



**ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ**
UNIVERSITY OF PATRAS

**ΣΧΟΛΗ ΟΙΚΟΝΟΜΙΚΩΝ ΕΠΙΣΤΗΜΩΝ ΚΑΙ ΔΙΟΙΚΗΣΗΣ
ΕΠΙΧΕΙΡΗΣΕΩΝ**

ΤΜΗΜΑ ΔΙΟΙΚΗΤΙΚΗΣ ΕΠΙΣΤΗΜΗΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ

(πρώην Τμήμα Διοίκησης Επιχειρήσεων – Μεσολόγγι)

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΑΛΕΞΑΝΔΡΟΣ ΑΦΕΝΔΡΑΣ

ΜΕ Α.Μ. : 15046

«ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΡΥΘΜΩΝ»

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΔΡ.ΑΡΙΣΤΟΓΙΑΝΝΗΣ ΓΑΡΜΠΗΣ

ΜΕΣΟΛΟΓΓΙ 2019

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τον καθηγητή μου κο. Αριστογιάννη Γαρμπή ολόψυχα, γιατί η παρούσα εργασία έγινε με την καθοδήγησή του και τις συμβουλές του. Χάρη σε αυτόν πραγματοποιήθηκε και ολοκληρώθηκε.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΕΧΟΜΕΝΑ.....	- 3 -
ΠΕΡΙΛΗΨΗ	- 7 -
<i>ΕΙΣΑΓΩΓΗ</i>	- 8 -
ΚΕΦΑΛΑΙΟ 1	- 10 -
1. ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ –ΔΙΑΚΡΙΣΗ – ΚΑΙ ΙΣΤΟΡΙΚΗ ΕΞΕΛΙΞΗ.....	- 10 -
1.1 ΟΡΙΣΜΟΣ ΓΛΩΣΣΑΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ.....	- 10 -
1.2 ΔΙΑΚΡΙΣΗ ΤΗΣ Γλώσσα Προγραμματισμού.....	- 10 -
1.3 Ιστορία των γλωσσών προγραμματισμού	- 11 -
1.3.1 Πριν από το 1940.....	- 11 -
1.3.2 Η δεκαετία του 1940	- 12 -
1.3.3 1967-1978: καθιέρωση θεμελιωδών παραδειγμάτων	- 13 -
1.3.4 Η δεκαετία του '80: ενοποίηση, ενότητες, επιδόσεις	- 14 -
1.3.5 Η δεκαετία του '90: η ηλικία του Διαδικτύου	- 15 -
1.3.6 Τρέχουσες τάσεις.....	- 16 -
1.4 Οι κορυφαίες γλώσσες προγραμματισμού.....	- 16 -
ΚΕΦΑΛΑΙΟ 2	- 19 -
2. Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΡΥΘΜΟΝ.....	- 19 -
Το λογότυπο	- 19 -
2.1 Η ΙΣΤΟΡΙΑ ΤΗΣ ΡΥΘΜΟΝ	- 20 -
2.2 ΒΑΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΗΣ ΡΥΘΜΟΝ.....	- 21 -
2.1.1 Φιλικότητα προς τους αρχαίους	- 21 -
2.1.2 Πολύ ευέλικτη	- 22 -
2.1.3 Ευνότητα.....	- 22 -
2.1.4 Απλή και εύκολη στη μάθηση.....	- 22 -
2.1.5 Ερμηνευμένη Γλώσσα.....	- 23 -
2.1.6 Γλώσσα πολλαπλών πλατφορμών.....	- 23 -
2.1.7 Δωρεάν και ανοιχτή πηγή	- 24 -
2.1.8 Αντικειμενοστρεφής γλώσσα	- 24 -

2.1.9 Εκτεταμένες Βιβλιοθήκες.....	- 24 -
2.1.10 Ολοκληρωμένη.....	- 25 -
2.1.11 Βάσεις δεδομένων Συνδεσιμότητα.....	- 25 -
2.2 Η ΕΓΚΑΤΑΣΤΑΣΗ ΤΗΣ ΡΥΘΗΟΝ.....	- 26 -
2.2.1 ΕΓΚΑΤΑΣΤΑΣΗ ΤΗΣ ΡΥΘΗΟΝ ΣΤΑ Windows	- 26 -
ΕΚΤΕΛΕΣΗ ΠΡΟΓΡΑΜΜΑΤΟΣ ΡΥΘΗΟΝ ΣΤΑ WINDOWS ΜΕ ΕΝΤΟΛΕΣ (<i>command line</i>)	- 29 -
Εκτέλεση προγράμματος μέσω <i>Idle</i>	- 30 -
2.2.2 ΕΓΚΑΤΑΣΤΑΣΗ ΤΗΣ ΡΥΘΗΟΝ ΣΕ LINUX	- 36 -
2.2.3 ΕΓΚΑΤΑΣΤΑΣΗ ΤΗΣ ΡΥΘΗΟΝ ΣΕ MAC	- 36 -
ΚΕΦΑΛΑΙΟ 3	- 37 -
3. ΕΝΣΩΜΑΤΩΜΕΝΟΙ ΤΥΠΟΙ ΕΝΤΟΛΩΝ ΣΤΗΝ ΡΥΘΗΟΝ	- 37 -
3.1 Ενότητα 1 ^η Έλεγχος για αλήθεια.....	- 37 -
3.2 Ενότητα 2 ^η Boolean Operatios- and, or not	- 38 -
3.3 Ενότητα 3 ^η Συγκρίσεις	- 39 -
3.4 Ενότητα 4 ^η Αριθμητικοί τύποι-int, float, long, complex.....	- 40 -
3.5 Ενότητα 5 ^η Διαδικές λειτουργίες σε τύπους ακεραίων	- 44 -
3.6 Ενότητα 6 ^η Τύποι Iterator.....	- 45 -
3.7 Ενότητα 7 ^η Οι γεννήτριες της Python	- 46 -
3.8 Ενότητα 8 ^η Αρχεία.....	- 46 -
3.9 Ενότητα 9 ^η Σχόλια.....	- 47 -
3.10 Ενότητα 10 ^η Συνάρτηση if.....	- 47 -
3.11 Ενότητα 11 ^η Τύποι ακολουθίας.....	- 47 -
<i>Λίστες</i>	- 47 -
<i>Συμβολοσειρές</i>	- 48 -
<i>Οι συμβολοσειρές Unicode</i>	- 48 -
3.12 Ενότητα 12 ^η Μέθοδοι στοιχειοσειράς.....	- 51 -
<i>str. decode ([κωδικοποίηση [, σφάλματα]])</i>	- 51 -
<i>str. encode ([κωδικοποίηση [, σφάλματα]])</i>	- 52 -
<i>str. endswith (επίθημα [, έναρξη [, τέλος]])</i>	- 52 -
<i>str. expandtabs ([tabsize])</i>	- 52 -
<i>str. find (υπο [, έναρξη [, τέλος]])</i>	- 53 -
<i>str. format (* args , ** kwargs)</i>	- 53 -
<i>str. isalnum ()</i>	- 53 -

ΚΕΦΑΛΑΙΟ 4	- 54 -
4. ΤΡΟΠΟΙ ΑΝΑΠΤΥΞΗΣ ΕΝΟΣ ΠΡΟΓΡΑΜΜΑΤΟΣ	- 54 -
4.1 Βήματα ανάπτυξης.....	- 54 -
4.1.1 Γνώση τεχνολογίας	- 54 -
4.1.2 Ορολογία	- 55 -
4.1.3 Γλώσσα προγραμματισμού.....	- 55 -
4.1.4 Επεξεργασία κώδικα	- 56 -
ΚΕΦΑΛΑΙΟ 5	- 57 -
5. ΑΝΑΛΥΣΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ ΡΥΘΜΩΝ	- 57 -
5.1 ΕΦΑΡΜΟΓΕΣ ΚΑΙ ΠΑΡΑΔΕΙΓΜΑΤΑ	- 57 -
5.1.1 Πρόγραμμα Python έκδοση 2.7.4.....	- 57 -
5.1.2 Πρώτη καρτέλα στο Πρόγραμμα Python File –Αρχείο.....	- 58 -
5.1.3 Δεύτερη καρτέλα στο Πρόγραμμα Python Edit.....	- 59 -
5.1.4 Τρίτη καρτέλα στο Πρόγραμμα Python Shell.....	- 59 -
5.1.5 Τέταρτη καρτέλα στο Πρόγραμμα Python Debug.....	- 60 -
5.1.6 Πέμπτη καρτέλα στο Πρόγραμμα Python Options	- 61 -
5.1.7 Έκτη καρτέλα στο Πρόγραμμα Python Windows	- 62 -
5.1.8 Έβδομη καρτέλα στο Πρόγραμμα Python Help.....	- 62 -
5.2.1 Παράδειγμα 1 <i>print</i>	- 63 -
5.2.2 Παράδειγμα 2 <i>print, int, str, error</i> , σχόλια.....	- 64 -
5.2.3 Παράδειγμα 3 <i>print x</i>	- 65 -
5.2.4 Παράδειγμα 4 <i>print, syntax error</i> , μεταβλητή	- 66 -
5.2.5 Παράδειγμα 5 <i>sum, print sum</i>	- 66 -
5.2.6 Παράδειγμα 6 , συνάρτηση <i>min, max</i>	- 67 -
5.2.7 Παράδειγμα 7 , μεταβλητή <i>-var</i>	- 68 -
5.2.8 Παράδειγμα 8 , μαθηματικές πράξεις	- 69 -
5.2.9 Παράδειγμα 9 , μαθηματικές πράξεις και λάθη	- 70 -
5.2.10 Παράδειγμα 10, λάθη	- 70 -
5.2.11 Παράδειγμα 11, συνάρτηση <i>if</i>	- 71 -
5.2.12 Παράδειγμα 12 συνάρτηση <i>if</i> και συνθήκη <i>True</i>	- 72 -
5.2.13 Παράδειγμα 13, συνάρτηση <i>range</i>	- 73 -
5.2.14 Παράδειγμα 14, συνάρτηση <i>for, in, range</i>	- 73 -
5.2.15 Παράδειγμα 15 συνθήκη αληθή ή ψευδή	- 74 -
5.2.16 Παράδειγμα 16, <i>translate</i>	- 75 -

5.2.17 Παράδειγμα 17, dict.....	- 76 -
5.2.18 Παράδειγμα 18 and, print	- 77 -
5.2.19 Παράδειγμα 19, συνάρτηση if, else.....	- 77 -
5.2.20 Παράδειγμα 20.....	- 78 -
5.2.21 Παράδειγμα 21 program to add two matrices using nested loop.....	- 80 -
5.2.22 Παράδειγμα 22 print my_string.....	- 81 -
5.2.23 Παράδειγμα 23 print string1, string2.....	- 82 -
5.2.24 Παράδειγμα 24 program to swap two variables.....	- 83 -
5.2.25 Παράδειγμα 25 find the factorial of a number.....	- 84 -
5.2.26 Παράδειγμα 26 print my_tuple.....	- 85 -
5.2.27 Παράδειγμα 27 class string, class tuple.....	- 87 -
5.2.28 Παράδειγμα 28 use of break statement inside loop	- 88 -
5.2.29 Παράδειγμα 29 my_dict 1	- 89 -
5.2.30 Παράδειγμα 30 print my_dict 2.....	- 89 -
5.2.31 Παράδειγμα 31 conversion factor.....	- 90 -
5.2.32 Παράδειγμα 32 convert temperature in Celsius to Fahrenheit	- 91 -
5.2.33 Παράδειγμα 33 program to calculate the square root.....	- 92 -
5.2.34 Παράδειγμα 34 solve the quadratic equation $ax^{**2}+bx+c=0$	- 93 -
5.2.35 Παράδειγμα 35 program to display calendar of given month of the year- 95 -	-
5.2.36 Παράδειγμα 36 program to find numbers divisible, list.....	- 96 -
5.2.37 Παράδειγμα 37 class def.....	- 97 -
5.2.38 Παράδειγμα 38 print, if, else, def, or.....	- 98 -
ΣΥΜΠΕΡΑΣΜΑΤΑ	- 100 -
<i>Εφαρμογές της Python</i>	- 100 -
<i>Οφέλη της Python</i>	- 100 -
ΠΕΡΙΕΧΟΜΕΝΑ ΠΙΝΑΚΩΝ	- 110 -
ΠΕΡΙΕΧΟΜΕΝΑ ΕΙΚΟΝΩΝ	- 111 -
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	- 113 -
ΕΛΛΗΝΙΚΗ ΒΙΒΛΙΟΓΡΑΦΙΑ.....	- 113 -
ΞΕΝΗ ΒΙΒΛΙΟΓΡΑΦΙΑ.....	- 115 -
ΙΣΤΟΣΕΛΙΔΕΣ.....	- 116 -

ΠΕΡΙΛΗΨΗ

Η εργασία που ακολουθεί, αναφέρεται στον προγραμματισμό της γλώσσας Python και αποτελείται από πέντε κεφάλαια. Στο πρώτο κεφάλαιο αναφέρονται γενικά στατιστικά και η ιστορία των γλωσσών προγραμματισμού, αντίστοιχα στο δεύτερο κεφάλαιο αρχίζει να αναπτύσσεται η γλώσσα Python και η εγκατάσταση της σε Windows Mac και Linux. Στο τρίτο κεφάλαιο γίνεται αναφορά σε εντολές και συναρτήσεις και μεταβλητές που υπάρχουν στον προγραμματισμό για τη γλώσσα Python. Στο τέταρτο κεφάλαιο γίνεται μία αναφορά στον τρόπο ανάπτυξης των προγραμμάτων. Τέλος στο πέμπτο κεφάλαιο υπάρχουν παραδείγματα από μικρά προγραμματάκια και εντολές στη γλώσσα Python.

ΕΙΣΑΓΩΓΗ

Η παρακάτω πτυχιακή εργασία με θέμα:

«Η γλώσσα προγραμματισμού Python – Τρόποι εκμάθησης», αποτελεί την ολοκλήρωση των σπουδών μου στο τμήμα «Διοίκηση Επιχειρήσεων» και στη σχολή «Διοίκηση και Οικονομία».

Το θέμα της έχει σχέση με τη γλώσσα προγραμματισμού Python. Τα κεφάλαια, που θα ακολουθήσουν είναι σχετικά με την εγκατάσταση και λειτουργία, καθώς και την κωδικοποίηση της συγκεκριμένης γλώσσας προγραμματισμού. Είναι δεδομένο ότι η τεχνολογία εξελίσσεται συνεχώς και ο προγραμματισμός, θεωρείται εξαιρετικά σπουδαίο επίτευγμα της τεχνολογίας και πολύ σημαντικός για να διεκπεραιωθούν πολλές λειτουργίες στην καθημερινή μας ζωή.

Ο προγραμματισμός είναι σημαντικός για την εκμάθηση της καινοτομίας, για τη δημιουργία φιλικών προς το περιβάλλον λύσεων για παγκόσμια προβλήματα. Ο προγραμματισμός είναι σημαντικός στην καθημερινή μας ζωή για την ενίσχυση και αύξηση της ισχύος των υπολογιστών και του διαδικτύου. Ο προγραμματισμός είναι σημαντικός για την επιτάχυνση των διαδικασιών εισόδου και εξόδου σε ένα μηχάνημα.

Ο προγραμματισμός είναι σημαντικός για την αυτοματοποίηση, τη συλλογή, τη διαχείριση, τον υπολογισμό, την ακριβή ανάλυση των δεδομένων και των πληροφοριών. Ο προγραμματισμός είναι σημαντικός για τη δημιουργία λογισμικού και εφαρμογών που βοηθούν τους χρήστες υπολογιστών και κινητών στην καθημερινή ζωή.

Λόγω όλων αυτών των λόγων, είναι πραγματικά σημαντικό να μάθουμε πώς να χρησιμοποιούμε γλώσσες προγραμματισμού στην καθημερινή μας ζωή. Για να δημιουργηθεί η παρούσα εργασία χρησιμοποιήθηκε βιβλιογραφία και ιντερνετικές πηγές. Όλα τα στοιχεία παρατίθενται στο τέλος της.

ΚΕΦΑΛΑΙΟ 1

1. ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ –ΔΙΑΚΡΙΣΗ – ΚΑΙ ΙΣΤΟΡΙΚΗ ΕΞΕΛΙΞΗ

1.1 ΟΡΙΣΜΟΣ ΓΛΩΣΣΑΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Μια γλώσσα προγραμματισμού είναι μια γλώσσα υπολογιστή σχεδιασμένη για να δημιουργήσει μια τυποποιημένη μορφή εντολών. Αυτές οι εντολές μπορούν να ερμηνευτούν σε έναν κώδικα κατανοητό από μια μηχανή. Τα προγράμματα δημιουργούνται μέσω γλωσσών προγραμματισμού για τον έλεγχο της συμπεριφοράς και της εξόδου ενός μηχανήματος μέσω ακριβών αλγορίθμων, παρόμοιων με τη διαδικασία ανθρώπινης επικοινωνίας. Μια γλώσσα προγραμματισμού είναι επίσης γνωστή ως σύστημα προγραμματισμού, γλώσσα υπολογιστή ή σύστημα υπολογιστή.¹

1.2 ΔΙΑΚΡΙΣΗ ΤΗΣ Γλώσσα Προγραμματισμού

Μια γλώσσα προγραμματισμού χωρίζεται συνήθως σε δύο στοιχεία: τη σύνταξη και τη σημασιολογία. Υπάρχει σχεδόν πάντα ένα έγγραφο προδιαγραφών για τον καθορισμό και των δύο στοιχείων. Για παράδειγμα,

¹ <http://www.ba.teiwest.gr/Nea%20Mathimata/c/LECTURE%2001.pdf>

ένα πρότυπο ISO ορίζει το C, ενώ το Perl έχει μια κυρίαρχη εφαρμογή που χρησιμοποιείται ως αναφορά.²

Παρουσιάζεται ένας αλγόριθμος χρησιμοποιώντας τη γλώσσα προγραμματισμού. Οι γλώσσες προγραμματισμού ονομάζονται συνήθως γλώσσες υπολογιστών.

Ωστόσο, ορισμένοι συγγραφείς θεωρούν ότι οι γλώσσες προγραμματισμού αποτελούν υποσύνολο γλωσσών υπολογιστών. Από τις παλαιότερες μορφές γλωσσών προγραμματισμού όπως το COBOL και το FORTRAN, έχουν αναπτυχθεί χιλιάδες γλώσσες υπολογιστών.³

1.3 Ιστορία των γλωσσών προγραμματισμού

1.3.1 Πριν από το 1940

Οι πρώτες γλώσσες προγραμματισμού προηγούνται του σύγχρονου υπολογιστή. Αρχικά, οι γλώσσες ήταν κωδικοί. Κατά τη χρονική περίοδο 1842-1843, μεταφράστηκαν από τον Ιταλό μαθηματικό Luigi, τα απομνημονεύματα της Menabrea. Το νεότερο προτεινόμενο μηχάνημα του Charles Babbage, ήταν ο Αναλυτικός κινητήρας. Επισύναψε ένα σύνολο σημειώσεων που διευκρίνιζε λεπτομερώς μια μέθοδο υπολογισμού. Ο Bernoulli αριθμεί με τον κινητήρα, αναγνωρισμένο από μερικούς ιστορικούς ως τον πρώτο υπολογιστή στον κόσμο πρόγραμμα.⁴

Ο Herman Hollerith συνειδητοποίησε ότι μπορούσε να κωδικοποιήσει τις πληροφορίες χαρτιών, όταν το παρατήρησε αυτό οι αγωγοί αμαξοστοιχίας θα κωδικοποιούσαν την εμφάνιση των κατόχων εισιτηρίων στα εισιτήρια τρένων χρησιμοποιώντας το θέση των διάτρητων οπών στα εισιτήρια. Στη συνέχεια ο Hollerith προχώρησε στην κωδικοποίηση των δεδομένων απογραφής του 1890 στις κάρτες διάτρησης. Οι πρώτοι κωδικοί υπολογιστών εξειδικεύτηκαν για τις εφαρμογές. Στις πρώτες δεκαετίες του εικοστού αιώνα, οι αριθμητικοί υπολογισμοί βασίστηκαν σε δεκαδικά ψηφία. Τελικά το συνειδητοποίησε αυτό

² <http://pages.cs.aueb.gr/courses/epl131/files/K6-Γλώσσες%20Προγραμματισμού-2013.pdf>

³ <https://www.techopedia.com/definition/24815/programming-language&prev=search>

⁴ https://the-eye.eu/public/Books/IT%20Various/history_of_programming_languages.pdf

η λογική θα μπορούσε να αναπαρασταθεί με αριθμούς, καθώς και με λέξεις. Παράδειγμα, που χρησιμοποιείται στις τηλεφωνικές εταιρείες.

1.3.2 Η δεκαετία του 1940

Τη δεκαετία του 1940 δημιουργήθηκαν οι πρώτοι αναγνωρίσιμα σύγχρονοι ηλεκτρικά υπολογιστές. Η περιορισμένη ταχύτητα και η χωρητικότητα μνήμης ανάγκασαν τους προγραμματιστές να γράψουν χειροκίνητη συμβολική γλώσσα προγραμμάτων. Σύντομα ανακαλύφθηκε, ότι ο προγραμματισμός στη γλώσσα συναρμολόγησης απαιτούσε μεγάλη πνευματική προσπάθεια και ήταν επιρρεπής σε σφάλματα.

Το 1948, ο Konrad Zuse δημοσίευσε ένα άρθρο σχετικά με τη γλώσσα προγραμματισμού του Plankalkül. Ωστόσο, δεν εφαρμόστηκε στην εποχή του και οι αρχικές συνεισφορές του απομονώθηκαν άλλες εξελίξεις. Ορισμένες σημαντικές γλώσσες, που αναπτύχθηκαν κατά την περίοδο αυτή περιλαμβάνουν:

- 1943 - Plankalkül
- 1943 - Σύστημα κωδικοποίησης ENIAC
- 1949 - C-10H δεκαετία του 1950 και του 1960⁵

Στη δεκαετία του 1950, σχεδιάστηκαν οι τρεις πρώτες σύγχρονες γλώσσες προγραμματισμού, των οποίων οι απόγονοι βρίσκονται ακόμα στο ευρεία χρήση σήμερα. Αυτές είναι:

- FORTRAN (1955), το "FOR mula TRAN slator", που εφευρέθηκε από το John Backus.
- LISP , το "LISt Processor", που εφευρέθηκε από τον John McCarthy.
- COBOL , η ΕΝΙΣΧΥΣΗ ΕΙΣΟΔΟΥ ΣΤΗΝ ΕΝΕΡΓΕΙΑΚΗ ΑΓΩΓΗ.

Ένα άλλο ορόσημο στα τέλη της δεκαετίας του 1950 ήταν η έκδοση, από επιτροπή αμερικανικών και ξένων Ευρωπαίων επιστημόνων υπολογιστών, μιας «νέας γλώσσας για αλγορίθμους».

- 1952 - Autocode
- 1954 - FORTRAN

⁵ https://repository.kallipos.gr/bitstream/11419/778/1/lpTechniques_Chapter1.pdf

- 1955 - FLOW-MATIC (πρόδρομος του COBOL)
- 1957 - COMTRAN (πρόδρομος στο COBOL)
- 1958 - ALGOL 58
- 1959 - FACT (πρόδρομος στο COBOL)
- 1959 - COBOL
- 1962 - APL
- 1962 - Simula
- 1964 - PL / I⁶

1.3.3 1967-1978: καθιέρωση θεμελιωδών παραδειγμάτων

Η περίοδος από τα τέλη της δεκαετίας του '60 έως τα τέλη της δεκαετίας του '70 έφερε μια μεγάλη ανθοφορία στις γλώσσες προγραμματισμού. Τα περισσότερα από τα κύρια παραδείγματα γλωσσών που χρησιμοποιούνται σήμερα εφευρέθηκαν στην περίοδο αυτή:

- Simula, που εφευρέθηκε στα τέλη της δεκαετίας του 1960 από τον Nygaard και Dahl ως υπερέκδο του Algol 60, ήταν η πρώτη γλώσσα σχεδιασμένη να υποστηρίζει αντικειμενοστραφής προγραμματισμός.
- C, νωρίς γλώσσα προγραμματισμού συστημάτων, αναπτύχθηκε από την Dennis Ritchie και Γνώση Thompson στο Bell Labs μεταξύ 1969 και 1973.
- Το Smalltalk (στα μέσα της δεκαετίας του 1970) παρείχε ένα ολοκληρωμένο σχεδιασμό βασισμένο σε αντικείμενα Γλώσσα.
- Prolog , που σχεδιάστηκε το 1972 από τον Colmerauer, Roussel, και Kowalski , ήταν ο πρώτος λογική γλώσσα προγραμματισμού.
- Το ML δημιούργησε ένα σύστημα πολυμορφικού τύπου (εφευρέθηκε από το Robin Milner το 1973) στην κορυφή του Lisp, πρωτοποριακό στατικά πληκτρολογούμενες λειτουργικές γλώσσες προγραμματισμού.

Η δεκαετία του 1960 και της δεκαετίας του 1970 παρουσίασε επίσης σημαντική συζήτηση σχετικά με τα πλεονεκτήματα της «δομημένο

⁶ <http://eclass.sch.gr/modules/document/file.php/EL19107/ΑΕΠΠ%20-%20Βιβλίο%20-%20Τετράδιο%20Μαθητή/kef6.pdf>

προγραμματισμό», που ουσιαστικά σήμαινε τον προγραμματισμό χωρίς τη χρήση του go to.⁷

Ορισμένες σημαντικές γλώσσες που αναπτύχθηκαν κατά την περίοδο αυτή περιλαμβάνουν:

- 1970 - Pascal
- 1972 - Prolog
- 1973 - ML
- •1978- SQL (αρχικά μόνο μια γλώσσα ερωτήματος, που αργότερα επεκτάθηκε με δομές προγραμματισμού)⁸

1.3.4 Η δεκαετία του '80: ενοποίηση, ενότητες, επιδόσεις

Η δεκαετία του 1980 ήταν σχετικής ενοποίησης. Η C ++ συνδυασμένη με αντικειμενοστραφή συστήματα προγραμματισμού. Στην Ιαπωνία και αλλού, δαπανήθηκαν τεράστια ποσά διερευνώντας τα λεγόμενα γλώσσες προγραμματισμού πέμπτης γενιάς, που ενσωμάτωσαν τη λογική προγραμματισμός κατασκευών. Η κοινότητα λειτουργικών γλωσσών μεταφέρθηκε στην τυποποίηση ML και Lisp.⁹

Αντί να εφεύρουν νέα παραδείγματα, όλες αυτές οι κινήσεις επεξεργάστηκαν τις εφευρεθείσες ιδέες κατά την προηγούμενη δεκαετία.

Ωστόσο, μια σημαντική νέα τάση στο γλωσσικό σχεδιασμό ήταν η αυξημένη εστίαση στον προγραμματισμό μεγάλης κλίμακας συστήματα μέσω της χρήσης *ενοτήτων*, ή μεγάλης κλίμακας οργανωτικές μονάδες κώδικα.

Αν και δεν εμφανίστηκαν σημαντικά νέα παραδείγματα για τις γλώσσες προγραμματισμού, πολλοί ερευνητές επεκτάθηκε στις ιδέες των προηγούμενων γλωσσών και τους προσαρμοσε σε νέα πλαίσια. Η δεκαετία του 1980 έφερε επίσης πρόοδο στην εφαρμογή της γλώσσας προγραμματισμού. Η τεχνολογία των γλωσσών συνεχίστηκε με αυτές τις κατευθύνσεις και στη δεκαετία του 1990.

⁷ https://repository.kallipos.gr/bitstream/11419/778/1/lpTechniques_Chapter1.pdf

⁸ <http://pliroforikiatschool.blogspot.com/2013/06/infographic.html>

⁹ Κουρκουμπέτης, διαφάνειες Κεφάλαιο 6 γλώσσες προγραμματισμού (pdf)

Ορισμένες σημαντικές γλώσσες που αναπτύχθηκαν κατά την περίοδο αυτή περιλαμβάνουν:

- 1983 - Ada
- 1983 - C ++
- 1985 - Eiffel
- 1987 - Perl
- 1989 - FL (Backus)¹⁰

1.3.5 Η δεκαετία του '90: η ηλικία του Διαδικτύου

Η δεκαετία του 1990 δεν έβλεπε καμία θεμελιώδη καινοτομία, αλλά πολύ ανασυνδυασμό, καθώς και ωρίμανση παλαιών ιδέες. Μια μεγάλη φιλοσοφία οδήγησης ήταν η παραγωγικότητα του προγραμματιστή. Πολλές "γρήγορες εφαρμογές ανάπτυξης "(RAD) γλώσσες που προέκυψαν, η οποία συνήθως έφερε ένα IDE , συλλογή απορριμμάτων , και ήταν απόγονοι παλαιότερων γλωσσών.

Όλες αυτές οι γλώσσες ήταν αντικειμενοστραφής. Αυτά περιλαμβάνονται Object Pascal , Visual Basic και C # .

Ορισμένες σημαντικές γλώσσες που αναπτύχθηκαν κατά την περίοδο αυτή περιλαμβάνουν:

- 1991 - Python
- 1991 - Java
- 1994 - ANSI Common Lisp
- 1995 - JavaScript
- 1995 - PHP
- 2000 - NTO#
- 2008 - JavaFX Script¹¹

¹⁰ <http://www.ba.teiwest.gr/Nea%20Mathimata/c/LECTURE%2001.pdf>

¹¹ <http://pliroforikiatschool.blogspot.com/2013/06/infographic.html>

1.3.6 Τρέχουσες τάσεις

Η εξέλιξη της γλώσσας προγραμματισμού συνεχίζεται, τόσο στη βιομηχανία όσο και στην έρευνα. Μερικές από τις τρέχουσες τάσεις περιλαμβάνουν:

- Μηχανισμούς για την προσθήκη επαλήθευσης ασφάλειας και αξιοπιστίας στη γλώσσα με εκτεταμένο στατικό έλεγχο, και έλεγχο ροής πληροφοριών.
- Εναλλακτικούς μηχανισμούς για modularity.
- Ανάπτυξη λογισμικού με γνώμονα τα στοιχεία.
- Μεταπρογραμματισμός
- Μεγαλύτερη έμφαση στη διανομή και την κινητικότητα.
- Ενσωμάτωση με βάσεις δεδομένων, συμπεριλαμβανομένων των XML και σχεσιακές βάσεις δεδομένων .
- Υποστήριξη για Unicode έτσι ώστε ο πηγαίος κώδικας (κείμενο προγράμματος) δεν περιορίζεται στους χαρακτήρες, επιτρέποντας, για παράδειγμα, τη χρήση μη-λατινικών,
- που βασίζονται σε δέσμες ενεργειών
- ή παρατεταμένη σίξη.
- XML για γραφική διεπαφή (XUL , XAML).¹²

1.4 Οι κορυφαίες γλώσσες προγραμματισμού

Ο ανοιχτός κώδικας αποτελεί το πρότυπο της βιομηχανίας για την πρακτική ανάπτυξη λογισμικού, και η εμπορική επιτυχία του οφείλεται εν μέρει στην υιοθέτηση από μεγάλες εταιρείες τεχνολογίας. Πριν από λίγους μήνες, ο ιστότοπος αναζήτησης εργασίας Indeed στις 14 Μαρτίου 2018 ανακοίνωσε την συμμετοχή του στο Cloud Native Computing Foundation (CNCF), και δημοσίευσε μια έρευνα για τις πιο δημοφιλείς γλώσσες προγραμματισμού και δεξιότητες που αναζητούν οι εταιρίες που αναπτύσσουν και χρησιμοποιούν ανοιχτό κώδικα.

¹² http://thecorememory.com/Programming_Languages_-_Sammet.pdf



Εικόνα 1 Open Source

13

Εδώ λοιπόν, σύμφωνα με την έρευνα είναι οι κορυφαίες 10 γλώσσες προγραμματισμού και οι δεξιότητες σε υψηλή ζήτηση, σύμφωνα με την Indeed:

- 1. Java**
- 2. Python**
- 3. Git**
- 4. JavaScript**
- 5. Node.js**

¹³ <https://opensource.ellak.gr/2018/03/14/i-korifees-10-glosses-programmatismou-ke-dexiotites-pou-chriazeste-gia-na-ergastite-se-open-source/>

- 6. Docker**
- 7. AngularJS**
- 8. Jenkins**
- 9. Amazon Web Services (AWS)**
- 10. Agile¹⁴**

¹⁴ <https://www.techrepublic.com>

ΚΕΦΑΛΑΙΟ 2

2. Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΡΥΘΟΝ

Η Python είναι μια γλώσσα γενικής χρήσης, που σημαίνει ότι μπορεί να χρησιμοποιηθεί για την κατασκευή σχεδόν οτιδήποτε, το οποίο θα γίνει εύκολα με τα σωστά εργαλεία και βιβλιοθήκες. Η Python είναι ιδανική για ανάπτυξη ιστοσελίδων, ανάλυση δεδομένων, τεχνητή νοημοσύνη και επιστημονική πληροφορική.

Πολλοί προγραμματιστές έχουν επίσης χρησιμοποιήσει την Python για την ανάπτυξη εργαλείων παραγωγικότητας, παιχνιδιών και εφαρμογών για υπολογιστές.¹⁵

Το λογότυπο



16

¹⁵ <https://www.python.org/psf/about/>

¹⁶ <https://www.python.org/community/logos/>

2.1 Η ΙΣΤΟΡΙΑ ΤΗΣ PYTHON

Η πρώτη έκδοση της Python (δηλαδή η Python 1.0) εισήχθη το 1991. Από την έναρξή της και την εισαγωγή της έκδοσης 1, η εξέλιξη της Python έφτασε μέχρι την έκδοση 3.x (μέχρι το 2017).¹⁷

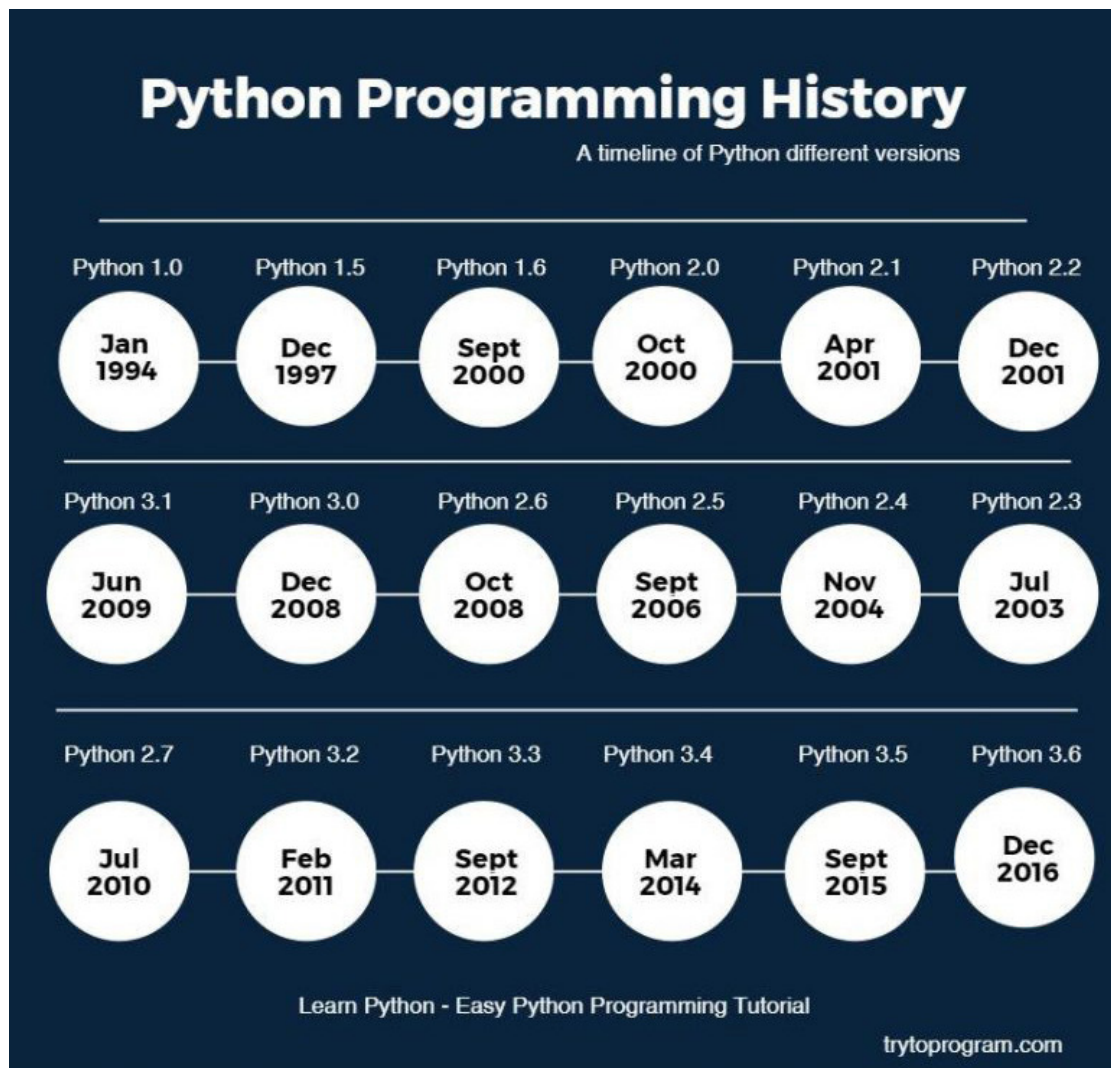
Ο Guido van Rossum, ο δημιουργός της γλώσσας Python, ονόμασε τη γλώσσα από την εκπομπή "Monty Python's Flying Circus" του BBC. Η Python 2.0 κυκλοφόρησε το 2000. Το 2008 κυκλοφόρησε η έκδοση 3.0. Πολλά από τα καινούργια χαρακτηριστικά αυτής της έκδοσης έχουν μεταφερθεί στις εκδόσεις 2.6 και 2.7, που είναι προς τα πίσω συμβατές. Η Python 3 είναι ιστορικά η πρώτη γλώσσα προγραμματισμού που σπάει την προς τα πίσω συμβατότητα με προηγούμενες εκδόσεις.^{18, 19}

Ακολουθεί το σύντομο διάγραμμα που απεικονίζει το χρονοδιάγραμμα της έκδοσης διαφορετικών εκδόσεων της γλώσσας προγραμματισμού Python.

