



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ
ΣΠΟΥΔΩΝ
ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΗ ΒΙΟΙΑΤΡΙΚΗ

**«Ανάπτυξη εκπαιδευτικού υλικού για τη γλώσσα
προγραμματισμού Python με αξιοποίηση σεναρίων σε Scratch»**

Σόρκου Ευτυχία

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Υπεύθυνος

Κακαρούντας Αθανάσιος

Λαμία, 2019



UNIVERSITY OF THESSALY
SCHOOL OF SCIENCE
INFORMATICS AND COMPUTATIONAL BIOMEDICINE

**«Exploitation of Scratch scripts for the development of Python
educational material»**

Sorkou Eftychia

Master thesis

Kakarountas Athanasios

Lamia, 2019



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΔΙΑΤΜΗΜΑΤΙΚΟ ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΗ ΒΙΟΙΑΤΡΙΚΗ
ΚΑΤΕΥΘΥΝΣΗ

«ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ
ΕΠΙΚΟΙΝΩΝΙΩΝ ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ»

«Ανάπτυξη εκπαιδευτικού υλικού για τη γλώσσα
προγραμματισμού Python με αξιοποίηση σεναρίων σε Scratch»

Σόρκου Ευτυχία

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Επιβλέπων

Κακαρούντας Αθανάσιος

«Υπεύθυνη Δήλωση μη λογοκλοπής και ανάληψης προσωπικής ευθύνης»

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, και γνωρίζοντας τις συνέπειες της λογοκλοπής, δηλώνω υπεύθυνα και ενυπογράφως ότι η παρούσα εργασία με τίτλο «Ανάπτυξη εκπαιδευτικού υλικού για τη γλώσσα προγραμματισμού Python με αξιοποίηση σεναρίων σε Scratch» αποτελεί προϊόν αυστηρά προσωπικής εργασίας και όλες οι πηγές από τις οποίες χρησιμοποίησα δεδομένα, ιδέες, φράσεις, προτάσεις ή λέξεις, είτε επακριβώς (όπως υπάρχουν στο πρωτότυπο ή μεταφρασμένες) είτε με παράφραση, έχουν δηλωθεί κατάλληλα και ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο/Η ΔΗΛΩΝ/-ΟΥΣΑ

Ημερομηνία

Υπογραφή

Λαμία, 2019

**«Ανάπτυξη εκπαιδευτικού υλικού για τη γλώσσα
προγραμματισμού Python με αξιοποίηση σεναρίων σε Scratch»**

Σόρκου Ευτυχία

Τριμελής Επιτροπή:

Κακαρούντας Αθανάσιος

Λουκόπουλος Αθανάσιος

Δελήμπασης Κωνσταντίνος

Επιστημονικός Σύμβουλος:

Ονοματεπώνυμο.....

Περίληψη

Στις μέρες μας οι έννοιες προγραμματισμός και κώδικας είναι ευρέως διαδεδομένες και γνωστές. Ειδικότερα στην εκπαίδευση κατέχουν σημαντικό ρόλο. Για το λόγο αυτό οι μαθητές της Α΄θμιας εκπαίδευσης (Δημοτικό σχολείο) έρχονται σε επαφή με τον προγραμματισμό στις τελευταίες τάξεις του σχολείου.

Ο εκπαιδευτικός χρησιμοποιεί ένα ελκυστικό και εύχρηστο λογισμικό στη διδασκαλία του προγραμματισμού, έτσι ώστε να κεντρίσει έντονα το ενδιαφέρον και την προσοχή των μαθητών. Στις περισσότερες περιπτώσεις το λογισμικό που χρησιμοποιείται είναι το SCRATCH.

Η γλώσσα προγραμματισμού Scratch είναι μια γλώσσα δυναμικού προγραμματισμού για εκπαιδευτικούς σκοπούς που απευθύνονται κυρίως σε μαθητές δημοτικού. Στη συνέχεια της εκπαιδευτικής τους πορείας οι μαθητές έρχονται σε επαφή και με άλλες γλώσσες προγραμματισμού. Μία από αυτές είναι η Python.

Η γνώση μιας γλώσσας προγραμματισμού υψηλού επιπέδου είναι βασική προϋπόθεση για τον ψηφιακό εγγραμματισμό των μαθητών. Παρόλα αυτά η διδασκαλία μιας γλώσσας προγραμματισμού υψηλού επιπέδου έχει αρκετά σημεία δυσκολίας για τους μαθητές.

Η παρούσα εργασία παρουσιάζει μία πρόταση διδασκαλίας με στόχο την υποστήριξη της μετάβασης από τον προγραμματισμό στο Scratch, στον προγραμματισμό με τη γλώσσα Python, χρησιμοποιώντας το εκπαιδευτικό λογισμικό υπερμέσων, δηλαδή τη δημιουργία ιστολογίου.

Λέξεις κλειδιά: Προγραμματισμός, Scratch, Python, γλώσσα προγραμματισμού, διδασκαλία, λογισμικό, ιστολόγιο

Abstract

Nowadays, programming and code concepts are widespread and familiar to the most. In particular, they play an important role in education. For this reason pupils of primary school, are in contact with coding in the last classes of the school.

The teacher, uses, an attractive and easy-to-use software in programming teaching, so as to strongly stimulate the interest and attention of the students. In most cases the software used is SCRATCH.

The Scratch programming language is a dynamic programming language for educational purposes aimed primarily at primary school students. Continuing their educational pathway, students also come in contact with high-level programming languages. One of them is Python.

The knowledge of a high level programming language is a basic prerequisite for the digital literacy of students. However, teaching a high-level programming language has several difficulty points for students.

This paper presents a teaching suggestion to support the transition from Scratch programming to Python, using the hypermedia educational software, creating a site.

Key words: Programming, Scratch, Python, programming language, coding teaching, software.

Ευχαριστίες

Στο σημείο αυτό θα ήθελα να ευχαριστήσω όλους όσους στάθηκαν δίπλα μου, πρακτικά και συναισθηματικά, κατά τη διάρκεια της εκπόνησης της παρούσας εργασίας. Πρωτίστως τις θερμές μου ευχαριστίες δίνω στον κ. Κακαρούντα Αθανάσιο, αναπληρωτή καθηγητή και επιβλέποντα της εργασίας μου, για την καθοδήγηση του κατά τη διάρκεια της συνεργασίας μας για την επιτυχή διεκπεραίωση της εργασίας.

Στη συνέχεια θα ήθελα να ευχαριστήσω συνολικά όλους τους καθηγητές και διδάσκοντες, των οποίων τα μαθήματα παρακολούθησα κατά τη διάρκεια του μεταπτυχιακού προγράμματος, όπως επίσης και τους φοιτητές του τμήματος Πληροφορικής με εφαρμογές στην Βιοϊατρική, που συμμετείχαν στην αξιολόγηση του ιστολογίου.

Έπειτα, σειρά έχουν τα πρόσωπα της οικογένειάς μου. Αρχικά, ευχαριστώ θερμά τους γονείς μου, Βασίλη και Σοφία, για την αμέριστη βοήθειά τους όλο αυτό το διάστημα, και που χωρίς την παρουσία τους δεν θα ήταν εφικτή η προσπάθειά μου.

Στο τέλος, ευχαριστώ το σύζυγο μου Κώστα και τα δυο παιδιά μου, τον υιό μου Δημήτρη και την κόρη μου Σοφία, για την κατανόηση και την υπομονή τους κατά τη διάρκεια της φοίτησής μου στο μεταπτυχιακό πρόγραμμα.

Την παρούσα εργασία την αφιερώνω σε όλους με πολλή αγάπη!

Περιεχόμενα

Περίληψη	6
Abstract	7
Ευχαριστίες	8
1. Εισαγωγή.....	12
2. Η γλώσσα προγραμματισμού Scratch	13
2.1 Σύντομη ιστορία του Scratch	13
2.2 Θεωρητικό πλαίσιο πάνω στο οποίο βασίζεται το Scratch	14
2.2.1 Εποικοδομισμός	14
2.2.2 Οπτικός Προγραμματισμός	15
2.2.3 Ταυτόχρονος προγραμματισμός	15
2.3 Προγραμματισμός και Scratch	16
2.3.1 Ο Προγραμματισμός και η χρησιμότητά του.	16
2.3.2 Χαρακτηριστικά του Scratch	16
2.4 Περιγραφή του Scratch	19
3. Η γλώσσα προγραμματισμού Python	26
3.1 Σύντομη ιστορία της Python	26
3.2 Βασικά χαρακτηριστικά της Python	27
3.3 Η Python στην εκπαίδευση	29
3.4 Κατάταξη της Python σε σχέση με τις άλλες γλώσσες προγραμματισμού.....	32
4. Μεθοδολογία.....	36
4.1 Οι δυσκολίες στην εκμάθηση του προγραμματισμού.....	36
4.2 Εκπαιδευτικό λογισμικό	38
4.3 Μοντέλο θεωρίας μάθησης που χρησιμοποιήθηκε.....	39
4.4 Επισκόπηση των μελετών αξιολόγησης	41
4.4.1 Η Μελέτη των Mathrani, A., Christian, S., & Ponder-Sutton, A.(2016). PlayIT: Game Based Learning Approach for Teaching Programming Concepts. Educational Technology & Society, 19(2), 5-17.	41
4.4.2. Διπλωματική εργασία του Σιδέρη Γρηγόριου με θέμα: «Ανάπτυξη παιχνιδιού σοβαρού σκοπού για την εκμάθηση εννοιών προγραμματισμού με τη γλώσσα Python» (2019), Θεσσαλονίκη.....	42

4.4.3 Διπλωματική εργασία των Μαγκιτούρα Νορμπέρτα, Μουτζούρη Βασιλική, με θέμα «Συγκριτική Μελέτη Περιβαλλόντων και Πλατφορμών Για τη Διδασκαλία του Προγραμματισμού», Βόλος, 2010.....	43
4.4.4 Η ιστοσελίδα codeacademy ως εργαλείο για την εκμάθηση προγραμματισμού, Θεοφανέλλης Τιμολέων, 2016.....	44
4.5 Ενότητες προς εκμάθηση	45
4.5.1 Περιγραφή των εννοιών προς εκμάθηση στην Python.....	45
4.5.2 Πρόταση μετάβασης από το Scratch στην Python.	53
5. Υλοποίηση και παρουσίαση ιστολογίου.....	56
6. Αξιολόγηση.....	148
6.1 Στόχος της έρευνας	148
6.2 Μεταβλητές προς έρευνα.....	148
6.3 Περιγραφή δείγματος.....	149
6.4 Μέσο συλλογής δεδομένων	149
6.5 Ερωτήματα της έρευνας.....	152
6.6 Ερευνητικές υποθέσεις	153
6.7 Στατιστική Ανάλυση Δεδομένων.....	154
6.7.1 Περιγραφική Στατιστική.....	154
6.7.2 Στατιστικός έλεγχος υποθέσεων	159
6.7.3 Στατιστική Ανάλυση Ερευνητικών Ερωτημάτων.....	160
7. Συμπέρασμα.....	166
Παράρτημα.....	168
Ερωτηματολόγιο USE.....	168

1. Εισαγωγή

Τα τελευταία χρόνια στο δημοτικό σχολείο γίνεται προσπάθεια εισαγωγής των μαθητών στις βασικές έννοιες προγραμματισμού. Το βασικό λογισμικό για την επίτευξη του συγκεκριμένου στόχου είναι η γλώσσα προγραμματισμού Scratch. Στη συνέχεια της μαθητικής τους ζωής οι μαθητές έρχονται σε επαφή με γλώσσες προγραμματισμού υψηλού επιπέδου με χαρακτηριστικότερη την Python. Η μετάβαση από μια γλώσσα σαν το Scratch σε μια γλώσσα όπως η Python, μπορεί να επιτευχθεί με διάφορους και ποικίλους τρόπους. Στην παρούσα διπλωμακή εργασία προτείνεται ως τρόπος μετάβασης, η περιήγηση σε ένα ιστολόγιο που δημιουργήθηκε και περιέχει παραδείγματα σε Scratch και πως αυτά αντιστοιχούν σε προγράμματα Python, επιτελώντας την ίδια λειτουργία.

Στο δεύτερο κεφάλαιο αναλύεται η γλώσσα προγραμματισμού Scratch, η χρησιμότητά της, το θεωρητικό πλαίσιο στο οποίο βασίζεται και τα βασικά χαρακτηριστικά της. Στο τρίτο κεφάλαιο, παρουσιάζεται συνοπτικά η γλώσσα προγραμματισμού Python, αναφέρονται τα βασικά χαρακτηριστικά της, καθώς και τα πλεονεκτήματα και η δημοτικότητά της σε σχέση με άλλες γλώσσες προγραμματισμού.

Στη συνέχεια, στο τέταρτο κεφάλαιο, τονίζονται οι δυσκολίες στην εκμάθηση του προγραμματισμού και αναλύεται η τεχνολογία υπερμέσων που χρησιμοποιήθηκε για την δημιουργία του ιστολογίου, όπως και αναλύονται οι βασικές έννοιες προς εκμάθηση. Το πέμπτο κεφάλαιο, αναφέρει το λογισμικό που χρησιμοποιήθηκε για την κατασκευή του ιστολογίου και περιέχει κάποια ενδεικτικά στιγμιότυπα οθόνης για την παρουσίασή του.

Έπειτα ακολουθεί το έκτο κεφάλαιο, το οποίο αναφέρεται στην αξιολόγηση της μεθόδου μάθησης μέσω του ιστολογίου και στην ανάλυση των αποτελεσμάτων που προέκυψαν, επικεντρώνοντας σε τέσσερις βασικές κατηγορίες: την χρησιμότητα, την ευκολία στη μάθηση, την ευχρηστία και την ικανοποίηση, σύμφωνα με το ερωτηματολόγιο USE του Lund.

2. Η γλώσσα προγραμματισμού Scratch

Το Scratch¹ είναι μια δυναμική οπτική γλώσσα προγραμματισμού. Ο βασικός στόχος του είναι η εκμάθηση βασικών εννοιών προγραμματισμού και προορίζεται κυρίως για παιδιά, εφήβους και γενικότερα είναι κατάλληλη για αρχάριους προγραμματιστές. Βοηθά τα άτομα στις μικρότερες ηλικίες να δημιουργήσουν παιχνίδια, βίντεο και μουσική. Μεταφορτώνεται δωρεάν και η χρήση του ποικίλει σε δραστηριότητες εντός και εκτός σχολείου.

Το όνομα του παραπέμπει στην τεχνική του scratching στα παλαιά πικάπ. Το scratching στην μουσική είναι η επαναχρησιμοποίηση κομματιών. Ομοίως στο Scratch όλα τα αλληλεπιδραστικά αντικείμενα, γραφικά και ήχοι μπορούν εύκολα να γίνουν είσοδος σε νέο πρόγραμμα και να συνδυαστούν με νέους τρόπους. Με τον τρόπο αυτό οι αρχάριοι προγραμματιστές έχουν την δυνατότητα να λάβουν γρήγορα αποτελέσματα και αποκτούν επιπλέον κίνητρο για περαιτέρω προσπάθεια.

2.1 Σύντομη ιστορία του Scratch

Το εγχείρημα για τη δημιουργία του Scratch ξεκίνησε το 2003, ενώ το λογισμικό και η ιστοσελίδα που το υποστηρίζει έγιναν ευρέως γνωστά το 2007. Δημιουργήθηκε από την Lifelong Kindergarten Group του MIT Media Lab με πρωτεργάτη τον Mitchel Resnick. Ο αρχικός σχεδιασμός του Scratch υποκινήθηκε από τις ανάγκες και τα ενδιαφέροντα των παιδιών και εφήβων (ηλικίας 8 έως 16 ετών) οι οποίοι παρακολουθούσαν κέντρα υπολογιστών μετά το σχολείο [1]. Επομένως, το Scratch, αρχικά χρησιμοποιήθηκε κυρίως σε χώρους άτυπης μάθησης, όπως κοινοτικά κέντρα, συλλόγους μετά το σχολείο, βιβλιοθήκες και σπίτια. [1]

Το περιβάλλον του Scratch ενθάρρυνε τους νέους να δημιουργήσουν διαδραστικές ιστορίες, κινούμενα σχέδια, ηλεκτρονικά παιχνίδια, μουσική. Το προαναφερόμενο σε συνδυασμό με το γεγονός ότι η «κοινότητα» που έχει δημιουργηθεί γύρω από το Scratch, παρέχει την ευκαιρία της ανταλλαγής ιδεών και απόψεων μεταξύ των δημιουργών ώστε όλοι μαζί να εμπλέκονται ενεργά στην πρακτική και τη μάθηση (χωρίς να είναι απαραίτητη η άμεση διδασκαλία όπως συνέβαινε μέχρι εκείνη τη

¹ scratch.mit.edu

στιγμή με τις προυπάρχουσες γλώσσες προγραμματισμού) καθιστά το scratch καινοτόμο και ευχάριστο [2].

Διατίθεται δωρεάν μέσω της ιστοσελίδας του, στην οποία υπάρχουν περίπου 700.000 εγγεγραμμένα μέλη και περίπου 200.000 προγραμματιστές που δημοσιεύουν τα προγράμματα τους στον συγκεκριμένο ιστοχώρο².

2.2 Θεωρητικό πλαίσιο πάνω στο οποίο βασίζεται το Scratch.

Στην ενότητα αυτή παρουσιάζονται οι θεωρίες πάνω στις οποίες βασίζεται το Scratch.

2.2.1 Εποικοδομισμός

Το σύγχρονο παιδαγωγικό και γνωστικό πλαίσιο βασίζεται στη θεωρία του εποικοδομισμού. Σύμφωνα με τη θεωρία αυτή οι γνώσεις δεν μεταδίδονται αλλά οικοδομούνται. Ο εκπαιδευόμενος είναι υπεύθυνος για την οικοδόμηση της γνώσης βασιζόμενος σε προηγούμενες εμπειρίες, συμμετέχοντας στην διερεύνηση της πραγματικότητας και σε ομαδικές καταστάσεις όπου «ανθήζει» η συνεργασία και η επικοινωνία. Με τον τρόπο αυτό εμφανίζονται γνωστικές «συγκρούσεις» οι οποίες συμβάλλουν στην οικοδόμηση της γνώσης [3]. Σύμφωνα με τους [4], ο εποικοδομητισμός χαρακτηρίζεται από τα 3 C: “context”, “construction” και “collaboration”.

- **Context** (πλαίσιο προβλημάτων): τα προβλήματα θέτουν ένα πλαίσιο, δίνοντας στον εκπαιδευόμενο κίνητρο για εφαρμογή της γνώσης και εμπλοκή σε διαδικασίες αναστοχασμού και διερεύνησης.
- **Construction** (οικοδόμηση γνώσης): οι εκπαιδευόμενοι οικοδομούν μόνοι τους τη γνώση συμμετέχοντας σε δραστηριότητες μάθησης.
- **Collaboration** (συνεργασία): οι εκπαιδευόμενοι μπορούν να μάθουν μέσω της συνεργασίας και της αλληλεπίδρασης. Ο καθένας παρουσιάζει τις απόψεις του και τις γνώσεις του, ώστε να προκύψει η «νέα» άποψη, η οποία αποτελεί και τον προπομπό της λύσης του προβλήματος.

2 scratch.mit.edu

2.2.2 Οπτικός Προγραμματισμός

Ο οπτικός προγραμματισμός, χρησιμοποιεί γραφικά, κίνηση και εικόνες. Το Scratch αποτελεί γλώσσα οπτικού προγραμματισμού.

Τα βασικότερα πλεονεκτήματα του οπτικού προγραμματισμού είναι τα εξής [5]:

1. Χρησιμοποιεί δραστηριότητες από την καθημερινή ζωή και κάνει ευκολότερο τον προγραμματισμό.
2. Υποστηρίζει την μεταφορά προγραμμάτων, μιας και τα προγράμματα που δημιουργούνται σε τέτοιου είδους περιβάλλοντα, μπορούν και εκτελούνται σε οποιοδήποτε υπολογιστή χωρίς να είναι απαραίτητη η εγκατάσταση του περιβάλλοντος προγραμματισμού.
3. Συσχετίζει εύκολα το γραφικό με το πραγματικό νόημα των εντολών.
4. Ο χρήστης αποφεύγει τα συντακτικά λάθη, αφού τα δομικά κομμάτια των γλωσσών οπτικού προγραμματισμού έχουν ειδικές υποδοχές να ενώνονται μεταξύ τους.

2.2.3 Ταυτόχρονος προγραμματισμός

Ο ταυτόχρονος προγραμματισμός χρησιμοποιεί ακολουθιακά τμήματα προγραμμάτων που εκτελούνται παράλληλα [6]. Για τα ακολουθιακά τμήματα προγραμμάτων, τα οποία συνθέτουν την ολότητα του προγράμματος, χρησιμοποιείται ο όρος διεργασία. Το πρόγραμμα αποτελεί το σύνολο των διεργασιών. Η διδασκαλία του ταυτόχρονου προγραμματισμού βοηθά στην επίτευξη των στόχων της εκπαίδευσης στην επιστήμη των υπολογιστών, επεκτείνοντας ταυτόχρονα κάποιες από τις βασικές αρχές της [7].

Θεωρώντας πως αυτές οι γλώσσες είναι πιο «φυσικές» για τον εκπαιδευόμενο, επειδή στην πραγματική ζωή ανεξάρτητες υπάρξεις λειτουργούν ταυτόχρονα και επικοινωνούν μεταξύ τους, έχουν υπάρξει σημαντικές προτάσεις για χρήση των ταυτόχρονων γλωσσών στην πρώτη επαφή των εκπαιδευομένων με τον προγραμματισμό [8].

2.3 Προγραμματισμός και Scratch

2.3.1 Ο Προγραμματισμός και η χρησιμότητά του.

Σύμφωνα με [9] η υπολογιστική σκέψη είναι οι διαδικασίες σκέψης που αφορούν στην διατύπωση ενός προβλήματος και στην περιγραφή της λύσης του με τέτοιο τρόπο ώστε αυτές να είναι κατανοητές από άνθρωπο και υπολογιστή. Η υπολογιστική σκέψη είναι άρρηκτα δεμένη με τον προγραμματισμό υπολογιστών.

Ο τρόπος διδασκαλίας του προγραμματισμού αποτελεί ένα πολύ σημαντικό αντικείμενο μελέτης. Η δραστηριότητα του προγραμματισμού θεωρείται διαδικασία επίλυσης προβλήματος. Ο τρόπος για να επιτευχθεί η επίλυση ενός προβλήματος εδράζεται στην ανάπτυξη στρατηγικών και στην εγκαθίδρυση διαδικασιών που επιτρέπουν τη σύνδεση ανάμεσα στην αρχική και τελική κατάσταση [10].

Επομένως βασικός και πρωταρχικός στόχος της διδασκαλίας του προγραμματισμού πρέπει να είναι η ανάπτυξη της ικανότητας των εκπαιδευομένων να χρησιμοποιούν και να εφαρμόζουν τις γνώσεις που έχουν κατακτήσει ώστε να επιλύσουν προβλήματα που δεν έχουν διδαχθεί σε προηγούμενο χρόνο [11].

Ο προγραμματισμός αποτελεί μια δραστηριότητα η οποία εμφανίστηκε μαζί με την Πληροφορική και δεν υπήρχε πριν από αυτή. Επίσης δεν θυμίζει κάποια άλλη δραστηριότητα του ανθρώπου. Ο προγραμματισμός υπολογιστών θεωρείται ένας τρόπος για την καλλιέργεια και ανάπτυξη νοητικών δεξιοτήτων υψηλού επιπέδου, θυμίζοντας μια νέα μεθοδολογία της σκέψης.

«Η μάθηση του προγραμματισμού μπορεί να οδηγήσει στην εγκαθίδρυση νέων τρόπων αντίληψης, μοντελοποίησης και επεξεργασίας προβλημάτων και για το λόγο αυτό συνιστά ένα εξαιρετικά ενδιαφέρον πεδίο στο πρόγραμμα σπουδών που δεν προσεγγίζεται από κανένα άλλο γνωστικό αντικείμενο στο χώρο της δευτεροβάθμιας εκπαίδευσης», [10].

2.3.2 Χαρακτηριστικά του Scratch

Βασικό χαρακτηριστικό της γλώσσας προγραμματισμού Scratch είναι ότι η δημιουργία προγραμμάτων με αυτό δεν είναι μια βαρετή και δύσκολη διαδικασία, αντίθετα προκαλεί το ενδιαφέρον και γεννά κίνητρα για επιπλέον ενασχόληση. Το Scratch δεν είναι γλώσσα ακολουθιακού προγραμματισμού αλλά γλώσσα

ταυτόχρονου προγραμματισμού [12]. Ταυτόχρονο πρόγραμμα ονομάζεται ένα σύνολο από ακολουθιακά τμήματα προγραμμάτων που εκτελούνται παράλληλα [6].

Επιπλέον τα παιδιά και οι έφηβοι που χρησιμοποιούν το Scratch για εκμάθηση, έρχονται από νωρίς σε επαφή με σημαντικές και βασικές έννοιες διαφόρων επιστημών όπως των μαθηματικών και της φυσικής [13].

Σημαντικό χαρακτηριστικό του Scratch είναι, ότι υποστηρίζει στο περιβάλλον του την ελληνική γλώσσα, κάτι το οποίο είναι ιδιαίτερο βοηθητικό για τους αρχάριους προγραμματιστές και τα άτομα μικρής ηλικίας(δημοτικό σχολείο).

Ο σχεδιασμός και η λογική του Scratch βασίζονται πρωταρχικά στη δημιουργία σεναρίων (scripts) χρησιμοποιώντας τη μέθοδο «σύρε και άσε» (drag and drop), για όλους τους χαρακτήρες (sprites) που βρίσκονται πάνω σε μία σκηνή (background). Δίνεται η δυνατότητα να χρησιμοποιηθούν οι βασικές Δομές που υπάρχουν σε όλες τις γλώσσες προγραμματισμού (Δομή ακολουθίας, Δομή Επιλογής και Δομή Επανάληψης). Επιπλέον των βασικών δομών ελέγχου, η γλώσσα προγραμματισμού Scratch παρέχει πυροδότες γεγονότων (when-clicked, when-key-pressed).

Η καινοτομία λοιπόν του Scratch βασίζεται στο γεγονός ότι ο εκπαιδευόμενος δεν απαιτείται να πληκτρολογήσει εντολές για να δημιουργήσει το πρόγραμμά του. Οι εντολές είναι σε στυλ «μπλοκ-πλακιδίων», έτσι ο εκπαιδευόμενος θα πρέπει απλώς να συναρμολογήσει τα πλακίδια μεταξύ τους. Τα πλακίδια έχουν διαφορετικά σχήματα και χρώματα, τα οποία αφορούν στην εργασία που επιτελεί το καθένα.

Τα συντακτικά λάθη ουσιαστικά εξαλείφονται επειδή τα πλακίδια έχουν τέτοιες υποδοχές οι οποίες δεν επιτρέπουν την ένωση σε κομμάτια που δεν είναι συμβατά και θα προκαλούσαν συντακτικό λάθος Αυτό έχει ως αποτέλεσμα να μην είναι αναγκαία η γνώση του συντακτικού της γλώσσας (προαπαιτούμενο για τις υπόλοιπες γλώσσες προγραμματισμού), γιατί τα προγράμματα στο Scratch δομούνται όπως ένα παζλ, όπου η ένωση λάθος κομματιών είναι αδύνατη.

Το μόνο που θα πρέπει να κάνει ο εκπαιδευόμενος είναι να σύρει μπλοκ (κομμάτια κώδικα) από το αριστερό μέρος της οθόνης και να τα «κολλήσει» με όποιον τρόπο θέλει στο μέσον της, ώστε ο κώδικας-σενάριο που θα προκύψει να έχει το επιθυμητό αποτέλεσμα.

Σύμφωνα με τους [14], στο άρθρο τους «Teaching the TV Generation», η διδασκαλία και τελικά η μάθηση γίνονται καλύτερες όταν ο τρόπος της διδασκαλίας είναι ανάλογος με τον τρόπο της μάθησης. Πρότειναν λοιπόν, οι εκπαιδευόμενοι που μεγαλώνουν σε ένα περιβάλλον που κατακλύζεται από εικόνες καλό θα είναι να μη διδάσκονται μόνο με λεκτικό τρόπο (παραδοσιακή διδασκαλία).

Έχοντας υπόψη λοιπόν την προηγούμενη πρόταση, καταλήγουμε ότι το Scratch είναι θελκτικό και ενδιαφέρον διότι το παραγόμενο πρόγραμμα-σενάριο παραμένει «ζωντανό» κατά τη διάρκεια της εκτέλεσης, επομένως δίνεται η δυνατότητα στον εκπαιδευόμενο να επιφέρει αλλαγές σε αυτό και να παρατηρήσει άμεσα τα αποτελέσματα.

Σημαντικό είναι όμως και το γεγονός ότι το περιβάλλον Scratch δίνει την νέα δυνατότητα στον εκπαιδευτή να εισάγει τους εκπαιδευόμενους στην έννοια του προγραμματισμού αλλά και σε πολυπλοκότερες έννοιες, δημιουργώντας εύκολα διαδραστικό εκπαιδευτικό υλικό.

Δύο από τα σημαντικότερα χαρακτηριστικά του είναι ότι [5]:

- I. **Ποικιλομορφία:** υποστήριξη αρκετών και διαφορετικών τύπων εργασιών, όπως: ιστορίες, παιχνίδια, κινούμενα γραφικά και προσομοιώσεις. Αυτό έχει ως απώτερο στόχο ο εκπαιδευόμενος να δημιουργεί εργασίες ανάλογα με τα εκάστοτε ενδιαφέροντά του.
- II. **Οικειοποίηση:** δυνατότητα εισαγωγής φωτογραφιών, ήχου (μουσικής αλλά και ηχογραφημένης φωνής) και γραφικών. Με αυτό επιτυγχάνεται ο εκπαιδευόμενος να αισθάνεται την εργασία του περισσότερο οικεία, σαν να είναι «κομμάτι» του εαυτού του.

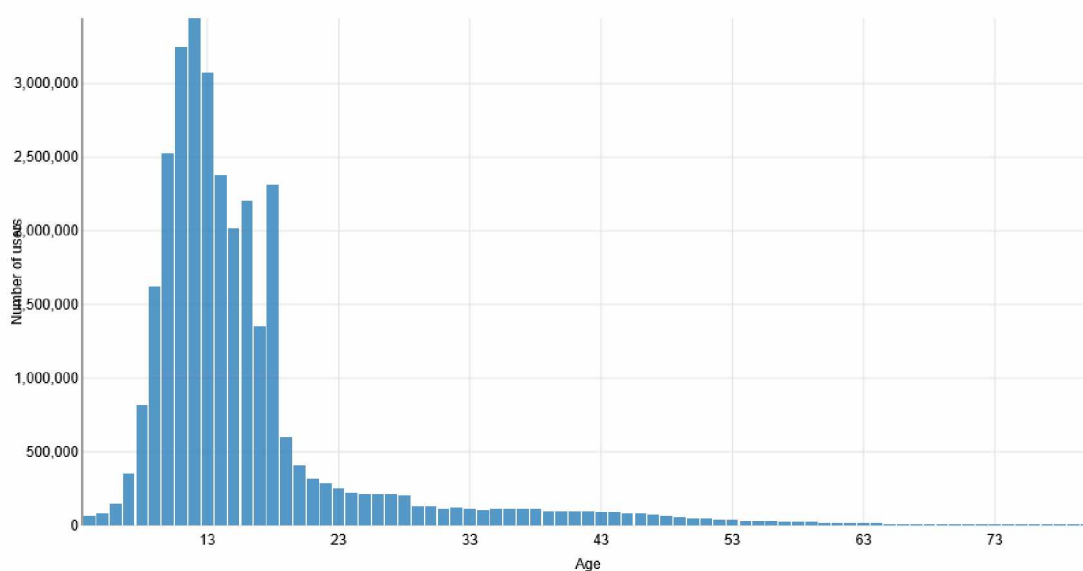
Σημαντικό είναι επίσης ότι το Scratch διαχειρίζεται προγράμματα τα οποία περιλαμβάνουν χρήση εικόνων, ήχου, βίντεο και μουσικής, γεγονός που το κάνει αυτόματα περισσότερο ενδιαφέρον στις μικρότερες ηλικίες. Αυτό αποτελεί σημαντική διαφορά με τον κλασικό προγραμματισμό που αφορά χειρισμό αριθμών και κειμένου.

Επιπλέον αξίζει να σημειωθεί ότι οι δημιουργοί του Scratch βασίστηκαν σε πρότερη εμπειρία, όπως το LEGO Mindstorms, και το σχεδίασαν με τρόπο τέτοιο ώστε οι

χρήστες του να μπορούν να προγραμματίσουν πραγματικά αντικείμενα, όπως προγραμματίζουν εικονικά αντικείμενα στο περιβάλλον του.

Όπως τελικά μπορούμε να κατανοήσουμε, σύμφωνα με τα προαναφερόμενα, η γλώσσα προγραμματισμού Scratch είναι κατάλληλα διαμορφωμένη ώστε να απευθύνεται σε παιδιά επειδή είναι ελκυστική, όπως και γιατί η σύνταξη και η σημασιολογία της είναι μικρή και απλή. Με τον τρόπο αυτό επιτυγχάνεται η μεταφορά της λύσης που έχουν στο μυαλό τους, σε πληροφορία με τη μορφή εντολών, που θα είναι κατανοητές και εκτελέσιμες από τον υπολογιστή.

Στην εικόνα 2.1 ένα γράφημα που απεικονίζει τις ηλικίες των νέων χρηστών του Scratch (από : <https://scratch.mit.edu/statistics/>).

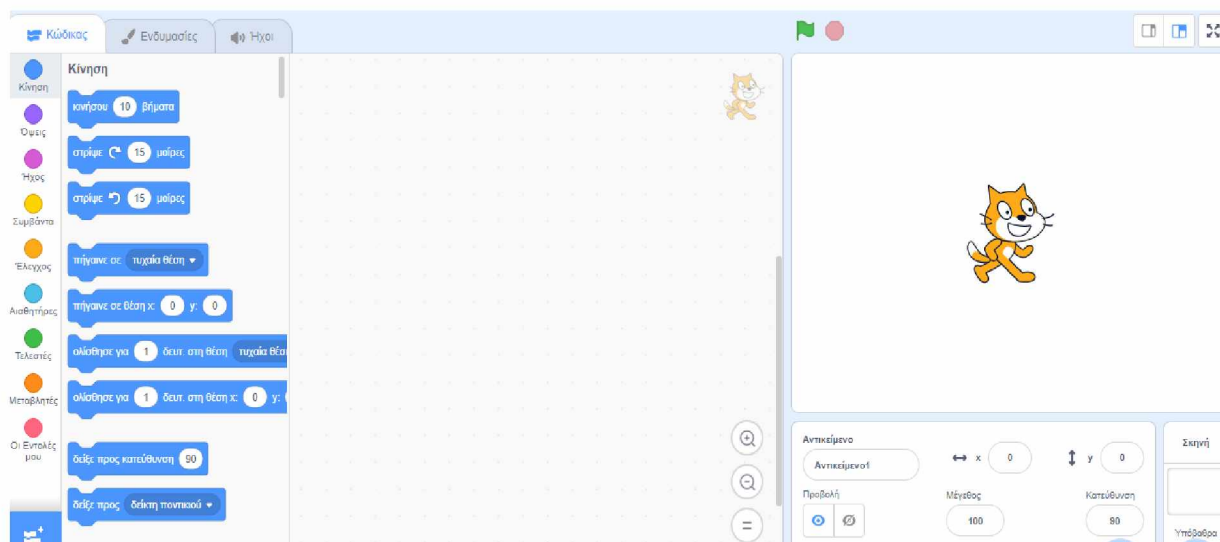


Εικόνα 2.1 : Ηλικίες νέων χρηστών του Scratch

Μόλις οι εκπαιδευόμενοι αποκτήσουν αρκετή εμπειρία στο γραφικό περιβάλλον του Scratch, μπορούν με ευκολία να συνεχίσουν με γλώσσες υψηλότερου επιπέδου, όπως η Python. Αυτό ήδη συμβαίνει στο ελληνικό σχολείο, στον τομέα της πληροφορικής της Γ΄ τάξης του Επαγγελματικού Λυκείου [15].

2.4 Περιγραφή του Scratch

Στην εικόνα 2.2 βλέπουμε ένα στιγμιότυπο οθόνης του περιβάλλοντος Scratch.



Εικόνα 2.2: Στιγμιότυπο οθόνης του Scratch

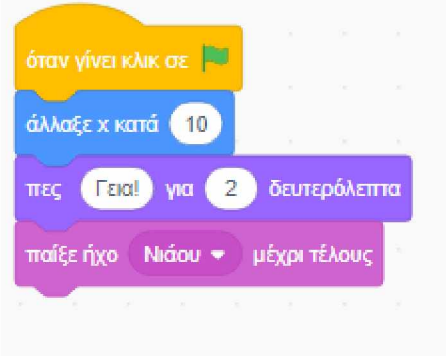
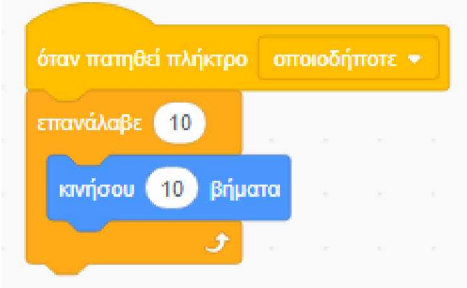
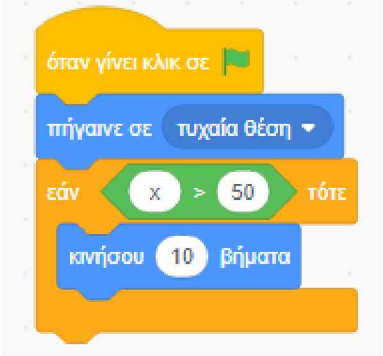
Η σύλληψη της ιδέας για τη δημιουργία του Scratch επηρεάστηκε από τον τρόπο με τον οποίο τα παιδιά παίζουν και δημιουργούν με τα τουβλάκια Lego. Βασισμένο σε αυτό λοιπόν το Scratch χρησιμοποιεί δομικούς λίθους (blocks), οι οποίοι συνδυάζονται και δημιουργούν τα σεναρία-προγράμματα. Όπως προαναφέρθηκε τα blocks ενώνονται μόνο στην περίπτωση που υπάρχει συντακτικό νόημα στο πρόγραμμα, (βασικός λόγος που οι αρχάριοι προγραμματιστές αποφεύγουν τα συντακτικά λάθη).


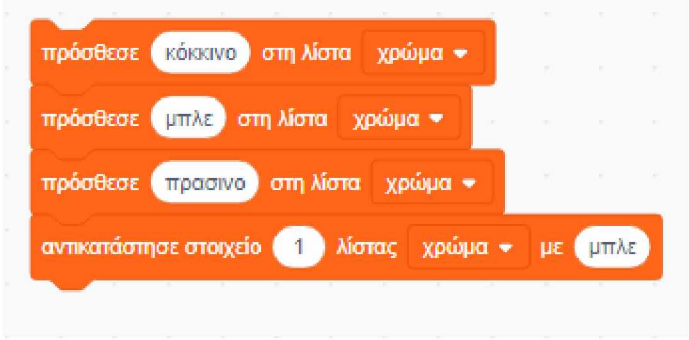
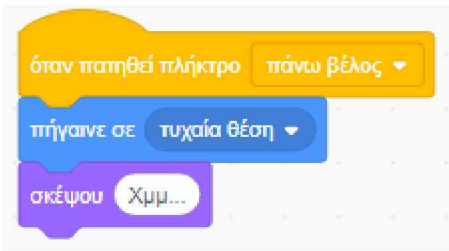
Επιπλέον τα blocks ελέγχου, έχουν ένα C σχήμα που βοηθούν τον χρήστη να κατανοήσει ότι μπορεί να τοποθετήσει άλλα blocks μέσα σε αυτά. Επίσης τα blocks που επιστρέφουν τιμές έχουν σχήμα ανάλογο με την τιμή την οποία επιστρέφουν, δηλαδή οβάλ για τα blocks που επιστρέφουν αριθμητικές τιμές και εξαγωνικά για τα blocks που επιστρέφουν Boolean τιμές.

