

ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΤΜΗΜΑΤΟΣ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ ΜΕ ΤΗΝ ΡΥΤΗΟΝ: ΠΕΡΙΒΑΛΛΟΝΤΑ
ΚΑΙ ΕΡΓΑΛΕΙΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ, ΔΙΔΑΚΤΙΚΕΣ ΠΡΟΣΕΓΓΙΣΕΙΣ ΚΑΙ
ΜΕΛΕΤΕΣ

Διπλωματική Εργασία

της

Λιάκου Βάγιας

Θεσσαλονίκη, Ιούνιος 2020

ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ ΜΕ ΤΗΝ ΡΥΤΗΟΝ: ΠΕΡΙΒΑΛΛΟΝΤΑ
ΚΑΙ ΕΡΓΑΛΕΙΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ, ΔΙΔΑΚΤΙΚΕΣ ΠΡΟΣΕΓΓΙΣΕΙΣ ΚΑΙ
ΜΕΛΕΤΕΣ

Λιάκου Βάγια

Πτυχίο Μηχανικών Πληροφορικής, ΑΤΕΙ Θεσ/κης, 2014

Διπλωματική Εργασία

υποβαλλόμενη για τη μερική εκπλήρωση των απαιτήσεων του

ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΤΙΤΛΟΥ ΣΠΟΥΔΩΝ ΣΤΗΝ ΕΦΑΡΜΟΣΜΕΝΗ
ΠΛΗΡΟΦΟΡΙΚΗ

Επιβλέπων Καθηγητής
ΞΥΝΟΓΑΛΟΣ ΣΤΥΝΙΑΝΟΣ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την ηη/μμ/εεεε

Ξυνόγαλος Στυλιανός

Σατρατζέμη Μάγια

Σακελλαρίου Ηλίας

ΛΙΑΚΟΥ ΒΑΓΙΑ

Περίληψη

Η διδασκαλία του προγραμματισμού αποτελεί ένα δύσκολο έργο καθώς οι αρχάριοι προγραμματιστές αντιμετωπίζουν αρκετές δυσκολίες με τις βασικές αρχές του προγραμματισμού. Η επιλογή της γλώσσας προγραμματισμού που θα χρησιμοποιηθεί για την εισαγωγή των μαθητών στον προγραμματισμό αποτελεί σημαντικό παράγοντα. Αρκετοί ερευνητές θεωρούν ότι οι δυσκολίες αυτές οφείλονται και στα προγραμματιστικά εργαλεία και περιβάλλοντα που χρησιμοποιούν οι αρχάριοι προγραμματιστές για την ανάπτυξη του κώδικα τους.

Αναλύονται τα βασικά στοιχεία της γλώσσας προγραμματισμού Python και παρουσιάζονται όλα εκείνα τα χαρακτηριστικά που κάνουν την Python μια πολύ καλή επιλογή γλώσσας για την εισαγωγή των μαθητών στις βασικές αρχές του προγραμματισμού. Επίσης παρουσιάζονται και αναλύονται τα χαρακτηριστικά από τα προγραμματιστικά περιβάλλοντα προγραμματισμού που χρησιμοποιούνται για την ανάπτυξη κώδικα. Ακολουθεί μια συγκριτική ανάλυση των κοινών χαρακτηριστικών και των διαφορών που έχουν αυτά τα περιβάλλοντα.

Παρουσιάζονται τα οφέλη της Python στη διδασκαλία των βασικών εννοιών του προγραμματισμού μέσα από την εφαρμογή της σε εισαγωγικά μαθήματα που διδάσκονται σε πανεπιστήμια αλλά και σε σχολεία. Επιπρόσθετα, γίνεται μια ανασκόπηση των διδακτικών προσεγγίσεων που χρησιμοποιούνται για τη γρήγορη και εύκολη εκμάθηση της γλώσσας από τους αρχάριους προγραμματιστές.

Τέλος, παρουσιάζονται κριτήρια που βασίζονται σε αποφάσεις σχεδιασμού της γλώσσας που χρησιμοποιείται για τη διδασκαλία και την εκμάθηση από τους αρχάριους προγραμματιστές. Με βάση αυτά τα κριτήρια γίνεται μια συγκριτική μελέτη μεταξύ της γλώσσας Python και των γλωσσών προγραμματισμού που χρησιμοποιούνται για τη διδασκαλία των εισαγωγικών εννοιών του προγραμματισμού.

Λέξεις Κλειδιά: Εισαγωγή στον προγραμματισμό, γλώσσα προγραμματισμού Python, εκπαιδευτικά περιβάλλοντα προγραμματισμού, σύγκριση με άλλες γλώσσες προγραμματισμού, διδακτικές προσεγγίσεις

Abstract

Teaching programming is a difficult task due to the fact that novice programmers encounter a number of difficulties regarding the basic principles of programming. The choice of the programming language which is going to be used for the introduction of the students to programming is an important factor. A considerable number of researchers believe that these difficulties are also due to the programming tools and integrated development environments which are used by novice programmers for the development of their code.

The basic components of the Python programming language are analyzed and all the features that make Python a very good choice of language for the introduction of students to the basic principles of programming are presented. Furthermore, the features of the integrated development environments used for the development of code are presented and analyzed. What follows is a comparative analysis of the common features and the differences which are present in these environments.

The benefits of Python in the teaching of the fundamental concepts of programming are presented through its application to introductory courses which are taught in universities as well as in schools. Moreover, the teaching approaches which are implemented for the language to be acquired more quickly and easily by the novice programmers are reviewed.

Finally, criteria which are based on decisions regarding the designing of the language used for teaching and learning by the novice programmers are presented. Under these criteria, a comparative study is conducted between the Python language and the programming languages which are used for the teaching of introductory concepts of programming.

Keywords: Introduction to programming, Python programming language, educational programming environments, didactic approaches

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή του Τμήματος Εφαρμοσμένης Πληροφορικής του Πανεπιστημίου Μακεδονίας κ. Στυλιανό Ξυνόγαλο, για την εμπιστοσύνη, το ενδιαφέρον και την καθοδήγηση του καθ' όλη την διάρκεια της εκπόνησης της διπλωματικής μου εργασίας.

Επίσης θα ήθελα να ευχαριστήσω την οικογένεια μου για την ψυχολογική στήριξη που μου παρείχε όλο αυτό το διάστημα.

Περιεχόμενα

| | | |
|-------|---------------------------------------------------|----|
| 1 | Εισαγωγή | 5 |
| 1.1 | Πρόβλημα - Σημαντικότητα του θέματος | 5 |
| 1.2 | Σκοπός – Στόχοι | 5 |
| 1.3 | Συνεισφορά | 5 |
| 1.4 | Διάρθρωση της μελέτης | 6 |
| 2 | Η γλώσσα προγραμματισμού Python | 8 |
| 2.1 | Χαρακτηριστικά της Python | 9 |
| 2.1.1 | Απλή | 9 |
| 2.1.2 | Φορητή | 9 |
| 2.1.3 | Ελεύθερη και ανοιχτού κώδικα | 10 |
| 2.1.4 | Επεκτάσιμη και με δυνατότητα ενσωμάτωσης | 10 |
| 2.1.5 | Γλώσσα υψηλού επιπέδου και διερμηνευόμενη | 10 |
| 2.1.6 | Αντικειμενοστρεφής | 10 |
| 2.1.7 | Εκτεταμένες βιβλιοθήκες | 11 |
| 3 | Περιβάλλοντα Προγραμματισμού | 12 |
| 3.1 | Επεξεργαστές κειμένου-κώδικα (Code Editors) | 12 |
| 3.1.1 | Επισήμανση σύνταξης | 12 |
| 3.1.2 | Αναδίπλωση κώδικα (Code folding) | 12 |
| 3.1.3 | Εντολές σύνταξης και εκτέλεσης | 13 |
| 3.2 | Ολοκληρωμένα Προγραμματιστικά Περιβάλλοντα (IDEs) | 13 |
| 3.2.1 | Επεξεργαστής κειμένου | 13 |
| 3.2.2 | Εργαλεία αποσφαλμάτωσης | 14 |
| 3.2.3 | Μεταγλωττιστής | 14 |
| 3.2.4 | Αυτόματη συμπλήρωση κώδικα | 14 |
| 3.2.5 | Υποστήριξη γλώσσας προγραμματισμού | 14 |
| 3.3 | Προγραμματιστικά περιβάλλοντα της Python | 15 |
| 3.3.1 | IDLE | 15 |
| 3.3.2 | Thonny | 17 |
| 3.3.3 | Spyder | 18 |
| 3.4 | Εργαλεία εκμάθησης προγραμματισμού | 20 |
| 3.4.1 | PILet | 20 |

| | | |
|-------|-----------------------------------------------------------------|----|
| 3.4.2 | Jype | 22 |
| 3.4.3 | MyPyTutor | 26 |
| 3.4.4 | Pythy | 28 |
| 3.4.5 | Online Python Tutor | 30 |
| 3.4.6 | CodeSkulptor | 31 |
| 4 | Σύγκριση Περιβαλλόντων Προγραμματισμού | 34 |
| 4.1 | Βασικά γνωρίσματα περιβαλλόντων προγραμματισμού | 34 |
| 4.2 | Μελέτες Αξιολόγησης Περιβαλλόντων | 38 |
| 4.3 | Συμπεράσματα | 40 |
| 5 | Εισαγωγή στον Προγραμματισμό με την γλώσσα Python | 41 |
| 5.1 | Οφέλη της Python στη διδασκαλία βασικών εννοιών προγραμματισμού | 42 |
| 5.2 | Δυσκολίες που αντιμετωπίζουν οι σπουδαστές | 43 |
| 5.3 | Διδακτικές Προσεγγίσεις | 46 |
| 6 | Μελέτες σύγκρισης με άλλες γλώσσες προγραμματισμού | 50 |
| 7 | Επίλογος | 57 |
| 7.1 | Σύνοψη και συμπεράσματα | 57 |
| 7.2 | Όρια και περιορισμοί της έρευνας | 58 |
| 7.3 | Μελλοντικές Επεκτάσεις | 59 |

Κατάλογος Εικόνων

| | |
|----------------------------------------------------------------------------------------|----|
| Εικόνα 1 - Η διεπαφή του προγράμματος IDLE | 16 |
| Εικόνα 2 - Κώδικας Python στο εργαλείο IDLE..... | 16 |
| Εικόνα 3 - Εκτέλεση προγράμματος και έξοδος προγράμματος..... | 17 |
| Εικόνα 4 - Η διεπαφή του εργαλείου Thonny | 18 |
| Εικόνα 5 - Η διεπαφή του εργαλείου Spyder | 19 |
| Εικόνα 6 - Αρχιτεκτονική του εργαλείου PILEt..... | 20 |
| Εικόνες 7 - Το παζλ προγραμματισμού Parson..... | 22 |
| Εικόνα 8 - Η κεντρική διεπαφή του εργαλείου Jupyter..... | 23 |
| Εικόνα 9 - Απεικόνιση της βασικής ροής δεδομένων | 24 |
| Εικόνα 10 - Απεικόνιση μεταβλητών..... | 25 |
| Εικόνα 11 - Παράδειγμα MyPyTutor: Περιγραφή προβλήματος και Παράθυρο Εκτέλεσης..... | 27 |
| Εικόνα 12 - Παράθυρο επεξεργασίας κώδικα | 27 |
| Εικόνα 13 - Το παράθυρο επεξεργασίας κώδικα του εργαλείου Pythy | 29 |
| Εικόνα 14- Η διεπαφή του εργαλείου OnlinePythonTutor | 30 |
| Εικόνα 15 - Η διεπαφή του εργαλείου CodeSkulptor | 32 |
| Εικόνα 16 - Εμφάνιση μηνύματος..... | 32 |

Κατάλογος Πινάκων

| | |
|-----------------------------------------------------------------------------|----|
| Πίνακας 1 - Χαρακτηριστικά διεπαφής (GUI)..... | 35 |
| Πίνακας 2 - Χαρακτηριστικά συντάκτη | 36 |
| Πίνακας 3 - Χαρακτηριστικά μεταγλώττισης | 36 |
| Πίνακας 4 - Χαρακτηριστικά οπτικοποίησης..... | 38 |
| Πίνακας 5 - Βασικές κατηγορίες δυσκολιών που αντιμετωπίζουν οι μαθητές..... | 46 |
| Πίνακας 6 – Σύγκριση γλωσσών προγραμματισμού [27] | 55 |

1 Εισαγωγή

1.1 Πρόβλημα - Σημαντικότητα του θέματος

Η διδασκαλία των βασικών αρχών και εννοιών του προγραμματισμού προκαλεί αρκετές δυσκολίες στους αρχάριους προγραμματιστές. Οι δυσκολίες αυτές οφείλονται συνήθως στην γλώσσα προγραμματισμού που χρησιμοποιείται στα εισαγωγικά μαθήματα προγραμματισμού αλλά και στα προγραμματιστικά περιβάλλοντα που υπάρχουν και στις μεθόδους διδασκαλίας που χρησιμοποιούνται.

Παρακάτω αναλύεται η γλώσσα προγραμματισμού Python για την εισαγωγή των αρχάριων προγραμματιστών στις αρχές του προγραμματισμού, τα εκπαιδευτικά προγραμματιστικά περιβάλλοντα που χρησιμοποιούνται για την εκμάθηση των βασικών δομών της γλώσσας καθώς και οι διδακτικές προσεγγίσεις που ακολουθούνται ώστε οι αρχάριοι προγραμματιστές να ξεπεράσουν αυτές τις δυσκολίες.

1.2 Σκοπός – Στόχοι

Σκοπός της παρούσας διπλωματικής εργασίας είναι η ανάλυση της γλώσσας προγραμματισμού Python ως εισαγωγικής γλώσσας προγραμματισμού, οι διδακτικές προσεγγίσεις και οι σχετικές μελέτες που έχουν γίνει, και κυρίως η συγκριτική ανάλυση των περιβαλλόντων και των εργαλείων προγραμματισμού που χρησιμοποιούνται για την εισαγωγή μαθητών ή σπουδαστών στον προγραμματισμό με τη γλώσσα αυτή.

1.3 Συνεισφορά

Η μελέτη και η συγκριτική ανάλυση των περιβαλλόντων που έγινε μπορεί να αξιοποιηθεί από διδάσκοντες μαθημάτων προγραμματισμού προκειμένου να επιλέξουν το καταλληλότερο περιβάλλον προγραμματισμού για ένα εισαγωγικό μάθημα προγραμματισμού με την Python, αλλά και οποιονδήποτε άλλον ενδιαφέρεται.

Επίσης, η συγκριτική ανάλυση των περιβαλλόντων μπορεί να αξιοποιηθεί από ερευνητές που ενδιαφέρονται για την σχεδίαση εκπαιδευτικών περιβαλλόντων προγραμματισμού.

Τέλος, η καταγραφή των διδακτικών προσεγγίσεων για την Python μπορεί να αποτελέσει το έναυσμα για μια πιο συστηματική καταγραφή τους που μπορεί να αξιοποιηθεί από εκπαιδευτικούς για την αποτελεσματικότερη διδασκαλία σχετικών μαθημάτων.

1.4 Διάρθρωση της μελέτης

Αρχικά στο Κεφάλαιο 2, γίνεται μια ανάλυση της γλώσσας προγραμματισμού Python, των κύριων χαρακτηριστικών/δομών της αλλά και των τεχνικών προγραμματισμού που υποστηρίζει, όπως δομημένο προγραμματισμό, αντικειμενοστρεφή προγραμματισμό και προγραμματισμό διαδικτύου.

Στο Κεφάλαιο 3, αναλύονται τα περιβάλλοντα προγραμματισμού καθώς και τα εκπαιδευτικά εργαλεία που χρησιμοποιούνται για τη διδασκαλία της γλώσσας προγραμματισμού Python και πραγματοποιείται μια συγκριτική μελέτη αλλά και αξιολόγηση αυτών, με βάση κάποια βασικά γνωρίσματα που χαρακτηρίζουν τα προγραμματιστικά περιβάλλοντα.

Στα Κεφάλαια 4 και 5 πραγματοποιείται διερεύνηση για το αν η γλώσσα προγραμματισμού Python χρησιμοποιείται στη Δευτεροβάθμια και την Τριτοβάθμια εκπαίδευση για την εισαγωγή των μαθητών /σπουδαστών στον προγραμματισμό και τις σχετικές διδακτικές προσεγγίσεις που εφαρμόζονται. Καταγράφονται επίσης οι δυσκολίες και οι παρανοήσεις που αντιμετωπίζουν οι μαθητές/σπουδαστές κατά την εισαγωγή τους στον προγραμματισμό με την γλώσσα Python και διερευνάται ο βαθμός δυσκολίας εκμάθησης της γλώσσας σε σχέση με άλλες γλώσσες προγραμματισμού, όπως για παράδειγμα οι C και Java, οι οποίες χρησιμοποιούνται για εισαγωγή στον προγραμματισμό.

Στο Κεφάλαιο 6, γίνεται μια συγκριτική μελέτη των γλωσσών προγραμματισμού που χρησιμοποιούνται στην διδασκαλία των εισαγωγικών μαθημάτων προγραμματισμού με βάση κριτήρια που βασίζονται σε αποφάσεις σχεδιασμού.

Τέλος, στο Κεφάλαιο 7, γίνεται μια σύνοψη της διπλωματικής εργασίας. Παρουσιάζονται αποτελέσματα και συμπεράσματα από την συγκριτική μελέτη και των

αξιολογήσεων των περιβαλλόντων προγραμματισμού που χρησιμοποιούνται στα εισαγωγικά μαθήματα προγραμματισμού με την Python. Επιπρόσθετα, καταγράφονται τα όρια και οι περιορισμοί που προέκυψαν κατά την έρευνα και διατυπώνονται προτάσεις για μελλοντική έρευνα.

2 Η γλώσσα προγραμματισμού Python

Η Python είναι μια γλώσσα προγραμματισμού υψηλού επιπέδου, η οποία δημιουργήθηκε το 1990 από τον Guido van Rossum, ως μια γλώσσα σεναρίων, με κύριο στόχο την αναγνωσιμότητα του κώδικά της και την ευκολία χρήσης της[14]. Η Python επιτρέπει στους προγραμματιστές να γράφουν σενάρια (scripts) σε λιγότερες γραμμές κώδικα από ότι θα ήταν δυνατόν σε γλώσσες προγραμματισμού όπως η C++ ή η Java. Είναι κατάλληλη και για αρχάριους και για έμπειρους προγραμματιστές. Μπορεί να χρησιμοποιηθεί τόσο για εκπαιδευτικούς σκοπούς όσο και για την ανάπτυξη ολοκληρωμένων εφαρμογών.

Αρχικά, χρησιμοποιούνταν στο λειτουργικό σύστημα Amoeba ως γλώσσα σεναρίων, και αργότερα μετεξελίχθηκε σε γλώσσα προγραμματισμού. Είναι γλώσσα ανοιχτού λογισμικού (open source) και η διαχείριση της γίνεται από τον οργανισμό Python Software Foundation. Η Python θεωρείται διάδοχος της γλώσσας προγραμματισμού ABC, μιας και αυτή υπήρξε η βασική πηγή έμπνευσης για τη δημιουργία της. Η πρώτη δημόσια κυκλοφορία της ήταν το 1991 [54]. Η Python 2.0 κυκλοφόρησε τον Οκτώβριο του 2000 και τον Δεκέμβριο του 2008, κυκλοφόρησε η έκδοση 3.0 (γνωστή και ως py3k ή python 3000) [36]. Ήταν μια σημαντική αναθεώρηση της γλώσσας που δεν είναι εντελώς συμβατή προς τα πίσω με τις προηγούμενες εκδόσεις. Η ημερομηνία λήξης της ζωής της Python 2.7 ορίστηκε το 2015 και κατόπιν αναβλήθηκε για το 2020 από την ανησυχία ότι ένα μεγάλο μέρος του υφιστάμενου κώδικα δεν θα μπορούσε εύκολα να μεταφερθεί στην έκδοση Python 3.

2.1 Χαρακτηριστικά της Python

Η Python υποστηρίζει πολλές τεχνικές προγραμματισμού, συμπεριλαμβανομένων του αντικειμενοστρεφούς και του διαδικαστικού προγραμματισμού. Διαθέτει ένα δυναμικό τύπο συστήματος, αυτόματη διαχείριση της μνήμης και έχει μια μεγάλη και ολοκληρωμένη τυποποιημένη βιβλιοθήκη [54]. Οι διερμηνείς της Python είναι διαθέσιμοι για εγκατάσταση σε πολλά λειτουργικά συστήματα, επιτρέποντας έτσι την εκτέλεση του κώδικα της σε ένα ευρύ φάσμα συστημάτων. Είναι μια γλώσσα που υποστηρίζει ταυτόχρονα και πλήρως τον αντικειμενοστρεφή προγραμματισμό (object-oriented programming), τον δομημένο προγραμματισμό (structured programming) και υπάρχουν ορισμένα χαρακτηριστικά γνωρίσματα της γλώσσας τα οποία υποστηρίζουν τον συναρτησιακό προγραμματισμό (functional programming) και τον θεματοστρεφή προγραμματισμό (aspect-oriented programming) [56]. Ένα σημαντικό χαρακτηριστικό της γλώσσας προγραμματισμού Python, επειδή είναι δυναμική γλώσσα, είναι ότι δεν δηλώνονται οι τύποι των μεταβλητών πριν χρησιμοποιηθούν. Ο τύπος που τους αποδίδεται εξαρτάται από την τιμή που θα πάρουν κατά τη διάρκεια της εκτέλεσης του προγράμματος. Παρακάτω, αναλύονται τα πιο σημαντικά χαρακτηριστικά της γλώσσας.

2.1.1 Απλή

Είναι μια γλώσσα υψηλού επιπέδου που έχει πολλές πηγές εκμάθησης. Η Python υποστηρίζει μια μεγάλη ποικιλία εργαλείων που την καθιστούν πολύ πιο εύκολη στη χρήση και παρακινεί τους χρήστες της να συνεχίσουν να την χρησιμοποιούν. Η Python είναι απλή στη σύνταξη με αποτέλεσμα να είναι πιο εύκολο στον χρήστη να γράφει και να διαβάζει προγράμματα Python σε σύγκριση με άλλες γλώσσες όπως η C++, Java, C#.

2.1.2 Φορητή

Τα σενάρια (scripts) της Python μπορούν να χρησιμοποιηθούν σε διαφορετικά λειτουργικά συστήματα όπως: Windows, Linux, UNIX, Amigo, Mac OS κ.λ.π. Τα προγράμματα της Python μπορούν να μεταφερθούν από μια πλατφόρμα σε μια άλλη και να εκτελεστούν χωρίς να γίνει κάποια αλλαγή σε αυτά [37]. Εκτελείται άψογα σε όλες σχεδόν τις πλατφόρμες συμπεριλαμβανομένων των Windows, Mac OS X, Linux.

2.1.3 Ελεύθερη και ανοιχτού κώδικα

Η γλώσσα προγραμματισμού Python, είναι μια γλώσσα ανοιχτού κώδικα και δεν υπάρχει περιορισμός στο να διαβάσει κάποιος τον κώδικά της, να κάνει αλλαγές σε αυτόν και να χρησιμοποιήσει κομμάτια του σε νέα ελεύθερα προγράμματα. Πέρα από το ότι μπορεί κάποιος να την χρησιμοποιεί ελεύθερα, μπορεί επίσης και να την διανείμει και για εμπορική χρήση. Η Python υποστηρίζεται από μια μεγάλη κοινότητα που την βελτιώνει συνεχώς.