¹⁷ <https://eclass.upatras.gr/modules/document/file.php/ECON1332/LAB-Intro-Python.pdf>

¹⁸ <https://repository.kallipos.gr/bitstream/11419/2745/2/ALL-KOY.pdf>

¹⁹ <https://startupper.gr/h-python-είναι-η-πιο-δημοφιλής-γλώσσα-στις-αν/>



Εικόνα 2 το χρονοδιάγραμμα της python

20

2.2 ΒΑΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΗΣ PYTHON

2.1.1. Φιλικότητα προς τους αρχαίους

Η Python σχεδιάστηκε για να είναι εύκολη στην κατανόηση και τη διασκέδαση στη χρήση (το όνομά της προέρχεται από τον Monty Python. Η διασκέδαση είναι ένα σπουδαίο κίνητρο και αφού έχουν τη δυνατότητα να δημιουργούν γρήγορα πρωτότυπα και εργαλεία με την Python, πολλοί βρίσκουν στην Python την κωδικοποίηση ικανοποιητική εμπειρία. Η εξαιρετική συντακτική απλότητα της Python, επιτρέπει στον αρχάριο να χρησιμοποιεί βασικές ή προηγμένες έννοιες προγραμματισμού χωρίς πολύ κωδικό λέβητα όπως οι

²⁰ <http://www.trytoprogram.com/python-programming/history-of-python/>

περισσότερες άλλες γλώσσες, για παράδειγμα για να εμφανίζει την έξοδο στην οθόνη είναι μία γραμμή σε σύγκριση με τις περισσότερες άλλες γλώσσες που απαιτούν πολύ περισσότερα “setup”.²¹

Έτσι, η Python έχει αποκτήσει δημοτικότητα ως γλώσσα φιλική προς τους αρχάριους και έχει αντικαταστήσει την Java ως την πιο δημοφιλή εισαγωγική γλώσσα στα κορυφαία πανεπιστήμια των ΗΠΑ και της Ευρώπης.

2.1.2 Πολύ ευέλικτη

Ως γλώσσα δυναμικά πληκτρολογημένη, η Python είναι πραγματικά ευέλικτη. Αυτό σημαίνει ότι δεν υπάρχουν σκληροί κανόνες για τον τρόπο κατασκευής χαρακτηριστικών και υπάρχει μεγαλύτερη ευελιξία για την επίλυση προβλημάτων χρησιμοποιώντας διαφορετικές μεθόδους (αν και η φιλοσοφία της Python ενθαρρύνει τη χρήση του προφανή τρόπο για την επίλυση των πραγμάτων). Επιπλέον, η Python είναι επίσης πιο επιεικής στα σφάλματα, έτσι ώστε οι χρήστες μπορούν να είναι σε θέση να καταρτίζονται και να εκτελούν το πρόγραμμά τους μέχρι να διορθώσουν το προβληματικό μέρος.

2.1.3 Ευνόητη

Όντας μια γλώσσα πολύ υψηλού επιπέδου, Η Python επιτρέπει στους αρχάριους να επικεντρωθούν στην εκμάθηση των εννοιών του προγραμματισμού και να μην χρειάζεται να ανησυχούν για πάρα πολλές λεπτομέρειες, δεδομένου ότι μπορούν να τη χρησιμοποιήσουν από τα αρχικά τους κίονες βήματα.²²

2.1.4 Απλή και εύκολη στη μάθηση

Η Python είναι εξαιρετικά εύκολη για να ξεκινήσει κάποιος τον προγραμματισμό. Προσφέρει εύκολη κατανόηση της σύνταξης, απλή

²¹ <https://eclass.upatras.gr/modules/document/file.php/ECON1362/Διαλέξεις-AM/5.1%20Python%20Γενικά-Υπολογισμοί%20%26%20μεταβλητές.pdf>

²² https://arch.icte.uowm.gr/docs/Mini_Python_Notebook.pdf

εγκατάσταση και πολλές πρακτικές εφαρμογές στην ανάπτυξη ιστού. Η σύνταξη δεν είναι υπερβολικά ενοχλητική σε σύγκριση με άλλες γλώσσες και μπορεί ο χρήστης να εισάγει μια δέσμη ενοτήτων, που συχνά μπορούν να κάνουν τον κώδικά του πολύ μικρότερο. Υπάρχουν άριστα και απλά εργαλεία για να εργαστεί με τον κώδικα Python, ειδικά τον διαδραστικό διερμηνέα. Δεν χρειάζεται να μάθει κάποιος χρήστης, ένα σύστημα δημιουργίας, IDE, ειδικό πρόγραμμα επεξεργασίας κειμένου ή οτιδήποτε άλλο για να αρχίσει να χρησιμοποιεί το Python. Το μόνο που χρειάζεται είναι μόνο μια γραμμή εντολών και ο διαδραστικός επεξεργαστής.

2.1.5 Ερμηνευμένη Γλώσσα

Η Python είναι μια γλώσσα που ερμηνεύεται, δηλαδή ο διερμηνέας εκτελεί την γραμμή κώδικα ανά γραμμή τη φορά. Όταν χρησιμοποιεί ο χρήστης μια γλώσσα που έχει ερμηνευτεί όπως η Python, δεν υπάρχουν ξεχωριστά βήματα σύνταξης και εκτέλεσης. Απλώς τρέχει το πρόγραμμα από τον πηγαίο κώδικα. Αυτό καθιστά εύκολο τον εντοπισμό σφαλμάτων και έτσι είναι κατάλληλο για αρχάριους. Εσωτερικά, η Python μετατρέπει τον πηγαίο κώδικα σε μια ενδιάμεση μορφή που ονομάζεται `bytecodes` και στη συνέχεια μεταφράζει αυτό στη μητρική γλώσσα του συγκεκριμένου υπολογιστή και στη συνέχεια το τρέχει. Απλώς τρέχει τα προγράμματά και ποτέ δεν χρειάζεται να ανησυχεί κάποιος για τη σύνδεση και φόρτωση με βιβλιοθήκες κ.λπ.²³

2.1.6 Γλώσσα πολλαπλών πλατφορμών

Η Python μπορεί να τρέχει εξίσου σε διαφορετικές πλατφόρμες όπως τα Windows, Linux, Unix, Macintosh κλπ. Ένα πρόγραμμα Python γραμμένο σε έναν υπολογιστή Macintosh θα τρέξει σε ένα σύστημα Linux και αντίστροφα. Έτσι, η Python είναι μια φορητή γλώσσα .

²³ http://python.org.gr/phocadownload/Tutorials/tutorial_by_example.pdf

2.1.7 Δωρεάν και ανοιχτή πηγή

Η Python είναι ένα παράδειγμα ενός FLOSS (Ελεύθερου Λογισμικού και Λογισμικού Ανοικτού Κώδικα). Με απλά λόγια, μπορεί κάποιος χρήστης να διανείμει ελεύθερα αντίγραφα αυτού του λογισμικού, να διαβάσει τον πηγαίο κώδικα του λογισμικού, να κάνει αλλαγές σε αυτόν, να χρησιμοποιήσει κάποιος χρήστης, κομμάτια του σε νέα δωρεάν προγράμματα και είναι γνωστό, ότι μπορεί να κάνει αυτά τα πράγματα. Η γλώσσα Python είναι ελεύθερα διαθέσιμη στη διεύθυνση www.python.org . Ο πηγαίος κώδικας είναι επίσης διαθέσιμος. Ως εκ τούτου, είναι ανοικτού κώδικα.²⁴

2.1.8 Αντικειμενοστρεφής γλώσσα

Η Python υποστηρίζει χαρακτηριστικά αντικειμενοστραφή . Σε σύγκριση με άλλες γλώσσες προγραμματισμού, ο μηχανισμός κλάσης της Python προσθέτει τάξεις με ελάχιστη σύνταξη και σημασιολογία. Πρόκειται για ένα μείγμα των μηχανισμών κατηγορίας που βρέθηκαν σε C ++ και Modula-3 . Οι κλάσεις Python παρέχουν όλα τα τυπικά χαρακτηριστικά της αντικειμενοστρεφούς γλώσσας προγραμματισμού εκτός από την ισχυρή ενσωμάτωση , η οποία είναι μόνο ένα από τα πολλά χαρακτηριστικά που σχετίζονται με τον όρο "αντικειμενοστρεφές".²⁵

2.1.9 Εκτεταμένες Βιβλιοθήκες

Η Πλανητική Βιβλιοθήκη Python είναι πραγματικά τεράστια. Η βιβλιοθήκη Python περιέχει ενσωματωμένες ενότητες (γραμμένες σε C) που παρέχουν πρόσβαση στη λειτουργικότητα του συστήματος, όπως το αρχείο I / O που

²⁴ http://python.org.gr/phocadownload/Tutorials/tutorial_by_example.pdf

²⁵ <https://slideplayer.gr/slide/2309211/>

διαφορετικά θα ήταν απρόσιτο για τους προγραμματιστές της Python, καθώς και μονάδες γραμμένες στη Python που παρέχουν τυποποιημένες λύσεις για πολλά προβλήματα που εμφανίζονται καθημερινό προγραμματισμό. Μπορεί να βοηθήσει το χρήστη, να κάνει διάφορα πράγματα που περιλαμβάνουν κανονικές εκφράσεις, παραγωγή τεκμηρίωσης, δοκιμές μονάδων, σπειρώματα , βάσεις δεδομένων κλπ.²⁶

2.1.10 Ολοκληρωμένη

Η Python μπορεί εύκολα να ενσωματωθεί σε γλώσσες όπως C, C ++, JAVA κλπ., με αρκετά εύκολο και προσιτό τρόπο.

2.1.11 Βάσεις δεδομένων Συνδεσιμότητα

Η Python παρέχει διεπαφή σε όλες τις εμπορικές βάσεις δεδομένων.²⁷



28

Εικόνα 3 Τομείς που μπορεί να χρησιμοποιηθεί η Python

²⁶ www.python.org

²⁷ <https://repository.kallipos.gr/bitstream/11419/2745/2/ALL-KOY.pdf>

²⁸ <http://www.bestprogramminglanguagefor.me/why-learn-python>

2.2 Η ΕΓΚΑΤΑΣΤΑΣΗ ΤΗΣ PYTHON

Εγκατάσταση της Python Η Python μπορεί να χρησιμοποιηθεί σε 21 διαφορετικά Λειτουργικά Συστήματα και περιβάλλοντα. Η διαδικασία εγκατάστασης είναι απλή.

2.2.1 ΕΓΚΑΤΑΣΤΑΣΗ ΤΗΣ PYTHON ΣΤΑ Windows

1. Επισκεπτόμαστε τον ιστότοπο της Python: www.python.org
2. Επιλέγουμε τον σύνδεσμο Downloads και κατεβάζουμε το νεότερο αρχείο εγκατάστασης της σειράς 3.x για Windows. Μπορούμε να διαλέξουμε μεταξύ 32-bit και 64-bit εγκατάστασης
3. Εκτελούμε το αρχείο της εγκατάστασης.

Εγκατάσταση Python σε Windows

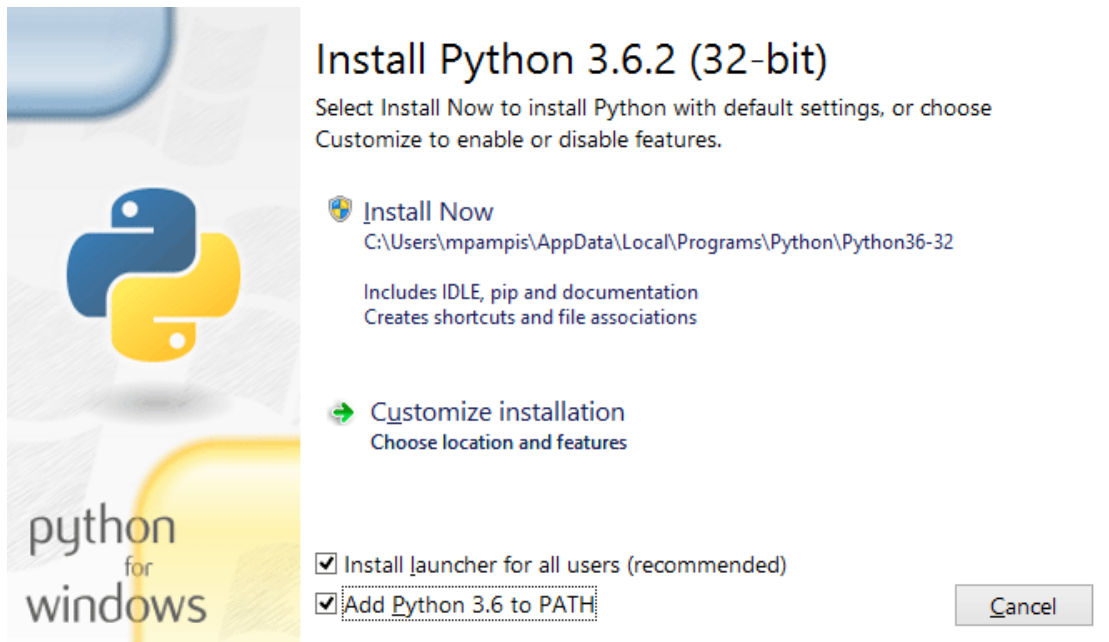
- Το κατεβάζουμε εύκολα από την επίσημη ιστοσελίδα www.python.org/download
- Επιλέγουμε την κατάλληλη έκδοση ανάλογα με το σύστημά μας.
- Μετά την εγκατάσταση-> path %path%; [C:\Python](#) (command prompt)



Εικόνα 4 Εγκατάσταση στα Windows

29

²⁹<https://www.google.gr/imgres?imgurl=https%3A%2F%2Fslideplayer.gr%2Fslide%2F2309211%2F8%2Fimages%2F13%2F%25CE%2595%25CE%25B3%25CE%25BA%25CE%25B1%25CF%2584%25CE%25AC%25CF%2583%25CF%2584%25CE%25B1%25CF%2583%25CE%25B7%2BPython%2B%25CF%2583%25CE%25B5%2BWindows.jpg&imgrefurl=https%3A%2F%2Fslideplayer.gr%2Fslide%2F2309211%2F&docid=T6cYGIRPPyDEM&tbnid=VA9rxo323e1u0M%3A&vet=10ahUKEwjGzYzv95nfAhWk4sKHX1jBXoQMwhOKBEwEQ..i&w=960&h=720&client=opera&bih=674&biw=1462&q=εγκατάσταση%20της%20python&ved=0ahUKEwjGzYzv95nfAhWk4sKHX1jBXoQMwhOKBEwEQ&iact=src&uact=8>



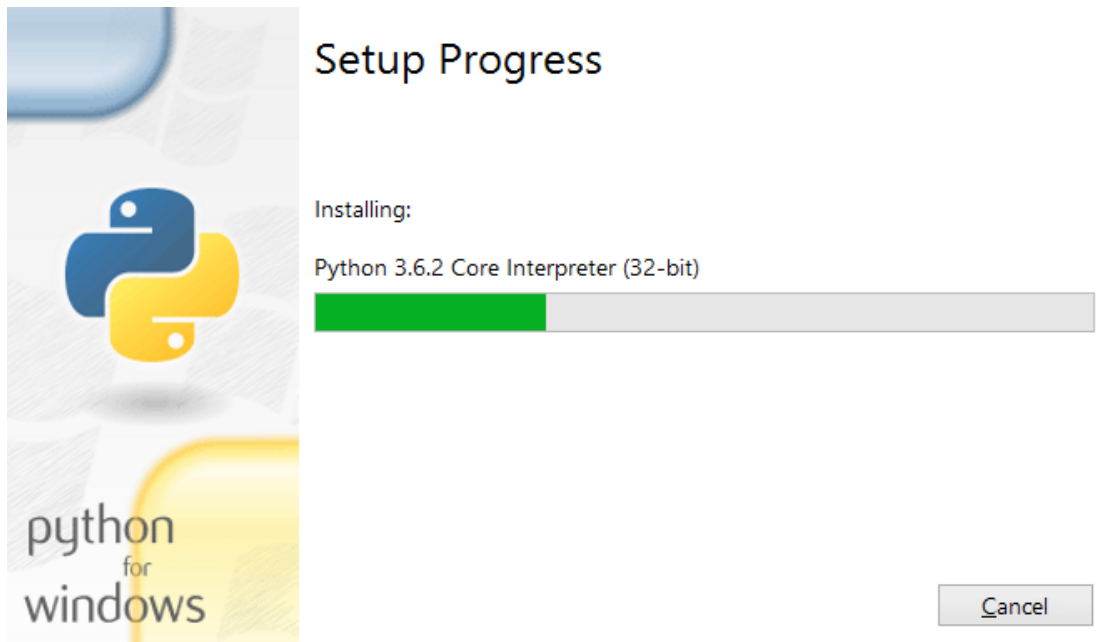
Εικόνα 5 Install Python

30

Επιλέγουμε είτε "Install Now" είτε "Customize installation" αν θέλουμε να μεταβάλλουμε την τοποθεσία της εγκατάστασης, προσέχοντας σε κάθε περίπτωση να κάνουμε τσεκ και την επιλογή για προσθήκη της Python στο PATH (κάτι που παλαιότερα έπρεπε να κάνουμε χειροκίνητα).

Το PATH είναι μια μεταβλητή συστήματος στα Windows. Προσθέτοντας την Python, ενημερώνουμε ουσιαστικά τη γραμμή εντολών σε ποιους φακέλους πρέπει να κοιτάει όταν επιχειρεί να εντοπίσει ένα αρχείο.

³⁰ <https://www.pcsteps.gr/221565-μαθήματα-προγραμματισμού-python-1/>



Εικόνα 6 Πρόοδος install setup Python

31

Με την ολοκλήρωση της εγκατάστασης, έχουμε στο σύστημά μας τον interpreter της Python, μαζί με μερικά ακόμα χρήσιμα εργαλεία.

ΕΚΤΕΛΕΣΗ ΠΡΟΓΡΑΜΜΑΤΟΣ PYTHON ΣΤΑ WINDOWS ΜΕ ΕΝΤΟΛΕΣ (command line)

Πλέον μπορούμε να επιχειρήσουμε να τρέξουμε τον πρώτο μας κώδικα. Έχουμε τη δυνατότητα να το κάνουμε, είτε με τη χρήση της γραμμής εντολών των Windows, είτε μέσα από τον IDLE της Python.

Πατώντας στην έναρξη, μπορούμε να πληκτρολογήσουμε "cmd" για να ανοίξουμε την κλασική γραμμή εντολών των Windows (ή "powershell" για το Powershell αντίστοιχα).

Προκειμένου να ελέγξουμε αν όλα έχουν πάει καλά με την εγκατάσταση, γράφουμε:

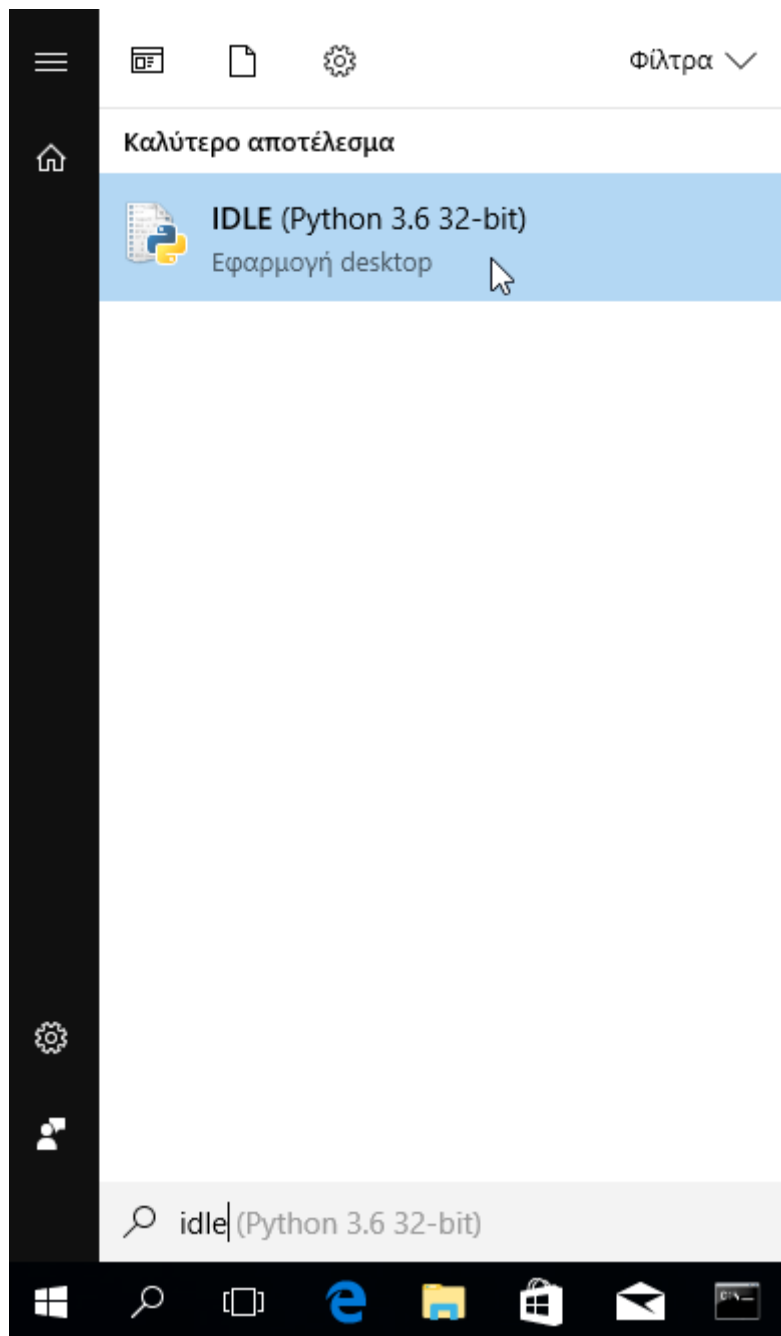
```
python
```

³¹ <https://www.pcsteps.gr/221565-μαθήματα-προγραμματισμού-python-1/>

Θα εμφανιστούν οι πληροφορίες της έκδοσής μας, ενώ ταυτόχρονα εισερχόμαστε στο περιβάλλον του interpreter.

Εκτέλεση προγράμματος μέσω Idle

Εναλλακτικά, πηγαίνοντας στην έναρξη μπορούμε να πληκτρολογήσουμε "IDLE" για να ανοίξουμε το λεγόμενο Python Shell.



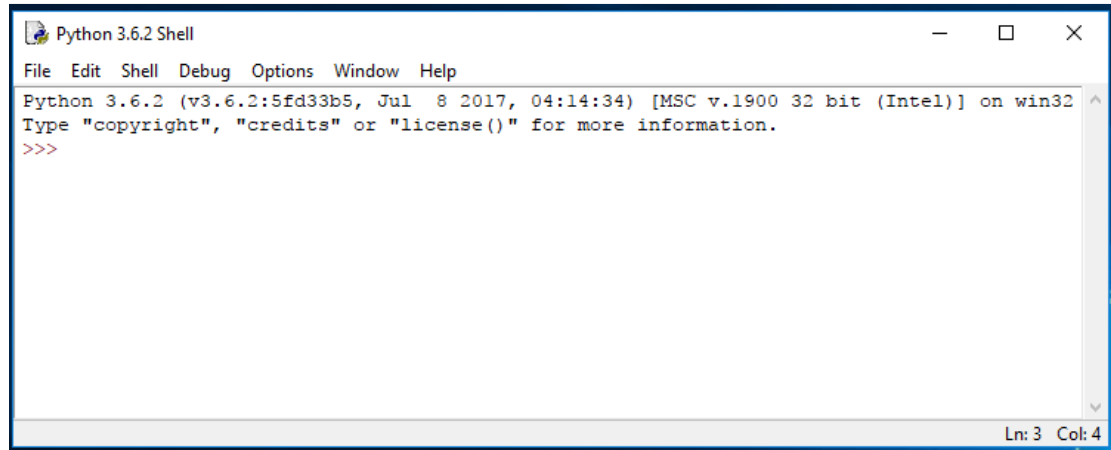
Εικόνα 7 Idle

32

Το IDLE (Integrated Development & Learning Environment) είναι ένα μινιμαλιστικό περιβάλλον ανάπτυξης (IDLE) που έρχεται από προεπιλογή μαζί με την εγκατάσταση του πακέτου της Python.

³² <https://www.pcsteps.gr/221565-μαθήματα-προγραμματισμού-python-1/>

Είναι γραμμένο αποκλειστικά με χρήση της Python και του πακέτου GUI Tkinter. Είναι cross-platform - δηλαδή έχει εκδόσεις για διαφορετικά λειτουργικά συστήματα - και, λόγω της απλότητάς του, είναι κατάλληλο για εκπαιδευτικούς σκοπούς.



Εικόνα 8 Python Shell

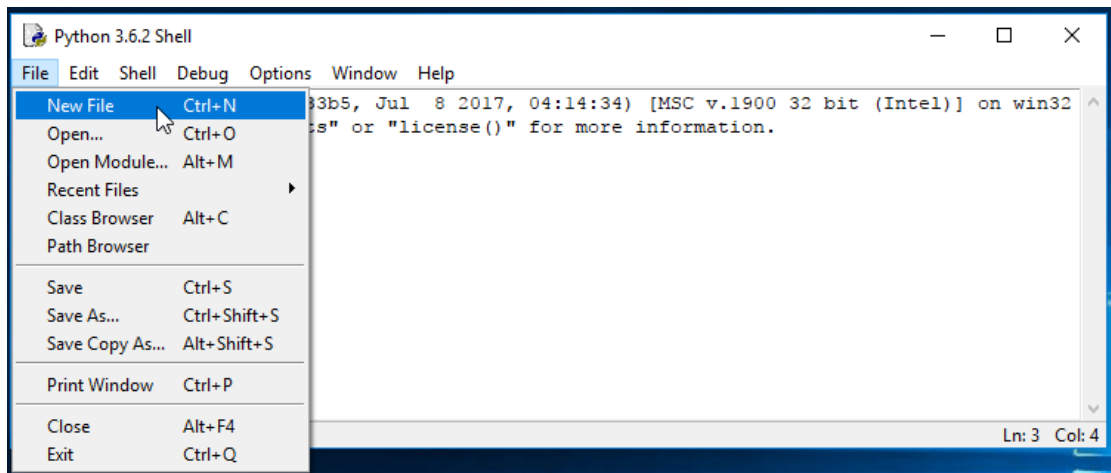
33

Περιλαμβάνει το δικό του περιβάλλον editor για να γράψουμε κώδικα - δεν χρειάζεται δηλαδή το Notepad++ ή κάποια άλλη εφαρμογή. Ο editor διαθέτει αρκετά κλασικά εργαλεία, ενώ παραμένει αρκετά ελαφρύ για το σύστημα σε σχέση με άλλα γνωστά IDLE.

Το μόνο μειονέκτημα του IDLE σε σχέση με το Notepad++ είναι πως δεν έχει εξελλητισμένο περιβάλλον.

Αν προτιμούμε να χρησιμοποιήσουμε το IDLE, δεν έχουμε παρά να επιλέξουμε να δημιουργήσουμε μέσα από το μενού του ένα νέο αρχείο, το οποίο θα ανοίξει στον native επεξεργαστή του IDLE.

³³ <https://www.pcsteps.gr/221565-μαθήματα-προγραμματισμού-python-1/>



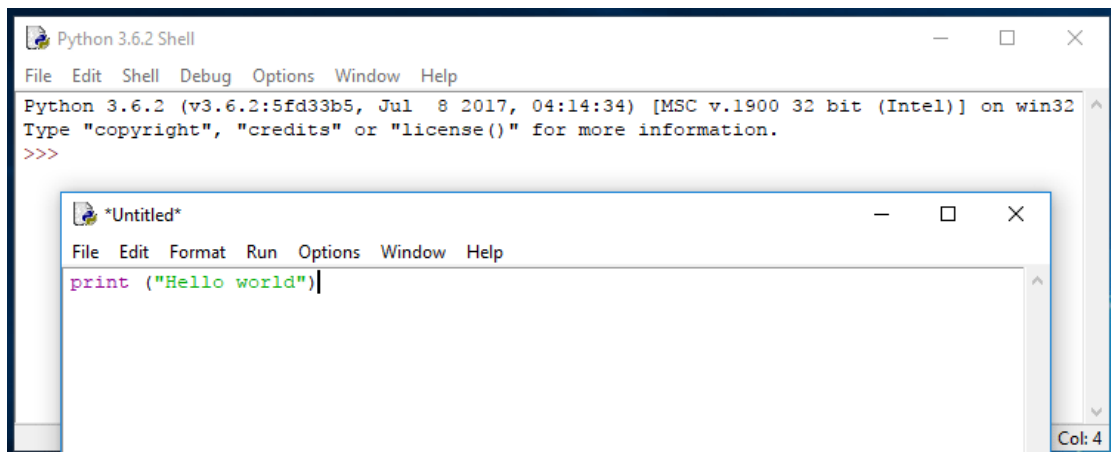
Εικόνα 9 Ανοιγμα Αρχείου

34

Γράφουμε κανονικά τον κώδικά μας, όπως είδαμε και προηγουμένως.

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

Ο επεξεργαστής του IDLE προσθέτει αυτόματα την κατάλληλη μορφοποίηση, ξεχωρίζοντας την εντολή και το κείμενο στην προκειμένη περίπτωση.



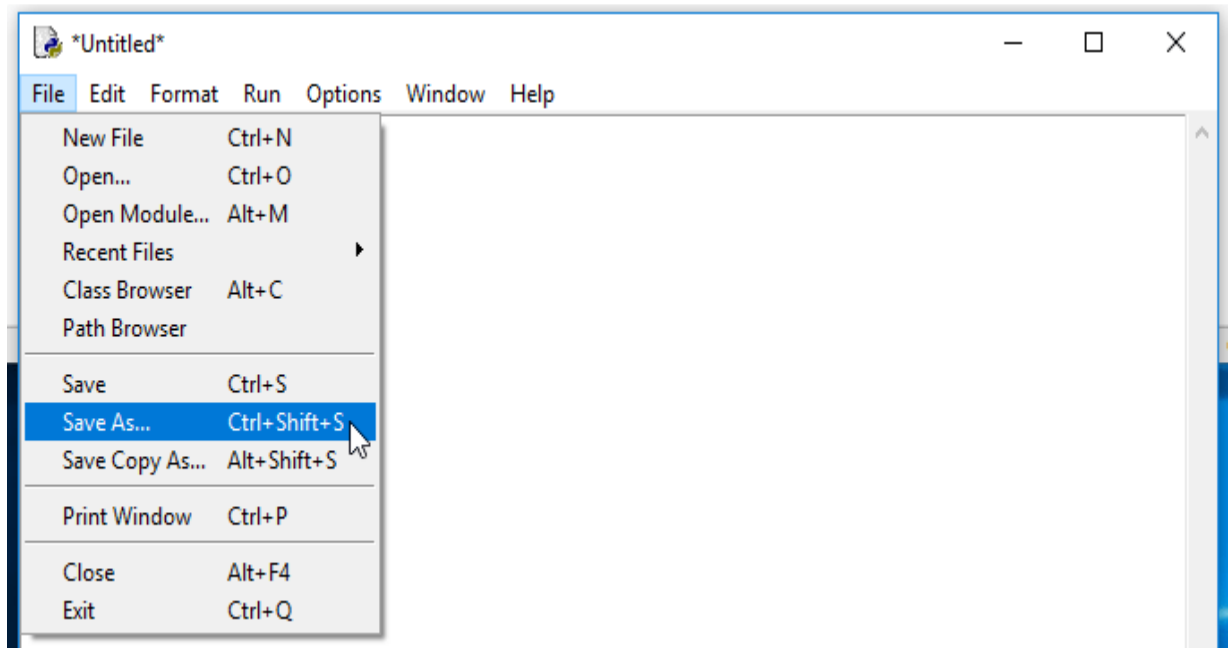
Εικόνα 10 Εγγραφή Κώδικα

35

³⁴ <https://www.pcsteps.gr/221565-μαθήματα-προγραμματισμού-python-1/>

³⁵ <https://www.pcsteps.gr/221565-μαθήματα-προγραμματισμού-python-1/>

Αφού γράψουμε τον κώδικα μας, μπορούμε να το αποθηκεύσουμε.