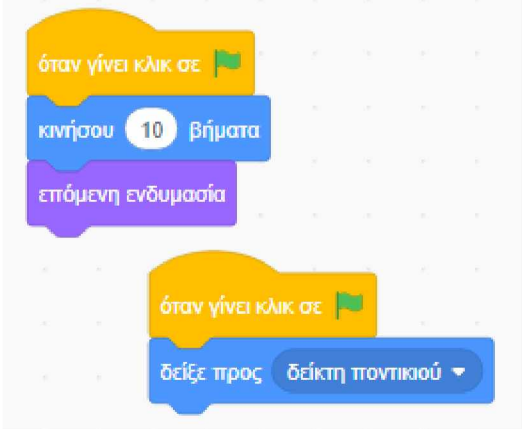
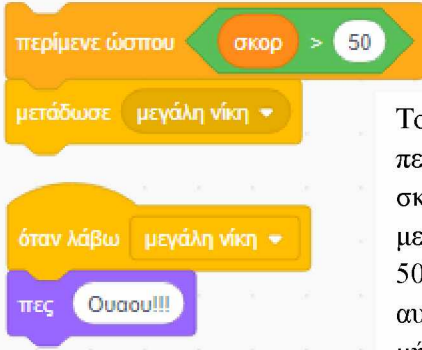
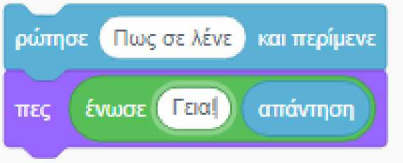
Η εκτέλεση του προγράμματος στο Scratch ξεκινά κάνοντας διπλό κλικ πάνω σε μια στοιβία από blocks. Λόγω του ότι είναι λογισμικό ταυτόχρονου προγραμματισμού, υποστηρίζει παράλληλη εκτέλεση διαφορετικών στοιβών από blocks.

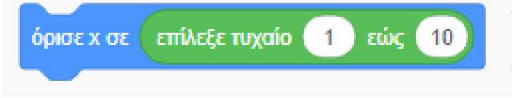
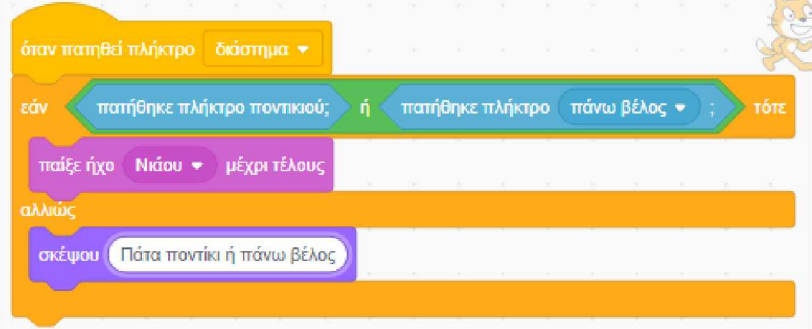
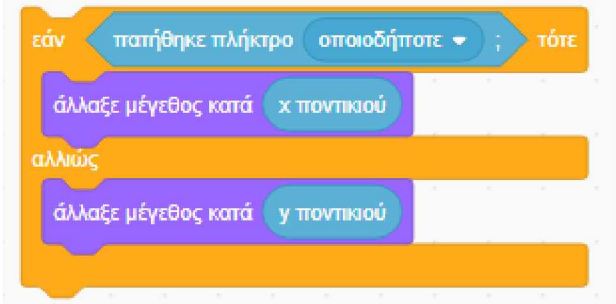
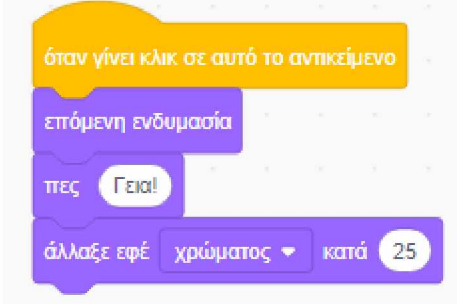
Στη συνέχεια παρουσιάζονται μερικές προγραμματιστικές βασικές έννοιες του Scratch, όπως επίσης και ένα παράδειγμα της καθεμιάς.

Πίνακας 2.1-Προγραμματιστικές έννοιες

<i>Έννοια</i>	<i>Ερμηνεία</i>	<i>Παράδειγμα</i>
Δομή ακολουθίας	Δημιουργώντας πρόγραμμα πρέπει να μελετήσουμε τη σειρά των βημάτων που θα ακολουθηθούν (αλγόριθμος)	
Δομή επανάληψης	Οι εντολές για πάντα επανέλαβε και επανέλαβε ώσπου χρησιμοποιούνται ως βρόγχοι επανάληψης	
Δομή ελέγχου	Οι εντολές εάν ... τότε, εάν τότε ...αλλιώς και περίμενε ώσπου, ελέγχουν αν ικανοποιείται μια συνθήκη	

<p>Μεταβλητές</p>	<p>Υπάρχουν οι κατάλληλες εντολές για δημιουργία και χρήση μεταβλητών. Οι μεταβλητές μπορούν να αποθηκεύσουν αριθμούς και αλφαριθμητικά.</p>	
<p>Λίστες</p>	<p>Υπάρχουν εντολές για αποθήκευση σε λίστες αριθμών και αλφαριθμητικών.</p>	
<p>Γεγονότα</p>	<p>Οι εντολές όταν το πλήκτρο πατηθεί και όταν γίνει κλικ, είναι παραδείγματα εντολών διαχείρισης γεγονότων και εκτελούνται από πρόκληση του χρήστη ή από άλλο τμήμα του προγράμματος</p>	

<p>Παράλληλη εκτέλεση</p>	<p>Η ταυτόχρονη μεταπήδηση σε 2 διαφορετικά σενάρια δημιουργεί δυο παράλληλες εκτελέσεις.</p>	
<p>Συγχρονισμός και συνεργασία</p>	<p>Οι εντολές μετάδωσε και όταν λάβω υποστηρίζουν τη συνεργασία μεταξύ των μορφών. Η εντολή μετάδωσε και περίμενε επιτρέπει συγχρονισμό</p>	 <p>Το σενάριο περιμένει μέχρι το σκορ γίνει μεγαλύτερο από το 50. Μόλις γίνει αυτό, μεταδίδει το μήνυμα μεγάλη νίκη. Όταν ληφθεί το μεγάλη νίκη, το αντικείμενο λέει Ουασου!!!</p>
<p>Ανάγνωση από το πληκτρολόγιο</p>	<p>Η εντολή ρώτησε και περίμενε καλεί τον χρήστη να δώσει μια απάντηση, η οποία αποθηκεύεται στην εντολή απάντηση</p>	

<p>Τυχαίοι αριθμοί</p>	<p>Η εντολή επίλεξε τυχαία επιστρέφει τυχαίους ακέραιους που ανήκουν σε ορισμένο διάστημα τιμών.</p>	
<p>Boole</p>	<p>Οι εντολές και, ή και όχι είναι παραδείγματα Boolean λογικής</p>	
<p>Δυναμική αλληλεπίδραση</p>	<p>Με τις εντολές ποντίκι x, ποντίκι y και ένταση μπορούμε να δώσουμε δυναμική είσοδο για αλληλεπίδραση πραγματικού χρόνου</p>	
<p>Σχεδίαση επιφάνειας χρήστη</p>	<p>Το Scratch υποστηρίζει σχεδίαση διαδραστικής επιφάνειας χρήστη</p>	

Από τα προηγούμενα κατανοούμε ότι το Scratch υποστηρίζει μια πληθώρα προγραμματιστικών εννοιών. Παρά το γεγονός αυτό υπάρχουν και κάποιες έννοιες οι οποίες δεν υποστηρίζονται ακόμη από το Scratch.

- Συναρτήσεις, ρουτίνες
- Αναδρομικοί αλγόριθμοι
- Επιστροφή τιμών
- Κλάσεις αντικειμένων
- Είσοδος/έξοδος αρχείων
- Κληρονομικότητα

3. Η γλώσσα προγραμματισμού Python

Η Python είναι μια γλώσσα προγραμματισμού γενικής χρήσης, υψηλού επιπέδου, απλή και εύκολη στην εκμάθηση, δυναμική, αποδοτική, παραγωγική και επεκτάσιμη. Είναι χρήσιμη για αρχάριους αλλά και για περισσότερο έμπειρους προγραμματιστές. Η χρήση της είναι σημαντική στην εκπαίδευση αλλά και στην ανάπτυξη ολοκληρωμένων εφαρμογών [16].

3.1 Σύντομη ιστορία της Python

Το 1989 ο ολλανδός προγραμματιστής Guido van Rossum, συνέταξε έναν διερμηνευτή για μια νέα γλώσσα προγραμματισμού την οποία ονόμασε Python. Για περισσότερες λεπτομέρειες μπορείτε να δείτε στο διαδίκτυο³.

Ο δημιουργός της νέας γλώσσας βάσισε τον σχεδιασμό στην λειτουργία της εικονικής μηχανής. Όταν σε μια γλώσσα προγραμματισμού χρησιμοποιείται εικονική μηχανή, η γλώσσα αυτή και τα προγράμματά της αποκτούν τη δυνατότητα της φορητότητας σε όλες τις πλατφόρμες.

Σε παγκόσμιο επίπεδο, οι προγραμματιστές έχουν υποδεχθεί θερμα την Python γράφοντας πολλά προγράμματα σε αυτή και αναπτύσσοντας τις βιβλιοθήκες της. Η δημοτικότητά της οφείλεται στην φορητότητα της και στη δυνατότητα που κατέχει να προσφέρει βιβλιοθήκες και σε άλλους. Επιπλέον η Python είναι μια γλώσσα προγραμματισμού που χαρακτηρίζεται από επεκτασιμότητα. Η επεκτασιμότητα είναι η ικανότητα προσδιορισμού των βιβλιοθηκών των κλάσεων για την διαχείριση και τελικά επίλυση προβλημάτων σε ποικίλους τομείς. Η Python βρίσκει εφαρμογή στον προγραμματισμό του διαδικτύου, στη δημιουργία σεναρίων για διακομιστές, στα γραφικά υπολογιστών, στα μαθηματικά και σε διάφορους επιπλέον τομείς.

Στις μέρες μας οι προγραμματιστές που γνωρίζουν Python έχουν αυξήσει σημαντικά τον αριθμό τους. Η Python έχει δυο εκδόσεις, την Python2 και την Python3. Οι διαφορετικές αυτές εκδόσεις δεν είναι συμβατές μεταξύ τους. Το γεγονός αυτό έδωσε στους προγραμματιστές την αφορμή να την βελτιστοποιήσουν [17].

³ <http://python-history.blogspot.com>

3.2 Βασικά χαρακτηριστικά της Python

Όπως προαναφέρθηκε η Python είναι μια ισχυρή και εύκολη στην εκμάθηση γλώσσα προγραμματισμού. Το συντακτικό της (που θυμίζει φυσική γλώσσα) και οι δυναμικοί τύποι της καθώς και το γεγονός ότι είναι διερμηνευόμενη την κατατάσσουν στις ιδανικότερες γλώσσες προγραμματισμού για ανάπτυξη εφαρμογών σε ποικίλους τομείς. Στη συνέχεια αναφέρονται αναλυτικά τα βασικά της χαρακτηριστικά σύμφωνα με το [18].

- ❖ *Εύκολη στην εκμάθηση*: Το συντακτικό της είναι εντυπωσιακά απλό.
- ❖ *Απλή*: Η ομοιότητά της με ψευδοκώδικα αποτελεί το κύριο πλεονέκτημά της.
- ❖ *Ελεύθερη και Ανοικτού Κώδικα*: Αποτελεί παράδειγμα ΕΛΛΑΚ (Ελεύθερο Λογισμικό και Λογισμικό Ανοικτού Κώδικα). Το ΕΛΛΑΚ είναι βασισμένο στην ιδέα μιας κοινότητας για ελεύθερο διαμοιρασμό της γλώσσας. Αυτό έχει σαν αποτέλεσμα να παρέχεται η δυνατότητα της διανομής αντιγράφων του λογισμικού, ανάγνωσης του πηγαίου κώδικα όπως και αλλαγές σε αυτόν, καθώς και χρήσης διάφορων στοιχείων του για τη δημιουργία νέων προγραμμάτων.
- ❖ *Γλώσσα υψηλού επιπέδου*: Κατά τη δημιουργία των προγραμμάτων δεν είναι αναγκαία η γνώση της λειτουργίας της γλώσσας μηχανής.
- ❖ *Φορητή*: Πολλές πλατφόρμες την υποστηρίζουν χωρίς να χρειάζονται αλλαγές στα προγράμματά της ώστε αυτά να εκτελεστούν.
- ❖ *Διερμηνευόμενη*: Σε μερικές μεταγλωττιζόμενες γλώσσες προγραμματισμού όταν τις χρησιμοποιούμε για την δημιουργία προγράμματος αυτό μετετρέπεται από την πηγαία γλώσσα σε μια γλώσσα μηχανής (δυναμικός κώδικας 0 και 1) χρησιμοποιώντας ένα μεταγλωττιστή. Ο συνδέτης στη συνέχεια εκτελεί το πρόγραμμα αφού προηγουμένως το έχει γράψει στη μνήμη. Σε αντίθεση με τα προαναφερόμενα, η Python δεν χρειάζεται μεταγλώττιση σε δυαδικό αρχείο. Αυτό που κάνει είναι να εκτελεί το πρόγραμμα απ' ευθείας απο τον πηγαίο κώδικα. Η Python

μετατρέπει τον πηγαίο κώδικα σε μια ενδιάμεση μορφή που ονομάζεται bytecode και μετά το μεταφράζει σε γλώσσα μηχανής, πριν το εκτελέσει.

- ❖ *Αντικειμενοστρεφής*: Μπορεί με ευκολία να υποστηρίξει τον διαδικασιοστρεφή προγραμματισμό αλλά και τον αντικειμενοστρεφή. Ο διαδικασιοστρεφής προγραμματισμός έχει το γνώρισμα ότι το πρόγραμμα βασίζεται πάνω σε διαδικασίες και συναρτήσεις, οι οποίες αποτελούν επαναχρησιμοποιήσιμα κομμάτια από προγράμματα. Ο αντικειμενοστρεφής προγραμματισμός χαρακτηρίζεται από το γεγονός ότι τα προγράμματα δομούνται πάνω σε αντικείμενα.
- ❖ *Επεκτάσιμη*: Ένα πρόγραμμα εκτελείται γρήγορα αν πρώτα γραφτεί σε C ή C++ και μετά μεταφερθεί και χρησιμοποιηθεί από την Python.
- ❖ *Ενσωματώσιμη*: Μπορεί να ενσωματωθεί στα προγράμματα που είναι γραμμένα σε C και C++, ώστε να δίνεται η δυνατότητα scripting στους χρήστες.
- ❖ *Εκτεταμένες βιβλιοθήκες*: Η ποικιλία και το μέγεθος της βιβλιοθήκης της Python, βοηθά αποτελεσματικά σε θέματα που άπτονται σε: κανονικές εκφράσεις, δημιουργία τεκμηρίωσης, δοκιμές μονάδων, νημάτωση, βάσεις δεδομένων, περιηγητές ιστού, CGI, FTP, email, XML, XML-RPC, HTML, αρχεία WAN, κρυπτογράφηση, γραφικές διεπαφές χρήστη και άλλα.
- ❖ *Χρόνος εκτέλεσης*: Το γεγονός ότι ένα πρόγραμμα σε Python δεν μεταγλωττίζεται σε δυαδικό κώδικα μηχανής ο οποίος εκτελείται άμεσα από τον υπολογιστή, κάνει τον χρόνο εκτέλεσης του αρκετά μεγαλύτερο απ' ότι στις μεταγλωττιζόμενες γλώσσες προγραμματισμού. Αυτό λοιπόν που αρχικά φαίνεται σαν μειονέκτημα τελικά δεν είναι διότι είναι σημαντικότερο το γεγονός της εξοικονόμησης χρόνου που συμβαίνει κατά την ανάπτυξη μιας εφαρμογής σε Python [16]. Ένα σημαντικό εργαλείο βελτιστοποίησης της εκτέλεσης Python προγραμμάτων είναι το σύστημα Cython. Το σύστημα αυτό είναι ουσιαστικά μια υβριδική γλώσσα η οποία συνδυάζει τον κώδικα Python έχοντας τη δυνατότητα κλήσης λειτουργιών C, όπως επίσης και τη χρήση δηλώσεων τύπου C για μεταβλητές, παραμέτρους και χαρακτηριστικά κλάσης. Η Cython μπορεί να αποδειχθεί

αρκετά χρήσιμη στη χρήση εξωτερικών βιβλιοθηκών της C όπως και στην κωδικοποίηση των αποτελεσματικών επεκτάσεων C για την Python [19].

3.3 Η Python στην εκπαίδευση

Στην ενότητα αυτή θα περιγράψουμε τους λόγους, σύμφωνα με τον [20], για τους οποίους η γλώσσα προγραμματισμού Python θεωρείται εξαιρετική επιλογή για διδασκαλία σε αρχάριους προγραμματιστές.

- ❖ **Δεν είναι αναγκαίο να δηλωθούν οι τύποι των μεταβλητών:** Όταν ένας εκπαιδευόμενος αρχικά μαθαίνει να προγραμματίζει σε μια γλώσσα όπως η C++ ή η Java, συνήθως οι πρώτες γραμμές που πρέπει να γράψει είναι οι δηλώσεις των μεταβλητών του προγράμματος. Αυτές οι γραμμές κώδικα δεν κάνουν τίποτα άλλο εκτός από το να χρησιμεύουν ως εικονική «σύγχυση» και επισκιάζει τις εντολές που εκτελούν πραγματικές ενέργειες. Όταν οι αρχάριοι προγραμματιστές γράφουν ένα πρόγραμμα, οι δηλώσεις των μεταβλητών καταλαμβάνουν το 1/3 ως το 1/4 όλων των εντολών του κώδικα. Οι έμπειροι προγραμματιστές θεωρούν ότι η διαδικασία της δήλωσης των μεταβλητών είναι δευτερεύουσας σημασίας.
- ❖ **Λάθη τύπου run-time αντί για τύπου compile-time:** Τα λάθη τύπου compile-time είναι ενοχλητικά για τους εκπαιδευόμενους λόγω του αδιεξόδου που δημιουργούν. Είναι προτιμότερο για αυτούς να αφήνουν να «τρέχει» το πρόγραμμά τους και να εκτελεί κάποιες λειτουργίες απ' το να αποτυγχάνει να μεταγλωττίσει και να μην κάνει απολύτως τίποτα. Είναι απογοητευτικό να είσαι αρχάριος που θέλει απλώς τον κώδικα του να τρέχει και να κάνει κάτι (ακόμα και αν δεν συμπεριφέρεται τέλεια), μόνο για να διαπιστώσει ότι αυτός ο μυστηριώδης «τύραννος» ονομάζεται compiler αρνείται να μετατρέψει τον κώδικα του σε ένα εκτελέσιμο πρόγραμμα.
- ❖ **Η εντολή print είναι πολυμορφική και αυτόματα εισάγει κενά και νέες γραμμές.**

Οι αρχάριοι προγραμματιστές θα χρησιμοποιούν συχνά την εντολή print, τόσο για την δημιουργία του προγράμματος όσο και σαν βοήθεια για τον εντοπισμό πιθανών σφαλμάτων. Αντί να είναι αναγκαίο οι εκπαιδευόμενοι να μάθουν την μερδεμένη σύνταξη της εντολής printf της γλώσσας C, η εντολή print

στην Python, μπορεί να τυπώσει σωστά μεταβλητές όλων των τύπων, και πιο απλές και πιο σύνθετες.

❖ **Συμβολοσειρές, λίστες και λεξικά είναι ενσωματώσιμα που λειτουργούν με αρκετά ομοιόμορφη σύνταξη.**

Ένας αρχάριος προγραμματιστής μπορεί να βελτιωθεί, απλά χρησιμοποιώντας συμβολοσειρές (αποθήκευση δεδομένων κειμένου), λίστες (αποθήκευση ακολουθιών) και λεξικά (αποθήκευση συσχετίσεων). Με την Python είναι πολύ εύκολη η χρήση αυτών των 3 δομών δεδομένων, χρησιμοποιώντας μια ομοιόμορφη σύνταξη. Οι αρχάριοι προγραμματιστές δεν ενδιαφέρονται για τις δηλώσεις τύπου, τον παραμετρικό πολυμορφισμό ή οποιαδήποτε άλλη παράξενη γλώσσα προγραμματισμού-θέλουν απλώς τον κώδικα τους να λειτουργεί όπως αυτοί επιθυμούν με όσο το δυνατόν λιγότερη πληκτρολόγηση!

❖ **Όμορφη, «καθαρή» σύνταξη.**

Είναι αδιαμφισβήτητο ότι ο κώδικας Python φαίνεται καθαρός και όμορφος και ότι ο καθαρός κώδικας είναι ευκολότερο στο να δημιουργηθεί, να αναγνωστεί και να κατανοηθεί. Ωστόσο, ο κώδικας της Python δεν είναι τόσο σύντομος ώστε να προκαλεί σύγχυση, όπως συμβαίνει συχνά με τον κώδικα Perl. Χαρακτηρίζεται από μεγάλη ισορροπία ανάμεσα στη συνοπτικότητα και τη σαφήνεια.

❖ **Μια διαδραστική γραμμή εντολών και τη δυνατότητα να επαναφορτωθούν δυναμικά ενότητες χωρίς επανεκκίνηση μιας διαδραστικής συνεδρίας**

Κάθε φορά που οι αρχάριοι προγραμματιστές έχουν μια ερώτηση σχετικά με το πώς ακριβώς λειτουργεί μια λειτουργία ή ένα γλωσσικό χαρακτηριστικό, ο καλύτερος τρόπος για να απαντήσετε είναι μέσω της επιστημονικής μεθόδου: δοκίμασε το πρώτα να δεις τι θα συμβεί, στη συνέχεια βελτίωσε την κατανόησή σου και επανέλαβε τον πειραματισμό σου. Ο καλύτερος τρόπος πειραματισμού είναι μέσω μιας διαδραστικής γραμμής εντολών, επειδή δεν είναι αναγκαία η αποθήκευση ενός πηγαίου αρχείου, η επαναμεταγλώττισή του και η επαναδιατύπωσή του. Χρησιμοποιώντας την Python, είναι εφικτό να

δημιουργηθούν δομές δεδομένων σε ένα διαδραστικό περιβάλλον και στην συνέχεια να δοκιμαστούν σε πραγματικό χρόνο.

❖ **Λειτουργίες οι οποίες επιστρέφουν νέες τιμές αντί να μεταφέρουν υπάρχοντα αντικείμενα**

Η λειτουργική αυτή ιδιότητα είναι εξαιρετικά χρήσιμη επειδή επιτρέπει στους χρήστες να συνεχίσουν να πειραματίζονται στο ίδιο αντικείμενο χωρίς να είναι αναγκαίο αυτό το αντικείμενο να μεταλλαχθεί. Ένας από τους καλύτερους τρόπους ώστε να μάθει αποτελεσματικά προγραμματισμό ένας αρχάριος, είναι να πειραματιστεί και η προαναφερόμενη ιδιότητα αυτό το καθιστά εύκολα εφικτό.

❖ **Περιεκτική βιβλιοθήκη και πολλές διαθέσιμες δωρεάν βιβλιοθήκες τρίτων.**

Η τεράστια βιβλιοθήκη της Python βοηθά σημαντικά τους εκπαιδευόμενους να δημιουργήσουν προγράμματα και εφαρμογές που εξυπηρετούν διάφορους και διαφορετικούς σκοπούς. Χωρίς την βοήθεια της βιβλιοθήκης της Python οι εκπαιδευόμενοι δεν θα κατάφερναν να προγραμματίσουν αποτελεσματικά, ώστε να αποκτήσουν κίνητρο για επιπλέον γνώση στον προγραμματισμό.

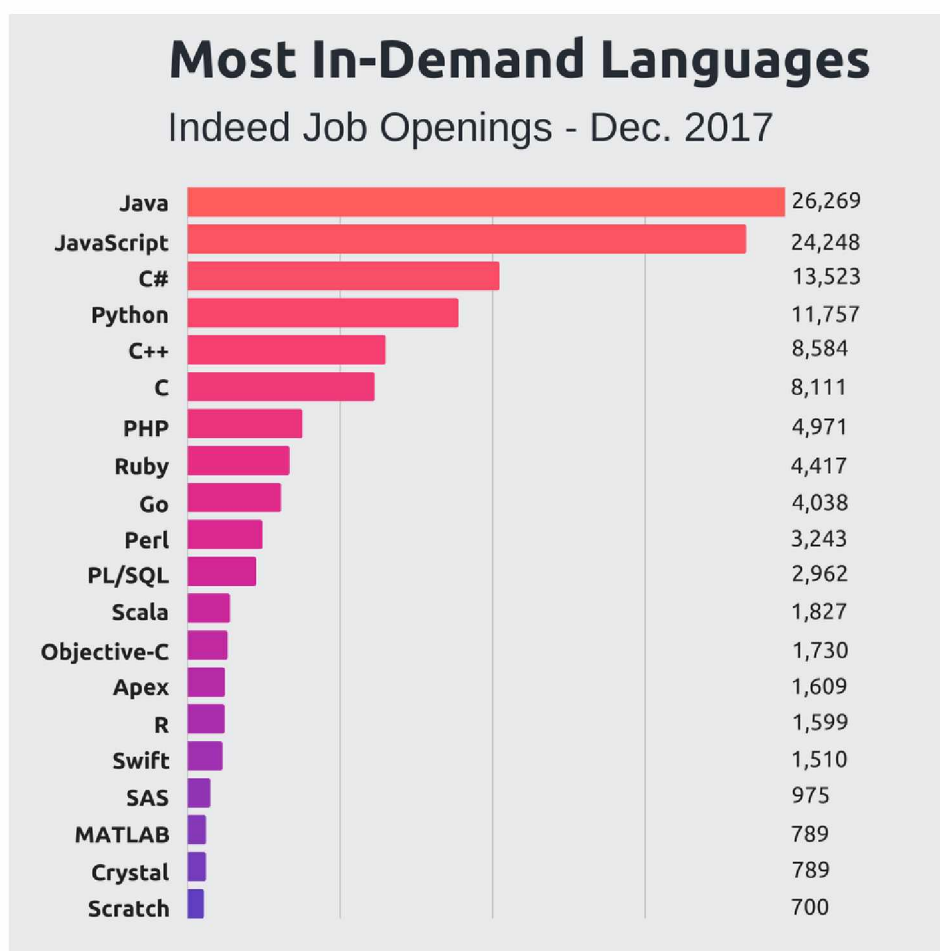
❖ **Οι κατάλληλοι τελεστές είναι υπερφορτωμένοι σε κατάλληλο βαθμό.**

Η υπερφόρτωση των τελεστών μπορεί να κάνει τον κώδικα να εμφανίζεται πιο συνοπτικός και αναγνώσιμος, αλλά μπορεί επίσης να παρεμποδίσει την λειτουργία του κώδικα όταν χρησιμοποιείται απερίσκεπτα. Ο βαθμός στον οποίο μια γλώσσα επιτρέπει στους τελεστές της την υπερφόρτωση (και ποιοι πραγματικά είναι υπερφορτωμένοι στην βιβλιοθήκη) καθορίζει την αναγνωσιμότητα του κώδικα. Από τη μία μεριά, στην C ή στην Java, σχεδόν κανένας τελεστής δεν είναι υπερφορτωμένος, γι 'αυτό είναι απαραίτητο να γίνουν λεπτομερείς κλήσεις λειτουργίας / μεθόδου για να εκτελεστούν οι περισσότερες λειτουργίες. Από την άλλη, στην Perl, πολλοί τελεστές είναι υπερφορτωμένοι και έχουν διαφορετικές έννοιες σε διαφορετικά πλαίσια, καθιστώντας τον κώδικα να μοιάζει κρυπτογραφημένος.

3.4 Κατάταξη της Python σε σχέση με τις άλλες γλώσσες προγραμματισμού.

Σύμφωνα με τον [21], οι εγκυρότερες και οι πιο αξιόπιστες μετρήσεις για την αξιολόγηση των κορυφαίων γλωσσών προγραμματισμού είναι:

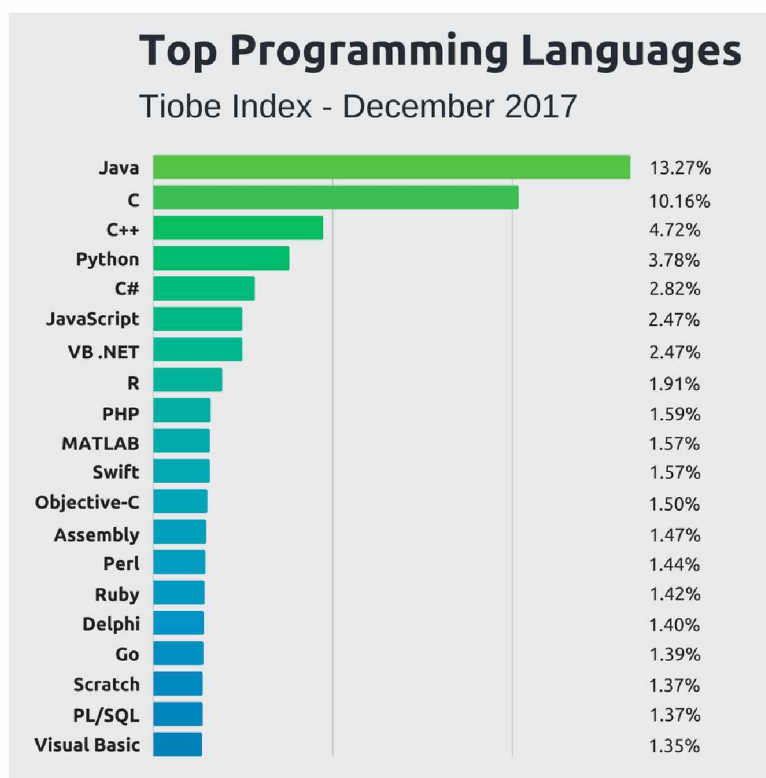
- **Indeed.com:** Η [Indeed](#) είναι παγκοσμίως η μεγαλύτερη μηχανή αναζήτησης εργασίας. Παρουσιάζει μια επαρκής μέτρηση των περισσότερο σε ζήτηση εργασιών που σχετίζονται με τον προγραμματισμό. Ειδικότερα, εξετάστηκε ο αριθμός των θέσεων εργασίας για τις 50 κορυφαίες γλώσσες προγραμματισμού, σύμφωνα με τον δείκτη Tiobe.



Εικόνα 3.1: Ευκαιρίες στην εργασία βασισμένες στις δημοφιλείς γλώσσες προγραμματισμού

Στην εικόνα 3.1, απεικονίζεται ένα γράφημα το οποίο δείχνει τις ευκαιρίες εργασίας οι οποίες σχετίζονται με τις 50 δημοφιλέστερες γλώσσες προγραμματισμού, σύμφωνα με τον δείκτη Tiobe. Όπως είναι αναμενόμενο οι

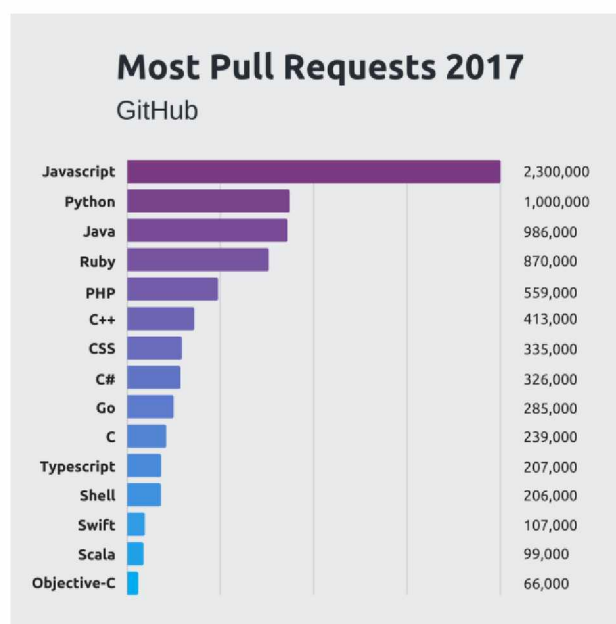
δημοφιλέστερες γλώσσες να έχουν περισσότερες ευκαιρίες στην εργασία (η Python βρίσκεται στην τρίτη θέση). Παρατηρούμε όμως, ότι λιγότερο δημοφιλείς γλώσσες όπως η Apex, βρίσκεται στις 20 πρώτες γλώσσες (εικόνα 3.1) σε δημοφιλία. Το γεγονός αυτό, υπάρχει πιθανότητα να καταδεικνύει την ανοδική τάση στη δημοτικότητα τέτοιου είδους γλωσσών.



Εικόνα 3.2: κορυφαίες γλώσσες προγραμματισμού (δείκτης Tiobe)

➤ **GitHub**

Το [GidHub](https://github.com) είναι παγκοσμίως ένα από τα μεγαλύτερα αποθετήρια κώδικα. Ετησίως δημιουργείται μια έκθεση με το όνομα «Year in Review», η οποία δίνει στατιστικά στοιχεία που αφορούν τη δική του κοινότητα προγραμματιστών. Ο δείκτης μέτρησης δημοτικότητας γλωσσών που χρησιμοποιούν, ονομάζεται «Pull Requests Opened».



Εικόνα 3.3: Αιτήματα έλξης γλωσσών προγραμματισμού

Στο γράφημα της εικόνας 3.3, απεικονίζονται οι 15 γλώσσες προγραμματισμού που αποτελούν αίτημα έλξης της κοινότητας του GitHub. Το αίτημα έλξης είναι δείκτης του ποσού του κώδικα που γράφεται. Η Javascript έχει μεγάλο αποτύπωμα στο αποθετήριο GitHub, με την επόμενη γλώσσα να έχει λιγότερο από τις μισές αιτήσεις έλξης. Όπως φαίνεται στο γράφημα, το 2017 η Python βρέθηκε πάνω από τη Java, καταλαμβάνοντας τη δεύτερη θέση.

Τα τελευταία 15 χρόνια, η δημοτικότητα της Python αυξάνεται σταθερά. Ένας σημαντικός λόγος για αυτό, είναι ότι η Python έχει εμπλακεί σε αρκετές από τις σημερινές τεχνολογίες, όπως στην μηχανική, την τεχνητή νοημοσύνη, στα μεγάλα δεδομένα, και στη ρομποτική.

➤ **IEEE Spectrum** [22]

Το IEEE Spectrum, παρουσιάζει την Πέμπτη ετήσια κατάταξη των κορυφαίων γλωσσών προγραμματισμού, χρησιμοποιώντας συνδυαστικά μετρήσεις διάφορες πηγές, σε 47 γλώσσες.

Language Rank	Types	Spectrum Ranking
1. Python	🌐 🖥️ 📱	100.0
2. C++	📱 🖥️ 📱	99.7
3. Java	🌐 📱 🖥️	97.5
4. C	📱 🖥️ 📱	96.7
5. C#	🌐 📱 🖥️	89.4
6. PHP	🌐	84.9
7. R	🖥️	82.9
8. JavaScript	🌐 📱	82.6
9. Go	🌐 🖥️	76.4
10. Assembly	📱	74.1

Εικόνα 3.4: Κατάταξη των κορυφαίων γλωσσών προγραμματισμού

Όπως απεικονίζεται στην εικόνα 3.4, η Python διατηρεί την πρώτη θέση στην κατάταξη. Η C++ κατέχει 99,7 βαθμούς την ίδια στιγμή που η C βρίσκεται στην τέταρτη θέση με 97,7 βαθμούς. Η Python εμφανίζεται ως ενσωματωμένη γλώσσα και αυτό υπάρχει περίπτωση να είναι ένας λόγος για την αύξηση της δημοτικότητάς της. Μια σημαντική περίπτωση χρήσης της Python είναι η ευχρηστία της σε ορισμένες εφαρμογές, όπως η δυναμική ενεργοποίηση της λειτουργίας των scripts.

Επίσης, ένας επιπλέον λόγος της αυξανόμενης δημοτικότητας της Python είναι η μικρή πτώση της γλώσσας προγραμματισμού R, η οποία έφτασε στο No. 5 το 2016, έπεσε στο No. 6 το 2017 και τώρα κατέχει την έβδομη θέση. Η γλώσσα προγραμματισμού είναι εξειδικευμένη στη διαχείριση στατιστικών στοιχείων και μεγάλου όγκου δεδομένων. Έχοντας υπόψη ότι το παγκόσμιο ενδιαφέρον στα δεδομένα μεγάλου όγκου, έχει στραφεί στις εφαρμογές τους στη μηχανική μάθηση και στην ύπαρξη υψηλής ποιότητας βιβλιοθηκών της Python, καθιστά πιθανότατα την Python ελκυστικότερη από την R.

4. Μεθοδολογία

4.1 Οι δυσκολίες στην εκμάθηση του προγραμματισμού

Παρόλο που είδαμε σε προηγούμενο κεφάλαιο ότι ο προγραμματισμός βοηθά γενικότερα στη μάθηση, οι εκπαιδευόμενοι αντιμετωπίζουν σοβαρά προβλήματα στην εκμάθηση του προγραμματισμού και πολλές φορές τον θεωρούν σύνθετη και πολύπλοκη διαδικασία. Σύμφωνα με τους [23], ορισμένες δυσκολίες εκδηλώνονται όταν ο εκπαιδευόμενος προγραμματιστής, προσπαθεί να προβλέψει ή να διορθώσει τη λειτουργία ενός αλγορίθμου. Μερικές από τις δυσκολίες συναντώνται κατά τρόπο συστηματικό και είναι σχεδόν ανεξάρτητες από τη μέθοδο διδασκαλίας του προγραμματισμού.

Υπάρχουν διάφοροι λόγοι και αιτίες που συμβαίνει αυτό. Αρχικά όπως είναι γνωστό η διαδικασία του προγραμματισμού απαιτεί χρήση συμβόλων για τη διαχείριση και αποθήκευση δεδομένων – μεταβλητών. Η κατανόηση της έννοιας της μεταβλητής είναι πρωταρχικής σημασίας αλλά και παράλληλα κάτι δύσκολο. Οι εκπαιδευόμενοι συγχέουν την εκχώρηση τιμής σε μια μεταβλητή με την ισότητα στα μαθηματικά [10].

Αρκετές και σημαντικές δυσκολίες κατανόησης απαντώνται και στις βασικές δομές, όπως στη δομή ελέγχου και στην επαναληπτική δομή. Ειδικότερα η κατανόηση της δομής επανάληψης αποτελεί μεγάλο «αγκάθι» για τους αρχάριους προγραμματιστές καθώς δεν είναι ένας αυθόρμητος και συνηθισμένος τρόπος επίλυσης προβλημάτων.

Ο [24], υποστηρίζει ότι τα λάθη εκπαιδευομένων ταξινομούνται στις επόμενες τρεις κατηγορίες.

1. **Λάθος εφαρμογή της αναλογίας:** η μεταβλητή παρομοιάζεται ως ένα κουτί αποθήκευσης, ώστε αρκετοί εκπαιδευόμενοι να πιστεύουν ότι μπορεί να αποθηκεύσει παραπάνω από μία τιμές.
2. **Υπεργενίκευση:** πολλές φορές πιστεύεται ότι όταν κάτι λειτουργεί σωστά για έναν τύπο αντικειμένου θα πρέπει να λειτουργεί σωστά και για κάποιον άλλο.
3. **Λάθος χειρισμός της πολυπλοκότητας και της αλληλεπίδρασης:** κακή διαχείριση και τοποθέτηση σε λάθος σημεία των υποπρογραμμάτων.

Επίσης ο [24], παρατηρεί ότι οι εκπαιδευόμενοι πιστεύουν αρκετές φορές ότι το πρόγραμμα θα επιστρέψει ως αποτέλεσμα αυτό που ίδιοι θέλουν ως αποτέλεσμα, χωρίς να κατανοούν ότι το αποτέλεσμα προκύπτει από την εκτέλεση των εντολών του προγράμματος που οι ίδιοι έχουν δημιουργήσει. Σημειώνει επίσης ότι, οι εκπαιδευόμενοι αρκετές φορές αγνοούν πώς να επιλύσουν κατάλληλα ένα πρόβλημα ώστε αυτό να οδηγεί σε λάθος ενέργειες από μέρους τους ή ακόμα και σε υπερπροσπάθεια για ένα απλό συντακτικό λάθος.

Η «δυσκολία» και η μη εμπέδωση βασικών εννοιών του προγραμματισμού παρουσιάζεται με την αδυναμία της επίλυσης απλών προβλημάτων και υπάρχει πιθανότητα να συνδέεται με τον κλασσικό τρόπο της διδασκαλίας του προγραμματισμού. Αυτό σημαίνει ότι οι αρχάριοι προγραμματιστές δεν μπορούν να χρησιμοποιήσουν δομικά εργαλεία της γλώσσας προγραμματισμού, όταν αυτά δεν αντιστοιχούν σε καθημερινές εμπειρίες και συνήθειες.