2.1.4 Επεκτάσιμη και με δυνατότητα ενσωμάτωσης

Ας υποθέσουμε ότι μια εφαρμογή που δημιουργεί κάποιος απαιτεί υψηλή απόδοση. Με τη χρήση της γλώσσας προγραμματισμού Python, μπορεί κάποιος εύκολα να συνδυάσει κομμάτια κώδικα από τις C και C++, ή και από άλλες γλώσσες προγραμματισμού ώστε να δώσει στην εφαρμογή υψηλή απόδοση καθώς και την δυνατότητα scripting για τους χρήστες της εφαρμογής.

2.1.5 Γλώσσα υψηλού επιπέδου και διερμηνυόμενη

Σε αντίθεση με τις γλώσσες προγραμματισμού C ή C++, δεν υπάρχει λόγος ανησυχίας όσον αφορά τη διαχείριση της μνήμης του υπολογιστή που χρησιμοποιείται από τα προγράμματα που έχουν γραφτεί. Επίσης, σε αντίθεση με την C, C++ οι οποίες μετατρέπουν τον πηγαίο κώδικα, σε μια γλώσσα που μιλάει ο υπολογιστής (δυναμικός κώδικας) με την χρήση ενός μεταγλωττιστή, η Python εκτελεί το πρόγραμμα απευθείας από τον πηγαίο κώδικα. Εσωτερικά, η Python μετατρέπει τον πηγαίο κώδικα σε μια ενδιάμεση μορφή που ονομάζεται bytecode και μετά το μεταφράζει στη γλώσσα του υπολογιστή και μετά το τρέχει [41]. Όλη αυτή η διαδικασία κάνει την χρήση της Python πολύ πιο εύκολη αφού δεν υπάρχει λόγος ανησυχίας για την μεταγλώττιση του προγράμματος όπως και για την σύνδεση με τις κατάλληλες βιβλιοθήκες.

2.1.6 Αντικειμενοστρεφής

Η Python υποστηρίζει τόσο τον διαδικαστικό προγραμματισμό (procedure oriented) όσο και τον αντικειμενοστρεφή προγραμματισμό (object oriented). Στον διαδικαστικό προγραμματισμό, το πρόγραμμα δομείται πάνω σε διαδικασίες ή συναρτήσεις. Στις αντικειμενοστρεφείς γλώσσες, το πρόγραμμα δομείται πάνω σε αντικείμενα τα οποία συνδυάζουν δεδομένα και λειτουργικότητα. Η Python έχει έναν

πολύ ισχυρό, αλλά πολύ απλό τρόπο για αντικειμενοστρεφή προγραμματισμό σε σχέση με άλλες γλώσσες όπως η C++ ή η Java.

2.1.7 Εκτεταμένες βιβλιοθήκες

Η πρότυπη βιβλιοθήκη της Python είναι πολύ μεγάλη. Μπορεί να βοηθήσει κάποιον να κάνει διάφορα πράγματα σχετικά με τις κανονικές εκφράσεις, βάσεις δεδομένων, περιηγητές ιστού, FTP, email, XML, HTML, αρχεία WAV, κρυπτογράφηση και άλλες λειτουργίες που εξαρτώνται από το σύστημα [41]. Επιπλέον, από την πρότυπη βιβλιοθήκη υπάρχουν διάφορες βιβλιοθήκες υψηλής ποιότητας όπως η wxPython, η Twisted, η PythonImagingLibrary και πολλές άλλες.

3 Περιβάλλοντα Προγραμματισμού

3.1 Επεξεργαστές κειμένου-κώδικα (Code Editors)

Ένας επεξεργαστής κειμένου κώδικα είναι ένα εργαλείο το οποίο χρησιμοποιείται για να γράφει κανείς και να επεξεργάζεται κώδικα. Οι περισσότεροι επεξεργαστές κειμένου είναι απλά (ελαφριά), προγράμματα που μπορούν να τρέξουν σε κάθε υπολογιστή [47]. Οι απλοί επεξεργαστές κειμένου είναι ιδανικοί για μικρά προγράμματα – έργα λογισμικού που δεν έχουν μεγάλη πολυπλοκότητα. Επίσης είναι ιδανικοί για μικρές διορθώσεις στον κώδικα. Ένα τελευταίο στοιχείο είναι ότι οι απλοί επεξεργαστές κειμένου είναι ιδανικοί για μικρές και πολύ μικρές ομάδες λογισμικού που μπορούν να συνεννοούνται για τον κώδικα πολύ εύκολα. Ωστόσο, όταν τα προγράμματα και τα έργα λογισμικού είναι μεγαλύτερα και πιο πολύπλοκα τότε χρειαζόμαστε πιο σύνθετα και ολοκληρωμένα προγραμματιστικά περιβάλλοντά όπως είναι τα IDE (Integrated Development Environment) [12]. Παρότι είναι απλοί επεξεργαστές κειμένου κώδικα, έχουν κάποια χρήσιμα χαρακτηριστικά για τους προγραμματιστές:

3.1.1 Επισήμανση σύνταξης

Είναι μια λειτουργία των επεξεργαστών κειμένου που χρησιμοποιούνται για προγραμματισμό. Η λειτουργία αυτή εμφανίζει κείμενο, ειδικά τον πηγαίο κώδικα, σε διαφορετικά χρώματα και γραμματοσειρές [48]. Αυτή η δυνατότητα, διευκολύνει τη συγγραφή κώδικα σε μια γλώσσα προγραμματισμού, καθώς τόσο οι δομές όσο και τα σφάλματα σύνταξης είναι οπτικά διακριτά. Με αυτό τον τρόπο γίνεται πιο εύκολη η ανάγνωση του κώδικα και ο εντοπισμός τυχόν σφαλμάτων. Για παράδειγμα, μπορεί ένας χρήστης που γράφει ένα πρόγραμμα σε Python, να ρυθμίσει την επισήμανση σύνταξης έτσι ώστε οι λέξεις-κλειδιά (keywords) της Python να είναι με χρώμα μπλε, τα σχόλια (comments) να είναι πράσινα, οι συμβολοσειρές (strings) να είναι πορτοκαλί και ούτω καθεξής.

3.1.2 Αναδίπλωση κώδικα (Code folding)

Είναι ένα χαρακτηριστικό κάποιων επεξεργαστών κειμένου - κώδικα, που επιτρέπει στον χρήστη να επιλέγει την απόκρυψη ή την εμφάνιση τμημάτων κώδικα στο αρχείο που επεξεργάζεται εκείνη την στιγμή [47]. Με αυτό τον τρόπο δίνεται η

δυνατότητα στον χρήστη να διαχειρίζεται «μεγάλες ποσότητες» κώδικα και να προβάλει μόνο εκείνα τα κομμάτια κώδικα που χρειάζεται να επεξεργαστεί εκείνη την στιγμή.

3.1.3 Εντολές σύνταξης και εκτέλεσης

Κάποιοι επεξεργαστές κειμένου - κώδικα δίνουν τη δυνατότητα στον χρήστη να εκτελέσει και να μεταγλωττίσει το πρόγραμμά του χωρίς να χρειάζεται να ανοίξει κάποιο παράθυρο τερματικού. Επομένως, το debugging στον κώδικα μπορεί να γίνει σε ένα μόνο σημείο.

3.2 Ολοκληρωμένα Προγραμματιστικά Περιβάλλοντα (IDEs)

Τα ολοκληρωμένα προγραμματιστικά περιβάλλοντα (IDE – Integrated Development Environment) είναι εφαρμογές λογισμικού που βοηθούν τον προγραμματιστή στη ανάπτυξη προγραμμάτων [12]. Συνήθως το περιβάλλον περιλαμβάνει επεξεργαστή πηγαίου κώδικα, εργαλεία αυτόματης συμπλήρωσης και διερμηνευτή, όμως σε πολλές περιπτώσεις ενσωματώνονται και άλλες λειτουργίες, όπως μεταγλωττιστής, εργαλεία κατασκευής γραφικού περιβάλλοντος χρήστη (GUI – Graphic User Interface), προγράμματα κλάσεων και αντικειμένων για χρήση σε αντικειμενοστρεφή ανάπτυξη λογισμικού [47]. Ένα IDE έχει τις κατάλληλες μεθόδους που αναλύουν και καταλαβαίνουν τον κώδικα σε αντίθεση με τους απλούς κειμενογράφους που πολλές φορές μπορούν να προσφέρουν μόνο μια βασική χρωματική σήμανση στις εντολές. Ένα IDE μπορεί να προσθέτει αυτόματα τις κατάλληλες βιβλιοθήκες, να εντοπίζει και να διορθώνει συντακτικά λάθη καθώς επίσης να προσφέρει εργαλεία αποσφαλμάτωσης του κώδικα (debugging) και ελέγχου του κώδικα (testing). Παρακάτω παρουσιάζονται τα βασικά χαρακτηριστικά που έχουν τα ολοκληρωμένα προγραμματιστικά περιβάλλοντα.

3.2.1 Επεξεργαστής κειμένου

Σχεδόν κάθε ολοκληρωμένο προγραμματιστικό περιβάλλον (IDE), έχει έναν επεξεργαστή κειμένου σχεδιασμένο να μπορεί ο χρήστης να γράφει και να επεξεργάζεται πηγαίο κώδικα. Ορισμένα εργαλεία ενδέχεται να έχουν οπτικά στοιχεία για μεταφορά και απόθεση (drag and drop) αλλά τα περισσότερα διαθέτουν μια απλή διεπαφή με επισημάνσεις σύνταξης ανάλογα την γλώσσα που χρησιμοποιεί ο χρήστης.

3.2.2. Εργαλεία αποσφαλμάτωσης

Τα εργαλεία εντοπισμού σφαλμάτων βοηθούν τους χρήστες να εντοπίζουν και να διορθώνουν τα σφάλματα στον πηγαίο κώδικα τους. Συχνά προσομοιώνουν σενάρια του πραγματικού κόσμου για να μπορεί ο χρήστης να δοκιμάζει και να ελέγχει την λειτουργία και την απόδοση του κώδικά του. Οι προγραμματιστές και οι μηχανικοί λογισμικού δοκιμάζουν και ελέγχουν συνήθως τα διάφορα τμήματα του κώδικα ώστε να αναγνωρίζουν τυχόν σφάλματα πριν δημοσιεύσουν την εφαρμογή τους.

3.2.3 Μεταγλωττιστής

Οι μεταγλωττιστές είναι προγράμματα που μεταφράζουν την γλώσσα προγραμματισμού σε μια μορφή που μπορούν να επεξεργαστούν οι μηχανές, όπως ο κώδικας σε γλώσσα μηχανής. Ο μεταγλωττιστής αναλύει και βελτιστοποιεί τον κώδικα για να βελτιστοποιήσει την απόδοση.

3.2.4 Αυτόματη συμπλήρωση κώδικα

Οι λειτουργίες συμπλήρωσης κώδικα βοηθούν τους προγραμματιστές με την έξυπνη ταυτοποίηση και εισαγωγή τμημάτων κώδικα. Αποθηκεύουν τμήματα του κώδικα κατά τη διάρκεια που οι προγραμματιστές τον συντάσσουν, τον επεξεργάζονται ή τον δημιουργούν με αποτέλεσμα να μειώνουν την πιθανότητα τυπογραφικών και σφαλμάτων κώδικα.

3.2.5 Υποστήριξη γλώσσας προγραμματισμού

Ένα ολοκληρωμένο περιβάλλον προγραμματισμού (IDE) μπορεί να σχεδιαστεί και να αναπτυχθεί ως εργαλείο για μια γλώσσα προγραμματισμού αν και συνήθως τα περισσότερα υποστηρίζουν πολλές γλώσσες προγραμματισμού. Ως εκ τούτου, πριν ξεκινήσει κάποιος να γράφει κώδικα θα πρέπει να επιλέξει το κατάλληλο IDE. Η γλώσσα προγραμματισμού Python υποστηρίζεται από αρκετά προγραμματιστικά εργαλεία για τη συγγραφή κώδικα.

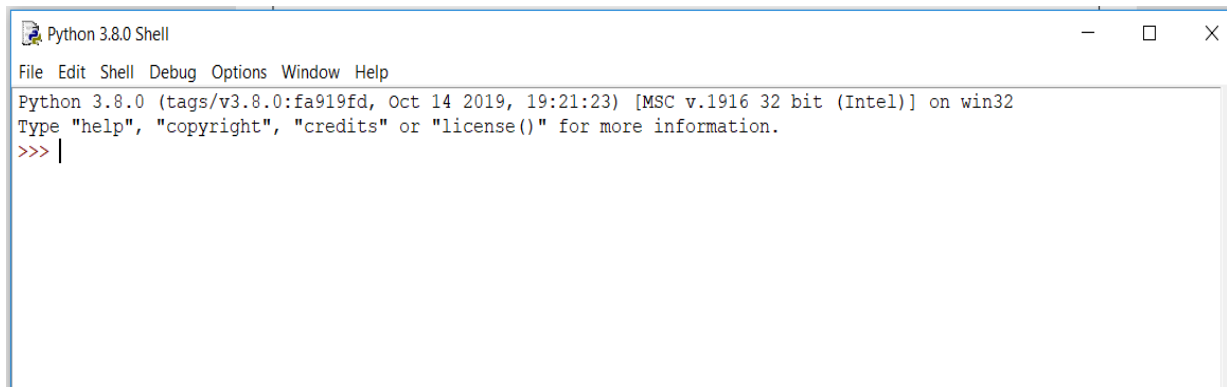
3.3 Προγραμματιστικά περιβάλλοντα της Python

Στην ενότητα αυτή θα παρουσιαστούν προγραμματιστικά περιβάλλοντα τα οποία χρησιμοποιούνται στη διδασκαλία και στην εισαγωγή στον προγραμματισμό με την γλώσσα Python, διανέμονται ελεύθερα και ικανοποιούν τις ιδιαίτερες απαιτήσεις των αρχάριων προγραμματιστών.

3.3.1 IDLE

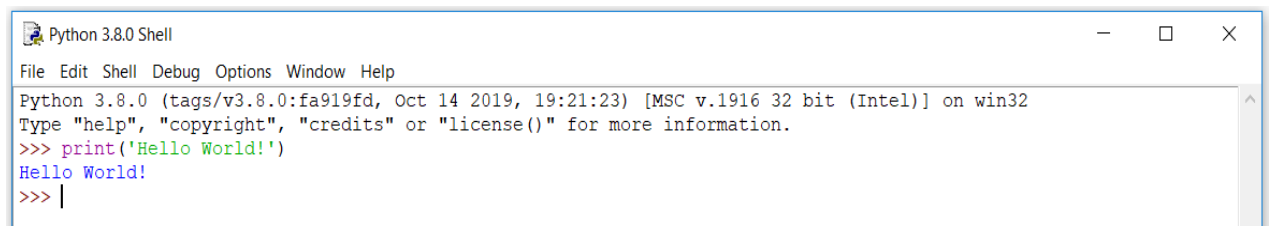
Το Ολοκληρωμένο Περιβάλλον Ανάπτυξης Προγραμμάτων IDLE (Integrated Development Environment) της Python, είναι ένα ελεύθερο λογισμικό, που κατασκευάστηκε από τον Guido van Rossum το δημιουργό της Python, έχοντας ως σκοπό να αποτελέσει ένα περιβάλλον ανάπτυξης προγραμμάτων της Python, κατάλληλο για αρχάριους προγραμματιστές και για εκπαιδευτικούς [13]. Το λογισμικό είναι διαθέσιμο για Windows, Linux, Mac OS X και βρίσκεται στην ιστοσελίδα <https://www.python.org/downloads/> από όπου μπορεί να ληφθεί δωρεάν και να εγκατασταθεί τοπικά στον υπολογιστή μας. Είναι ο ευκολότερος τρόπος για κάποιον να ξεκινήσει να γράφει κώδικα σε Python, καθώς δεν χρειάζονται περαιτέρω εγκαταστάσεις και ρυθμίσεις από την αρχική εγκατάσταση της Python. Το IDLE είναι στην πραγματικότητα ένα πρόγραμμα Python που χρησιμοποιεί το εργαλείο GUI της βιβλιοθήκης tkinter ώστε να δημιουργήσει τα παράθυρά του [36]. Στο περιβάλλον IDLE, ο προγραμματιστής έχει τη δυνατότητα να πληκτρολογεί εντολές απευθείας στο διερμηνευτή της Python, οι οποίες εκτελούνται και εμφανίζουν τα αποτελέσματα άμεσα. Επιπλέον, το περιβάλλον διαθέτει λειτουργίες αυτόματων εσοχών όπως επίσης επισήμανσης και αυτόματης συμπλήρωσης εντολών. Τέλος, σε περίπτωση συντακτικού λάθους το περιβάλλον εμφανίζει μηνύματα σχετικά με το είδος και το σημείο του σφάλματος, προκειμένου να διευκολυνθούν οι χρήστες στη διόρθωσή του.

Το Python IDLE υπάρχει συνήθως ως καταχώρηση στο μενού του κουμπιού «Εναρξη» στα Windows. Όταν ξεκινήσει, εμφανίζει κάποιες πληροφορίες σχετικά με την έκδοση Python και το λειτουργικό σύστημα (Εικόνα 1).



Εικόνα 1 - Η διεπαφή του προγράμματος IDLE

Γράφουμε τον κώδικα μας μετά την εντολή `>>>` και όταν πατηθεί το Enter τότε εκτελείται ο κώδικας μας (Εικόνα 2).

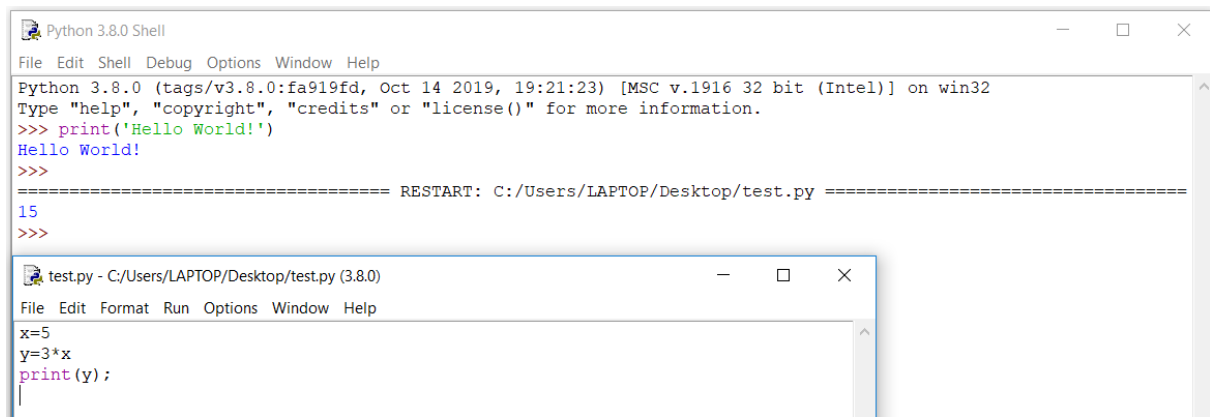


Εικόνα 2 - Κώδικας Python στο εργαλείο IDLE

Για την συγγραφή ολοκληρωμένων προγραμμάτων δεν χρησιμοποιείτε το Shell, αντ' αυτού το Python IDLE έρχεται με έναν ενσωματωμένο επεξεργαστή κειμένου που μπορείτε να χρησιμοποιήσετε για να γράψετε και να αποθηκεύσετε τον κώδικα σας. Ανοίγουμε τον επεξεργαστή από το μενού File -> New File και γράφουμε τον κώδικα μας. Το αρχείο που δημιουργούμε και στο οποίο γράφουμε τον κώδικα Python, το αποθηκεύουμε με κατάληξη .py

Για την εκτέλεση του προγράμματος ακολουθούμε τα εξής βήματα στο μενού του IDLE:

- Ανοίγουμε το αρχείο και το «φορτώνουμε» στο περιβάλλον επιλέγοντας File -> Open
- Στη συνέχεια επιλέγουμε στο μενού Run -> Run Module ή (με το F5).



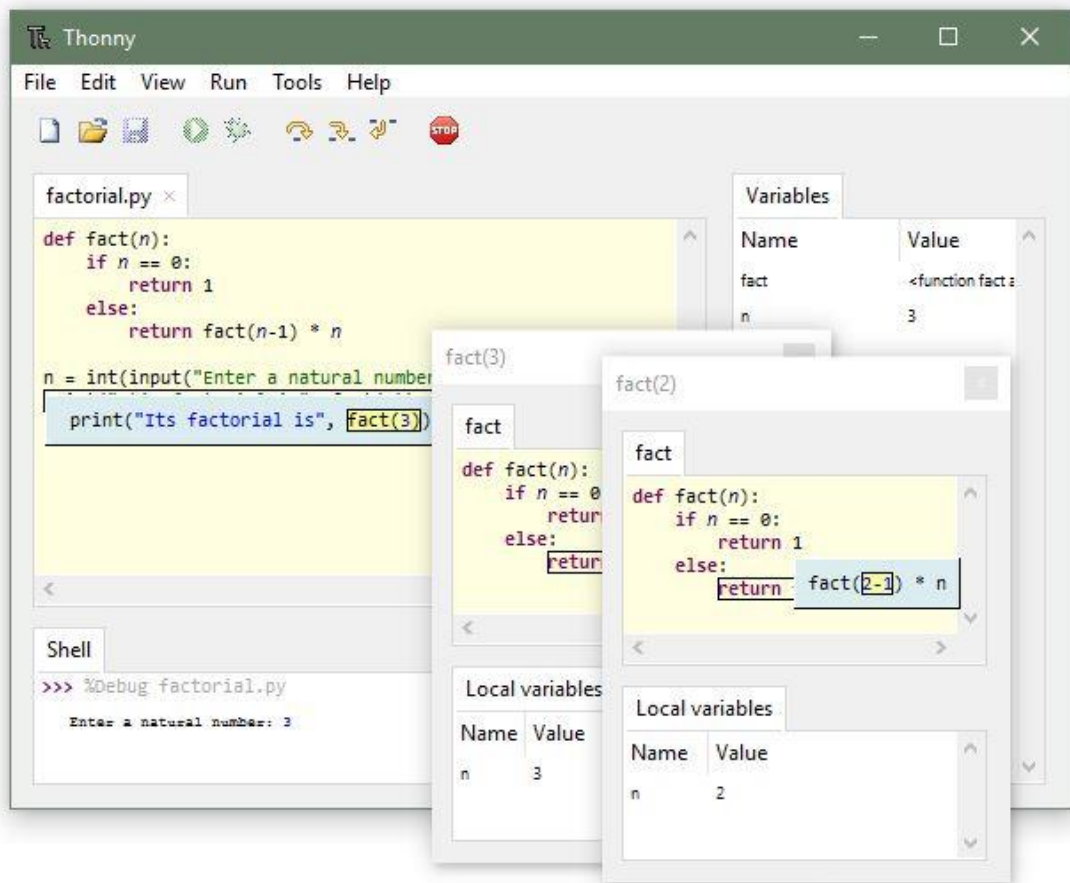
```
Python 3.8.0 Shell
File Edit Shell Debug Options Window Help
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:21:23) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print('Hello World!')
Hello World!
>>>
===== RESTART: C:/Users/LAPTOP/Desktop/test.py =====
15
>>>

test.py - C:/Users/LAPTOP/Desktop/test.py (3.8.0)
File Edit Format Run Options Window Help
x=5
y=3*x
print(y);
|
```

Εικόνα 3 - Εκτέλεση προγράμματος και έξοδος προγράμματος

3.3.2 Thonny

Το Thonny είναι ένα ολοκληρωμένο προγραμματιστικό περιβάλλον της Python, το οποίο έχει σχεδιαστεί και υλοποιηθεί από τον Aivar Annamaa στο Πανεπιστήμιο Τάρτου της Εσθονίας (University of Tartu). Χρησιμοποιήθηκε από φοιτητές για το εισαγωγικό μάθημα στον προγραμματισμό στο Πανεπιστήμιο της Τάρτου, ως εναλλακτική λύση στην χρήση του περιβάλλοντος IDLE, το οποίο είναι το προεπιλεγμένο περιβάλλον για προγραμματισμό Python [1]. Παρέχει ένα φιλικό και κατάλληλο περιβάλλον προγραμματισμού για αρχάριους προγραμματιστές αλλά και για μαθητές, προσφέροντας παράλληλα πολλά από τα χαρακτηριστικά και τις επιλογές που έχουν τα σύγχρονα IDE's. Περιέχει σημαντικά χαρακτηριστικά που χρειάζονται οι αρχάριοι προγραμματιστές, όπως εργαλεία εντοπισμού σφαλμάτων και επίσης μια διεπαφή όπου εμφανίζει πληροφορίες για την κάθε μια εντολή του κώδικα που εκτελείται ξεχωριστά. Αυτή η λειτουργία της «βήμα-βήμα» εκτέλεσης του προγράμματος είναι ένας πολύ ισχυρό μέσο κατανόησης του τρόπου με τον οποίο λειτουργεί μια γλώσσα προγραμματισμού [44]. Για παράδειγμα, όταν ξεκινάει μια έκφραση, ο Thonny παρουσιάζει όλες τις ενδιάμεσες λειτουργίες και εμφανίζει την τρέχουσα τιμή της έκφρασης και των μεταβλητών. Αυτή η δυνατότητα απεικόνισης της λειτουργίας του προγράμματος είναι εξαιρετικά χρήσιμη για τους αρχάριους προγραμματιστές και επίσης για τον εντοπισμό σφαλμάτων στον κώδικα τους.