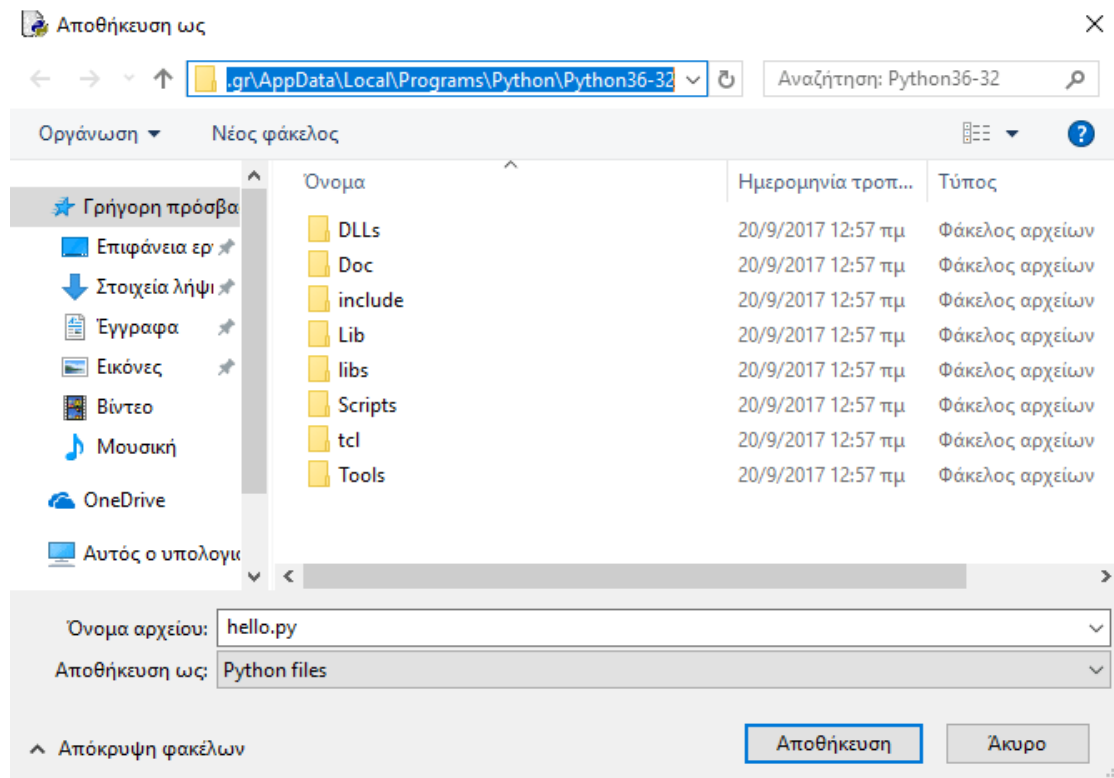


Εικόνα 11 Αποθήκευση

36

Από προεπιλογή το IDLE αποθηκεύει το αρχείο μας με κατάληξη .py, στον φάκελο εγκατάστασης της Python στο σύστημά μας (πχ στο AppData -> Local -> Programs -> Python -> Python36-32).

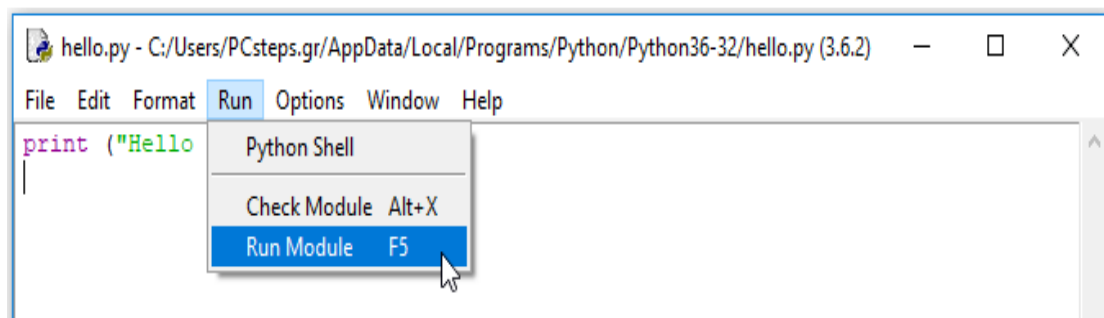
³⁶ <https://www.pcsteps.gr/221565-μαθήματα-προγραμματισμού-python-1/>



Εικόνα 12 Αποθήκευση αρχείου python

37

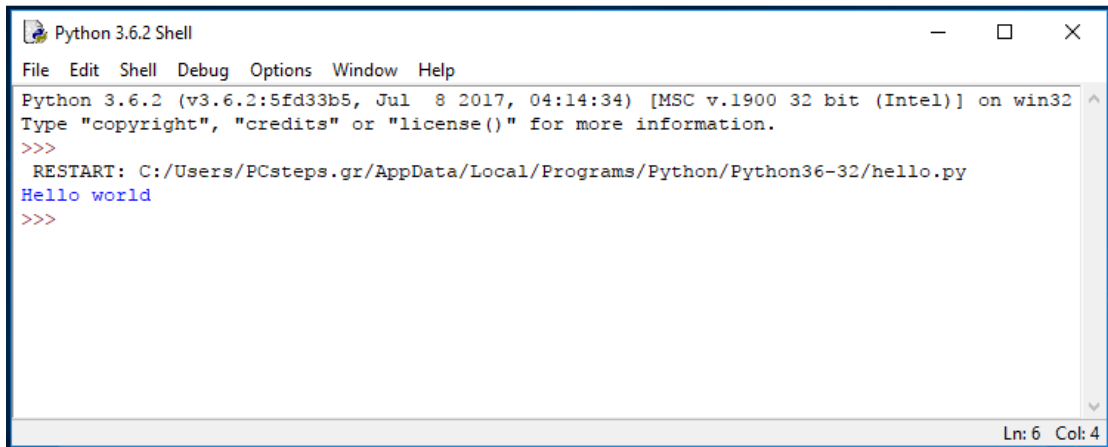
Πλέον μπορούμε να τρέξουμε άμεσα το πρόγραμμα. Απλά πατάμε Run -> Run Module ή "F5", χωρίς να χρειάζεται να κάνουμε κάποια άλλη ενέργεια από τη γραμμή εντολών ή να γυρεύουμε σε ποιον φάκελο είχαμε αποθηκεύσει το αρχείο μας.



Εικόνα 13 Run Module

³⁷ <https://www.pcsteps.gr/221565-μαθήματα-προγραμματισμού-python-1/>

Στο βασικό παράθυρο του IDLE βλέπουμε αμέσως το αποτέλεσμα.



```
Python 3.6.2 Shell
File Edit Shell Debug Options Window Help
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Users/PCsteps.gr/AppData/Local/Programs/Python/Python36-32/hello.py
Hello world
>>>
```

Εικόνα 14 Παράθυρο Idle

38

2.2.2 ΕΓΚΑΤΑΣΤΑΣΗ ΤΗΣ PYTHON ΣΕ LINUX

Η Python πρέπει να είναι ήδη εγκατεστημένη, μάλλον όμως μια παλαιότερη έκδοση. Για να εγκαταστήσει τη νεότερη έκδοση, επομένως μπορεί ο χρήστης, είτε να κατεβάσει τον πηγαίο κώδικα από τον ιστότοπο της Python και να τον μεταγλωττίσει ή να την εγκαταστήσει χρησιμοποιώντας τον διαχειριστή πακέτων λογισμικού (package manager) που συνοδεύει το λειτουργικό του σύστημα.

2.2.3 ΕΓΚΑΤΑΣΤΑΣΗ ΤΗΣ PYTHON ΣΕ MAC

Mac OS X: Πρέπει να υπάρχει ήδη προεγκατεστημένη μια έκδοση της Python. Επειδή το πιθανότερο είναι να πρόκειται για μια παλαιότερη, θα πρέπει να κατεβάσει κάποιος τη νεότερη έκδοση που θα βρείτε στον ιστότοπο της Python.³⁹

³⁸ <https://www.pcsteps.gr/221565-μαθήματα-προγραμματισμού-python-1/>

³⁹ www.python.org/downloads/mac-osx

ΚΕΦΑΛΑΙΟ 3

3. ΕΝΣΩΜΑΤΩΜΕΝΟΙ ΤΥΠΟΙ ΕΝΤΟΛΩΝ ΣΤΗΝ ΡΥΤΗΘ

Οι ακόλουθες ενότητες περιγράφουν τους τυπικούς τύπους, που είναι ενσωματωμένοι στον διερμηνέα, δηλαδή στον Idle. Οι κύριοι ενσωματωμένοι τύποι είναι αριθμοί, ακολουθίες, αντιστοιχίσεις, αρχεία, κλάσεις, παρουσίες και εξαιρέσεις.

Ορισμένες λειτουργίες υποστηρίζονται από διάφορους τύπους αντικειμένων. Συγκεκριμένα, σχεδόν όλα τα αντικείμενα μπορούν να συγκριθούν, να ελεγχθούν για την τιμή αλήθειας και να μετατραπούν σε μια συμβολοσειρά με τη συνάρτηση repr () ή τη ελαφρά διαφορετική λειτουργία str (). Η τελευταία λειτουργία χρησιμοποιείται, όταν ένα αντικείμενο έχει γραφτεί από τη λειτουργία print ().

3.1 Ενότητα 1^η Έλεγχος για αλήθεια

Οποιοδήποτε αντικείμενο μπορεί να δοκιμαστεί για την τιμή αλήθειας, για χρήση σε κατάσταση if ή while ή ως operand των λειτουργιών Boolean παρακάτω. Οι παρακάτω τιμές θεωρούνται ψευδείς:

- None

- False
- μηδέν οποιουδήποτε αριθμητικού τύπου, για παράδειγμα, 0 , 0L , 0.0 , 0j .
- κάθε κενή ακολουθία, για παράδειγμα, " , () , [] .

Πίνακας 1 Boolean Operations

Λειτουργία	Αποτέλεσμα	Σημειώσεις
x or y	εάν το x είναι ψευδές, τότε y , αλλιώς x	(1)
x and y	εάν το x είναι ψευδές, τότε το x , αλλιώς y	(2)
not x	εάν το x είναι ψευδές, τότε το True , αλλιώς το False	(3)

Όλες οι άλλες τιμές θεωρούνται αληθείς - έτσι αντικείμενα πολλών τύπων είναι πάντα αληθινά. Οι λειτουργίες και οι ενσωματωμένες λειτουργίες που έχουν ένα αποτέλεσμα Boolean επιστρέφουν πάντα 0 ή False για ψευδείς και 1 ή True για αληθείς, εκτός εάν δηλώνεται διαφορετικά.

3.2 Ενότητα 2^η Boolean Operatios- and, or not

Πίνακας 2 Λειτουργίες Σύγκρισης

Λειτουργία	Έννοια	Σημειώσεις
<	αυστηρά λιγότερο από	
<=	μικρότερη ή ίση	
>	αυστηρά μεγαλύτερη από	
>=	μεγαλύτερη ή ίση	
==	Ίσος	
!=	δεν είναι ίσο	(1)
is	ταυτότητα αντικειμένου	
is not	ανυπαρξία ταυτότητας αντικειμένου	

Αυτές είναι οι λειτουργίες Boolean, που έχουν ταξινομηθεί με αύξουσα προτεραιότητα:

Σημειώσεις:

- 1.Πρόκειται για χειριστή βραχυκυκλώματος, επομένως αξιολογεί μόνο το δεύτερο όρισμα, εάν το πρώτο είναι ψευδές.
- 2.Πρόκειται για χειριστή βραχυκυκλώματος, επομένως αξιολογεί μόνο το δεύτερο όρισμα, εάν το πρώτο είναι αληθές.
- 3.not έχει χαμηλότερη προτεραιότητα από τους μη-Boolean operators, οπότε $\text{not } a == b$ ερμηνεύεται ως $\text{not } (a == b)$, και $a == \text{not } b$ είναι σφάλμα σύνταξης.

3.3 Ενότητα 3^η Συγκρίσεις

Οι λειτουργίες σύγκρισης υποστηρίζονται από όλα τα αντικείμενα. Όλοι έχουν την ίδια προτεραιότητα (η οποία είναι μεγαλύτερη από αυτή των λειτουργιών Boolean). Οι συγκρίσεις μπορούν να συνδεθούν αυθαίρετα. για παράδειγμα, το $x < y <= z$ είναι ισοδύναμο με το $x < y$ and $y <= z$, εκτός από το ότι το y αξιολογείται μόνο μία φορά (αλλά και στις δύο περιπτώσεις το z δεν αξιολογείται καθόλου όταν το $x < y$ θεωρείται ψευδές)

Ο πίνακας αυτός συνοψίζει τις λειτουργίες σύγκρισης:

Σημειώσεις:

1.!= μπορεί επίσης να γραφτεί <> , αλλά αυτό είναι μια απαρχαιωμένη χρήση που διατηρείται μόνο για συμβατότητα προς τα πίσω. Ο νέος κώδικας θα πρέπει πάντα να χρησιμοποιείται !=

Αντικείμενα διαφορετικών τύπων, εκτός από διαφορετικούς αριθμητικούς τύπους και διαφορετικούς τύπους συμβολοσειρών, ποτέ δεν συγκρίνονται ίσοι. αυτά τα αντικείμενα διατάσσονται με συνέπεια αλλά αυθαίρετα (έτσι ώστε η ταξινόμηση μιας ετερογενούς συστοιχίας να αποδίδει ένα συνεπές αποτέλεσμα). Επιπλέον, ορισμένοι τύποι (για παράδειγμα, αντικείμενα αρχείου) υποστηρίζουν μόνο μια εκφυλισμένη έννοια σύγκρισης όπου δύο αντικείμενα αυτού του τύπου είναι άνισα. Και πάλι, τέτοια αντικείμενα παραγγέλλονται αυθαίρετα αλλά σταθερά. Οι χειριστές < , <= , > και >= θα αυξήσουν την εξαίρεση TypeError όταν ο τελεστής είναι ένας σύνθετος αριθμός.⁴⁰

Λεπτομέρειες εφαρμογής CPython: Αντικείμενα διαφορετικών τύπων εκτός από αριθμούς ταξινομούνται βάσει των ονομάτων τους τύπου. τα αντικείμενα των ίδιων τύπων που δεν υποστηρίζουν τη σωστή σύγκριση διατάσσονται από τη διεύθυνσή τους.

3.4 Ενότητα 4^η Αριθμητικοί τύποι-int, float, long, complex

Υπάρχουν τέσσερις διαφορετικοί αριθμητικοί τύποι:

- απλοί ακέραιοι αριθμοί ,
- μακρύ ακέραιοι αριθμοί ,
- αριθμοί κυμαινόμενου σημείου
- και πολύπλοκοι αριθμοί .

⁴⁰ <https://www.openbook.gr/eisagwgi-ston-antikeimenostrafi-programmatismo-me-python/>

Επιπλέον, οι Booleans είναι υποτύποι απλών ακέραιων αριθμών. Οι απλοί ακέραιοι αριθμοί (που ονομάζονται και *ακέραιοι αριθμοί*) υλοποιούνται χρησιμοποιώντας το **long** C, το οποίο τους δίνει τουλάχιστον 32 bits ακρίβειας (το `sys.maxint` είναι πάντοτε ρυθμισμένο στην μέγιστη τιμή απλού ακέραιου αριθμού για την τρέχουσα πλατφόρμα, η ελάχιστη τιμή είναι `-sys.maxint - 1`)⁴¹.

Οι ακέραιοι αριθμοί έχουν απεριόριστη ακρίβεια. Οι αριθμοί κυμαινόμενου σημείου συνήθως υλοποιούνται χρησιμοποιώντας **double** στο C πληροφορίες σχετικά με την ακρίβεια και την εσωτερική αναπαράσταση των αριθμών κυμαινόμενου σημείου για το μηχάνημα στο οποίο εκτελείται το πρόγραμμα διατίθενται στο `sys.float_info`. Οι σύνθετοι αριθμοί έχουν ένα πραγματικό και φανταστικό κομμάτι, το κάθε ένα είναι ένας αριθμός κινητής υποδιαστολής.

Οι αριθμοί δημιουργούνται με αριθμητικά γράμματα ή ως αποτέλεσμα ενσωματωμένων λειτουργιών και χειριστών. Τα αδιατάρακτα ακέραια γράμματα (συμπεριλαμβανομένων των δυαδικών, δεκαεξαδικών και οκταδικών αριθμών) αποδίδουν απλούς ακέραιους αριθμούς εκτός αν η τιμή που υποδηλώνουν είναι υπερβολικά μεγάλη για να αναπαρασταθεί ως απλός ακέραιος αριθμός, οπότε αποδίδουν έναν μακρύ ακέραιο αριθμό. Τα ακέραια γράμματα με ένα επίθεμα 'L' ή 'l' αποδίδουν μακρούς ακέραιους ('L' προτιμάται επειδή το ll φαίνεται πάρα πολύ σαν έντεκα!).

Τα αριθμητικά κυκλώματα που περιέχουν ένα δεκαδικό σημείο ή ένα σήμα εκθέτη αποδίδουν αριθμούς κυμαινόμενου σημείου. Η προσθήκη του 'j' ή του 'J' σε ένα αριθμητικό γράμμα δίνει έναν φανταστικό αριθμό (έναν σύνθετο αριθμό με μηδενικό πραγματικό τμήμα) που μπορείτε να προσθέσετε σε έναν ακέραιο αριθμό ή σε ένα float για να πάρετε έναν πολύπλοκο αριθμό με πραγματικά και φανταστικά μέρη.

⁴¹ <https://www.openbook.gr/eisagwgi-ston-programmatismo-me-arwgo-ti-glwssa-python/>

Η Python υποστηρίζει πλήρως την μεικτή αριθμητική: όταν ένας δυαδικός αριθμητικός χειριστής έχει τελεστές διαφορετικών αριθμητικών τύπων, ο τελεστής με τον "στενότερο" τύπο διευρύνεται με αυτόν του άλλου, όπου ο απλός ακέραιος είναι στενότερος από τον μακρύ ακέραιο είναι στενότερος από το πλωτό σημείο είναι στενότερος συγκρότημα. Οι συγκρίσεις μεταξύ αριθμών μικτού τύπου χρησιμοποιούν τον ίδιο κανόνα. Οι κατασκευαστές `int()`, `long()`, `float()`, και `complex()` μπορούν να χρησιμοποιηθούν για την παραγωγή αριθμών συγκεκριμένου τύπου.

Πίνακας 3 Αριθμητικοί Τύποι

Λειτουργία	Αποτέλεσμα	Σημειώσεις
$x + y$	το άθροισμα των x και y	
$x - y$	διαφορά x και y	
$x * y$	προϊόν των x και y	
x / y	πηλίκο των x και y	(1)
$x // y$	(πάτωμα) του x και y	(4) (5)
$x \% y$	υπόλοιπο x / y	(4)
$-x$	x αναιρείται	
$+x$	x αμετάβλητο	
<code>abs(x)</code>	απόλυτη τιμή ή μέγεθος x	(3)
<code>int(x)</code>	x μετατρέπεται σε ακέραιο	(2)
<code>long(x)</code>	x μετατρέπεται σε μακρύ ακέραιο	(2)
<code>float(x)</code>	x μετατρέπεται σε κυμαινόμενο σημείο	(6)
<code>complex(re,im)</code>	ένα πολύπλοκο αριθμό με πραγματικό μέρος re , φανταστικό μέρος im . Οι προεπιλεγμένες ρυθμίσεις είναι μηδενικές.	
<code>c.conjugate()</code>	σύζευξη του	

Λειτουργία	Αποτέλεσμα	Σημειώσεις
	σύνθετου αριθμού c . (Ταυτότητα σε πραγματικούς αριθμούς)	
<code>divmod(x, y)</code>	το ζεύγος $(x // y, x \% y)$	(3) (4)
<code>pow(x, y)</code>	x στη δύναμη y	(3) (7)
<code>x ** y</code>	x στη δύναμη y	(7)

Όλοι οι ενσωματωμένοι αριθμητικοί τύποι υποστηρίζουν τις ακόλουθες λειτουργίες.

Σημειώσεις:

1.Για διαίρεση ακέραιου ή μήκους, το αποτέλεσμα είναι ένας ακέραιος αριθμός. Το αποτέλεσμα είναι πάντα στρογγυλεμένο προς το μείον άπειρο: $1/2$ είναι 0, $(-1) / 2$ είναι -1, $1 / (-2)$ είναι -1 και $(-1) / (-2)$ είναι 0. Το αποτέλεσμα είναι ένας μακρής ακέραιος αριθμός εάν ένας τελεστής είναι ένας μακρής ακέραιος, ανεξάρτητα από την αριθμητική τιμή.

2.Η μετατροπή χρησιμοποιώντας `int ()` ή `long ()` περικόπτει προς το μηδέν, όπως η σχετική συνάρτηση, `math.trunc()` .Η συνάρτηση `math.floor()` είναι για να στρογγυλοποιηθούν προς τα κάτω και το `math.ceil()` για να στρογγυλοποιηθούν τα νούμερα προς τα πάνω.

Λειτουργία	Αποτέλεσμα
<code>math.trunc(x)</code>	x κατατεμαχισμένο στο Integral
<code>round(x[, n])</code>	x στρογγυλεμένο σε n ψηφία, στρογγυλοποιώντας τους δεσμούς μακριά από το μηδέν. Αν το n παραλειφθεί, τότε το προεπιλεγμένο είναι 0.
<code>math.floor(x)</code>	το μεγαλύτερο ακέραιο σαν float $\leq x$
<code>math.ceil(x)</code>	το μικρότερο ακέραιο σαν float $\geq x$

Πίνακας 4 Λειτουργίες μαθηματικών

Όλοι οι numbers.Real τύποι (int , long και float) περιλαμβάνουν επίσης τις παραπάνω λειτουργίες.⁴²

3.5 Ενότητα 5^η Δυαδικές λειτουργίες σε τύπους ακεραίων

Οι δυαδικές πράξεις έχουν νόημα μόνο για ακέραιους αριθμούς. Οι αρνητικοί αριθμοί θεωρούνται ως η συμπληρωματική τιμή του 2 (αυτό προϋποθέτει έναν αρκετά μεγάλο αριθμό μπιτ που δεν εμφανίζεται υπερχείλιση κατά τη διάρκεια της λειτουργίας).

Οι προτεραιότητες των δυαδικών πράξεων είναι όλες χαμηλότερες από τις αριθμητικές πράξεις και είναι υψηλότερες από τις συγκρίσεις, η μονόπλευρη λειτουργία ~ έχει την ίδια προτεραιότητα με τις άλλες οριακές αριθμητικές λειτουργίες (+ και -).⁴³

⁴² Μάνης Γ., (2015), «Εισαγωγή στον προγραμματισμό με αρωγό τη γλώσσα Python.» έκδοση: Σύνδεσμος ελληνικών ακαδημαϊκών βιβλιοθηκών. (pdf)

⁴³ <https://www.openbook.gr/eisagwgi-ston-programmatismo-me-arwgo-ti-glwssa-python/>

Αυτός ο πίνακας παραθέτει τις ταξινομημένες σε αύξουσα προτεραιότητα λειτουργίες με δυαδικά ψηφία

Πίνακας 5 Λειτουργίες με δυαδικά ψηφία

Λειτουργία	Αποτέλεσμα	Σημειώσεις
$x y$	bit ή x και y	
$x \wedge y$	bitwise <i>exclusive</i> ή x και y	
$x \& y$	bit και x και y	
$x \ll n$	x μετακινήθηκε αριστερά από n bits	(1) (2)
$x \gg n$	x μετατοπισμένο δεξιά από n bits	(1) (3)
$\sim x$	τα δυαδικά ψηφία x ανεστραμμένα	

Σημειώσεις:

1.Οι αρνητικοί αριθμοί μετατόπισης είναι παράνομοι και προκαλούν αύξηση του ValueError .

2.Μια αριστερή μετατόπιση με n bits είναι ισοδύναμη με τον πολλαπλασιασμό με $\text{pow}(2, n)$. Ένας μακρύς ακέραιος αριθμός επιστρέφεται αν το αποτέλεσμα υπερβαίνει το εύρος των απλών ακεραίων.

3.Μια δεξιά μετατόπιση με n bits είναι ισοδύναμη με τη διαίρεση από $\text{pow}(2, n)$.

3.6 Ενότητα 6^η Τύποι Iterator

Η Python υποστηρίζει μια ιδέα επανάληψης πάνω από τα δοχεία. Αυτό εφαρμόζεται με δύο διαφορετικές μεθόδους, αυτά χρησιμοποιούνται για να επιτρέπουν σε κλάσεις που ορίζονται από το χρήστη να υποστηρίζουν την επανάληψη. Οι ακολουθίες, που περιγράφονται παρακάτω λεπτομερέστερα, υποστηρίζουν πάντα τις μεθόδους επανάληψης.

Μια μέθοδος πρέπει να οριστεί για αντικείμενα κοντέινερ για την παροχή υποστήριξης επανάληψης:

```
container. __iter__ ()
```

Θα επιστρέψει ένα αντικείμενο `iterator`. Το αντικείμενο απαιτείται για να υποστηρίξει το πρωτόκολλο `iterator` που περιγράφεται παρακάτω. Εάν μια πλατφόρμα υποστηρίζει διαφορετικούς τύπους επανάληψης, μπορούν να παρασχεθούν πρόσθετες μέθοδοι για να ζητήσουν συγκεκριμένα `iterators` για αυτούς τους τύπους επανάληψης.

Τα ίδια τα αντικείμενα του `iterator` πρέπει να υποστηρίζουν τις ακόλουθες δύο μεθόδους, οι οποίες μαζί συνθέτουν το *πρωτόκολλο* του `iterator` :

```
iterator. __iter__ ()  
iterator. next ()
```

Η Python ορίζει πολλά αντικείμενα `iterator` που υποστηρίζουν την επανάληψη γενικών και ειδικών τύπων αλληλουχίας, λεξικών και άλλων πιο εξειδικευμένων μορφών. Οι συγκεκριμένοι τύποι δεν είναι σημαντικοί πέρα από την εφαρμογή του πρωτοκόλλου `iterator`.

3.7 Ενότητα 7^η Οι γεννήτριες της Python

Οι γεννήτριες της Python παρέχουν έναν βολικό τρόπο για την εφαρμογή του πρωτοκόλλου `iterator`. Εάν η μέθοδος `__iter__()` του αντικειμένου του `__iter__()` υλοποιείται ως γεννήτρια, θα επιστρέψει αυτόματα ένα αντικείμενο `iterator` (τεχνικά, ένα αντικείμενο γεννήτριας) που παρέχει τις `__iter__()` και `next()` .

3.8 Ενότητα 8^η Αρχεία

Η Python μας επιτρέπει να αποθηκεύουμε τον κώδικα μας σε αρχεία (που ονομάζονται επίσης ενότητες). Αυτό είναι πολύ χρήσιμο για περισσότερα σοβαρό προγραμματισμό, όπου δεν θέλουμε να επαναπροσδιορίσουμε έναν ορισμό μακροχρόνιας λειτουργίας από την ίδια αρχίζοντας ακριβώς να αλλάξει ένα λάθος. Με αυτόν τον τρόπο, καθορίζουμε ουσιαστικά τις δικές μας

ενότητες, όπως και οι ενότητες που έχουν ήδη οριστεί στη βιβλιοθήκη Python.⁴⁴

3.9 Ενότητα 9^η Σχόλια

Σε εντολή Python, οτιδήποτε μετά από ένα σύμβολο # είναι ένα σχόλιο. Για παράδειγμα: εκτύπωση "Hello world" # αυτό είναι ανόητο. Τα σχόλια δεν αποτελούν μέρος της εντολής, αλλά προορίζονται μάλλον ως τεκμηρίωση για οποιονδήποτε διαβάσει ο κώδικας. Τα σχόλια πολλαπλών γραμμών είναι επίσης δυνατά και περικλείονται από τριπλά σύμβολα διπλής προσφοράς"" "Αυτό είναι ένα παράδειγμα ενός μακρού σχολίου που συνεχίζεται και επάνω. "" "

3.10 Ενότητα 10^η Συνάρτηση if

Η συνάρτηση if, εφαρμόζεται συνήθως σε δομές ροής ελέγχου. Επιλέγει ακριβώς μία από τις σουίτες αξιολογώντας τις εκφράσεις μία προς μία μέχρις ότου διαπιστωθεί ότι είναι αληθές, τότε αυτή η σουίτα εκτελείται (και κανένα άλλο μέρος της εντολής if δεν εκτελείται ή δεν αξιολογείται). Εάν όλες οι εκφράσεις είναι ψευδείς, εκτελείται η ακολουθία της άλλης ρήτρας, εάν υπάρχει.

3.11 Ενότητα 11^η Τύποι ακολουθίας

Υπάρχουν επτά τύποι ακολουθίας: συμβολοσειρές, συμβολοσειρές Unicode, λίστες, πλειάδες, bytearray, buffer και αντικείμενα xrange.

Λίστες

Άλλοι σημαντικοί τύποι ακολουθίας που χρησιμοποιούνται στην Python περιλαμβάνουν λίστες και πλειάδες. Δημιουργείται ένας τύπος αλληλουχίας με τη συναρμολόγηση ορισμένων άλλων τύπων σε μια σειρά. Με την τοποθέτηση λιστών σε λίστες με αυτό τον τρόπο, μπορούμε να δημιουργήσουμε περίπλοκες δομές.⁴⁵

⁴⁴ <https://www.openbook.gr/eisagwgi-ston-programmatismo-me-arwgo-ti-glwssa-python/>

Συμβολοσειρές

Μια συμβολοσειρά στην Python είναι μια ακολουθία χαρακτήρων. Σε ορισμένες περιπτώσεις, οι χορδές δηλαδή οι ακολουθίες είναι παρόμοιες με τις λίστες, ωστόσο, υπάρχουν σημαντικές διαφορές. Μία σημαντική διαφορά είναι ότι οι χορδές της Python είναι αμετάβλητες, ότι δεν μας επιτρέπεται να αλλάζουμε μεμονωμένα μέρη αυτών, όπως μπορούμε για μια λίστα.

Οι συμβολοσειρές Unicode

Οι συμβολοσειρές Unicode μοιάζουν πολύ με συμβολοσειρές, αλλά καθορίζονται στη σύνταξη χρησιμοποιώντας έναν προηγούμενο χαρακτήρα 'u' : `u'abc'` , `u"def"` . Εκτός από τη λειτουργικότητα που περιγράφεται εδώ, υπάρχουν επίσης μέθοδοι που αφορούν συγκεκριμένες συμβολοσειρές που περιγράφονται στην ενότητα μέθοδοι στοιχειοσειρών .

Οι λίστες κατασκευάζονται με αγκύλες, χωρίζοντας τα στοιχεία με κόμματα: `[a, b, c]` . Τα κουμπιά κατασκευάζονται από τον χειριστή των κόμματος (όχι μέσα σε αγκύλες), με ή χωρίς εγκάρσια παρενθέσεις, αλλά μια κενή πλειάδα πρέπει να έχει τις παρεμβαλλόμενες παρενθέσεις, όπως `a, b, c` ή `()` . Μια ενιαία πλειάδα στοιχείων πρέπει να έχει ένα τελικό κόμμα, όπως `(d,)` ..

Τα αντικείμενα `buffer` δεν υποστηρίζονται άμεσα από τη σύνταξη της Python, αλλά μπορούν να δημιουργηθούν καλώντας το ενσωματωμένο `buffer()` . Δεν υποστηρίζουν σύνδεση ή επανάληψη.

Τα αντικείμενα τύπου `xrange` είναι παρόμοια με τα `buffer`, επειδή δεν υπάρχει συγκεκριμένη σύνταξη για τη δημιουργία τους, αλλά δημιουργούνται χρησιμοποιώντας τη `xrange()` . Δεν υποστηρίζουν τον τεμαχισμό, τη σύζευξη ή την επανάληψη, και η χρήση `in` , `not in` , `min()` ή `max()` πάνω τους είναι αναποτελεσματική.

⁴⁵ <https://www.openbook.gr/eisagwgi-ston-antikeimenostrafi-programmatismo-me-python/>

Πίνακας 6 Λειτουργίες συγκρίσεων και συνθήκη αληθής ή ψευδής

Λειτουργία	Αποτέλεσμα	Σημειώσεις
$x \text{ in } s$	True αν ένα στοιχείο του s είναι ίσο με το x , αλλιώς False	(1)
$x \text{ not in } s$	False αν ένα στοιχείο s είναι ίσο με το x , αλλιώς True	(1)
$s + t$	η συνένωση των s και t	(6)
$s * n, n * s$	ισοδύναμο με την προσθήκη s στον εαυτό του n φορές	(2)
$s[i]$	το πρώτο στοιχείο της προέλευσης 0	(3)
$s[i:j]$	s από i σε j	(3) (4)
$s[i:j:k]$	s από το i στο j με το βήμα k	(3) (5)
$\text{len}(s)$	μήκος του s	
$\text{min}(s)$	το μικρότερο στοιχείο του s	
$\text{max}(s)$	μεγαλύτερο στοιχείο του s	
$s.\text{index}(x)$	δείκτης της πρώτης εμφάνισης του x σε s	
$s.\text{count}(x)$	ο συνολικός αριθμός περιστατικών x στο s	

Οι περισσότεροι τύποι ακολουθίας υποστηρίζουν τις ακόλουθες λειτουργίες. Οι εσωτερικές και not in λειτουργίες έχουν τις ίδιες προτεραιότητες με τις λειτουργίες σύγκρισης. Οι λειτουργίες + και * έχουν την ίδια προτεραιότητα με τις αντίστοιχες αριθμητικές λειτουργίες. Παρέχονται πρόσθετες μέθοδοι για τους τύπους μεταβλητών ακολουθιών .

Αυτός ο πίνακας παραθέτει τις λειτουργίες ακολουθίας που ταξινομούνται με προτεραιότητα προς τα πάνω. Στον πίνακα, s και t είναι αλληλουχίες του ίδιου τύπου. n , i και j είναι ακέραιοι αριθμοί:

Οι τύποι ακολουθιών υποστηρίζουν επίσης τις συγκρίσεις. Συγκεκριμένα, οι πλειάδες και οι λίστες συγκρίνονται λεξικογραφικά συγκρίνοντας τα αντίστοιχα στοιχεία. Αυτό σημαίνει ότι για να συγκρίνετε ίσο, κάθε στοιχείο πρέπει να

συγκρίνει ίσο και οι δύο αλληλουχίες πρέπει να είναι του ίδιου τύπου και να έχουν το ίδιο μήκος.⁴⁶

Σημειώσεις:

1. Όταν το *s* είναι μια συμβολοσειρά ή ένα αντικείμενο συμβολοσειράς Unicode, οι *in* και *not in* πράξεις λειτουργούν σαν μια δοκιμή υποζώνης.

2. Οι τιμές του *n* μικρότερες του 0 αντιμετωπίζονται ως 0 (η οποία αποδίδει μια κενή ακολουθία του ίδιου τύπου με *s*). Σημειώστε ότι τα στοιχεία της ακολουθίας *s* δεν αντιγράφονται. αναφέρονται πολλές φορές. Αυτό συχνά στοιχειώνει νέους προγραμματιστές της Python. Παράδειγμα:

1. `>>> lists = [[]] * 3`
2. `>>> lists`
3. `[[], [], []]`
4. `>>> lists [0] . append (3)`
5. `>>> lists`

```
[[3], [3], [3]]
```

Αυτό που συνέβη είναι ότι `[[]]` είναι ένας κατάλογος ενός στοιχείου που περιέχει μια κενή λίστα και έτσι και τα τρία στοιχεία του `[[]] * 3` είναι αναφορές σε αυτή τη μοναδική κενή λίστα. Η τροποποίηση οποιουδήποτε στοιχείου των *lists* τροποποιεί αυτή τη μοναδική λίστα. Μπορεί να δημιουργήσει μια λίστα διαφορετικών λιστών με αυτόν τον τρόπο:

⁴⁶ <https://www.openbook.gr/eisagwgi-ston-antikeimenostrafi-programmatismo-me-python/>

```
>>> lists = [[] for i in range ( 3 )]
```

```
>>> lists [ 0 ] . append ( 3 )
```

```
>>> lists [ 1 ] . append ( 5 )
```

```
>>> lists [ 2 ] . append ( 7 )
```

```
>>> lists
```

```
[[3], [5], [7]]47
```

3.12 Ενότητα 12^η Μέθοδοι στοιχειοσειράς

Παρακάτω παρατίθενται οι μέθοδοι συμβολοσειρών που υποστηρίζουν και οι δύο συμβολοσειρές 8-bit και τα αντικείμενα Unicode. Μερικά από αυτά είναι επίσης διαθέσιμα σε αντικείμενα bytearray .