Συνήθως ο εκπαιδευτής παρουσιάζει βασικές προγραμματιστικές έννοιες και δομές, συνήθως, χρησιμοποιώντας συγκεκριμένη γλώσσα προγραμματισμού. Επίσης επιμένει στην διδασκαλία του συντακτικού της συγκεκριμένης γλώσσας προγραμματισμού.

Σύμφωνα με τους [25], τα σημαντικότερα προβλήματα που αντιμετωπίζουν οι αρχάριοι προγραμματιστές που διδάσκονται τις αρχές προγραμματισμού με την κλασσική προσέγγιση είναι τα εξής :

- Οι γλώσσες προγραμματισμού γενικού σκοπού διαθέτουν ένα μεγάλο μενού εντολών και είναι πολύπλοκες.
- Η προσοχή των μαθητών εστιάζεται περισσότερο στην εκμάθηση του συντακτικού της γλώσσας και όχι στην καλλιέργεια ικανοτήτων της επίλυσης προβλημάτων.
- Η διαδικασία εκτέλεσης του προγράμματος δεν είναι εύκολα αντιληπτή από τον μαθητή γιατί δεν είναι ορατή.
- Η πολυπλοκότητα που απαιτεί η δημιουργία ενός αλγορίθμου σε μια γλώσσα προγραμματισμού είναι μεγάλη, λόγω της φύσης της γλώσσας.
- Η επίλυση προβλημάτων με ενδιαφέρον, έχει ως προαπαιτούμενο τη γνώση μεγάλου ποσοστού της γλώσσας.

Εξέχουσας σημασίας είναι επίσης ότι, τα προβλήματα που καλούνται να επιλύσουν χρησιμοποιώντας προγραμματισμό οι εκπαιδευόμενοι είναι μαθηματικού περιεχομένου. Αυτό έχει ως αποτέλεσμα ο εκπαιδευόμενος να μην έχει κίνητρο και ενδιαφέρον για αυτό.

Επιπλέον, ιδιαίτερα σημαντικό είναι το γεγονός ότι οι περισσότερες γλώσσες προγραμματισμού, δεν έχουν ελκυστικό περιβάλλον, κάτι το οποίο είναι καθοριστικό ειδικά για τους «μικρούς» προγραμματιστές.

Σημαντικό είναι επίσης ότι οι εκπαιδευόμενοι δεν κατανοούν τις λειτουργίες των προγραμματιστικών εννοιών και δυσκολεύονται να κατακτήσουν ικανότητες στη σχεδίαση αποτελεσματικών προγραμμάτων.

Αυτό πολλές φορές έχει ως αποτέλεσμα σύμφωνα με τον [24], οι εκπαιδευόμενοι λόγω των αρνητικών τους εμπειριών να εγκαταλείπουν την προσπάθειά τους μόλις έλθουν αντιμέτωποι με ένα πρόβλημα. Επιπλέον ο [23], συμπεραίνει ότι η πρώτη επαφή και εμπειρία με τον προγραμματισμό, όταν οι εκπαιδευόμενοι σχεδιάζουν το πρόγραμμά τους, είναι εξαιρετικά σημαντική.

4.2 Εκπαιδευτικό λογισμικό

Στη παρούσα εργασία, δημιουργήθηκε εκπαιδευτικό υλικό το οποίο παρουσιάστηκε χρησιμοποιώντας ιστολόγιο (τεχνολογία υπερμέσων). Σύμφωνα με τους [26], το εκπαιδευτικό λογισμικό είναι μια κατηγορία του λογισμικού εφαρμογών και δημιουργείται για την επίτευξη καθορισμένων μαθησιακών στόχων. Μπορεί να το χρησιμοποιήσει ο εκπαιδευτής σαν συμπληρωματικό μέσο διδασκαλίας ή να παίξει το ρόλο του υποστηρικτικού μέσου αυτοδιδασκαλίας από τον εκπαιδευόμενο. Η εκπαιδευτική διαδικασία χρησιμοποιώντας εκπαιδευτικό λογισμικό μπορεί να αποδειχθεί εξαιρετικά αποτελεσματική. Η διδασκαλία μπορεί να γίνει εμπλουτισμένη, διαθεματική και να δώσει την επιπλέον δυνατότητα διερεύνησης.

Στις μέρες μας λόγω της τεράστιας ανάπτυξης των τεχνολογιών υλικού και λογισμικού, όλες οι κατηγορίες εκπαιδευτικού λογισμικού, χρησιμοποιούν υπερμεσικά χαρακτηριστικά. Με τη χρήση υπερμεσικών εφαρμογών ο εκπαιδευόμενος έχει πλέον τη δυνατότητα να επιλέξει ποια πληροφορία θα παρακολουθήσει.

Σύμφωνα με τους [27], από παιδαγωγική άποψη οι εκπαιδευτικές εφαρμογές υπερμέσων καλό θα είναι να ακολουθούν τις επόμενες αρχές:

1. Να παρέχουν προσανατολισμό στον εκπαιδευόμενο
 - Να θέτουν στόχους
 - Να παρουσιάζουν τα επιθυμητά αποτελέσματα
2. Να περιλαμβάνουν αλληλεπιδράσεις
3. Να περιέχουν αξιολόγηση

Τα τελευταία χρόνια αναπτύσσεται δυναμικά η εξ αποστάσεως εκπαίδευση, μέσω των εκπαιδευτικών εφαρμογών υπερμέσων. Σύμφωνα με την [28], οι εκπαιδευόμενοι προτιμούν να μαθαίνουν με τη χρήση του διαδικτύου για τους εξής λόγους:

- Επιλέγουν την ώρα εκμάθησης
- Οι μαθητές θεωρούν ότι τα εκπαιδευτικά συστήματα δίνουν έμφαση στην στείρα απομνημόνευση, ανταμείβοντάς τη.
- Δίνεται η δυνατότητα οι εκπαιδευόμενοι να εστιάσουν στην προσπάθειά τους, χωρίς περισπαστικούς παράγοντες από τους συν-εκπαιδευόμενούς τους.
- Οι εκπαιδευόμενοι έχουν τη δυνατότητα να ακολουθούν το δικό τους προσωπικό ρυθμό.
- Μέσω της παροχής των διαδικτυακών παροχών, υπάρχουν περισσότερες ευκαιρίες ενασχόλησης με θέματα που άπτονται των ενδιαφερόντων τους.
- Υπάρχει η δυνατότητα του προγραμματισμού της ενασχόλησής τους ανάλογα τις διαφορετικότητας των αναγκών εκμάθησης.
- Δίνεται η δυνατότητα εκμάθησης σε άτομα που περιορίζονται από κάποια θέματα υγείας τα οποία αποτελούν εμπόδιο σε ένα παραδοσιακό σκηνικό τάξης.

4.3 Μοντέλο θεωρίας μάθησης που χρησιμοποιήθηκε

Όπως προαναφέρθηκε, στη συγκεκριμένη εργασία δημιουργήθηκε ένα ιστολόγιο για την ανάπτυξη εκπαιδευτικού υλικού, με τη χρήση παραδειγμάτων, έτοιμα παραδείγματα σε Scratch και τα αντίστοιχα σε γλώσσα προγραμματισμού Python, το οποίο βασίστηκε στο μοντέλο Gagnè [29], [30]. Σύμφωνα με αυτό το μοντέλο,

υποστηρίζεται ότι μια θεωρία σχεδίασης της διδασκαλίας χρησιμοποιείται για την ανάπτυξη υλικού το οποίο μπορεί να αναλάβει το ρόλο του διδάσκοντα. Οι θέσεις που ανέπτυξε ο Gagnè έχουν βασιστεί στα πρότυπα των συμπεριφοριστών, στις γνωστικές θεωρίες και τέλος στη θεωρία επεξεργασίας πληροφοριών [31]. Σύμφωνα με τον [32], το μοντέλο έχει ως στόχο την αποτελεσματική διδασκαλία και είναι ένα «αθροιστικό» μοντέλο, διότι η νέα γνώση είναι συνέχεια της προηγούμενης γνώσης του εκπαιδευόμενου και η μάθηση είναι αποτέλεσμα της αλληλεπίδρασης του ατόμου με το περιβάλλον.

Σύμφωνα με το μοντέλο Gagnè [33], οι συνθήκες της μάθησης διακρίνονται σε εσωτερικές και εξωτερικές. Οι εσωτερικές αποτελούν την προϋπάρχουσα γνώση του εκπαιδευόμενου, τα κίνητρα και τους στόχους του. Οι εξωτερικές, αποτελούν τα ερεθίσματα (συμπεριφορισμός) που παρουσιάζονται στον εκπαιδευόμενο. Επίσης οι [29] και [34], υποστηρίζουν ότι για να επιτευχθεί η εσωτερική διαδικασία της μάθησης απαιτούνται τα επόμενα εννέα εκπαιδευτικά γεγονότα:

- Προσέλκυση της προσοχής του εκπαιδευόμενου
- Ενημέρωση για τους επιθυμητούς στόχους
- Ανάκληση προϋπάρχουσας γνώσης
- Σαφήνεια στην παρουσίαση του εκπαιδευτικού υλικού
- Καθοδήγηση της μάθησης
- Πρόκληση ενεργειών από τον εκπαιδευόμενο
- Ανάδραση
- Εκτίμηση απόδοσης
- Ενίσχυση της συγκράτησης γνώσης και προσπάθεια μεταβίβασής της σε άλλους τομείς.

Ο [35], θεωρεί ότι τα προηγούμενα εννέα συμβάντα δείχνουν τη μετατόπιση της αρχικής θεωρίας από τη συμπεριφοριστική μάθηση προς μια γνωστική θεωρία.

Σημαντικό είναι επίσης να αναφερθεί ότι τα παραδείγματα χρησιμοποιούνται για την αύξηση της διαδραστικότητας της προς εκμάθησης υλικού, σε σχέση με τον εκπαιδευόμενο.

4.4 Επισκόπηση των μελετών αξιολόγησης

Οι μελέτες αξιολόγησης που αναφέρονται στη συνέχεια έχουν θέμα παρεμφερές με την παρούσα εργασία. Αυτό σημαίνει ότι είτε έχουν μελετήσει τον τρόπο της εκμάθησης της γλώσσας προγραμματισμού Python, χρησιμοποιώντας περιβάλλοντα παρόμοιας λογικής με το Scratch, είτε έχουν μελετήσει τον τρόπο πως χρησιμοποιήθηκε το Scratch για την εκμάθηση μιας γλώσσας προγραμματισμού υψηλού επιπέδου.

4.4.1 Η Μελέτη των Mathrani, A., Christian, S., & Ponder-Sutton, A.(2016). *PlayIT: Game Based Learning Approach for Teaching Programming Concepts. Educational Technology & Society, 19(2), 5-17.*

Οι [36] έκαναν μια μελέτη σχετικά με τη χρήση του εκπαιδευτικού παιχνιδιού Lightbot 2.0 ως στοιχείο ενός μαθήματος. Στη συνέχεια παρουσιάζονται τα βασικά στοιχεία της μελέτης.

Ο στόχος της μελέτης ήταν να εντοπιστούν τα προβλήματα των φοιτητών οι οποίοι παρακολουθούν μαθήματα σχετικά με την επιστήμη των υπολογιστών σε έναν μη πανεπιστημιακό φορέα εκπαίδευσης. Επίσης σημαντικό σημείο αποτέλεσε η διερεύνηση της περίπτωσης τα εκπαιδευτικά παιχνίδια να αποτελούν βοήθεια ώστε να ξεπεραστούν τα προαναφερόμενα προβλήματα. Επιπλέον να διαφανεί ότι η χρήση ενός παιχνιδιού στη διδακτική πράξη ενισχύει και κεντρίζει το ενδιαφέρον των φοιτητών για μάθηση.

Στη μελέτη συμμετείχαν φοιτητές που παρακολουθούσαν τα μαθήματα Πληροφορικής του Εθνικού πτυχίου στην Πληροφορική των οποίων η ηλικία και το φύλο τους δεν αναφέρονται στη μελέτη. Οι συμμετέχοντες χωρίστηκαν σε δύο ομάδες. Στην πρώτη ομάδα υπήρχαν άτομα (20), τα οποία δεν είχαν καμία εμπειρία προγραμματισμού ενώ στη δεύτερη ομάδα υπήρχαν άτομα (24), τα οποία είχαν εμπειρία σε βασικά θέματα προγραμματισμού.

Η μελέτη διήρκεσε 4 μήνες και διεξήχθη σε δύο φάσεις. Στην πρώτη φάση, έγιναν συνεντεύξεις σε καθηγητές Πληροφορικής, για να εντοπιστούν οι δυσκολίες που

αντιμετωπίζουν οι φοιτητές στον προγραμματισμό και να καταγραφούν οι ιδέες και οι προτάσεις τους για την επίλυσή τους. Στη δεύτερη φάση έγινε η χρήση του εκπαιδευτικού παιχνιδιού LightBot 2.0 από τους φοιτητές σε εργαστήριο πληροφορικής, παίζοντας για μία ώρα.

Η επιλογή του συγκεκριμένου παιχνιδιού έγινε λόγω της συνάφειάς του με το αναλυτικό πρόγραμμα σπουδών του διδασκόμενου μαθήματος. Οι διάφορες δραστηριότητες του παιχνιδιού αντιστοιχήθηκαν στις έννοιες του μαθήματος ανάλογα με την δυσκολία τους.

Ολοκληρώνοντας την προηγούμενη διαδικασία οι 44 φοιτητές κλήθηκαν να συμπλρώσουν ένα ερωτηματολόγιο «online», με ερωτήσεις ανοικτού τύπου και ερωτήσεις με απαντήσεις βάσει της κλίμακας Likert. Δίνοντας εξετάσεις οι φοιτητές στο διδασκόμενο μάθημα, κλήθηκαν να απαντήσουν σε ένα δεύτερο ερωτηματολόγιο 15 ερωτήσεων, οι φοιτητές που απάντησαν ήταν 15. Επιπλέον καταγράφηκαν οι επιδόσεις των φοιτητών στις εξετάσεις, που συμμετείχαν στην μελέτη σε σχέση με το υπόλοιπο τμήμα για το αν υπάρχουν ιδιαίτερες επιδόσεις στην πρώτη ομάδα φοιτητών. Τα δεδομένα που συλλέχθηκαν υποβλήθηκαν σε στατιστική επεξεργασία.

Σαν συμπέρασμα της μελέτης μετά από τα προαναφερόμενα, ήταν ότι οι πλειοψηφία των φοιτητών που συμμετείχαν ανέφεραν ότι με το παιχνίδι το μάθημα έγινε πιο διασκεδαστικό και κατανοητό. Επίσης η χρήση του παιχνιδιού βοήθησε στην ενίσχυση της αυτοπεποίθησης των φοιτητών καθώς πλέον ήταν ικανοί να εφαρμόσουν βασικές προγραμματιστικές έννοιες μέσω του παιχνιδιού.

Τελικά διαφάνηκε ότι η στρατηγική μάθησης βασισμένη σε παιχνίδι είναι παιδαγωγικά χρήσιμη διότι συμβάλλει αποτελεσματικά στην εκμάθηση δύσκολων εννοιών.

4.4.2. Διπλωματική εργασία του Σιδέρη Γρηγόριου με θέμα: «Ανάπτυξη παιχνιδιού σοβαρού σκοπού για την εκμάθηση εννοιών προγραμματισμού με τη γλώσσα Python» (2019), Θεσσαλονίκη.

Ο [37] στην διπλωματική του εργασία στα πλαίσια του μεταπτυχιακού του διπλώματος ειδίκευσης στα Πληροφορικά Συστήματα, δημιούργησε ένα εκπαιδευτικό παιχνίδι ώστε ο παίκτης να μάθει βασικές προγραμματιστικές έννοιες της γλώσσας

Python. Απευθύνεται κυρίως σε μαθητές οι οποίοι δεν έχουν εμπειρία προγραμματισμού και περιέχει έξι επίπεδα.

Για την αξιολόγηση του παιχνιδιού Py-rate Adventures, προσκλήθηκαν φοιτητές και απόφοιτοι του μεταπτυχιακού του Πανεπιστημίου Μακεδονίας, των τριών τελευταίων ετών. Αυτό που έπρεπε να κάνουν ήταν να κατεβάσουν το παιχνίδι και να το χρησιμοποιήσουν για ένα διάστημα δέκα ημερών, όπου επιθυμούσαν και για όσο χρόνο ημερησίως. Τελικά συμμετείχαν συνολικά 31 χρήστες διαφορετικού επιπέδου γνώσης προγραμματισμού. Στη συνέχεια κλήθηκαν να συμπληρώσουν ένα ανώνυμο ερωτηματολόγιο με 43 ερωτήσεις κλειστού τύπου με απαντήσεις βασισμένες στη κλίμακα Likert και 3 ερωτήσεις ανοικτού τύπου για σχόλια και προτάσεις βελτίωσης του παιχνιδιού.

Τα συμπεράσματα της αξιολόγησης του παιχνιδιού περιείχαν γενικά θετικά αποτελέσματα στην εμπειρία του παίκτη και στα βραχυχρόνια μαθησιακά αποτελέσματα του παιχνιδιού.

Η κατηγορία που επιδέχεται σημαντική βελτίωση είναι αυτή που αφορά την εστίαση της προσοχής του παίκτη. Το σημαντικότερο είναι όμως ότι τα αποτελέσματα στην κατηγορία της μαθησιακής αντίληψης ήταν ενθαρρυντικά, καθώς αναδείχθηκε ο εκπαιδευτικός χαρακτήρας του παιχνιδιού και επιτεύχθηκαν οι εκπαιδευτικοί του στόχοι. Επίσης το ενδιαφέρον των συμμετεχόντων για να παίξουν το παιχνίδι περισσότερο από μία φορά ήταν αυξημένο καθώς επιθυμούσαν να το τερματίσουν.

4.4.3 Διπλωματική εργασία των Μαγκιτούρα Νορμπέρτα, Μουτζούρη Βασιλική, με θέμα «Συγκριτική Μελέτη Περιβαλλόντων και Πλατφορμών Για τη Διδασκαλία του Προγραμματισμού», Βόλος, 2010.

Οι [38], στα πλαίσια της διπλωματικής τους εργασίας, πραγματοποίησαν έρευνα-πείραμα σε μαθητές Λυκείου, ώστε να αξιολογήσουν την χρήση των πλατφορμών και περιβαλλόντων, Lego Mindstorms NXT, Alice, Scribbler Robot και Jeliot 3, για τη διδασκαλία του προγραμματισμού. Οι [38], δίδαξαν ογδόντα τέσσερις διδακτικές ώρες τμήμα της προβλεπόμενης διδακτέας ύλης από το Υπουργείο Παιδείας, του μαθήματος «Ανάπτυξη Εφαρμογών σε προγραμματιστικό Περιβάλλον». Ο αριθμός

των μαθητών που συμμετείχαν στην μελέτη ήταν εβδομήντα και ήταν μαθητές της Α΄ και Β΄ Λυκείου, χωρισμένοι σε τρία τμήματα.

Στη συνέχεια το κάθε τμήμα χωρίστηκε σε ετερογενής ομάδες των δύο ή τριών ατόμων και παρακολούθησε την διδασκαλία του κάθε περιβάλλοντος καλύπτοντας τη θεωρία των κεφαλαίων του σχολικού βιβλίου. Καθένα από τα τρία τμήματα παρακολούθησε 28 διδακτικές ώρες, επτά με κάθε περιβάλλον. Σημαντικό είναι να σημειωθεί ότι η διδασκαλία έλαβε χώρα στον εργαστηριακό χώρο του σχολείου.

Στη συνέχεια οι μαθητές συμπλήρωσαν ερωτήσεις πολλαπλής επιλογής και ασκήσεις εμπέδωσης πάνω στη διδαχθείσα ύλη, καθώς και προγραμματιστικές ασκήσεις. Επίσης, οι μαθητές συμπλήρωσαν και ένα ερωτηματολόγιο σύγκρισης των τεσσάρων προαναφερθέντων πλατφορμών-περιβαλλόντων.

Τελικά η μελέτη των [38], κατέληξε ότι τα περιβάλλοντα – πλατφόρμες εκμάθησης προγραμματισμού, όπως τα τέσσερα προαναφερόμενα, ενθαρρύνουν τη συμμετοχή των μαθητών στη μαθησιακή διδασκαλία και προωθούν αποτελεσματικά την μεταξύ τους συνεργασία. Επίσης κεντρίζουν έντονα το ενδιαφέρον των μαθητών και το μάθημα του προγραμματισμού γίνεται ευχάριστο και ελκυστικό. Επιπλέον προέκυψε ότι με την χρήση των περιβαλλόντων-πλατφορμών, η κατανόηση των βασικών εννοιών του προγραμματισμού καθιστάται εύκολη, αλλά η εμπέδωση και η χρήση τους παραμένει μια δύσκολότερη υπόθεση.

Γενικότερα η μελέτη κατέληξε ότι τα περιβάλλοντα μάθησης αποτελούν χρήσιμα διδακτικά εργαλεία για την εκμάθηση του προγραμματισμού, παρόλα αυτά δεν έχει υπάρξει κάποιο κυρίαρχο περιβάλλον το οποίο να ικανοποιεί όλες τις προδιαγραφές και να εκπληρώνει το στόχο του.

4.4.4 Η ιστοσελίδα codeacademy ως εργαλείο για την εκμάθηση προγραμματισμού, Θεοφανέλλης Τιμολέων, 2016.

Ο [39], στην εργασία του παρουσιάζει μια διαδικτυακή εφαρμογή που μπορεί να χρησιμοποιηθεί αποκλειστικά ή βοηθητικά για την εκμάθηση μιας γλώσσας προγραμματισμού. Προσπαθώντας να αξιολογηθεί η ιστοσελίδα, έγινε μιας ομάδας οκτώ φοιτητών οι οποίοι ενδιαφέρονταν να μάθουν τη γλώσσα HTML, ώστε να αξιολογηθούν σε μάθημα που την αφορούσε σε προπτυχιακό τους μάθημα.

Για να συντονιστεί η ομάδα των οκτώ φοιτητών προηγήθηκαν τρεις συναντήσεις. Στην πρώτη συνάντηση παρουσιάστηκε η ιστοσελίδα και προτάθηκε ο τρόπος χρήσης της. Όταν οι συμμετέχοντες ολοκλήρωσαν το 40% της ύλης, έλαβε χώρα η δεύτερη συνάντηση, ώστε να εντοπιστούν και να αντιμετωπιστούν προβλήματα που είχαν ενδεχομένως προκύψει. Κατά τη διάρκεια της συνάντησης συνέβη αναλυτική συζήτηση με παραγωγικό τρόπο, εφόσον υπήρξε σημαντική ανταλλαγή ιδεών και προτάσεων.

Στη τρίτη συνάντηση, η οποία πραγματοποιήθηκε μετά την ολοκλήρωση των μαθημάτων, οι συμμετέχοντες πρότειναν ιδέες σχετικές με την εμπειρία τους και την προετοιμασία για την αξιολόγηση του μαθήματος. Επίσης, η ομάδα των οκτώ φοιτητών ανέφεραν ότι ένιωσαν άνεση και εξοικείωση με το συγκεκριμένο περιβάλλον και ότι υπήρξε έντονο ενδιαφέρον και κίνητρο για να συνεχίσουν.

4.5 Ενότητες προς εκμάθηση

4.5.1 Περιγραφή των ενότητων προς εκμάθηση στην Python

Στη συγκεκριμένη θα περιγράψουμε αναλυτικά τις ενότητες της Python που επιχειρείται να κατανοήσουν οι εκπαιδευόμενοι μέσω της συγκεκριμένης εργασίας.

Οι τελεστές στη γλώσσα προγραμματισμού Python

Οι βασικοί τελεστές σε μια γλώσσα προγραμματισμού είναι οι **μαθηματικοί τελεστές**, οι **λογικοί τελεστές** και οι **τελεστές σύγκρισης**.

Οι μαθηματικοί τελεστές χρησιμοποιούνται στην δημιουργία μαθηματικών εκφράσεων, οι οποίες πραγματοποιούν κάποιους υπολογισμούς και έχουν αποτέλεσμα μια τιμή. Στον επόμενο πίνακα φαίνονται οι μαθηματικοί τελεστές της γλώσσας Python [15].

Πίνακας 4.1: Μαθηματικοί τελεστές

Σύμβολο	Λειτουργία	Περιγραφή
+	Πρόσθεση	Προσθέτει 2 αριθμούς
-	Αφαίρεση	Αφαιρεί έναν αριθμό από έναν άλλο
*	Πολλαπλασιασμός	Πολλαπλασιάζει έναν αριθμό με έναν άλλο
/	Διαίρεση	Διαιρεί έναν αριθμό με έναν άλλο και δίνει το αποτέλεσμα ως έναν αριθμό κινητής υποδιαστολής
//	Ακέραια διαίρεση	Διαιρεί έναν αριθμό με έναν άλλο και δίνει το αποτέλεσμα ως έναν ακέραιο αριθμό
%	Υπόλοιπο διαίρεσης	Διαιρεί έναν αριθμό με έναν άλλο και δίνει το υπόλοιπο
**	Ύψωση σε δύναμη	Υψώνει έναν αριθμό σε μία δύναμη

Σύμφωνα με τους [15], σε κάθε έκφραση στην οποία υπάρχουν αριθμητικοί τελεστές ακολουθείται μια προτεραιότητα πράξεων:

1. Ύψωση σε δύναμη
2. Πολλαπλασιασμός, διαίρεση, υπόλοιπο ακέραιας διαίρεσης.
3. Πρόσθεση, αφαίρεση.

Σε περίπτωση που επιθυμούμε να αλλάξουμε την προηγούμενη ιεραρχία, χρησιμοποιούμε παρενθέσεις.

Στη συνέχεια είναι οι λογικοί τελεστές. Για να επιτευχθεί ο συνδυασμός των Boolean τιμών γίνεται χρήση των 3 βασικών **λογικών τελεστών**: **not**, **and** και **or**. Οι τελεστές αυτοί συμβολίζουν λογικές πράξεις (όχι, και, ή) και βοηθούν στη λήψη αποφάσεων σε ένα πρόγραμμα. Συνδέουν δύο λογικές εκφράσεις σε μια σύνθετη έκφραση.

- **AND:** Για να είναι αληθής η σύνθετη έκφραση θα πρέπει και οι δύο εκφράσεις που συμμετέχουν σε αυτή να είναι αληθείς.
- **OR:** Για να είναι αληθής η σύνθετη έκφραση, θα πρέπει η μία ή και οι δύο υπο-εκφράσεις να είναι αληθείς.
- **NOT (τελεστής):** Ο τελεστής **not** αντιστρέφει την τιμή αληθείας του τελεστέου του.

Ας υποθέσουμε ότι p και q είναι 2 μεταβλητές που δείχνουν σε Boolean τιμές. Στον επόμενο πίνακα, ο οποίος ονομάζεται πίνακας αληθείας, περιέχονται τιμές οι οποίες επιστρέφονται από τις τρεις λογικές πράξεις, πάνω στα p και q .

Πίνακας 4.2: Λογικοί τελεστές

p	q	$\text{not } p$	$p \text{ and } q$	$p \text{ or } q$
False	False	True	False	False
False	True	True	False	True
True	False	False	False	True
True	True	False	True	True

Τελικά έχουμε τους τελεστές σύγκρισης (σχεσιακοί), οι οποίοι εμφανίζονται αναλυτικά στον επόμενο πίνακα. Οι τελεστές αυτοί χρησιμοποιούνται για τη σύγκριση δύο τιμών ή μεταβλητών, με το αποτέλεσμα της σύγκρισης να είναι Αληθής (True) ή Ψευδής (False).

Πίνακας 4.3: Σχεσιακοί τελεστές

$x == y$	X είναι ίσος με y
$x != y$	X είναι διάφορο του y
$x > y$	X είναι μεγαλύτερο του y
$x >= y$	X είναι μεγαλύτερο ή ίσο του y
$x < y$	X είναι μικρότερο του y
$x <= y$	X είναι μικρότερο ή ίσο του y

Μεταβλητές

Η μεταβλητή είναι το όνομα που αναριστά μια τιμή αποθηκευμένη στη μνήμη του υπολογιστή και χρησιμοποιείται για την αποθήκευση και την επεξεργασία των δεδομένων. Το στοιχείο που περιέχει μια μεταβλητή λέγεται τιμή. Μια μεταβλητή έχει:

- Τύπο
- Όνομα
- Τιμή (η οποία δηλώνεται: $\langle \text{όνομα} \rangle =$ τιμή ή έκφραση υπολογισμού τιμής

Για την είσοδο δεδομένων από το πληκτρολόγιο χρησιμοποιείται η συνάρτηση εισόδου *input*, η οποία συντάσσεται:

variable = input ('συμβολοσειρά μηνύματος προτροπής')

Σημαντικό είναι να αναφερθεί ότι η μεταβλητή στην προηγούμενη σύνταξη είναι συμβολοσειρά και όχι αριθμός. Για να επιτευχθεί η μετατροπή της σε αριθμητικό τύπο θα πρέπει να χρησιμοποιηθούν άλλες συναρτήσεις, όπως η “int” που μετατρέπει τη μεταβλητή σε ακέραιο και η “float” που την μετατρέπει σε δεκαδικό.

Όταν δίνεται όνομα σε μία μεταβλητή θα πρέπει να μη χρησιμοποιούνται λέξεις – κλειδιά, όπως: for, else, if, while, def, sum, and κ.α.

Επίσης καλό θα είναι τα ονόματα των μεταβλητών να περιγράφουν την τιμή την οποία αποθηκεύουν.

Για να εκχωρήσουμε μια τιμή σε μια μεταβλητή χρησιμοποιούμε τον τελεστή ‘=’, όπως για παράδειγμα value=30. Στο αριστερό μέρος δίνουμε το όνομα της μεταβλητής στη συνέχεια έχουμε τον τελεστή εκχώρησης ‘=’ και στο δεξιό μέρος έχουμε μία τιμή, μια άλλη μεταβλητή ή μια έκφραση που αποτέλεσμά της είναι μια τιμή. Αυτό που είναι σημαντικό είναι να μην συγχέεται ο τελεστής ‘=’ με το σύμβολο της ισότητας και τη σημασία του στα Μαθηματικά.

Εκτός από τις μεταβλητές συμβολοσειρών, υπάρχουν και οι αριθμητικοί τύποι μεταβλητών. Όταν ο διερμηνέας της Python διαβάζει έναν αριθμό, προσδιορίζει τον τύπο του ανάλογα με τον τρόπο που είναι γραμμένος. Αυτό σημαίνει ότι αν είναι γραμμένο ως αριθμός χωρίς υποδιαστολή, θεωρείται ακέραιος (**int**), ενώ αν υπάρχει υποδιαστολή θεωρείται δεκαδικός (**float**).

Για να επιτευχθεί η είσοδος τιμής από το πληκτρολόγιο χρησιμοποιείται η συνάρτηση **“input”** π.χ. **name = input(‘Πως σε λένε;’)**.

Κατά την εκτέλεση της προηγούμενης εντολής, το πρόγραμμα σταματά και περιμένει την είσοδο δεδομένων από το χρήστη. Η απάντηση που θα δοθεί από αυτόν, αποθηκεύεται στη μεταβλητή **“name”**.

Εντολή εκτύπωσης “print”

Η συνάρτηση εξόδου “print”, χρησιμοποιείται για εκτύπωση στην οθόνη. Υπάρχουν διάφοροι τρόποι σύνταξής της, ανάλογα με το τι επιθυμεί ο προγραμματιστής να έχει

ως αποτέλεσμα [40], [41]. Η πιο απλή σύνταξη είναι: `print("Το όνομα σου είναι:", name)` σε σχέση με το προηγούμενο παράδειγμα με την `"input"`.

Η Δομή Ακολουθίας στην Python.

Η δομή της ακολουθίας είναι η πιο απλή από τις τρεις δομές. Στη δομή αυτή οι εντολές που περιγράφουμε εκτελούνται όλες η μια μετά από την άλλη, ακολουθιακά δηλαδή η μία μετά την άλλη. Χρησιμοποιώντας τη δομή ακολουθίας αποκλειστικά, μπορούμε να λύσουμε περιορισμένα προβλήματα στα οποία [15]:

- Η σειρά των βημάτων είναι καθορισμένη
- Όλα τα βήματα εκτελούνται πάντα
- Δεν υπάρχουν εξαιρέσεις

Η Δομή Επιλογής - Απόφασης στην Python.

Για την υλοποίηση της συγκεκριμένης δομής χρησιμοποιούμε την εντολή `"if"`. Οι δομές απόφασης επιτρέπουν σε ένα πρόγραμμα να έχει πολλαπλές διαδρομές εκτέλεσης. Η εντολή `"if"` (**απλή δομή επιλογής**) επιτρέπει σε μία ή περισσότερες εντολές να εκτελεστούν μόνο εάν κάποια λογική έκφραση είναι αληθής, διαφορετικά η ροή του προγράμματος θα προσπεράσει τη δομή `'if'` και θα συνεχίσει από την εντολή που βρίσκεται αμέσως μετά το τέλος της. Οι εκφράσεις αυτές συνήθως σχηματίζονται με ένα σχεσιακό τελεστή ο οποίος προσδιορίζει την ύπαρξη ή μη κάποιας συγκεκριμένης σχέσης μεταξύ δύο τιμών.

Η εντολή `'if'` συντάσσεται:

if <συνθήκη ελέγχου>:

#εντολές που θα εκτελεστούν αν ισχύει η συνθήκη ελέγχου

Επιπλέον έχουμε και την εντολή `"if...else"` (**σύνθετη δομή επιλογής**), η οποία θα εκτελέσει ένα μπλοκ εντολών εάν η συνθήκη είναι αληθής ή ένα άλλο μπλοκ εάν η συνθήκη είναι ψευδής.

Η εντολή ελέγχου `'if...else'` συντάσσεται:

if <συνθήκη ελέγχου>:

εντολές που θα εκτελεστούν, αν η συνθήκη ελέγχου είναι αληθής

else:

#εντολές που θα εκτελεστούν, αν η συνθήκη ελέγχου είναι ψευδής

Αυτό που είναι φανερό στη σύνταξη της προηγούμενης εντολής είναι ότι οι ομάδες εντολών που θα εκτελεστούν, αν ισχύει μια συνθήκη, ορίζονται ως ένα μπλοκ με τη

χρήση εσοχής. Οι διαδοχικές εντολές που έχουν την ίδια εσοχή ανήκουν στο ίδιο μπλοκ.

Η εντολή “**if...elif...else**” (πολλαπλή επιλογή), αποτελεί την επέκταση της “if...else” λογικής για παραπάνω των δύο εναλλακτικών επιλογών της ίδιας λογικής έκφρασης. Η σύνταξη της είναι:

if <συνθήκη>:

<εντολές>

elif:

<εντολές>

else:

<εντολές>

Η Δομή Επανάληψης στην Python.

Μια δομή επανάληψης κάνει μια εντολή ή ένα σύνολο εντολών να εκτελούνται περισσότερο από μία φορά. Το σημαντικότερο σημείο στη συγκεκριμένη δομή είναι ο αριθμός των επαναλήψεων, δηλαδή ο τερματισμός του βρόγχου. Υπάρχουν δύο τρόποι ελέγχου του βρόγχου.

- **Βρόγχος ελεγχόμενος με μετρητή:** Επαναλαμβάνεται ένα συγκεκριμένο αριθμό φορών (εντολή **for**). Η εντολή “for” έχει την εξής σύνταξη:

for μεταβλητή_στόχος in [τιμή1, τιμή2, ...]:

σύνολο εντολών: 

Τις περισσότερες φορές χρησιμοποιείται η συνάρτηση **range**, η οποία είναι αρκετά βοηθητική στη διαδικασία σύνταξης της εντολής “for”. Η συνάρτηση “range”, δημιουργεί έναν τύπο αντικειμένου γνωστό ως λίστα τιμών βρόγχου, παρόμοιο της λίστας.

Σε αρκετές περιπτώσεις η μεταβλητή στόχος χρησιμοποιείται σε κάποιον υπολογισμό ή κάποια άλλη διεργασία μέσα στο βρόγχο.

- **Βρόγχος ελεγχόμενος με συνθήκη:** Χρησιμοποιεί μία συνθήκη αληθούς/ψευδούς για να ελέγξει πόσες φορές θα επαναληφθεί (εντολή **while**).

Αυτό συμβαίνει έως ότου κάποια συνθήκη είναι αληθής. Η σύνταξη της “while”:

while <συνθήκη>:

εντολή

εντολή

.....

Ο βρόγχος που χρησιμοποιεί την εντολή “**while**” είναι βρόγχος προελέγχου. Ονομάζεται έτσι διότι ελέγχει τη συνθήκη του πριν πραγματοποιήσει την πρώτη του επανάληψη και για το λόγο αυτό είναι αναγκαίες κάποιες ενέργειες πριν από το βρόγχο, ώστε να εξασφαλιστεί ότι θα εκτελεστεί τουλάχιστον μία φορά.

Μεγάλης σημασίας είναι εντός του βρόγχου να επηρεάζεται το αποτέλεσμα της συνθήκης ώστε αυτή να μην ικανοποιείται πάντα και το πρόγραμμα οδηγηθεί σε ατέρμων βρόγχο.

Επίσης ένας τρόπος για τον τερματισμό του βρόγχου είναι και η χρήση της τιμής τερματισμού. Η τιμή αυτή είναι μία ειδική τιμή η οποία σηματοδοτεί το τέλος μιας αλληλουχίας τιμών, όταν διαβαστεί από το πρόγραμμα.

Οι Λίστες στην Python

Η λίστα είναι η δομή δεδομένων που περιέχει πολλαπλά δεδομένα. Κάθε δεδομένο ονομάζεται στοιχείο. Τα δεδομένα που υπάρχουν στην λίστα μπορούν να αλλάζουν (αύξηση ή μείωση), κατά την εκτέλεση του προγράμματος. Για να επιτευχθεί η προσπέλαση των στοιχείων της λίστας χρησιμοποιείται **δείκτης** (index).

Για να δημιουργηθεί η δομή δεδομένων λίστα, χρησιμοποιείται η συνάρτηση **list()**. Παράλληλα με τη συνάρτηση **range()**, δημιουργούνται οι τιμές της λίστας, οι οποίες αποτελούν την είσοδο για τη λίστα.

Σημαντική συνάρτηση που αφορά στις λίστες είναι και η **len()**. Η συγκεκριμένη δέχεται ως είσοδο την λίστα και επιστρέφει το μήκος της.

Επιπλέον χρησιμοποιείται και η συνάρτηση **append()**, η οποία προσθέτει στο τέλος της λίστας το στοιχείο που έχει ως όρισμα. Σε αντίθεση, η συνάρτηση **remove()**, αφαιρεί από τη λίστα το στοιχείο που δέχεται ως όρισμα, στην πρώτη του εμφάνιση.

Σε περίπτωση που το στοιχείο δεν υπάρχει στη λίστα η συνάρτηση `remove()`, επιστρέφει την τιμή *ValueError*.

Οι Συναρτήσεις στην Python.

Οι συναρτήσεις είναι μέρη προγράμματος που επαναχρησιμοποιούνται. Αυτό επιτυγχάνεται δίνοντας ένα όνομα σε ένα σύνολο εντολών (έχοντας ως σκοπό την εκτέλεση μιας συγκεκριμένης εργασίας). Με αυτό τον τρόπο γίνεται δυνατό να εκτελούμε το συγκεκριμένο κομμάτι χρησιμοποιώντας το όνομα, οπουδήποτε μέσα στο πρόγραμμα και όσες φορές είναι αναγκαίο. Υπάρχουν δύο είδη συναρτήσεων στη γλώσσα προγραμματισμού Python.

1. Ενσωματωμένες συναρτήσεις της Python από τη standard library όπως οι `input`, `print`, `int`, `range` κλπ, και των αρθρωμάτων (modules), για τις οποίες απαιτείται η εντολή *import*.
2. Συναρτήσεις ορισμένες από τον προγραμματιστή, των οποίων η βασική σύνταξή τους είναι:

```
def όνομα_συνάρτησης():  
    εντολή  
    εντολή  
    ...  
Κλήση  
όνομα_συνάρτησης()
```

Μία συνάρτηση μπορεί να δεχθεί παραμέτρους, δηλαδή τιμές που δίνονται σε αυτή έτσι ώστε αξιοποιώντας αυτές τις τιμές να μπορεί να παρουσιάσει ένα αποτέλεσμα. Οι τιμές αυτές καθορίζονται μέσα στην παρένθεση και ονομάζονται ορίσματα. Κατά την κλήση της συνάρτησης δίνονται τιμές στα ορίσματα με τον ίδιο τρόπο. Οι ονομασίες που δίνονται στον ορισμό της συνάρτησης ονομάζονται παράμετροι ενώ οι τιμές που δίνονται όταν καλείται η συνάρτηση ονομάζονται ορίσματα.