Εικόνα 4 - Η διεπαφή του εργαλείου Thonny

Το Thonny είναι ένα ελεύθερο και ανοικτού κώδικα λογισμικό και είναι διαθέσιμο για Windows, Linux, Mac OS X και βρίσκεται στην ιστοσελίδα <https://www.python.org/downloads/> από όπου μπορεί να ληφθεί δωρεάν και να εγκατασταθεί τοπικά στον υπολογιστή μας.

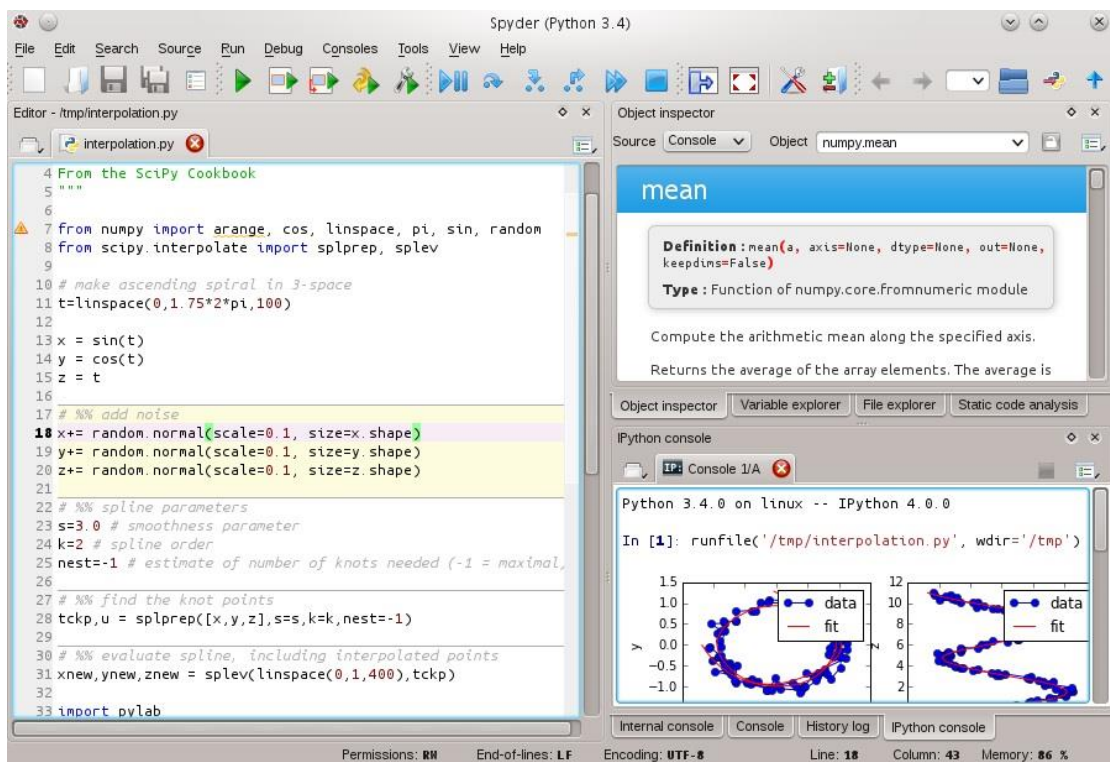
3.3.3 Spyder

Το Spyder είναι ένα ολοκληρωμένο προγραμματιστικό περιβάλλον (IDE) ανοικτού κώδικα που χρησιμοποιείται συνήθως στην ανάπτυξη προγραμμάτων για επιστημονικούς σκοπούς [40]. Η εγκατάσταση και η ενεργοποίηση του συγκεκριμένου IDE, απαιτεί επίσης και την εγκατάσταση της συλλογής πακέτων Anaconda.

Η Anaconda είναι μια ελεύθερη και ανοικτού κώδικα συλλογή πακέτων, των γλωσσών προγραμματισμού Python και R, για την επιστημονική πληροφορική (επιστήμη δεδομένων, εφαρμογές μηχανικής μάθησης, επεξεργασία δεδομένων μεγάλης κλίμακας,

προγνωστικά αναλύσεις κλπ.), που αποσκοπεί στην απλοποίηση της διαχείρισης πακέτων [2].

Το Spyder IDE είναι κατάλληλο εργαλείο για αρχάριους προγραμματιστές και χρήστες της Python. Το γραφικό του περιβάλλον είναι πολύ απλό με σκοπό οι αρχάριοι προγραμματιστές να βρίσκουν εύκολα ότι χρειάζονται. Σημαντικό χαρακτηριστικό του IDE Spyder, είναι ότι το χρησιμοποιεί πολύ το επιστημονικό κοινό που ασχολείται με την επιστήμη των δεδομένων και την ανάλυσή τους μέσω της γλώσσας προγραμματισμού Python, και ο λόγος είναι ότι ενσωματώνει πολλές βιβλιοθήκες της Python όπως οι SciPy, NumPy και Matplotlib.



Εικόνα 5 - Η διεπαφή του εργαλείου Spyder

Υποστηρίζει όλες τις λειτουργίες όπου μπορεί να διαθέτει ένα IDE, όπως την χρωματική επισήμανση σύνταξης (syntax highlighting), την αυτόματη συμπλήρωση εντολών (autocompletion) και το εργαλείο αποσφαλμάτωσης (debugger) για την παρακολούθηση της εκτέλεσης του κώδικα με σκοπό την εύρεση σφαλμάτων. Τέλος το συγκεκριμένο Python IDE, είναι δωρεάν διαθέσιμο για όλες τις πλατφόρμες όπως Windows, macOS και Linux και βρίσκεται στην ιστοσελίδα <https://www.spyder-ide.org/>.

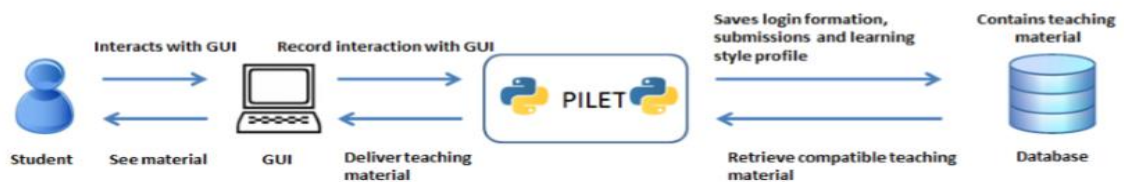
3.4 Εργαλεία εκμάθησης προγραμματισμού

3.4.1 PILEt

Είναι ένα διαδραστικό εργαλείο που έχει σχεδιαστεί για να διδάσκει την γλώσσα προγραμματισμού Python. Το εργαλείο ενσωματώνει διάφορες μεθόδους διδασκαλίας (οπτικοποίηση, παραδείγματα και ασκήσεις, πάζλ) κατάλληλες για διαφορετικούς εκπαιδευόμενους. Στο τέλος κάθε μαθήματος υπάρχουν ερωτήσεις πολλαπλών επιλογών, ώστε να αξιολογούν οι μαθητές/ σπουδαστές ως προς την κατανόηση των εννοιών στο μάθημα που έχουν διδαχθεί [4]. Με αυτό τον τρόπο, κάθε μαθητής μπορεί να μάθει με την τεχνική διδασκαλίας που είναι πιο κοντά σε αυτόν ή να χρησιμοποιήσει ένα μείγμα από διαφορετικούς τρόπους για να υποστηρίξουν την μάθηση του.

Το PILEt αποτελείται από τρία επίπεδα:

1. Ένα γραφικό περιβάλλον εργασίας του χρήστη (GUI) που εμφανίζει το εκπαιδευτικό υλικό στον χρήστη.
2. Το σύστημα PILEt που παρέχει διαφορετικές πηγές μάθησης.
3. Στο backend μέρος του συστήματος υπάρχει μια βάση δεδομένων που αποθηκεύει τις πληροφορίες σύνδεσης του χρήστη, τις υποβολές μαθημάτων, το προφίλ «μάθησης» του χρήστη καθώς και τις διαφορετικές πηγές μάθησης που χρησιμοποιούνται για τη διδασκαλία της κάθε ενότητας που επιλέγει ο μαθητής.



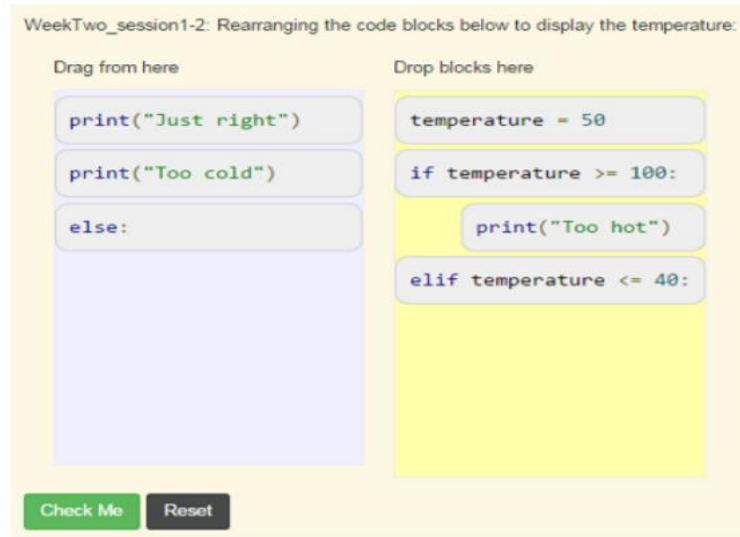
Εικόνα 6 - Αρχιτεκτονική του εργαλείου PILEt

Ο μαθητής έχει πρόσβαση στο υλικό του μαθήματος χρησιμοποιώντας το GUI. Το PILEt καταγράφει την συμπεριφορά του μαθητή με διαφορετικό GUI, και δημιουργεί ένα προφίλ μάθησης για τον μαθητή με βάση αυτές τις αλληλεπιδράσεις. Στη συνέχεια

το σύστημα PILEt χρησιμοποιεί το προφίλ του μαθητή για να επιλέξει το καταλληλότερο διδακτικό υλικό και να το μεταφέρει στην διεπαφή του χρήστη. Ως εκ τούτου, όσο περισσότερο χρησιμοποιεί ο μαθητής το εργαλείο, τόσο υψηλότερη είναι η αντιστοιχία μεταξύ του υλικού που παρουσιάζεται στον μαθητή και της μαθησιακής του προτίμησης.

Το υλικό αυτό είναι:

1. Διδακτικό υλικό με την μορφή σημειώσεων ή διαφανειών.
2. Μια εικονική επεξήγηση του μαθήματος είτε:
 - a. μέσω βίντεο με ζωντανή σύνδεση (live) επίδειξης του μαθήματος.
 - b. μέσω στατικών πινάκων, διαγραμμάτων ροής και εικόνων που περιγράφουν τις διάφορες καταστάσεις της δραστηριότητας ενός προγράμματος.
 - c. Μέσω του εργαλείου Online Python Tutor, το οποίο είναι ένα εργαλείο που ενσωματώνει την κινούμενη απεικόνιση βήμα προς βήμα εκτέλεσης του κώδικα και των δομών δεδομένων.
3. Παραδείγματα από εκτελέσιμο κώδικα σε Python που μπορεί ο μαθητής να τα επεξεργαστεί και να τα μεταγλωττίσει. Αυτό ενθαρρύνει τους μαθητές να πειραματιστούν με τον κώδικα π.χ. «Τι συμβαίνει αν κάνω αυτό, αντί αυτού;».
4. Ασκήσεις προγραμματισμού που μπορούν να υλοποιηθούν σε έναν ενσωματωμένο διερμηνέα, με μια λύση των ασκήσεων ως δείγμα και μια καρτέλα συζήτησης. Η λύση των ασκήσεων εμφανίζεται μετά από δέκα προσπάθειες που κάνει ο μαθητής. Μέσω της καρτέλας συζήτησης οι μαθητές ενθαρρύνονται να μιλούν για τις διαφορετικές λύσεις των προβλημάτων που αντιμετωπίζουν και για την αποτελεσματικότητα του κώδικα τους. Ο εκπαιδευτής προβάλλει τις υποβολές των μαθητών ώστε να διασφαλίσει ότι δεν θα δημοσιευθούν προφανείς αποδείξεις ή λύσεις.
5. Το παζλ προγραμματισμού Parson [10]: είναι μια άσκηση που περιλαμβάνει την αναδιάταξη τμημάτων κώδικα με την σωστή σειρά. Η φύση των ασκήσεων αυτών προάγει την συλλογιστική και τις λογικές δεξιότητες και δίνει κίνητρο στους μαθητές για μάθηση.



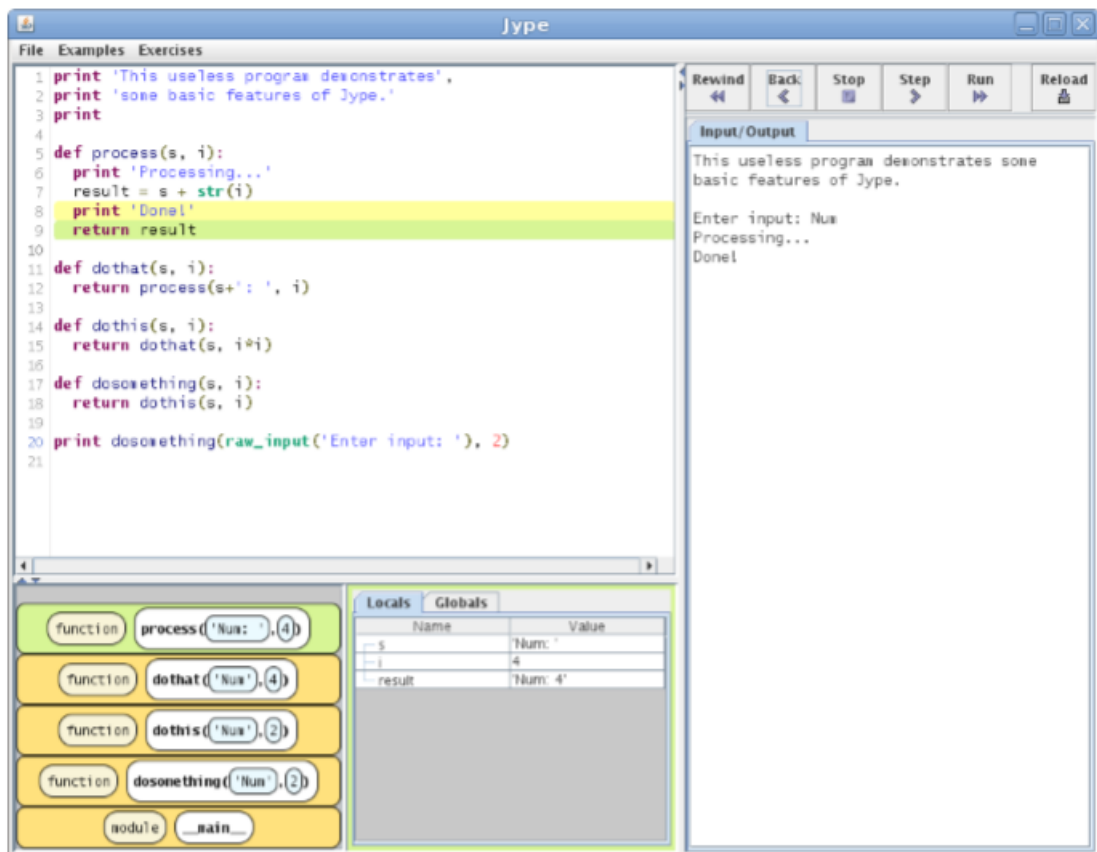
Εικόνες 7 - Το παζλ προγραμματισμού Parson

6. Αξιολόγηση στο τέλος κάθε συνεδρίας με τη μορφή ερωτήσεων πολλαπλών επιλογών. Αυτή η διαδικασία βοηθά τους μαθητές να αναθεωρήσουν, να διατηρήσουν μια επαφή με το μάθημα και να εντοπίσουν τυχόν κενά που μπορεί να έχουν, ώστε ο εκπαιδευτής να τους προσφέρει επιπλέον υποστήριξη.

Το εργαλείο Pilet συνδυάζει μερικές από τις πιο δημοφιλείς παιδαγωγικές δεξιότητες προγραμματισμού που αυξάνουν αποτελεσματικά το ενδιαφέρον των μαθητών και τους δίνει κίνητρο να μάθουν και απευθύνεται σε όλους τους μαθητές ανεξαρτήτως τις ικανότητες που έχει ο καθένας [4]. Επιτρέπει στον εκπαιδευτή να παρακολουθεί την εξέλιξη των μαθητών και να τους υποστηρίζει σε όποιο πρόβλημα και αν προκύψει.

3.4.2 *Jype*

Είναι ένα διαδικτυακό εργαλείο που παρέχει ένα περιβάλλον απεικόνισης της εκτέλεσης των προγραμμάτων Python γραμμή – γραμμή. Μέσω αυτού του εργαλείου μπορεί ο μαθητής να επιλύσει ασκήσεις προγραμματισμού και να χρησιμοποιήσει λειτουργίες για τον εντοπισμό σφαλμάτων στα προγράμματα που διαχειρίζεται.

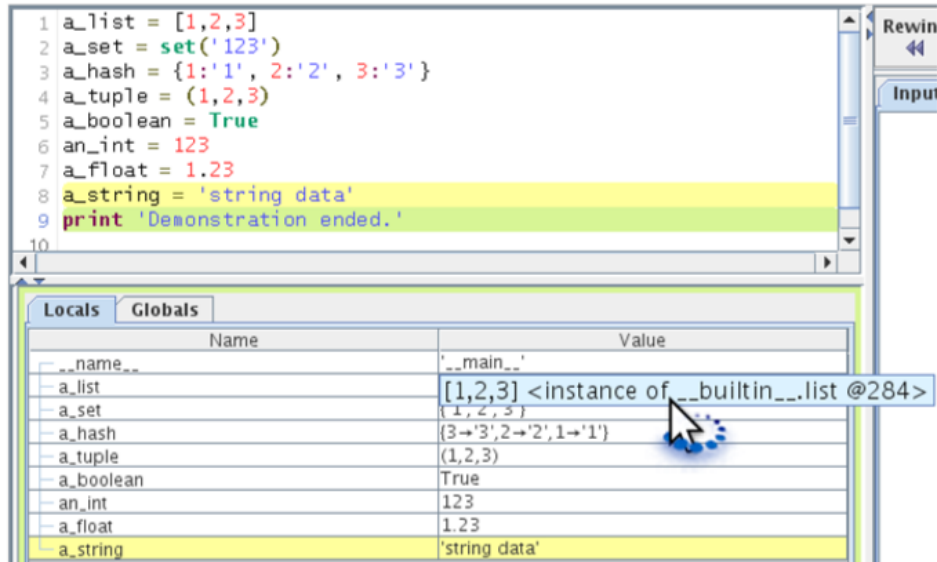


Εικόνα 8 - Η κεντρική διεπαφή του εργαλείου Jypy

Η διεπαφή του προγράμματος Jypy είναι χωρισμένη σε τέσσερα τμήματα. Το πρώτο τμήμα είναι ένας επεξεργαστής κειμένου-κώδικα που υποστηρίζει την επισήμανση σύνταξης. Το δεύτερο τμήμα κάτω από τον επεξεργαστή κειμένου είναι μια περιοχή που απεικονίζει τα δεδομένα και την κατάσταση των μεταβλητών κατά την εκτέλεση του κώδικα. Δίπλα υπάρχει η καρτέλα Input/Output όπου εμφανίζονται εντολές για εισαγωγή στοιχείων, σταθερών ή συμβολοσειρών και επίσης απεικονίζει το αποτέλεσμα εξόδου εκτέλεσης του προγράμματος. Τέλος, πάνω από την καρτέλα Input/Output υπάρχει μια μπάρα με εργαλεία ελέγχου.

Η εκτέλεση του προγράμματος μπορεί να εξεταστεί με βάση δύο κατηγορίες: ροή ελέγχου και ροή δεδομένων. Στο σύστημα Jypy, η πρόοδος της ροής ελέγχου απεικονίζεται στον κώδικα, όπου η τρέχουσα γραμμή εκτέλεσης επισημαίνεται με πράσινο χρώμα και η γραμμή που εκτελέστηκε προηγουμένως επισημαίνεται με κίτρινο χρώμα. Όταν δημιουργείται μια εξαίρεση που διακόπτει την εκτέλεση του προγράμματος, η γραμμή από την οποία προέκυψε η εξαίρεση, επισημαίνεται με κόκκινο χρώμα [21]. Συνολικά οι διάφορες οντότητες που απεικονίζονται στο εργαλείο Jypy

αναγνωρίζονται και διακρίνονται από το χρώμα και άλλα χαρακτηριστικά, τα οποία επιτρέπουν στον μαθητή να διακρίνει γρήγορα τις περιοχές στις οποίες πρέπει να επικεντρωθεί.



