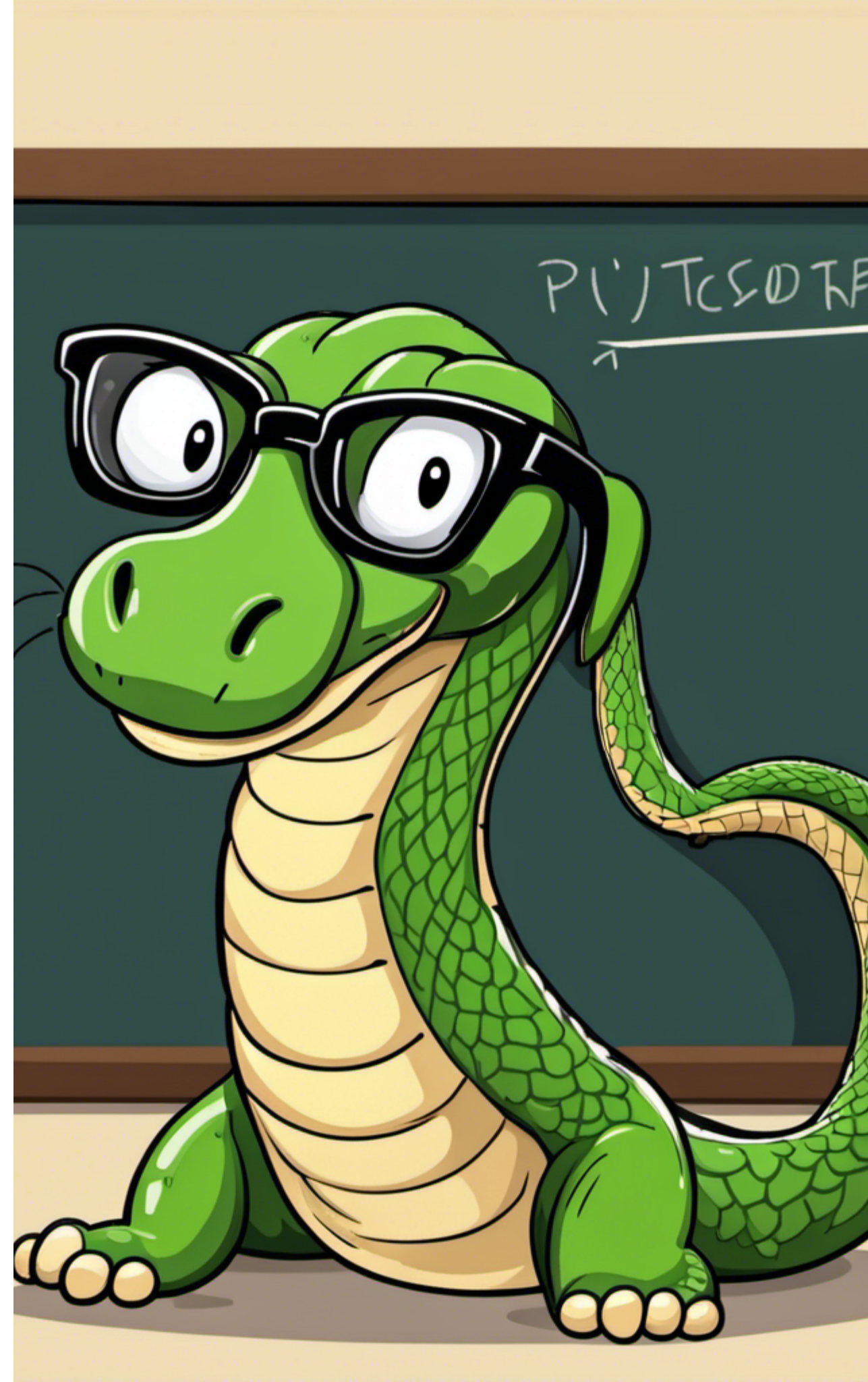
Αν το τρέξουμε ξανά, ακόμη και με τους ίδιους αριθμούς (4 και 2), θα δούμε πως βγαίνει λανθασμένο γινόμενο. Ο λόγος είναι απλός: η πρώτη μεταβλητή παίρνει μια τιμή “αλφαριθμητική” και όχι ακέραιο αριθμό! Θα πρέπει να την αλλάξουμε σε :

```
number1=int(input("Πληκτρολογήστε έναν αριθμό"))
```