Υπάρχει η δυνατότητα σε συναρτήσεις με πολλές παραμέτρους να δοθούν τιμές μόνο σε εκείνες τις παραμέτρους που επιθυμείται, ενώ στις υπόλοιπες παραμέτρους υπάρχουν οι προεπιλεγμένες τιμές ορίσματος.

Επίσης πρέπει να σημειωθεί ότι υπάρχουν συναρτήσεις που δεν επιστρέφουν τιμές και ονομάζονται *void*. Σε περίπτωση που γίνεται επιστροφή τιμής από μια συνάρτηση επιτυγχάνεται με την εντολή *return*.

Τα βασικότερα πλεονεκτήματα της χρήσης των συναρτήσεων είναι πρωτίστως η απλούστερη δομή του προγράμματος, το οποίο βοηθά στην ταχύτερη ανάπτυξη του και στον ευκολότερο, αποδοτικότερο έλεγχο του. Επιπλέον για ίδιες λειτουργίες δεν είναι αναγκαίο να επαναληφθεί η συγγραφή εντολών σε διαφορετικά μέρη του προγράμματος.

4.5.2 Πρόταση μετάβασης από το Scratch στην Python.

Σε προηγούμενο κεφάλαιο αναφέρθηκε ότι η παρούσα εργασία επιχειρεί να διευκολύνει την εκμάθηση της γλώσσας προγραμματισμού Python, χρησιμοποιώντας την προηγούμενη γνώση των εκπαιδευομένων, η οποία είναι ο προγραμματισμός στο περιβάλλον του Scratch. Αρχικά γίνεται υπενθύμιση σε θεωρητικό πλαίσιο στις βασικές προγραμματιστικές έννοιες.

Στη συνέχεια δημιουργηθήκαν απλά παραδείγματα σε Scratch, και αποθηκευτήκαν στο περιβάλλον scratch.mit.edu, τα οποία αφορούν τις προαναφερόμενες βασικές έννοιες του προγραμματισμού. Οι έννοιες αυτές είναι:

- 1. Καταχώριση τιμής σε μεταβλητή**
- 2. Εντολή ‘εμφάνισε’**
- 3. Δομή ακολουθίας**
- 4. Δομή επιλογής**
 - Η εντολή ‘εάν...’
 - Η εντολή ‘εάν..αλλιώς’
 - Η πολλαπλή επιλογή
- 5. Δομή επανάληψης**
 - Η εντολή ‘επανάλαβε ώσπου...’
 - Η εντολή ‘επανάλαβε για...’
- 6. Λίστες**
- 7. Συναρτήσεις**

Τα παραδείγματα είναι προοδευτικά αυξανόμενης δυσκολίας, χωρίς να φτάνουν σε ιδιαίτερα απαιτητικό επίπεδο προγραμματισμού. Σε κάθε παράδειγμα υπάρχει και το προτεινόμενο πρόγραμμα για την υλοποίησή του σε Python.

Παρόμοιες ενότητες με τις ίδιες έννοιες, δημιουργηθήκαν και για τα παραδείγματα προγραμμάτων στη γλώσσα προγραμματισμού Python (Python3), προσθέτοντας την έννοια των τελεστών της συγκεκριμένης γλώσσας. Δηλαδή οι ενότητες προς εκμάθηση σε Python είναι:

- 1. Οι τελεστές στην Python**
- 2. Καταχώριση τιμής σε μεταβλητή**
- 3. Εντολή 'print'**
 - Η εντολή 'print' στην Python
 - Διαδοχικές εντολές 'print'
 - Η εντολή 'print' εκτελώντας απλές πράξεις
- 4. Δομή ακολουθίας**
- 5. Δομή επιλογής**
 - Η εντολή 'if'
 - Η εντολή 'if...else'
 - Η πολλαπλή επιλογή
- 6. Δομή επανάληψης**
 - Απλή εντολή 'for'
 - Η εντολή 'for' με συνθήκη
 - Η εντολή 'while'
- 7. Λίστες**
- 8. Συναρτήσεις**

Το επίπεδο δυσκολίας των προγραμμάτων σε Python είναι αντίστοιχο με αυτό των παραδειγμάτων σε Scratch. Για την παρουσίαση και αποθήκευση των προγραμμάτων Python, χρησιμοποιήθηκε το περιβάλλον repl.it/languages/Python3, στο οποίο μπορεί ο καθένας να δει, να μελετήσει και να εκτελέσει το πρόγραμμα χωρίς να είναι αναγκαία η δημιουργία προσωπικού του λογαριασμού. Σημαντικό είναι επίσης να αναφερθεί ότι ο εκπαιδευόμενος δεν είναι αναγκαίο να εγκαταστήσει κάποια εφαρμογή ή πρόγραμμα τοπικά στον υπολογιστή του.

Το ιστολόγιο το οποίο βρίσκεται στη διεύθυνση blogs.sch.gr/efsork, μπορεί να χρησιμοποιηθεί αποκλειστικά ή επικουρικά στην «εκμάθηση» της Python. Είναι ένα εκπαιδευτικό υλικό δωρεάν και πάντα διαθέσιμο, οπότε χαρακτηρίζεται από ευελιξία.

5. Υλοποίηση και παρουσίαση ιστολογίου.

Στη συγκεκριμένη εργασία δημιουργήθηκε το ιστολόγιο blogs.sch.gr/efsork, το οποίο περιέχει κάποιες βασικές προγραμματιστικές έννοιες αρχικά και στη συνέχεια υπάρχουν παραδείγματα (σενάρια) σε Scratch τα οποία υλοποιούνται και με τη γλώσσα προγραμματισμού Python. Σε κάθε παράδειγμα σε Scratch υπάρχει και το αντίστοιχο πρόγραμμα σε Python.

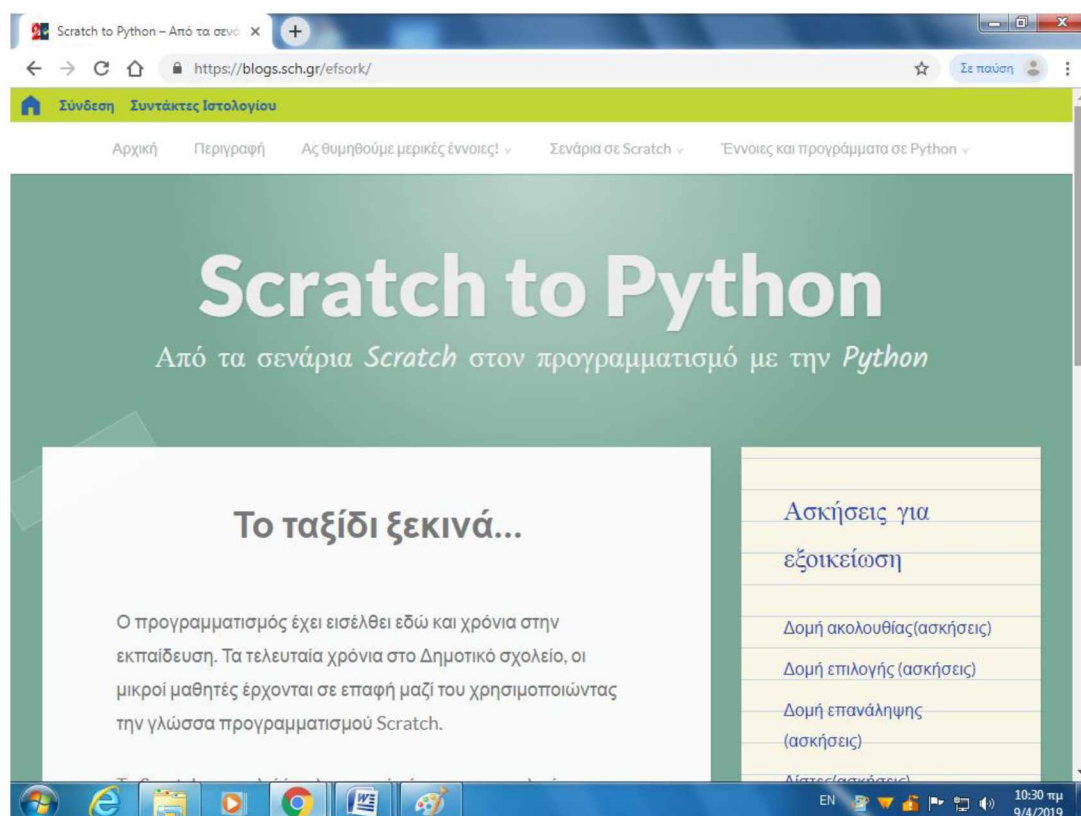
Για να δημιουργηθεί το ιστολόγιο χρησιμοποιήθηκε ο λογαριασμός της συγγραφέως στο Πανελλήνιο Σχολικό Δίκτυο⁴. Το ιστολόγιο έχει δημιουργηθεί με το **WordPress**.

Τα παραδείγματα σε Scratch έχουν δημιουργηθεί σε Scratch 3.0 και έχουν γίνει κοινόχρηστα στην κοινότητα Scratch. Με παρόμοιο τρόπο, τα προγράμματα σε Python3, δημιουργήθηκαν στο περιβάλλον replit/languages/Python3 και έγιναν κοινόχρηστα στην αντίστοιχη κοινότητα. Επίσης σε αριστερό μενού το ιστολόγιο περιέχει προτεινόμενες ασκήσεις για εξοικείωση με τη γλώσσα προγραμματισμού Python, έχοντας και προτεινόμενη λύση για κάθε πρόβλημα. Στη συνέχεια ακολουθεί η παρουσίαση του ιστολογίου δείχνοντας τις οθόνες που εμφανίζονται, ανάλογα την επιλογή μας.

Η επόμενη εικόνα αποτελεί την αρχική σελίδα του ιστολογίου. Η συγκεκριμένη σελίδα αποτελεί μια εισαγωγή στις έννοιες που παρουσιάζονται στην εργασία και ένα καλωσόρισμα στους χρήστες του ιστολογίου. Το κύριο μενού του ιστολογίου αποτελείται από τα εξής στοιχεία:

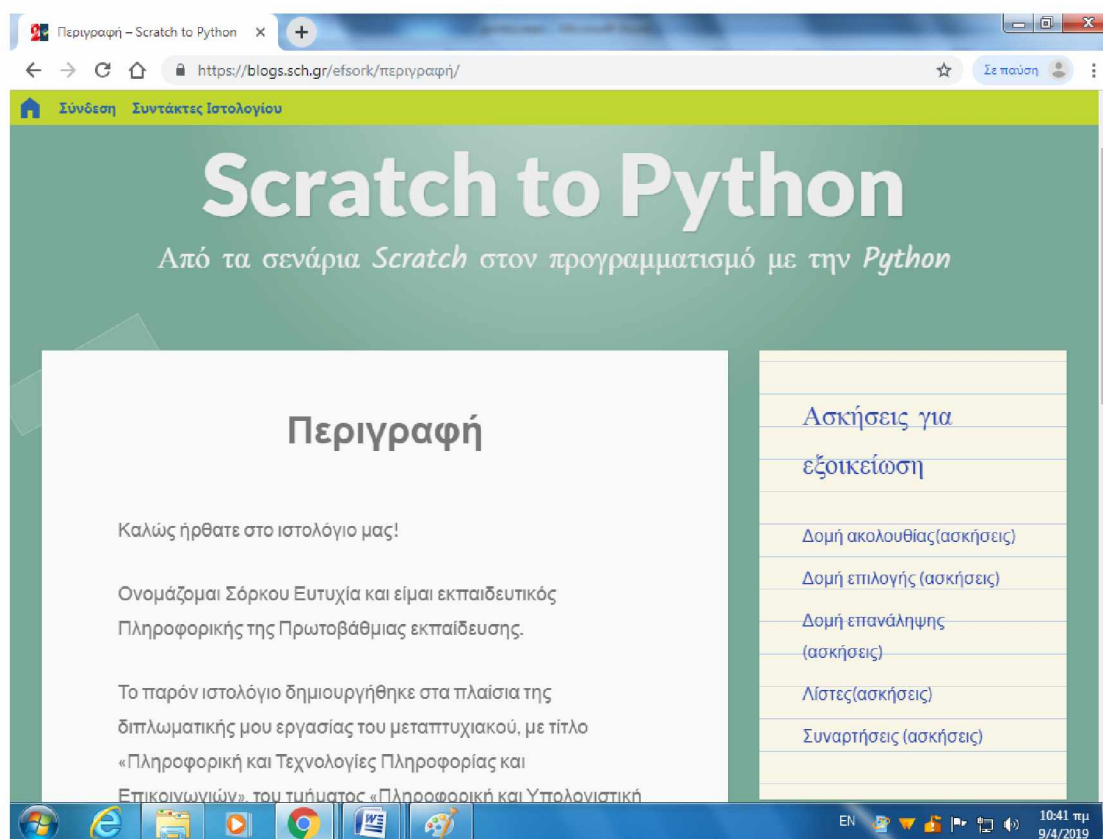
- *Αρχική*
- *Περιγραφή*
- *Ας θυμηθούμε μερικές έννοιες*
- *Σενάρια σε Scrtatch*
- *Έννοιες και προγράμματα σε Python*

⁴ <https://www.sch.gr/>



Εικόνα 5.1: Αρχική σελίδα

Στην επόμενη εικόνα εμφανίζονται τα περιεχόμενα του στοιχείου *‘Περιγραφή’*. Στη σελίδα αυτή γίνεται μια περιγραφή του λόγου δημιουργίας του ιστολογίου και παρουσιάζεται ο τρόπος με τον οποίο θα γίνει η προσπάθεια της εκμάθησης της γλώσσας προγραμματισμού Python, μέσω παραδειγμάτων πάνω σε βασικές προγραμματιστικές έννοιες, σε Scratch.



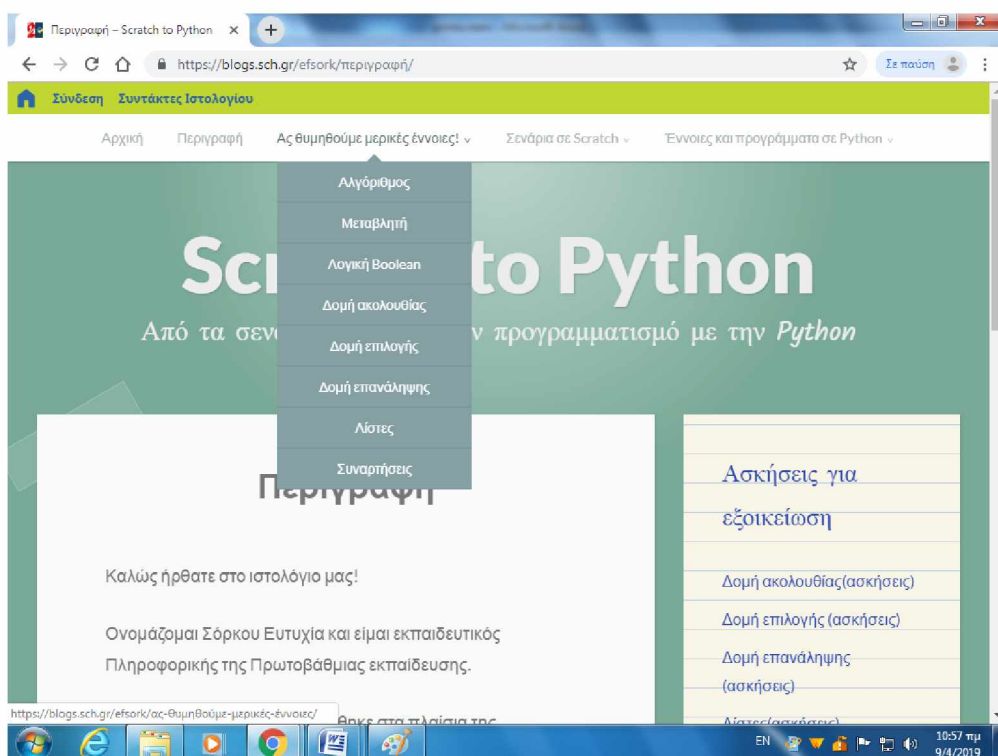
Εικόνα 5.2: Περιγραφή

- ***Το μενού ‘Ας θυμηθούμε μερικές έννοιες’***

Στη συνέχεια επιλέγοντας το στοιχείο μενού ‘*Ας θυμηθούμε μερικές έννοιες*’, εμφανίζονται τα περιεχόμενά του που είναι:

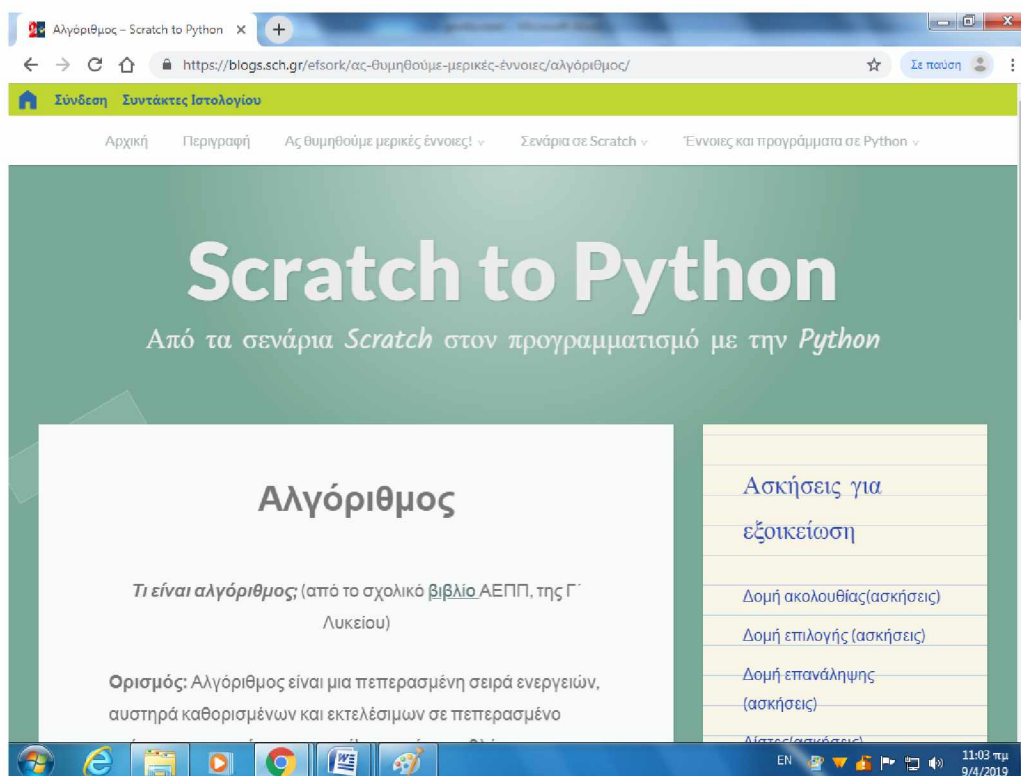
- ***Αλγόριθμος***
- ***Μεταβλητή***
- ***Λογική Boolean***
- ***Δομή Ακολουθίας***
- ***Δομή Επιλογής***
- ***Δομή Επανάληψης***
- ***Λίστες***
- ***Συναρτήσεις***

Το κάθε στοιχείο από τα προηγούμενα αποτελεί μια σελίδα στο ιστολόγιο.



Εικόνα 5.3: Ας θυμηθούμε μερικές έννοιες

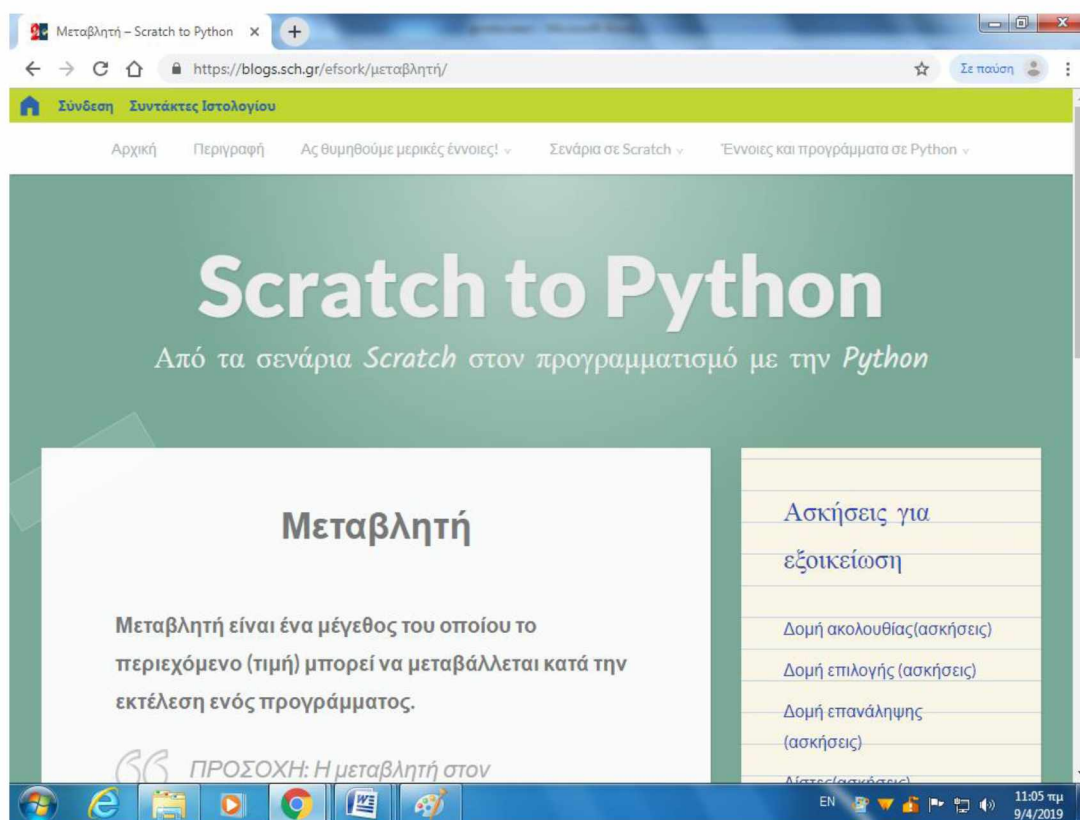
Όταν εισερχόμαστε στο στοιχείο ‘Αλγόριθμος’ εμφανίζεται στην οθόνη μας η επόμενη εικόνα.



Εικόνα 5.4: Αλγόριθμος

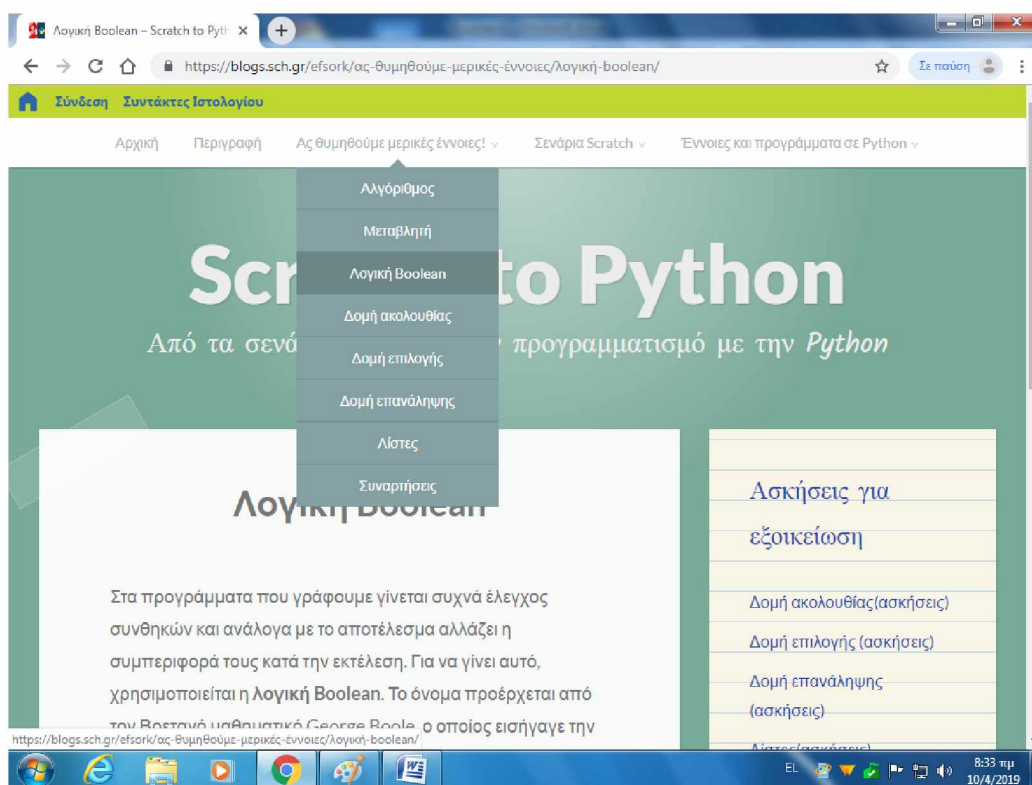
Στη συγκεκριμένη σελίδα γίνεται μια θεωρητική υπενθύμιση της έννοιας του αλγορίθμου και των χαρακτηριστικών του.

Στη συνέχεια επιλέγοντας το στοιχείο '**Μεταβλητή**', εμφανίζεται η ακόλουθη οθόνη, η οποία δίνει τον ορισμό και την έννοια της μεταβλητής χωρίς να γίνεται αναφορά σε συγκεκριμένη γλώσσα προγραμματισμού:



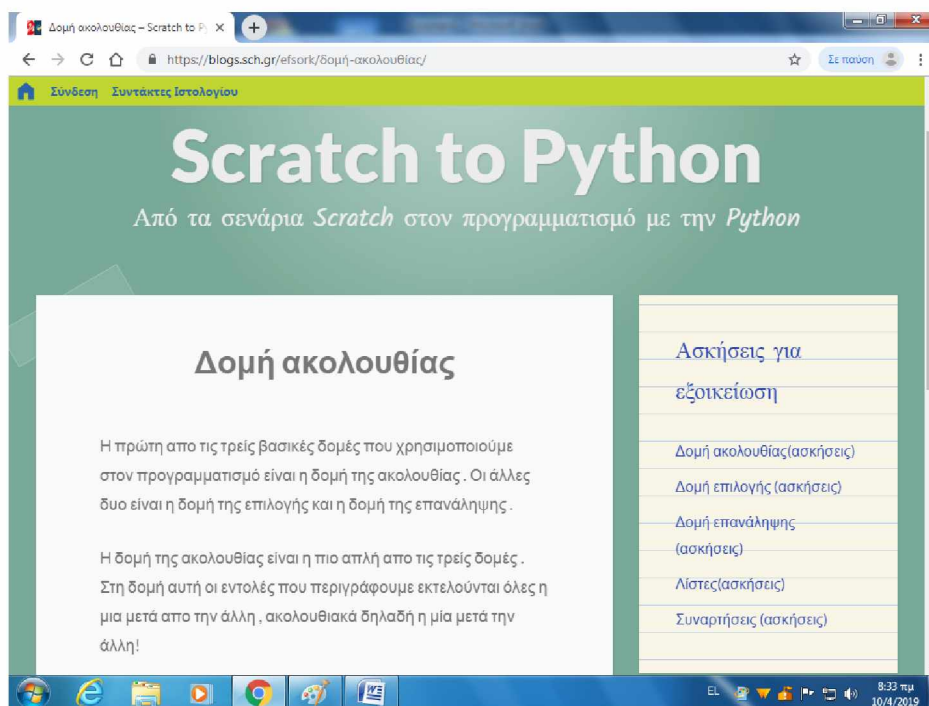
Εικόνα 5.5: Μεταβλητή

Το επόμενο στοιχείο του μενού είναι η σελίδα '**Λογική Boolean**' της οποίας τα περιεχόμενα εμφανίζονται στη συνέχεια. Στη σελίδα αυτή αναφέρεται μια σύντομη περιγραφή της Άλγεβρας Boole και για τον τρόπο που αυτή μας είναι αναγκαία στον προγραμματισμό και τις διάφορες δομές δεδομένων.



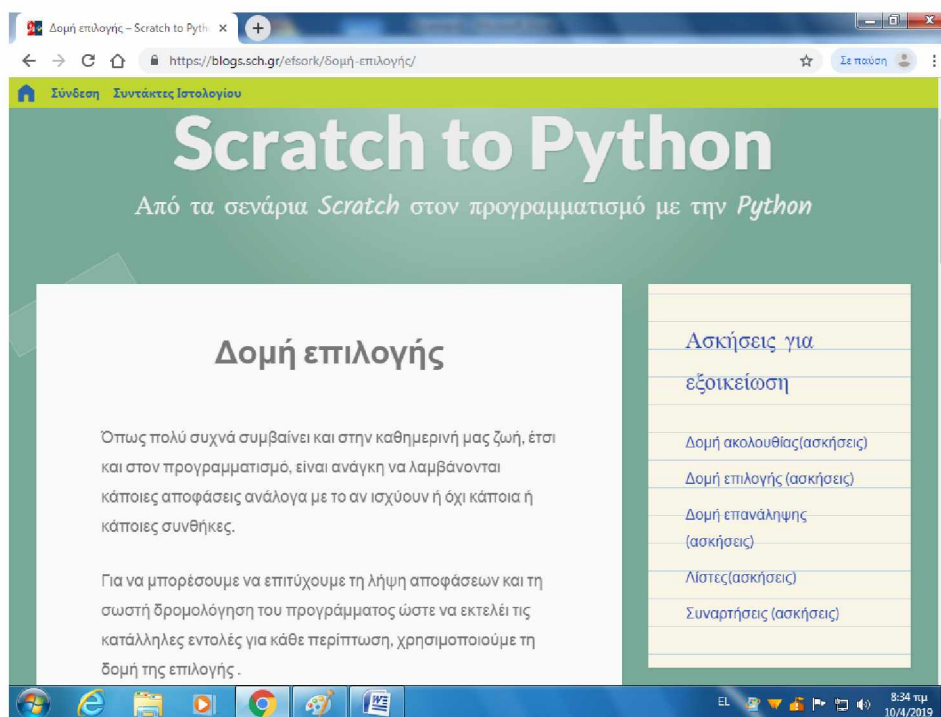
Εικόνα 5.6: Λογική Boolean

Η επόμενη σελίδα που υπάρχει στο μενού είναι η *‘Δομή Ακολουθίας’*. Στη σελίδα αυτή γίνεται μια σύντομη περιγραφή της δομής ακολουθίας και αναφέρεται και το γεγονός ότι είναι η ευκολότερη από τις τρεις δομές.



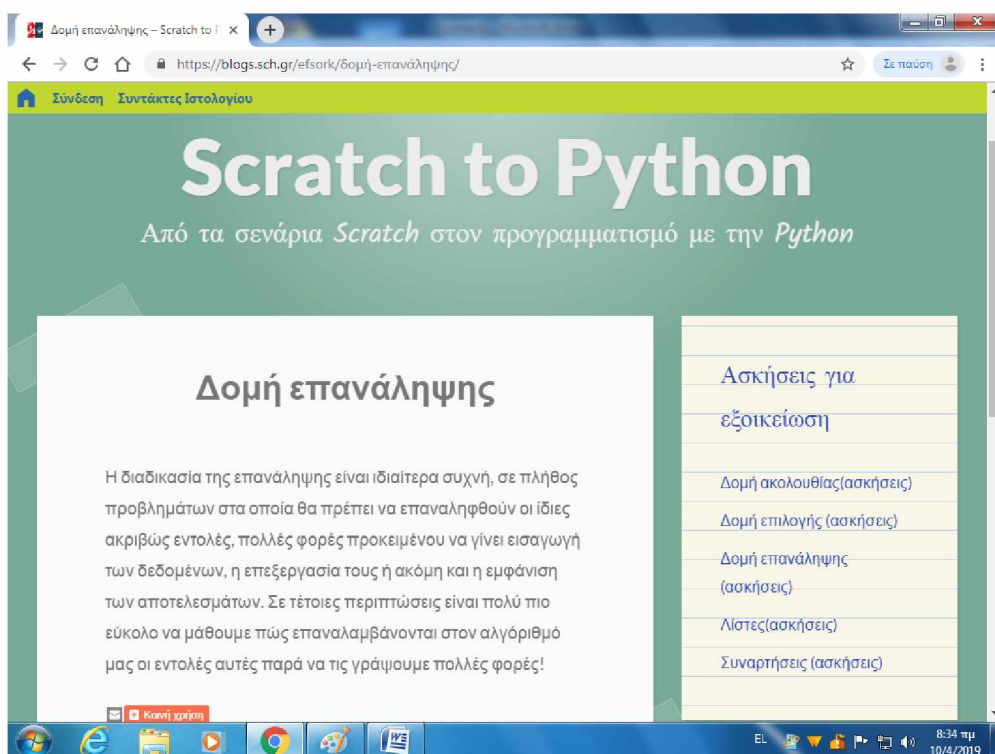
Εικόνα 5.7: Δομή ακολουθίας

Συνεχίζοντας την περιήγηση, στο μενού υπάρχει το στοιχείο **‘Δομή Επιλογής’**. Στη σελίδα αυτή γίνεται αναφορά στη χρήση της συγκεκριμένης δομής και παρουσιάζονται με συνοπτική περιγραφή τα είδη της.



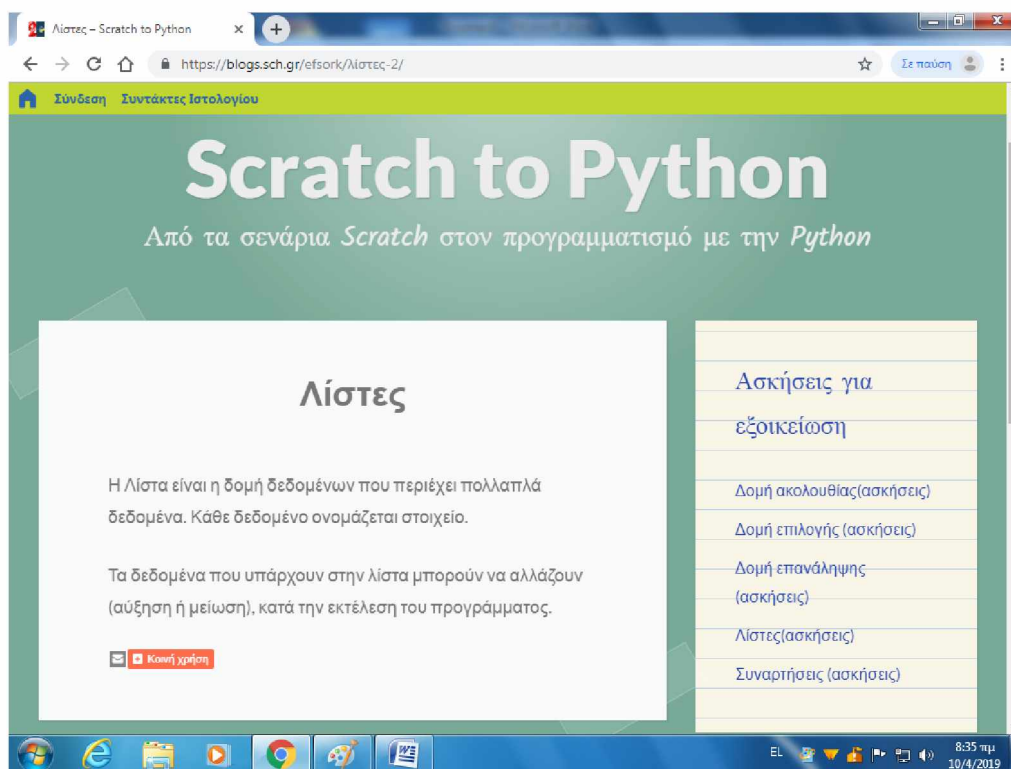
Εικόνα 5.8: Δομή Επιλογής

Στην συνέχεια του μενού **‘Ας θυμηθούμε μερικές έννοιες’**, βρίσκεται η σελίδα **‘Δομή επανάληψης’**. Στη συγκεκριμένη σελίδα παρουσιάζεται συνοπτικά η έννοια της δομής, χωρίς να αναφέρεται συγκεκριμένη γλώσσα προγραμματισμού.



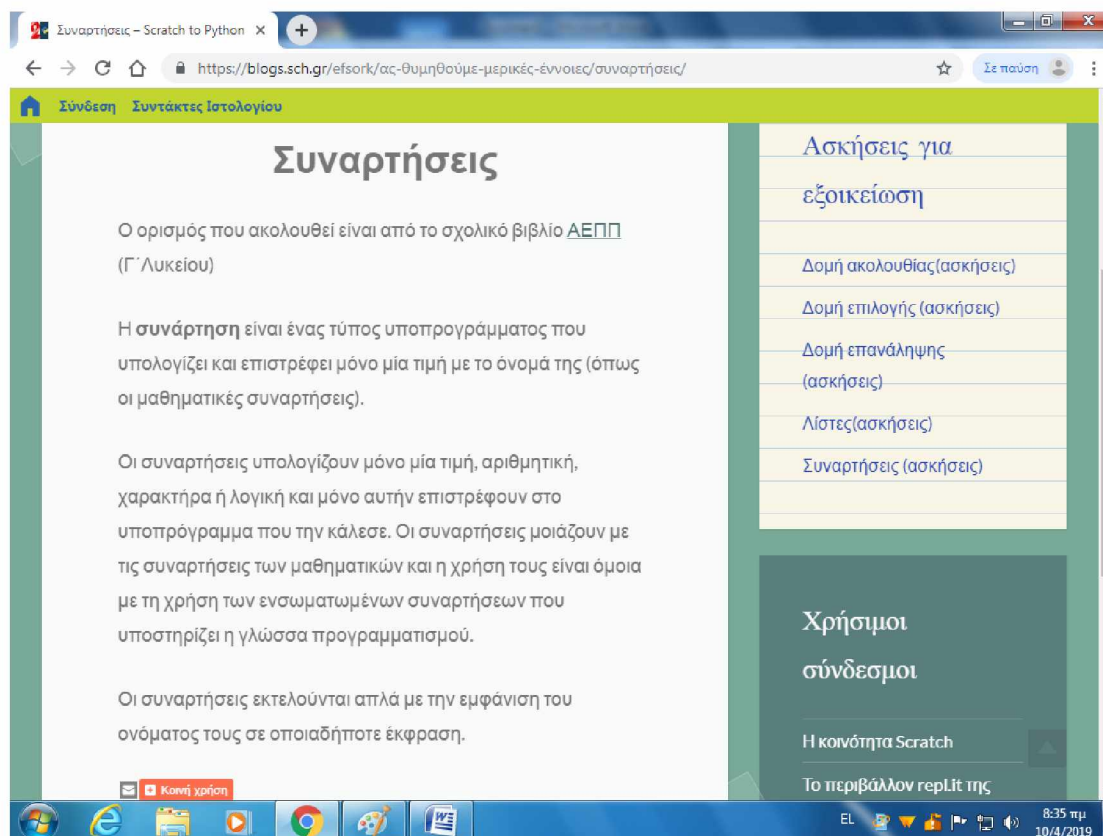
Εικόνα 5.9: Δομή Επανάληψης

Στη συνέχεια του μενού, υπάρχει το στοιχείο ‘*Λίστες*’. Στο περιεχόμενό της εμφανίζεται μια σύντομη περιγραφή της δομής .



Εικόνα 5.10: Λίστες

Το τελευταίο στοιχείο του μενού, είναι η σελίδα ‘*Συναρτήσεις*’. Στη συγκεκριμένη σελίδα αναφέρεται ένας ορισμός για τις συναρτήσεις στις γλώσσες προγραμματισμού γενικότερα. Ο ορισμός που έχει χρησιμοποιηθεί είναι από το σχολικό βιβλίο της Γ΄ Λυκείου «*Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον*».