Εικόνα 9 - Απεικόνιση της βασικής ροής δεδομένων

Οι παραπάνω εικόνες (Εικόνα 9) και (Εικόνα 10) δείχνουν την βασική απεικόνιση της ροής δεδομένων στο εργαλείο Jyре. Τα ονόματα και οι τιμές των μεταβλητών εμφανίζονται σε έναν πίνακα. Όταν μια νέα μεταβλητή εισάγεται, δηλαδή ένα όνομα μεταβλητής δεσμεύεται σε μια τιμή για πρώτη φορά, η αντίστοιχη γραμμή του πίνακα επισημαίνεται με κίτρινο χρώμα, ταιριάζοντας το χρώμα στην προβολή κώδικα. Όταν ένα υπάρχον όνομα μεταβλητής δεσμεύεται σε διαφορετική τιμή τότε παίρνει χρώμα κίτρινο μόνο η στήλη με τις τιμές.

```

1 class Slot(object):
2     def set_value(self, value):
3         self.value = value
4     def get_value(self, value):
5         return self.value
6
7 myslot = Slot()
8 myslot.set_value(7)
9 a_list = [1,2,3]
10 list_ref = a_list
11 a_list.append(4)
12 a_list.append(5)
13

```

| Locals | | Globals | |
|-----------|-------------------|---------|-------|
| Name | Value | Name | Value |
| __name__ | '__main__' | | |
| Slot | <class> | | |
| myslot | <Slot> | | |
| get_value | <instance method> | | |
| set_value | <instance method> | | |
| value | 7 | | |
| a_list | [1,2,3,4] | | |
| list_ref | [1,2,3,4] | | |

Εικόνα 10 - Απεικόνιση μεταβλητών

Η εκτέλεση των προγραμμάτων στο εργαλείο Jyve ελέγχεται με βασικά κουμπιά τύπου VCR που βρίσκονται πάνω δεξιά όπως φαίνεται στην Εικόνα 8. Ο χρήστης μπορεί εύκολα να εναλλάσσεται μεταξύ επεξεργασίας και εκτέλεσης του κώδικα. Δηλαδή, αν ο χρήστης τροποποιήσει τον κώδικα του ενώ εκτελείται, τότε η εκτέλεση του κώδικα σταματά αυτόματα. Το εργαλείο Jyve μπορεί να φιλοξενήσει δύο τύπους περιεχομένου: παραδείγματα και ασκήσεις. Τα παραδείγματα παρέχουν στους μαθητές παιδαγωγικά χαρακτηριστικά, διότι μέσα από την απεικόνιση του κώδικα, οι μαθητές βλέπουν πως έχει δομηθεί ο κώδικας και μπορούν να εντοπίσουν τυχόν σφάλματα που μπορεί να έχουν αυτά τα παραδείγματα. Οι ασκήσεις επιτρέπουν στους μαθητές να γράψουν γρήγορα και εύκολα μικρά προγράμματα με την βοήθεια άμεσων ανατροφοδοτήσεων (feedbacks).

3.4.3 *MyPyTutor*

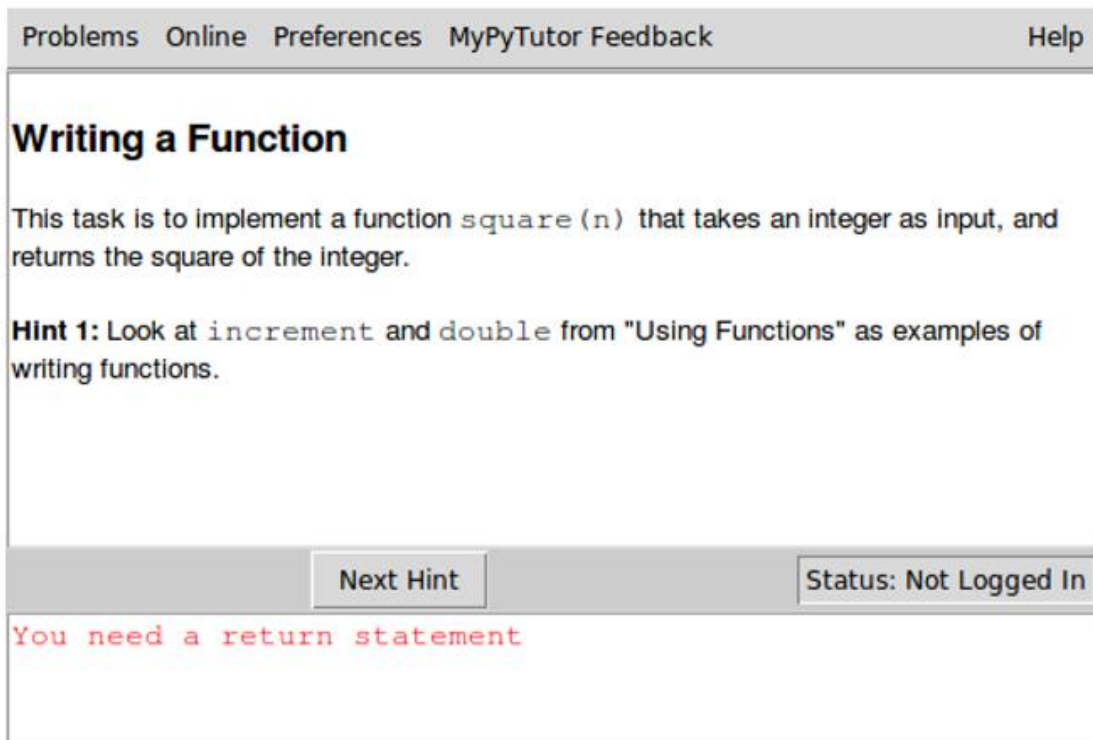
Το MyPyTutor είναι ένα διαδραστικό εκπαιδευτικό εργαλείο για την γλώσσα προγραμματισμού Python. Το εργαλείο αυτό παρέχει υποστήριξη τόσο για τους μαθητές όσο και για τους εκπαιδευτές. Το MyPyTutor μπορεί να χρησιμοποιηθεί σε δύο λειτουργίες: είτε αυτόνομο, είτε με ηλεκτρονική υποστήριξη [34]. Για μαθήματα όπου το MyPyTutor χρησιμοποιείται για αξιολόγηση, μπορεί να ρυθμιστεί έτσι ώστε οι μαθητές να υποβάλουν τις σωστές λύσεις σε online προβλήματα.

Είναι μια εφαρμογή που οι φοιτητές κατεβάζουν στον υπολογιστή τους, όπως και τη συλλογή των μαθημάτων. Αυτό σημαίνει ότι το MyPyTutor μπορεί να εκτελεστεί με αυτόνομο τρόπο εάν χρησιμοποιείται απλώς ως εργαλείο μάθησης ή αν θέλει ένας μαθητής να εργαστεί σε προβλήματα χωρίς να συνδεθεί. Εάν το εργαλείο χρησιμοποιείται για αξιολόγηση τότε μπορεί να ρυθμιστεί έτσι ώστε οι μαθητές να μπορούν να συνδεθούν και να υποβάλουν απαντήσεις σε προβλήματα.

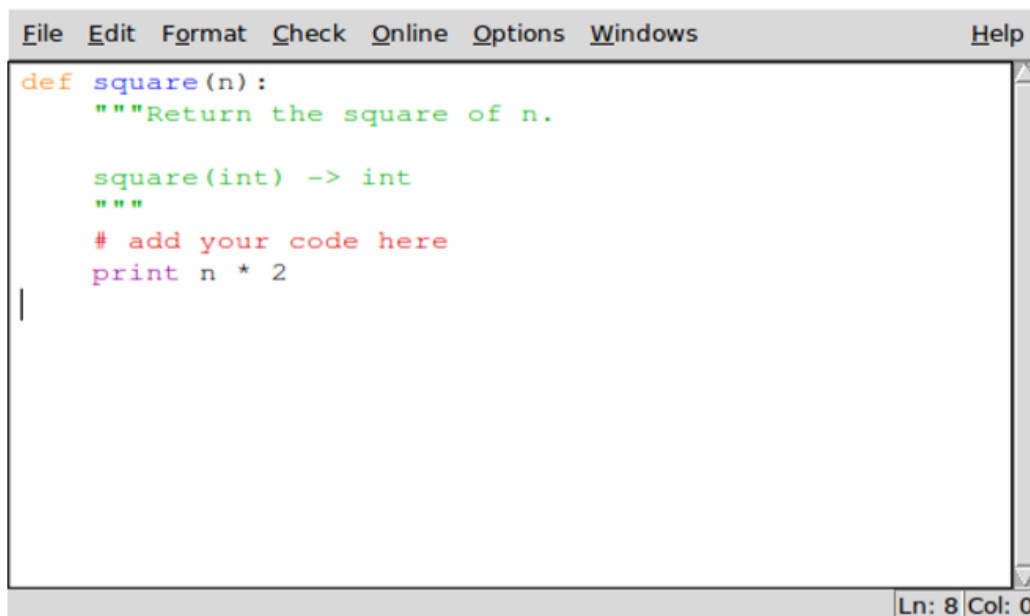
Το εργαλείο MyPyTutor αποτελείται από δύο παράθυρα. Ένα παράθυρο για την επεξεργασία κώδικα το οποίο είναι ίδιο με το παράθυρο επεξεργασίας κώδικα του εργαλείου IDLE, εκτός από κάποιες μικρές αλλαγές στο μενού. Το άλλο παράθυρο αποτελείται από δύο μικροεφαρμογές (widgets) κειμένου. Στην μια μικροεφαρμογή εμφανίζεται το πρόβλημα που έχει να επεξεργαστεί ο μαθητής, το οποίο είναι σε μορφή HTML. Η δεύτερη μικροεφαρμογή χρησιμοποιείται για την εξαγωγή των αποτελεσμάτων μετά τον έλεγχο του κώδικα των μαθητών και χρησιμοποιείται για την εμφάνιση μηνυμάτων επιτυχίας όταν ο κώδικας είναι σωστός ή αποτυχίας αν υπάρχει κάποιο σφάλμα.

Συνήθως, οι μαθητές ξεκινούν την εφαρμογή και επιλέγουν ένα πρόβλημα για να δουλέψουν. Αφού δουλέψουν στον κώδικα μπορούν να τον ελέγξουν για την ορθότητα του. Εάν ο κώδικας περάσει την δοκιμή και είναι συνδεδεμένοι οι μαθητές, μπορούν να υποβάλουν την απάντησή τους στον διακομιστή για βαθμολόγηση. Αν ο κώδικας αποτύχει στην δοκιμή μπορούν οι μαθητές να επεξεργαστούν τον κώδικά τους και να τον ελέγξουν ξανά. Όταν οι μαθητές ελέγξουν αν ο κώδικάς τους είναι σωστός το MyPyTutor θα απαντήσει με ένα μήνυμα που δηλώνει ότι ο κώδικάς τους είναι σωστός και αν έχει ρυθμιστεί η εφαρμογή για υποβολή απαντήσεων τότε θα υπενθυμίσει στους μαθητές να υποβάλουν τις απαντήσεις τους. Εάν ο κώδικας είναι λανθασμένος, το MyPyTutor αποκρίνεται εμφανίζοντας ένα μήνυμα σφάλματος, ανάλογα με το σφάλμα

που έχει συμβεί στον κώδικα. Οι Εικόνες 11 και 12 απεικονίζουν ένα παράδειγμα του MyPyTutor.



Εικόνα 11 - Παράδειγμα MyPyTutor: Περιγραφή προβλήματος και Παράθυρο Εκτέλεσης



Εικόνα 12 - Παράθυρο επεξεργασίας κώδικα

Εκτός από την πλευρά του μαθητή στο εργαλείο MyPyTutor υπάρχει και η πλευρά του προγραμματιστή και του διαχειριστή. Ο προγραμματιστής είναι αυτός που δημιουργεί τα προβλήματα των σεμιναρίων και των μαθημάτων. Ο διαχειριστής είναι αυτός που διαχειρίζεται κάθε μάθημα στο εργαλείο MyPyTutor και ειδικά όταν αυτό χρησιμοποιείται για αξιολόγηση. Ο προγραμματιστής διαχειρίζεται τρία εργαλεία: ένα εργαλείο ώστε να δημιουργήσει το πρόβλημα που θα πρέπει να λύσουν οι μαθητές, ένα εργαλείο ελέγχου των προβλημάτων και ένα εργαλείο δημιουργίας σεμιναρίων. Το εργαλείο δημιουργίας προβλήματος χρησιμοποιείται για την κατασκευή προβλημάτων για το κάθε σεμινάριο. Έχει δύο παράθυρα επεξεργασίας. Στο ένα παράθυρο υπάρχει η περιγραφή του προβλήματος που είναι γραμμένη σε γλώσσα HTML και στο άλλο παράθυρο γράφεται ο κώδικας δοκιμής. Το εργαλείο δοκιμής προβλημάτων είναι μια περιορισμένη έκδοση του MyPyTutor που ο προγραμματιστής μπορεί να χρησιμοποιήσει για να ελέγξει αν το πρόβλημα που έχει γράψει, συμπεριφέρεται σωστά από την πλευρά του μαθητή.

Αυτά τα δύο εργαλεία έχουν σχεδιαστεί για να συνεργάζονται, διευκολύνοντας την επεξεργασία και τον έλεγχο των προβλημάτων. Το εργαλείο δημιουργίας σεμιναρίων (tutorials) χρησιμοποιείται από τον προγραμματιστή για τη δημιουργία μιας συλλογής μαθημάτων που θα χρησιμοποιηθούν σε ένα σεμινάριο. Αποτελείται από ένα παράθυρο επεξεργασίας στο οποίο ο προγραμματιστής μπορεί να τοποθετήσει τα τμήματα του σεμιναρίου. Όταν ο προγραμματιστής έχει ολοκληρώσει τον σχεδιασμό του σεμιναρίου, το σεμινάριο μπορεί να εξαχθεί για χρήση από τον μαθητή.

Το εργαλείο διαχειριστή χρησιμοποιείται για τη διαχείριση των πληροφοριών των μαθητών όταν χρησιμοποιείται το MyPyTutor σε online κατάσταση λειτουργίας. Ο διαχειριστής συνήθως ξεκινά με την εγγραφή των μαθητών δηλαδή να δημιουργεί ονόματα χρήστη και κωδικούς πρόσβασης [34]. Ανά πάσα στιγμή, ο διαχειριστής μπορεί να αλλάξει τον κωδικό πρόσβασης ενός μαθητή.

3.4.4 Pythy

Το Pythy είναι ένα διαδραστικό προγραμματιστικό περιβάλλον για την Python. Χρησιμοποιώντας ένα πρόγραμμα περιήγησης (browser), οι μαθητές μπορούν να ξεκινήσουν να γράφουν κώδικα, να παρακολουθούν την εκτέλεση και να έχουν πρόσβαση σε υλικό που θα τους βοηθήσει στην εκμάθηση καθώς και σεμινάρια (tutorials). Το εργαλείο απευθύνεται σε εισαγωγικά μαθήματα προγραμματισμού για την

γλώσσα προγραμματισμού Python [39]. Ο πρωταρχικός στόχος του εργαλείου είναι να προσφέρει μια απλή διαδρομή για τους μαθητές ώστε να μάθουν να προγραμματίζουν μέσω αναθέσεων και παραδειγμάτων, να μάθουν, να εκτελούν και να εντοπίζουν σφάλματα στον κώδικα.



```
1 # This problem set involves some mathematical formulas that can't be typed
2 # easily in Python comments (without just writing them as Python code!).
3 # Make sure you read the full problem set description by clicking the link
4 # above to see those formulas.
5
6 -----
7 # Problem 1: Greet a person by printing "Hello, ", followed by their name,
8 # followed by an exclamation point. The greeting should be printed entirely
9 # on one line.
10 def greet(name):
11     print("Hello, "+ name+ "!")
12
13 -----
14 # Problem 2: Print the amount that you would tip on a bill at a restaurant.
15 # billTotal is the amount of the bill in dollars, and tipPercent is the tip
16 # percentage as an integer value (for example, a 15% tip would be 15).
17 def tip(billTotal, tipPercent):
18     print(billTotal*tipPercent/100)
19
20 -----
21 # Problem 3: Compute and print the wind-chill temperature based on the
22 # outside temperature (in degrees Fahrenheit) and the wind speed (in miles
23 # per hour) by writing a Python expression that corresponds to the formula
24 # in the problem set write-up.
25 def windchill(temperature, speed):
26     print(35.74 + (0.6215 * temperature) - (35.75 * (speed ** 0.16)) + (0.4275 * temperatu
```

In Class Week 2.1: longer name in class w2 (CS 2984) Colin Falten

There are five small problems given in the code below. Complete them, try the code yourself by clicking "Run", then when you think you have it working, check it by clicking **Check**. You may check it as many times as you want before the due date; your latest check counts!

Important Dates

This assignment is due at 09/04/2013 17:30 (2 days ago). This assignment may not be checked after 09/06/2013 23:55.

Εικόνα 13 - Το παράθυρο επεξεργασίας κώδικα του εργαλείου Pythy

Χρησιμοποιώντας το Pythy, οι μαθητές δεν χρειάζεται να εγκαταστήσουν κάποιο περιβάλλον ανάπτυξης. Το μόνο λογισμικό που θα χρειαστεί ο μαθητής είναι ένας περιηγητής ιστού (browser) και μέσα σε λίγα δευτερόλεπτα μπορούν να ξεκινήσουν να γράφουν κώδικα, να παρακολουθούν την εκτέλεση του προγράμματος και να έχουν πρόσβαση σε υλικό που θα τους βοηθήσει στην εκμάθηση της γλώσσας.

Οι εκπαιδευτές μπορούν να χρησιμοποιήσουν το εργαλείο Pythy στην τάξη για να παρουσιάσουν παραδείγματα και οι μαθητές να μπορούν να τα επεξεργαστούν και να αλληλεπιδράσουν με αυτά. Καθώς ο εκπαιδευτής γράφει και επεξεργάζεται ένα παράδειγμα στην τάξη, την ίδια στιγμή οι μαθητές που βλέπουν το ίδιο παράδειγμα στο παράθυρο του περιηγητή τους, βλέπουν ενημερώσεις σε πραγματικό χρόνο και έτσι διατηρούν το περιεχόμενο τους σε συγχρονισμό με τις αλλαγές που γίνονται. Οι μαθητές μπορούν να τροποποιήσουν και να επεξεργαστούν αυτά τα παραδείγματα, να αλλάξουν

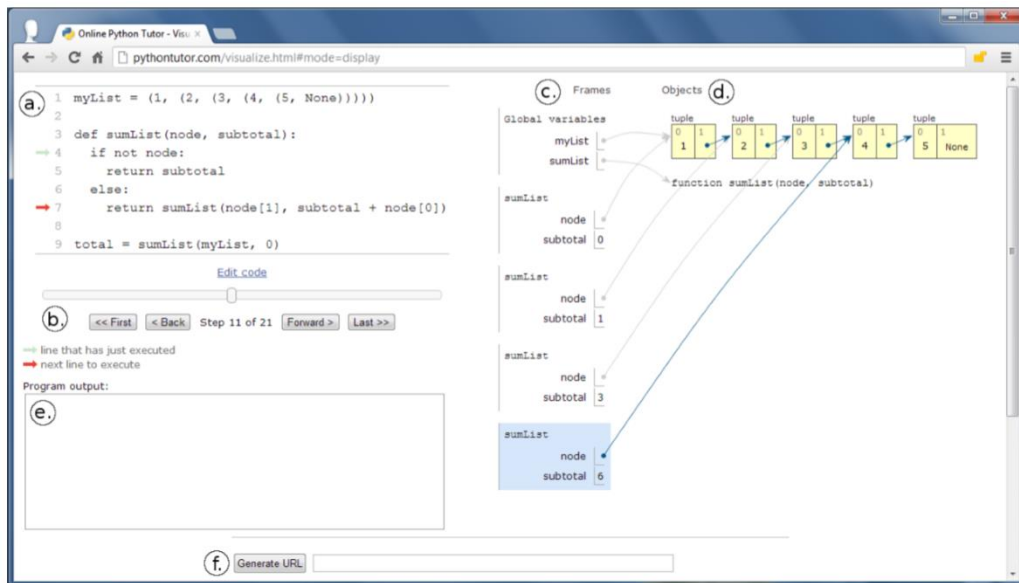
και να εκτελέσουν τον κώδικα τους. Όσο οι μαθητές αλλάζουν ή εκτελούν τον κώδικά τους, οι ενημερώσεις που γίνονται στα παραδείγματα σε πραγματικό χρόνο διακόπτονται προσωρινά.

Το εργαλείο Pythy αποθηκεύει την εργασία των μαθητών σε cloud storage, δίνοντας τους έτσι πρόσβαση στον κώδικά τους από οπουδήποτε. Η εργασία αποθηκεύεται αυτόματα κάθε φορά που ο μαθητής διακόπτει την επεξεργασία, οπότε η εργασία του δεν χάνεται ποτέ. Κάθε φορά που ένας μαθητής επιστρέφει σε μια σελίδα κώδικα που έχει γράψει, φαίνεται έτσι όπως τον άφησε την τελευταία φορά.

3.4.5 Online Python Tutor

Είναι ένα διαδικτυακό εργαλείο που χρησιμοποιείται για την εκμάθηση της γλώσσας προγραμματισμού Python. Το εργαλείο χρησιμοποιείται από μαθητές και από εκπαιδευτές και έχουν την δυνατότητα να γράψουν προγράμματα σε Python, απευθείας στο πρόγραμμα περιήγησης (web browser). Το Online Python Tutor είναι ένα δωρεάν εργαλείο και λογισμικό ανοιχτού κώδικα και είναι διαθέσιμο στο <http://pythontutor.com/>.

Η ευκολία πρόσβασης καθιστά το Online Python Tutor ένα πολύ δημοφιλές εργαλείο οπτικοποίησης και εκτέλεσης προγραμμάτων στην εκπαίδευση. Η χρήση του Online Python Tutor είναι τόσο απλή και εύκολη όσο η επίσκεψη σε έναν κανονικό ιστότοπο [35]. Η διεπαφή (GUI) του εργαλείου περιέχει δύο κύρια μέρη που σχετίζονται με δύο φάσεις: φάση επεξεργασίας και φάση οπτικοποίησης όπως φαίνεται παρακάτω (Εικόνα 14).



Εικόνα 14- Η διεπαφή του εργαλείου OnlinePythonTutor

Σε αυτό το στιγμιότυπο, ο χρήστης έχει ένα πρόγραμμα σε Python για τον υπολογισμό του αθροίσματος μιας συνδεδεμένης λίστας. Πάνω αριστερά στην ενότητα a) εμφανίζεται ο πηγαίος κώδικας του προγράμματος. Ένα κόκκινο βέλος στο αριστερό περιθώριο δείχνει την επόμενη γραμμή που πρέπει να εκτελεστεί. Το πράσινο ανοιχτό βέλος δείχνει την γραμμή του κώδικα που μόλις εκτελέστηκε και βοηθά τους χρήστες να παρακολουθούν την ροή ελέγχου του προγράμματός τους. Στην ενότητα b) απεικονίζεται μια μπάρα κύλισης και κουμπιά πλοήγησης και ένα κείμενο (step 11 of 21) το οποίο υποδεικνύει το τρέχον σημείο εκτέλεσης. Κάθε σημείο αντιπροσωπεύει μια μόνο γραμμή του κώδικα που έχει εκτελεσθεί. Ο χρήστης μπορεί να κάνει κλικ ή να σύρει το ποντίκι πάνω από την μπάρα κύλισης για να μεταβεί στο σημείο του κώδικα που θέλει.