Με αυτή την αλλαγή, ο αριθμός που θα δώσουμε στη μεταβλητή αυτή να αναγνωριστεί από τον κώδικα μας ως ακέραιος!

Τι μάθαμε μέχρι τώρα

- Στο κεφάλαιο 3 εργαστήκαμε με κώδικα πολλών γραμμών και δημιουργήσαμε ένα απλό παιχνίδι
- Με τις εντολές `print()` μπορούμε να εμφανίσουμε στην οθόνη, ταυτόχρονα, πολλές γραμμές με πληροφορίες
- Οι μεταβλητές χρησιμεύουν για να αποθηκεύουν τιμές (κείμενο, ακέραιος, δεκαδικός...) τις οποίες χρησιμοποιούμε στον κώδικα, συνήθως για σκοπούς σύγκρισης, μαθηματικών πράξεων κ.α.
- Με την εντολή `input()` μπορούμε να δώσουμε από το πληκτρολόγιο τιμή σε μια μεταβλητή
- Με την εντολή `if...else` μπορούμε να ελέγξουμε αν ισχύει μια συνθήκη (π.χ. $10 > 5$)
- Με το `type()` μπορούμε να διαπιστώσουμε τον τύπο της μεταβλητής
- Με την `int(input())` εισαγάγουμε μια τιμή ως ακέραιο.



Δραστηριότητες

1. Ποιο είναι το αποτέλεσμα της εκτέλεσης του πιο κάτω κώδικα; Προσπαθήστε να το γράψετε χωρίς να εκτελεστεί στον υπολογιστή.

```
name=input("Παρακαλώ γράψτε το όνομά σας")
surname=input("Παρακαλώ γράψτε το επίθετό σας")
print("Καλωσόρισες",name, surname)
```

2. Τι θα εμφανιστεί στην οθόνη όταν εκτελεστεί ο πιο κάτω κώδικας; Να το γράψετε χωρίς να εκτελεστεί στον υπολογιστή.

```
print("*****")
print("*    Υπολογισμός δυνάμεων    *")
print("*****")
print()
base=int(input("Παρακαλώ δώστε έναν αριθμό"))
power=int(input("Παρακαλώ δώστε τη δύναμη"))
print("Το",base,"στη δύναμη",power,"ισούται",base**power)
```

3. Να γράψετε ένα πρόγραμμα το οποίο να ζητά από το πληκτρολόγιο έναν αριθμό από το 1-10. Στη συνέχεια, να εμφανίζει τον πίνακα του πολλαπλασιασμού αυτού του αριθμού.
4. Να γράψετε ένα πρόγραμμα με το οποίο να ζητάτε τον "κωδικό" από το πληκτρολόγιο. Αν ο κωδικός είναι σωστός ("mypassword"), να εμφανίζεται το μήνυμα "Πολύ σωστά". Διαφορετικά να σας βγάζει το μήνυμα "κάνετε λάθος" και να σταματά.
5. Να διορθώσετε τον πιο κάτω κώδικα ώστε να λειτουργεί σωστά το πρόγραμμα.

```
@Υπολογισμός γινόμενου δύο αριθμών
print(Καλωσορίσατε στο πρόγραμμά μας)
number1=input("Δώστε τον πρώτο αριθμό")
number2=input("Δώστε τον δεύτερο αριθμό")
print("Το γινόμενό τους είναι",number1*Number2)
```