Επιπλέον, οι συμβολοσειρές της Python υποστηρίζουν τις μεθόδους τύπου αλληλουχίας που περιγράφονται στους τύπους ακολουθίας - str, unicode, list, tuple, bytearray, buffer, xrange .

str. decode ([κωδικοποίηση [, σφάλματα]])

Αποκωδικοποιεί τη συμβολοσειρά χρησιμοποιώντας τον κωδικοποιητή που έχει καταχωριστεί για κωδικοποίηση . κωδικοποίηση των προεπιλογών στην προεπιλεγμένη κωδικοποίηση συμβολοσειράς, μπορεί να δοθούν σφάλματα για τον καθορισμό διαφορετικού σχήματος χειρισμού σφαλμάτων.

Η προεπιλογή είναι 'strict', πράγμα που σημαίνει ότι τα σφάλματα κωδικοποίησης αυξάνουν το UnicodeError . Άλλες πιθανές τιμές είναι 'ignore', 'replace' και οποιοδήποτε άλλο όνομα καταχωρήθηκε μέσω codecs.register_error(), υπάρχει το τμήμα Codec Base Classes .

⁴⁷ Μάνης Γ., (2015), «Εισαγωγή στον προγραμματισμό με αρωγό τη γλώσσα Python.» έκδοση: Σύνδεσμος ελληνικών ακαδημαϊκών βιβλιοθηκών. (pdf)

str. encode ([κωδικοποίηση [, σφάλματα]])

Επιστρέφτε μια κωδικοποιημένη έκδοση της συμβολοσειράς. Η προεπιλεγμένη κωδικοποίηση είναι η τρέχουσα κωδικοποίηση συμβολοσειράς. Μπορεί να δοθούν *σφάλματα* για τον καθορισμό διαφορετικού σχήματος χειρισμού σφαλμάτων. Η προεπιλογή για τα *σφάλματα* είναι 'strict', που σημαίνει ότι τα σφάλματα κωδικοποίησης δημιουργούν ένα σφάλμα.

Άλλες πιθανές τιμές είναι οι 'ignore', 'replace', 'xmlcharrefreplace', 'backslashreplace' και κάθε άλλη ονομασία, που θα καταχωρηθεί μέσω `codecs.register_error()`.

str. endswith (επίθημα [, έναρξη [, τέλος]])

Επιστροφή True αν η συμβολοσειρά τελειώνει με το καθορισμένο *επίθημα*, διαφορετικά επιστρέφει το False. Το *επίθημα* μπορεί επίσης να είναι μια πλειάδα επιθημάτων. Με προαιρετική *εκκίνηση*, μπορεί κάποιος να δοκιμάσει να ξεκινάει από τη θέση αυτή.

str. expandtabs ([tabsize])

Μας επιστρέφει ένα αντίγραφο της συμβολοσειράς όπου όλοι οι χαρακτήρες καρτελών αντικαθίστανται από ένα ή περισσότερα κενά, ανάλογα με την τρέχουσα στήλη και το μέγεθος της δεδομένης καρτέλας. Οι θέσεις των καρτελών εμφανίζονται σε κάθε χαρακτήρες *καρτέλας* (η προεπιλογή είναι 8, δίνοντας θέσεις καρτελών στις στήλες 0, 8, 16 και ούτω καθεξής). Για να αναπτύξετε τη συμβολοσειρά, η τρέχουσα στήλη έχει οριστεί στο μηδέν και η συμβολοσειρά εξετάζεται χαρακτήρας κατά χαρακτήρα. Εάν ο χαρακτήρας είναι μια καρτέλα (`\t`), ένας ή περισσότεροι χαρακτήρες διαστήματος εισάγονται στο αποτέλεσμα μέχρι η τρέχουσα στήλη να είναι ίση με την επόμενη θέση καρτέλας. (Ο χαρακτήρας από μόνος του δεν αντιγράφεται.) Εάν ο χαρακτήρας είναι μια νέα γραμμή (`\n`) ή επιστρέφει (`\r`), αντιγράφεται και η τρέχουσα στήλη επαναφέρεται στο μηδέν. Οποιοσδήποτε άλλος χαρακτήρας αντιγράφεται αμετάβλητος και η τρέχουσα στήλη αυξάνεται κατά ένα, ανεξάρτητα από τον τρόπο με τον οποίο εκπροσωπείται ο χαρακτήρας κατά την εκτύπωση.

```
>>> '01 \t 012 \t 0123 \t 01234' . expandtabs ()
'01  012  0123  01234'
>>> '01 \t 012 \t 0123 \t 01234' . expandtabs ( 4 )
'01 012 0123  01234'
```

str. find (υπο [, έναρξη [, τέλος]])

Μας επιστρέφει το χαμηλότερο δείκτη στη συμβολοσειρά, όπου βρίσκεται το *sub* subring *sub* μέσα στο slice *s*[start:end] .⁴⁸

str. format (* args , ** kwargs)

Εκτελεί μια διαδικασία μορφοποίησης συμβολοσειρών. Η συμβολοσειρά στην οποία ονομάζεται αυτή η μέθοδος μπορεί να περιέχει κυριολεκτικά κείμενα ή πεδία αντικατάστασης που οριοθετούνται από τιράντες {}. Κάθε πεδίο αντικατάστασης περιέχει είτε το αριθμητικό ευρετήριο ενός επιχειρήματος θέσης είτε το όνομα ενός όρου λέξεων-κλειδιών. Επιστρέφει ένα αντίγραφο της συμβολοσειράς όπου κάθε πεδίο αντικατάστασης αντικαθίσταται με την τιμή συμβολοσειράς του αντίστοιχου επιχειρήματος.

Αυτή η μέθοδος μορφοποίησης συμβολοσειρών είναι το νέο πρότυπο της Python 3 και θα πρέπει να προτιμάται σε σχέση με τη μορφοποίηση %, και περιγράφεται στις λειτουργίες μορφοποίησης γραμμών σε νέο κώδικα.

str. isalnum ()

Επιστροφή *true* αν όλοι οι χαρακτήρες της συμβολοσειράς είναι αλφαριθμητικοί και υπάρχει τουλάχιστον ένας χαρακτήρας, ψευδώς διαφορετικά.

⁴⁸ <https://www.openbook.gr/eisagwgi-ston-programmatismo-me-arwgo-ti-glwssa-python/>

ΚΕΦΑΛΑΙΟ 4

4. ΤΡΟΠΟΙ ΑΝΑΠΤΥΞΗΣ ΕΝΟΣ ΠΡΟΓΡΑΜΜΑΤΟΣ

Πολλοί άνθρωποι έχουν δυσκολίες ή απογοητεύονται σχετικά με τις γλώσσες προγραμματισμού που χρησιμοποιούν καθημερινά. Μερικοί θέλουν τα πράγματα που χειρίζονται να είναι πιο αφηρημένα, ενώ άλλοι αντιπαθούν τα χαρακτηριστικά εφαρμογής που επιθυμούν «τυποποιημένα».

Για να δημιουργηθεί μια γλώσσα προγραμματισμού και να αναπτυχθεί ένας κώδικας, δηλαδή ένα πρόγραμμα υπάρχει μια σειρά από τρόπους και συγκεκριμένα βήματα, που πρέπει να ακολουθήσει κάποιος, για να επιτύχει το σκοπό του.

4.1 Βήματα ανάπτυξης

4.1.1 Γνώση τεχνολογίας

Αρχικά θα πρέπει να γνωρίζουμε την τεχνολογία. Δεν μπορούμε να δημιουργήσουμε μια γλώσσα προγραμματισμού εάν δεν ξέρουμε, πώς να χρησιμοποιήσετε έναν υπολογιστή.⁴⁹..

⁴⁹ <https://www.openbook.gr/eisagwgi-ston-programmatismo-me-arwgo-ti-glwssa-python/>

4.1.2 Ορολογία

Έπειτα θα πρέπει να γνωρίζουμε την ορολογία. Οι συγγραφείς κώδικα συχνά χρησιμοποιούν άγνωστη ορολογία. Θα είναι καλύτερα αν διαβάσουμε διάφορους ορισμούς προγραμματισμού, πριν προχωρήσουμε στην ανάπτυξη ενός προγράμματος και να είμαστε βέβαιοι για όσα θα πρέπει να γνωρίζουμε, ότι τα γνωρίζουμε πραγματικά.

4.1.3 Γλώσσα προγραμματισμού

Στη συνέχεια θα πρέπει να αποφασίσουμε αν η γλώσσα σας προγραμματισμού, που θα αναπτύξουμε θα αντιμετωπίζει ένα πρόβλημα συγκεκριμένου τομέα ή θα είναι μια γλώσσα γενικού σκοπού.

Οφείλουμε να οριοθετήσουμε τη σημασιολογία της γλώσσας μας, καθώς και τις έννοιες της.

- Πρόκειται να επιτρέψουμε την άμεση πρόσβαση ή όχι.
- Ποιοι θα είναι οι τύποι δεδομένων της γλώσσας μας;
- Είναι μια στατική ή δυναμική γλώσσα;
- Ποιο είναι το μοντέλο μνήμης;
- Πώς θα χειριστούμε την ταυτότητα; Πρόκειται να χρησιμοποιήσουμε ένα απλό μοντέλο κλειδώματος ή κάτι πιο περίπλοκο;
- Θα υπάρχουν πρωτόγονες λειτουργίες ενσωματωμένα στη γλώσσα ή θα προέρχονται όλα από μια βιβλιοθήκη;⁵⁰
- Ποιο είναι το πρότυπο ή τα παραδείγματα της γλώσσας σας; Λειτουργικός; Αντικειμενοστραφής (python)
Πρωτότυπο (όπως JavaScript);
Προσανατολισμός; Πρότυπο προσανατολισμένο; Ή κάτι εντελώς νέο;
- Πώς θα συνδέεται η γλώσσα μας με υπάρχουσες βιβλιοθήκες και γλώσσες (κυρίως C);

⁵⁰ <https://www.openbook.gr/eisagwgi-ston-antikeimenostrafi-programmatismo-me-python/>

Αυτό το σημείο είναι σημαντικό αν δημιουργούμε μια συγκεκριμένη γλώσσα.

Στη συνέχεια καλό θα είναι να σκεφτούμε κάποια συγκεκριμένα καθήκοντα που κάποιος θα ήθελε να είναι σε θέση να εκτελέσει με τη γλώσσα που θα αναπτύξουμε.

Για παράδειγμα, "μπορεί να θέλουν να κατευθύνουν ένα ρομπότ για να ακολουθήσουν μια γραμμή" ή "μπορεί να θέλουν να δημιουργήσουν σχετικά φορητά προγράμματα γραφικών σε αυτό" ή "μπορεί να θέλουν να δημιουργήσουν εφαρμογές ιστού με αυτό".

Φυσικά είναι αναγκαίο να πειραματιστούμε με τις ιδέες σύνταξης (το κείμενο της γλώσσας) για τα παραπάνω παραδείγματα.

Είναι κρίσιμο να αποφασίσουμε εάν η γλώσσα μας θα ερμηνευτεί ή θα μεταγλωττιστεί. Αυτό σημαίνει ότι στον ερμηνευμένο κόσμο ο χρήστης μας θα επεξεργαστεί συνήθως το πρόγραμμα μας σε ένα πρόγραμμα επεξεργασίας και θα το εκτελέσει απευθείας στον διερμηνέα, όπως και στην Python, ενώ στον κόσμο της συλλογής, ο χρήστης μας θα επεξεργαστεί το πρόγραμμά σας, θα το συντάξει, θα αποθηκεύσει το εκτελέσιμο αρχείο κάπου και θα το εκτελέσει.

4.1.4 Επεξεργασία κώδικα

- Ευκολία στη χρήση
- Επισήμανση σύνταξης
- Αυτόματη απόσπαση
- Προβολές με καρτέλες
- Αρίθμηση γραμμών
- Ικανότητα να προσαρμόσετε την εμφάνιση και την αίσθηση του εκδότη
- Διαθεσιμότητα πρόσθετων⁵¹

⁵¹ <https://www.openbook.gr/eisagwgi-ston-programmatismo-me-arwgo-ti-glwssa-python/>

ΚΕΦΑΛΑΙΟ 5

5.ΑΝΑΛΥΣΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ ΡΥΘΗΘ

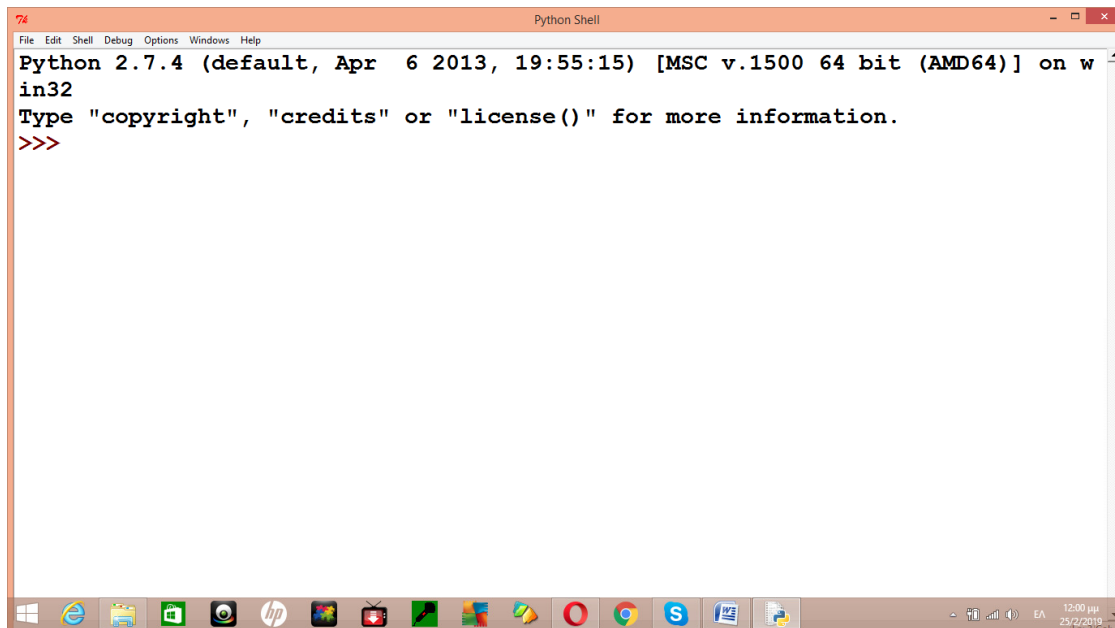
5.1ΕΦΑΡΜΟΓΕΣ ΚΑΙ ΠΑΡΑΔΕΙΓΜΑΤΑ

Όλα τα παραδείγματα έγιναν στο πρόγραμμα Python έκδοση 2.7.4 το οποίο, κατέβασα και έκανα εγκατάσταση στα windows.

5.1.1 Πρόγραμμα Python έκδοση 2.7.4

Είναι η αρχική εικόνα στον διερμηνευτή Idle, που ανοίγει, αφού έχει γίνει η εγκατάσταση της γλώσσας Προγραμματισμού Python.

Η συγκεκριμένη μας δείχνει την έκδοση του προγράμματος, και τι μπορούμε να πληκτρολογήσουμε, για να μάθουμε περισσότερες πληροφορίες για αυτό.

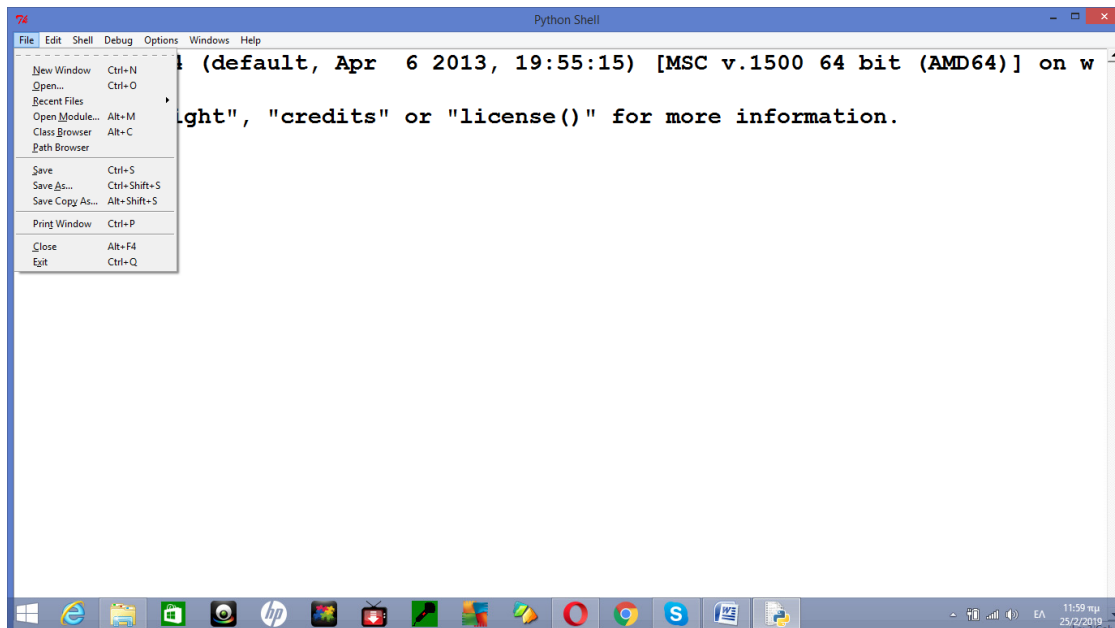


Εικόνα 15 Πρόγραμμα Python έκδοση 2.7.4

5.1.2 Πρώτη καρτέλα στο Πρόγραμμα Python File –Αρχείο

Σε αυτήν την καρτέλα εμφανίζονται όλα όσα είναι σχετικά με το μενού των αρχείων. Εδώ μπορούμε να δημιουργήσουμε ένα αρχείο, να αναζητήσουμε και να αποθηκεύσουμε ή και να εκτυπώσουμε.⁵²

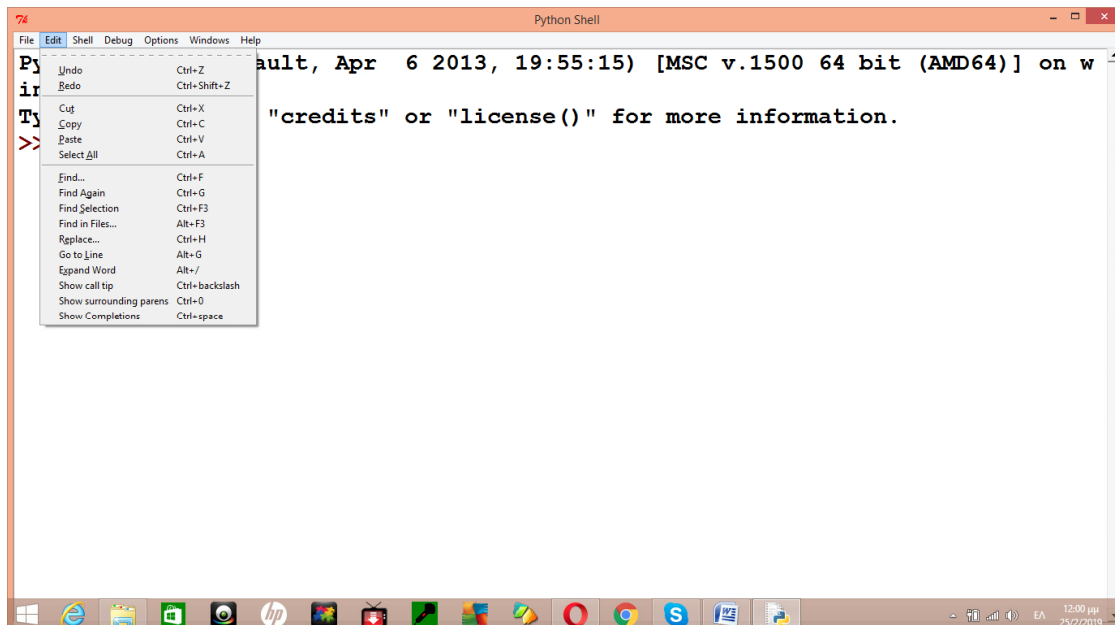
⁵² <https://www.openbook.gr/eisagwgi-ston-antikeimenostrafi-programmatismo-me-python/>



Εικόνα 16 Πρώτη Καρτέλα σε πρόγραμμα Python με τα Αρχεία

5.1.3 Δεύτερη καρτέλα στο Πρόγραμμα Python Edit

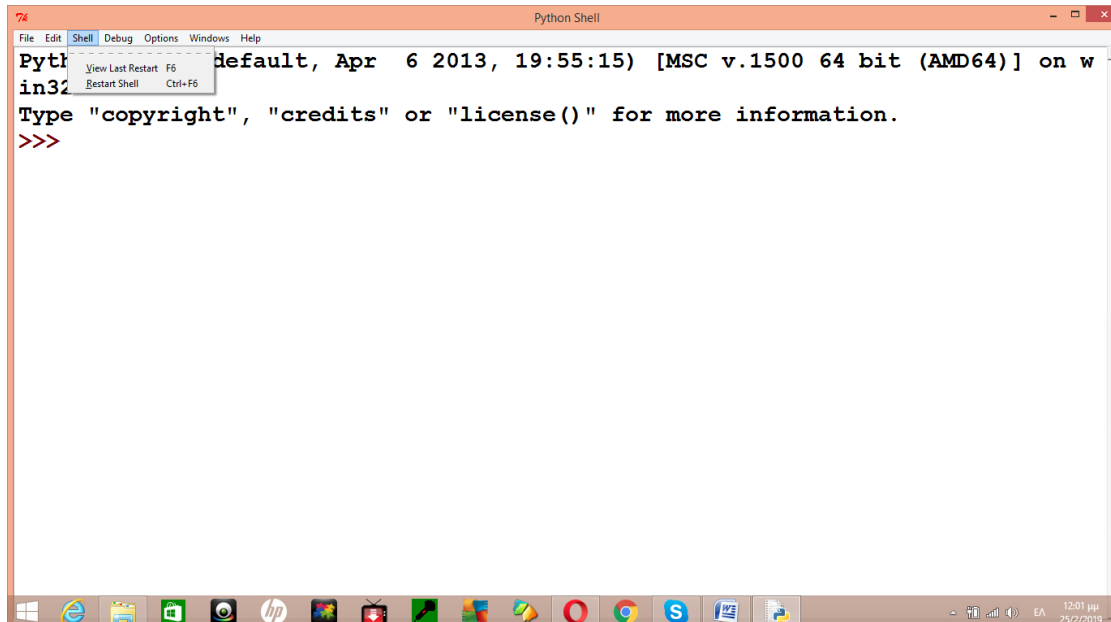
Στη συγκεκριμένη καρτέλα μπορούμε να κάνουμε επεξεργασία στον κώδικα ή στις εντολές που έχουμε γράψει.



Εικόνα 17 Δεύτερη καρτέλα σε Πρόγραμμα Python με την επεξεργασία

5.1.4 Τρίτη καρτέλα στο Πρόγραμμα Python Shell

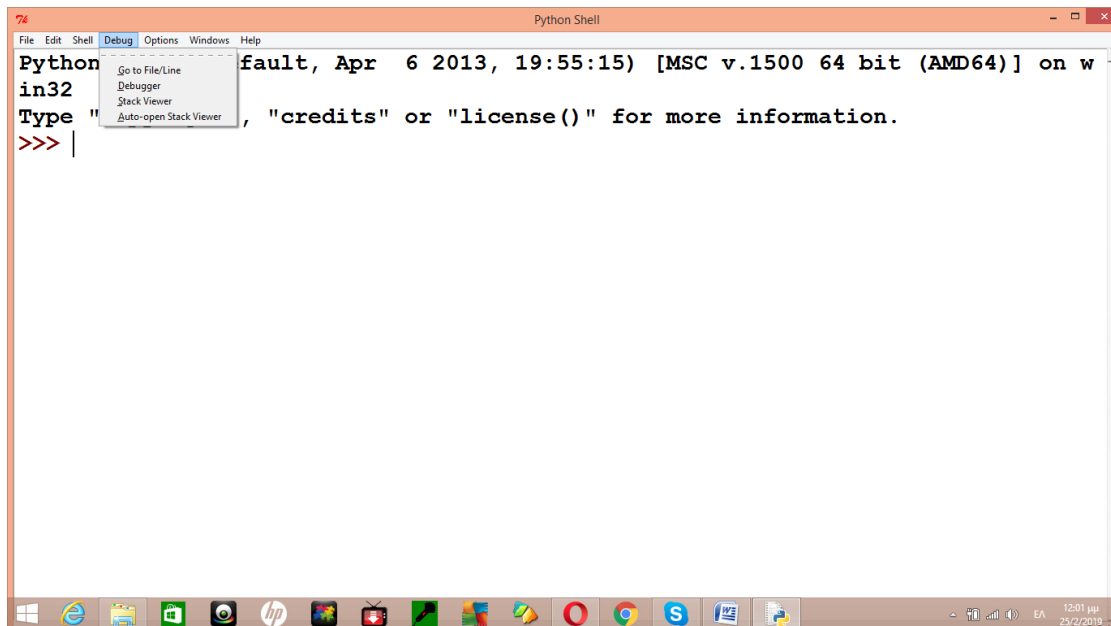
Σε αυτήν την καρτέλα από το μενού της γλώσσας προγραμματισμού μπορούμε να δούμε τις τελευταίες διαθέσιμες κινήσεις μας.



Εικόνα 18 Τρίτη Καρτέλα σε Πρόγραμμα Python Shell

5.1.5 Τέταρτη καρτέλα στο Πρόγραμμα Python Debug

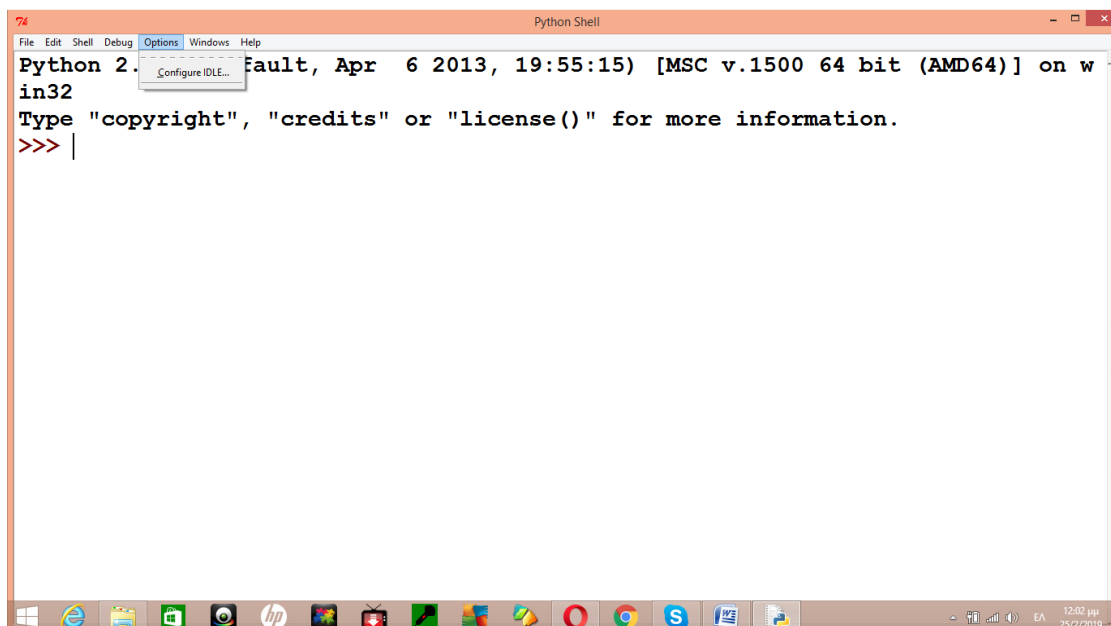
Στη καρτέλα αυτή, έχουμε τη δυνατότητα να μπορούμε να δούμε ολοκληρωμένο, ότι πρόγραμμα έχουμε φτιάξει μέχρι αυτή τη στιγμή.



Εικόνα 19 Τέταρτη καρτέλα σε πρόγραμμα Python Debug

5.1.6 Πέμπτη καρτέλα στο Πρόγραμμα Python Options

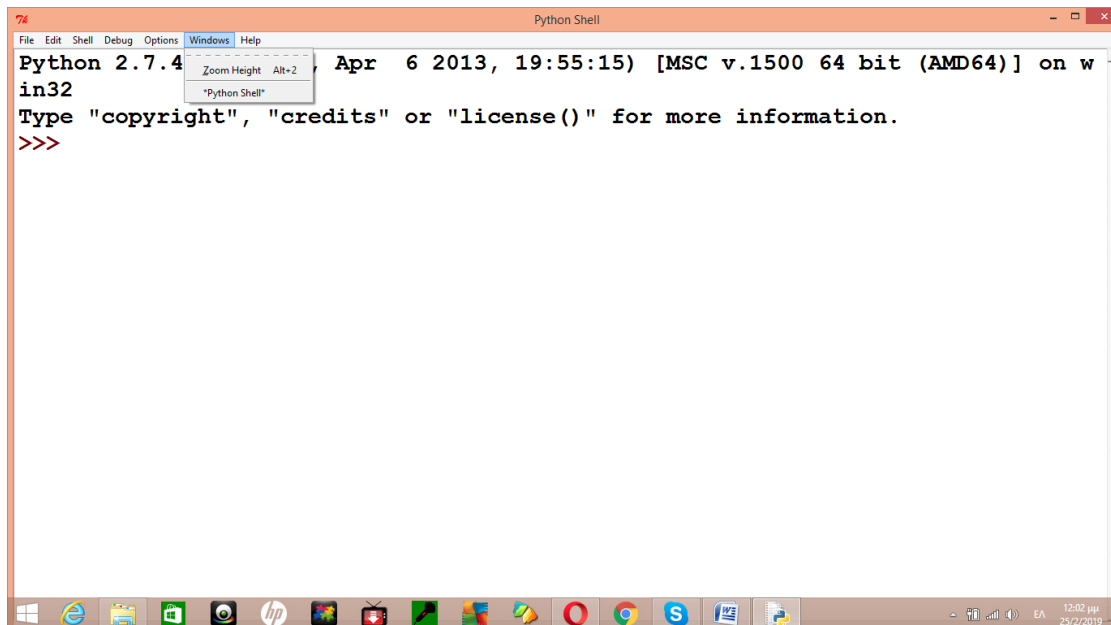
Σε αυτό το σημείο του μενού , μας δίνεται η δυνατότητα των επιλογών. Εδώ μπορούμε να μορφοποιήσουμε το παράθυρο του διερμηνευτή που δουλεύουμε. Από τύπους γραμματοσειρών μέχρι πιο εξειδικευμένα εργαλεία για τη μορφοποίηση του περιβάλλοντος της Python που επεξεργαζόμαστε.



Εικόνα 20 Πέμπτη καρτέλα σε πρόγραμμα Python Options

5.1.7 Έκτη καρτέλα στο Πρόγραμμα Python Windows

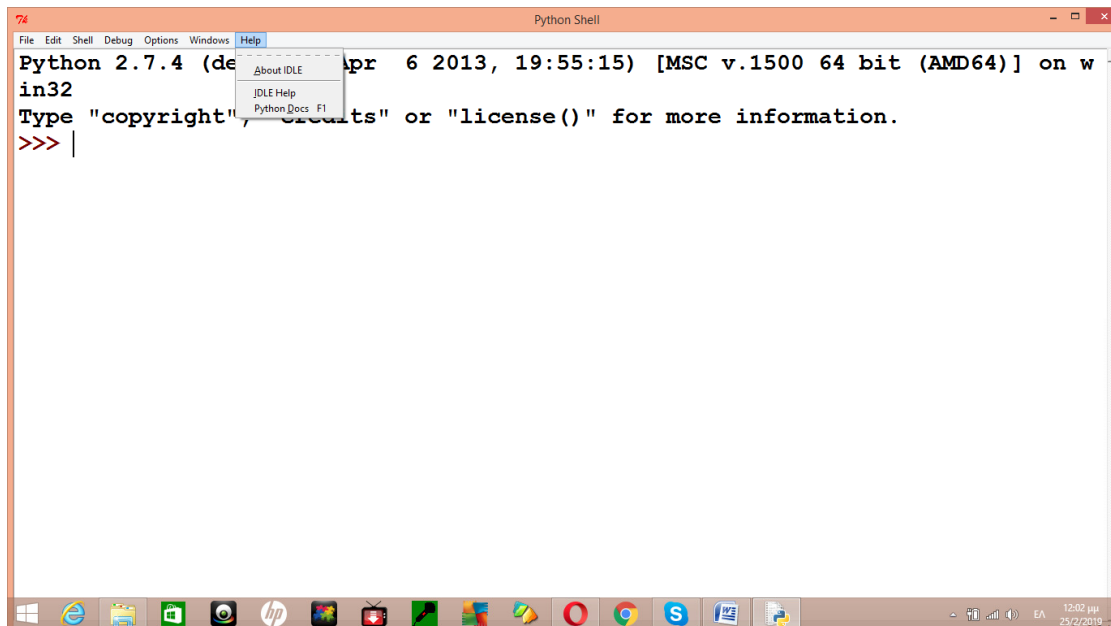
Η προτελευταία καρτέλα στο μενού της γλώσσας Python μας δείχνει που είναι εγκατεστημένο το πρόγραμμα μας.



Εικόνα 21 Έκτη καρτέλα σε Πρόγραμμα Python Windows

5.1.8 Έβδομη καρτέλα στο Πρόγραμμα Python Help

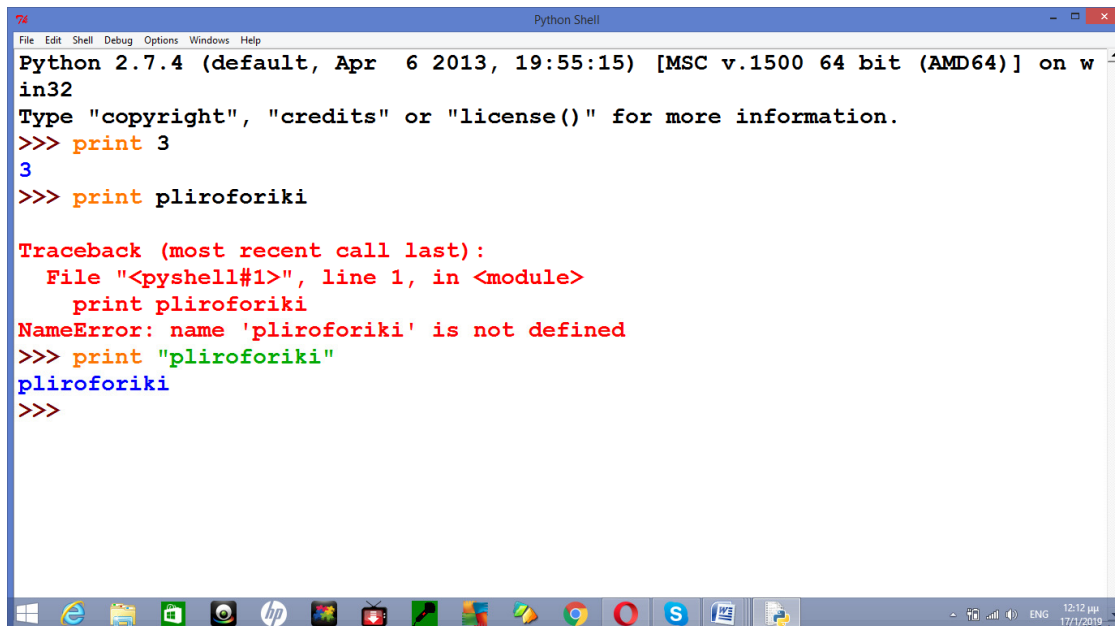
Αυτή είναι η τελευταία καρτέλα από το μενού της γλώσσας προγραμματισμού Python και μας δείχνει όλους τους σχετικούς τρόπους, που μπορούμε να αναζητήσουμε βοήθεια σε οποιοδήποτε στάδιο κι αν βρισκόμαστε. Περιέχει αναλυτικά πάρα πολλές πληροφορίες και είναι ιδιαίτερα προσιτή και ευκολονόητη.