Στην ενότητα c) εμφανίζονται πλαίσια (frames) με τις καθολικές (global) μεταβλητές και πλαίσια (frames) που εμφανίζουν το όνομα της συνάρτησης, στο παράδειγμα αυτό, η συνάρτηση sumList και τις τοπικές μεταβλητές. Το τρέχον ενεργό πλαίσιο καθώς και τα βέλη που προέρχονται από αυτό απεικονίζονται με μπλε χρώμα. Στην ενότητα d) εμφανίζονται οπτικές αναπαραστάσεις των αντικειμένων Python και τις αναφορές δεικτών προς αυτά. Το συγκεκριμένο παράδειγμα δείχνει μια συνδεδεμένη λίστα με ακέραιους αριθμούς που δημιουργήθηκαν από μια αλυσίδα πέντε πλειάδων. Η ενότητα e) εμφανίζει την έξοδο του προγράμματος και τέλος η ενότητα f) όπου εμφανίζεται το κουμπί «Generate URL» όπου δημιουργείται μια διεύθυνση URL που προσδιορίζει το τρέχον σημείο εκτέλεσης του προγράμματος. Ο χρήστης μπορεί να στείλει αυτή τη διεύθυνση URL σε ένα email ή να το δημοσιεύσει σε κάποιο φόρουμ. Όταν κάποιος κάνει κλικ σε αυτή την διεύθυνση URL, μεταφέρεται στο ακριβές σημείο εκτέλεσης που έχει καθορίσει ο αποστολέας.

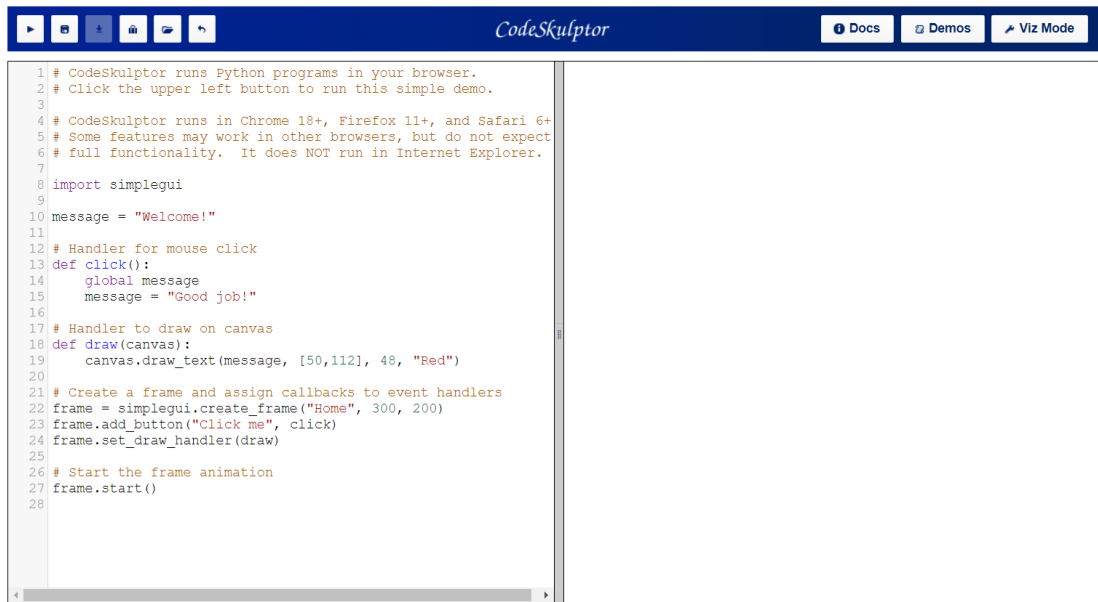
3.4.6 CodeSkulptor

Το CodeSkulptor είναι ένα διαδραστικό περιβάλλον προγραμματισμού σε γλώσσα Python. Το περιβάλλον του CodeSkulptor αναπτύχθηκε από τον Scott Rixner, καθηγητή του πανεπιστημίου Rice και είναι διαθέσιμο σε όλους τους χρήστες που έχουν πρόσβαση στο διαδίκτυο μέσω της ιστοσελίδας <http://www.codeskulptor.org/>

Κάποιες από τις πιο σημαντικές λειτουργίες του CodeSkulptor περιλαμβάνουν:

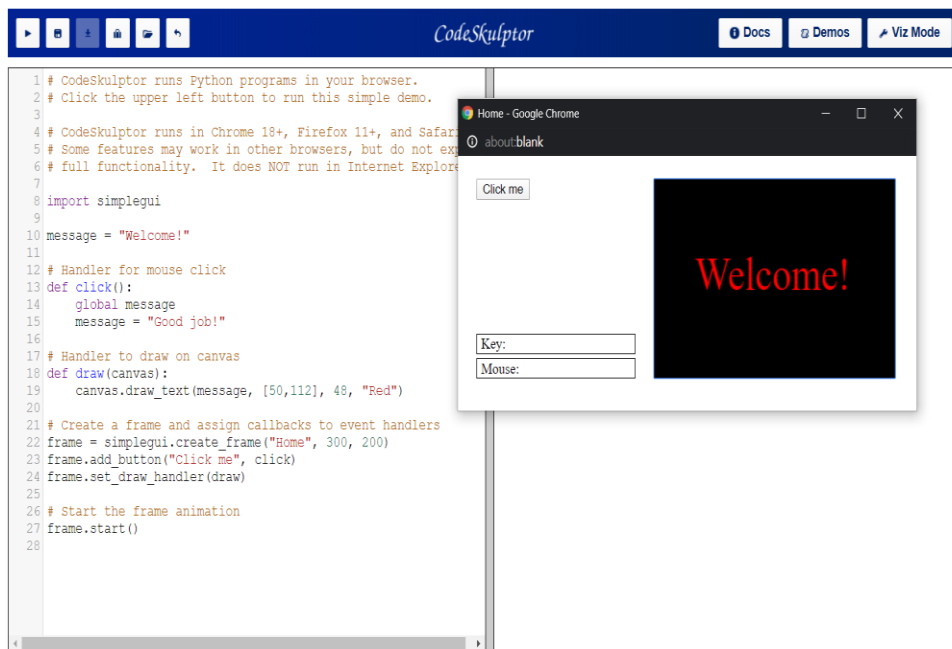
- a) Σύνταξη προγραμμάτων διαδραστικού χαρακτήρα.
- b) Επεξεργασία και αποθήκευση προγραμμάτων.
- c) Εμφάνιση μηνυμάτων «λάθους» για τον εντοπισμό σφαλμάτων.

d) Οπτική απεικόνιση της λειτουργίας του προγράμματος (Visualization mode) [42].



Εικόνα 15 - Η διεπαφή του εργαλείου CodeSkulptor

Η Εικόνα 16 απεικονίζει το CodeSkulptor όπου εκτελείται ένα απλό διαδραστικό πρόγραμμα το οποίο δημιουργεί ένα παράθυρο που εμφανίζει το μήνυμα «Welcome!». Επίσης εμφανίζεται και ένα κουμπί «Click me», και όταν το πατήσει ο χρήστης το μήνυμα αλλάζει σε «Good job!».



Εικόνα 16 - Εμφάνιση μηνύματος

Ο κώδικας Python για αυτό το πρόγραμμα εμφανίζεται στο πλαίσιο αριστερά, ενώ στο πλαίσιο δεξιά εμφανίζεται η έξοδος της εκτέλεσης του προγράμματος. Ο σχεδιασμός του εργαλείου CodeSkulptor αξιοποιεί πολλές υπάρχουσες ιδέες και ενσωματώνει πολλά νέα χαρακτηριστικά. Η δημιουργία του εργαλείου αυτού αποτελεί συγχώνευση των εργαλείων CodeMirror3 και Skulpt4. Τα προγράμματα στο Codeskulptor αποθηκεύονται σε ένα απομακρυσμένο κέντρο δεδομένων και όχι στον υπολογιστή. Η διαχείριση των προγραμμάτων καθίσταται πιο εύκολη με την αποθήκευση τους σε σελιδοδείκτες του browser .

4 Σύγκριση Περιβαλλόντων Προγραμματισμού

Στο προηγούμενο κεφάλαιο παρουσιάστηκαν εκπαιδευτικά εργαλεία προγραμματισμού και προγραμματιστικά περιβάλλοντα που χρησιμοποιούνται στην εκμάθηση της γλώσσας προγραμματισμού Python, τα οποία διανέμονται ελεύθερα και ικανοποιούν τις ιδιαίτερες απαιτήσεις των αρχάριων προγραμματιστών.

4.1 Βασικά γνωρίσματα περιβαλλόντων προγραμματισμού

Στην ενότητα αυτή θα συγκρίνουμε τα εκπαιδευτικά προγραμματιστικά περιβάλλοντα με βάση τις βασικές κατηγορίες γνωρισμάτων, οι οποίες θα πρέπει να λαμβάνονται υπόψη κατά τη συγκριτική ανάλυση και είναι οι εξής: γραφικό ενδιάμεσο, συντάκτης, αποσφαλματωτής, μηνύματα λάθους, οπτικοποίηση, μεμονωμένη εκτέλεση μεθόδων και αξιολόγηση από σπουδαστές [30]. Παρακάτω παρουσιάζονται οι ομοιότητες και οι διαφορές αυτών των περιβαλλόντων.

Γραφικό ενδιάμεσο (GUI): Από τα 9 περιβάλλοντα προγραμματισμού που έχουν αναλυθεί στο προηγούμενο κεφάλαιο, το PILEt, το MyPyTutor και το OnlinePythonTutor είναι αυτά τα εκπαιδευτικά προγραμματιστικά περιβάλλοντα, όπου μέσα από την διεπαφή τους αναδεικνύουν περισσότερο τη γλώσσα προγραμματισμού Python καθώς και διαφορετικές τεχνικές μάθησης της γλώσσας, χρησιμοποιώντας παραδείγματα κώδικα, ασκήσεις, παζλ και εκπαιδευτικό υλικό. Το PILEt και το OnlinePythonTutor χρησιμοποιούν στη διεπαφή τους διαγράμματα ροής, οπτικές αναπαραστάσεις των αντικειμένων καθώς και πλαίσια (frames), παρέχοντας έτσι στους μαθητές τη δυνατότητα άμεσης παρατήρησης της συμπεριφοράς μεμονωμένων μεθόδων και μεταβλητών. Τέλος, τα δύο αυτά εργαλεία, χρησιμοποιούν επιπλέον ένα γραφικό ενδιάμεσο όπου μέσα από αυτό οι μαθητές αξιολογούν τις γνώσεις τους μετά από κάθε συνεδρία. Η διεπαφή των περιβαλλόντων Thonny, Jypr, Pythy και CodeSkulptor παρέχει έναν επεξεργαστή κώδικα καθώς και σημαντικά εργαλεία για την απεικόνιση της εκτέλεσης του κώδικα «βήμα-βήμα» καθώς παραδείγματα και εκπαιδευτικό υλικό που βοηθούν τους αρχάριους προγραμματιστές. Το γραφικό ενδιάμεσο του IDLE είναι πάρα πολύ απλό χωρίς να παρουσιάζονται με οπτικό τρόπο οι μεταβλητές και οι μέθοδοι του

κώδικα που γράφουν οι μαθητές. Τέλος, το Spyder έχει ένα επιστημονικό γραφικό ενδιάμεσο, διότι χρησιμοποιείται περισσότερο από το επιστημονικό κοινό.

Παρακάτω στον Πίνακα 1 απεικονίζονται συγκεντρωτικά τα χαρακτηριστικά της διεπαφής (GUI) κάθε εκπαιδευτικού προγραμματιστικού περιβάλλοντος.

Πίνακας 1 - Χαρακτηριστικά διεπαφής (GUI)

| | Παραδείγματα κώδικα/ ασκήσεις | Οπτικές αναπαραστάσεις/παζλ | Επεξεργαστής κώδικα | Εργαλεία απεικόνισης της εκτέλεσης κώδικα |
|-------------------|-------------------------------|-----------------------------|---------------------|-------------------------------------------|
| IDLE | | | ✓ | |
| Thonny | | | ✓ | ✓ |
| Spyder | | | ✓ | ✓ |
| Jype | | | ✓ | ✓ |
| Pythy | ✓ | | ✓ | ✓ |
| PILet | ✓ | ✓ | ✓ | ✓ |
| MyPyTutor | ✓ | ✓ | ✓ | ✓ |
| OnlinePythonTutor | ✓ | ✓ | ✓ | |
| CodeSkulptor | ✓ | | ✓ | ✓ |

Συντάκτης (Editor): Όλα τα εκπαιδευτικά προγραμματιστικά περιβάλλοντα διαθέτουν τις βασικές λειτουργίες ενός συντάκτη κώδικα. Επίσης, όλα τα περιβάλλοντα υποστηρίζουν την χρωματική επισήμανση των στοιχείων της γλώσσας. Το IDLE και το Thonny δεν εμφανίζουν αρίθμηση των γραμμών κώδικα. Δυνατότητα για αυτόματη συμπλήρωση κώδικα κατά την ανάπτυξη προγραμμάτων διαθέτουν τα IDLE και Spyder. Παρακάτω στον Πίνακα 2 απεικονίζονται συγκεντρωτικά τα χαρακτηριστικά του συντάκτη (editor) κάθε εκπαιδευτικού προγραμματιστικού περιβάλλοντος.

Πίνακας 2 - Χαρακτηριστικά συντάκτη

| | Επισημανση σύνταξης | Αναδίπλωση κώδικα | Εντολές σύνταξης και εκτέλεσης | Αρίθμηση γραμμών κώδικα |
|-------------------|------------------------|----------------------|--------------------------------------|----------------------------|
| IDLE | ✓ | | ✓ | ✓ |
| Thonny | ✓ | | ✓ | ✓ |
| Spyder | ✓ | ✓ | ✓ | ✓ |
| Jype | ✓ | | ✓ | ✓ |
| Pythy | ✓ | | ✓ | ✓ |
| PILet | ✓ | | ✓ | ✓ |
| MyPyTutor | ✓ | | ✓ | |
| OnlinePythonTutor | ✓ | ✓ | ✓ | ✓ |
| CodeSkulptor | ✓ | ✓ | ✓ | ✓ |

Μεταγλώττιση – μηνύματα λάθους: Όλα τα περιβάλλοντα επιτρέπουν την αλληλεπίδραση μεταξύ του μηνύματος λάθους και της αντίστοιχης γραμμής κώδικα. Το IDLE σε περίπτωση συντακτικού λάθους, εμφανίζει μηνύματα σχετικά με το είδος και το σημείο του σφάλματος, προκειμένου να διευκολυνθούν οι αρχάριοι προγραμματιστές στην διόρθωση του. Το περιβάλλον PILEt χρησιμοποιεί παραδείγματα από εκτελέσιμο κώδικα, που μπορεί ο αρχάριος προγραμματιστής να τα επεξεργαστεί και να τα μεταγλωττίσει. Τέλος τα περιβάλλοντα MyPyTutor και Pythy υποστηρίζουν τον προγραμματιστή στον εντοπισμό του λάθους προφέροντας βοήθεια με πληροφορίες σχετικές με τις πιθανές αιτίες που προκάλεσαν το συντακτικό λάθος.

Πίνακας 3 - Χαρακτηριστικά μεταγλώττισης

| | Αντιστοίχιση γραμμής κώδικα με το μήνυμα λάθους | Εμφάνιση είδους σφάλματος | Εμφάνιση πληροφοριών με τις αιτίες πρόκλησης του σφάλματος |
|--------|-------------------------------------------------------|------------------------------|---------------------------------------------------------------------|
| IDLE | ✓ | ✓ | |
| Thonny | ✓ | | |
| Spyder | ✓ | | |
| Jype | ✓ | | |

| | | | |
|-------------------|---|---|---|
| Pythy | ✓ | | ✓ |
| PILet | ✓ | | |
| MyPyTutor | ✓ | | ✓ |
| OnlinePythonTutor | ✓ | ✓ | |
| CodeSkulptor | ✓ | | |

Αποσφαλματοτής (debugger): Όλα τα περιβάλλοντα διαθέτουν ενσωματωμένο αποσφαλματοτή επιτρέποντας στους αρχάριους προγραμματιστές να εντοπίζουν σφάλματα στον κώδικα τους, εξετάζοντας «βήμα – βήμα» την κάθε γραμμή του κώδικα και να παρατηρούν τις μεταβλητές που αλλάζουν τιμές. Τα εργαλεία Spyder και Codeskulptor επιτρέπουν την προβολή και τον έλεγχο σημείων διακοπής (breakpoints) και της ροής εκτέλεσης απευθείας μέσα από το γραφικό ενδιάμεσο των περιβαλλόντων.

Οπτικοποίηση (Visualization): Τα περιβάλλοντα Thonny και Jupyter εμφανίζουν στην διεπαφή τους πληροφορίες για την κάθε μια εντολή του κώδικα που εκτελείται ξεχωριστά. Το Spyder διαθέτει διαδραστικές επιλογές οπτικοποίησης δεδομένων (γραφική αναπαράσταση, ιστόγραμμα, εικόνες), χρησιμοποιώντας την βιβλιοθήκη Matplotlib. Το περιβάλλον PILEt, διαθέτει εικονική επεξήγηση του μαθήματος στον αρχάριο προγραμματιστή, μέσω βίντεο, στατικών πινάκων, διαγραμμάτων ροής και εικόνων που περιγράφουν τις διάφορες καταστάσεις της δραστηριότητας ενός προγράμματος. Το περιβάλλον OnlinePythonTutor διαθέτει και αυτό εργαλείο οπτικοποίησης εκτέλεσης των προγραμμάτων, εμφανίζοντας γραφικές αναπαραστάσεις των αντικειμένων κατά την εκτέλεση του κώδικα. Τέλος, το Codeskulptor, χρησιμοποιεί το εργαλείο Viz που υποστηρίζει την ίδια βασική λειτουργικότητα οπτικοποίησης που υπάρχει στο περιβάλλον OnlinePythonTutor. Το εργαλείο αυτό οπτικοποιεί την εκτέλεση προγραμμάτων Python που έχουν μέγεθος εκατοντάδων γραμμών, ο αρχάριος προγραμματιστής μπορεί να ορίσει σημεία διακοπής στον κώδικά του και να μεταβεί είτε προς το εμπρός, είτε προς τα πίσω και μπορεί να ελέγξει τη σειρά και τον ρυθμό με τον οποίο ενεργοποιούνται τα γεγονότα στον κώδικα.

Πίνακας 4 - Χαρακτηριστικά οπτικοποίησης

| | Εμφάνιση πληροφοριών για κάθε μια εντολή | Γραφική αναπαράσταση/ ιστόγραμμα | Βίντεο / Εικόνες | Διαγράμματα ροής |
|-------------------|------------------------------------------|----------------------------------|------------------|------------------|
| IDLE | | | | |
| Thonny | ✓ | | | |
| Spyder | | ✓ | | ✓ |
| Jype | ✓ | | | |
| Pythy | ✓ | | | |
| PILet | | | ✓ | ✓ |
| MyPyTutor | | | | |
| OnlinePythonTutor | | ✓ | ✓ | ✓ |
| CodeSkulptor | | ✓ | ✓ | ✓ |

Μεμονωμένη εκτέλεση μεθόδων – δημιουργία αντικειμένων: Όλα τα περιβάλλοντα επιτρέπουν την μεμονωμένη εκτέλεση μεθόδων με την βοήθεια του γραφικού τους ενδιάμεσου. Το περιβάλλον Thonny, κατά την εκτέλεση μιας μεθόδου, ανοίγει ξεχωριστό παράθυρο στην διεπαφή του που εμφανίζει τον πίνακα με τις τοπικές μεταβλητές της μεθόδου. Το περιβάλλον OnlinePythonTutor, αναπαριστά τις μεθόδους και τα αντικείμενα με γραφικό τρόπο (frames).

4.2 Μελέτες Αξιολόγησης Περιβαλλόντων

Thonny: Το προγραμματιστικό περιβάλλον Thonny, χρησιμοποιήθηκε για ένα εξάμηνο στο εισαγωγικό μάθημα προγραμματισμού στο πανεπιστήμιο του Γάρτου. Το μάθημα είχε 280 συμμετέχοντες. Σε κάποιες διαλέξεις χρησιμοποιήθηκε ο τρόπος εκτέλεσης του κώδικα «βήμα-βήμα» που παρέχει το περιβάλλον Thonny για την εμφάνιση των μεταβλητών, τη ροή ελέγχου του προγράμματος, τις κλήσεις των μεθόδων και την αναδρομή. Οι αξιολογήσεις στο τέλος αυτών των διαλέξεων αποκάλυψαν ότι η πλειοψηφία των φοιτητών (άνω του 70%), πιστεύει ότι το προγραμματιστικό περιβάλλον Thonny, τους βοήθησε να καταλάβουν αυτές τις αρχές. Τουλάχιστον 70 φοιτητές εγκατέστησαν το περιβάλλον Thonny στους υπολογιστές

τους και προσπάθησαν να το χρησιμοποιήσουν [1]. Δεν προσφέρθηκε κάποιο υλικό υποστήριξης για τη χρήση των λειτουργιών του προγράμματος. Οι διαλέξεις ήταν αρκετές για να δώσουν στους ενδιαφερόμενους φοιτητές τη δυνατότητα να εκτελέσουν τα προγράμματα τους με το Thonny. Λήφθηκαν αρχεία καταγραφής από τις ασκήσεις προγραμματισμού και απαντήσεις από 44 φοιτητές. Οι φοιτητές ανέφεραν ότι τους άρεσε ο Thonny διότι ο επεξεργαστής κώδικα (code editor) και το κέλυφος (shell) βρίσκονται στο ίδιο παράθυρο.

OnlinePythonTutor: Καθηγητές και διδάσκοντες σε δώδεκα πανεπιστήμια έχουν χρησιμοποιήσει το εργαλείο OnlinePythonTutor στη διδασκαλία τους. Στο εισαγωγικό μάθημα προγραμματισμού στο πανεπιστήμιο της Ουάσιγκτον, χρησιμοποίησαν το συγκεκριμένο εργαλείο πάνω από 900 φοιτητές. Αρκετοί εκπαιδευτές αναφέρανε ότι το εργαλείο αυτό τους βοήθησε να διευκρινίσουν στους φοιτητές τους, έννοιες όπως ανίχνευση των παραμέτρων και τις επιστρεφόμενες τιμές από τις κλήσεις των μεθόδων [35]. Παρόλο, που δεν υπάρχουν λεπτομερείς δημογραφικές πληροφορίες για τους χρήστες, υπάρχει τουλάχιστον μια σημαντική πηγή χρηστών: οι μαθητές που λαμβάνουν δωρεάν διαδικτυακά μαθήματα εισαγωγής προγραμματισμού που προσφέρονται από το Udacity [45], όπως το μάθημα «Εισαγωγή στην Επιστήμη Υπολογιστών» που το έχουν παρακολουθήσει 186000 μαθητές συνολικά και το μάθημα «Σχεδιασμός προγραμμάτων υπολογιστών» με σχεδόν 40000 μαθητές. Μερικοί μαθητές βρήκαν το εργαλείο ιδιαίτερα χρήσιμο κατά τον εντοπισμό σφαλμάτων καθώς και στο να κατανοήσουν τις έννοιες των δομών όπως οι λίστες και οι πίνακες.