Εικόνα 5.11: Συναρτήσεις

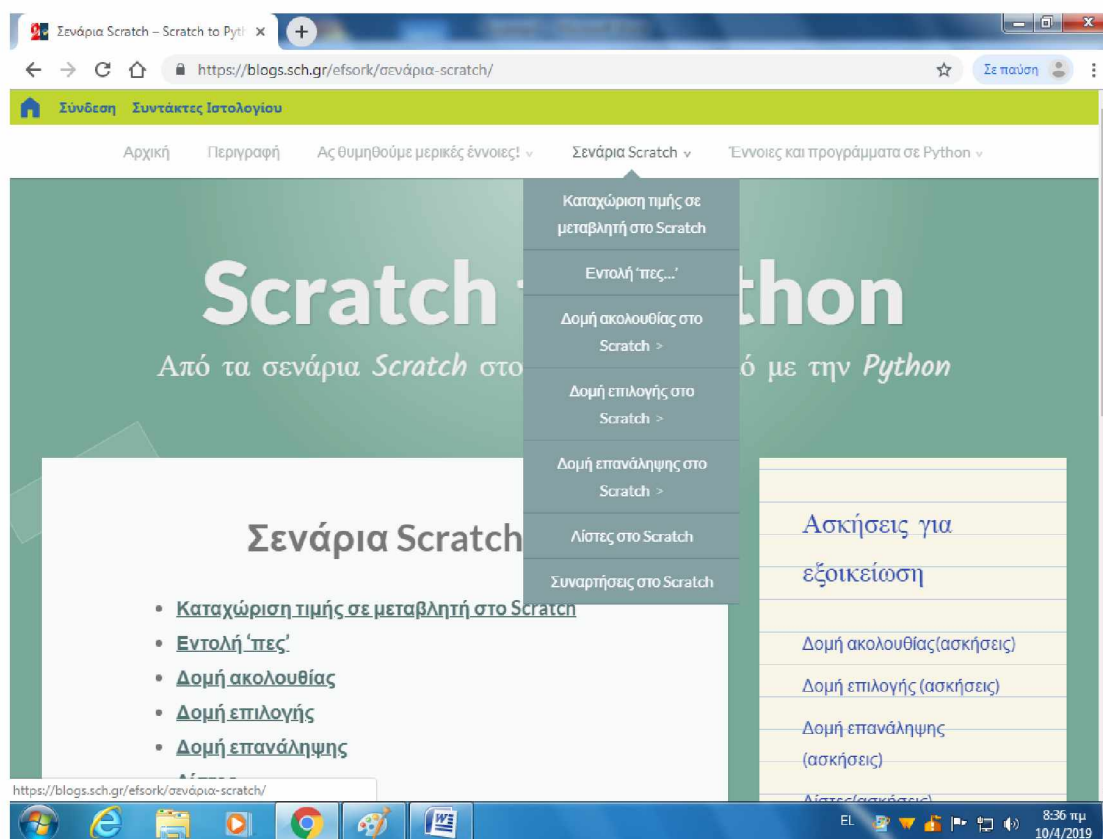
- ***Το μενού ‘Σενάρια σε Scratch’***

Το επόμενο στοιχείο του βασικού μενού του ιστολογίου, είναι το ‘*Σενάρια σε Scratch*’. Το εν λόγω στοιχείο περιέχει σελίδες οι οποίες αναφέρονται στις βασικές προγραμματιστικές δομές και πως μπορούμε να τις παρουσιάσουμε χρησιμοποιώντας έτοιμα σενάρια στο περιβάλλον του Scratch. Οι έννοιες που παρουσιάζονται και αποτελούν τα στοιχεία του ‘*Σενάρια σε Scratch*’, είναι:

- **Καταχώριση τιμής σε μεταβλητή στο Scratch**
- **Εντολή ‘πες’**
- **Δομή ακολουθίας στο Scratch**

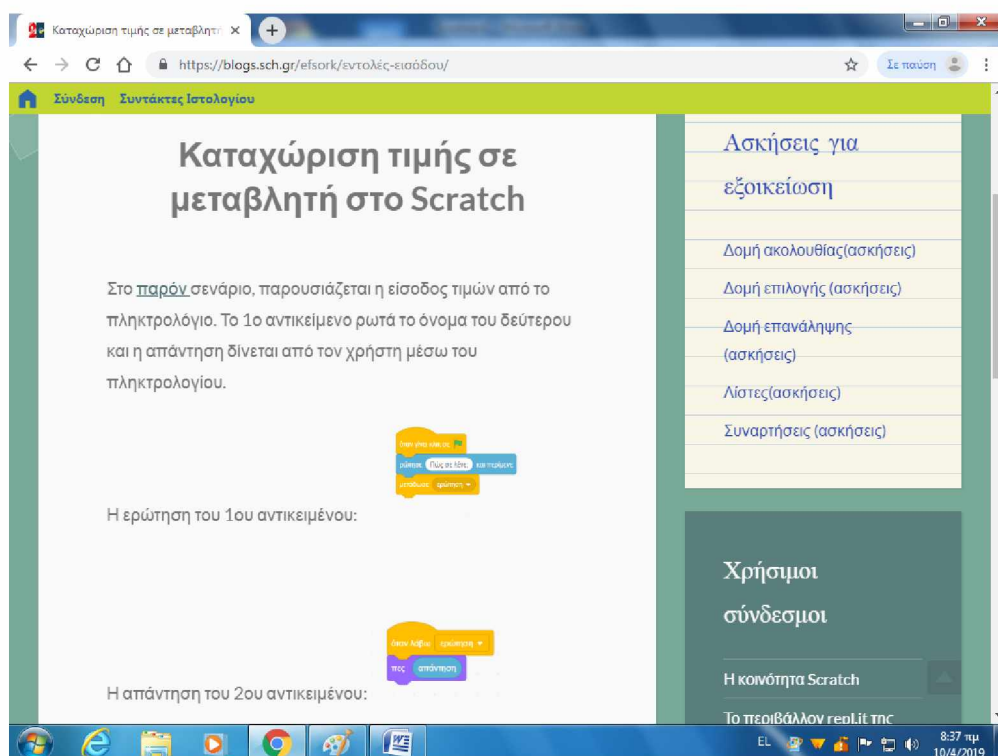
- Δομή επιλογής στο Scratch
- Δομή επανάληψης στο Scratch
- Λίστες στο Scratch
- Συναρτήσεις στο Scratch

Στην επόμενη εικόνα, εμφανίζεται το αποτέλεσμα στην οθόνη όταν επιλέγουμε το στοιχείο ‘Σενάρια σε Scratch’.



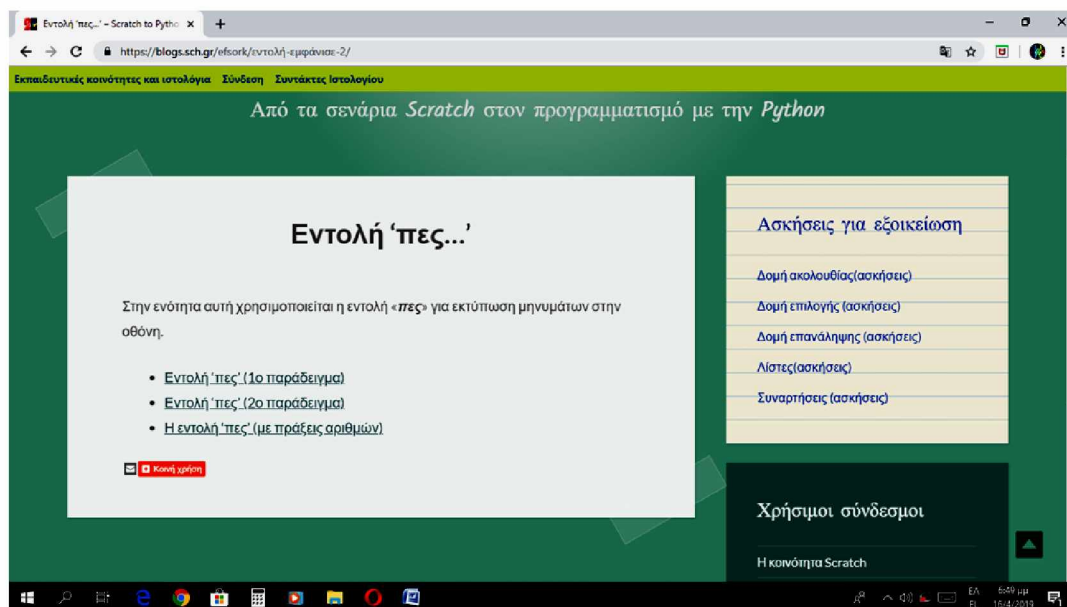
Εικόνα 5.12: Σενάρια σε Scratch

Στο πρώτο στοιχείο του προηγούμενου μενού εμφανίζεται η σελίδα ‘**Καταχώριση τιμής σε μεταβλητή στο Scratch**’. Στη σελίδα αυτή υπάρχει ένα απλό παράδειγμα καταχώρισης τιμής σε μεταβλητή, με είσοδο τιμών από το πληκτρολόγιο. Το πρώτο αντικείμενο ρωτά το όνομα του δεύτερου και η απάντηση δίνεται από τον χρήστη μέσω του πληκτρολογίου. Στην εικόνα που ακολουθεί εμφανίζεται ένα στιγμιότυπο οθόνης του προαναφερόμενου.



Εικόνα 5.13: Καταχώριση τιμής σε μεταβλητή στο Scratch

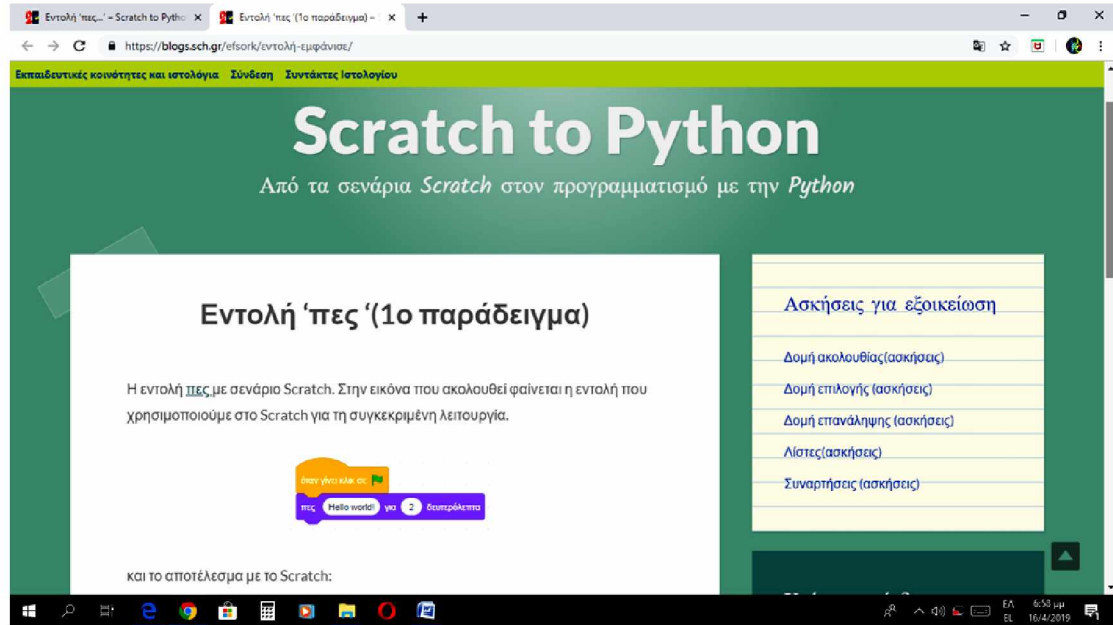
Έπειτα το στοιχείο που ακολουθεί είναι η σελίδα 'Εντολή 'πες...'', η οποία εμφανίζεται στη συνέχεια.



Εικόνα 5.14: Εντολή 'πες...'

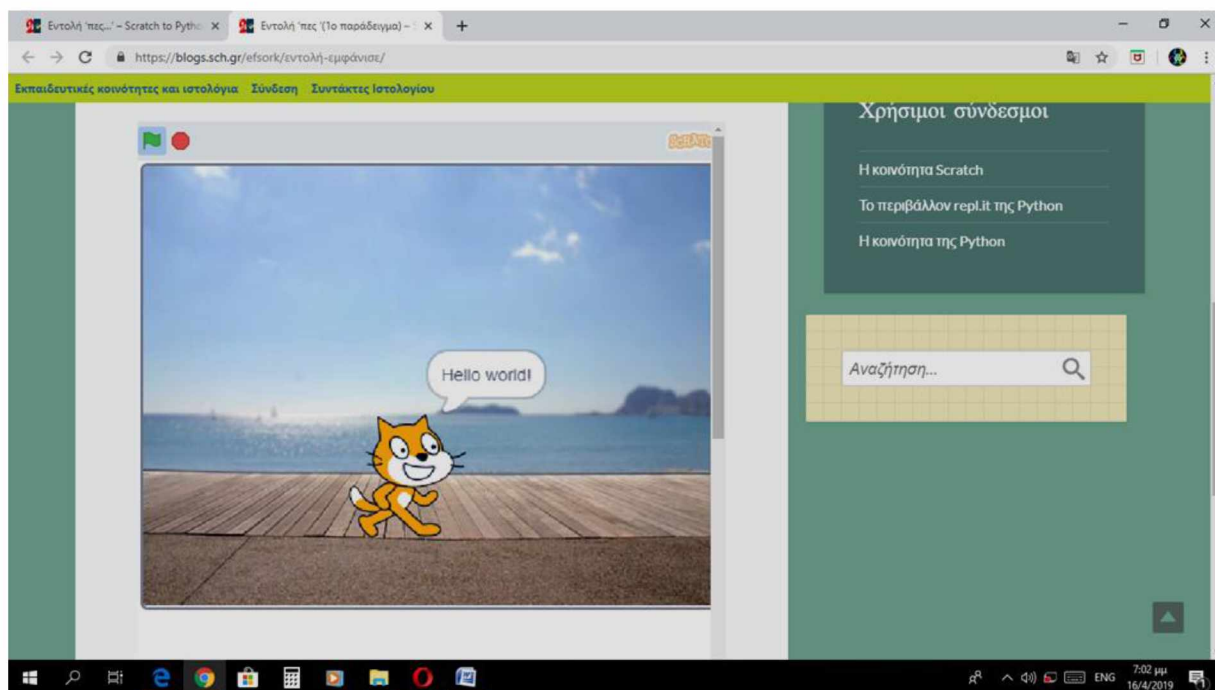
Η εντολή 'πες...', χρησιμοποιείται στο Scartch για την εκτύπωση μηνυμάτων στην οθόνη και συνήθως είναι μηνύματα που τα εκφέρει ένα αντικείμενο. Στη σελίδα αυτή,

υπάρχουν τρία παραδείγματα για την εντολή ‘πες...’. Η σελίδα του πρώτου παραδείγματος εμφανίζεται στη συνέχεια.



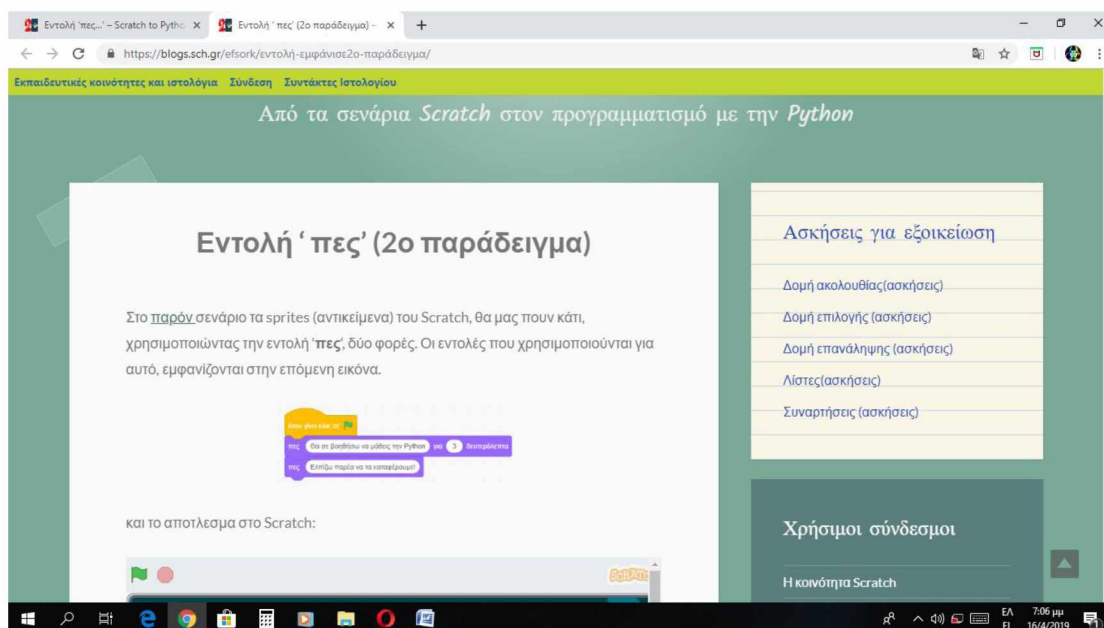
Εικόνα 5.15: Εντολή ‘πες...’ (1^ο παράδειγμα)

Στο συγκεκριμένο παράδειγμα, χρησιμοποιείται η εντολή ‘πες...’ για το γνωστό μας ‘Hello world’. Το αποτέλεσμα της εκτέλεσης του σεναρίου υπάρχει στη σελίδα σαν ενσωματωμένο βίντεο αλλά και ολόκληρο το σενάριο αποθηκευμένο στην κοινότητα του Scratch.



Εικόνα 5.16: Εκτέλεση σεναρίου με την εντολή 'πες...'(1^ο)

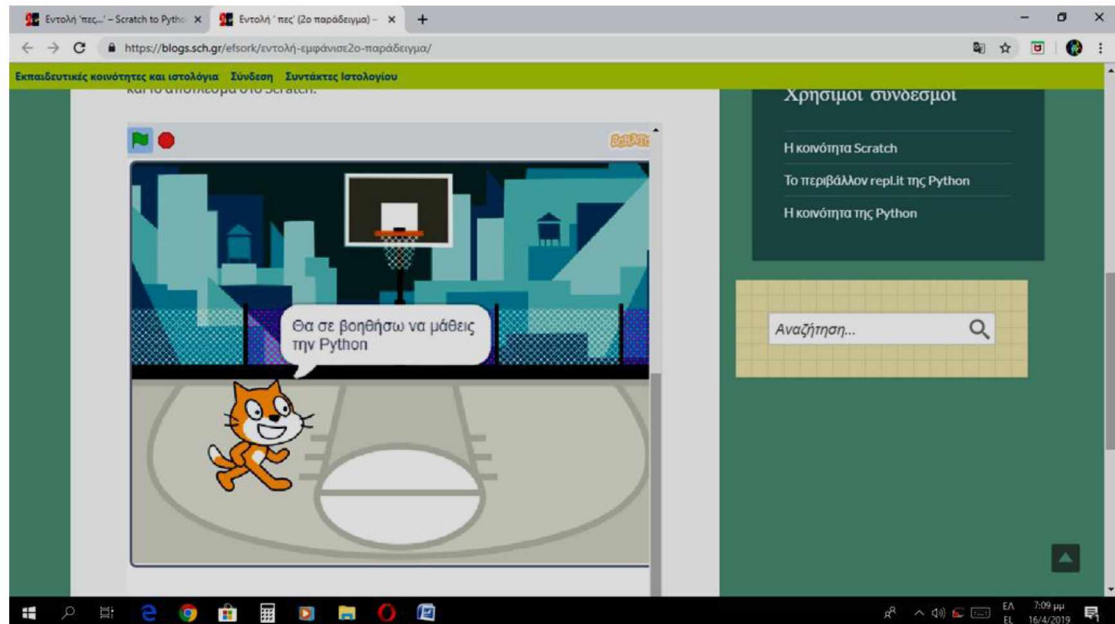
Το δεύτερο παράδειγμα της εντολή 'πες...' παρουσιάζεται στην επόμενη εικόνα



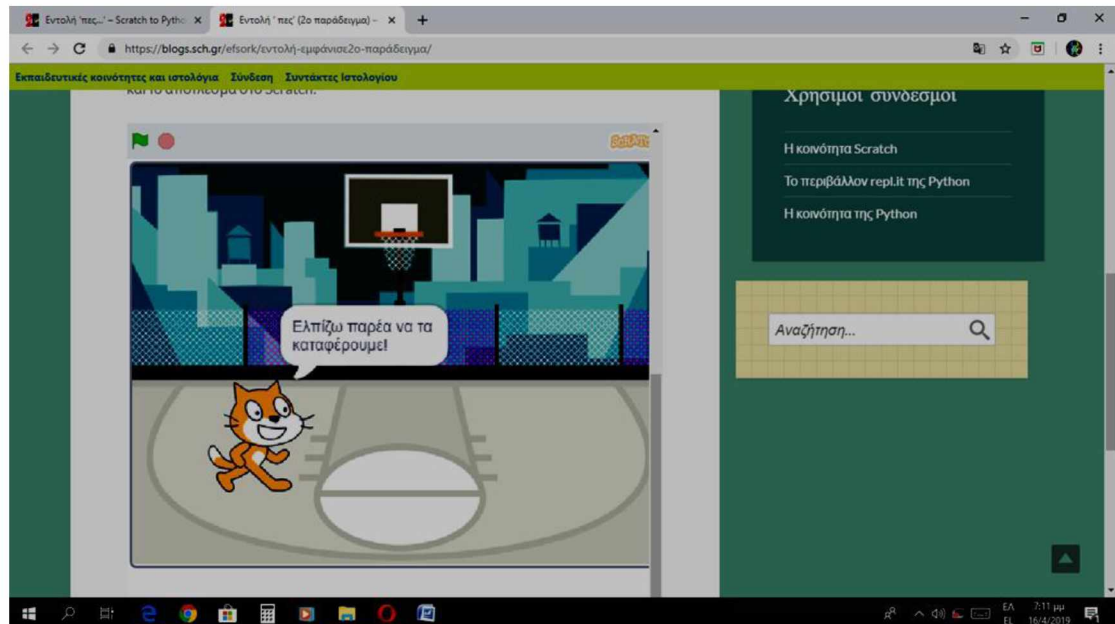
Εικόνα 5.17: Εντολή 'πες...' (2^ο)

Στο συγκεκριμένο παράδειγμα χρησιμοποιείται η εντολή 'πες...' διαδοχικά για το ίδιο αντικείμενο. Για να μπορέσει να εμφανιστεί το δεύτερο μήνυμα θα πρέπει το πρώτο μήνυμα να έχει καθορισμένη χρονική διάρκεια σε δευτερόλεπτα. Η εκτέλεση

του σεναρίου υπάρχει και σε αυτό το παράδειγμα σαν ενσωματωμένο βίντεο στην σελίδα.

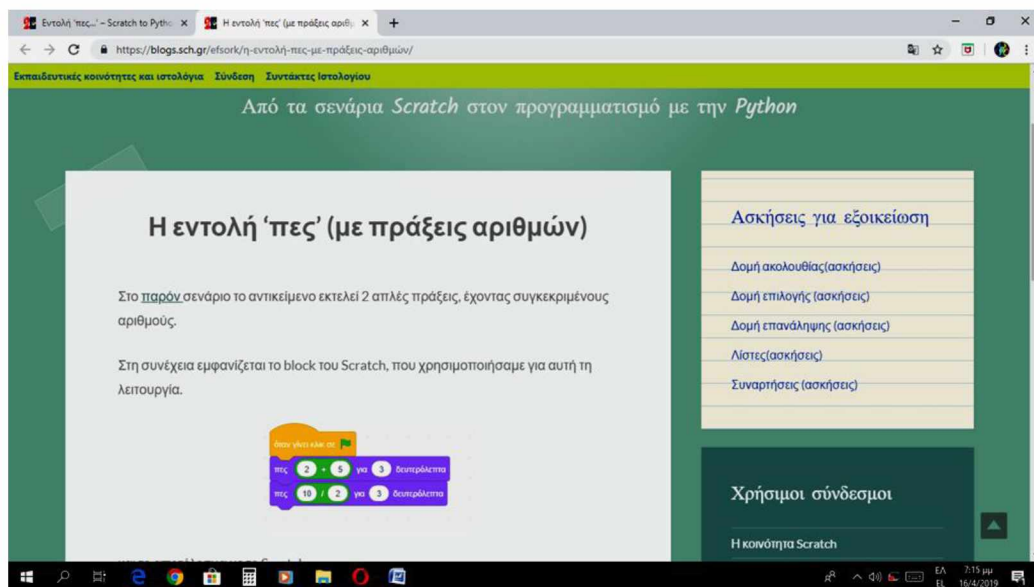


Εικόνα 5.18: Εμφάνιση 1^{ου} μηνύματος



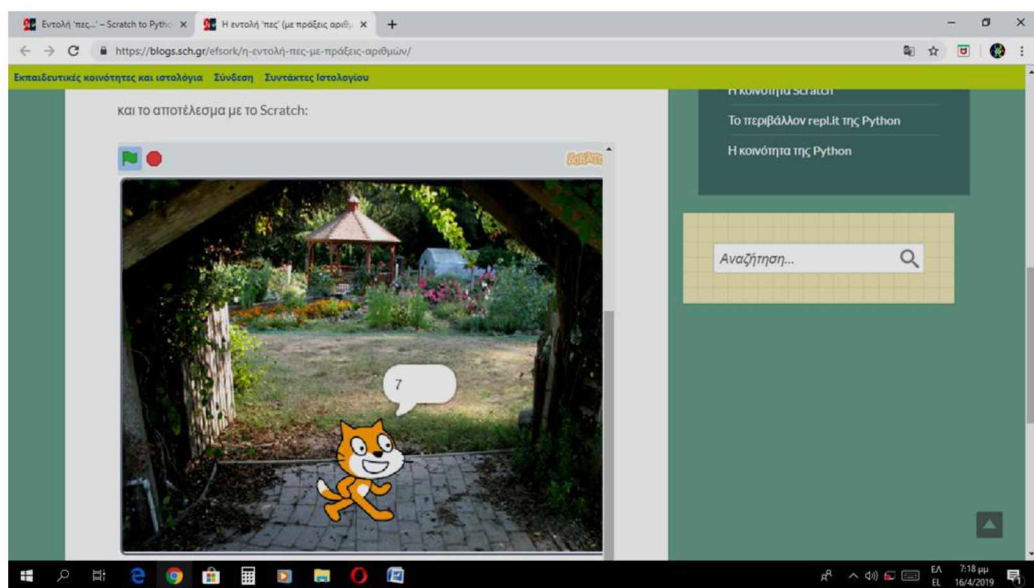
Εικόνα 5.19: Εμφάνιση 2^{ου} μηνύματος

Ακολουθεί το τρίτο παράδειγμα της εντολής ‘πες...’. Στη σενάριο αυτό οι διαδοχικές εντολές ‘πες...’, περιέχουν πράξεις αριθμών. Σαν αποτέλεσμα αυτό έχει το αντικείμενο να εκφέρει το αποτέλεσμα των πράξεων.

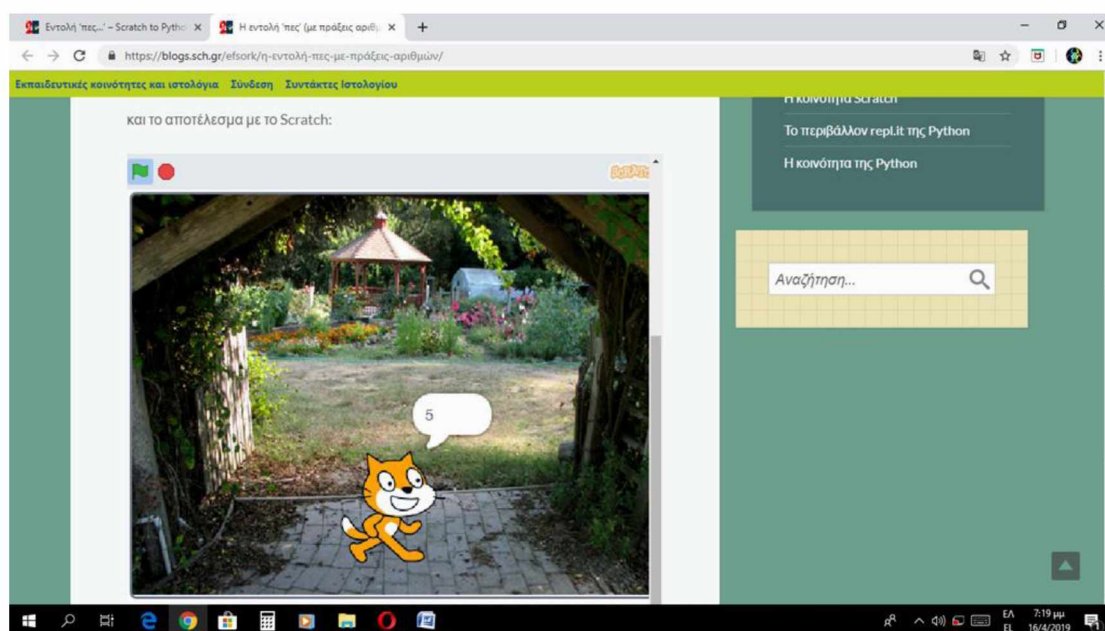


Εικόνα 5.20: Η εντολή ‘πες..’ (με πράξεις αριθμών)

Επίσης στο συγκεκριμένο παράδειγμα, το αποτέλεσμα της εκτέλεσης του σεναρίου, βρίσκεται ενσωματωμένο στη σελίδα του παραδείγματος, όπως εικονίζεται στη συνέχεια.

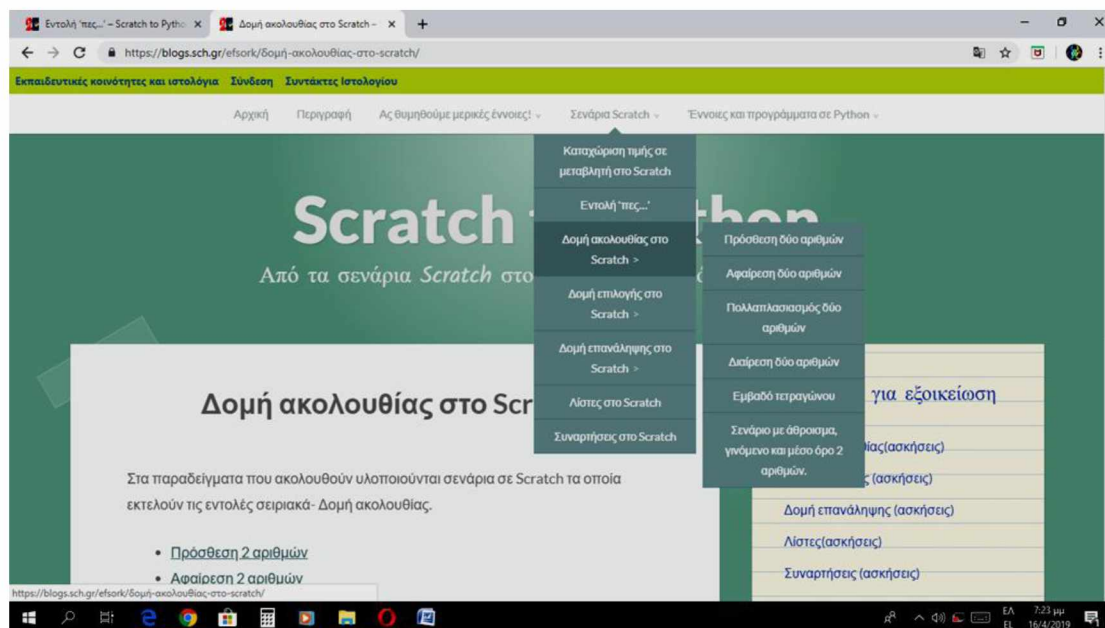


Εικόνα 5.21: Η εντολή ‘πες...’ με πράξεις (1^η)



Εικόνα 5.22: Η εντολή ‘πες...’ με πράξεις (2^η)

Το επόμενο στοιχείο του μενού ‘*Σενάρια σε Scratch*’, είναι η σελίδα ‘*Δομή ακολουθίας*’ η οποία εμφανίζεται στην επόμενη εικόνα.



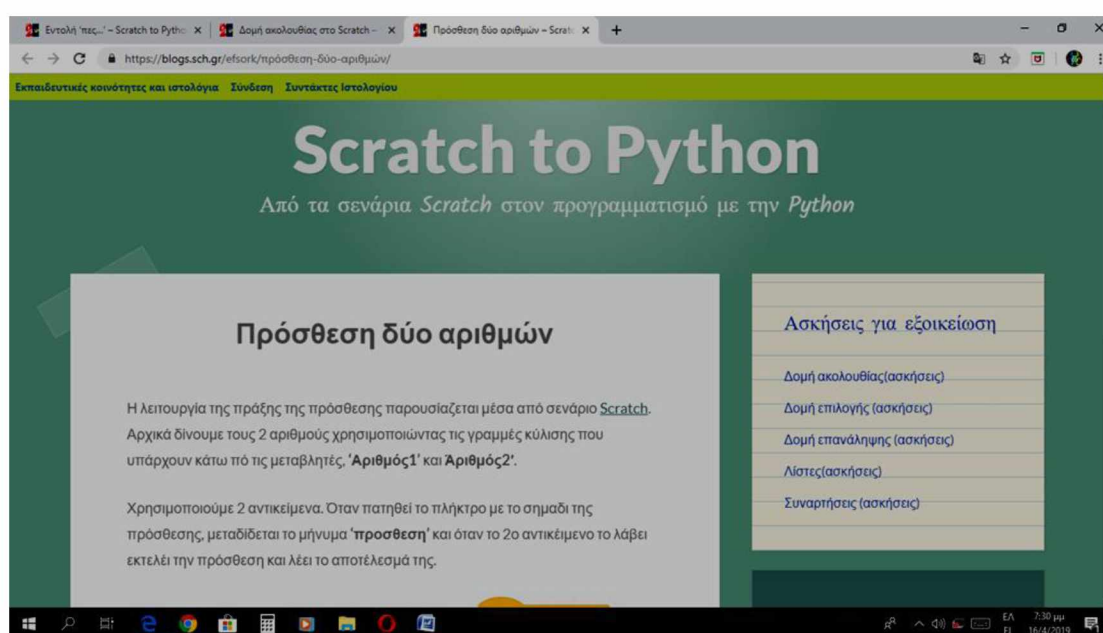
Εικόνα 5.23: Δομή ακολουθίας

Όπως έχει προαναφερθεί η ευκολότερη δομή των δομών προγραμματισμού, είναι η δομή ακολουθίας. Η συγκεκριμένη σελίδα περιέχει έξι παραδείγματα:

- Πρόσθεση δύο αριθμών

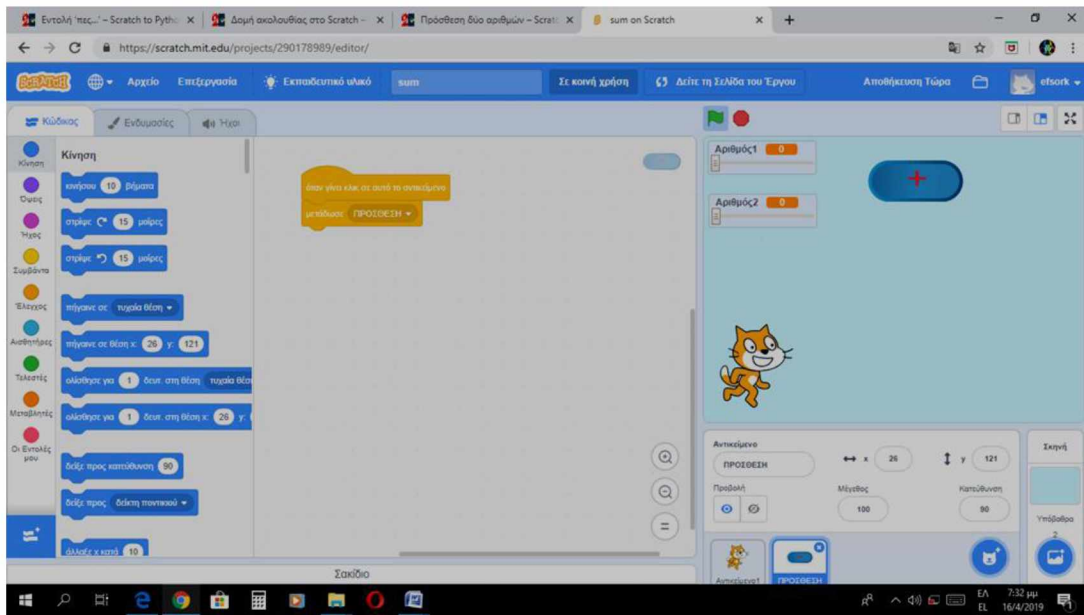
- Αφαίρεση δύο αριθμών
- Πολλαπλασιασμός δύο αριθμών
- Διαίρεση δύο αριθμών
- Εμβασό τετραγώνου
- Σενάριο με άθροισμα, γινόμενο και μέσο όρο δύο αριθμών

Στο πρώτο παράδειγμα της πρόσθεσης αριθμών με το Scratch, χρησιμοποιούνται δύο αντικείμενα. Η σελίδα του παραδείγματος είναι η επόμενη εικόνα.

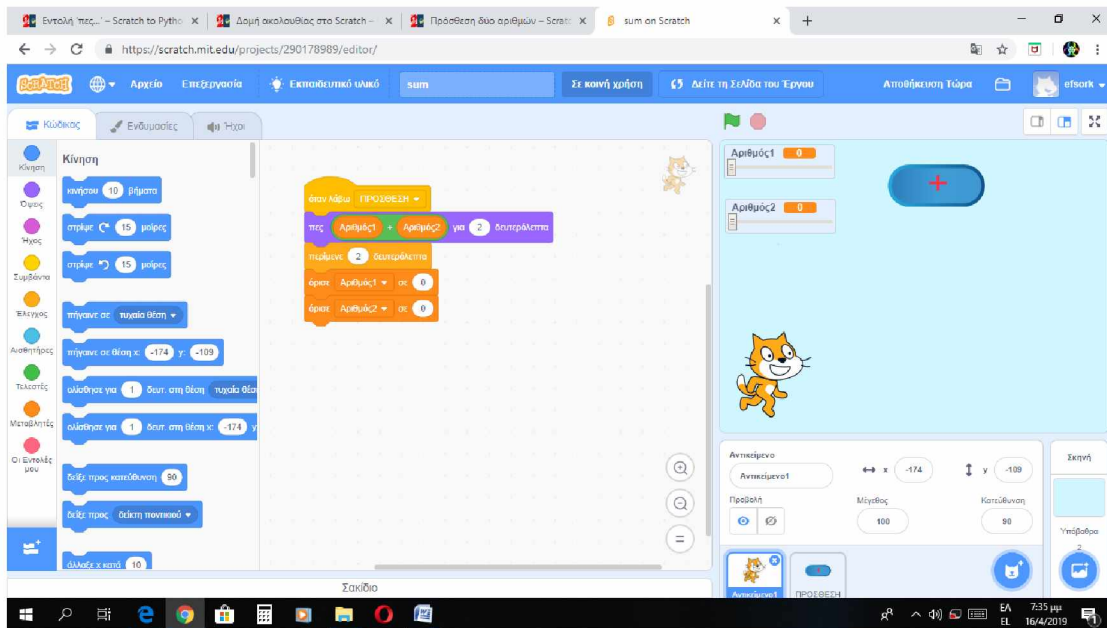


Εικόνα 5.24: Πρόσθεση δύο αριθμών

Στη συνέχεια εμφανίζονται οι εντολές των δύο αντικειμένων στο περιβάλλον του Scratch.