Εικόνα 22 Έβδομη καρτέλα σε Πρόγραμμα Python Help

5.2.1 Παράδειγμα 1 print

Σε αυτό το παράδειγμα που ακολουθεί βλέπουμε την εντολή print στην Python. γράφοντας πριν και τον αριθμό 3 βλέπουμε ότι το πρόγραμμα το εκτελεί και πληκτρολογεί μετά τον αριθμό 3 ως αποτέλεσμα της εντολής πριν που δώσαμε προηγουμένως. στη συνέχεια διαπιστώνουμε ότι αν δώσουμε την εντολή print και τη λέξη πληροφορική, είπα στον δεν θα δώσω το αναμενόμενο αποτέλεσμα. Ο λόγος είναι γιατί στην Python με εντολή πριν προκειμένου να γραφτεί κείμενο θα πρέπει να είναι μέσα το κείμενο σε εισαγωγικά. το παράδειγμα μας στη συνέχεια γίνεται print εισαγωγικά πληροφορική εισαγωγικά και με τον τρόπο αυτό έχουμε το αποτέλεσμα το επιθυμητό.



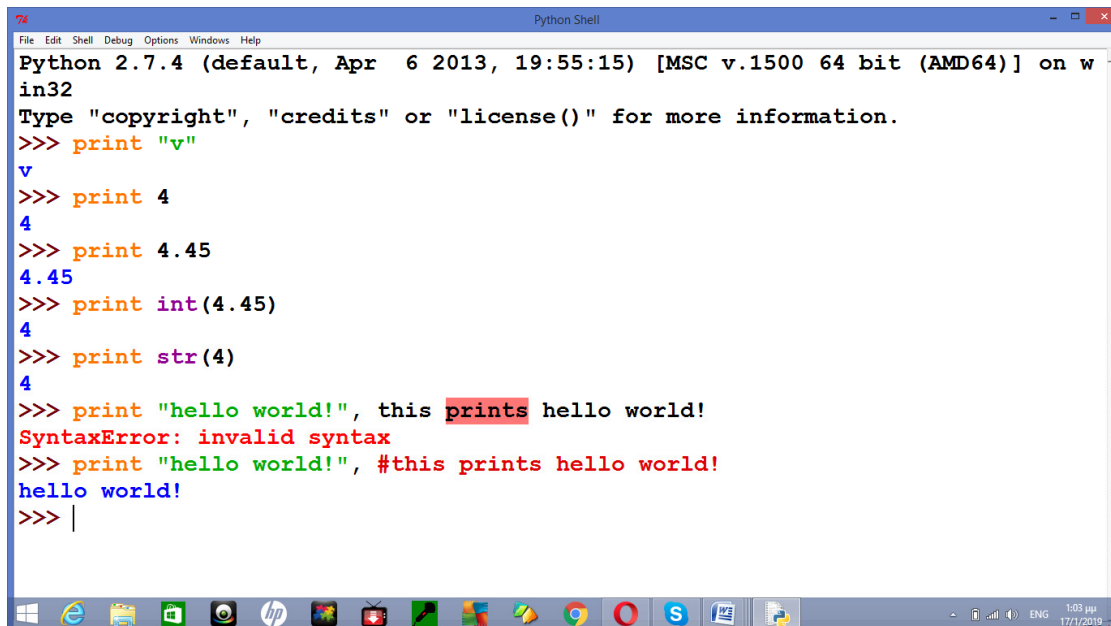
```
Python Shell
Python 2.7.4 (default, Apr 6 2013, 19:55:15) [MSC v.1500 64 bit (AMD64)] on w
in32
Type "copyright", "credits" or "license()" for more information.
>>> print 3
3
>>> print pliroforiki

Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    print pliroforiki
NameError: name 'pliroforiki' is not defined
>>> print "pliroforiki"
pliroforiki
>>>
```

Εικόνα 23 Παράδειγμα 1 print

5.2.2 Παράδειγμα 2 print, int, str, error, σχόλια.

διαπιστώνουμε επίσης ότι η γλώσσα προγραμματισμού Python όταν δέχεται λάθος σύνταξη κειμένου τότε επιστρέφει κόκκινα γράμματα με τη λέξη error. στο παράδειγμα νούμερο 2 βλέπουμε ότι μπορούμε να δώσουμε την εντολή print και να τυπώσουμε αριθμούς όπως το 4 ή να δώσουμε τη εντολή print 4,45 και να μας επιστραφεί το 4.45. όμως η γλώσσα προγραμματισμού Python Έχει και άλλες εντολές όπως είπαμε στο κεφάλαιο 3 για στο συγκεκριμένο Λοιπόν παράδειγμα μπορούμε να δώσουμε την εντολή print int 4,45 και να μας επιστρέψει το πρόγραμμα το νούμερο 4 Αυτό γίνεται γιατί έτσι έχουμε δώσει την εντολή από όλο τον αριθμό να τυπώσει μόνο τον ακέραιο και όχι το δεκαδικό του τμήμα. αντίστοιχα όπως θα δούμε Η εντολή print string μαζί με το προηγούμενο αριθμό δηλαδή το 4 ότι είχε επαναληφθεί επαναλαμβάνεται και τώρα. επιπλέον μπορούμε να διαπιστώσουμε ότι όταν γράφουμε ένα σχόλιο στην Python το γράφουμε με τη δίσωση όταν γράφουμε το σύμβολο της δίσωσης το πρόγραμμα δέχεται τα σχόλια τα σχόλια μπορούμε να τα γράψω προκειμένου να ερμηνεύσουμε κάποιο κώδικα ή κάποια εντολή.

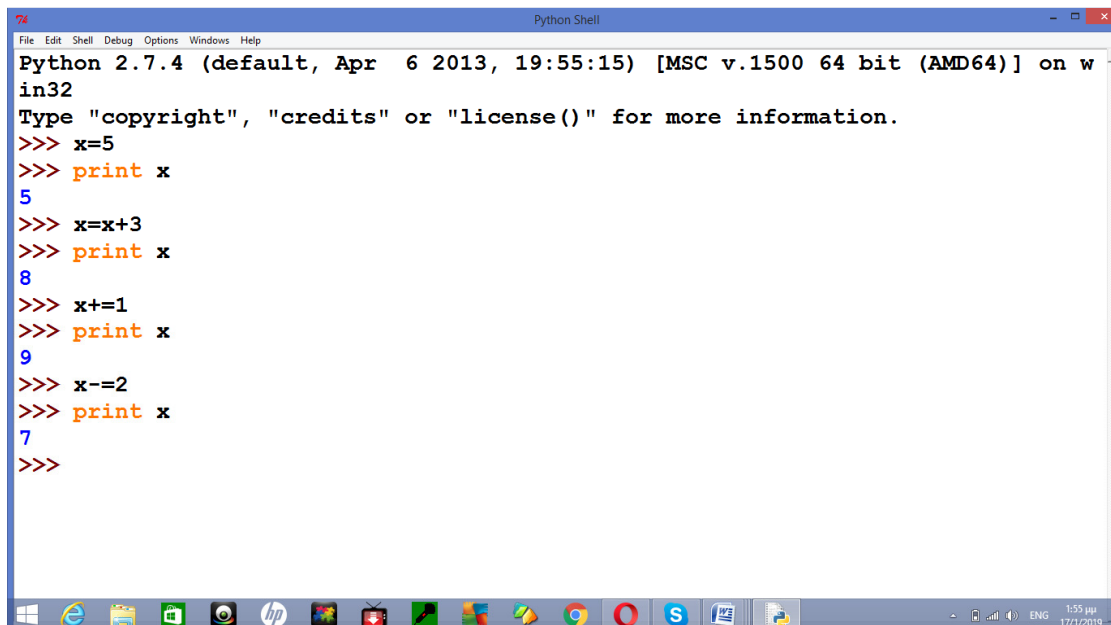


```
Python Shell
Python 2.7.4 (default, Apr 6 2013, 19:55:15) [MSC v.1500 64 bit (AMD64)] on w
in32
Type "copyright", "credits" or "license()" for more information.
>>> print "v"
v
>>> print 4
4
>>> print 4.45
4.45
>>> print int(4.45)
4
>>> print str(4)
4
>>> print "hello world!", this prints hello world!
SyntaxError: invalid syntax
>>> print "hello world!", #this prints hello world!
hello world!
>>> |
```

Εικόνα 24 Παράδειγμα 2 print,int, str,error, σχόλια

5.2.3 Παράδειγμα 3 print x

Στο παράδειγμα 3 βλέπουμε, πως μπορούμε να θέσουμε οτιδήποτε σαν μεταβλητή Έστω ότι έχουμε το $X = 5$ αν πληκτρολογήσουμε print μας δίνει αποτέλεσμα τον αριθμό που θέσαμε ίσο με τη μεταβλητή δηλαδή το 5. αντίστοιχα βλέπουμε πως αν πληκτρολογήσουμε print x+3 θα μας δώσει τον αριθμό 8.

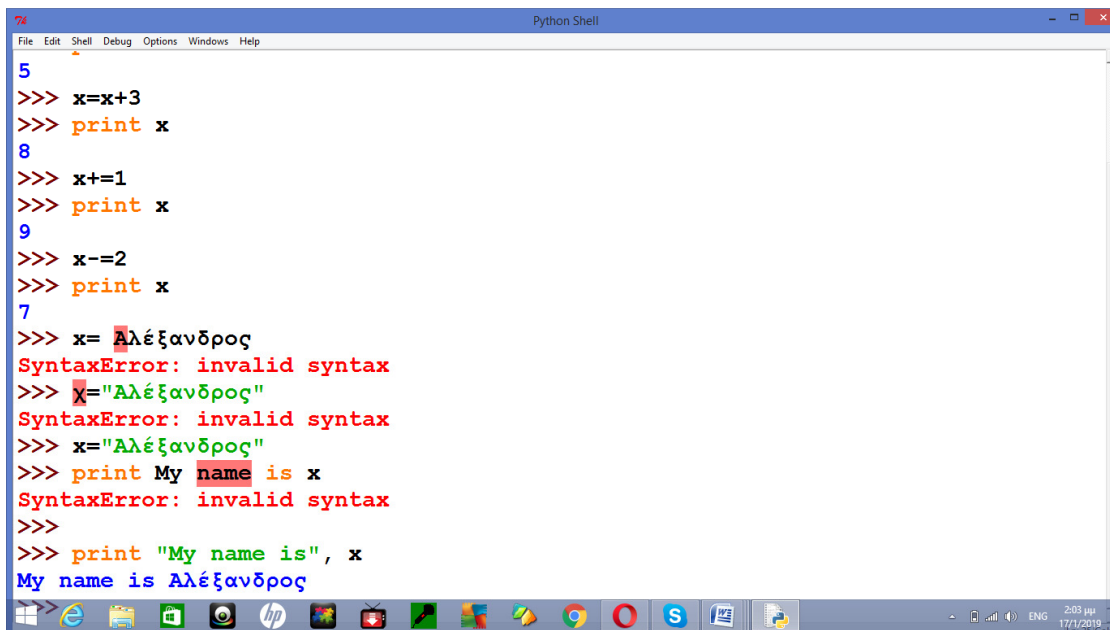


```
Python Shell
Python 2.7.4 (default, Apr 6 2013, 19:55:15) [MSC v.1500 64 bit (AMD64)] on w
in32
Type "copyright", "credits" or "license()" for more information.
>>> x=5
>>> print x
5
>>> x=x+3
>>> print x
8
>>> x+=1
>>> print x
9
>>> x-=2
>>> print x
7
>>>
```

Εικόνα 25 Παράδειγμα 3 print x

5.2.2 Παράδειγμα 4 *print, syntax error, μεταβλητή*

Στο επόμενο παράδειγμα δηλαδή το 4 βλέπουμε πως αν θέσουμε τη μεταβλητή X ίση με το όνομα Αλέξανδρος μέσα σε εισαγωγικά, και δώσουμε την εντολή `print` μέσα σε εισαγωγικά `my name is` κλείνοντας τα εισαγωγικά δώσουμε και τη μεταβλητή X, θα μας επιστρέψει το πρόγραμμα το `My name is Αλέξανδρος`.⁵³



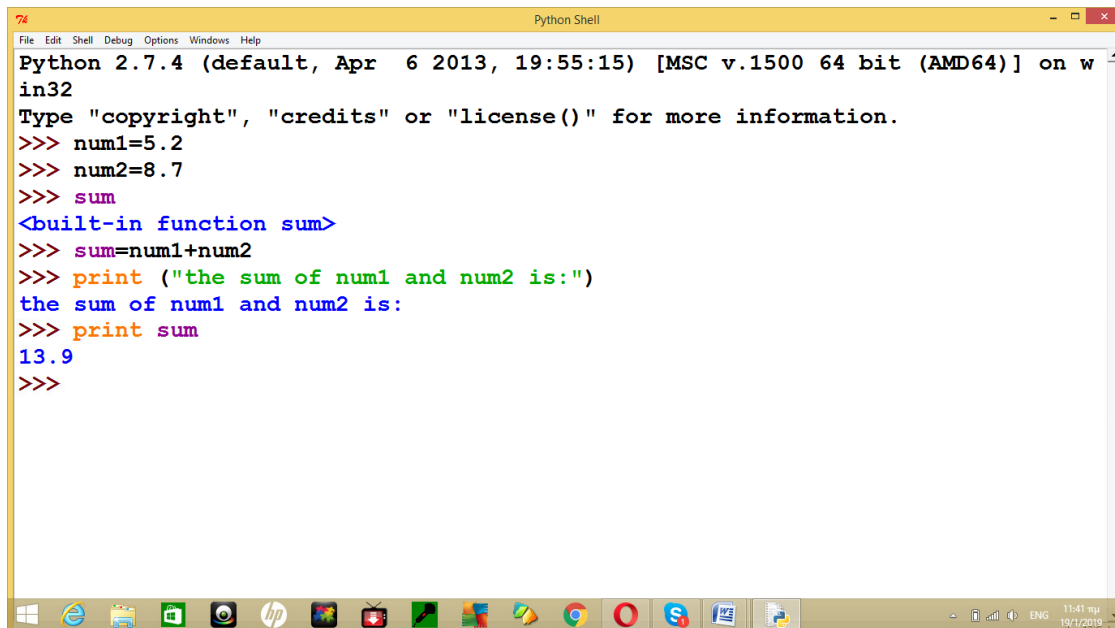
```
Python Shell
File Edit Shell Debug Options Windows Help
5
>>> x=x+3
>>> print x
8
>>> x+=1
>>> print x
9
>>> x-=2
>>> print x
7
>>> x= Αλέξανδρος
SyntaxError: invalid syntax
>>> x="Αλέξανδρος"
SyntaxError: invalid syntax
>>> x="Αλέξανδρος"
>>> print My name is x
SyntaxError: invalid syntax
>>>
>>> print "My name is", x
My name is Αλέξανδρος
```

Εικόνα 26 Παράδειγμα 4 *print, syntax error, μεταβλήτη*

5.2.5 Παράδειγμα 5 *sum, print sum*

Στο παράδειγμα 5 διαπιστώνουμε ότι αν δώσουμε σαν ένα νούμερο συνάρτησης `num 1` έναν αριθμό και αντίστοιχα `num 2` έναν άλλον αριθμό μπορούμε να χτίσουμε τη συνάρτηση `sum` που αποτελεί τη συνάρτηση του αθροίσματος και πληκτρολογώντας την εντολή `print` να μας επιστρέψει γλώσσα Python το αποτέλεσμα του αθροίσματος των δύο αριθμών που δώσαμε στην αρχή.

⁵³ Μάνης Γ., (2015), «Εισαγωγή στον προγραμματισμό με αρωγό τη γλώσσα Python.» έκδοση: Σύνδεσμος ελληνικών ακαδημαϊκών βιβλιοθηκών. (pdf)



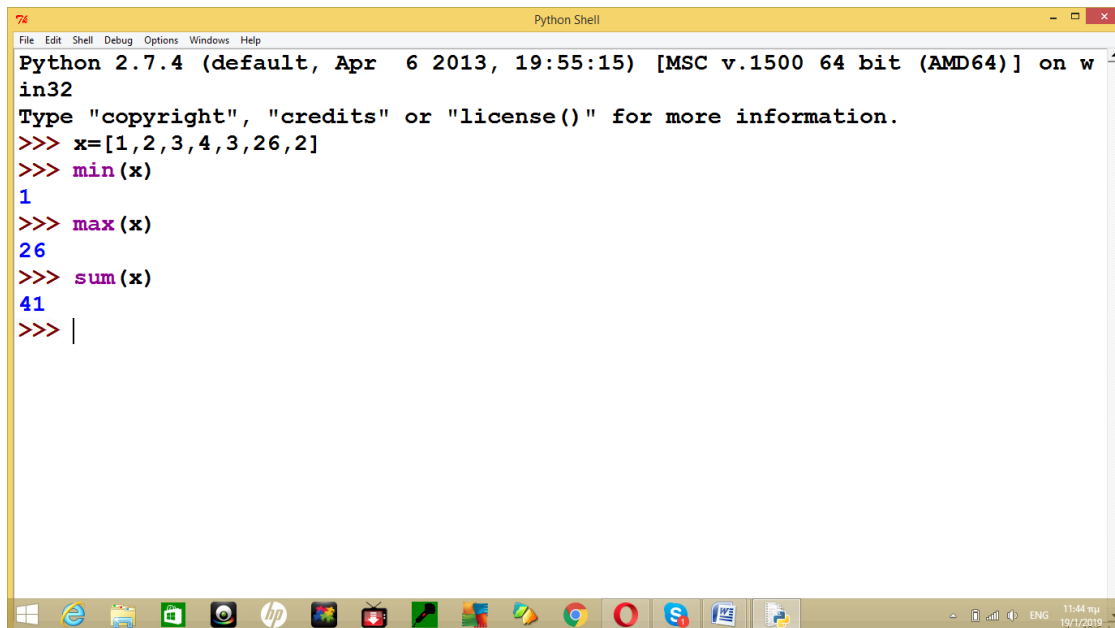
```
Python Shell
Python 2.7.4 (default, Apr 6 2013, 19:55:15) [MSC v.1500 64 bit (AMD64)] on w
in32
Type "copyright", "credits" or "license()" for more information.
>>> num1=5.2
>>> num2=8.7
>>> sum
<built-in function sum>
>>> sum=num1+num2
>>> print ("the sum of num1 and num2 is:")
the sum of num1 and num2 is:
>>> print sum
13.9
>>>
```

Εικόνα 27 Παράδειγμα 5 sum, print sum

5.2.6 Παράδειγμα 6 , συνάρτηση min, max

Στο παράδειγμα 6 βλέπουμε ότι μπορούμε αντίστοιχα να δώσουμε τις συναρτήσεις min και Max ανάμεσα σε κάποιες μεταβλητές ή αριθμούς και να μας δώσουν το αντίστοιχο Επιθυμητό αποτέλεσμα του ελάχιστου και του μέγιστου.⁵⁴

⁵⁴ Λεβαδεας Δ., (2010), «Οδηγός Python μέσω παραδειγμάτων.», έκδοση: ομάδα TasPython. (pdf)

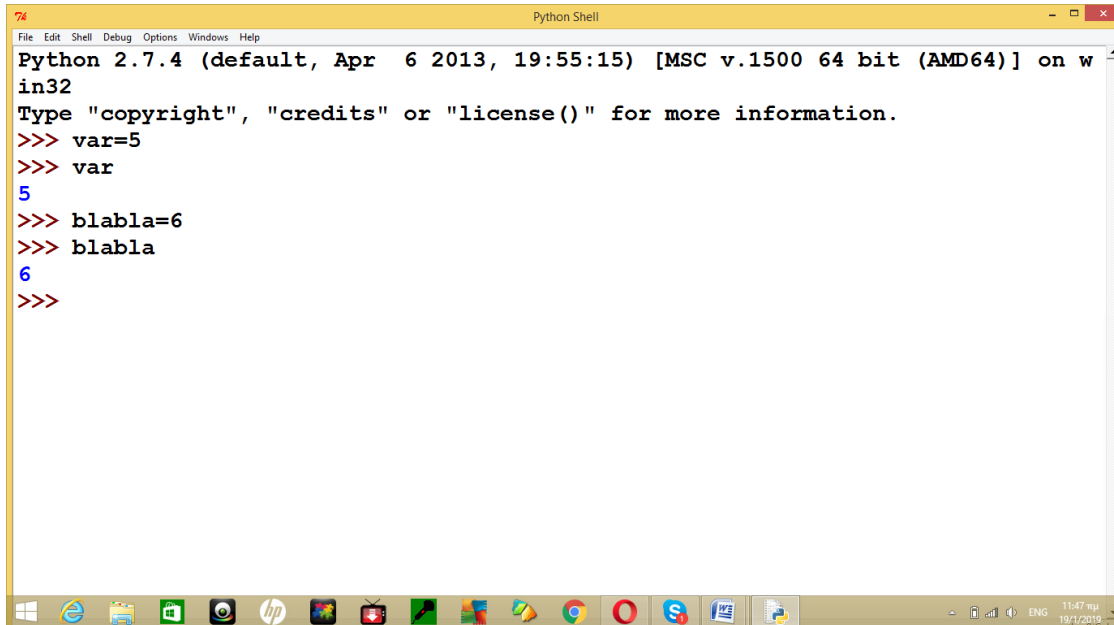


```
Python Shell
Python 2.7.4 (default, Apr 6 2013, 19:55:15) [MSC v.1500 64 bit (AMD64)] on w
in32
Type "copyright", "credits" or "license()" for more information.
>>> x=[1,2,3,4,3,26,2]
>>> min(x)
1
>>> max(x)
26
>>> sum(x)
41
>>> |
```

Εικόνα 28 Παράδειγμα 6 συναρτήσεων min, max, sum

5.2.7 Παράδειγμα 7 , μεταβλητή -var

Στο παράδειγμα 7 διαπιστώνουμε ότι μπορούμε να δώσουμε ως εντολή var, δηλαδή μεταβλητή οποιοδήποτε όνομα μπορούμε να δώσουμε τον αριθμό , μπορεί να δώσουμε ακόμα και οτιδήποτε θέλουμε εμείς σαν μεταβλητή ακόμα και κείμενο.

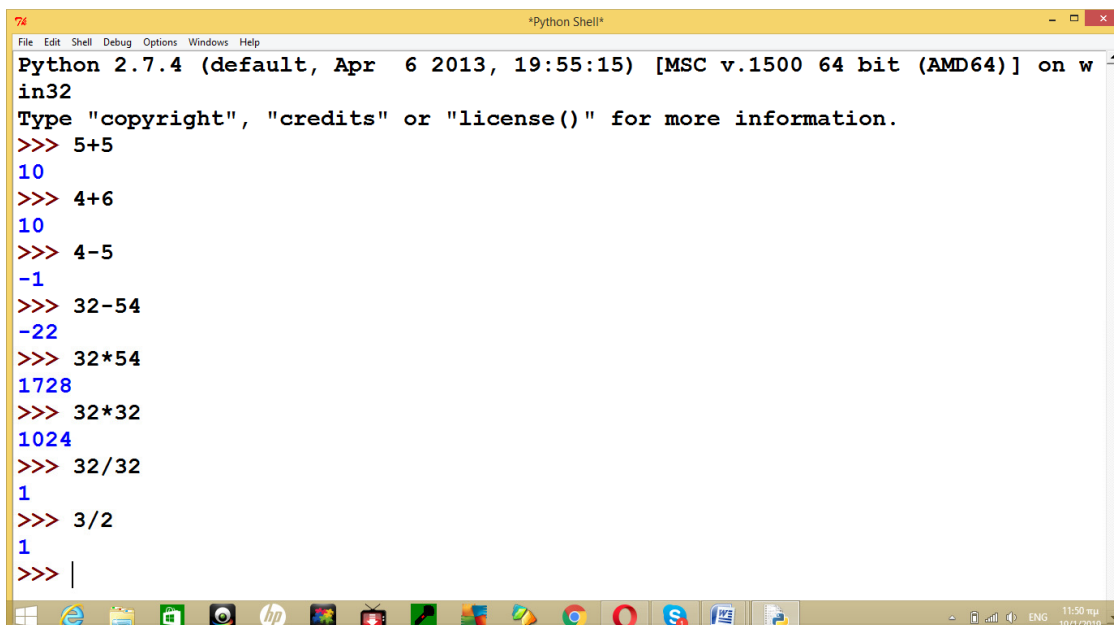


```
Python 2.7.4 (default, Apr 6 2013, 19:55:15) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> var=5
>>> var
5
>>> blabla=6
>>> blabla
6
>>>
```

Εικόνα 29 Παράδειγμα 7, μεταβλητή var

5.2.8 Παράδειγμα 8, μαθηματικές πράξεις

Στο παράδειγμα 8 διαπιστώνουμε ότι μπορούμε να κάνουμε οποιεσδήποτε μαθηματικές πράξεις όπως πρόσθεση-αφαίρεση πολλαπλασιασμό και διαίρεση.

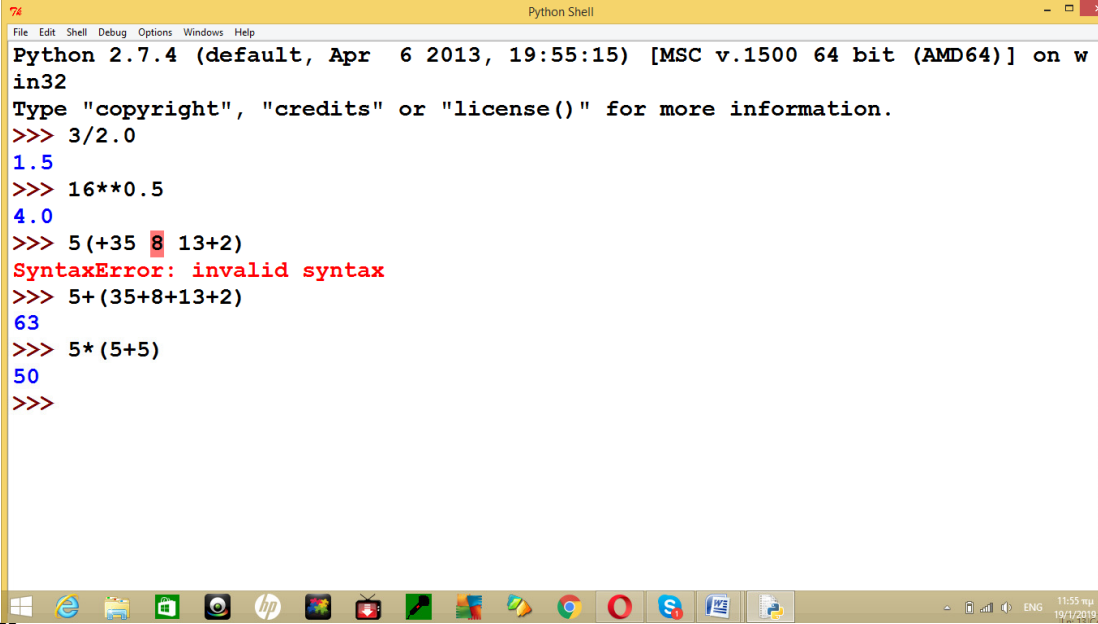


```
Python 2.7.4 (default, Apr 6 2013, 19:55:15) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 5+5
10
>>> 4+6
10
>>> 4-5
-1
>>> 32-54
-22
>>> 32*54
1728
>>> 32*32
1024
>>> 32/32
1
>>> 3/2
1
>>> |
```

Εικόνα 30 Παράδειγμα 8, μαθηματικές πράξεις

5.2.9 Παράδειγμα 9, μαθηματικές πράξεις και λάθη

Στο παράδειγμα 9 διαπιστώνουμε ότι για να κάνουμε διάφορες πράξεις θα πρέπει να μην υπάρχουν κενά ανάμεσα σε αριθμητικές παραστάσεις και να υπάρχουν και παρενθέσεις.



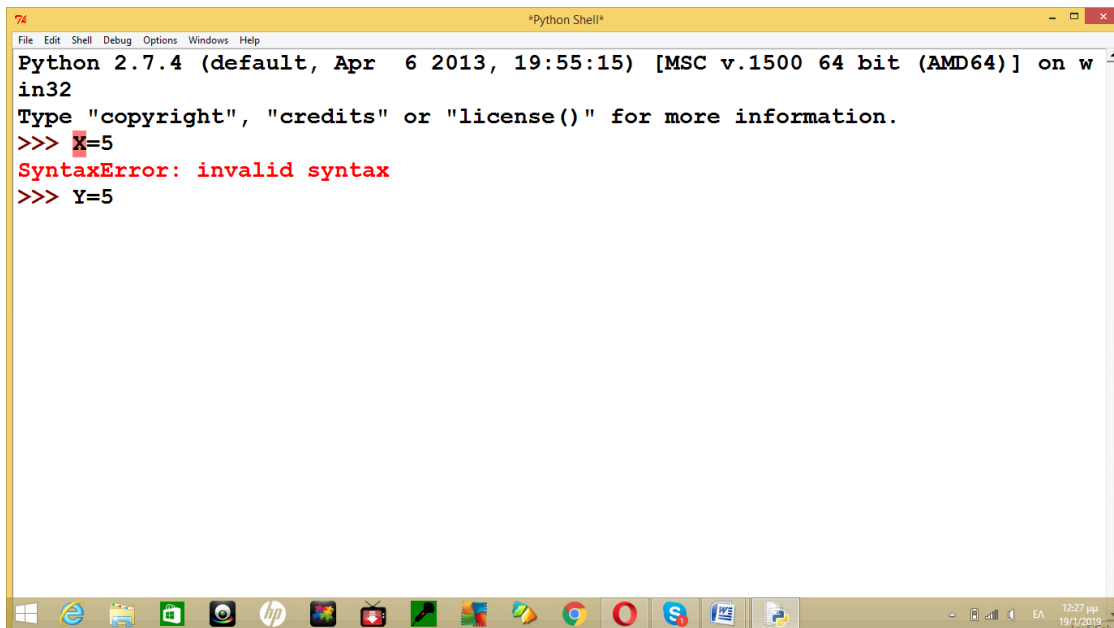
```
Python Shell
Python 2.7.4 (default, Apr 6 2013, 19:55:15) [MSC v.1500 64 bit (AMD64)] on w
in32
Type "copyright", "credits" or "license()" for more information.
>>> 3/2.0
1.5
>>> 16**0.5
4.0
>>> 5(+35 8 13+2)
SyntaxError: invalid syntax
>>> 5+(35+8+13+2)
63
>>> 5*(5+5)
50
>>>
```

Εικόνα 31 Παράδειγμα 9 μαθηματικές πράξεις και λάθη

5.2.10 Παράδειγμα 10, λάθη

στο παράδειγμα 10 διαπιστώνουμε λάθη σε σύνταξη.

⁵⁵ <https://www.openbook.gr/eisagwgi-ston-antikeimenostrafi-programmatismo-me-python/>

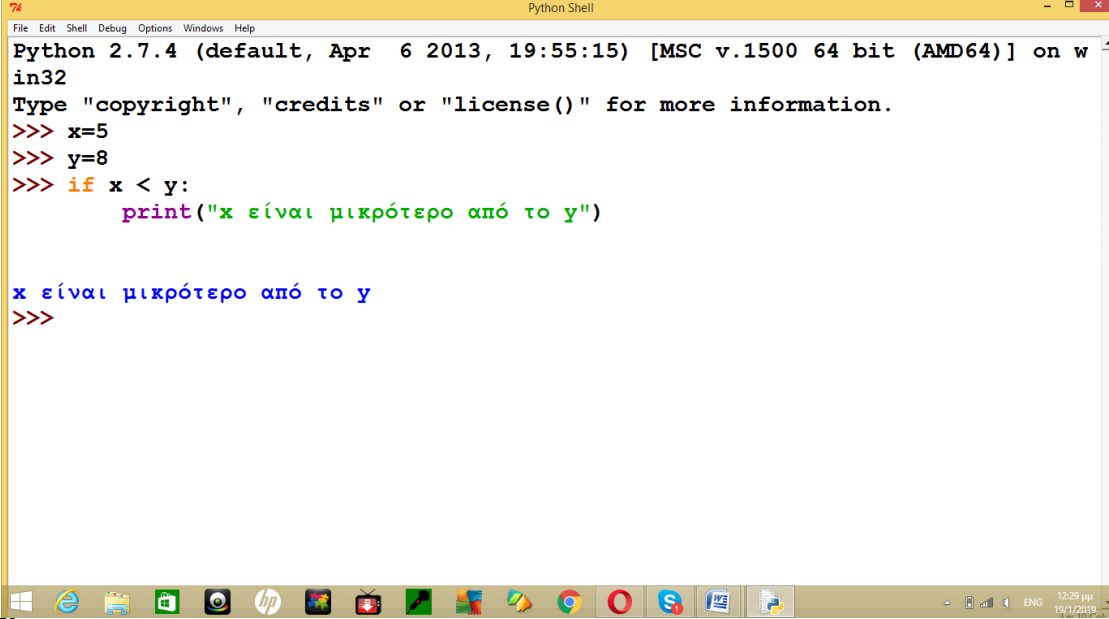


```
Python Shell
File Edit Shell Debug Options Windows Help
Python 2.7.4 (default, Apr 6 2013, 19:55:15) [MSC v.1500 64 bit (AMD64)] on w
in32
Type "copyright", "credits" or "license()" for more information.
>>> X=5
SyntaxError: invalid syntax
>>> Y=5
```

Εικόνα 32 Παράδειγμα 10 λάθη σύνταξης

5.2.11 Παράδειγμα 11, συνάρτηση if

Στο παράδειγμα 11 χρησιμοποιούμε τη συνάρτηση if δίνοντας τη μεταβλητή $X = 5$ και $\psi = 8$ θέλουμε την εντολή print ώστε όταν το X είναι μικρότερο από το ψ θα μας επιστρέψει το πρόγραμμα το αντίστοιχο αποτέλεσμα.



The screenshot shows a Python Shell window with the following content:

```
Python 2.7.4 (default, Apr 6 2013, 19:55:15) [MSC v.1500 64 bit (AMD64)] on w
in32
Type "copyright", "credits" or "license()" for more information.
>>> x=5
>>> y=8
>>> if x < y:
        print("x είναι μικρότερο από το y")

x είναι μικρότερο από το y
>>>
```

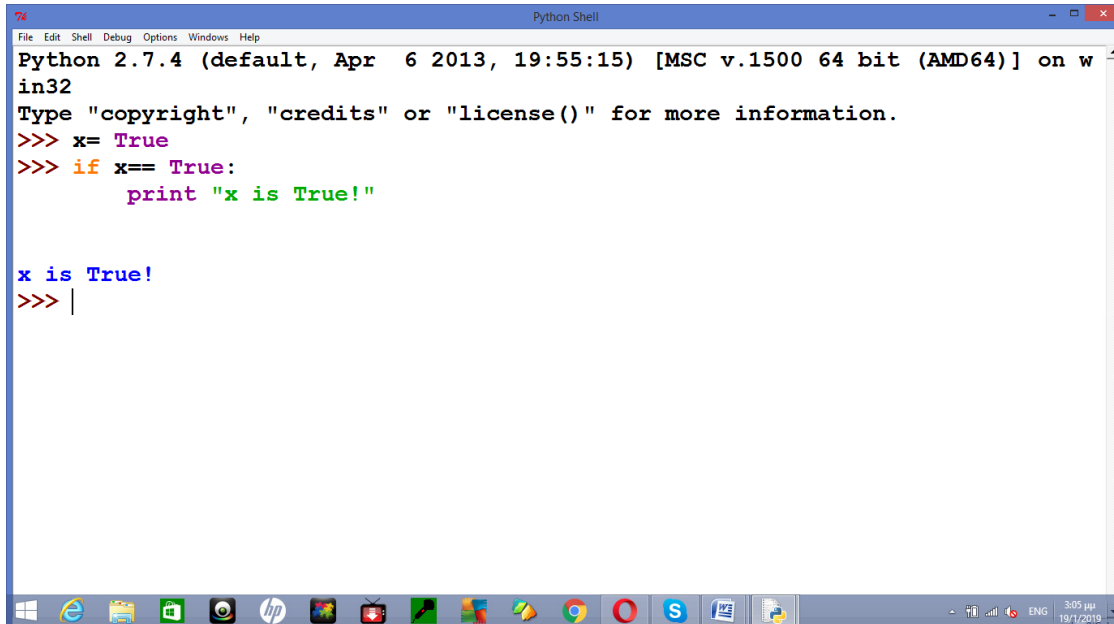
The window title is "Python Shell" and the taskbar at the bottom shows various application icons and the system clock.