Pythy: Το εκπαιδευτικό εργαλείο προγραμματισμού Pythy χρησιμοποιήθηκε για πρώτη φορά στο μάθημα προγραμματισμού «Υπολογιστική επίλυση προβλημάτων στην Python» που διδάχθηκε τον Ιανουάριο του 2013 στο Πολυτεχνικό Ινστιτούτο της Βιρτζίνια [39]. Στο συγκεκριμένο μάθημα χρησιμοποιήθηκε το βιβλίο του Miller [5] ως πρωτεύον εκπαιδευτικό υλικό. Το μάθημα παρακολούθησαν 53 φοιτητές και χρησιμοποίησαν το εργαλείο Pythy για όλες τις ασκήσεις και τις εργασίες προγραμματισμού. Ελήφθησαν συνολικά 35 απαντήσεις από την έρευνα που διεξήχθη όπου οι 19 (36%) χρησιμοποίησαν το εργαλείο Pythy. Τα αποτελέσματα της έρευνας έδειξαν ότι οι φοιτητές που χρησιμοποίησαν το εργαλείο Pythy, ότι ήταν πολύ εύκολη

η χρήση του, ότι παρέχει μεγάλη υποστήριξη στις εργασίες των φοιτητών και ότι τους διευκόλυνε με την χρήση των παραδειγμάτων να καταλάβουν τους λόγους που τα προγράμματα τους δεν δούλεψαν σωστά.

4.3 Συμπεράσματα

Από την παραπάνω παρουσίαση και συγκριτική ανάλυση των χαρακτηριστικών των εκπαιδευτικών προγραμματιστικών περιβαλλόντων φαίνεται ότι από τα 9 εκπαιδευτικά περιβάλλοντα, το Thonny, το CodeSkulptor και το OnlinePythonTutor, ενσωματώνουν δυνατότητες που βοηθούν τον αρχάριο προγραμματιστή στην εκμάθηση της γλώσσας προγραμματισμού Python μέσα από διάφορες διδακτικές προσεγγίσεις. Τα περιβάλλοντα αυτά χρησιμοποιούν διεπαφές που βοηθάνε τους αρχάριους προγραμματιστές να συντάξουν τον κώδικα τους και αναδεικνύουν περισσότερο τη γλώσσα προγραμματισμού Python καθώς και διαφορετικές τεχνικές εκμάθησης της γλώσσας, χρησιμοποιώντας παραδείγματα κώδικα, ασκήσεις, παζλ και εκπαιδευτικό υλικό. Τα περιβάλλοντα αυτά επιτρέπουν την αλληλεπίδραση μεταξύ του μηνύματος λάθους κατά την εκτέλεση των προγραμμάτων. Τέλος, με τη λειτουργία της οπτικοποίησης που παρέχουν τα περιβάλλοντα αυτά, βοηθούν τους αρχάριους προγραμματιστές να μάθουν τη γλώσσα και να διαχειρίζονται τα τυχόν λάθη τους μέσα από οπτικές αναπαραστάσεις.

5 Εισαγωγή στον Προγραμματισμό με την γλώσσα Python

Όλο και περισσότερο η γλώσσα προγραμματισμού Python με το διαδραστικό της περιβάλλον χρησιμοποιείται σε εισαγωγικά μαθήματα προγραμματισμού σε πανεπιστημιακό επίπεδο. Είναι σχετικά απλή στην σύνταξη της [31, 23], και επιτρέπει περισσότερη εστίαση στην επίλυση προβλημάτων στα πρώτα μαθήματα προγραμματισμού. Υπάρχουν αρκετές ενδείξεις ότι η διδασκαλία της Python στο πρώτο μάθημα είναι τουλάχιστον τόσο καλή όσο η διδασκαλία οποιασδήποτε άλλης γλώσσας προγραμματισμού υψηλού επιπέδου. Συγκεκριμένα, μια μελέτη [38] διαπίστωσε ότι οι φοιτητές που είχαν πρώτο μάθημα με την γλώσσα προγραμματισμού Python και αργότερα πήγαν σε ένα μάθημα με γλώσσα προγραμματισμού την C++, τα πήγαν το ίδιο καλά, με τους φοιτητές που είχαν στο πρώτο τους μάθημα προγραμματισμού την C++.

Η χρήση της γλώσσας Python σε συνδυασμό με προγράμματα καθοδήγησης και άλλες διδακτικές προσεγγίσεις, μπορεί να είναι χρήσιμη για τη βελτίωση του τρόπου που οι μαθητές μαθαίνουν και κατανοούν τα μαθήματα προγραμματισμού. Πρόσφατα, οι ερευνητές ενδιαφέρθηκαν ιδιαίτερα για την ανάπτυξη διαδικτυακών περιβαλλόντων [12] για τον προγραμματισμό κώδικα Python, καθώς αυτά καταργούν τυχόν εμπόδια που σχετίζονται με το λογισμικό στους αρχάριους προγραμματιστές.

Η Python επιτρέπει σε έναν αρχάριο μαθητή να προχωρά γρήγορα σε ενδιαφέροντα έργα. Περιέχει ενσωματωμένα χαρακτηριστικά που καθιστούν τη γλώσσα εξαιρετικά χρήσιμη και για αρχάριους προγραμματιστές. Περιέχει πακέτα γραφιστικά [46] που επιτρέπουν στους αρχάριους μαθητές να γράφουν προγράμματα με γραφικά στοιχεία ακόμη και στο πρώτο μάθημα. Επειδή η Python είναι ελεύθερο λογισμικό, δεν επιβαρύνει με κάποιο κόστος τους μαθητές.

Η Python έχει μια ενεργή κοινότητα προγραμματιστών που ασχολούνται με τον προγραμματισμό ανοιχτού κώδικα (open-source) και παρέχουν τόσο ζωντανές συζητήσεις πάνω σε θέματα προγραμματισμού της Python όσο και χρήσιμες συμβουλές [19]. Με αυτό τον τρόπο οι αρχάριοι προγραμματιστές αντιλαμβάνονται ότι είναι και αυτοί μέρος μιας ευρύτερης κοινότητας προγραμματιστών.

Τέλος, η Python παρέχει ένα διαδραστικό κέλυφος (shell), το οποίο ενθαρρύνει τους μαθητές να πειραματιστούν τη στιγμή που γράφουν κώδικα. Μαθαίνουν την βασική ικανότητα χρήσης εργαλείων προγραμματισμού για να διερευνήσουν τον τρόπο με τον οποίο γίνεται η ανάπτυξη εφαρμογών.

5.1 Οφέλη της Python στη διδασκαλία βασικών εννοιών προγραμματισμού

Ένα χαρακτηριστικό της Python, είναι ότι δεν είναι καθαρά εκπαιδευτική γλώσσα, ούτε καθαρά εμπορική γλώσσα προγραμματισμού [3]. Σχεδιάστηκε αρχικά για την εκπαίδευση, αλλά σύντομα κέρδισε τη δημοτικότητα των προγραμματιστών που έγιναν και η κινητήρια δύναμη της στην εξέλιξη της.

Το μεγαλύτερο μέρος της σύνταξης των εντολών στην γλώσσα Python είναι αρκετά εύκολο για τους αρχάριους προγραμματιστές. Οι εντολές ελέγχου if, οι εντολές επανάληψης και ούτω καθεξής που μπορεί να μάθει κάποιος με την Python, είναι χρήσιμες και βασικές για τις επόμενες γλώσσες προγραμματισμού. Μια σημαντική ιδιαιτερότητα που έχει η γλώσσα Python κατά την σύνταξη κώδικα είναι η χρήση εσοχών για την ομαδοποίηση εντολών.

Η Python διαθέτει εξαιρετικά εργαλεία επεξεργασίας των συμβολοσειρών (strings) και των κανονικών εκφράσεων (regular expression). Οι μαθητές μπορούν να ξεκινήσουν νωρίς τον προγραμματισμό συμβολοσειρών, αυξάνοντας με αυτό τον τρόπο, το εύρος των προβλημάτων που μπορούν να τους ανατεθούν και να τα υλοποιήσουν. Υποστηρίζει επίσης τις θεμελιώδεις δομές δεδομένων. Οι λίστες (lists) στην Python, είναι συντακτικά παρόμοιες με τους πίνακες (arrays) στις περισσότερες γλώσσες προγραμματισμού, αλλά είναι πλήρως ετερογενείς [56]. Η εντολή της Python for loop, παρέχει έναν σαφή μηχανισμό επανάληψης σε λίστες. Επίσης παρέχει έναν ενσωματωμένο τύπο λεξικού (dictionary). Οι αρχάριοι προγραμματιστές επωφελούνται από την πρώιμη εξοικείωση τους μέσα από αυτές τις δομές δεδομένων.

Η Python υποστηρίζει μεθόδους (functions) και πρόσθετα (modules). Ένα πρόσθετο στην Python είναι ένα αρχείο κώδικα με μεθόδους. Η κοινότητα των προγραμματιστών της Python επεκτείνει συνεχώς την γλώσσα χρησιμοποιώντας πρόσθετα. Αυτή η διαδικασία προσφέρει αρκετά οφέλη στους αρχάριους προγραμματιστές [23]. Τους δίνεται η δυνατότητα να αρχίζουν αμέσως να οργανώνουν και να κωδικοποιούν τις δικές τους λύσεις σε προβλήματα.

5.2 Δυσκολίες που αντιμετωπίζουν οι σπουδαστές

Η Python έχει γίνει μια δημοφιλής γλώσσα προγραμματισμού για την παράδοση εισαγωγικών μαθημάτων προγραμματισμού. Δύο λόγοι για αυτό, είναι η ευκολία και η συντακτική απλότητα της Python, δίνοντας έτσι στους αρχάριους προγραμματιστές τη δυνατότητα επίλυσης σύνθετων προβλημάτων με μικρά αποσπάσματα κώδικα. Ωστόσο, οι μαθητές παρουσιάζουν εκτεταμένες παρανοήσεις σχετικά με τις έννοιες των βασικών δομών της γλώσσας, με αποτέλεσμα να αντιμετωπίζουν προβλήματα στην κατανόηση θεμελιωδών εννοιών [11]. Η πολυπλοκότητα των αλγορίθμων και των δομών δεδομένων, οδηγούν συχνά τους μαθητές να μην κατανοούν ούτε τις βασικές αρχές του προγραμματισμού, όπως είναι η ανάθεση τιμής σε μια μεταβλητή, η δομή ελέγχου και η δομή επανάληψης.

Το ακαδημαϊκό έτος 2000-2001 το κολέγιο “Centre College” άρχισε να χρησιμοποιεί στο εισαγωγικό μάθημα προγραμματισμού την γλώσσα προγραμματισμού Python. Μετά από έρευνα που έγινε σε 86 σπουδαστές που παρακολούθησαν το εισαγωγικό μάθημα προγραμματισμού, προέκυψαν κάποια ζητήματα που αντιμετώπισαν οι σπουδαστές κατά την διδασκαλία του μαθήματος [19]. Η έλλειψη της δήλωσης του τύπου μια μεταβλητής στην Python, μπορεί να απλοποιεί πολύ την σύνταξη της, αλλά δημιουργεί και αρκετά προβλήματα στους αρχάριους προγραμματιστές όταν έρθουν σε επαφή με μια άλλη γλώσσα προγραμματισμού όπως η Java, όπου εκεί η δήλωση του τύπου της μεταβλητής είναι απαραίτητη και υποχρεωτική. Κατά την εκτέλεση ενός κώδικα, η Python δεν γνωρίζει τον τύπο της μεταβλητής έως αυτή να εκτελεστεί. Αυτό που συμβαίνει είναι, να αποθηκεύει αυτή την τιμή σε κάποια θέση μνήμης και στη συνέχεια, δεσμεύει το όνομα της μεταβλητής σε αυτό τον χώρο μνήμης. Με αυτό τον τρόπο, καθιστά τα δεδομένα που βρίσκονται στην μνήμη να είναι προσβάσιμα μέσω του ονόματος της μεταβλητής. Οι μαθητές μπορούν να μάθουν, να δημιουργούν και να χρησιμοποιούν μεταβλητές, αλλά δεν μαθαίνουν πολλά για τον τύπο της μεταβλητής. Θα πρέπει να είναι πολύ προσεκτικοί στην δήλωση των μεταβλητών, διότι μπορούν να προκαλέσουν συντακτικά λάθη στον κώδικά τους και να μην λειτουργεί σωστά. Για παράδειγμα, αν ένας μαθητής δηλώσει στον κώδικά του την παρακάτω γραμμή, $x=1$ τότε θα δημιουργηθεί η μεταβλητή x , ενώ αν δηλώσει την μεταβλητή $X=3$ τότε δημιουργεί την μεταβλητή X . Για αρχάριους προγραμματιστές, η διαφορά μεταξύ x και X είναι δύσκολο να εντοπιστεί και η γλώσσα δεν δίνει κάποιες επιλογές ώστε να τους βοηθήσει.

Σε μια άλλη μελέτη [8] οι φοιτητές προσπάθησαν να ολοκληρώσουν μια άσκηση αντικειμενοστρεφή προγραμματισμού στο πλαίσιο ενός εισαγωγικού μαθήματος προγραμματισμού, χρησιμοποιώντας την γλώσσα Python. Οι δυσκολίες που αντιμετώπισαν κυρίως οι φοιτητές είχαν να κάνουν με τις παραμέτρους σε μεθόδους καθώς και με τις αναφορές των αντικειμένων (object-dot notation). Οι δυσκολίες αυτές είναι ιδιαίτερα κρίσιμες για την εκμάθηση προηγμένων αντικειμενοστρεφών εννοιών για τη γλώσσα προγραμματισμού Python. Υπάρχουν αρκετές αναφορές [32] για τους φοιτητές που δυσκολεύονται να μάθουν αντικειμενοστρεφείς αρχές, συμπεριλαμβανομένης της κληρονομικότητας. Επίσης μια άλλη μελέτη [19] από τα αντικειμενοστρεφή προγράμματα των φοιτητών, έδειξε ότι οι φοιτητές συνήθως παρερμηνεύουν την σχέση μεταξύ μεθόδων και αντικειμένων και τον τρόπο με τον οποίο λειτουργούν οι μέθοδοι.

Επίσης, σε μια έρευνα που διεξήχθη στο πανεπιστήμιο ΝτεΠολ [7] όπου έλαβαν μέρος 36 φοιτητές (6 γυναίκες και 30 άνδρες), καταγράφηκαν τα στοιχεία του αντικειμενοστρεφή προγραμματισμού με την χρήση της γλώσσας Python που οδηγούν σε μαθητικές δυσκολίες. Χρησιμοποιήθηκε μια άσκηση όπου οι φοιτητές καλούνται να ορίσουν μια μέθοδο που απαιτεί αναφορές σε αντικείμενα και έχει καταγραφεί η συχνότητα των σφαλμάτων που προκύπτουν από αυτά τα στοιχεία. Από τα παραπάνω προέκυψαν τα παρακάτω σφάλματα από τις απαντήσεις των φοιτητών. Οι δυσκολίες έχουν να κάνουν με την παράλειψη του αντικειμένου self στη λίστα παραμέτρων (53%) ή την αποτυχία αναφοράς των δεδομένων λίστας από το αντικείμενο self (36%). Οι περισσότερες από τις άλλες δυσκολίες αφορούν αναφορές που έχουν να κάνουν με την μεταβλητή αντιστοίχισης είτε με την επιστρεφόμενη μεταβλητή (19%).

Όσον αφορά επίσης τον αντικειμενοστρεφή προγραμματισμό, ο ορισμός της κλάσης (class) στην Python, δεν περιέχει συντακτικές δηλώσεις ορατότητας (π.χ. public, private, protected). Η Python δεν επιβάλλει αυστηρά κανένα έλεγχο πρόσβασης. Αυτό προκαλεί μια δυσφορία καθώς ο τυπικός έλεγχος πρόσβασης είναι ένα σημαντικό εργαλείο για τους προγραμματιστές λογισμικού [30]. Επιπρόσθετα, αρκετές μελέτες έχουν επισημάνει ότι οι αρχάριοι προγραμματιστές έχουν δυσκολία με τις παραμέτρους και τα στοιχεία αναφοράς των αντικειμένων [19]. Η εκμάθηση προγραμματισμού με την Python περιλαμβάνει και αυτή το πρόβλημα με την μεταβίβαση των παραμέτρων αλλά εισάγει επίσης ένα νέο ζήτημα με τη μάθηση ορισμού μιας κλάσης (class). Όπως πολλές αντικειμενοστρεφείς γλώσσες που χρησιμοποιούνται σε μαθήματα εισαγωγής στον

προγραμματισμό, όπως η Java, για να καλέσουμε μια μέθοδο (function) χρησιμοποιούμε την παρακάτω σύνταξη:

```
obj1.move(steps)
```

Ωστόσο, στον ορισμό της μεθόδου θα πρέπει να οριστεί μια επιπλέον παράμετρος για τη λήψη του αντικειμένου (instance object).

```
def move(self, distance)
```

Ο επεξεργαστής της Python μετατρέπει την μέθοδο (move) έτσι ώστε το αντικείμενο (obj1) να περάσει ως πρώτη παράμετρος της μεθόδου. Η εκμάθηση του ορισμού μιας κλάσης στην Python, απαιτεί μεγάλη κατανόηση του τρόπου με τον οποίο γίνεται το πέρασμα των παραμέτρων, προκειμένου να αποφευχθεί σύγχυση στους αρχάριους προγραμματιστές. Οι φοιτητές χρειάζονται αρκετή εξάσκηση για το πέρασμα των παραμέτρων, ώστε να αποφευχθούν τα λάθη και να κατανοήσουν επαρκώς την λειτουργικότητα της γλώσσας.

Τέλος, πολλοί φοιτητές δυσκολεύονται να χρησιμοποιήσουν σωστά την αναφορά σε ένα αντικείμενο με την χρήση της τελείας (dot notation). Συγχέουν την αναφορά στο ίδιο το αντικείμενο (self) με μια αναφορά σε ένα χαρακτηριστικό που ανήκει στο ίδιο το αντικείμενο (π.χ. self.s) [8]. Γενικά, τα σφάλματα που κάνουν οι αρχάριοι προγραμματιστές με την σωστή χρήση της αναφοράς, είναι κοινά και σε άλλες γλώσσες προγραμματισμού.

Στο Πειραματικό Γυμνάσιο Πειραιά συμμετείχαν 80 μαθητές της Α τάξης στο τέλος του σχολικού έτους 2016-2017, σε μια έρευνα που διεξήχθη για τις δυσκολίες που αντιμετωπίζουν οι μαθητές με την έννοια της μεταβλητής στον προγραμματισμό με τη γλώσσα Python. Οι μαθητές δεν δυσκολεύτηκαν να κατανοήσουν τις ιδιότητες της έννοιας της μεταβλητής. Παρόλα αυτά υπήρχε δυσκολία με τις εντολές εκχώρησης σε ένα πρόγραμμα αλλά και με τη δυναμική τροποποίηση των τιμών της μεταβλητής (30%) [53]. Επίσης οι μαθητές δυσκολεύτηκαν στην κατανόηση των λογικών τιμών true ή false.

Παρακάτω παρουσιάζεται ο Πίνακας με τις βασικές κατηγορίες δυσκολιών που αντιμετωπίζουν οι μαθητές κατά την εκμάθηση της γλώσσας προγραμματισμού Python.

Πίνακας 5 - Βασικές κατηγορίες δυσκολιών που αντιμετωπίζουν οι μαθητές

| Βασικές κατηγορίες δυσκολιών |
|--------------------------------------------------------------------|
| 1. Έλλειψη δήλωσης του τύπου των μεταβλητών [19][53] |
| 2. Παράλειψη αντικειμένου self στη λίστα παραμέτρων [7][8] |
| 3. Μεταβλητές αντιστοίχισης και επιστρεφόμενες μεταβλητές [19][53] |
| 4. Έλλειψη συντακτικών δηλώσεων ορατότητας των κλάσεων [30] |
| 5. Δυσκολία της χρήσης αναφοράς σε ένα αντικείμενο [7][8][32] |
| 6. Δυσκολία δυναμικής τροποποίησης των τιμών μιας μεταβλητής [53] |

5.3 Διδακτικές Προσεγγίσεις

Η διδασκαλία του προγραμματισμού είναι μια πολύπλοκη διαδικασία και αρκετά δύσκολη στον εισαγωγικό προγραμματισμό. Ένα από τα κοινά προβλήματα που αντιμετωπίζουν τα τμήματα της Επιστήμης Υπολογιστών είναι η έλλειψη βασικών δεξιοτήτων προγραμματισμού που έχουν αναφερθεί από τους εκπαιδευτές των εισαγωγικών μαθημάτων προγραμματισμού και πως να εξοπλίσουν τους μαθητές με καλύτερες δεξιότητες μετά από τα εισαγωγικά μαθήματα. Τα πρώτα μαθήματα προγραμματισμού συνήθως δίνουν έμφαση στις αρχές του αντικειμενοστρεφούς προγραμματισμού και του σχεδιασμού από την αρχή. Επίσης κάποια μαθήματα στην αρχή βασίζονται περισσότερο σε μια πιο παραδοσιακή διαδικαστική προσέγγιση.

Υπάρχουν διάφορες προσεγγίσεις για τη διδασκαλία του προγραμματισμού αλλά υπάρχει και μια συνεχιζόμενη συζήτηση στην εκπαιδευτική κοινότητα σχετικά με την καλύτερη προσέγγιση για τη διδασκαλία του εισαγωγικού προγραμματισμού [9].

Υπάρχουν κάποιες πειραματικές μελέτες που συγκρίνουν την αντικειμενοστρεφή προσέγγιση με την παραδοσιακή διαδικαστική.

Οι αρχάριοι προγραμματιστές μπορούν να επωφεληθούν από την χρήση της Python ως κύριας γλώσσας προγραμματισμού στα εισαγωγικά μαθήματα. Καθώς η Python κερδίζει δημοτικότητα, εμφανίζονται διάφορα μοντέλα για την επιτυχή εφαρμογή των εισαγωγικών μαθημάτων. Στόχος είναι, οι αρχάριοι προγραμματιστές να εστιάσουν σε πτυχές της επίλυσης προβλημάτων, της ανάπτυξης και της κατανόησης αλγορίθμων. Δυστυχώς, πολλές σύγχρονες γλώσσες προγραμματισμού απαιτούν από τους μαθητές μεταβούν σε πιο προηγμένες έννοιες προγραμματισμού σε ένα σημείο που είναι πολύ νωρίς για αυτούς. Αυτό τους δημιουργεί μια πιθανή αποτυχία, όχι λόγω της επιστήμης των υπολογιστών, αλλά λόγω του σχήματος που χρησιμοποιείται. Η Python είναι μια εξαιρετική γλώσσα για τη διδασκαλία εισαγωγικών μαθημάτων πληροφορικής. Έχει πολλά πλεονεκτήματα που μπορούν να βοηθήσουν στην ελαχιστοποίηση του σεναρίου αποτυχίας που αναφέρεται παραπάνω. Παρακάτω παρουσιάζονται μια ποικιλία μοντέλων μαθημάτων και προσεγγίσεων που χρησιμοποιούνται στην Python.