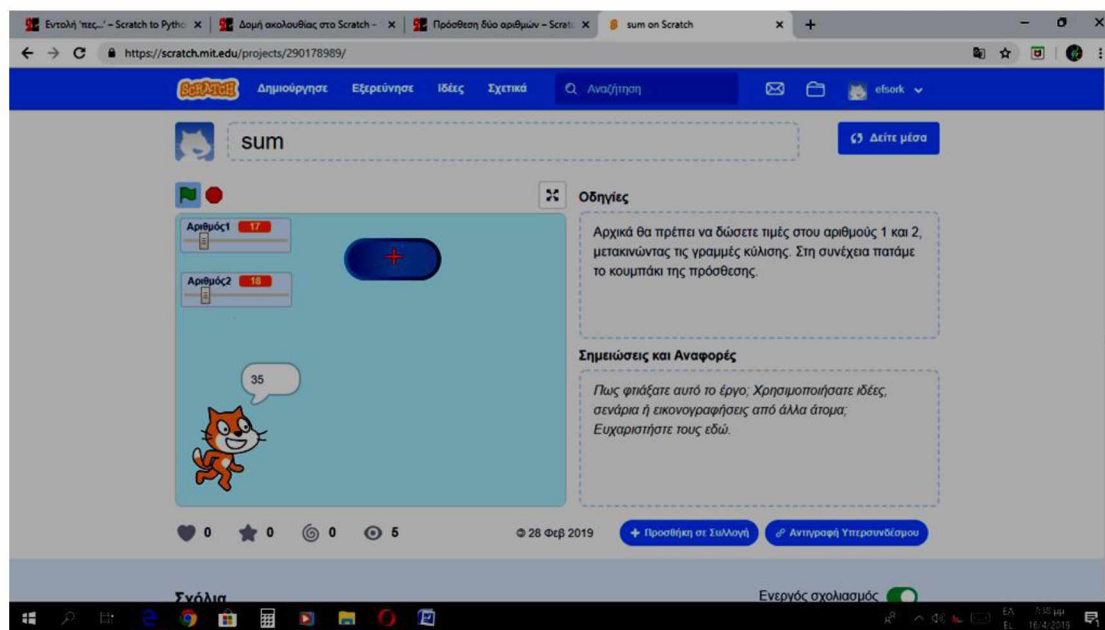


Εικόνα 5.25: Εντολές στο Scratch 1^{ου} αντικειμένου



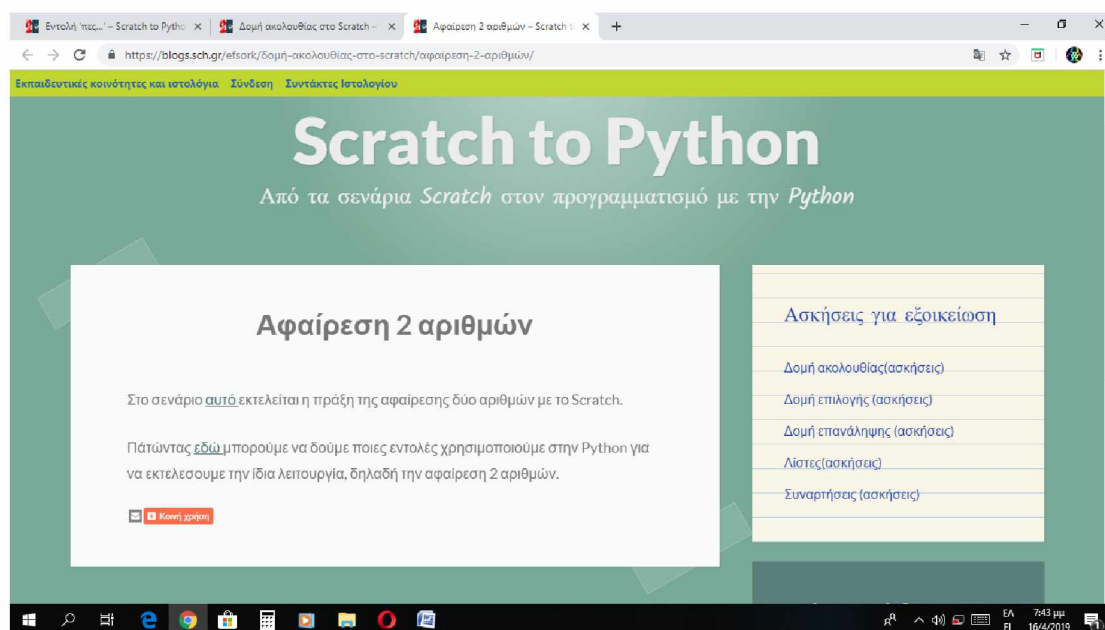
Εικόνα 5.26: Εντολές στο Scratch 2^{ου} αντικειμένου

Δίνοντας τιμές ο χρήστης στις δύο μεταβλητές που θα προστεθούν, χρησιμοποιώντας την μπάρα κύλισης κάτω από τη καθεμία, και πατώντας το αντικείμενο με το σύμβολο της πρόσθεσης, το αντικείμενο γάτα εκφέρει το αποτέλεσμα της πρόσθεσης, όπως στην εικόνα που ακολουθεί.



Εικόνα 5.27: Αποτέλεσμα της πρόσθεσης

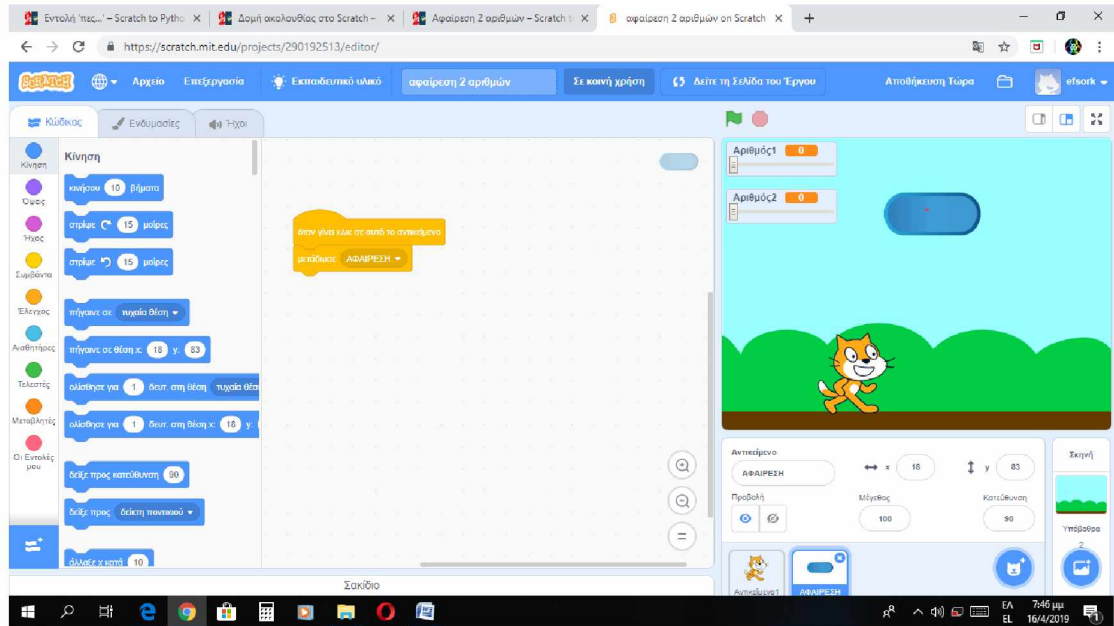
Το δεύτερο παράδειγμα της δομής ακολουθίας είναι η πράξη της αφαίρεσης μεταξύ δύο αριθμών, το οποίο απεικονίζεται στη συνέχεια.



Εικόνα 5.28: Αφαίρεση δύο αριθμών

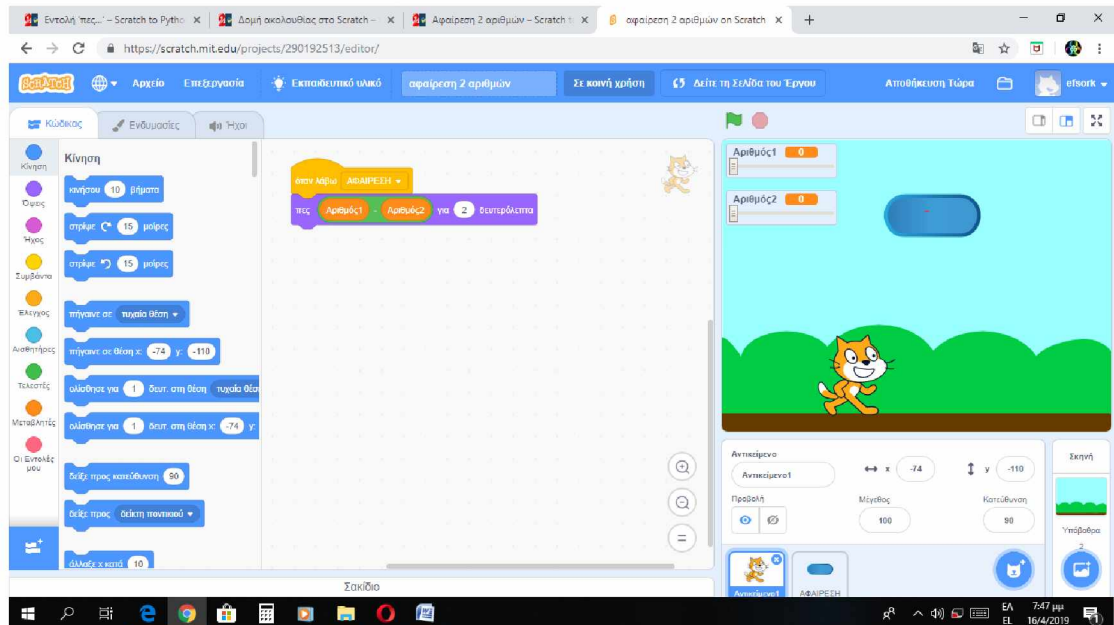
Χρησιμοποιώντας τον υπερσύνδεσμο της σελίδας, μεταφερόμαστε στο περιβάλλον του Scratch, στα αντικείμενα και τις εντολές που χρησιμοποιήθηκαν.

Οι εντολές του πρώτου αντικειμένου είναι:



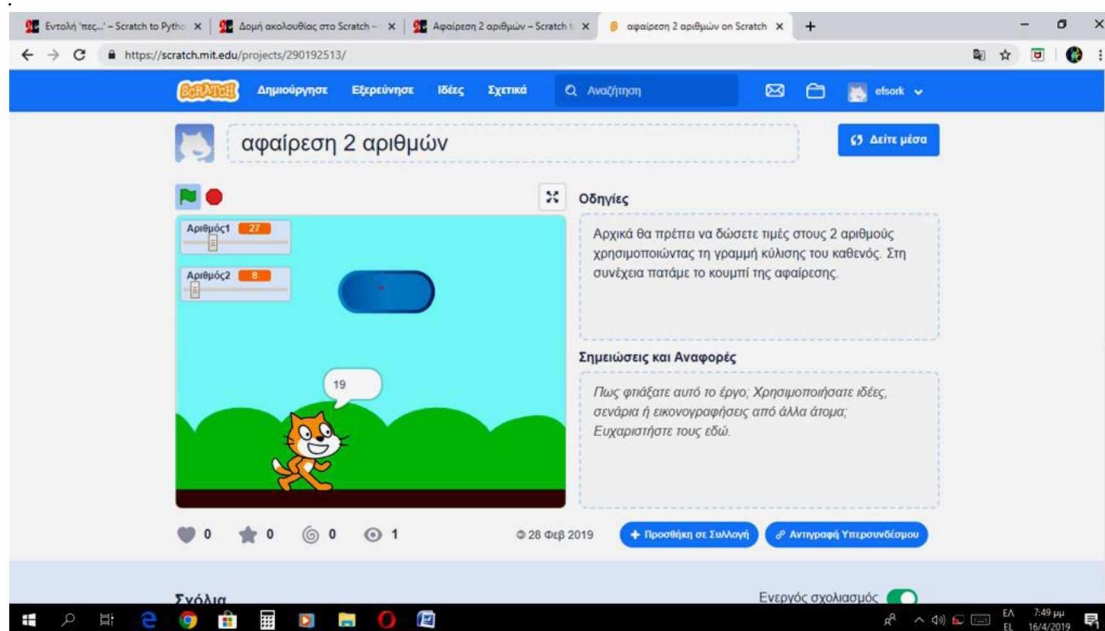
Εικόνα 5.29: Εντολές 1^{ου} αντικειμένου

Ενώ οι εντολές του δεύτερου αντικειμένου είναι:



Εικόνα 5.30: Εντολές 2^{ου} αντικειμένου

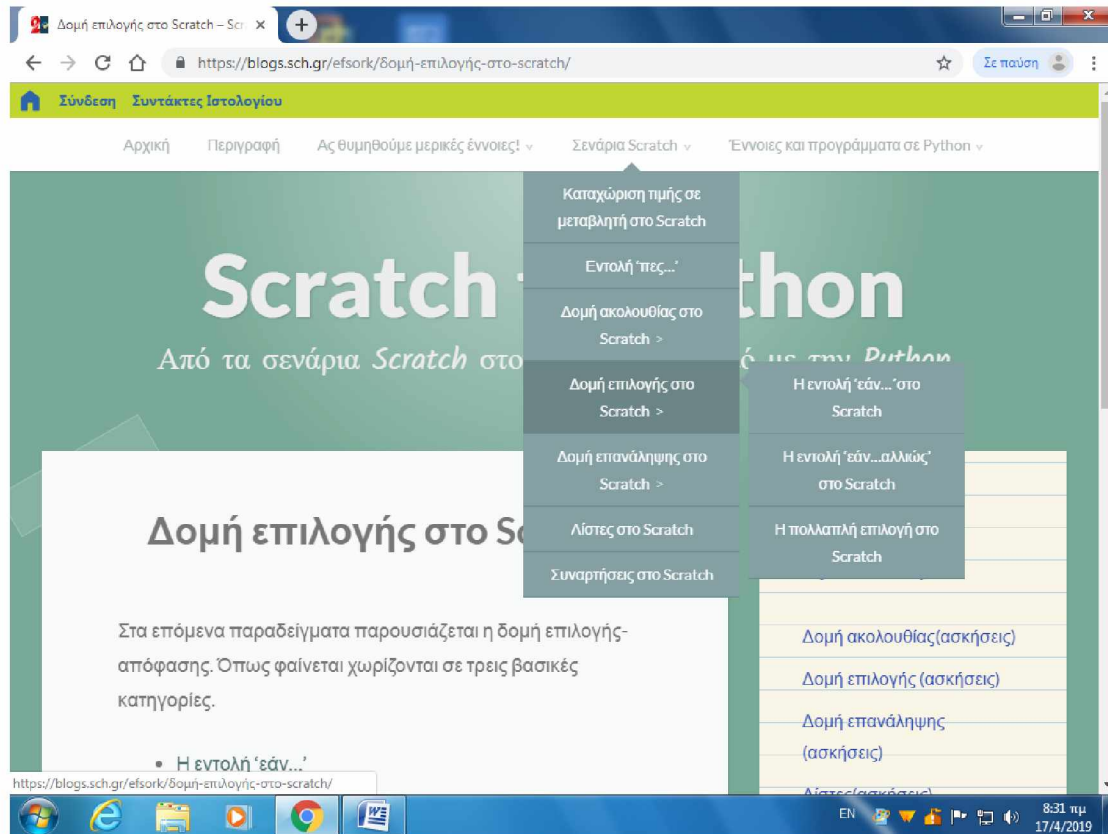
Εκτελώντας το σενάριο αφού πρώτα δοθούν τιμές στις δύο μεταβλητές το αποτέλεσμα φαίνεται στη συνέχεια και πατώντας το αντικείμενο με το σύμβολο της αφαίρεσης.



Εικόνα 5.31: Αφαίρεση δύο αριθμών

Με παρόμοιο τρόπο λειτουργούν και τα σεναρία που ακολουθούν και υπάρχουν στο μενού 'Δομή ακολουθίας', ο πολλαπλασιασμός, η διαίρεση το εμβαδό τετραγώνου και το άθροισμα, γινόμενο και μέσος όρος δύο αριθμών.

Το επόμενο στοιχείο που ακολουθεί στο μενού 'Σενάρια σε Scratch', είναι η 'Δομή επιλογής'. Στην εικόνα που ακολουθεί παρουσιάζεται η οθόνη που προκύπτει επιλέγοντας το εν λόγω στοιχείο.

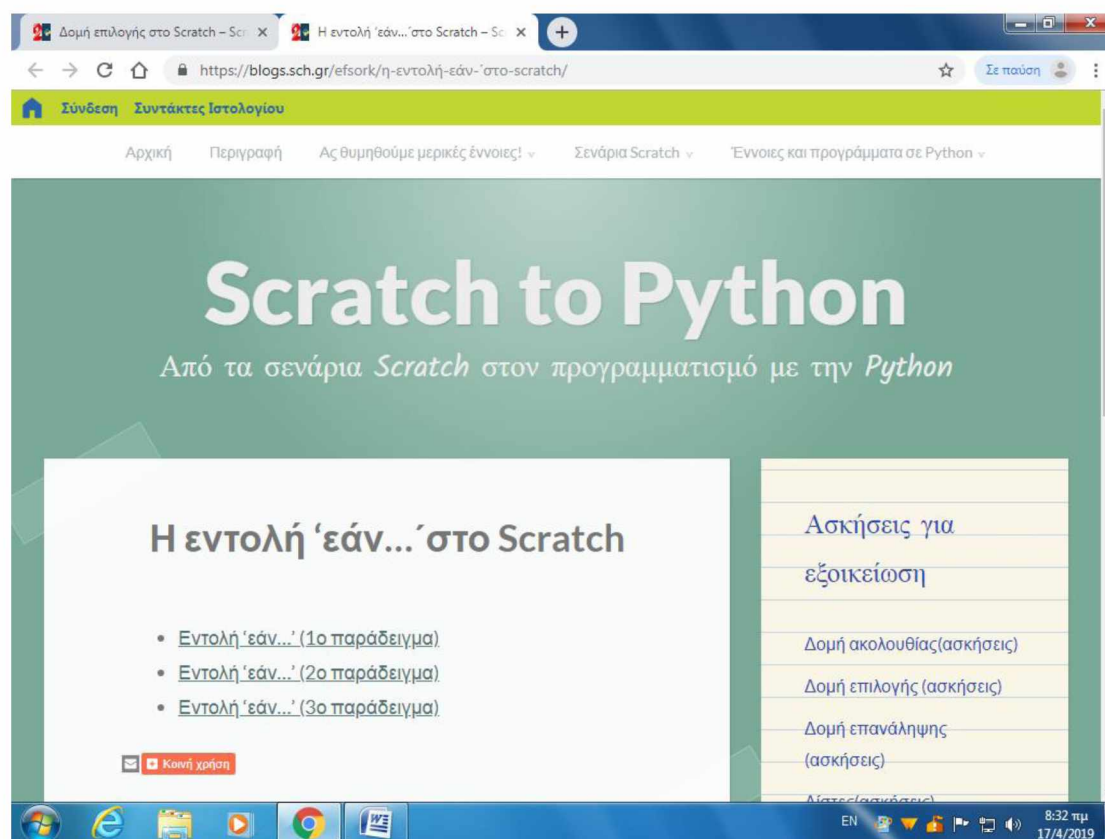


Εικόνα 5.32: Δομή ακολουθίας

Τα περιεχόμενα στοιχεία της ‘Δομής ακολουθίας’ είναι:

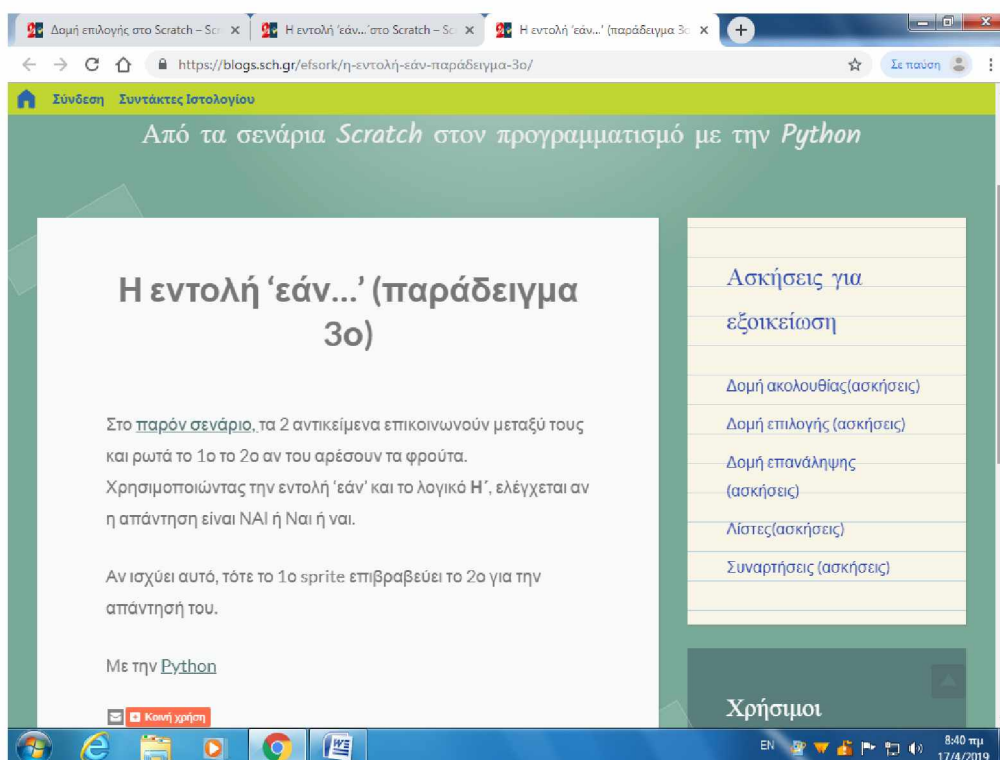
- **Η εντολή ‘εάν...’ στο Scratch**
- **Η εντολή ‘εάν...αλλιώς’ στο Scratch**
- **Η πολλαπλή επιλογή στο Scratch**

Επιλέγοντας το πρώτο στοιχείο, εμφανίζονται τρία παραδείγματα της απλής επιλογής υλοποιημένα σε Scratch και σε Python, όπως στην επόμενη εικόνα.



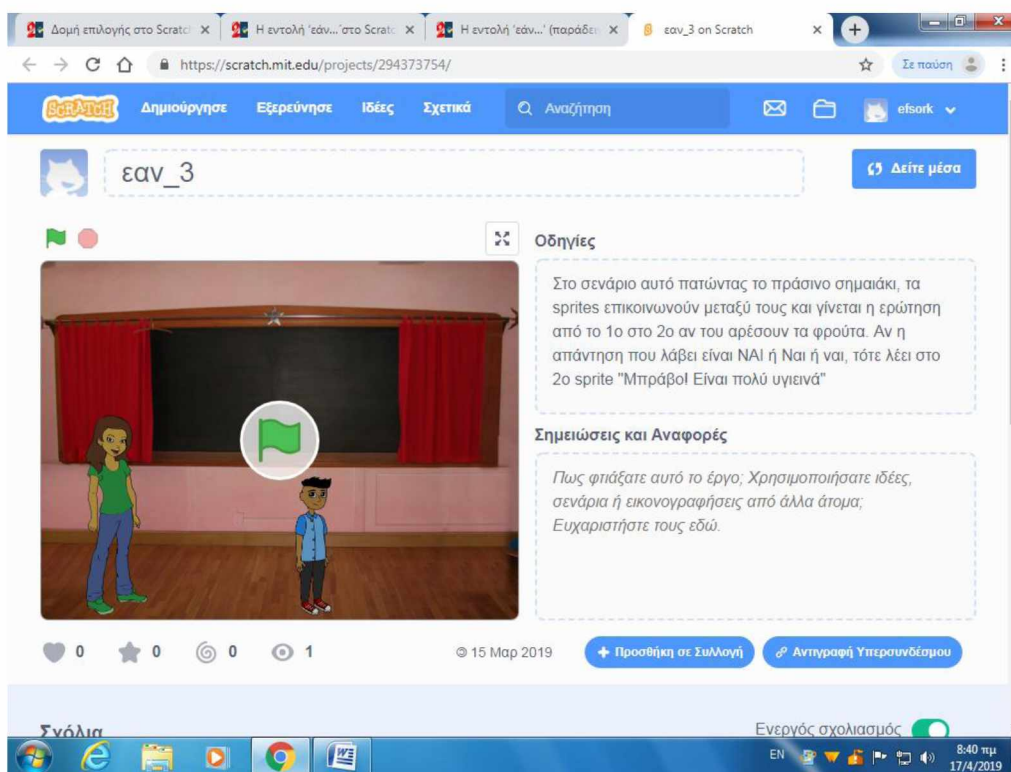
Εικόνα 5.33: Η εντολή 'εάν...' στο Scratch

Επιλέγουμε να περιγράψουμε το τρίτο παράδειγμα το οποίο περιέχει δύο αντικείμενα. Το πρώτο αντικείμενο ρωτά το δεύτερο αν του αρέσουν τα φρούτα, αν η απάντηση που θα πάρει είναι 'ΝΑΙ' ή 'Ναι' ή 'ναι', τότε το δεύτερο αντικείμενο επιβραβεύεται για την απάντησή του. Για να επιτευχθεί αυτό, χρησιμοποιείται η εντολή 'εάν...' και ο λογικός τελεστής 'Η'. Στην επόμενη εικόνα εμφανίζεται η σελίδα του παραδείγματος.



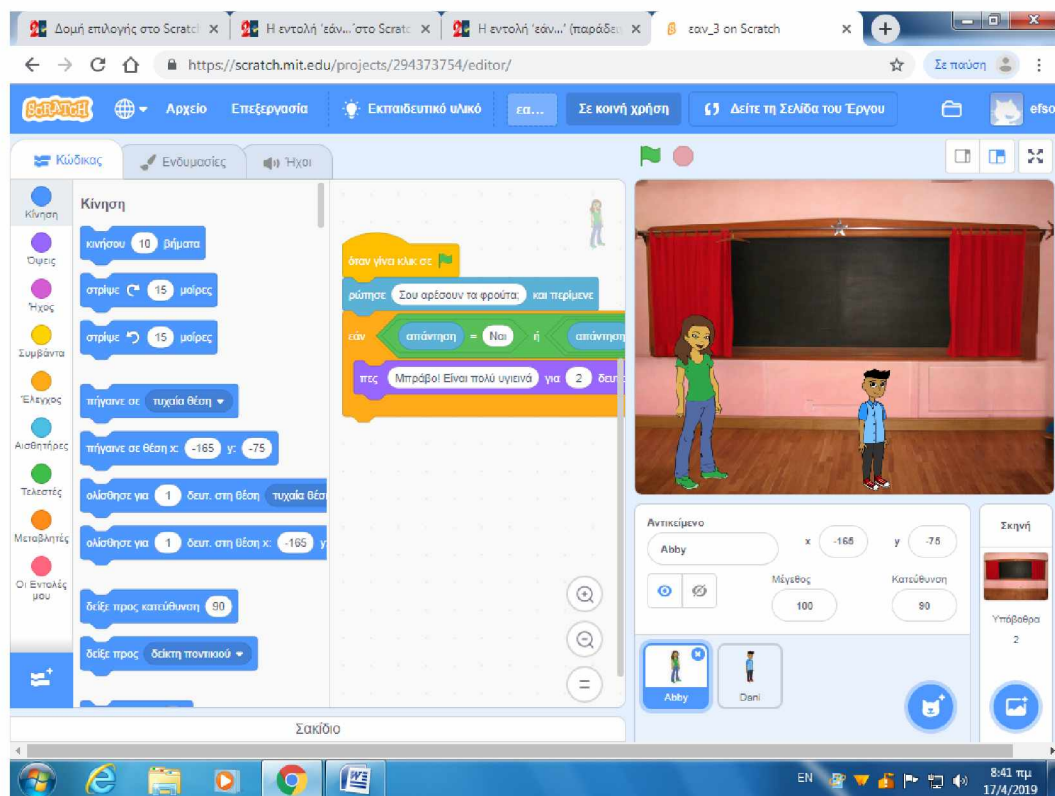
Εικόνα 5.34: Η εντολή ‘εάν...’ (παράδειγμα 3^ο)

Έπειτα παρουσιάζεται το παράδειγμα στο περιβάλλον του Scratch, στο οποίο περιέχεται και μια περιγραφή του σεναρίου.

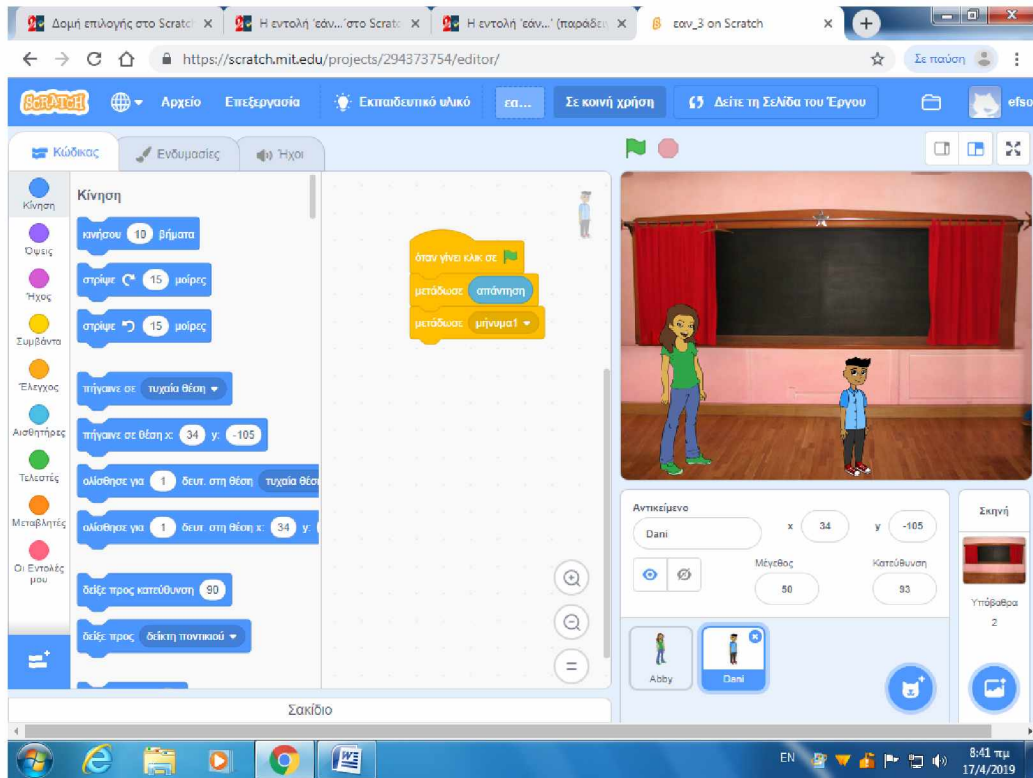


Εικόνα 5.35: Το περιβάλλον Scratch, εντολή ‘εάν..’ (παράδειγμα 3^ο)

Στη συνέχεια εμφανίζονται οι εντολές που χρησιμοποιήθηκαν για τα δύο αντικείμενα και το αποτέλεσμα της εκτέλεσης του σεναρίου.

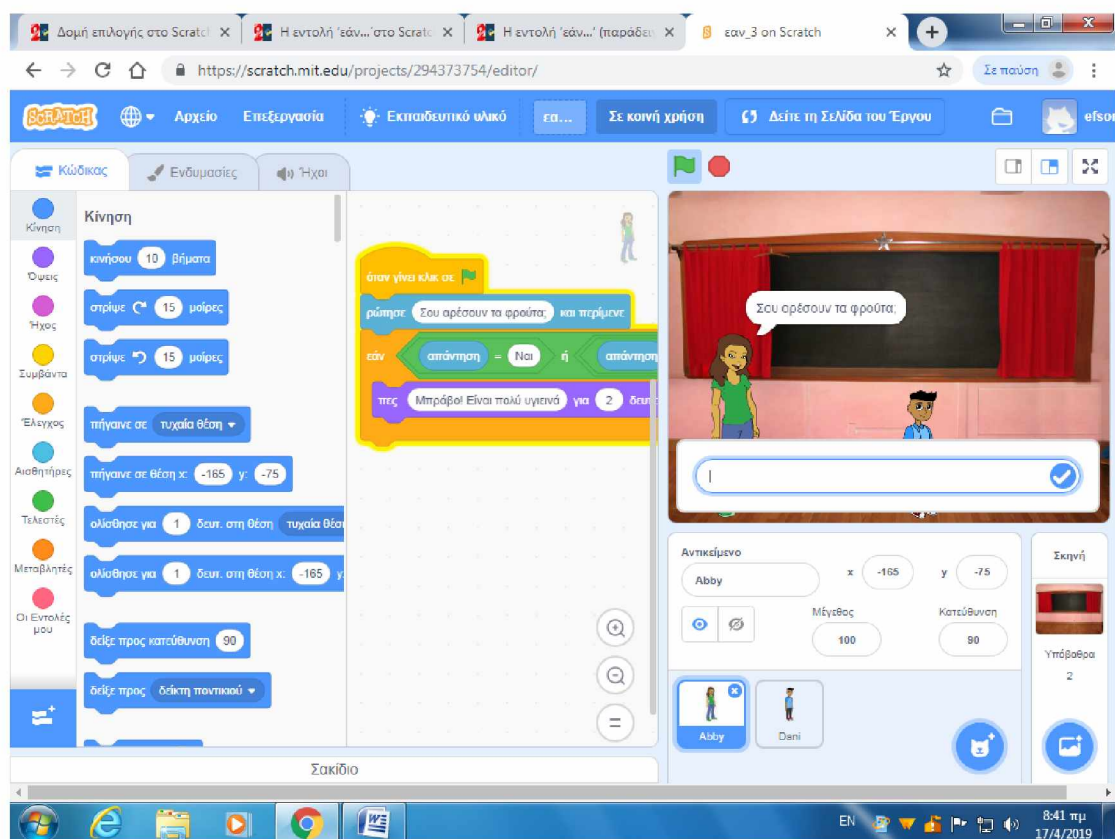


Εικόνα 5.36: Εντολές 1^{ου} αντικειμένου, εντολή 'εάν...' (παράδειγμα 3^ο)



Εικόνα 5.37: Εντολές 2^{ου} αντικειμένου, εντολή ‘εάν...’ (παράδειγμα 3^ο)

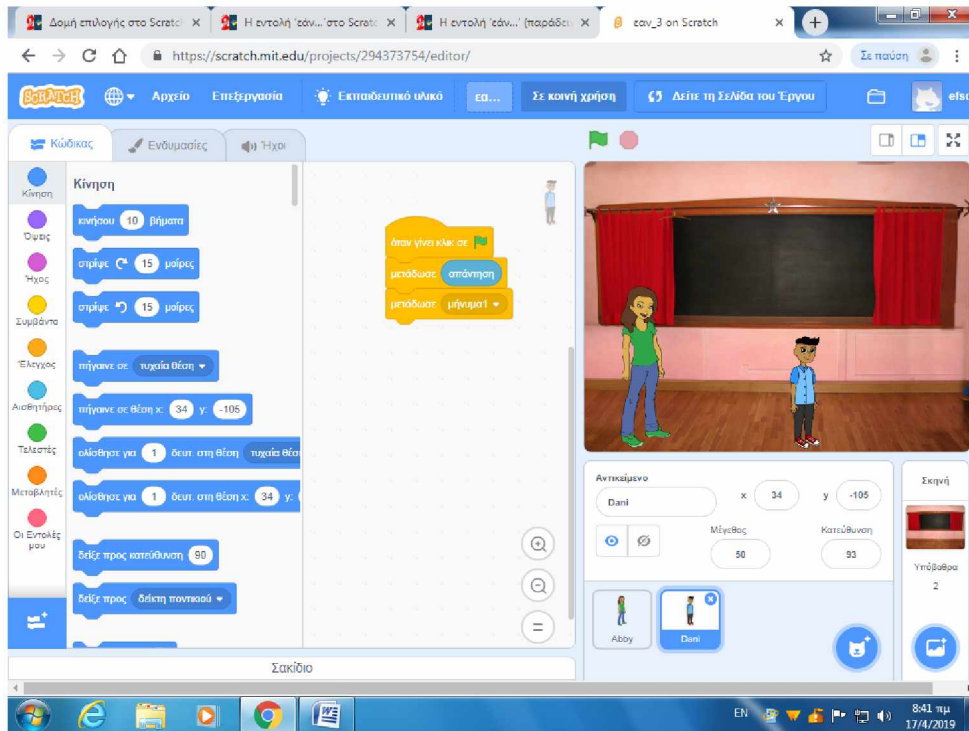
Στη συνέχεια κατά την εκτέλεση του σεναρίου, παρουσιάζονται οι οθόνες που προκύπτουν και η ‘επικοινωνία’, ανάμεσα στα δύο αντικείμενα. Στην εικόνα που ακολουθεί εμφανίζεται η ερώτηση του πρώτου αντικειμένου στο δεύτερο για το εάν του αρέσουν τα φρούτα.



Εικόνα 5.38: Ερώτηση 1^{ου} αντικείμενου, εντολή 'εάν...', (παράδειγμα 3^ο)

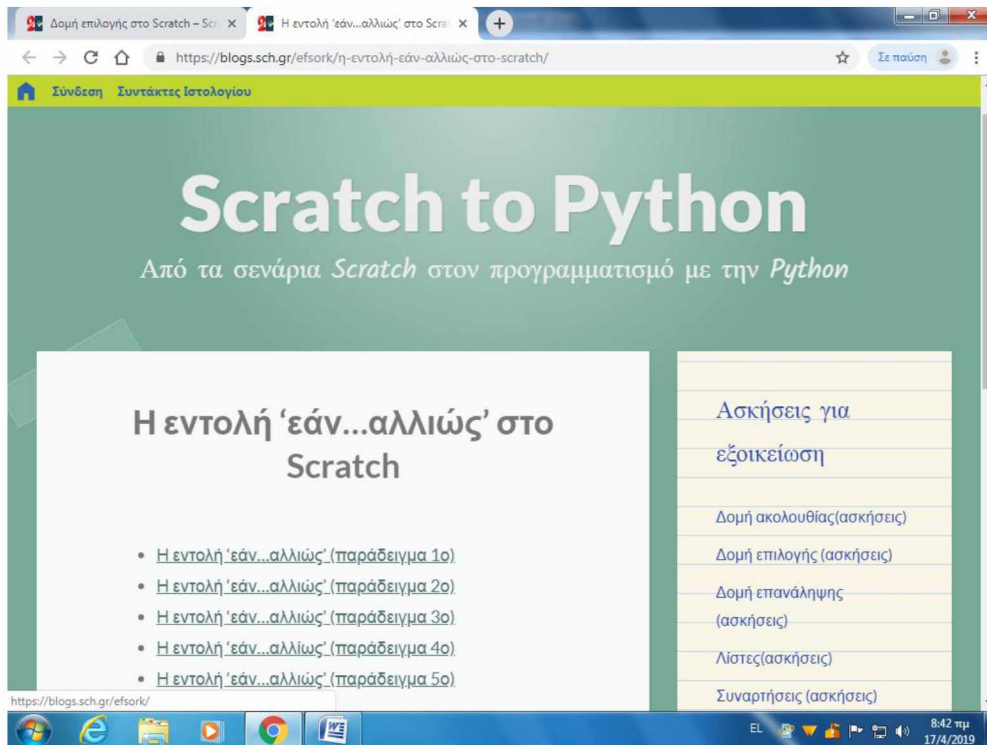
Όταν το δεύτερο αντικείμενο λάβει την ερώτηση, ο χρήστης πληκτρολογεί την απάντηση. Σε περίπτωση που η απάντηση που δοθεί είναι 'ΝΑΙ' ή 'Ναι' ή 'ναι', τότε το αντικείμενο που ρωτά επιβραβεύει εκείνο που απαντά.

Το σενάριο των εντολών του δεύτερου αντικείμενου, εμφανίζεται στην επόμενη εικόνα.



Εικόνα 5.39: Απάντηση 2^ο αντικειμένου, η εντολή ‘εάν..’ (παράδειγμα 3^ο)

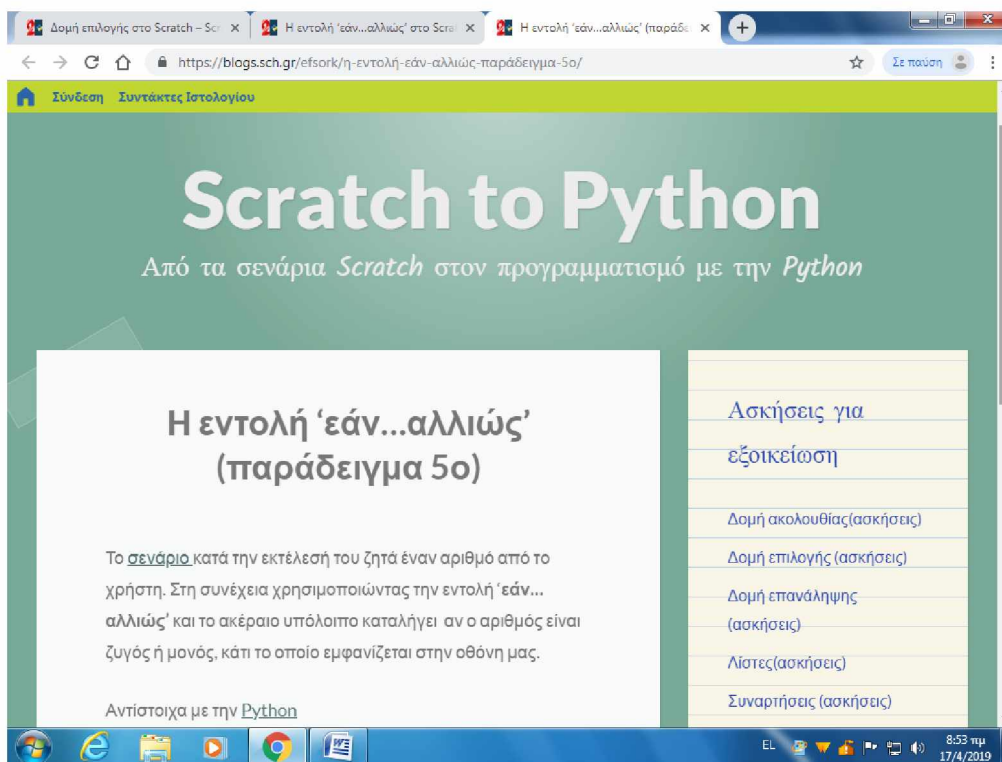
Έπειτα ακολουθεί η εντολή ‘εάν...αλλιώς’, του μενού, ‘Δομή επιλογής’. Στη επόμενη εικόνα παρουσιάζεται η σελίδα των παραδειγμάτων για τη συγκεκριμένη εντολή.