Εικόνα 33 Παράδειγμα 11 συνάρτηση if

5.2.12 Παράδειγμα 12 συνάρτηση if και συνθήκη True

Στο παράδειγμα 12 χρησιμοποιούμε τη συνάρτηση if και τη συνθήκη True, για να μας επιστρέψει το πρόγραμμα κάτι που είναι αληθές.

⁵⁶ <https://www.openbook.gr/odigos-python-mesw-paradeigmatwn/>



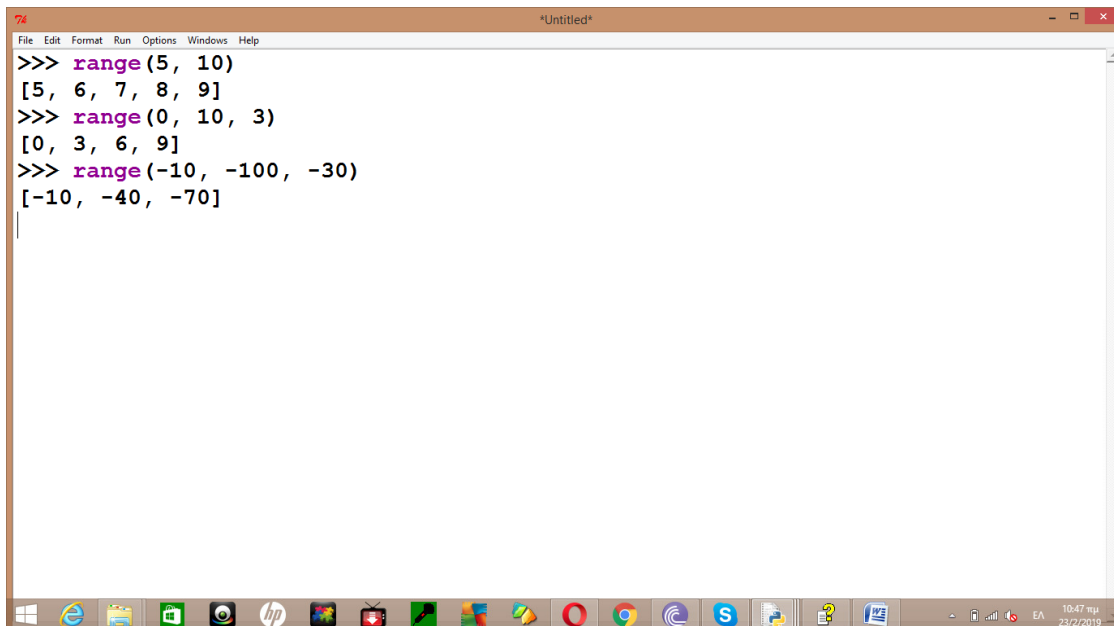
```
Python Shell
File Edit Shell Debug Options Windows Help
Python 2.7.4 (default, Apr 6 2013, 19:55:15) [MSC v.1500 64 bit (AMD64)] on w
in32
Type "copyright", "credits" or "license()" for more information.
>>> x= True
>>> if x== True:
    print "x is True!"

x is True!
>>> |
```

Εικόνα 34 Παράδειγμα 12 συνάρτηση if -true

5.2.13 Παράδειγμα 13, συνάρτηση range

Στο παράδειγμα 13 χρησιμοποιούμε τη συνάρτηση range.

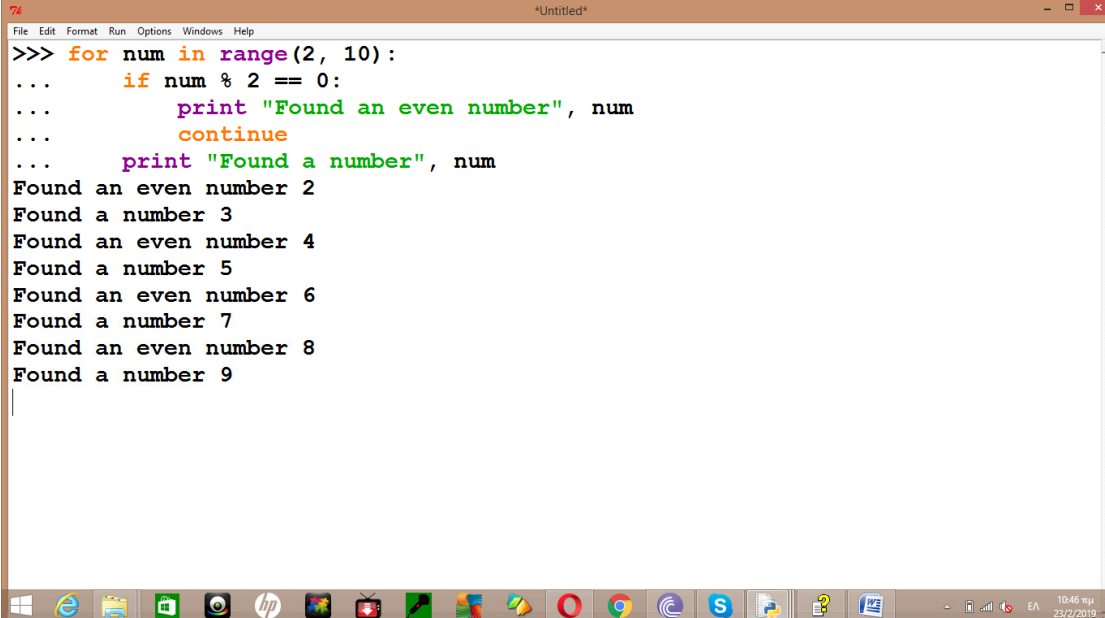


```
*Untitled*
File Edit Format Run Options Windows Help
>>> range(5, 10)
[5, 6, 7, 8, 9]
>>> range(0, 10, 3)
[0, 3, 6, 9]
>>> range(-10, -100, -30)
[-10, -40, -70]
```

Εικόνα 35 Παράδειγμα 13 συνάρτηση range

5.2.14 Παράδειγμα 14, συνάρτηση for, in, range

Στο παράδειγμα 14 χρησιμοποιούμε τη συνάρτηση for, in , range, ανάμεσα στα νούμερα (2, 10).



```
>>> for num in range(2, 10):
...     if num % 2 == 0:
...         print "Found an even number", num
...         continue
...     print "Found a number", num
Found an even number 2
Found a number 3
Found an even number 4
Found a number 5
Found an even number 6
Found a number 7
Found an even number 8
Found a number 9
```

57

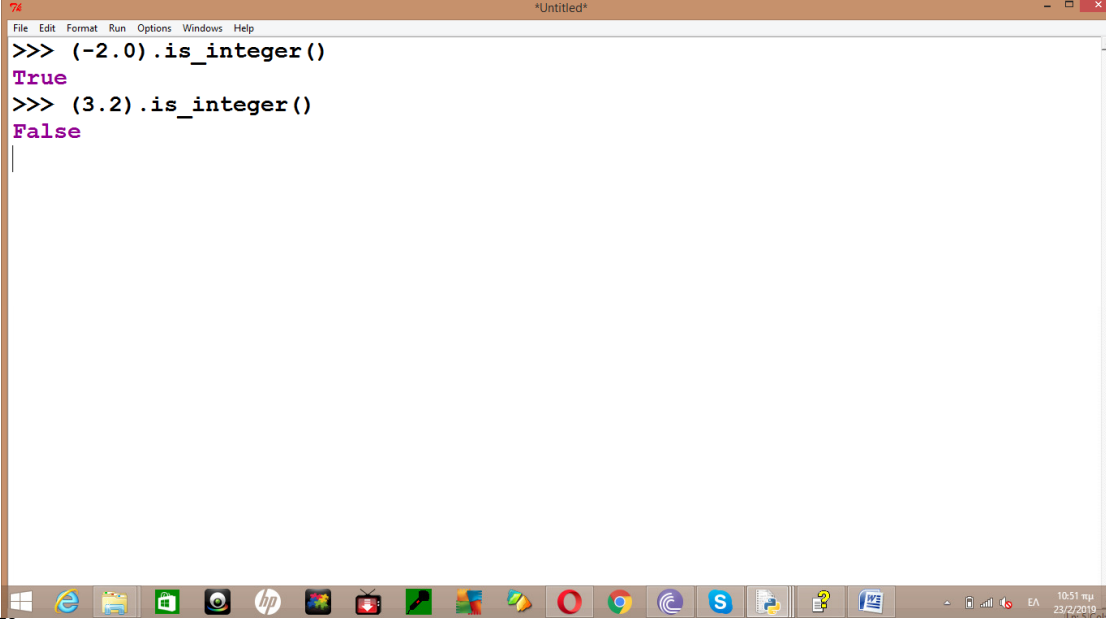
Εικόνα 36 Παράδειγμα 14 συνάρτηση for , in range

5.2.15 Παράδειγμα 15 συνθήκη αληθή ή ψευδή

Στο παράδειγμα 15 χρησιμοποιούμε συνθήκη αληθή ή ψευδή, ανάμεσα στους αριθμούς (2,0).⁵⁸

⁵⁷ <https://www.openbook.gr/odigos-python-mesw-paradeigmatwn/>

⁵⁸ <https://www.openbook.gr/eisagwgi-ston-antikeimenostrafi-programmatismo-me-python/>



```
File Edit Format Run Options Windows Help
>>> (-2.0).is_integer()
True
>>> (3.2).is_integer()
False
```

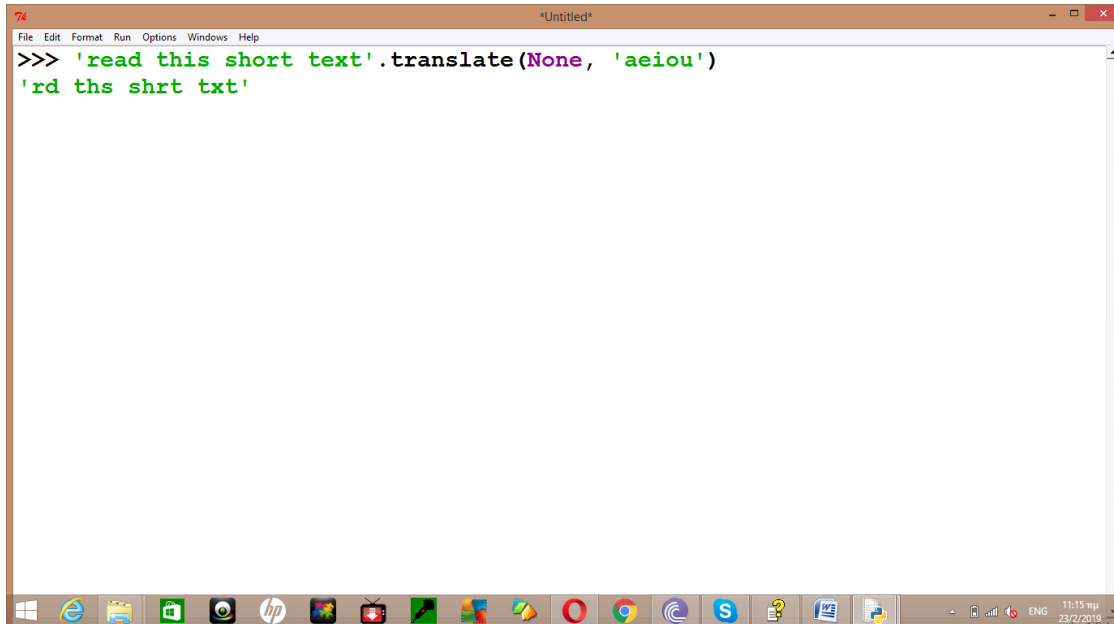
59

Εικόνα 37 Παράδειγμα 15 συνθήκη αληθής ή ψευδής

5.2.16 Παράδειγμα 16, *translate*

Στο παράδειγμα 16 χρησιμοποιούμε την `translate` προκειμένου να μορφοποιήσουμε ένα κείμενο χωρίς φωνήεν.

⁵⁹ Λεβαδεας Δ., (2010), «Οδηγός Python μέσω παραδειγμάτων.», έκδοση: ομάδα TasPython. (pdf)

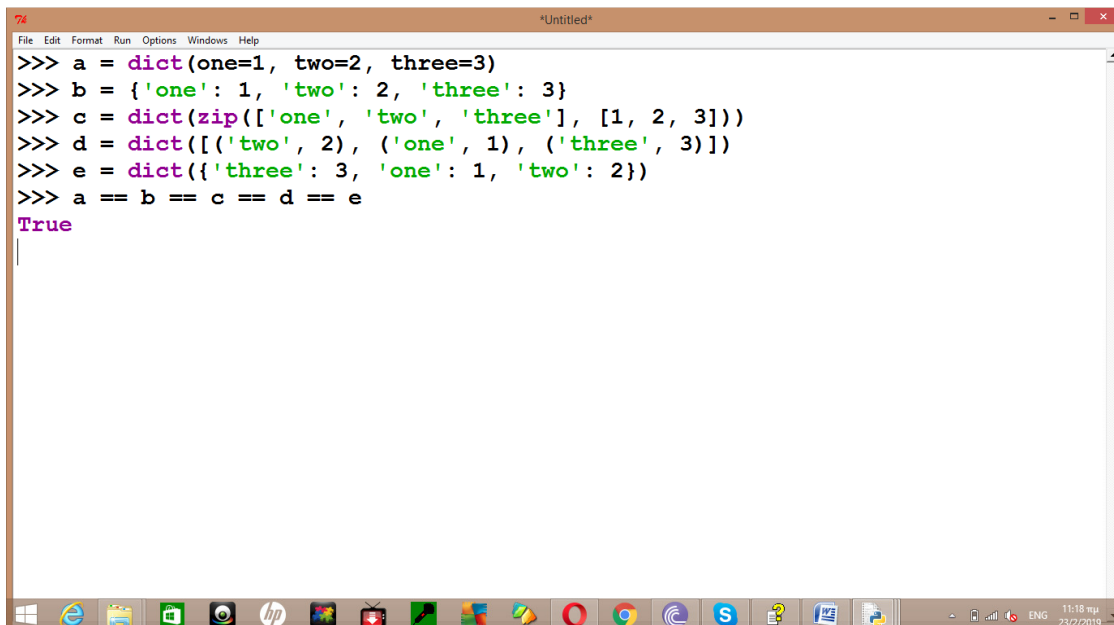


```
File Edit Format Run Options Windows Help
>>> 'read this short text'.translate(None, 'aeiou')
'rd ths shrt txt'
```

Εικόνα 38 Παράδειγμα 16 κείμενο χωρίς φωνήεν

5.2.17 Παράδειγμα 17, dict

Στο παράδειγμα 17 θέτουμε κάποια γράμματα για να φτιάξουμε ένα λεξικό.

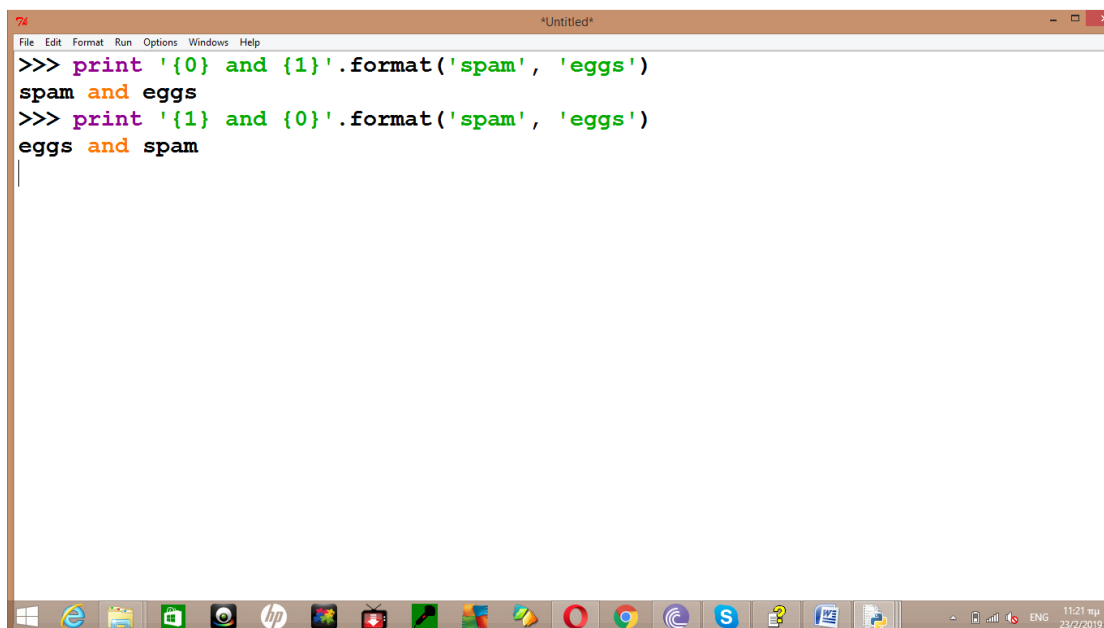


```
File Edit Format Run Options Windows Help
>>> a = dict(one=1, two=2, three=3)
>>> b = {'one': 1, 'two': 2, 'three': 3}
>>> c = dict(zip(['one', 'two', 'three'], [1, 2, 3]))
>>> d = dict([('two', 2), ('one', 1), ('three', 3)])
>>> e = dict({'three': 3, 'one': 1, 'two': 2})
>>> a == b == c == d == e
True
```

Εικόνα 39 Παράδειγμα 17 θέτωντας το dict

5.2.18 Παράδειγμα 18 and, print

Στο παράδειγμα 18 υπάρχει η συνάρτηση and καθώς και το αποτέλεσμα της με την εντολή print.



```
>>> print '{0} and {1}'.format('spam', 'eggs')
spam and eggs
>>> print '{1} and {0}'.format('spam', 'eggs')
eggs and spam
```

Εικόνα 40 Παράδειγμα 18 συνάρτηση and

5.2.19 Παράδειγμα 19, συνάρτηση if, else

Τέλος στο παράδειγμα 19, δείχνουμε τα σχόλια που αγνοούνται και τη συνάρτηση if σε συνδυασμό με τη συνάρτηση else, καθώς και το αποτέλεσμα τους με την εντολή print.

60

⁶⁰ <https://www.openbook.gr/odigos-python-mesw-paradeigmatwn/>

```
Python Shell
Python 2.7.4 (default, Apr 6 2013, 19:55:15) [MSC v.1500 64 bit (AMD64)] on w
in32
Type "copyright", "credits" or "license()" for more information.
>>> # comments are ignored
>>> x = 12
>>> x
12
>>> if x == 13:
...     print "yes"
... else:
...     print "no"
...     print "NO"
...     print "NO!!!"
...
no
NO
NO!!!
>>>
```

Εικόνα 41 Παράδειγμα 19 συνάρτηση if-else

5.2.20 Παράδειγμα 20

Στο παρακάτω παράδειγμα έχουμε πρόγραμμα, το οποίο μπορεί να βρει τα παραγοντικά σε μια συνάρτηση.

```
Untitled*
# Python Program to find the factors of a number

# define a function
def print_factors(x):
    # This function takes a number and prints the factors

    print("The factors of",x,"are:")
    for i in range(1, x + 1):
        if x % i == 0:
            print(i)

# change this value for a different result.
num = 320

# uncomment the following line to take input from the user
#num = int(input("Enter a number: "))

print_factors(num)|
```

Εικόνα 42 Παράδειγμα 20 factors of a number

61

⁶¹ <https://www.openbook.gr/odigos-python-mesw-paradeigmatwn/>

Το αποτέλεσμα είναι:

The factors of 320 are:

1

2

4

5

8

10

16

20

40

64

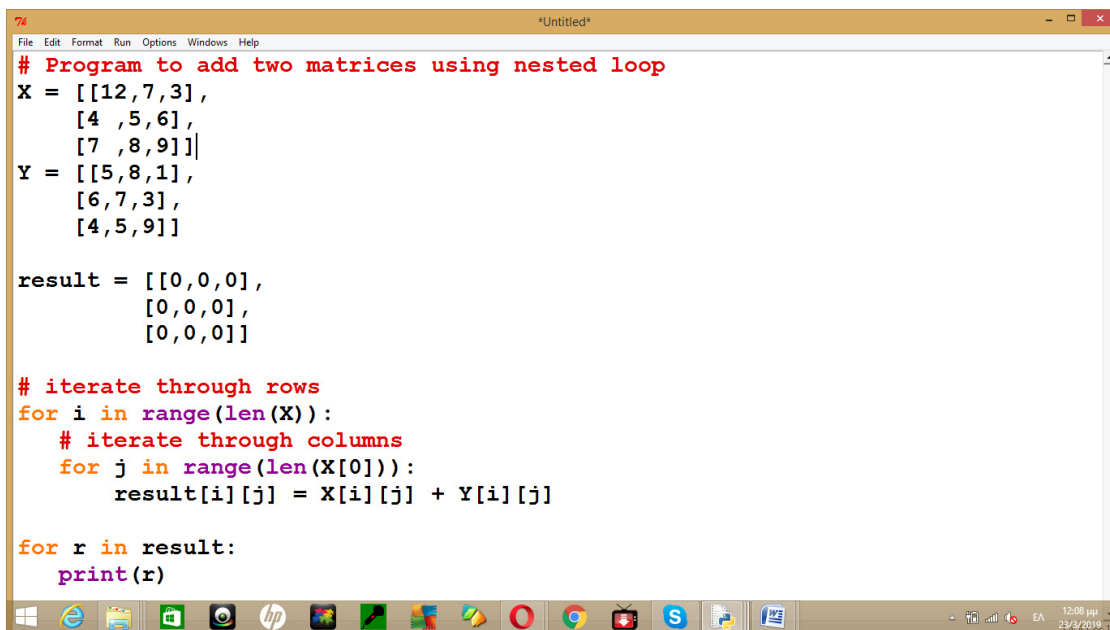
80

160

320

5.2.21 Παράδειγμα 21 program to add two matrices using nested loop

Στο παράδειγμα που ακολουθεί υπάρχει πρόγραμμα, που βρίσκει το άθροισμα δύο πινάκων.



```
# Program to add two matrices using nested loop
X = [[12,7,3],
     [4 ,5,6],
     [7 ,8,9]]
Y = [[5,8,1],
     [6,7,3],
     [4,5,9]]

result = [[0,0,0],
          [0,0,0],
          [0,0,0]]

# iterate through rows
for i in range(len(X)):
    # iterate through columns
    for j in range(len(X[0])):
        result[i][j] = X[i][j] + Y[i][j]

for r in result:
    print(r)
```

Εικόνα 43 Παράδειγμα 21 PROGRAM TO ADD TWO MATRICES

Το αποτέλεσμα είναι:

62

```
[17, 15, 4]

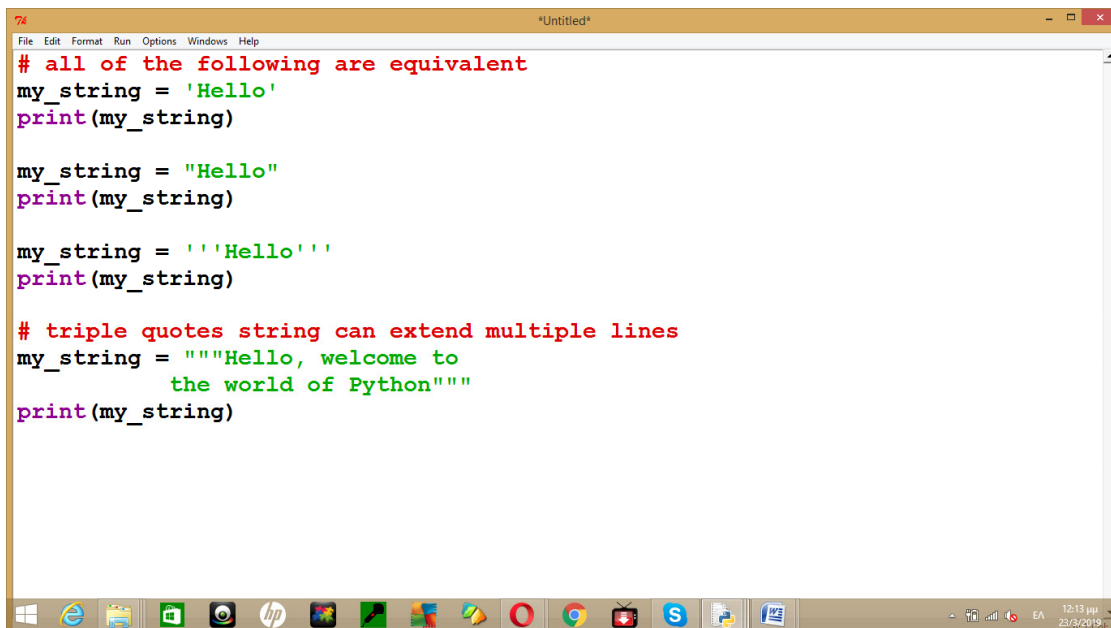
[10, 12, 9]

[11, 13, 18]
```

⁶² <https://www.openbook.gr/odigos-python-mesw-paradeigmatwn/>

5.2.2 Παράδειγμα 22 `print my_string`

Στο παράδειγμα αυτό έχουμε τύπωση της ακολουθίας



```
74
File Edit Format Run Options Windows Help
# all of the following are equivalent
my_string = 'Hello'
print(my_string)

my_string = "Hello"
print(my_string)

my_string = '''Hello'''
print(my_string)

# triple quotes string can extend multiple lines
my_string = """Hello, welcome to
               the world of Python"""
print(my_string)
```

Εικόνα 44 Παράδειγμα 22 `print my_string`

Το αποτέλεσμα είναι:

63

Hello

Hello

63 <https://www.openbook.gr/odigos-python-mesw-paradeigmatwn/>

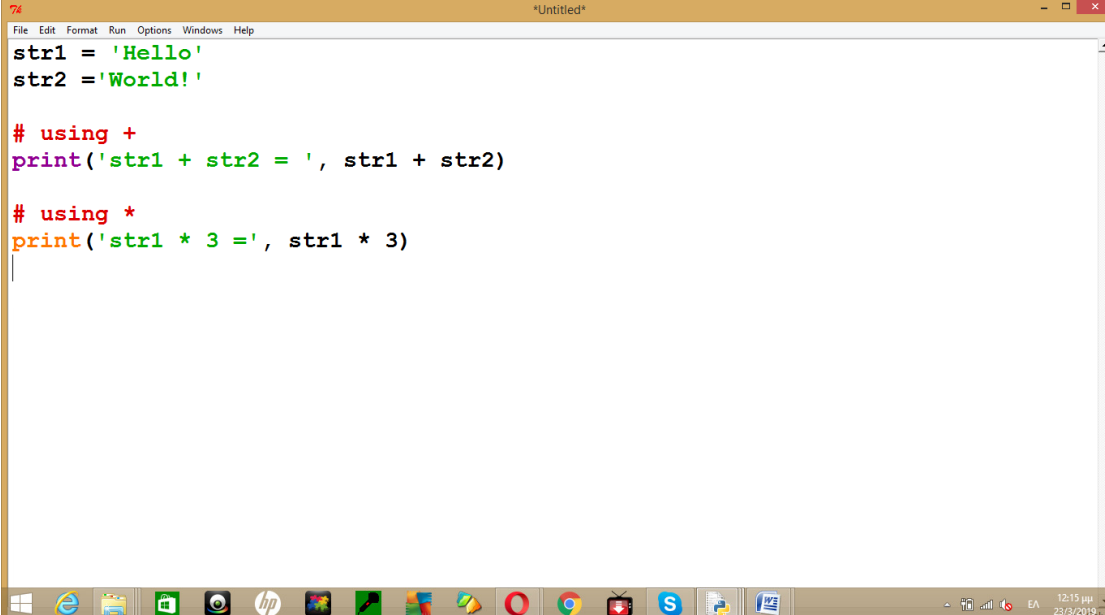
Hello

Hello, welcome to

the world of Python

5.2.23 Παράδειγμα 23 print string1, string2

Στο παράδειγμα αυτό έχουμε το άθροισμα 2 ακολουθιών.



```
str1 = 'Hello'
str2 = 'World!'

# using +
print('str1 + str2 = ', str1 + str2)

# using *
print('str1 * 3 =', str1 * 3)
```

Εικόνα 45 Παράδειγμα 23 print string1, string2 using +,*

Το αποτέλεσμα είναι:

```
str1 + str2 = HelloWorld!
```

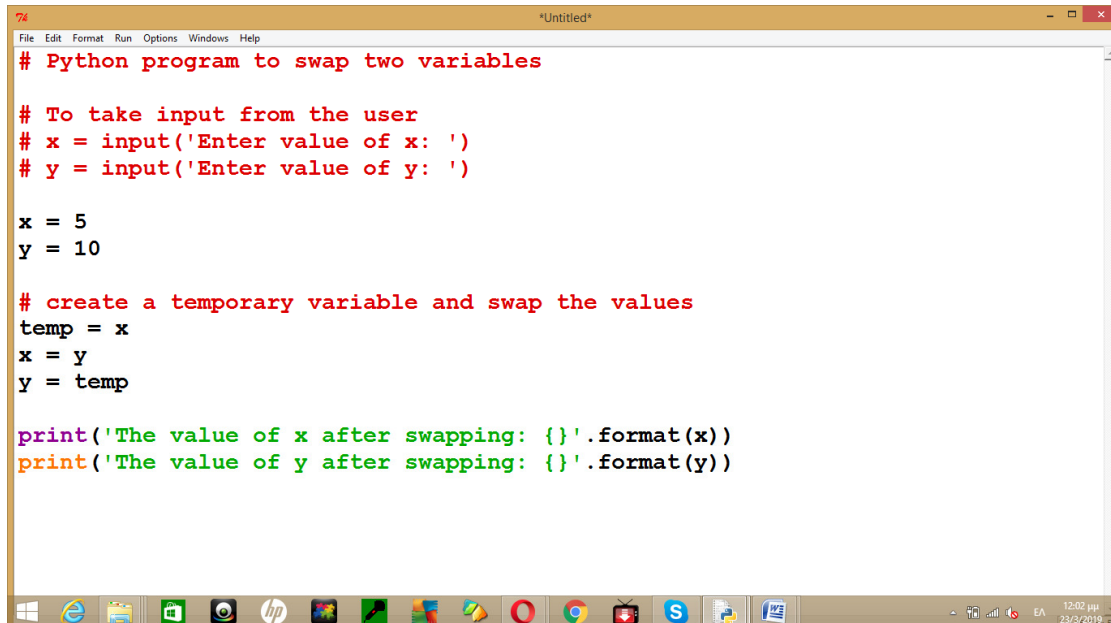
```
str1 * 3 = HelloHelloHello
```

64

⁶⁴ <https://www.openbook.gr/odigos-python-mesw-paradeigmatwn/>

5.2.24 Παράδειγμα 24 program to swap two variables

Στο παράδειγμα που ακολουθεί έχουμε την εναλλαγή ανάμεσα σε δύο μεταβλητές.



```
# Python program to swap two variables

# To take input from the user
# x = input('Enter value of x: ')
# y = input('Enter value of y: ')

x = 5
y = 10

# create a temporary variable and swap the values
temp = x
x = y
y = temp

print('The value of x after swapping: {}'.format(x))
print('The value of y after swapping: {}'.format(y))
```

Εικόνα 46 Παράδειγμα 24 program to swap two variables

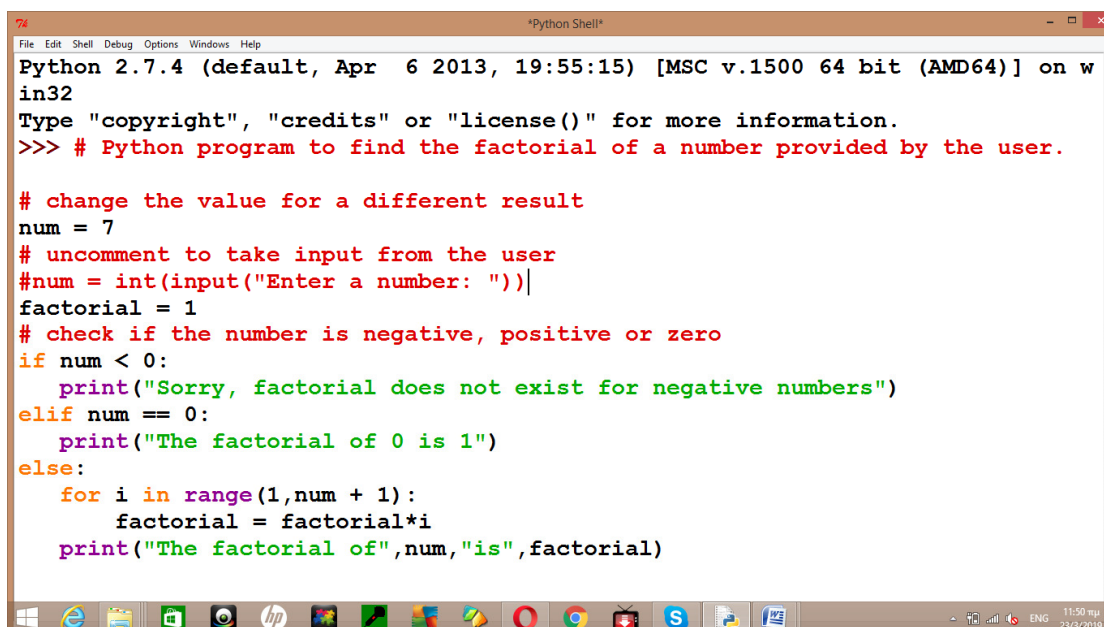
Το αποτέλεσμα είναι:

```
The value of x after swapping: 10
```

```
The value of y after swapping: 5
```