Introduction to Media Computation

Ο Mark Guzdial [9], αναφέρει την χρήση της γλώσσας Python σε εισαγωγικό μάθημα προγραμματισμού. Στο μάθημα, «Introduction to Media Computation», οι φοιτητές μπορούν να μάθουν εισαγωγικό προγραμματισμό μέσω χειρισμού πολυμέσων, όπως δημιουργία φίλτρων εικόνας, συναρμολόγηση ήχων, δημιουργία HTML από βάσεις δεδομένων και δημιουργία βίντεο και κινούμενων εικόνων με ψηφιακά ειδικά εφέ. Το συγκεκριμένο μάθημα το παρακολουθούν περίπου 300 φοιτητές το εξάμηνο, με πολύ υψηλό ποσοστό επιτυχίας (περίπου 90%). Το ίδιο αυτό μάθημα έχει αναπτυχθεί και σε Java και έχει χρησιμοποιηθεί σε ένα καλοκαιρινό εργαστήριο. Η διαπίστωση της παραπάνω προσέγγισης είναι ότι οι φοιτητές που χρησιμοποίησαν Python κάνανε ενδιαφέροντα έργα σε λιγότερο χρόνο λόγω της λιγότερο περίπλοκης σύνταξης της γλώσσας.

Python for S60 (PyS60)

Το 2006 δημιουργήθηκε ένα πρακτικό σεμινάριο το οποίο παρουσίαζε έναν νέο τρόπο δημιουργικού προγραμματισμού πολυμέσων και εφαρμογών για κινητά τηλέφωνα. Το συγκεκριμένο σεμινάριο είναι παιδαγωγικά διαμορφωμένο να διδάσκει

τον προγραμματισμό ενός μεγάλου συνόλου λειτουργιών του κινητού τηλεφώνου όπως η κάμερα, η ανταλλαγή μηνυμάτων, τα γραφικά, το Bluetooth, το Wi-Fi [20] κ.α Το σεμινάριο αυτό, χρησιμοποιεί το Python for S60 (PyS60) , όπου παρέχει ένα περιβάλλον εκτέλεσης κώδικα Python για το κινητό τηλέφωνο. Αναπτύχθηκε από τη Nokia Research ως έργο ανοιχτού κώδικα [50]. Το σεμινάριο χρησιμοποιεί το PyS60 σε συνδυασμό με ένα σύνολο παραδειγμάτων κώδικα. Επιτρέπει την ανάπτυξη εφαρμογών για κινητά ακόμη και από αρχάριους προγραμματιστές. Οι μαθητές μπορούν να δημιουργήσουν ισχυρές εφαρμογές με βάση τις δικές τους ιδέες και να συμβάλουν στον χώρο της ανάπτυξης εφαρμογών για κινητά τηλέφωνα. Αυτό το σεμινάριο έχει διδαχθεί σε περισσότερες από 40 περιπτώσεις σε ακαδημαϊκούς και βιομηχανικούς κλάδους και σε περισσότερες από 17 χώρες. Μεταξύ αυτών είναι οι Yahoo Research, NTU Taiwan, MIT και Stanford University [20]. Η προσέγγιση που ακολουθείται σε αυτό το σεμινάριο είναι η πρακτική εξάσκηση με κώδικα Python, που σημαίνει ότι οι συμμετέχοντες ακούνε πρώτα την εξήγηση του κώδικα και στη συνέχεια χρησιμοποιούν τον υπολογιστή τους για να γράψουν τον κώδικα σε μορφή σεναρίου και στη συνέχεια μεταφέρουν τον κώδικα στο κινητό τηλέφωνο για δοκιμή. Αυτή η διαδικασία που αναλύθηκε παραπάνω, προσφέρει στους αρχάριους προγραμματιστές ισχυρή ανατροφοδότηση και υψηλό κίνητρο για μάθηση.

Εισαγωγή στην Python με την παιδαγωγική μέθοδο POGIL

Στο κολέγιο Chestnut Hill της Φιλαδέλφεια, το φθινόπωρο του 2012, στο μάθημα προγραμματισμού με τη γλώσσα Python χρησιμοποιήθηκε η παιδαγωγική μέθοδος POGIL. Η μέθοδος στηρίζεται στη θεωρία της συνεργατικής μάθησης [49]. Στο μάθημα αναπτύχθηκαν 14 δραστηριότητες που επέτρεψαν στους φοιτητές να εξερευνήσουν τη γλώσσα από μόνοι τους σε ομάδες τριών ή τεσσάρων φοιτητών, αναλαμβάνοντας ο καθένας από έναν ρόλο όπως ορίζονται με την μέθοδο POGIL [26]. Ο σκοπός της συγκεκριμένης προσέγγισης ήταν, αφού οι φοιτητές είχαν ήδη μάθει τις βασικές δομές προγραμματισμού στο πρώτο μέρος του μαθήματος χρησιμοποιώντας την παραδοσιακή μορφή εκμάθησης, το δεύτερο μέρος να γίνει ενσωμάτωση της μεθόδου POGIL για να τονίσει την σημασία της μάθησης μέσα από την συνεργασία των φοιτητών. Βοήθησε τους φοιτητές να κατανοήσουν τη σημασία της ενεργού συμμετοχής στη μάθηση και επίλυση προβλημάτων μέσα σε ένα συνεργατικό περιβάλλον.

Προγραμματισμός Python με την χρήση του ρομπότ Scribbler

Σε ένα εισαγωγικό μάθημα προγραμματισμού με τη γλώσσα Python επιλέχθηκε το ρομπότ Scribbler [51] για τη διδασκαλία του μαθήματος [33]. Οι μαθητές χωρίστηκαν σε ομάδες των δύο ατόμων και εκτέλεσαν τα προγράμματα τους με την χρήση του ρομπότ. Με αυτό τον τρόπο διδασκαλίας, οι μαθητές ενθουσιάστηκαν που μπορούσαν να κάνουν αρκετά πράγματα με τον νέο τρόπο προγραμματισμού. Αυτό το εργαστήριο παρείχε στους εκπαιδευτικούς χρήσιμες συμβουλές και οδηγίες για την ενσωμάτωση του προγράμματος με την χρήση του ρομπότ και παρουσιάστηκαν στατιστικά στοιχεία σχετικά με τις αξιολογήσεις των μαθητών.

Το πακέτο γραφικών Python cs1graphics

Το 2005 στο Πανεπιστήμιο του Σαιντ Λούις, επανασχεδιάστηκε το εισαγωγικό μάθημα προγραμματισμού χρησιμοποιώντας την Python ως εκπαιδευτική γλώσσα καθώς και την ενσωμάτωση ενός πακέτου γραφικών το cs1graphics [29]. Το πακέτο είναι απλό στην χρήση αλλά και αρκετά πλούσιο για να παρέχει σημαντική βοήθεια στην εκμάθηση της γλώσσας. Ο μαθητής μπορεί να δημιουργήσει ενδιαφέρουσες εικόνες μέσα από αυτό το πακέτο μόνο με μια μικρή καθοδήγηση από τους διδάσκοντες. Μέσα από αυτή την προσέγγιση, οι μαθητές μπορούν να διδαχθούν τις αντικειμενοστρεφείς αρχές, την ροή ελέγχου ενός προγράμματος, τις κλάσεις, την κληρονομικότητα και την αναδρομή. Αποτελεί ένα πολύτιμο εργαλείο για τους μαθητές των εισαγωγικών μαθημάτων προγραμματισμού και είναι διαθέσιμο στο www.cs1graphics.org.

Από τα παραπάνω προκύπτουν ότι η γλώσσα προγραμματισμού Python, είναι μια βιώσιμη γλώσσα για τη διδασκαλία εισαγωγικών μαθημάτων της επιστήμης των υπολογιστών. Μετά την ανάλυση των παραπάνω διδακτικών προσεγγίσεων οδηγούμαστε στο συμπέρασμα της ευκολίας με την οποία οι μαθητές μπορούν να μάθουν τη γλώσσα μέσα από αυτούς τους τρόπους εκμάθησης, όπως επίσης αυξάνεται η παρακίνηση τους να ασχοληθούν επαγγελματικά με την επιστήμη των υπολογιστών.

6 Μελέτες σύγκρισης με άλλες γλώσσες προγραμματισμού

Το ζήτημα της γλώσσας που πρέπει να χρησιμοποιηθεί στην εισαγωγή στον προγραμματισμό έχει προκαλέσει αρκετές συζητήσεις, συχνά βασισμένες σε συναισθηματικές απόψεις. Έχουν διεξαχθεί αρκετές μελέτες σχετικά με τα οφέλη μεμονωμένων γλωσσών ή συγκρίσεις μεταξύ δύο γλωσσών, αλλά εξακολουθεί να υπάρχει έλλειψη αντικειμενικών δεδομένων που χρησιμοποιούνται για την ενημέρωση αυτών των συγκρίσεων.

Η επιλογή της γλώσσας προγραμματισμού, που θα χρησιμοποιηθεί στην εκπαίδευση υπήρξε επίκαιρο ζήτημα εδώ και αρκετό καιρό. Στις αρχές της δεκαετίας του 1980, ο Tharp [43] έκανε μια γλωσσική σύγκριση των COBOL, FORTRAN, Pascal, PL-I και Shobol, όπου επικεντρώθηκε κυρίως στην αποτελεσματικότητα της σύνταξης και την ταχύτητα της εφαρμογής κώδικα, προκειμένου να παρέχει στους εκπαιδευτικούς, πληροφορίες που απαιτούνται για να επιλέξουν την κατάλληλη γλώσσα προγραμματισμού. Σήμερα, οι σκέψεις επικεντρώνονται περισσότερο στις παιδαγωγικές ανησυχίες και το φάσμα των γλωσσών είναι ακόμη ευρύτερο.

Παρακάτω παρουσιάζονται και αναλύονται κριτήρια που βασίζονται σε αποφάσεις σχεδιασμού [27] που χρησιμοποιούνται από διακεκριμένους δημιουργούς γλωσσών προγραμματισμού. Τα κριτήρια αυτά, χρησιμοποιούνται για να συγκρίνουν γλώσσες που χρησιμοποιούνται επί του παρόντος σε εισαγωγικά μαθήματα προγραμματισμού. Γίνονται συστάσεις για το πώς αυτά τα κριτήρια μπορούν να χρησιμοποιηθούν ή να προσαρμοστούν σε διαφορετικές καταστάσεις.

Η γλώσσα είναι κατάλληλη για διδασκαλία

Αυτό το κριτήριο έχει προταθεί από τον Niklaus Wirth [52], δημιουργό της γλώσσας Pascal, όπου επισημαίνει ότι οι ευρέως χρησιμοποιούμενες γλώσσες προγραμματισμού δεν είναι απαραίτητα και οι καλύτερες γλώσσες για διδασκαλία. Για να ικανοποιηθεί αυτό το κριτήριο, η γλώσσα προγραμματισμού θα πρέπει να είναι σχεδιασμένη με γνώμονα την διδασκαλία. Θα πρέπει να έχει απλή σύνταξη και φυσική σημασιολογία χωρίς να περιέχει δύσκολες έννοιες που θα προκαλούν σύγχυση στους αρχάριους προγραμματιστές. Επίσης, τα εργαλεία που θα χρησιμοποιηθούν για την ανάπτυξη κώδικα θα πρέπει να είναι και αυτά εύχρηστα.

Η γλώσσα θα πρέπει να προσφέρει ένα γενικό πλαίσιο

Ο πρωταρχικός στόχος κάθε εισαγωγικού μαθήματος προγραμματισμού είναι να εισάγει τους μαθητές στον προγραμματισμό. Ως εκ τούτου, η ίδια η γλώσσα δεν είναι το επίκεντρο της διδασκαλίας και οι δεξιότητες που μαθαίνονται σε μια γλώσσα πρέπει να μεταφέρονται σε άλλες κοινές γλώσσες [28]. Για να ικανοποιηθεί αυτό το κριτήριο, η γλώσσα θα πρέπει να επιτρέπει την εκμάθηση θεμελιωδών αρχών του προγραμματισμού, οι οποίες θα χρησιμεύουν ως μια εξαιρετική βάση για την εκμάθηση άλλων γλωσσών προγραμματισμού αργότερα.

Η γλώσσα να προωθεί μια νέα προσέγγιση για τη διδασκαλία λογισμικού

Σε ένα εισαγωγικό μάθημα προγραμματισμού, η γλώσσα είναι μόνο ένα μέρος της μάθησης για έναν αρχάριο προγραμματιστή. Είναι επίσης πολύ σημαντικό όταν η ίδια η γλώσσα και τα σχετικά εργαλεία μπορούν να βοηθήσουν στην εκμάθηση της εφαρμογής της γλώσσας. Για να επιτευχθεί αυτό, η γλώσσα δεν πρέπει να περιορίζεται μόνο στην εφαρμογή, αλλά να καλύπτει πολλές πτυχές της διαδικασίας ανάπτυξης λογισμικού [28]. Η γλώσσα πρέπει να σχεδιαστεί ως μια ολόκληρη μεθοδολογία για την κατασκευή λογισμικού με βάση 1) την γλώσσα και 2) ένα σύνολο αρχών εργαλείων και βιβλιοθηκών.

Η γλώσσα να είναι διαδραστική και να διευκολύνει την ταχεία ανάπτυξη κώδικα

Η δυνατότητα εφαρμογής νέων ιδεών προγραμματισμού χωρίς να απαιτείται το πλαίσιο ενός πλήρους προγράμματος είναι πολύτιμη για τους αρχάριους προγραμματιστές. Η δυνατότητα της γρήγορης έναρξης συγγραφής και κατανόησης απλών προγραμμάτων παρακινεί και εμπνέει τους αρχάριους προγραμματιστές [15]. Για να ικανοποιηθεί αυτό το κριτήριο, η γλώσσα και τα περιβάλλοντα που υποστηρίζουν την χρήση της θα πρέπει να επιτρέπουν στους αρχάριους προγραμματιστές να εφαρμόζουν

νέες ιδέες που αποκτήθηκαν, χωρίς να χρειάζεται να καθορίσουν το πλαίσιο μιας πλήρους εφαρμογής. Το γλωσσικό περιβάλλον πρέπει να παρέχει στους μαθητές διαδραστικό και άμεσο σχόλιο σχετικά με την πρόοδο τους.

Η γλώσσα θα πρέπει να προωθεί τη συγγραφή σωστών προγραμμάτων

Ο στόχος αυτού του κριτηρίου είναι να απομακρυνθούμε από την επικρατούσα προσέγγιση «δοκιμή και σφάλμα» στην κατασκευή λογισμικού. Για να επιτευχθεί αυτό, πρέπει να δοθούν στους μαθητές τρόποι να διασφαλίσουν ότι ο κώδικας που γράφουν είναι σωστός και δεν περιέχει σφάλματα.

Η γλώσσα να επιτρέπει την επίλυση προβλημάτων σε «μικρά κομμάτια κώδικα»

Είναι επιθυμητό για κάθε προγραμματιστή να μπορεί να επικεντρώνεται σε ένα κομμάτι ενός προβλήματος πριν προχωρήσει στο επόμενο πρόβλημα. Μια γλώσσα που υποστηρίζει την αποσύνθεση των προβλημάτων είναι επιθυμητή. Για να συμβεί αυτό, η γλώσσα θα πρέπει να υποστηρίζει την διαμόρφωση (modularization) σε συναρτήσεις και διαδικασίες.

Η γλώσσα παρέχει ένα αξιόπιστο περιβάλλον ανάπτυξης

Όταν ένα αρχάριος προγραμματιστής αρχίζει να προγραμματίζει είναι σημαντικό να μπορεί να κατανοήσει τη διαδικασία με την οποία ένα πηγαίος κώδικας γίνεται ένα εκτελέσιμο πρόγραμμα. Ορισμένα περιβάλλοντα προγραμματισμού αποκρύπτουν λεπτομέρειες αυτής της διαδικασίας χάριν απλότητας και ταχύτητας, που μπορεί να είναι επωφελής για έναν επαγγελματία προγραμματιστή αλλά λιγότερο για έναν αρχάριο. Άλλα περιβάλλοντα, βοηθούν στην γεφύρωση του χάσματος μεταξύ σχεδιασμού και υλοποίησης, μετατρέποντας για παράδειγμα τα αρχιτεκτονικά διαγράμματα σε κώδικα [28]. Για να επιτευχθεί αυτό το κριτήριο, το περιβάλλον ανάπτυξης θα πρέπει να διαθέτει μια διαισθητική επαφή (GUI) για σχεδιασμό και υλοποίηση, παρέχοντας πρόσβαση σε λειτουργίες και βιβλιοθήκες, τόσο για τον βασικό όσο και για τον προχωρημένο προγραμματισμό.

Η γλώσσα θα πρέπει να παρέχει κοινότητα υποστήριξης

Είτε μια γλώσσα προγραμματισμού είναι εμπορική, είτε είναι ανοιχτού κώδικα, η «μακροζωία» (longevity) εξαρτάται από την υποστήριξη της γλώσσας στην ευρύτερη προγραμματιστική κοινότητα [15]. Όταν η υποστήριξη είναι περιορισμένη, οι πόροι και η υποστήριξη της γλώσσας, ενδέχεται να αποτελούν περιορισμό στους εκπαιδευτές και στους μαθητές. Για να ικανοποιηθεί αυτό το κριτήριο, πρέπει να υπάρχει επαρκής υποστήριξη στους φοιτητές, στους καθηγητές καθώς και σε όσους άλλους ενδιαφέρονται να μάθουν και να χρησιμοποιούν την γλώσσα. Αυτή η υποστήριξη μπορεί να διατίθεται σε διάφορες μορφές, όπως ιστοσελίδες, βιβλία, σεμινάρια, ασκήσεις κτλ.

Η γλώσσα θα πρέπει να είναι ανοιχτού κώδικα, ώστε ο καθένας να μπορεί να συμβάλει στην ανάπτυξη της

Ένα από τα οφέλη ενός λογισμικού ανοιχτού κώδικα είναι η μείωση του κόστους. Ένα άλλο πλεονέκτημα είναι η διαλειτουργικότητα του λογισμικού, δηλαδή να μπορεί να είναι συμβατό με άλλα συστήματα. Για να επιτευχθεί αυτό το κριτήριο, η γλώσσα θα πρέπει να είναι εφεύρεση μιας ομάδας που δεν επιδιώκει να δημιουργήσει ένα εμπορικό προϊόν και στην οποία κάποιος να μπορεί να συνεισφέρει αν θέλει.

Η γλώσσα να υποστηρίζεται με συνέπεια σε όλα τα περιβάλλοντα

Ο προγραμματισμός πραγματοποιείται σε διαφορετικά λειτουργικά συστήματα και σε διαφορετικά μηχανήματα. Είναι χρήσιμο να προσφέρονται εργαλεία στους μαθητές, τα οποία μπορούν να χρησιμοποιηθούν σε πολλά περιβάλλοντα και όχι μόνο σε ένα. Για να γίνει αυτό η γλώσσα θα πρέπει να είναι διαθέσιμη σε διάφορες πλατφόρμες.

Η γλώσσα να είναι ελεύθερη και εύκολα διαθέσιμη

Για τους μαθητές, το κόστος μιας γλώσσας ή ενός περιβάλλοντος προγραμματισμού είναι ένα σημαντικό ζήτημα. Οι μαθητές θα σταματήσουν να προγραμματίζουν αν μια γλώσσα ή ένα IDE είναι ακριβό. Άρα για να αποφευχθεί αυτό, η γλώσσα θα πρέπει να είναι απαλλαγμένη από κόστη ή συνδρομές και να είναι διαθέσιμη σε όλο τον κόσμο χωρίς περιορισμούς.

Η γλώσσα να υποστηρίζεται με καλό διδακτικό υλικό

Είναι πολύ σημαντικό πλεονέκτημα τόσο για τους εκπαιδευτές όσο και για τους μαθητές εάν το εκπαιδευτικό υλικό είναι διαθέσιμο για μια συγκεκριμένη γλώσσα προγραμματισμού [27]. Για να επιτευχθεί αυτό το κριτήριο, βιβλία και άλλο υλικό θα πρέπει να είναι διαθέσιμα για χρήση στην τάξη.

Η γλώσσα να μην χρησιμοποιείται μόνο στην εκπαίδευση

Οι μαθητές μπορεί να έχουν περισσότερα κίνητρα από μια γλώσσα που δεν χρησιμοποιείται απλώς σε ένα εκπαιδευτικό περιβάλλον [27]. Άρα η γλώσσα θα πρέπει να σχετίζεται επίσης και με άλλους τομείς εκτός από την εκπαίδευση όπως για παράδειγμα στην βιομηχανία, και να είναι κατάλληλη για την ανάπτυξη μεγάλων πραγματικών εφαρμογών.

Η γλώσσα να είναι επεκτάσιμη

Οι αρχάριοι προγραμματιστές μπορεί να μην αναμένεται να γράφουν πρόσθετα επέκτασης (extension modules) σε ένα εισαγωγικό μάθημα προγραμματισμού αλλά η χρήση υπαρχουσών επεκτάσεων γλώσσας έχει τη δυνατότητα να κάνει τη μάθηση πιο παρακινητική και συναρπαστική [27]. Η επεκτασιμότητα επιτρέπει σε μια γλώσσα να μπορεί να εφαρμοστεί σε μια μεγάλη ποικιλία προβλημάτων τόσο στη μάθηση όσο και αργότερα στην επαγγελματική χρήση της. Μια γλώσσα η οποία μπορεί να

χρησιμοποιηθεί αποτελεσματικά μόνο με ένα μικρό υποσύνολο χαρακτηριστικών, διευκολύνει την πρόσβαση σε προηγμένες λειτουργίες που δεν είναι άμεσα προσβάσιμες.

Η γλώσσα είναι αξιόπιστη και αποτελεσματική

Η ταχύτητα εκτέλεσης εξακολουθεί να διαφοροποιεί ορισμένες γλώσσες. Μπορεί ένας αρχάριος προγραμματιστής να μην χρησιμοποιεί σαν γνώμονα αυτό το κριτήριο για να μάθει μια γλώσσα προγραμματισμού, αλλά μπορεί μέσα σε ένα ακαδημαϊκό περιβάλλον, να εξερευνήσει αυτό το ζήτημα, εξετάζοντας την ταχύτητα εκτέλεσης των προγραμμάτων του σε σύγκριση με άλλες γλώσσες προγραμματισμού.

Παρακάτω στον Πίνακα 6 παρουσιάζονται συγκεντρωτικά οι γλώσσες προγραμματισμού που χρησιμοποιούνται στην εισαγωγή στον προγραμματισμό καθώς και ποια κριτήρια που αναφέρθηκαν πιο πάνω πληρούν οι γλώσσες αυτές.