Εικόνα 5.40: Η εντολή ‘εάν...αλλιώς’, στο Scratch

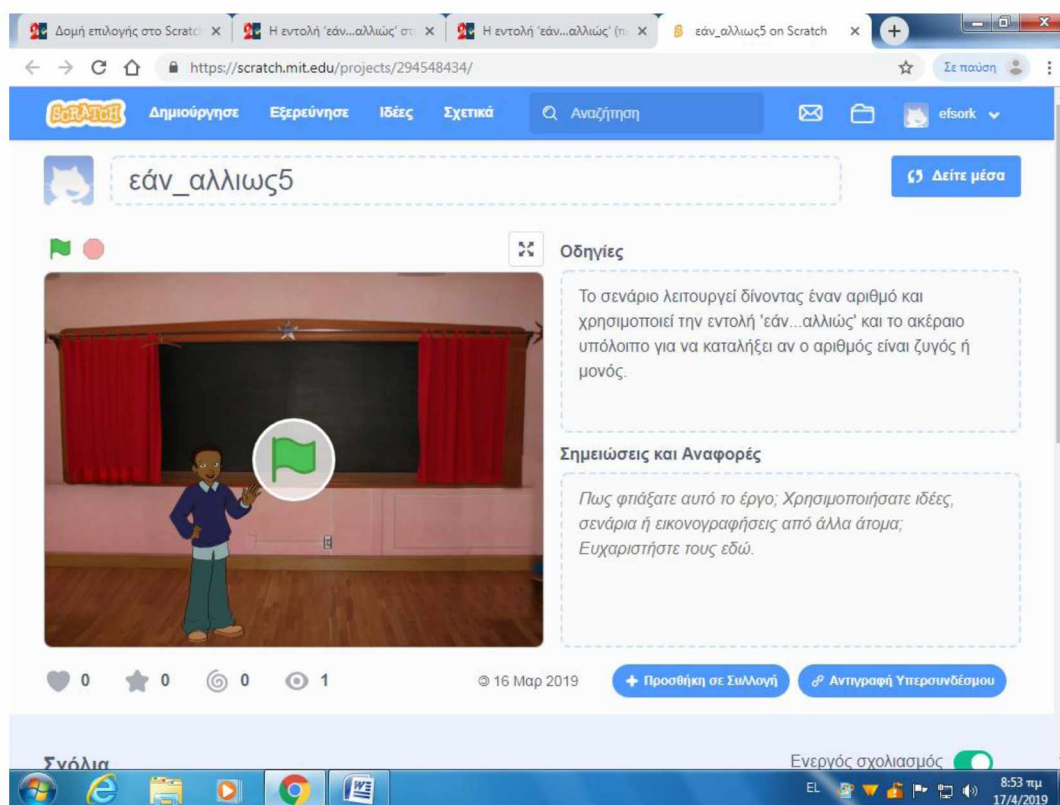
Αυτό που επιλέγεται να παρουσιαστεί είναι το πέμπτο παράδειγμα, όπως στην εικόνα που ακολουθεί. Κατά την εκτέλεση του συγκεκριμένου παραδείγματος ζητείται να δοθεί ένας αριθμός από το χρήστη.

Στη συνέχεια το σενάριο ελέγχει αν ο αριθμός που δίνεται είναι μονός ή ζυγός και το αποτέλεσμα εμφανίζεται στην οθόνη.



Εικόνα 5.41: Η εντολή 'εάν...αλλιώς' (παράδειγμα 5^ο)

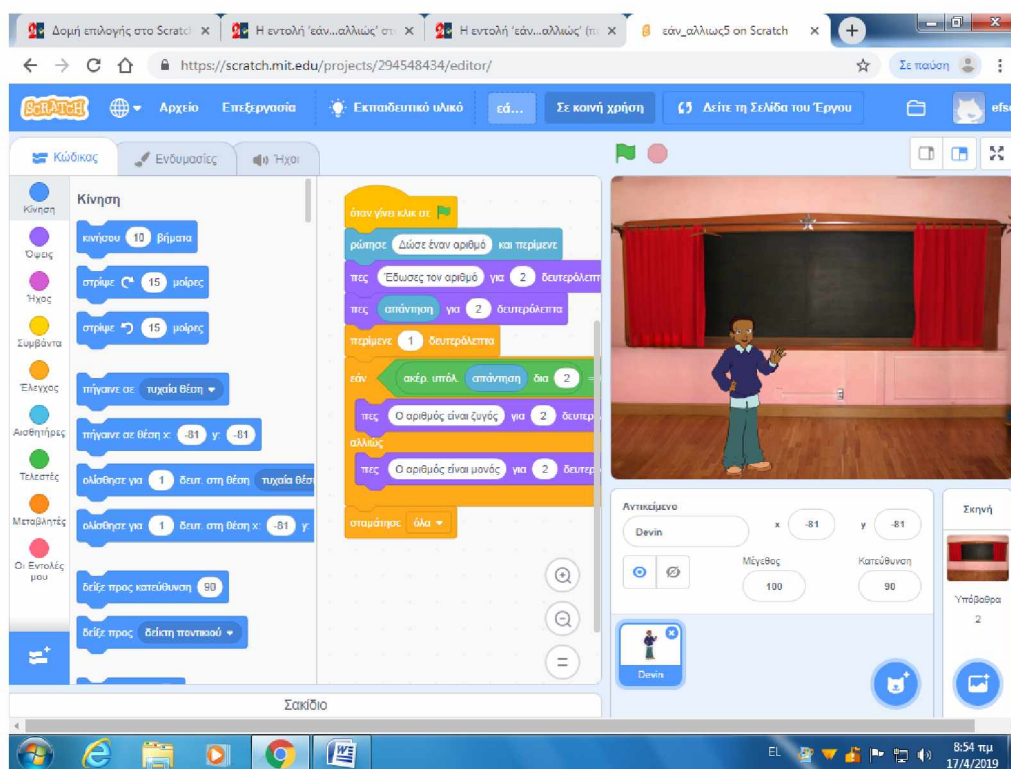
Ακολουθώντας τον υπερσύνδεσμο της σελίδας για το σενάριο σε Scratch, προκύπτει η επόμενη εικόνα, στην οποία όπως παρατηρείται υπάρχει ένα αντικείμενο το οποίο ζητά τον αριθμό από το χρήστη. Επιπλέον στη σελίδα που είναι αποθηκευμένο το σενάριο, υπάρχει περιγραφή για τον τρόπο εκτέλεσης του σεναρίου και το αποτέλεσμα που θα προκύψει.



Εικόνα 5.42: Περιβάλλον Scratch, παράδειγμα 'εάν...αλλιώς', (παράδειγμα 5^ο)

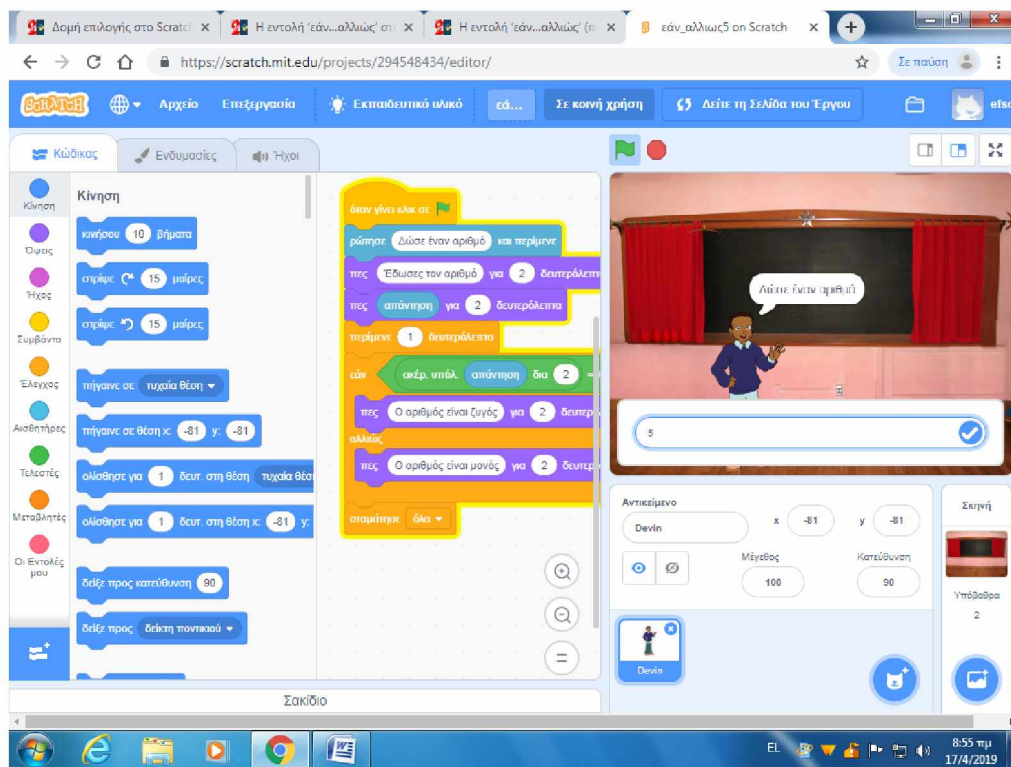
Στη συνέχεια όταν εισερχόμαστε στο σενάριο των εντολών του παραδείγματος, το αντικείμενο ζητά τον αριθμό από το χρήστη. Όταν το σενάριο λάβει τον αριθμό, τον εμφανίζει για δύο δευτερόλεπτα στην οθόνη. Έπειτα, ελέγχεται με την εντολή 'εάν...αλλιώς' και χρησιμοποιώντας το ακέραιο υπόλοιπο με το δύο, η περίπτωση που αριθμός είναι ζυγός. Εάν αυτό ισχύει τότε εμφανίζεται στην οθόνη ότι ο αριθμός είναι ζυγός. Σε αντίθετη περίπτωση εμφανίζεται στην οθόνη μας ότι ο αριθμός είναι μονός.

Στην επόμενη εικόνα, παρουσιάζονται οι εντολές του σεναρίου για το αντικείμενο.



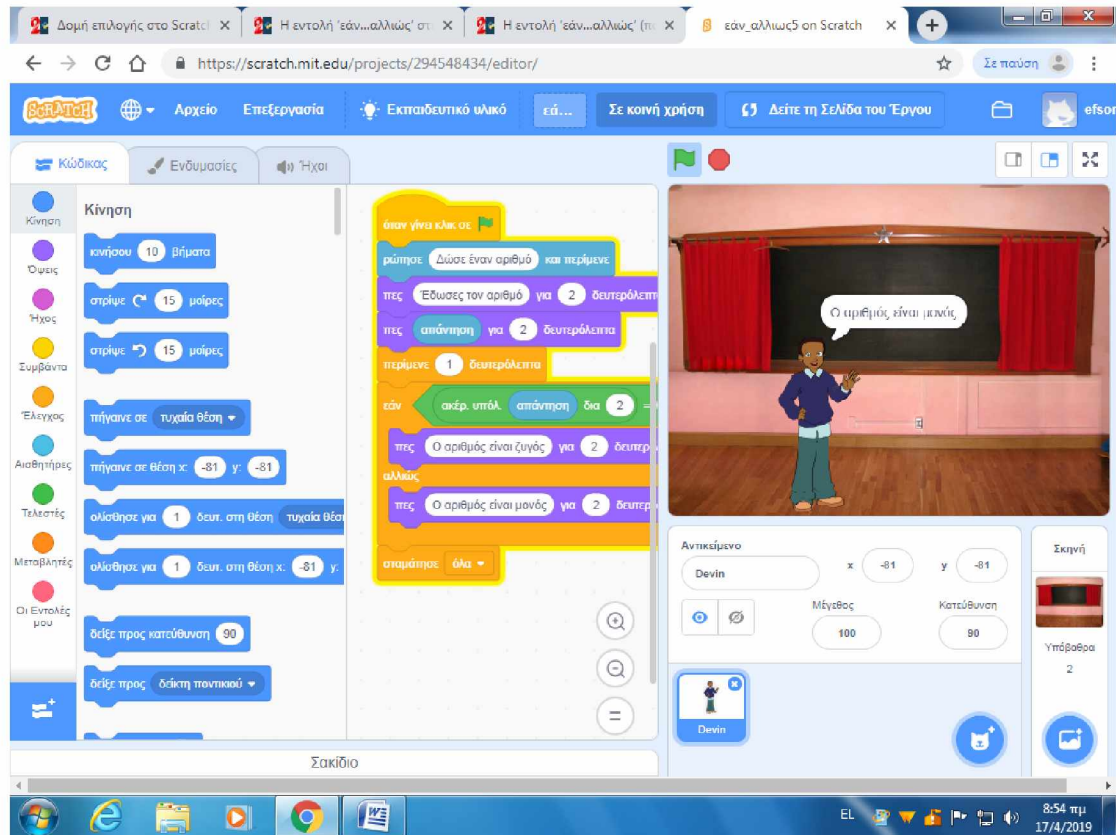
Εικόνα 5.43: Σενάριο εντολών, η εντολή ‘εάν...αλλιώς’, (παράδειγμα 5^ο)

Όταν εκτελείται το σενάριο προκύπτει η επόμενη εικόνα



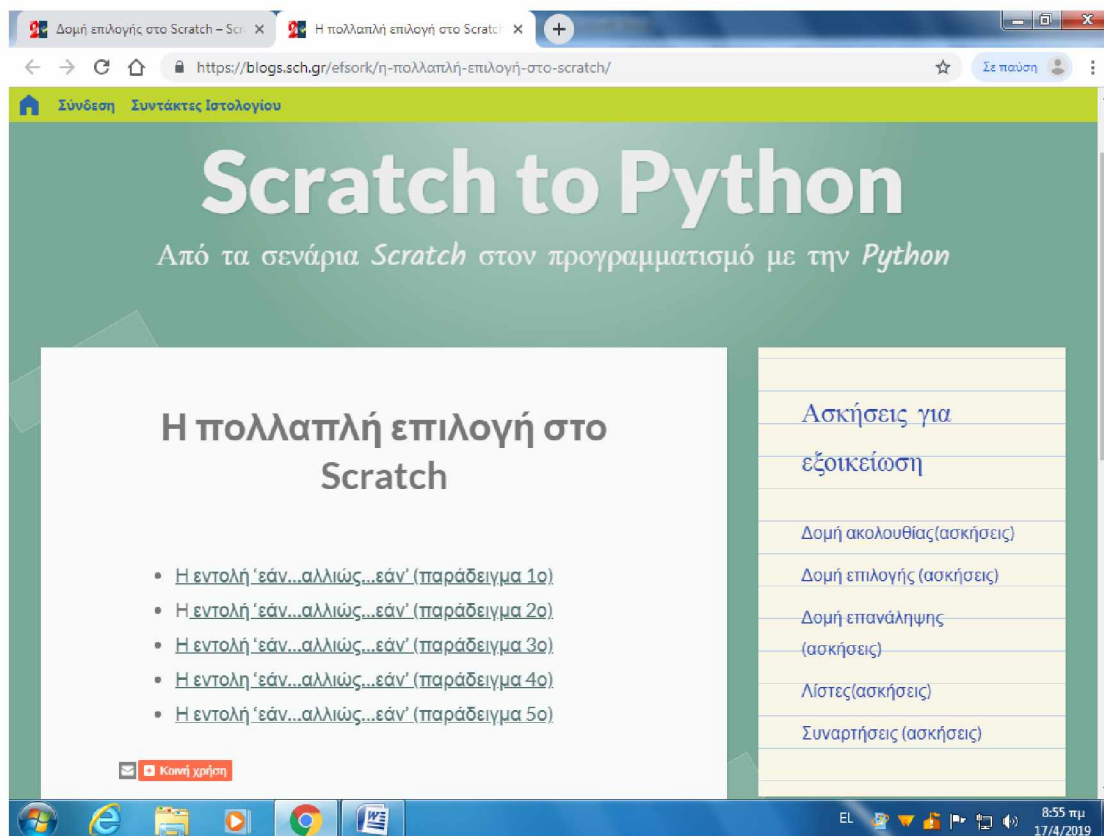
Εικόνα 5.44: Ερώτηση αντικειμένου για αριθμό, εντολή ‘εάν...αλλιώς’, (παράδειγμα 5^ο)

Σύμφωνα με την περιγραφή που προηγήθηκε, όταν δοθεί μονός αριθμός η εικόνα που εμφανίζεται στην οθόνη είναι:



Εικόνα 5.45: Μονός αριθμός, η εντολή 'εάν...αλλιώς', (παράδειγμα 5^ο)

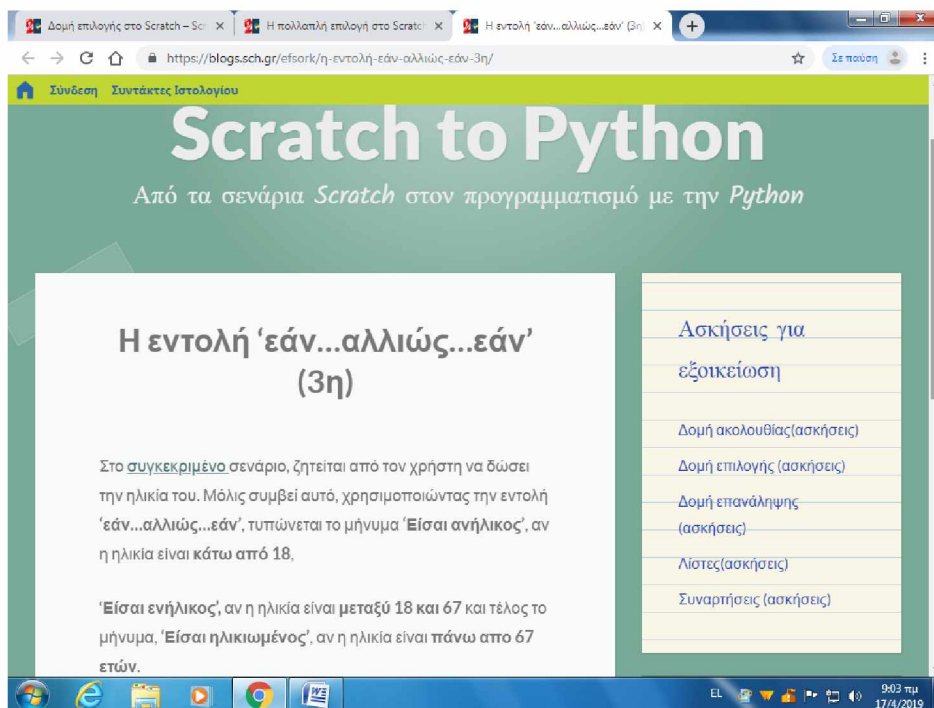
Έπειτα, το στοιχείο που παρουσιάζεται είναι η πολλαπλή επιλογή. Η σελίδα η οποία περιέχει πέντε διαφορετικά παραδείγματα, απεικονίζεται στη συνέχεια.



Εικόνα 5.46: Η πολλαπλή επιλογή στο Scratch

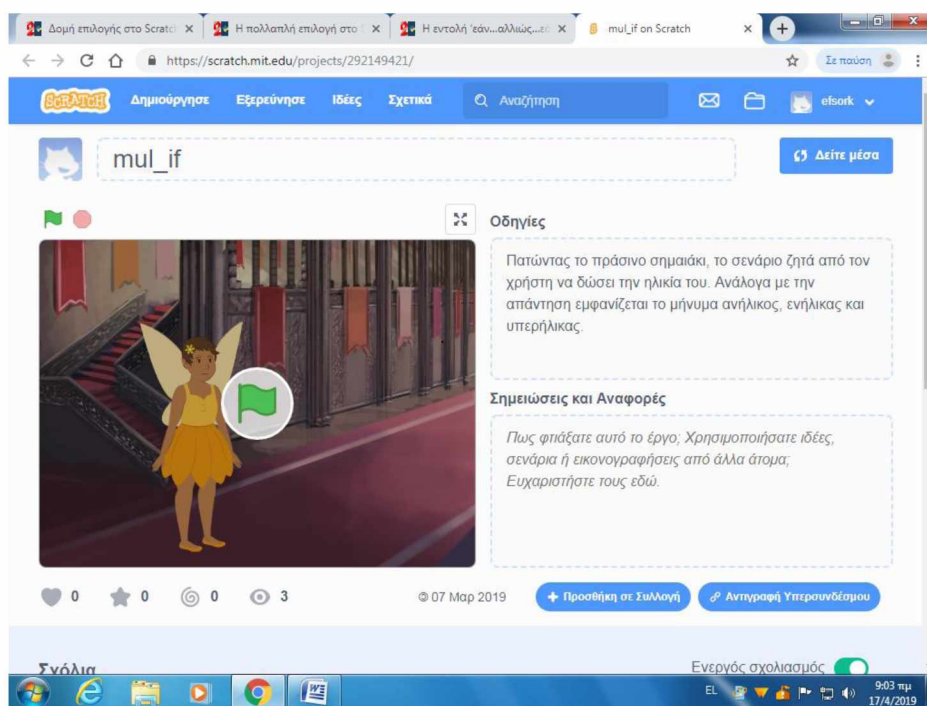
Το παράδειγμα που θα παρουσιαστεί είναι το τρίτο. Στο συγκεκριμένο παράδειγμα, το σενάριο μέσω του αντικειμένου, ζητά από το χρήστη να δώσει την ηλικία του. Αν ο αριθμός που δοθεί είναι μικρότερος από 18, τυπώνεται στην οθόνη: 'Είσαι ανήλικος'. Αν ο αριθμός είναι μεγαλύτερος από το 67, τυπώνεται στην οθόνη το μήνυμα: 'Είσαι ηλικιωμένος' ενώ σε κάθε άλλη περίπτωση, τυπώνεται το μήνυμα 'Είσαι ενήλικας'. Τα προαναφερόμενα, επιτυγχάνονται, χρησιμοποιώντας την εντολή 'εάν...αλλιώς...εάν'.

Στην επόμενη εικόνα εμφανίζεται η οθόνη του παραδείγματος.



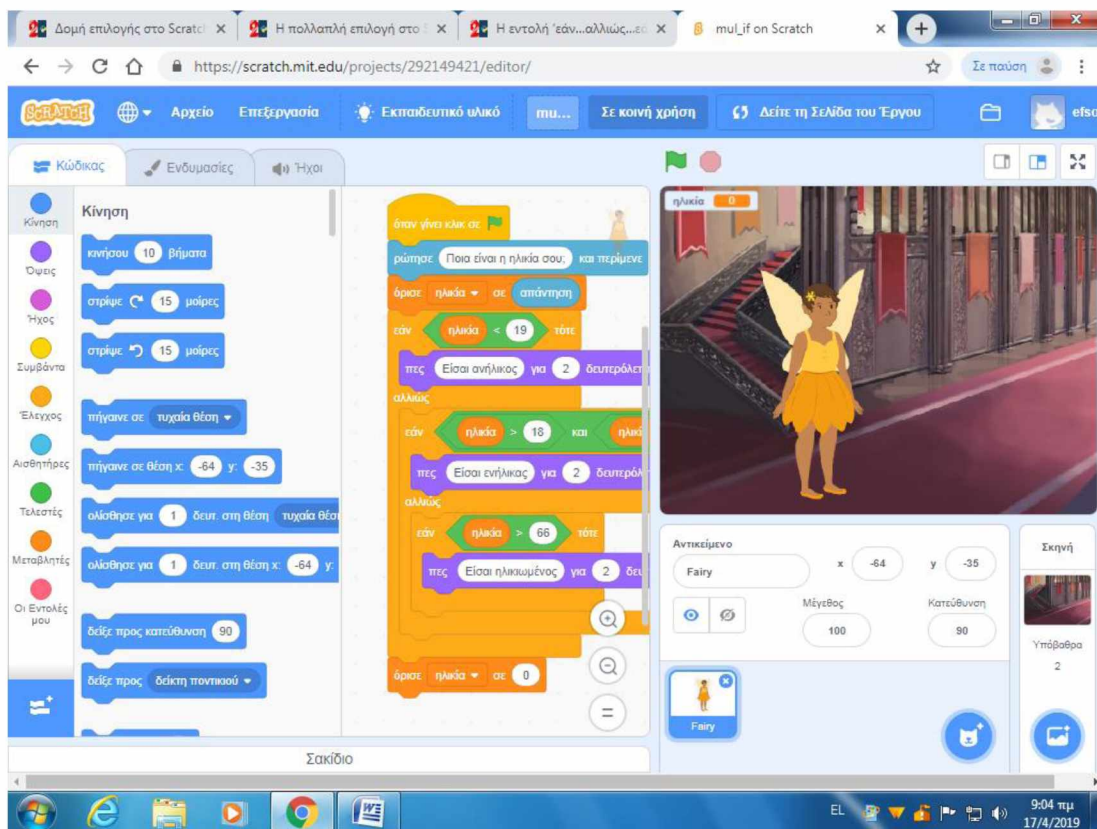
Εικόνα 5.47: Η εντολή ‘εάν...αλλιώς...εάν’, (παράδειγμα 3^ο)

Επιλέγοντας τον υπερσύνδεσμο, οδηγούμαστε στην οθόνη του περιβάλλοντος Scratch, που είναι αποθηκευμένο το παράδειγμα, όπως εμφανίζεται στην επόμενη εικόνα.



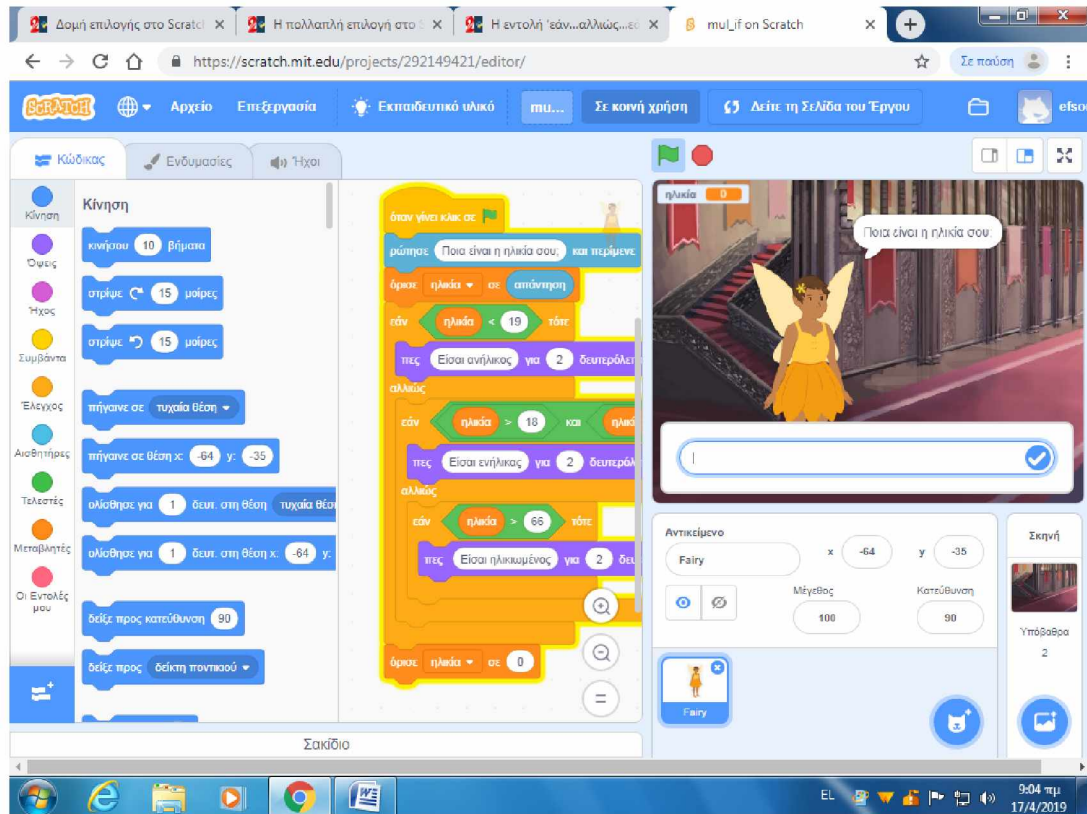
Εικόνα 5.48: Περιβάλλον Scratch, εντολή ‘εάν...αλλιώς...αλλιώς’, (παράδειγμα 3^ο)

Όταν ο χρήστης εισέρχεται στις εντολές του σεναρίου η σελίδα που προκύπτει, απεικονίζεται στη συνέχεια.



Εικόνα 5.49: Εντολές σεναρίου, η εντολή ‘εάν... αλλιώς... εάν’, (παράδειγμα 3^ο)

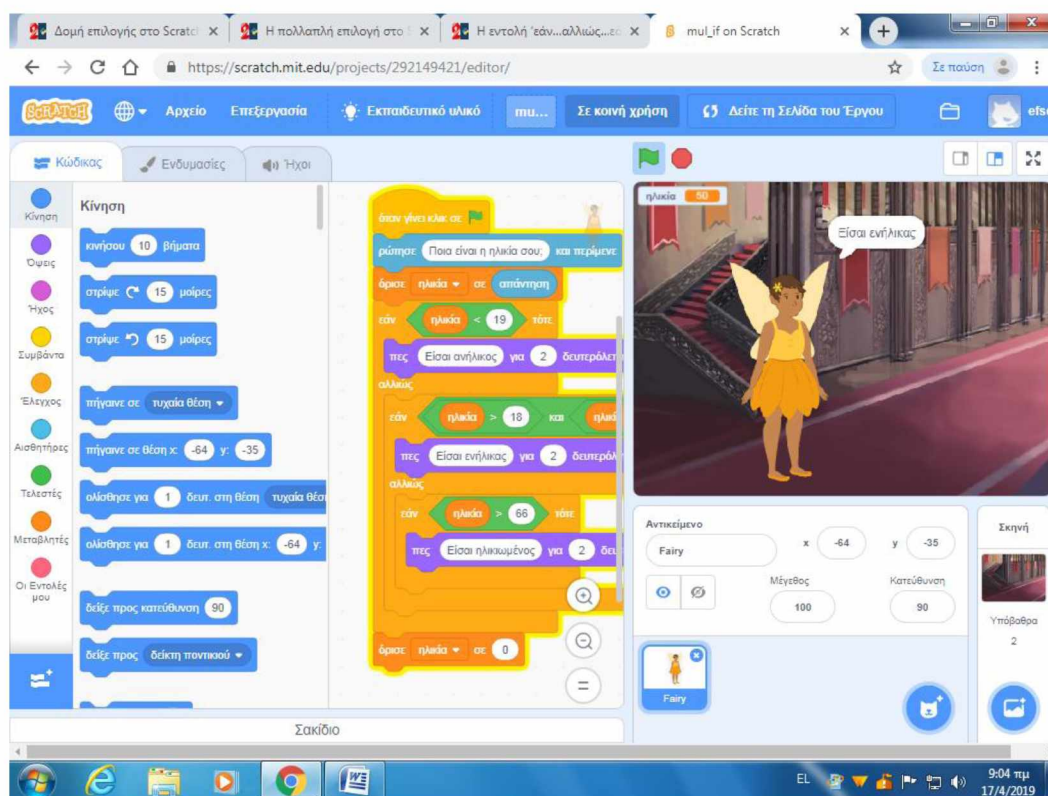
Στη συνέχεια εκτελώντας το σενάριο και πατώντας το πράσινο σημαϊάκι που υπάρχει στο περιβάλλον του Scratch, το αντικείμενο ρωτά την ηλικία του χρήστη, όπως στην επόμενη εικόνα.



Εικόνα 5.50: Ερώτηση αντικειμένου, εντολή ‘εάν...αλλιώς...εάν’, (παράδειγμα 3^ο)

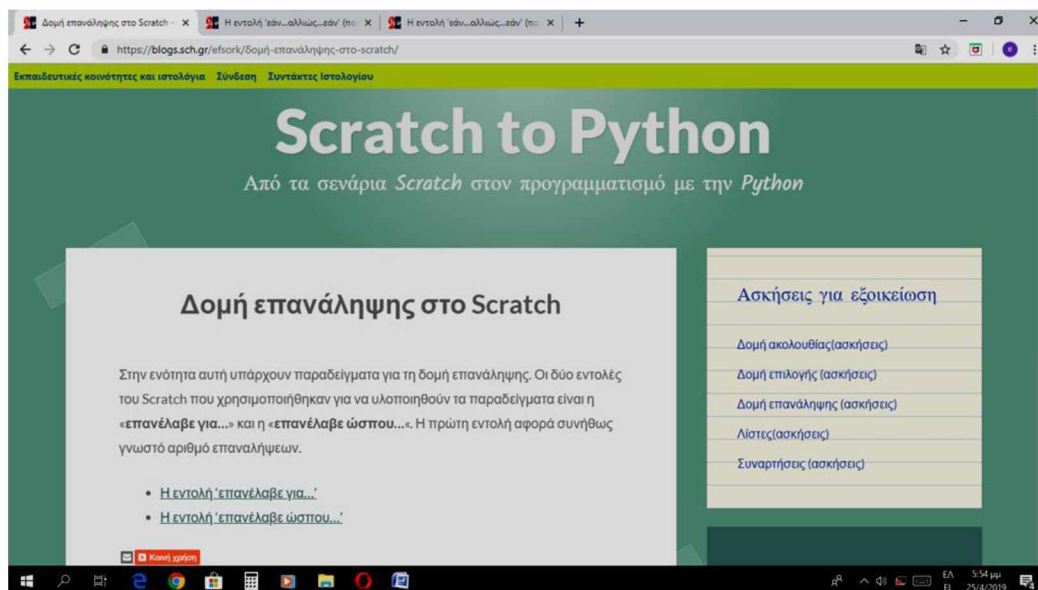
Όταν ο χρήστης πληκτρολογήσει έναν αριθμό, ο οποίος αποθηκεύεται στην μεταβλητή ‘ηλικία’ και είναι ορισμένη στο σενάριο, κάνοντας τους κατάλληλους ελέγχους, χρησιμοποιώντας την ‘εάν...αλλιώς...εάν’, τυπώνει στην οθόνη το αντίστοιχο μήνυμα.

Στην επόμενη εικόνα, παρουσιάζεται η τιμή της μεταβλητής και το μήνυμα που τυπώνεται στην οθόνη.



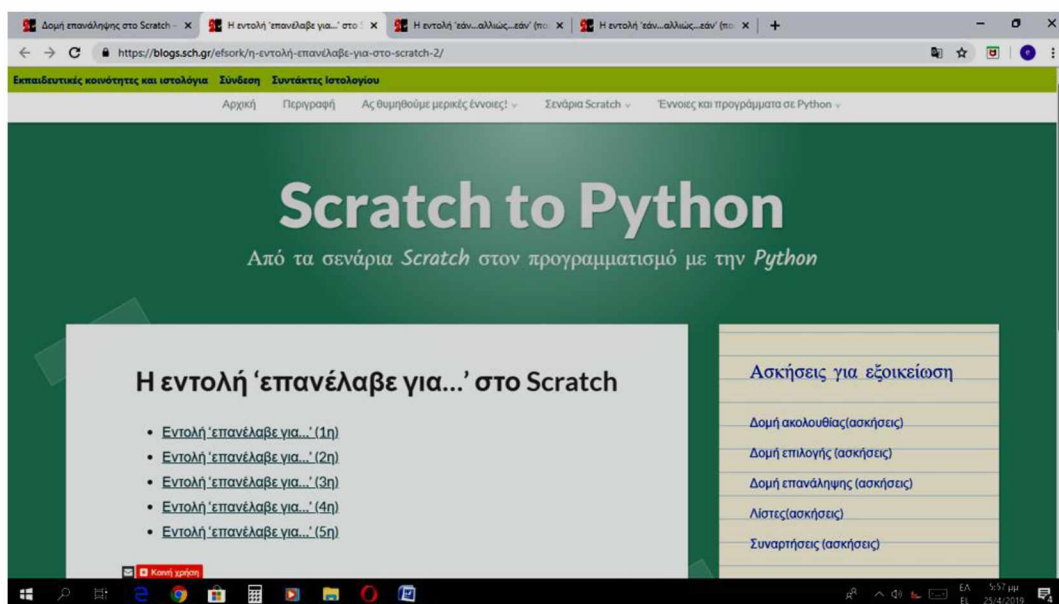
Εικόνα 5.51: Τελικό αποτέλεσμα, εντολή ‘εάν...αλλιώς...εάν’, (παράδειγμα 3^ο)

Το επόμενο στοιχείο του μενού ‘Σενάρια σε Scratch’, είναι η ‘*Δομή επανάληψης*’. Στη σελίδα αυτή, παρουσιάζονται παραδείγματα σε Scratch, χρησιμοποιώντας τις εντολές ‘επανάλαβε για...’ και ‘επανάλαβε ώσπου...’. Η εν λόγω σελίδα απεικονίζεται στη συνέχεια.



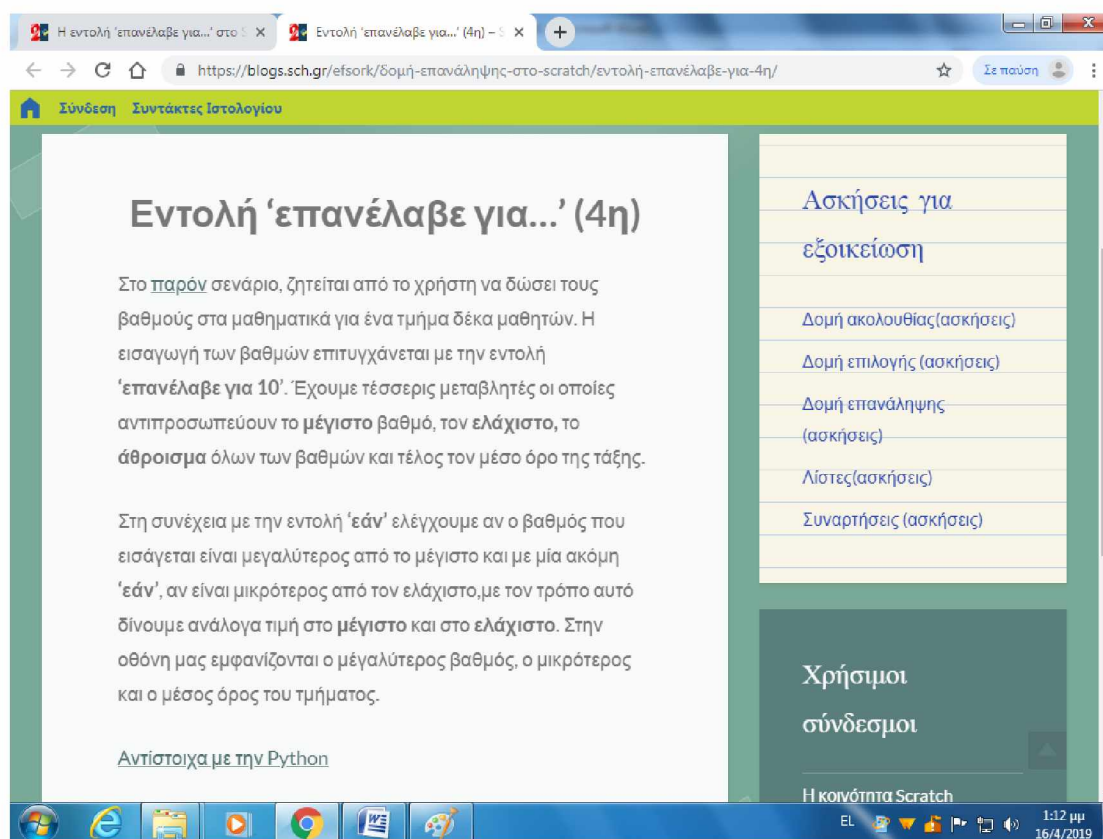
Εικόνα 5.52: Δομή επανάληψης στο Scratch

Σε περίπτωση που ο χρήστης επιλέξει τον υπερσύνδεσμο, ‘Η εντολή επανέλαβε για...’, εμφανίζεται η επόμενη οθόνη, η οποία περιέχει πέντε παραδείγματα της εντολής.