5.2.25 Παράδειγμα 25 find the factorial of a number

Το παρακάτω παράδειγμα μας δείχνει τα παραγοντικά σε ένα νούμερο.



```
Python 2.7.4 (default, Apr 6 2013, 19:55:15) [MSC v.1500 64 bit (AMD64)] on w
in32
Type "copyright", "credits" or "license()" for more information.
>>> # Python program to find the factorial of a number provided by the user.

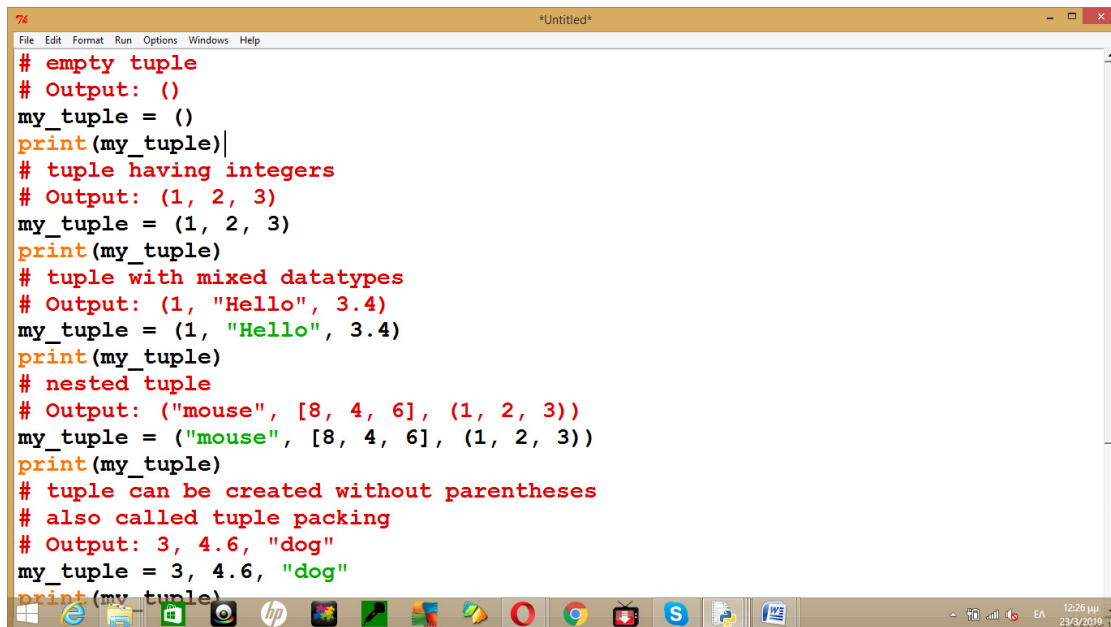
# change the value for a different result
num = 7
# uncomment to take input from the user
#num = int(input("Enter a number: "))
factorial = 1
# check if the number is negative, positive or zero
if num < 0:
    print("Sorry, factorial does not exist for negative numbers")
elif num == 0:
    print("The factorial of 0 is 1")
else:
    for i in range(1,num + 1):
        factorial = factorial*i
    print("The factorial of",num,"is",factorial)
```

Εικόνα 47 Παράδειγμα 25 find the factorial of a number

Το αποτέλεσμα είναι:

The factorial of 7 is 5040

5.2.26 Παράδειγμα 26 print my_tuple



```
File Edit Format Run Options Windows Help
# empty tuple
# Output: ()
my_tuple = ()
print(my_tuple)
# tuple having integers
# Output: (1, 2, 3)
my_tuple = (1, 2, 3)
print(my_tuple)
# tuple with mixed datatypes
# Output: (1, "Hello", 3.4)
my_tuple = (1, "Hello", 3.4)
print(my_tuple)
# nested tuple
# Output: ("mouse", [8, 4, 6], (1, 2, 3))
my_tuple = ("mouse", [8, 4, 6], (1, 2, 3))
print(my_tuple)
# tuple can be created without parentheses
# also called tuple packing
# Output: 3, 4.6, "dog"
my_tuple = 3, 4.6, "dog"
print(my_tuple)
```

Εικόνα 48 Παράδειγμα 26 print my_tuple

Το αποτέλεσμα είναι:

()

(1, 2, 3)

(1, 'Hello', 3.4)

('mouse', [8, 4, 6], (1, 2, 3))

(3, 4.6, 'dog')

3

4.6

dog

()

(1, 2, 3)

(1, 'Hello', 3.4)

('mouse', [8, 4, 6], (1, 2, 3))

(3, 4.6, 'dog')

3

4.6

dog

()

(1, 2, 3)

(1, 'Hello', 3.4)

('mouse', [8, 4, 6], (1, 2, 3))

(3, 4.6, 'dog')

3

4.6

dog

()

(1, 2, 3)

(1, 'Hello', 3.4)

('mouse', [8, 4, 6], (1, 2, 3))

(3, 4.6, 'dog')

3

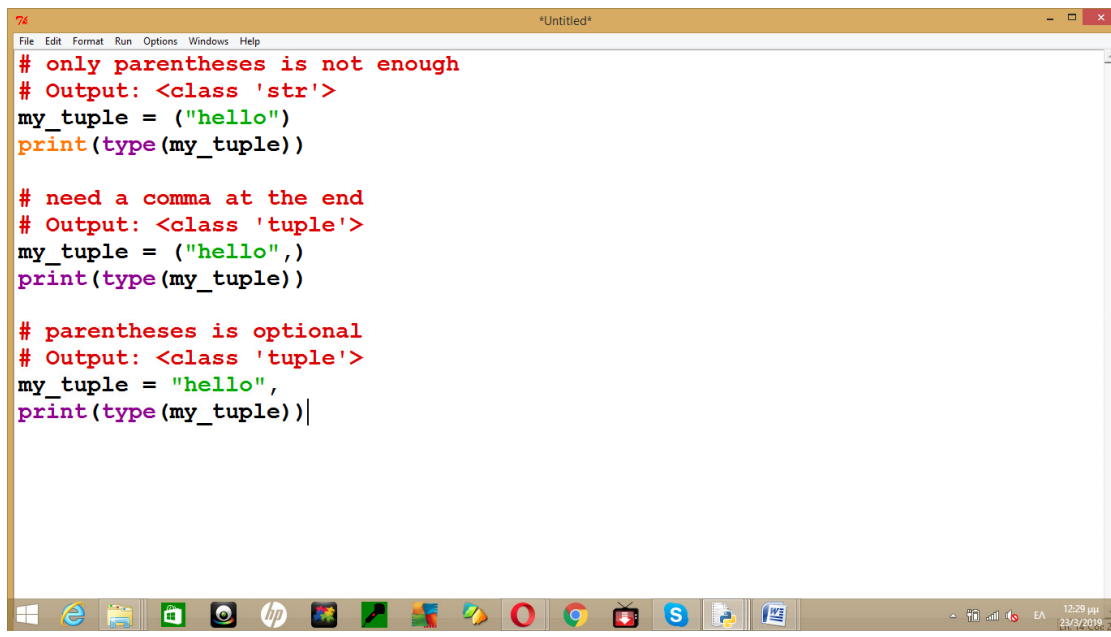
4.6

dog

67

⁶⁷ <https://www.openbook.gr/odigos-python-mesw-paradeigmatwn/>

5.2.27 Παράδειγμα 27 class string, class tuple



```
File Edit Format Run Options Windows Help
# only parentheses is not enough
# Output: <class 'str'>
my_tuple = ("hello")
print(type(my_tuple))

# need a comma at the end
# Output: <class 'tuple'>
my_tuple = ("hello",)
print(type(my_tuple))

# parentheses is optional
# Output: <class 'tuple'>
my_tuple = "hello",
print(type(my_tuple))
```

Εικόνα 49 Παράδειγμα 27 class string, class tuple

Το αποτέλεσμα είναι:

<class 'str'>

<class 'tuple'>

<class 'tuple'>

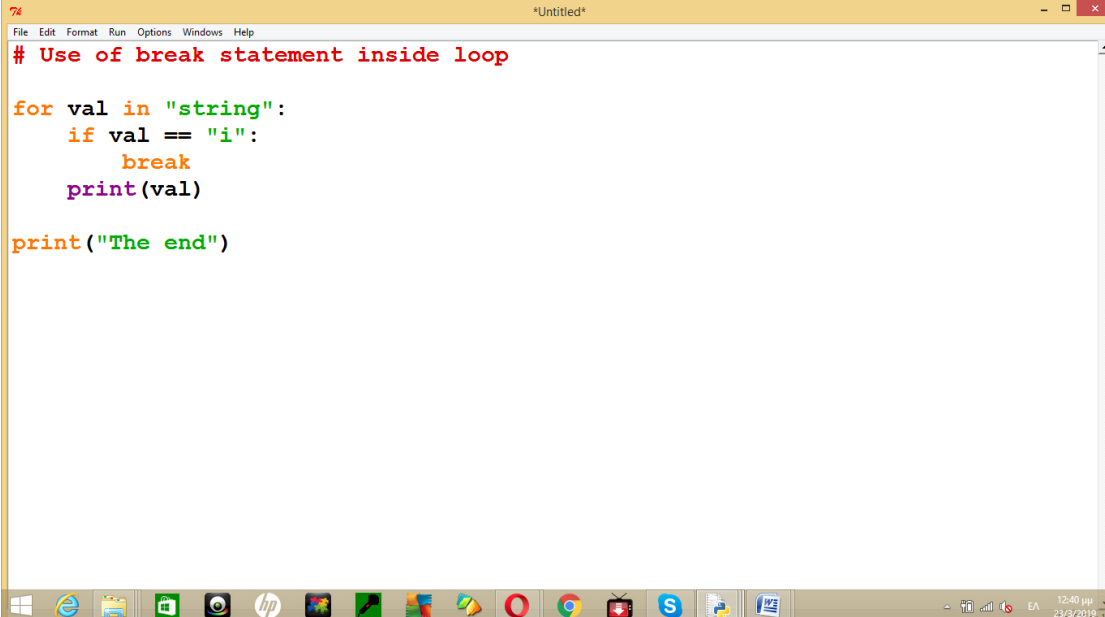
<class 'str'>

<class 'tuple'>

<class 'tuple'>⁶⁸

⁶⁸<http://astronomi.erciyes.edu.tr/wp-content/uploads/astronom/pdf/Python%20-%20Andrew%20Johansen.pdf>

5.2.28 Παράδειγμα 28 use of break statement inside loop



```
File Edit Format Run Options Windows Help
# Use of break statement inside loop

for val in "string":
    if val == "i":
        break
    print(val)

print("The end")
```

Εικόνα 50 Παράδειγμα 48 use of break statement inside loop

Το αποτέλεσμα είναι:

s

t

r

The end

s

t

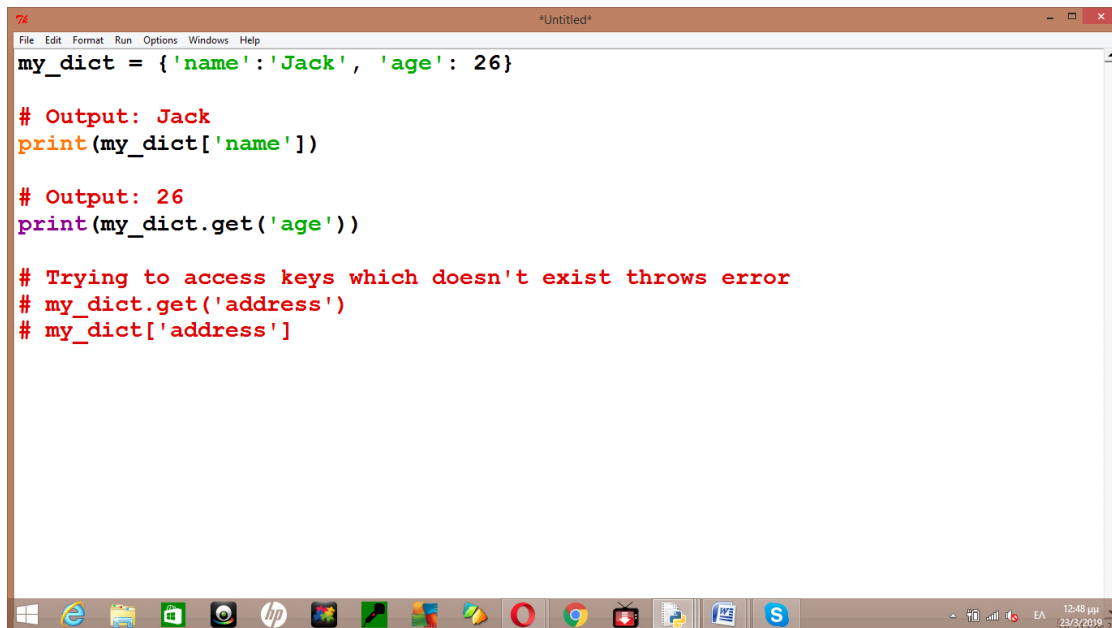
r

The end⁶⁹

⁶⁹ https://www.davekuhlman.org/python_book_01.pdf

5.2.29 Παράδειγμα 29 my_dict 1

Δημιουργία λεξικού μέσα στο πρόγραμμα



```
my_dict = {'name': 'Jack', 'age': 26}

# Output: Jack
print(my_dict['name'])

# Output: 26
print(my_dict.get('age'))

# Trying to access keys which doesn't exist throws error
# my_dict.get('address')
# my_dict['address']
```


Εικόνα 51 Παράδειγμα 29 my_dict

Το αποτέλεσμα είναι

Jack

26

5.2.30 Παράδειγμα 30 print my_dict 2



```
File Edit Format Run Options Windows Help
my_dict = {'name': 'Jack', 'age': 26}

# update value
my_dict['age'] = 27

#Output: {'age': 27, 'name': 'Jack'}
print(my_dict)

# add item
my_dict['address'] = 'Downtown'

# Output: {'address': 'Downtown', 'age': 27, 'name': 'Jack'}
print(my_dict)|
```

Εικόνα 52 Παράδειγμα 30 print my_dict 2

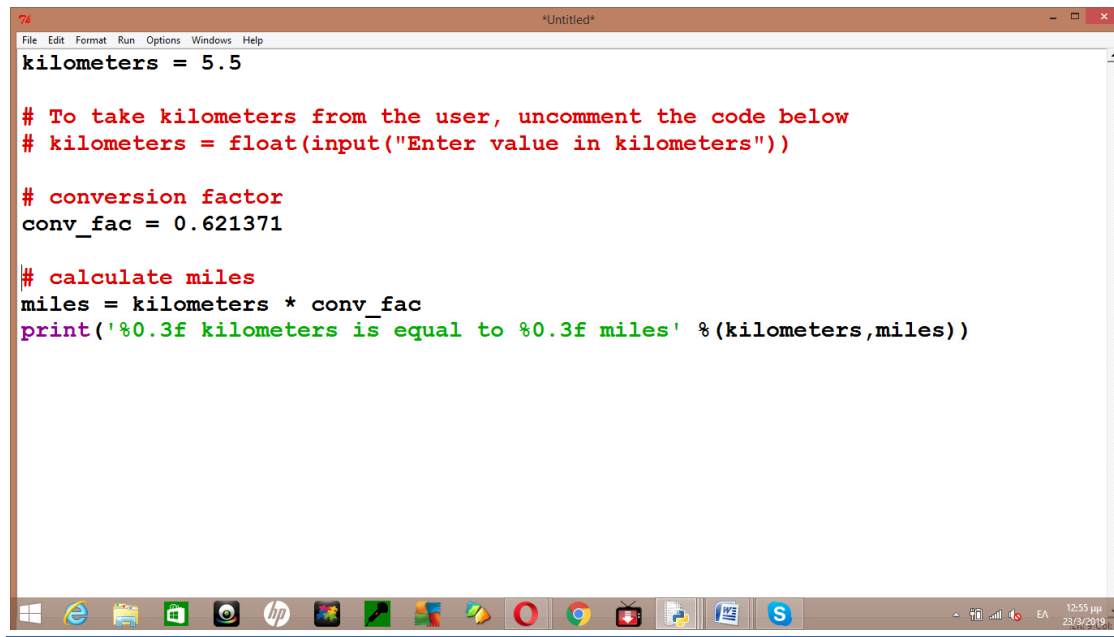
Το αποτέλεσμα είναι

{'age': 27, 'name': 'Jack'}

{'age': 27, 'address': 'Downtown', 'name': 'Jack'}

5.2.31 Παράδειγμα 31 conversion factor

Στο παράδειγμα που ακολουθεί γίνεται μετατροπή χιλιομέτρων σε μίλια.



```
File Edit Format Run Options Windows Help
kilometers = 5.5

# To take kilometers from the user, uncomment the code below
# kilometers = float(input("Enter value in kilometers"))

# conversion factor
conv_fac = 0.621371

# calculate miles
miles = kilometers * conv_fac
print('%0.3f kilometers is equal to %0.3f miles' % (kilometers,miles))
```

Εικόνα 53 Παράδειγμα 31 conversion factor

Το αποτέλεσμα είναι

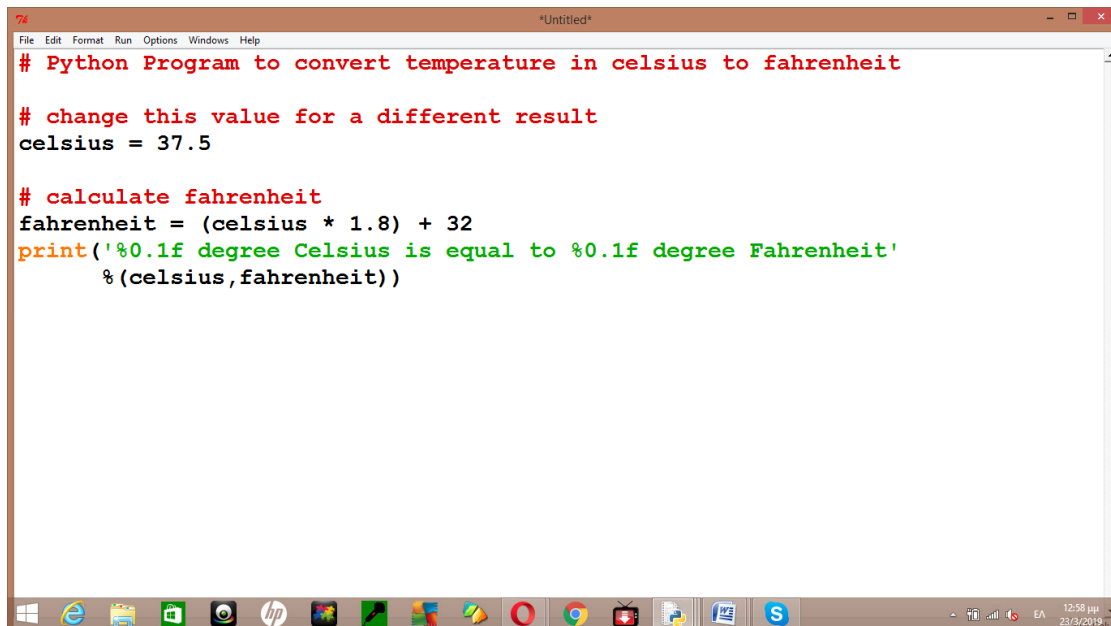
5.500 kilometers is equal to 3.418 miles

70

5.2.32 Παράδειγμα 32 convert temperature in Celsius to Fahrenheit

Το παράδειγμα αναφέρεται σε μετατροπή της θερμοκρασίας από βαθμούς κελσίου σε φαρναίτ.

⁷⁰ <https://www.openbook.gr/odigos-python-mesw-paradeigmatwn/>

A screenshot of a Python IDE window titled 'Untitled*'. The window contains the following Python code:

```
# Python Program to convert temperature in celsius to fahrenheit

# change this value for a different result
celsius = 37.5

# calculate fahrenheit
fahrenheit = (celsius * 1.8) + 32
print('%0.1f degree Celsius is equal to %0.1f degree Fahrenheit'
      %(celsius, fahrenheit))
```

The IDE window has a menu bar with 'File', 'Edit', 'Format', 'Run', 'Options', 'Windows', and 'Help'. The Windows taskbar is visible at the bottom, showing various application icons and the system tray with the date '23/3/2019' and time '12:58 μμ'.

Εικόνα 54 Παράδειγμα 32 convert temperature in Celsius to Fahrenheit

Το αποτέλεσμα είναι

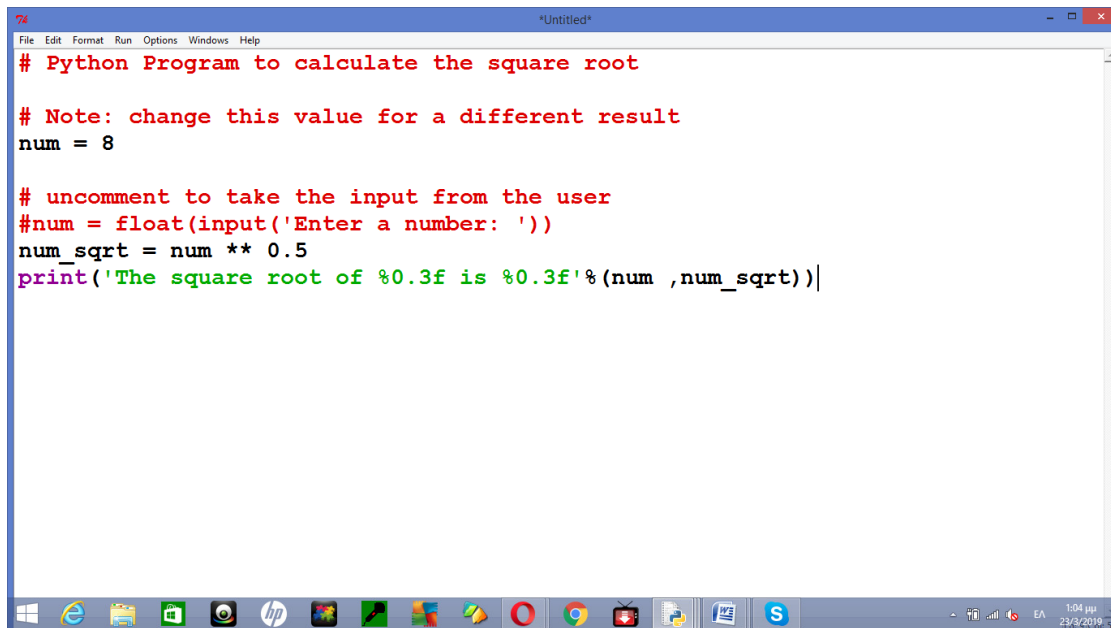
37.5 degree Celsius is equal to 99.5 degree Fahrenheit

71

5.2.33 Παράδειγμα 33 program to calculate the square root

Πρόγραμμα για υπολογισμό τετραγωνικής ρίζας.

⁷¹ <https://www.openbook.gr/odigos-python-mesw-paradeigmatwn/>



```
File Edit Format Run Options Windows Help
# Python Program to calculate the square root

# Note: change this value for a different result
num = 8

# uncomment to take the input from the user
#num = float(input('Enter a number: '))
num_sqrt = num ** 0.5
print('The square root of %0.3f is %0.3f'%(num ,num_sqrt))
```

Εικόνα 55 Παράδειγμα 33 program to square root

Το αποτέλεσμα είναι

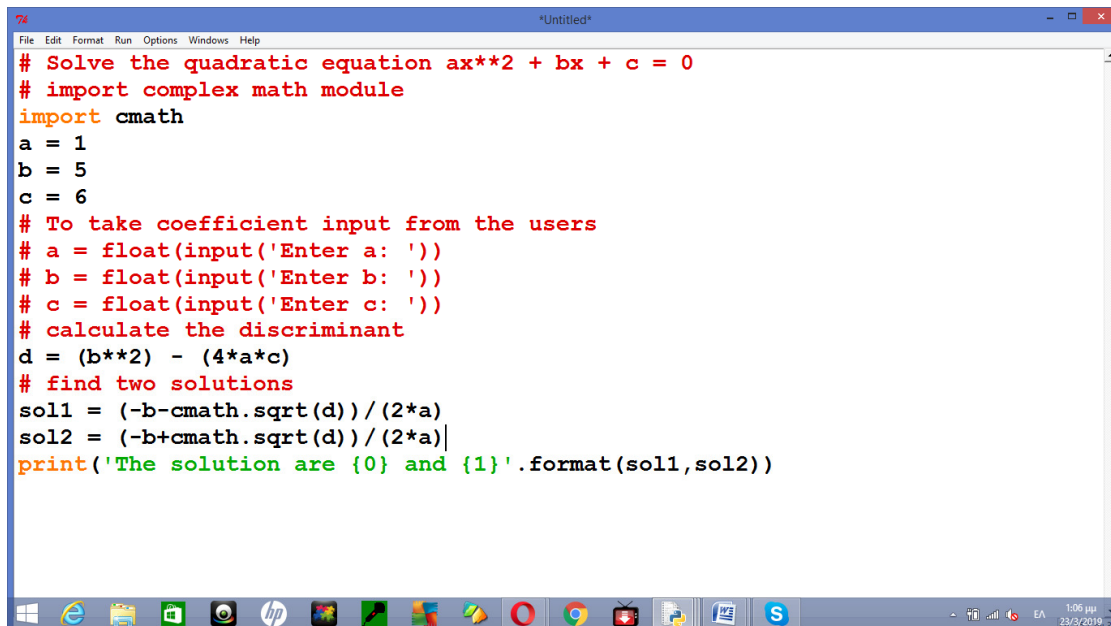
The square root of 8.000 is 2.828

72

5.2.34 Παράδειγμα 34 solve the quadratic equation $ax^2+bx+c=0$

Επίλυση εξίσωσης δευτέρου βαθμού.

⁷² <https://www.openbook.gr/odigos-python-mesw-paradeigmatwn/>



```
File Edit Format Run Options Windows Help
# Solve the quadratic equation ax**2 + bx + c = 0
# import complex math module
import cmath
a = 1
b = 5
c = 6
# To take coefficient input from the users
# a = float(input('Enter a: '))
# b = float(input('Enter b: '))
# c = float(input('Enter c: '))
# calculate the discriminant
d = (b**2) - (4*a*c)
# find two solutions
sol1 = (-b-cmath.sqrt(d))/(2*a)
sol2 = (-b+cmath.sqrt(d))/(2*a)
print('The solution are {0} and {1}'.format(sol1,sol2))
```

Εικόνα 56 ΠΑΡΑΔΕΙΓΜΑ 34 solve the quadratic equation $ax^2+bx+c=0$

Το αποτέλεσμα είναι

Enter a: 1

Enter b: 5

Enter c: 6

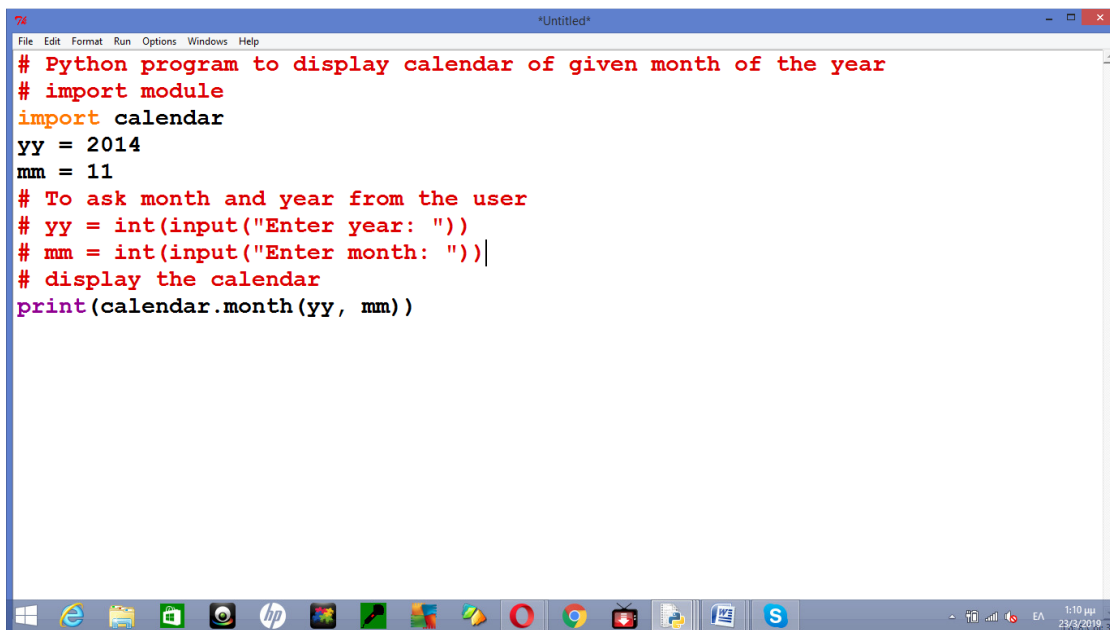
The solutions are $(-3+0j)$ and $(-2+0j)$

73

⁷³ <https://www.openbook.gr/odigos-python-mesw-paradeigmatwn/>

5.2.35 Παράδειγμα 35 program to display calendar of given month of the year

Παράδειγμα υπολογισμού ημερολογίου



```
# Python program to display calendar of given month of the year
# import module
import calendar
yy = 2014
mm = 11
# To ask month and year from the user
# yy = int(input("Enter year: "))
# mm = int(input("Enter month: "))
# display the calendar
print(calendar.month(yy, mm))
```

Εικόνα 57 Παράδειγμα 35 program to display calendar of given month of the year

74

Το αποτέλεσμα είναι

November 2014

Mo Tu We Th Fr Sa Su

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

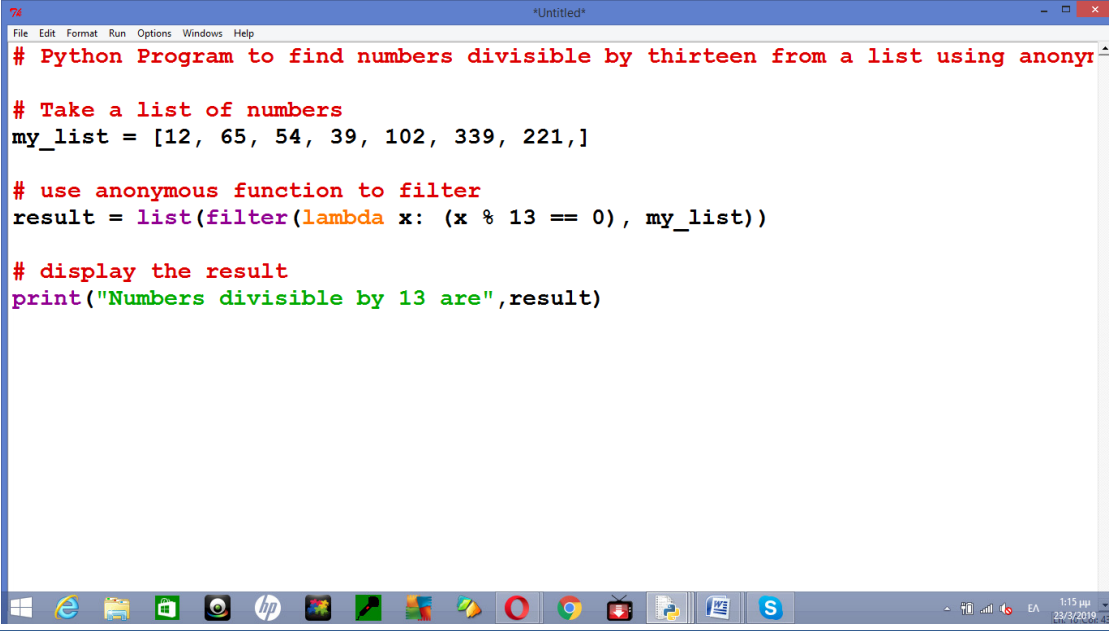
22 23 24 25 26 27 28

29 30

⁷⁴ https://www.davekuhlman.org/python_book_01.pdf

5.2.36 Παράδειγμα 36 program to find numbers divisible, list

Παράδειγμα λίστας.



```
# Python Program to find numbers divisible by thirteen from a list using anonymous function

# Take a list of numbers
my_list = [12, 65, 54, 39, 102, 339, 221,]

# use anonymous function to filter
result = list(filter(lambda x: (x % 13 == 0), my_list))

# display the result
print("Numbers divisible by 13 are",result)
```

Εικόνα 58 Παράδειγμα 36 program to find numbers divisible, list

Το αποτέλεσμα είναι :

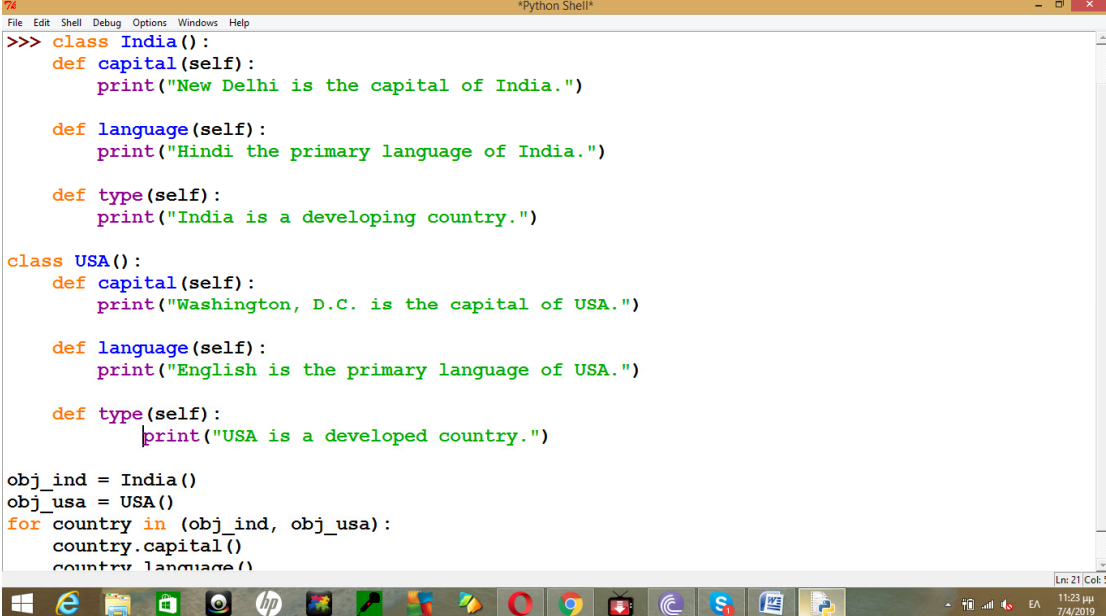
```
Numbers divisible by 13 are [65, 39, 221]
```

75

⁷⁵ <https://www.openbook.gr/odigos-python-mesw-paradeigmatwn/>

5.2.37 Παράδειγμα 37 class def

Παράδειγμα κλάσεων.



```
>>> class India():
    def capital(self):
        print("New Delhi is the capital of India.")

    def language(self):
        print("Hindi the primary language of India.")

    def type(self):
        print("India is a developing country.")

class USA():
    def capital(self):
        print("Washington, D.C. is the capital of USA.")

    def language(self):
        print("English is the primary language of USA.")

    def type(self):
        print("USA is a developed country.")

obj_ind = India()
obj_usa = USA()
for country in (obj_ind, obj_usa):
    country.capital()
    country.language()
```

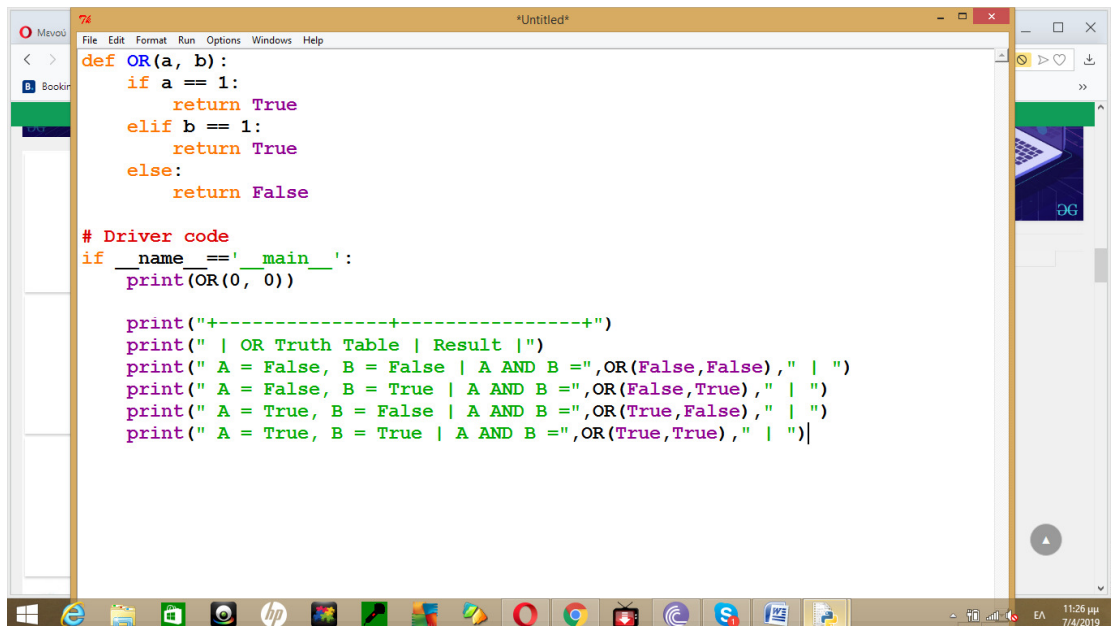
Εικόνα 59 Παράδειγμα 37 class def

```
New Delhi is the capital of India.
Hindi the primary language of India.
India is a developing country.
Washington, D.C. is the capital of USA.
English is the primary language of USA.
USA is a developed country.
```

76

⁷⁶<https://www.cse.unsw.edu.au/~en1811/python-docs/python-3.6.4-docs-pdf/tutorial.pdf>

5.2.38 Παράδειγμα 38 print, if, else, def, or



```
def OR(a, b):
    if a == 1:
        return True
    elif b == 1:
        return True
    else:
        return False

# Driver code
if __name__ == '__main__':
    print(OR(0, 0))

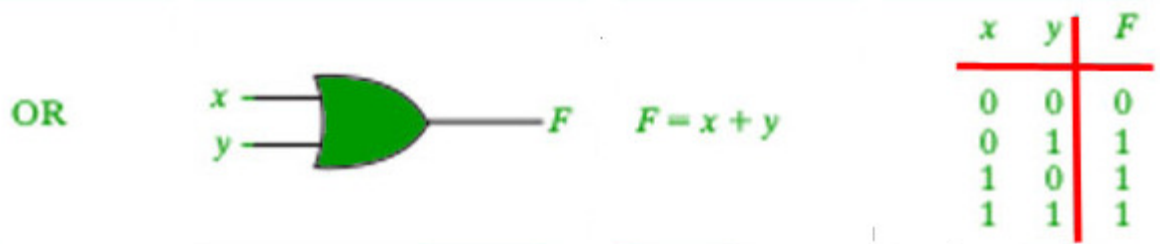
    print("+-----+-----+")
    print(" | OR Truth Table | Result |")
    print(" A = False, B = False | A AND B =",OR(False,False), " | ")
    print(" A = False, B = True | A AND B =",OR(False,True), " | ")
    print(" A = True, B = False | A AND B =",OR(True,False), " | ")
    print(" A = True, B = True | A AND B =",OR(True,True), " | ")
```

Εικόνα 60 Παράδειγμα 38 def, or, if, else

Το αποτέλεσμα είναι:

```
False
+-----+-----+
 | OR Truth Table |    Result |
A = False, B = False | A AND B = False |
A = False, B = True | A AND B = True |
A = True, B = False | A AND B = True |
A = True, B = True | A AND B = True |
```

The OR gate gives an output of 1 if either of the two inputs are 1, it gives 0 otherwise.