Πίνακας 6 – Σύγκριση γλωσσών προγραμματισμού [27]

| | C | C++ | Java | Pascal | Python | VB |
|--------------------------------------------------------------------------|----------|------------|-------------|---------------|---------------|-----------|
| Η γλώσσα είναι κατάλληλη για διδασκαλία | | | | ✓ | ✓ | |
| Η γλώσσα θα πρέπει να προσφέρει ένα γενικό πλαίσιο | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Η γλώσσα να προωθεί μια νέα προσέγγιση για τη διδασκαλία λογισμικού | | | ✓ | | | |
| Η γλώσσα είναι διαδραστική και να διευκολύνει την ταχεία ανάπτυξη κώδικα | | | | | ✓ | |
| Η γλώσσα θα πρέπει να προωθεί τη συγγραφή σωστών προγραμμάτων | | | | | | |
| Η γλώσσα να επιτρέπει την επίλυση προβλημάτων σε «μικρά κομμάτια κώδικα» | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Η γλώσσα να παρέχει ένα αξιόπιστο | | | | | | |

| περιβάλλον ανάπτυξης | | | | | | |
|------------------------------------------------------------------------------------------------------|----------|------------|-------------|---------------|---------------|-----------|
| | C | C++ | Java | Pascal | Python | VB |
| Η γλώσσα θα πρέπει να παρέχει κοινότητα υποστήριξης | ✓ | ✓ | ✓ | | ✓ | ✓ |
| Η γλώσσα θα πρέπει να είναι ανοιχτού κώδικα, οπότε ο καθένας να μπορεί να συμβάλει στην ανάπτυξη της | | | | | ✓ | |
| Η γλώσσα να υποστηρίζεται με συνέπεια σε όλα τα περιβάλλοντα | ✓ | ✓ | ✓ | ✓ | ✓ | |
| Η γλώσσα να είναι ελεύθερη και εύκολα διαθέσιμη | ✓ | ✓ | ✓ | ✓ | ✓ | |
| Η γλώσσα να υποστηρίζεται με καλό διδακτικό υλικό | | ✓ | ✓ | ✓ | ✓ | ✓ |
| Η γλώσσα να μην χρησιμοποιείται μόνο στην εκπαίδευση | ✓ | ✓ | ✓ | | ✓ | ✓ |
| Η γλώσσα να είναι επεκτάσιμη | ✓ | ✓ | ✓ | | ✓ | ✓ |
| Η γλώσσα είναι αξιόπιστη και αποτελεσματική | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

7 Επίλογος

Σκοπός της παραπάνω έρευνας ήταν η ανάλυση της γλώσσας προγραμματισμού Python, τα βασικά χαρακτηριστικά της γλώσσας, και κυρίως τα περιβάλλοντα προγραμματισμού που χρησιμοποιούνται για την εκμάθηση της από σπουδαστές για την εισαγωγή τους στον προγραμματισμό. Επίσης παρουσιάζονται τα οφέλη της γλώσσας στη διδασκαλία των βασικών εννοιών του προγραμματισμού καθώς και οι διδακτικές προσεγγίσεις που χρησιμοποιούνται σε κάποια ακαδημαϊκά ιδρύματα για την καλύτερη κατανόηση των δομών του προγραμματισμού από τους αρχάριους προγραμματιστές. Τέλος, γίνεται μια συγκριτική ανάλυση της γλώσσας Python με άλλες γλώσσες που χρησιμοποιούνται στην εισαγωγή στον προγραμματισμό όπου παρουσιάζονται τα αποτελέσματα σε έναν συγκριτικό πίνακα.

7.1 Σύνοψη και συμπεράσματα

Αρχικά στο Κεφάλαιο 2, πραγματοποιήθηκε μια ανάλυση της γλώσσας προγραμματισμού Python και των βασικών χαρακτηριστικών της, με σημαντικότερα το γεγονός ότι η γλώσσα Python είναι πολύ απλή και εύκολη στην σύνταξη της, ανοιχτού λογισμικού, υψηλού επιπέδου γλώσσας και αντικειμενοστρεφής.

Στο Κεφάλαιο 3 πραγματοποιήθηκε μια ανάλυση των περιβαλλόντων προγραμματισμού καθώς και των εκπαιδευτικών εργαλείων που χρησιμοποιούνται για τη διδασκαλία της γλώσσας Python. Έγινε καταγραφή των χαρακτηριστικών και των δυνατοτήτων από εννέα περιβάλλοντα καθώς και μια συγκριτική μελέτη των περιβαλλόντων με βάση κάποιες βασικές κατηγορίες γνωρισμάτων καταγράφοντας τις ομοιότητες και τις διαφορές τους, οι οποίες παρουσιάστηκαν σε συγκριτικούς πίνακες. Τέλος, παρουσιάζονται μελέτες αξιολόγησης από φοιτητές και αρχάριους προγραμματιστές που έχουν χρησιμοποιήσει αυτά τα περιβάλλοντα για την εκμάθηση της γλώσσας Python αλλά και για την διδασκαλία των βασικών εννοιών και δομών του προγραμματισμού. Το αποτέλεσμα της σύγκρισης και των αξιολογήσεων είναι ότι τα περιβάλλοντα προγραμματισμού Thonny, CodeSkulptor και OnlinePythonTutor ενσωματώνουν δυνατότητες και εργαλεία που μπορούν να βοηθήσουν τους αρχάριους προγραμματιστές στην εκμάθηση της γλώσσας Python μέσα από διάφορες διδακτικές προσεγγίσεις.

Κατόπιν στο Κεφάλαιο 5, καταγράφηκαν τα οφέλη που έχει η γλώσσα Python στην διδασκαλία. Κατά την ανάπτυξη κώδικα, η γλώσσα παρέχει την λειτουργία των εσοχών για την ομαδοποίηση των εντολών και διαθέτει εργαλεία επεξεργασίας συμβολοσειρών και κανονικών εκφράσεων, ώστε οι σπουδαστές να μπορούν να υλοποιήσουν ένα μεγάλο εύρος από προβλήματα που μπορούν να τους ανατεθούν. Οι σπουδαστές επωφελούνται μέσα από την πρώιμη εξοικείωση τους σε δομές δεδομένων όπως είναι οι λίστες και από την συγγραφή προσθέτων (modules), δίνοντας τους έτσι την δυνατότητα να δίνουν τις δικές τους λύσεις σε προβλήματα. Επίσης, πέρα από τα οφέλη που παρέχει η γλώσσα έγινε και μια καταγραφή των δυσκολιών που αντιμετωπίζουν οι αρχάριοι προγραμματιστές κατά την εισαγωγή τους στον προγραμματισμό με την γλώσσα Python. Μέσα από αυτή την καταγραφή, προέκυψαν κάποιες βασικές κατηγορίες δυσκολιών που έχει η γλώσσα, όπως η έλλειψη της δήλωσης του τύπου των μεταβλητών, η έλλειψη συντακτικών δηλώσεων ορατότητας των κλάσεων στον αντικειμενοστρεφή προγραμματισμό με αποτέλεσμα να οδηγούν συχνά τους σπουδαστές σε λάθη κατά την συγγραφή των προγραμμάτων τους και να μην κατανοούν τις βασικές αρχές του προγραμματισμού.

Τέλος στο Κεφάλαιο 6, έγινε μια καταγραφή και ανάλυση των κριτηρίων που βασίζονται σε αποφάσεις σχεδιασμού και χρησιμοποιούνται από δημιουργούς γλωσσών προγραμματισμού. Με βάση αυτά τα κριτήρια, έγινε μια συγκριτική μελέτη των γλωσσών προγραμματισμού που χρησιμοποιούνται στην διδασκαλία των εισαγωγικών μαθημάτων προγραμματισμού. Το αποτέλεσμα της μελέτης, είναι ότι η γλώσσα Python πληροί τα περισσότερα από αυτά τα κριτήρια.

7.2 Όρια και περιορισμοί της έρευνας

Μέσα από την μελέτη που πραγματοποιήθηκε για την γλώσσα προγραμματισμού Python, των προγραμματιστικών περιβαλλόντων και των διδακτικών προσεγγίσεων που χρησιμοποιούνται στον εισαγωγικό προγραμματισμό, προέκυψαν κάποιοι περιορισμοί που πρέπει να ληφθούν υπόψη.

Στη μελέτη των περιβαλλόντων προγραμματισμού, όπου καταγράφηκαν και αναλύθηκαν τα χαρακτηριστικά και οι δυνατότητες του κάθε περιβάλλοντος, δεν υπήρχαν στην βιβλιογραφία μελέτες αξιολόγησης για όλα τα περιβάλλοντα. Η

αξιολόγηση ενός περιβάλλοντος προγραμματισμού από τους σπουδαστές και τους διδάσκοντες αποτελεί βασικό γνώρισμα, με αποτέλεσμα αυτός ο περιορισμός να δυσκολεύει τη γενίκευση των αποτελεσμάτων της συγκριτικής μελέτης των περιβαλλόντων.

Επίσης, έναν ακόμη περιοριστικό παράγοντα αποτελεί το γεγονός ότι στην παρουσίαση των διδακτικών προσεγγίσεων για τη διδασκαλία της γλώσσας Python, δεν αναφέρεται στη βιβλιογραφία, για κάποιες από αυτές, ο αριθμός των σπουδαστών που συμμετείχαν σε αυτά τα μοντέλα μαθημάτων με αποτέλεσμα να μην γνωρίζουμε στατιστικά το ποσοστό επιτυχίας των σπουδαστών να μάθουν τις έννοιες και τις δομές του εισαγωγικού προγραμματισμού με αυτούς τους τρόπους διδασκαλίας.

Τέλος, η μελέτη επικεντρώθηκε στα περιβάλλοντα προγραμματισμού και απαιτείται περαιτέρω μελέτη των δυσκολιών των σπουδαστών και των διδακτικών προσεγγίσεων που έχουν προταθεί για την αντιμετώπισή τους.

7.3 Μελλοντικές Επεκτάσεις

Λαμβάνοντας υπόψη τους περιορισμούς και τα όρια που αναφέρθηκαν στην προηγούμενη ενότητα, προκύπτουν κάποιες προτάσεις για την επέκταση της παρούσας διπλωματικής εργασίας.

Μια επέκταση της έρευνας για περαιτέρω διερεύνηση του θέματος είναι τα προγραμματιστικά περιβάλλοντα που αναλύθηκαν παραπάνω, να χρησιμοποιηθούν κατά τη διδασκαλία της εισαγωγής προγραμματισμού με την γλώσσα Python σε πανεπιστήμια και σχολεία. Μαθητές σχολείων αλλά και φοιτητές θα μπορούσαν με τη σωστή καθοδήγηση των εκπαιδευτικών και με την χρήση του σωστού περιβάλλοντος να κάνουν τα πρώτα τους βήματα στον προγραμματισμό και στην εκμάθηση της γλώσσας Python. Μετά το πέρας της διδασκαλίας, οι σπουδαστές αλλά και οι εκπαιδευτικοί θα μπορούσαν να αξιολογήσουν τη διδασκαλία του μαθήματος, το αν η γλώσσα Python βοήθησε τους σπουδαστές στην εισαγωγή τους στον προγραμματισμό καθώς και αν το περιβάλλον προγραμματισμού που χρησιμοποιήθηκε ήταν το ιδανικό για την εκμάθηση των βασικών εννοιών και δομών της γλώσσας. Επιπλέον περαιτέρω έρευνα θα μπορούσε να γίνει και για τις διδακτικές προσεγγίσεις που χρησιμοποιούνται στα εισαγωγικά μαθήματα προγραμματισμού. Να μελετηθούν και να καταγράφουν και άλλες διδακτικές προσεγγίσεις που μπορεί να υπάρχουν, και να υπάρχει αξιολόγηση από τους σπουδαστές,

καθώς και η πραγματοποίηση μιας συγκριτικής μελέτης των προσεγγίσεων αυτών. Η σύγκριση των αποτελεσμάτων από την εφαρμογή αυτών των διδακτικών προσεγγίσεων θα οδηγούσε σε πολύ ενδιαφέροντα και βελτιωμένα συμπεράσματα.

Βιβλιογραφία

- 1 Aivar Annamaa. 2015. Thonny,: a Python IDE for Learning Programming. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '15)*. ACM, New York, NY, USA, 343-343. DOI: <http://dx.doi.org/10.1145/2729094.2754849>
- 2 Anaconda (Python Distribution) . In Wikipedia. Retrieved February 20, 2020, from [https://en.wikipedia.org/wiki/Anaconda_\(Python_distribution\)](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution))
- 3 Atanas Radenski. "Python First": A Lab-Based Digital Introduction to Computer Science. Chapman University Orange, CA 92869, USA
- 4 Bedour Alshaigy, Samia Kamal, Faye Mitchell, Clare Martin, and Arantza Aldea. 2015. PILET: an Interactive Learning Tool To Teach Python. In *Proceedings of the Workshop in Primary and Secondary Computing Education (WiPSCE '15)*. ACM, New York, NY, USA, 76-79. DOI: <http://dx.doi.org/10.1145/2818314.2818319>
- 5 Bedour Alshaigy. 2013. Development of an interactive learning tool to teach python programming language. In *Proceedings of the 18th ACM conference on Innovation and technology in computer science education (ITiCSE '13)*. ACM, New York, NY, USA, 344-344. DOI=<http://dx.doi.org/10.1145/2462476.2465601>
- 6 Charles Dierbach. 2014. Python as a first programming language. *J. Comput. Sci. Coll.* 29, 6 (June 2014), 153-154.
- 7 Craig S. Miller and Amber Settle. 2016. Some Trouble with Transparency: An Analysis of Student Errors with Object-oriented Python. In *Proceedings of the 2016 ACM Conference on International Computing Education Research (ICER '16)*. ACM, New York, NY, USA, 133-141. DOI: <http://dx.doi.org/10.1145/2960310.2960327>
- 8 Craig S. Miller, Amber Settle, and John Lalor. 2015. Learning Object-Oriented Programming in Python: Towards an Inventory of Difficulties and Testing Pitfalls. In *Proceedings of the 16th Annual Conference on Information Technology Education (SIGITE '15)*. ACM, New York, NY, USA, 59-64. DOI: <http://dx.doi.org/10.1145/2808006.2808017>

- 9 David Ranum, Bradley Miller, John Zelle, and Mark Guzdial. 2006. Successful approaches to teaching introductory computer science courses with python. In *Proceedings of the 37th SIGCSE technical symposium on Computer science education* (SIGCSE '06). ACM, New York, NY, USA, 396-397. DOI=<http://dx.doi.org/10.1145/1121341.1121465>
- 10 D. Parsons and P. Haden. Parson's programming puzzles: a fun and effective learning tool for first programming courses. In *Proceedings of the 8th Australasian Conference on Computing Education-Volume 52*, pages 157–163. Australian Computer Society, Inc., 2006
- 11 Fionnuala Johnson, Stephen McQuistin, John O'Donnell, 2020. Analysis of Student Misconceptions using Python as an Introductory Programming Language. In *CEP*, January 9, 2020, Durham United Kingdom
- 12 Integrated development environment. (n.d.). In Wikipedia. Retrieved March 20, 2020, from https://en.wikipedia.org/wiki/Integrated_development_environment
- 13 IDLE. (n.d.). In Wikipedia. Retrieved March 20, 2020, from <https://en.wikipedia.org/wiki/IDLE>
- 14 Guido van Rossum. (2011). *An Introduction to Python*. Network Theory Ltd. ISBN-10: 1906966133, ISBN-13: 978-1906966133.
- 15 Guido van Rossum. (1999) "*Computer Programming for Everybody*." Proposal to the Corporation for National Research Initiatives.
- 16 Heda Samimi. 2013. Introduction to the Python programming language. *J. Comput. Sci. Coll.* 29, 1 (October 2013), 8-9.
- 17 Jen Fritz, Margie Matthews, Tom Wulf, Jim Scott, and Jamie Fritz. 2016. University of Cincinnati and Saint Ursula Academy Partnership: Introducing Female High School Students to the Field of Information Technology, Year 2. In *Proceedings of the 17th Annual Conference on Information Technology Education* (SIGITE '16). ACM, New York, NY, USA, 109-109. DOI: <http://dx.doi.org/10.1145/2978192.2978199>
- 18 John Aycock. 2012. μ Python: non-majors programming from the very first lecture. In *Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education* (ITiCSE '12). ACM, New York, NY, USA, 345-350. DOI=<http://dx.doi.org/10.1145/2325296.2325376>

- 19 Joseph D. Oldham 2005. What happens after Python in CS1?. *JCSC 20,6 (June 2005)*, Centre College, Danville, KY.
- 20 Jörgen Scheible. 2007. Mobile phone programming for multimedia. In *Proceedings of the 15th ACM international conference on Multimedia (MM '07)*. ACM, New York, NY, USA, 12-13. DOI=<http://dx.doi.org/10.1145/1291233.1291240>
- 21 Juha Helminen and Lauri Malmi. 2010. Jype - a program visualization and programming exercise tool for Python. In *Proceedings of the 5th international symposium on Software visualization (SOFTVIS '10)*. ACM, New York, NY, USA, 153-162. DOI=<http://dx.doi.org/10.1145/1879211.1879234>
- 22 Juha Helminen, Petri Ihantola, and Ville Karavirta. 2013. Recording and analyzing in-browser programming sessions. In *Proceedings of the 13th Koli Calling International Conference on Computing Education Research (Koli Calling '13)*. ACM, New York, NY, USA, 13-22. DOI=<http://dx.doi.org/10.1145/2526968.2526970>
- 23 K. K. Agarwal and A. Agarwal. Python for CS1, CS2 and beyond. *J. Comput. Sci. Coll.*, 20(4):262–270, Apr. 2005.
- 24 Krishna K. Agarwal, Achla Agarwal, and Catherine Hiller. 2013. Rapid applications development in Python. *J. Comput. Sci. Coll.* 28, 4 (April 2013), 42-48.
- 25 Landofcode.com, *Code editor features*. Διαθέσιμο: <http://www.landofcode.com/code-editors/code-editors-features.php> (22 Ιανουαρίου, 2020).
- 26 Lisa M. Olivieri. 2013. Piloting POGIL in an introductory Python programming course. *J. Comput. Sci. Coll.* 28, 6 (June 2013), 194-195.
- 27 Mannila Linda, Michael de Raadt. 2006. An Objective Comparison of Languages for Teaching Introductory Programming. In *Proceedings of Koli Calling 2006*.
- 28 Meyer, B. Towards an Object-Oriented Curriculum. In *Proceedings of 11th international TOOLS conference*. (Santa Barbara, United States, August 1993). Prentice Hall 1993, 1993, 585 - 594.
- 29 Michael H. Goldwasser and David Letscher. 2008. A python graphics package for the first day and beyond. In *Proceedings of the 13th annual conference on*

- Innovation and technology in computer science education* (ITiCSE '08). ACM, New York, NY, USA, 326-326. DOI=<http://dx.doi.org/10.1145/1384271.1384369>
- 30 McIver, L., (2002), *Evaluating Languages and Environments for Novice Programmers*, Fourteenth Annual Workshop of the Psychology of Programming Interest Group (PPIG 2002), Middlesex, UK, 100-110.
- 31 Michael H. Goldwasser and David Letscher. 2008. Teaching an object-oriented CS1 -: with Python. *SIGCSE Bull.* 40, 3 (June 2008), 42-46. DOI=<http://dx.doi.org/10.1145/1597849.1384285>
- 32 N. Liberman, C. Beerli, and Y. Ben-David Kolikant. Difficulties in learning inheritance and polymorphism. *ACM Transactions on Computing Education* (TOCE), 11(1):4, 201
- 33 Patricia Roth Pierce. 2010. Adding a robot project to a CS1 course. In *Proceedings of the 48th Annual Southeast Regional Conference* (ACM SE '10). ACM, New York, NY, USA, , Article 79 , 1 pages. DOI=<http://dx.doi.org/10.1145/1900008.1900115>
- 34 Peter J. Robinson. 2011. MyPyTutor: an interactive tutorial system for Python. In *Proceedings of the Thirteenth Australasian Computing Education Conference - Volume 114* (ACE '11), John Hamer and Michael de Raadt (Eds.), Vol. 114. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 155-160.
- 35 Philip J. Guo. 2013. Online python tutor: embeddable web-based program visualization for cs education. In *Proceeding of the 44th ACM technical symposium on Computer science education* (SIGCSE '13). ACM, New York, NY, USA, 579-584. DOI: <http://dx.doi.org/10.1145/2445196.2445368>
- 36 Python.org. *Graphical User Interfaces with Tk*. Διαθέσιμο: <https://docs.python.org/3/library/tk.html> (15 Φεβρουαρίου, 2020).
- 37 Python. (n.d.). In Wikipedia. Retrieved February 15, 2020, from <https://el.wikipedia.org/wiki/Python>
- 38 R. J. Enbody and W. F. Punch. Performance of Python CS1 students in mid-level non-Python CS courses. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education, SIGCSE '10*, pages 520–523, New York, NY, USA, 2010. ACM
- 39 Stephen H. Edwards, Daniel S. Tilden, and Anthony Allevato. 2014. Pythy: improving the introductory python programming experience. In *Proceedings of the*

- 45th ACM technical symposium on Computer science education (SIGCSE '14)*. ACM, New York, NY, USA, 641-646. DOI: <http://dx.doi.org/10.1145/2538862.2538977>
- 40 Spyder. (n.d.). In Wikipedia. Retrieved March 15, 2020, from [https://en.wikipedia.org/wiki/Spyder_\(software\)](https://en.wikipedia.org/wiki/Spyder_(software))
- 41 Swaroop C H. (2013). *A byte of Python*. Swaroop CH.
- 42 Terry Tang, Scott Rixner, and Joe Warren. 2014. An environment for learning interactive programming. In *Proceedings of the 45th ACM technical symposium on Computer science education (SIGCSE '14)*. ACM, New York, NY, USA, 671-676. DOI: <http://dx.doi.org/10.1145/2538862.2538908>
- 43 Tharp, A.L. Selecting the “right” programming language. In *Proceedings of the thirteenth SIGCSE technical symposium on Computer science education*. (Indianapolis, Indiana, United States). ACM Press, 1982, 151 – 155
- 44 Thonny Python IDE for beginners. Retrieved January 15, 2020, from <https://thonny.org/>
- 45 Udacity. *Introduction to Python Programming*. Διαθέσιμο: <https://www.udacity.com/course/introduction-to-python--ud1110>
- 46 VPython. Διαθέσιμο: <https://vpython.org/>
- 47 Wikizero. *Development Environments*. Διαθέσιμο: https://www.wikizero.com/en/Programming_environment (5 Φεβρουαρίου, 2020).
- 48 Wikipedia, *Source-code Editor*. Διαθέσιμο: https://en.wikipedia.org/wiki/Source-code_editor (20 Φεβρουαρίου, 2020).
- 49 Wikipedia, *POGIL*. Διαθέσιμο: <https://en.wikipedia.org/wiki/POGIL> (15 Μάϊου 2020).
- 50 Wikipedia, *Python for S60*. Διαθέσιμο: https://en.wikipedia.org/wiki/Python_for_S60 (20 Μάϊου 2020).
- 51 Wikipedia, *Scribbler (robot)*. Διαθέσιμο: [https://en.wikipedia.org/wiki/Scribbler_\(robot\)](https://en.wikipedia.org/wiki/Scribbler_(robot)) (5 Μάϊου 2020).
- 52 Wirth, N., The programming language Pascal. *Acta Informatica*, 1, 1971, 35 - 63.
- 53 Zahid Hussain, Muhammad Siyab Khan. Introducing Python Programming for Engineering Scholars. In *IJCSNS International Journal of Computer Science and Network Security*, VOL.18 No.12, December 2018.

- 54 Αγγελιδάκης Νικόλαος (2015). *Εισαγωγή στον προγραμματισμό με την Python*. Creative Commons BY-SA, ISBN: 978-960-93-7364-7
- 55 Ευρυπίδης Βραχνός, Μαρία Κατσένη (2017). *Δυσκολίες των μαθητών στην κατανόηση των τύπων δεδομένων και μεταβλητών της γλώσσας προγραμματισμού Python*. Ζάννειο Πειραματικό Γυμνάσιο Πειραιά, Γυμνάσιο Θρακομακεδόνων.
- 56 Μανής Γεώργιος (2015). *Εισαγωγή στον Προγραμματισμό με Αρωγό τη Γλώσσα Python*. ΣΕΑΒ, ΚΑΛΛΙΠΙΟΣ, ISBN: 978-960-603-415-2