77

⁷⁷[https://www.brianheinold.net/python/A Practical Introduction to Python Programming Heinold.pdf](https://www.brianheinold.net/python/A_Practical_Introduction_to_Python_Programming_Heinold.pdf)

ΣΥΜΠΕΡΑΣΜΑΤΑ

Εφαρμογές της Python

- GUI που βασίζονται σε επιτραπέζιες εφαρμογές
- Εφαρμογές επεξεργασίας εικόνας και γραφικού σχεδιασμού
- Επιστημονικές και υπολογιστικές εφαρμογές
- Παιχνίδια
- Πλαίσια ιστού και εφαρμογές ιστού
- Επιχειρήσεις και επιχειρηματικές εφαρμογές
- Λειτουργικά συστήματα
- Γλωσσική ανάπτυξη
- Πρωτότυπα⁷⁸

Οφέλη της Python

Η ποικίλη εφαρμογή της γλώσσας Python είναι αποτέλεσμα του συνδυασμού χαρακτηριστικών που προσδίδουν στη γλώσσα αυτή ένα πλεονέκτημα έναντι των άλλων.

Μερικά από τα πλεονεκτήματα του προγραμματισμού στη Python περιλαμβάνουν:

1. Παρουσία ενοτήτων τρίτων

⁷⁸ <https://www.openbook.gr/eisagwgi-ston-antikeimenostrafi-programmatismo-me-python/>

Ο Python Package Index (PyPI) περιέχει πολλές μονάδες τρίτων που κάνουν την Python ικανή να αλληλεπιδρά με τις περισσότερες άλλες γλώσσες και πλατφόρμες.

2. Εκτεταμένες Βιβλιοθήκες Υποστήριξης

Η Python παρέχει ένα μεγάλη βιβλιοθήκη που περιλαμβάνει τομείς όπως πρωτόκολλα Διαδικτύου, λειτουργίες string, εργαλεία υπηρεσιών ιστού και διεπαφές λειτουργικού συστήματος. Πολλές εργασίες προγραμματισμού υψηλής χρήσης έχουν ήδη γραφτεί στη συνηθισμένη βιβλιοθήκη, η οποία μειώνει το μήκος του κώδικα που πρέπει να γραφτεί σημαντικά.

3. Open Source και Κοινοτική Ανάπτυξη

Η γλώσσα Python αναπτύσσεται με άδεια ανοικτού κώδικα εγκεκριμένη από OSI, η οποία την καθιστά ελεύθερη για χρήση και διανομή, συμπεριλαμβανομένων και για εμπορικούς σκοπούς.

Επιπλέον, η ανάπτυξή της καθοδηγείται από την κοινότητα που συνεργάζεται για τον κώδικα της μέσω φιλοξενίας συνεδρίων και λιστών αλληλογραφίας και προβλέπει τις πολυάριθμες ενότητες της.⁷⁹

4. Εκμάθηση ευκολίας και υποστήριξης διαθέσιμη

Η Python προσφέρει εξαιρετική αναγνωσιμότητα και εύχρηστη σύνταξη απλού στη μάθηση, η οποία βοηθά τους αρχάριους να χρησιμοποιούν αυτή τη γλώσσα προγραμματισμού. Οι οδηγίες, παρέχουν ένα σύνολο κανόνων για τη διευκόλυνση της μορφοποίησης του κώδικα. Επιπλέον, η ευρεία βάση χρηστών και ενεργών προγραμματιστών έχει οδηγήσει σε μια πλούσια τράπεζα πόρων διαδικτύου για να ενθαρρύνει την ανάπτυξη και τη συνεχή υιοθέτηση της γλώσσας.

5. Ευέλικτες δομές δεδομένων

⁷⁹ Μάνης Γ., (2015), «Εισαγωγή στον προγραμματισμό με αρωγό τη γλώσσα Python.» έκδοση: Σύνδεσμος ελληνικών ακαδημαϊκών βιβλιοθηκών. (pdf)

Η Python έχει ενσωματωμένες δομές δεδομένων και λίστας δεδομένων, οι οποίες μπορούν να χρησιμοποιηθούν για την κατασκευή δομών δεδομένων ταχείας εκτέλεσης. Επιπλέον, η Python παρέχει επίσης τη δυνατότητα δυναμικής πληκτρολόγησης δεδομένων υψηλού επιπέδου, η οποία μειώνει το μήκος του κωδικού υποστήριξης που απαιτείται.

6. Παραγωγικότητα και Ταχύτητα

Η Python έχει καθαρό αντικειμενοστραφή σχεδιασμό, παρέχει βελτιωμένες δυνατότητες ελέγχου της διαδικασίας και διαθέτει ισχυρές δυνατότητες ολοκλήρωσης και επεξεργασίας κειμένου καθώς και δικό της πλαίσιο δοκιμών μονάδων, τα οποία συμβάλλουν στην αύξηση της ταχύτητας και της παραγωγικότητάς της. Η Python θεωρείται βιώσιμη επιλογή για την κατασκευή σύνθετων εφαρμογών δικτύου πολλαπλών πρωτοκόλλων.

Όπως φαίνεται από τα παραπάνω σημεία, η Python προσφέρει πολλά πλεονεκτήματα για την ανάπτυξη λογισμικού. Καθώς η αναβάθμιση της γλώσσας συνεχίζεται, θα μπορούσε να αναπτυχθεί και η νομιμοποιημένη βάση της.

7. Δέσμες

Η Python θεωρείται εύκολο να μάθει και να τρέχει σχεδόν οπουδήποτε. Είναι χρήσιμο για πολλές εφαρμογές, όπως εκπαίδευση, ανάλυση δεδομένων και ανάπτυξη ιστού. Μερικές από τις μεγαλύτερες εταιρείες στον κόσμο βασίζονται σε μεγάλο βαθμό στην Python, συμπεριλαμβανομένων των Instagram και Google.

Πρόκειται για μια δυναμική, αντικειμενοστραφή γλώσσα προγραμματισμού (OO), παρόμοια με τις γλώσσες που βασίζονται στη Microsoft .NET ή Java, ως υπόστρωμα γενικού σκοπού για διάφορα είδη ανάπτυξης λογισμικού. Παρέχει ισχυρή υποστήριξη για την ενσωμάτωση με πολλές τεχνολογίες και υψηλότερη παραγωγικότητα προγραμματισμού σε ολόκληρο τον κύκλο ζωής της ανάπτυξης. Είναι ιδιαίτερα κατάλληλο για μεγάλα και σύνθετα έργα με μεταβαλλόμενες απαιτήσεις.⁸⁰

⁸⁰ <https://www.openbook.gr/eisagwgi-ston-antikeimenostrafi-programmatismo-me-python/>

Η Python είναι επίσης μια από τις γρηγορότερα αναπτυσσόμενες γλώσσες προγραμματισμού ανοιχτού κώδικα και χρησιμοποιείται σε κρίσιμες εφαρμογές για το μεγαλύτερο χρηματιστήριο στον κόσμο. Αποτελεί επίσης τη βάση για διάφορους ιστότοπους δημοσίευσης υψηλού επιπέδου, λειτουργεί σε αρκετά εκατομμύρια κινητά τηλέφωνα και χρησιμοποιείται σε βιομηχανίες όπως ο έλεγχος της εναέριας κυκλοφορίας, η κινηματογραφική ταινία μεγάλου μήκους και η ναυπηγική βιομηχανία.

8. Ευκολία στη χρήση και ανάγνωση

Οι περισσότεροι προγραμματιστές της Python θα συμφωνούσαν ότι το μεγαλύτερο πλεονέκτημα της Python είναι ότι είναι εύκολο να το πάρει. Η ευκολία στη χρήση και η εύκολη αναγνωσιμότητα είναι κάτι περισσότερο από απλή ευκολία. Η εύκολη χρησιμότητα μας βοηθά να σκεφτούμε με μεγαλύτερη σαφήνεια όταν γράφετε προγράμματα και για άλλους που πρέπει να ενισχύσουν ή να διατηρήσουν το πρόγραμμα.

Οι ειδικοί και οι αρχάριοι μπορούν εύκολα να κατανοήσουν τον κώδικα και μπορεί να γίνει κάποιος γρήγορος, παραγωγικός με αυτή τη γλώσσα. Δεδομένου ότι ο πηγαίος κώδικας του μοιάζει με τον ψευδοκώδικα, είναι επίσης εύκολο να μάθει. Μόλις αρχίσει κάποιος να μαθαίνει, μπορεί να ξεκινήσει την κωδικοποίηση αποτελεσματικά σχεδόν αμέσως.

Συνολικά, χρειάζεται λιγότερη προσπάθεια για να γράψει κανείς ένα πρόγραμμα στην Python από ό, τι θα χρησιμοποιούσε άλλες γλώσσες όπως η Java ή C ++. Αυτό είναι επίσης αρκετά δημοφιλές μεταξύ των πανεπιστημίων, με αποτέλεσμα μια μεγάλη δεξαμενή ταλέντων. Θεωρείται ένας πολύ παραγωγικός τρόπος γραφής κώδικα, και μερικά από αυτά προέρχονται από την αναγνωσιμότητα και την απλή σύνταξη. Ορισμένα προέρχονται από τις καλά σχεδιασμένες και πλούσιες ενσωματωμένες δυνατότητές του και την τυπική βιβλιοθήκη, καθώς και από τη διάθεση διαφόρων δομοστοιχείων και βιβλιοθηκών ανοιχτού κώδικα από τρίτους.⁸¹

⁸¹ Αγγελιδάκης Ν., «Εισαγωγή στον προγραμματισμό με την Python print Hello World», εκπαιδευτικός πληροφορικής (pdf)

Δεδομένου ότι είναι εύκολο να γίνει κατανοητό, είναι επίσης εύκολο να διατηρηθεί. Η γλώσσα είναι επίσης δυναμικά ευέλικτη και δακτυλογραφημένη, με κώδικα που δεν είναι τόσο λεπτομερής όσο άλλες γλώσσες. Αλλά αυτή η δυναμική δακτυλογράφηση θα μπορούσε επίσης να αποτελέσει μειονέκτημα, το οποίο θα συζητήσουμε αργότερα.

9. Απλή και γρήγορη

Η κοινότητα Python προσφέρει γρήγορη και αποτελεσματική υποστήριξη στους χρήστες και εκατοντάδες χιλιάδες προγραμματιστές εργάζονται σκληρά για να βρουν και να διορθώσουν σφάλματα και να αναπτύξουν νέες ενημερώσεις και βελτιώσεις στη γλώσσα. Προσφέρει επίσης γρήγορη ανατροφοδότηση με πολλούς τρόπους. Για ένα, οι προγραμματιστές μπορούν να παρακάμψουν διάφορες εργασίες που θα έπρεπε να γίνουν σε άλλες γλώσσες. Αυτό μειώνει τον χρόνο και το κόστος κάθε προγράμματος και τη συντήρηση που απαιτείται για το πρόγραμμα. Η Python επιτρέπει επίσης γρήγορη προσαρμογή του κώδικα. Η γλώσσα μπορεί να οριστεί ως έτοιμη για εκτέλεση, απαιτώντας απλά έναν απλό κώδικα για εκτέλεση. Η αναπαραγωγή και ο έλεγχος του κώδικα γίνεται πολύ πιο απλή με τη γλώσσα, η οποία προσφέρει επίσης ένα στυλ ανάπτυξης από κάτω προς τα πάνω για την εύκολη κατασκευή της εφαρμογής σας ελέγχοντας βασικές λειτουργίες στον διερμηνέα προτού αρχίσει ο χρήστης να γράφει κώδικα ανώτατου επιπέδου.⁸²

Ο διερμηνέας είναι εύκολα επεκτάσιμος, επιτρέποντάς σας να ενσωματώσετε τον κώδικα C με μια απλή μονάδα επέκτασης. Η Python ενθαρρύνει την επαναχρησιμοποίηση του προγράμματος με πακέτα και ενότητες. Ορισμένες μονάδες είναι ήδη διαθέσιμες με την τυπική βιβλιοθήκη, απαραίτητη για τη διανομή Python. Μπορεί ο χρήστης να μοιραστεί τη λειτουργικότητα μεταξύ διαφορετικών προγραμμάτων, διασπώντας τα σε διάφορες ενότητες.

⁸² learning to program with Python Richard L. Halterman, (2011), (pdf)

Η γλώσσα μπορεί να τρέξει σε πολλά συστήματα αλλά διατηρεί την παρόμοια διεπαφή και ο σχεδιασμός της δεν αλλάζει πολύ με το κάθε λειτουργικό σύστημα, αφού είναι γραμμένο σε φορητό ANSI C. Αυτό σημαίνει ότι μπορεί εύκολα να γράψει κάποιος Python σε Mac, αφού δοκιμάσει το ένα σύστημα Linux και μεταφόρτωση σε έναν υπολογιστή με Windows.⁸³

10. Ευχρηστία με IoT

Το Διαδίκτυο των πραγμάτων ή του Διαδικτύου έχει ανοίξει τεράστιες ευκαιρίες και η Python μπορεί να διαδραματίσει βασικό ρόλο στη χρήση αυτών των ευκαιριών. Η γλώσσα γίνεται μια δημοφιλή επιλογή για το IoT, με νέες πλατφόρμες όπως το Raspberry Pi που βασίζεται σε αυτό. Η τεκμηρίωση για το Raspberry P δηλώνει ότι η γλώσσα είναι εύκολη στη χρήση και τη δύναμη.

11. Ασύγχρονη κωδικοποίηση

Η Python έχει αποδειχθεί αρκετά αποτελεσματική για τη σύνταξη ασύγχρονου κώδικα, ο οποίος χρησιμοποιεί έναν ενιαίο βρόχο συμβάντων για να κάνει εργασία σε μικρές μονάδες αντί να γράφει χρήσεις. Αυτό συμβαίνει επειδή είναι ευκολότερο να γράφει καθώς και να διατηρεί χωρίς να δημιουργεί σύγχυση σε έρευνες ή αδιέξοδα ή άλλα ζητήματα. Αυτές οι γεννήτριες είναι πολύ χρήσιμες για τη διεμπλοκή λειτουργίας πολλών βρόχων επεξεργασίας.

12. Μία λιγότερο περιορισμένη προσέγγιση προγραμματισμού

Σε σύγκριση με την Java, η Python χρησιμοποιεί πολύ λιγότερο περιορισμένη προσέγγιση προγραμματισμού πολλαπλών παραδειγμάτων. Για παράδειγμα, δεν χρειάζεται να δημιουργήσει ξεχωριστή κλάση OO για εκτύπωση "Hello World" στην Python, αλλά πρέπει να το κάνει στην Java. Το Python είναι πολυπαραδειγματικό και υποστηρίζει λειτουργικά, διαδικαστικά και αντικειμενοστραφή στυλ προγραμματισμού.

⁸³ «Εισαγωγή στη γλώσσα προγραμματισμού Python βασισμένο στην παρουσίαση της Python του Michael DiRamio», 2005, 12-10-2018 TEI ΚΡΗΤΗΣ(pdf)

Στην Python, τίποτα και όλα μπορεί να είναι ένα αντικείμενο. Μπορεί να γράψει εφαρμογές στη γλώσσα χρησιμοποιώντας πολλά παραδείγματα προγραμματισμού και μπορεί ακόμα να γράψει καθαρό, σαφή και κατανοητό κώδικα OO.⁸⁴

13. Ενσωμάτωση Εφαρμογών Επιχειρήσεων

Η Python είναι μια εξαιρετική επιλογή για μια γλώσσα προγραμματισμού που περιλαμβάνει Enterprise Integration Application (EAI). Κάνει την ανάπτυξη υπηρεσιών ανάπτυξης ιστοσελίδων πιο εύκολη, επικαλούμενη CORBA ή COM στοιχεία και απευθείας κλήση από και προς Java, C + + ή C κώδικα. Παρέχει σημαντικές λειτουργίες ελέγχου διαδικασιών και υλοποιεί κοινές μορφές και πρωτόκολλα δεδομένων διαδικτύου, επεξεργάζεται γλώσσες σήμανσης όπως XL, εκτελείται από τον ίδιο κώδικα byte στα σύγχρονα λειτουργικά συστήματα και μπορεί να ενσωματωθεί ως γλώσσα δέσμης ενεργειών.

14. Η χρήση του στην ανάπτυξη ιστού

Η Python μπορεί να χρησιμοποιηθεί και χρησιμοποιείται εκτενώς για την ανάπτυξη ιστοσελίδων, για σκοπούς που κυμαίνονται από την ανάπτυξη εφαρμογών web υψηλής τεχνολογίας έως απλή scripting CGI σε πλαίσια μεγάλης κλίμακας όπως το TurboGears και το Django . Άλλα παραδείγματα χρήσης της Python στην ανάπτυξη ιστού περιλαμβάνουν το πλαίσιο εφαρμογής Quixote web, το σύστημα διαχείρισης περιεχομένου Plone και το διακομιστή εφαρμογής Zope.

Μπορεί εύκολα να δημιουργήσει κάποιος τη δική του λύση με βάση τις εύχρηστες και εκτεταμένες τυποποιημένες βιβλιοθήκες της Python. Η Python παρέχει διασυνδέσεις για τις περισσότερες βάσεις δεδομένων, λειτουργεί

⁸⁴ learning to program with Python Richard L. Halterman, (2011), (pdf)

καλά με άλλες τεχνολογίες ανάπτυξης ιστού και διαθέτει ισχυρές εγκαταστάσεις επεξεργασίας κειμένου και κειμένου.⁸⁵

15. Η χρήση του σε επιστημονικές και αριθμητικές εφαρμογές

Μπορεί να χρησιμοποιήσει τη βιβλιοθήκη απεικόνισης της Python, καθώς και τα εργαλεία εργαλείων απεικόνισης MayaVi και VTK 3D, καθώς και άλλα εργαλεία όπως το ScientificPython και το Numeric Python για την ανάπτυξη αριθμητικών και επιστημονικών εφαρμογών. Πολλές από αυτές τις εφαρμογές μπορούν επίσης να υποστηριχθούν από την Ενσωμάτωση της Διανομής Python.⁸⁶

16. Σάρωση εφαρμογών και δοκιμές λογισμικού

Η ισχυρή ενσωμάτωση της Python με το Java και το C και το C ++ καθιστά πολύ χρήσιμη τη χρήση scripting εφαρμογών. Σχεδιάστηκε από την αρχή για να είναι ενσωματωμένο και μπορεί να είναι μια εξαιρετική επιλογή για μια γλώσσα δέσμης ενεργειών για την προσαρμογή ή την επέκταση μεγαλύτερων εφαρμογών. Η Python μπορεί επίσης να χρησιμοποιηθεί για εκτεταμένες δοκιμές λογισμικού, χάρη στις ισχυρές δυνατότητες επεξεργασίας κειμένου και ενοποίησης. Στην πραγματικότητα, η Python έρχεται ακόμη και με το πολύ δικό της πλαίσιο δοκιμών μονάδων. Η Python μπορεί να χρησιμοποιηθεί και για την ανάπτυξη εφαρμογών GUI υψηλής ποιότητας. Μπορεί να χρησιμοποιήσει ανοιχτές τεχνολογίες για να αναπτύξει την εφαρμογή στα περισσότερα λειτουργικά συστήματα. Υποστήριξη για άλλα πλαίσια GUI όπως Motif, X11, Delphi, Carbon και MFC είναι επίσης διαθέσιμα.

17. Η χρήση του Python σε πρωτότυπα και πλεονεκτήματα ανοικτού κώδικα

Το πρωτότυπο σε Python είναι αρκετά εύκολο και γρήγορο, με αποτέλεσμα την ανάπτυξη του τελικού συστήματος σε αρκετές περιπτώσεις. Δεδομένου ότι

⁸⁵ Αγγελιδάκης Ν., «Εισαγωγή στον προγραμματισμό με την Python print Hello World», εκπαιδευτικός πληροφορικής (pdf)

⁸⁶ <https://www.openbook.gr/eisagwgi-ston-antikeimenostrafi-programmatismo-me-python/>

η Python είναι αρκετά ευέλικτη, μπορείτε εύκολα να αναδιαμορφώσετε κώδικα για γρήγορη ανάπτυξη από το πρώτο πρωτότυπο έως το τελικό προϊόν. Η φύση της ανοιχτής πηγής της Python είναι επίσης ένα τεράστιο πλεονέκτημα. Είναι καλά σχεδιασμένο, κλιμακωτό, φορητό, εύρωστο και γρήγορο λόγω της φύσης του. Η σύνταξή του είναι εύκολη και έχει ξεκάθαρα και καλά αναπτυγμένα προηγμένα γλωσσικά χαρακτηριστικά. Με πολλούς τρόπους, η Python ξεπερνά τα χαρακτηριστικά και τις δυνατότητες άλλων συγκρίσιμων λύσεων που διατίθενται στο εμπόριο.

Η άδεια ανοιχτού κώδικα της Python επιτρέπει επίσης την απεριόριστη τροποποίηση, αναδιανομή και χρήση της γλώσσας και των εφαρμογών που βασίζονται σε αυτήν. Η πλήρης πηγή είναι διαθέσιμη και δεν υπάρχει κόστος αδειοδότησης, το οποίο αποτελεί τεράστιο εξοικονόμηση κόστους. Η υποστήριξη διατίθεται δωρεάν μέσω πόρων του διαδικτύου.

18. Σάρωση διακομιστή⁸⁷

Η Python θεωρείται ισχυρή γλώσσα προγραμματισμού για διακομιστές. Ο κώδικας του μοιάζει με ψευδοκώδικα όπως και άλλες γλώσσες δέσμης ενεργειών και δεν έχει πλούσια ή περίπλοκη σύνταξη. Έχει κατασκευαστεί έτσι ώστε να εστιάζετε λιγότερα στην εντολή που θέλει να χρησιμοποιήσει και να επικεντρωθεί στους επιχειρηματικούς κανόνες της εφαρμογής.

19. Φορητότητα και διαδραστικότητα

Ένα άλλο τεράστιο πλεονέκτημα της Python είναι η φορητότητα και η διαδραστικότητα, καθιστώντας την πολύ πιο εύκολη στην εκμάθηση. Παρέχει δυναμική σημασιολογία και δυνατότητες γρήγορης δημιουργίας πρωτοτύπων. Συχνά θεωρείται ως γλώσσα κόλλας, συνδέοντας τα διαφορετικά υπάρχοντα εξαρτήματα. Είναι εξαιρετικά ενσωματωμένο σε εφαρμογές, ακόμη και σε εκείνες που χρησιμοποιούν άλλες γλώσσες προγραμματισμού. Αυτό μας δίνει

⁸⁷ «Εισαγωγή στη γλώσσα προγραμματισμού Python βασισμένο στην παρουσίαση της Python του Michael DiRamio», 2005, 12-10-2018 ΤΕΙ ΚΡΗΤΗΣ(pdf)

τη δυνατότητα να διορθώσουμε νέες ενότητες στην Python και να επεκτείνουμε το βασικό λεξιλόγιο της.⁸⁸

⁸⁸ «Εισαγωγή στη γλώσσα προγραμματισμού Python βασισμένο στην παρουσίαση της Python του Michael DiRamio», 2005, 12-10-2018 ΤΕΙ ΚΡΗΤΗΣ(pdf)

ΠΕΡΙΕΧΟΜΕΝΑ ΠΙΝΑΚΩΝ

Πίνακας 1 Boolean Operations.....	- 38 -
Πίνακας 2 Λειτουργίες Σύγκρισης.....	- 38 -
Πίνακας 3 Αριθμητικοί Τύποι.....	- 42 -
Πίνακας 4 Λειτουργίες μαθηματικών	- 44 -
Πίνακας 5 Λειτουργίες με δυαδικά ψηφία	- 45 -
Πίνακας 6 Λειτουργίες συγκρίσεων και συνθήκη αληθή ή ψευδή	- 49 -

ΠΕΡΙΕΧΟΜΕΝΑ ΕΙΚΟΝΩΝ

Εικόνα 1 Open Source	- 17 -
Εικόνα 2 το χρονοδιάγραμμα της python	- 21 -
Εικόνα 3 Τομείς που μπορεί να χρησιμοποιηθεί η Python ython.....	- 25 -
Εικόνα 4 Εγκατάσταση στα Windows.....	- 27 -
Εικόνα 5 Install Python.....	- 28 -
Εικόνα 6 Πρόοδος install setup Python.....	- 29 -
Εικόνα 7 Idle	- 31 -
Εικόνα 8 Python Shell	- 32 -
Εικόνα 9 Άνοιγμα Αρχείου.....	- 33 -
Εικόνα 10 Εγγραφή Κώδικα.....	- 33 -
Εικόνα 11 Αποθήκευση.....	- 34 -
Εικόνα 12 Αποθήκευση αρχείου python.....	- 35 -
Εικόνα 13 Run Module.....	- 35 -
Εικόνα 14 Παράθυρο Idle	- 36 -
Εικόνα 15 Πρόγραμμα Python έκδοση 2.7.4.....	- 58 -
Εικόνα 16 Πρώτη Καρτέλα σε πρόγραμμα Python με τα Αρχεία	- 59 -
Εικόνα 17 Δεύτερη καρτέλα σε Πρόγραμμα Python με την επεξεργασία	- 59 -
Εικόνα 18 Τρίτη Καρτέλα σε Πρόγραμμα Python Shell	- 60 -
Εικόνα 19 Τέταρτη καρτέλα σε πρόγραμμα Python Debug	- 61 -
Εικόνα 20 Πέμπτη καρτέλα σε πρόγραμμα Python Options.....	- 61 -
Εικόνα 21 Έκτη καρτέλα σε Πρόγραμμα Python Windows	- 62 -
Εικόνα 22 Έβδομη καρτέλα σε Πρόγραμμα Python Help.....	- 63 -
Εικόνα 23 Παράδειγμα 1 print.....	- 64 -
Εικόνα 24 Παράδειγμα 2 print,int, str,error, σχόλια	- 65 -
Εικόνα 25 Παράδειγμα 3 print x.....	- 65 -
Εικόνα 26 Παράδειγμα 4 print, syntax error, μεταβλήτη	- 66 -
Εικόνα 27 Παράδειγμα 5 sum, print sum	- 67 -
Εικόνα 28 Παράδειγμα 6 συναρτήσεων min, max, sum	- 68 -
Εικόνα 29 Παράδειγμα 7, μεταβλητή var.....	- 69 -
Εικόνα 30 Παράδειγμα 8, μαθηματικές πράξεις.....	- 69 -
Εικόνα 31 Παράδειγμα 9 μαθηματικές πράξεις και λάθη.....	- 70 -
Εικόνα 32 Παράδειγμα 10 λάθη σύνταξης.....	- 71 -
Εικόνα 33 Παράδειγμα 11 συνάρτηση if.....	- 72 -
Εικόνα 34 Παράδειγμα 12 συνάρτηση if -true	- 73 -
Εικόνα 35 Παράδειγμα 13 συνάρτηση range	- 73 -
Εικόνα 36 Παράδειγμα 14 συνάρτηση for , in range	- 74 -
Εικόνα 37 Παράδειγμα 15 συνθήκη αληθής ή ψευδής	- 75 -
Εικόνα 38 Παράδειγμα 16 κείμενο χωρίς φωνήεν	- 76 -
Εικόνα 39 Παράδειγμα 17 θέτωντας το dict.....	- 76 -
Εικόνα 40 Παράδειγμα 18 συνάρτηση and.....	- 77 -
Εικόνα 41 Παράδειγμα 19 συνάρτηση if-else.....	- 78 -

Εικόνα 42 Παράδειγμα 20 factors of a number	- 78 -
Εικόνα 43 Παράδειγμα 21 PROGRAM TO ADD TWO MATRICES	- 80 -
Εικόνα 44 Παράδειγμα 22 print my_string	- 81 -
Εικόνα 45 Παράδειγμα 23 print string1, string2 using +, *	- 82 -
Εικόνα 46 Παράδειγμα 24 program to swap two variables	- 83 -
Εικόνα 47 Παράδειγμα 25 find the factorial of a number	- 84 -
Εικόνα 48 Παράδειγμα 26 print my_tuple	- 85 -
Εικόνα 49 Παράδειγμα 27 class string, class tuple	- 87 -
Εικόνα 50 Παράδειγμα 48 use of break statement inside loop.....	- 88 -
Εικόνα 51 Παράδειγμα 29 my_dict.....	- 89 -
Εικόνα 52 Παράδειγμα 30 print my_dict 2	- 90 -
Εικόνα 53 Παράδειγμα 31 conversion factor	- 91 -
Εικόνα 54 Παράδειγμα 32 convert temperature in Celsius to Fahrenheit.....	- 92 -
Εικόνα 55 Παράδειγμα 33 program to square root.....	- 93 -
Εικόνα 56 ΠΑΡΑΔΕΙΓΜΑ 34 solve the quadratic equation $ax^{**2}+bx+c=0$	- 94 -
Εικόνα 57 Παράδειγμα 35 program to display calendar of given month of the year	- 95 -
-	
Εικόνα 58 Παράδειγμα 36 program to find numbers divisible, list.....	- 96 -
Εικόνα 59 Παράδειγμα 37 class def.....	- 97 -
Εικόνα 60 Παράδειγμα 38 def, or, if, else	- 98 -

ΒΙΒΛΙΟΓΡΑΦΙΑ

ΕΛΛΗΝΙΚΗ ΒΙΒΛΙΟΓΡΑΦΙΑ

- Αγγελιδάκης Ν., «Εισαγωγή στον προγραμματισμό με την Python print Hello World», εκπαιδευτικός πληροφορικής (pdf)
- Κουρκουμπετης, διαφάνειες Κεφάλαιο 6 γλώσσες προγραμματισμού (pdf)
- Λεβαντέας Δ., (2010), «Οδηγός Python μέσω παραδειγμάτων.», έκδοση: ομάδα TasPython. (pdf)
- Μαγκούτης Κ., Νικολάου Χ., (2015), «Εισαγωγή στον αντικειμενοστραφή προγραμματισμό με Python. μία προσέγγιση από την πλευρά των υπολογιστικών Συστημάτων.», έκδοση: Σύνδεσμος ελληνικών ακαδημαϊκών βιβλιοθηκών. (pdf)
- Μάνης Γ., (2015), «Εισαγωγή στον προγραμματισμό με αρωγό τη γλώσσα Python.» έκδοση: Σύνδεσμος ελληνικών ακαδημαϊκών βιβλιοθηκών. (pdf)
- Πανεπιστήμιο Πατρών University of Patras διαφάνειες (pdf)
- Σταματόπουλος Π., (2017), «Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών τμήμα πληροφορικής και τηλεπικοινωνιών σημειώσεις εισαγωγή στον προγραμματισμό.» (pdf)

- Χατζήνικολακάκης Γ., (2016), «Προγραμματισμός με Python στο Raspberry Pi.» Έκδοση σύλλογος εκπαιδευτικών πληροφορικής Χίου. (pdf)
- «Εισαγωγή στη γλώσσα προγραμματισμού Python βασισμένο στην παρουσίαση της Python του Michael DiRamio», 2005, 12-10-2018 ΤΕΙ ΚΡΗΤΗΣ(pdf)
- Κεφάλαιο 1 γλώσσες προγραμματισμού (pdf).
- Υπουργείο Πολιτισμού Παιδείας και Θρησκευμάτων Ινστιτούτο εκπαιδευτικής πολιτικής Αράπογλου Α., Κανίδης Ε., Βραχνός Ε., Μακρυγιάννης Π., Μπελεσιώτης Β., Δήμας Δ., «Αρχές προγραμματισμού υπολογιστών Β τάξη ΕΠΑΛ σημειώσεις μαθητή τεχνολογίας υπολογιστών», εκδόσεις: Διόφαντος

ΞΕΝΗ ΒΙΒΛΙΟΓΡΑΦΙΑ

- CS 16 Introduction to Python to (2018), Introduction to Algorithms and Data Structures. (pdf)
- (2009) Introduction to Python Heavily based on presentations by Matt Huenerfauth (Penn State), Guido van Rossum (google), Richard P. Muller (Caltech). (pdf)
- Think Python How to think like a computer scientist έκδοση 1.0.3 Σεπτέμβριος (2014) (pdf)
- learning to program with Python Richard L. Halterman, (2011), (pdf)
- .
- 1213 Python David mayer painting Cambridge fatmagul sebastopol Tokyo
- Artificial Intelligence with Python tutorials point Simply Easy learning www .tutorial point.com. (pdf)

ΙΣΤΟΣΕΛΙΔΕΣ

- <http://pages.cs.aueb.gr>
- <https://www.techopedia.com>
- <https://the-eye.eu>
- <http://pliroforikiatschool.blogspot.com>
- <http://thecorememory.com>
- <https://opensource.ellak.gr>
- <https://www.techrepublic.com>
- <https://startupper.gr>
- <http://www.trytoprogram.com>
- <https://arch.icte.uowm.gr>
- <https://eclass.upatras.gr>
- <https://slideplayer.gr>
- <https://repository.kallipos.gr>
- <http://www.bestprogramminglanguagefor.me>
- <https://www.google.gr>
- <https://www.pcsteps.gr>
- <https://www.openbook.gr>
- <http://www.ba.teiwest.gr>
- www.python.org
- <http://astronomi.erciyes.edu.tr/wp-content/uploads>
- <https://www.cse.unsw.edu.au/~en1811/>
- https://www.davekuhlman.org/python_book_01
- <https://www.brianheinold.net/python/>