Εικόνα 5.53: Η εντολή ‘επανάλαβε για...’ στο Scratch

Στην παρούσα αναφορά θα αναλυθεί και θα παρουσιαστεί το τέταρτο παράδειγμα της εντολής ‘επανάλαβε για...’. Στο συγκεκριμένο παράδειγμα, το σενάριο ζητά από το χρήστη να εισάγει τις βαθμολογίες στο μάθημα των μαθηματικών, για ένα τμήμα δέκα ατόμων. Όταν δοθούν οι δέκα βαθμοί, το σενάριο εμφανίζει τον μεγαλύτερο, τον μικρότερο και το μέσο όρο του τμήματος. Η σελίδα του παραδείγματος εμφανίζεται στη συνέχεια.



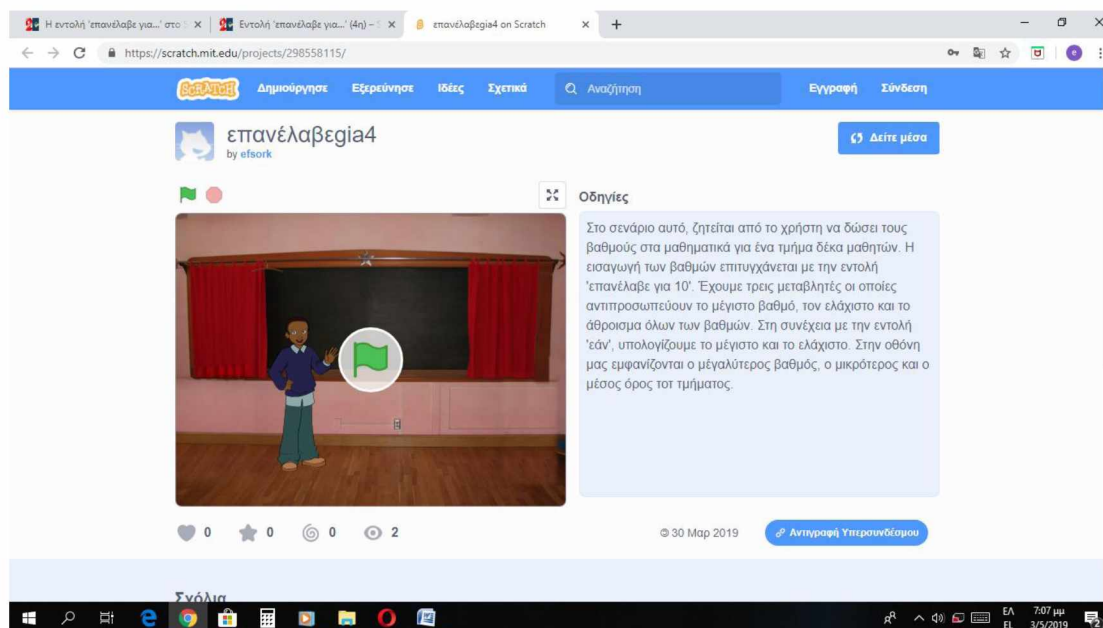
Εικόνα 5.54: Εντολή 'επανάλαβε για...' (4^η)

Για να επιτευχθεί η διαδοχική εισαγωγή των δέκα αριθμών, χρησιμοποιείται η εντολή 'επανάλαβε για 10'. Αυτό σημαίνει ότι το κομμάτι των εντολών που περιέχονται στην 'επανάλαβε για...', θα εκτελεστεί δέκα φορές, όσο είναι και οι αριθμοί. Στο σενάριο έχουν οριστεί τέσσερις μεταβλητές, μία για το μέγιστο βαθμό, μία για τον ελάχιστο, μία για το άθροισμα και τέλος μία για τον μέσο όρο.

Για την εύρεση του μεγαλύτερου βαθμού, χρησιμοποιείται η εντολή 'εάν' η οποία ελέγχει, αν ο κάθε αριθμός που εισάγεται είναι μεγαλύτερος από τον μέγιστο (μέχρι εκείνη τη στιγμή) και με μία ακόμη 'εάν', ελέγχεται αν ο νέος αριθμός είναι μικρότερος από τον ελάχιστο (μέχρι εκείνη τη στιγμή). Με τον τρόπο αυτό, γίνεται εφικτό να δοθεί τιμή στις μεταβλητές για τον μέγιστο και τον ελάχιστο.

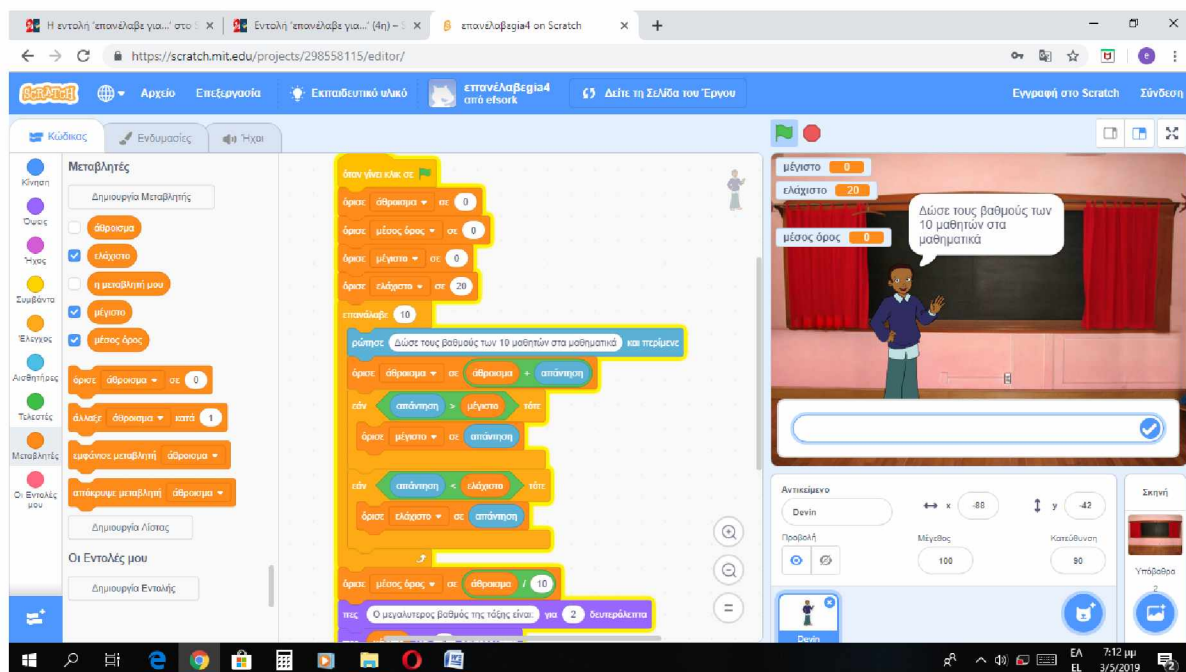
Στο τέλος της εκτέλεσης του σεναρίου, εμφανίζεται στην οθόνη ο μέγιστος βαθμός, ο μικρότερος βαθμός και ο μέσος όρος του τμήματος στο μάθημα των μαθηματικών.

Το περιβάλλον του Scratch που είναι αποθηκευμένο το σενάριο απεικονίζεται στη συνέχεια, στο οποίο περιέχεται μια σύντομη περιγραφή της λειτουργίας του σεναρίου.



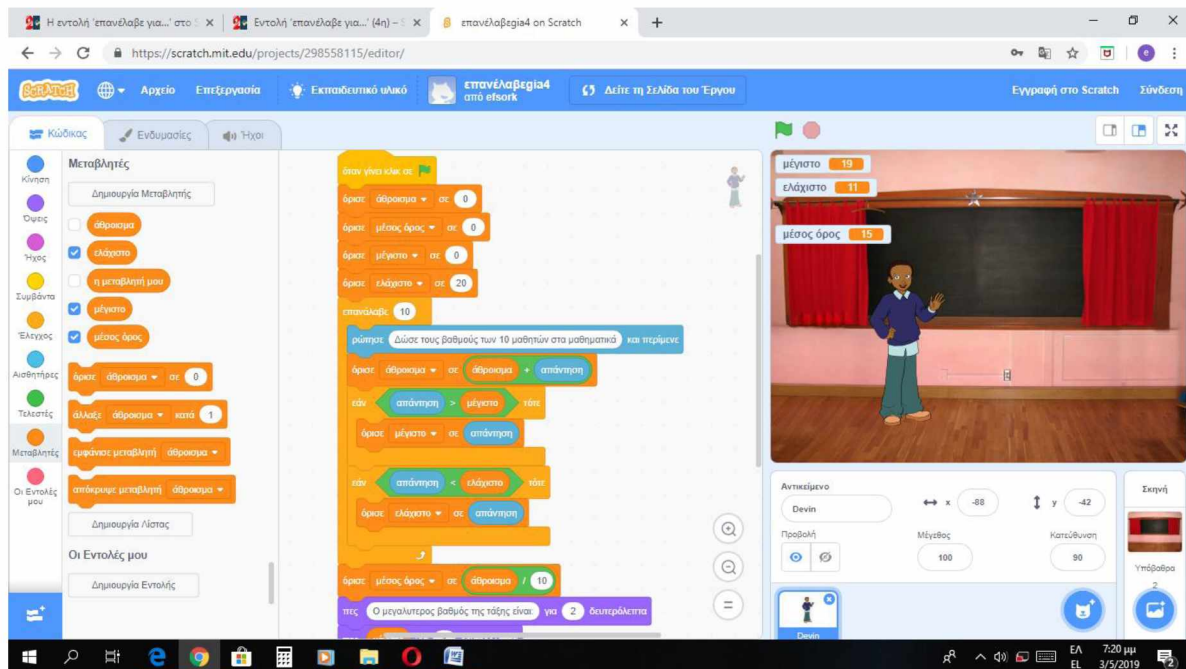
Εικόνα 5.55: Περιβάλλον Scratch, εντολή ‘επανέλαβε για...’ (4^η)

Έπειτα όταν εισερχόμαστε στις εντολές του σεναρίου και στο περιβάλλον δημιουργίας του σεναρίου, μπορούμε να δούμε αναλυτικά τον τρόπο με τον οποίο χρησιμοποιηθήκαν οι εντολές που παρέχει το Scratch, όπως εμφανίζεται στην επόμενη εικόνα. Πατώντας το πράσινο σημαϊάκι για την έναρξη της εκτέλεσης του σεναρίου, το αντικείμενο ζητά από το χρήστη να εισάγει τους δέκα αριθμούς.



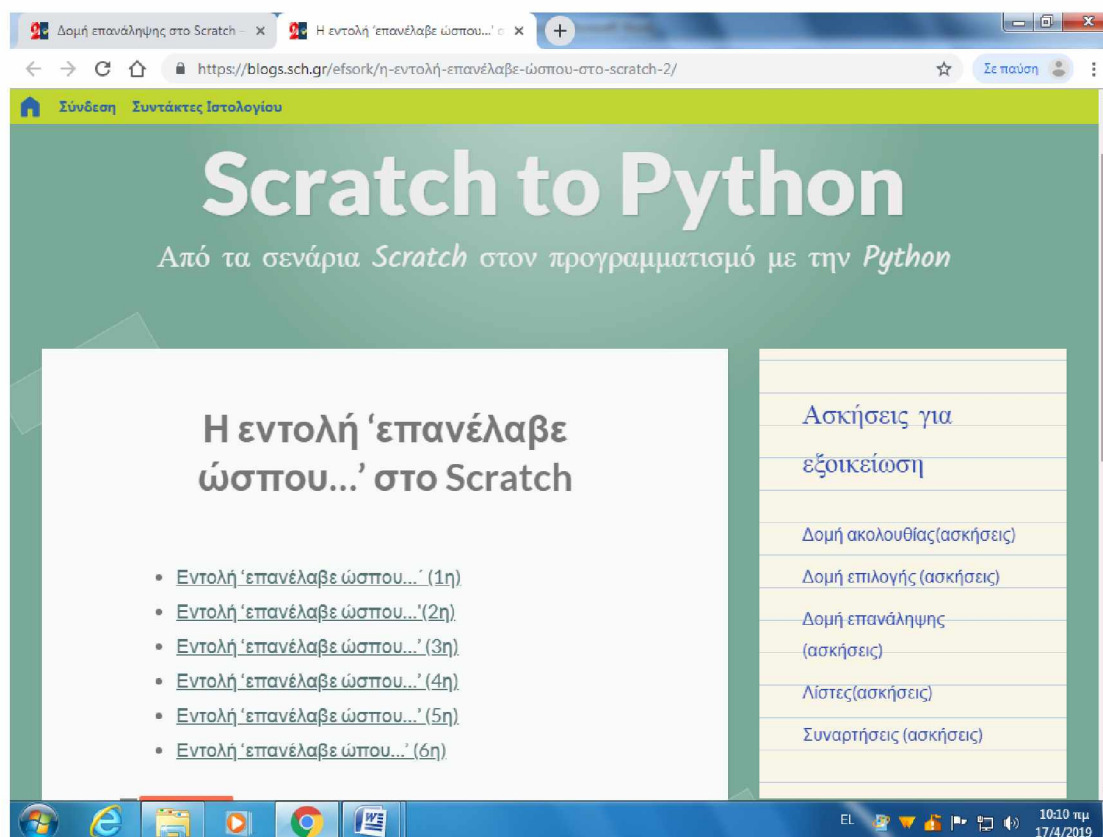
Εικόνα 5.56: Η εντολή ‘επανάλαβε για...’ στο Scratch (4^η)

Όταν δοθούν οι αριθμοί από το χρήστη ακολουθώντας τη λειτουργία που περιγράφηκε στα προηγούμενα, το σενάριο εμφανίζει τον μέγιστο, τον ελάχιστο και το μέσο όρο του τμήματος όπως απεικονίζεται στη συνέχεια.



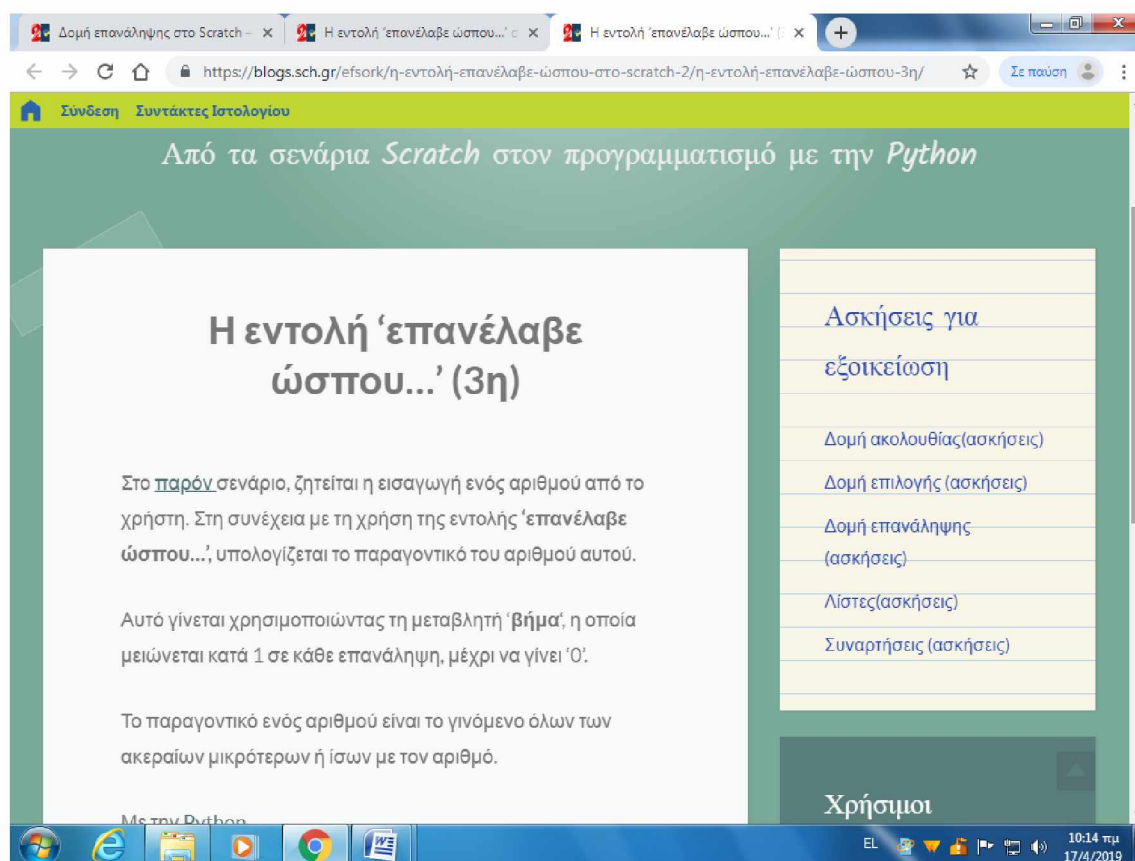
Εικόνα 5.57: Η εντολή ‘επανάλαβε για...’ (4^η)

Το επόμενο στοιχείο του μενού ‘Σενάρια σε Scratch’, είναι τα παραδείγματα που αφορούν την εντολή ‘επανάλαβε ώσπου...’. Η σελίδα με την εν λόγω εντολή περιέχει έξι παραδείγματα, και εμφανίζεται στην εικόνα που ακολουθεί.



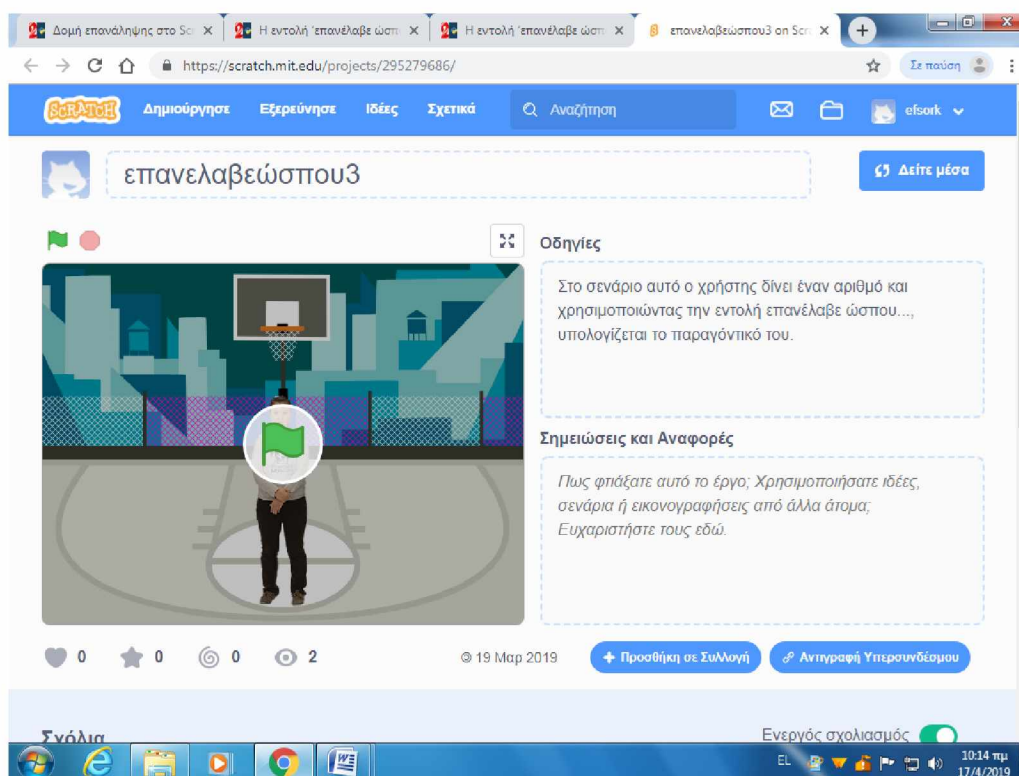
Εικόνα 5.58: Η εντολή ‘επανάλαβε ώσπου...’ στο Scratch

Αρχικά, το παράδειγμα που θα αναλυθεί είναι το τρίτο της συγκεκριμένης εντολής. Στο τρίτο παράδειγμα το σενάριο κατά την εκτέλεσή του, ζητά ως είσοδο έναν αριθμό από το χρήστη. Στη συνέχεια υπολογίζει και εμφανίζει το παραγοντικό του αριθμού αυτού. Το παραγοντικό ενός αριθμού είναι το γινόμενο όλων των αριθμών μικρότερων ή ίσων με τον αριθμό. Η σελίδα του παραδείγματος που εμφανίζεται στο ιστολόγιο ακολουθεί στην επόμενη εικόνα.



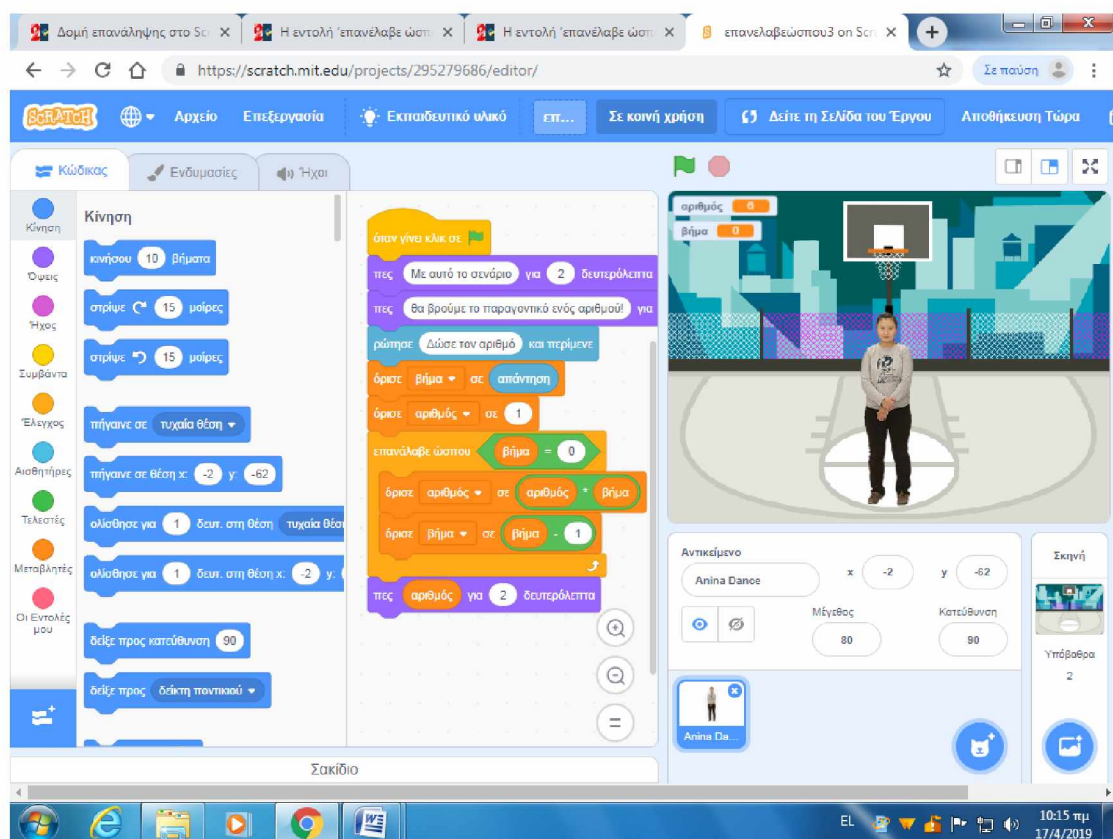
Εικόνα 5.59: Η εντολή 'επανάλαβε ώσπου...' (3^η)

Στο σενάριο χρησιμοποιείται η εντολή 'επανάλαβε ώσπου...' και η μεταβλητή 'βήμα', η οποία μειώνεται κατά ένα σε κάθε επανάληψη της εντολής, μέχρι να γίνει μηδέν. Στην επόμενη εικόνα παρουσιάζεται το περιβάλλον Scratch του τρίτου παραδείγματος.



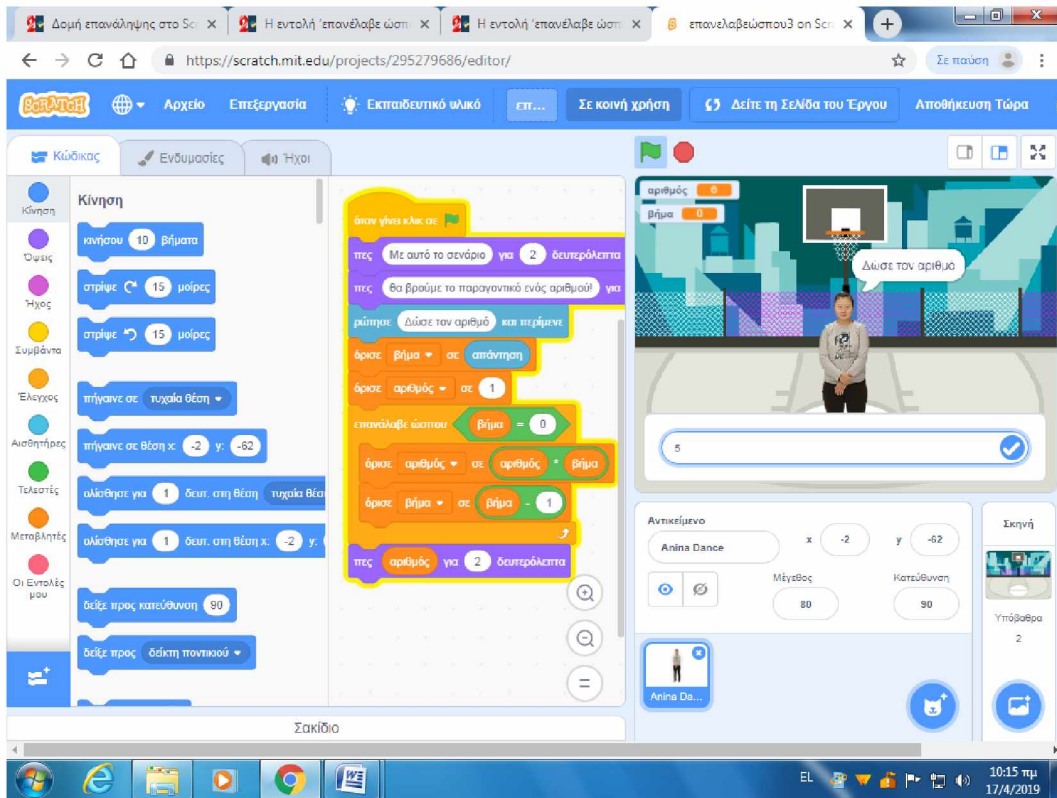
Εικόνα 5.60: Περιβάλλον Scratch, η εντολή ‘επανελάβε ώσπου...’ (3^η)

Όταν ο χρήστης εισέρχεται στο περιβάλλον εντολών του Scratch, έχει τη δυνατότητα να μελετήσει τις εντολές και τις μεταβλητές που χρησιμοποιήθηκαν για τον υπολογισμό του παραγοντικού του αριθμού που εισάγεται από το χρήστη, όπως απεικονίζεται στη συνέχεια.

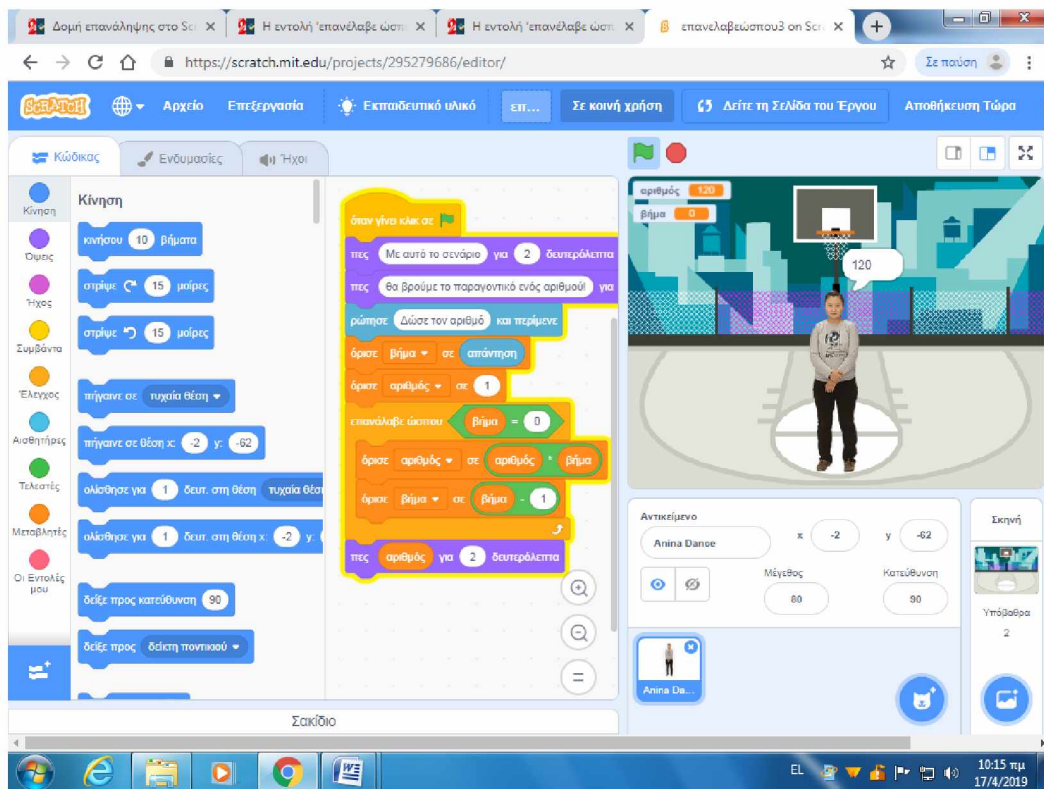


Εικόνα 5.61: Εντολές στο Scratch, η εντολή ‘επανέλαβε ώσπου...’ (3^η)

Όπως φαίνεται στην προηγούμενη εικόνα, η εντολή ‘επανέλαβε ώσπου...’, εκτελείται ώσπου η μεταβλητή ‘**βήμα**’ που έχει οριστεί στο σενάριο, γίνει μηδέν. Όταν γίνει μηδέν η ‘επανέλαβε ώσπου...’ δεν εκτελείται. Μέσα στο blog της επανάληψης, η μεταβλητή ‘**αριθμός**’, που δίνεται από το χρήστη πολλαπλασιάζεται με τη μεταβλητή ‘**βήμα**’. Στη συνέχεια η μεταβλητή ‘**βήμα**’ μειώνεται κατά ένα. Τελικά το αποτέλεσμα που θα εμφανιστεί στην οθόνη, είναι το γινόμενο των δύο μεταβλητών. Στην εικόνα που ακολουθεί εμφανίζεται η εκτέλεση του σεναρίου.



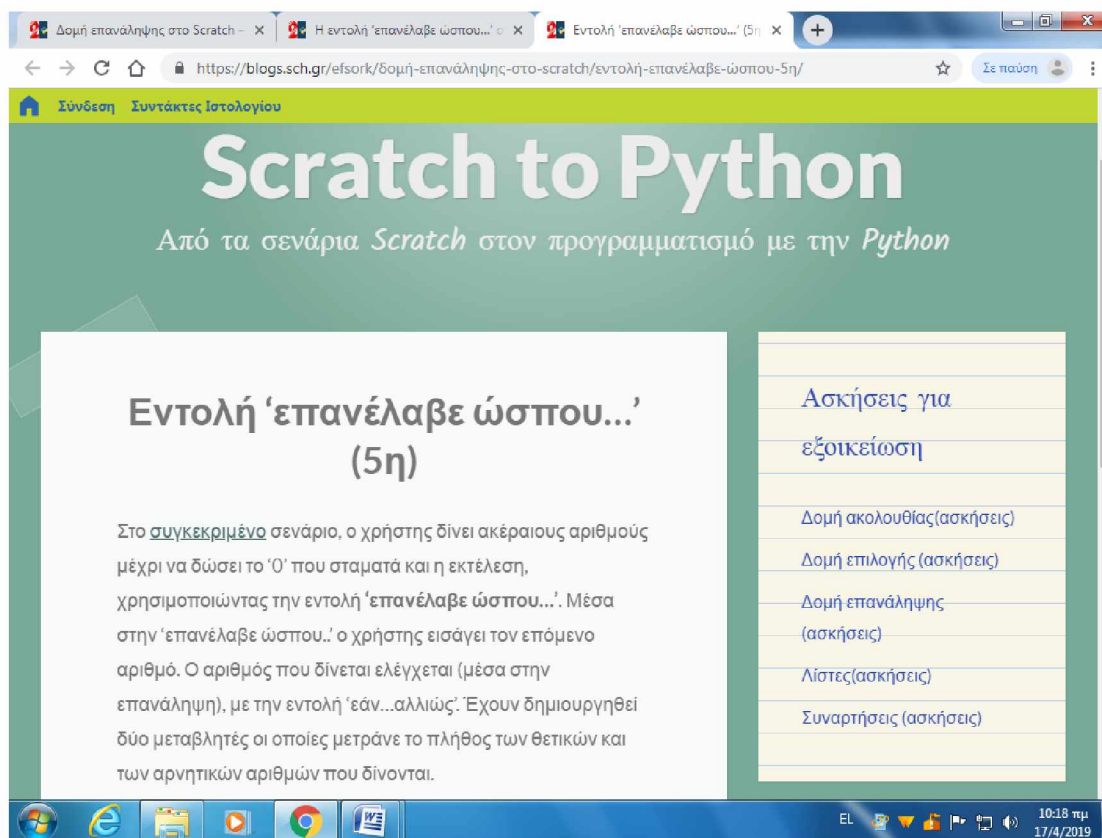
Εικόνα 5.62: Η εντολή ‘επανάλαβε ώσπου...’ (3^η), εισαγωγή αριθμού



Εικόνα 5.63: Η εντολή ‘επανάλαβε ώσπου...’, παραγοντικό αριθμού (3^η)

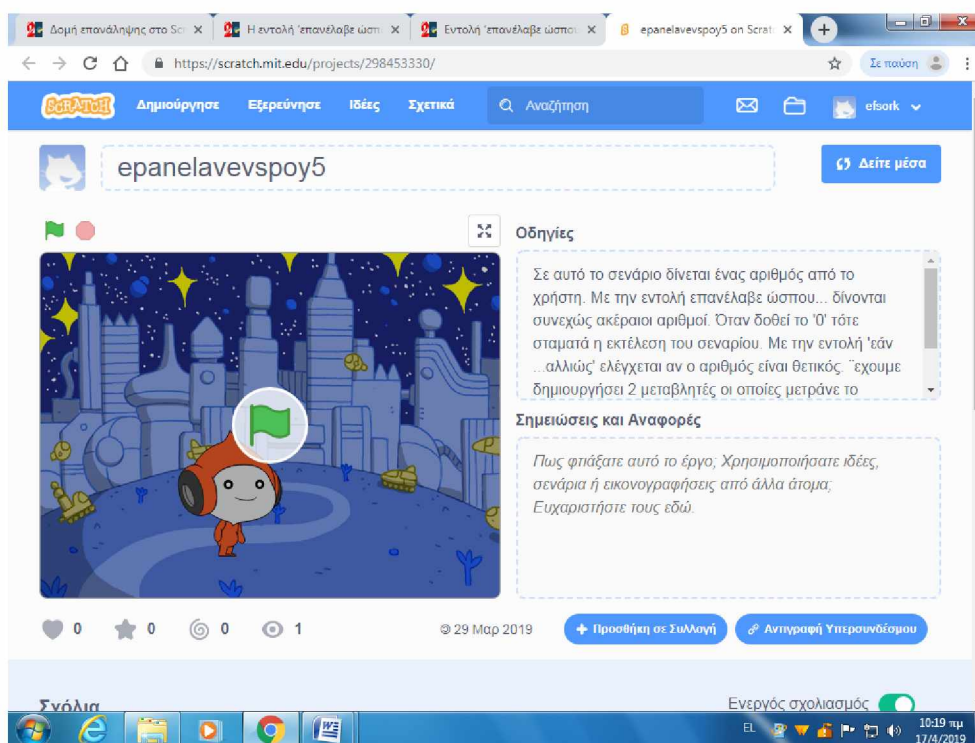
Στη συνέχεια επιλέχθηκε ένα επιπλέον παράδειγμα που αφορά στη συγκεκριμένη εντολή, αυτό είναι το πέμπτο. Στο συγκεκριμένο παράδειγμα, κατά την εκτέλεση του σεναρίου, ζητείται από το χρήστη να εισάγει ακέραιους αριθμούς. Όταν ο χρήστης δώσει ως είσοδο τον αριθμό 'μηδέν', τερματίζεται η εκτέλεση του σεναρίου. Η διαδοχική εισαγωγή αριθμών από το χρήστη επιτυγχάνεται με την εντολή 'επανάλαβε ώσπου...', ζητώντας αριθμούς μέχρι να δοθεί το μηδέν. Το σενάριο μετρά τους θετικούς και τους αρνητικούς αριθμούς που δίνονται. Για να επιτευχθεί αυτό, χρησιμοποιήθηκαν δύο μεταβλητές ως μετρητές, μία για τους θετικούς και μία για τους αρνητικούς αριθμούς.

Όταν ο χρήστης δώσει το μηδέν, το σενάριο σταματά την εκτέλεσή του και τυπώνει το πλήθος των θετικών και των αρνητικών αριθμών που δόθηκαν. Στην εικόνα που ακολουθεί παρουσιάζεται η σελίδα του πέμπτου παραδείγματος.



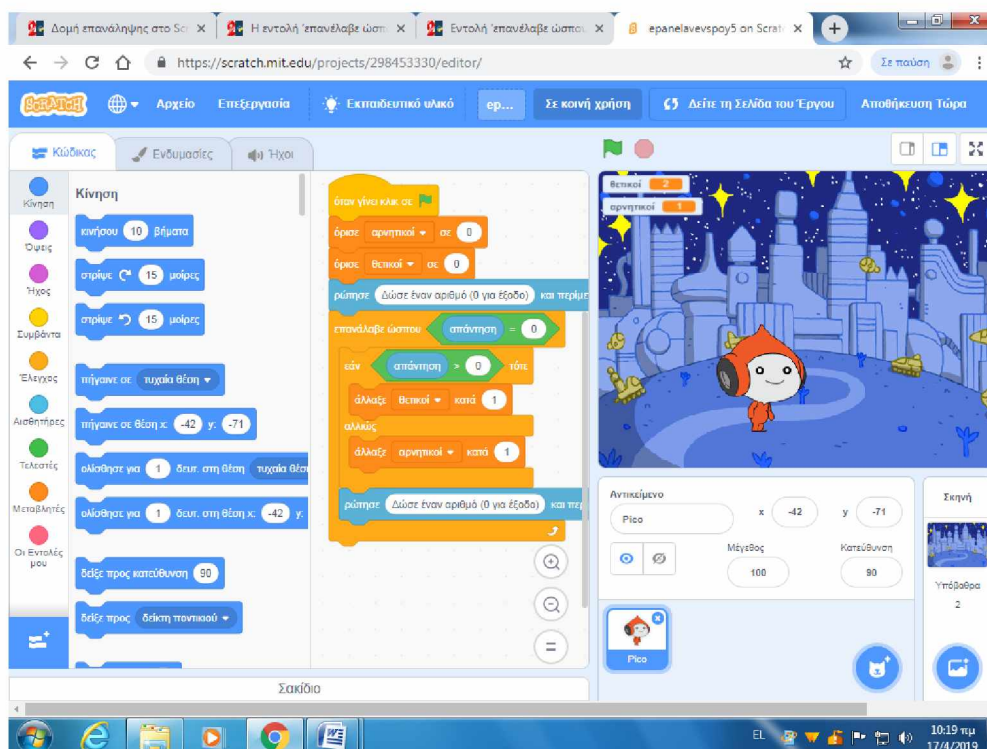
Εικόνα 5.64: Εντολή 'επανάλαβε ώσπου...' (5^η)

Ακολουθώντας τον υπερσύνδεσμο της σελίδας που αφορά το περιβάλλον του παραδείγματος μεταφερόμαστε στην επόμενη οθόνη.



Εικόνα 5.65: Περιβάλλον Scratch, η εντολή ‘επανάλαβε ώσπου...’ (5^η)

Σε περίπτωση που ο χρήστης επιλέξει να εισέλθει στο σενάριο των εντολών του Scratch μεταφέρεται στην επόμενη εικόνα.

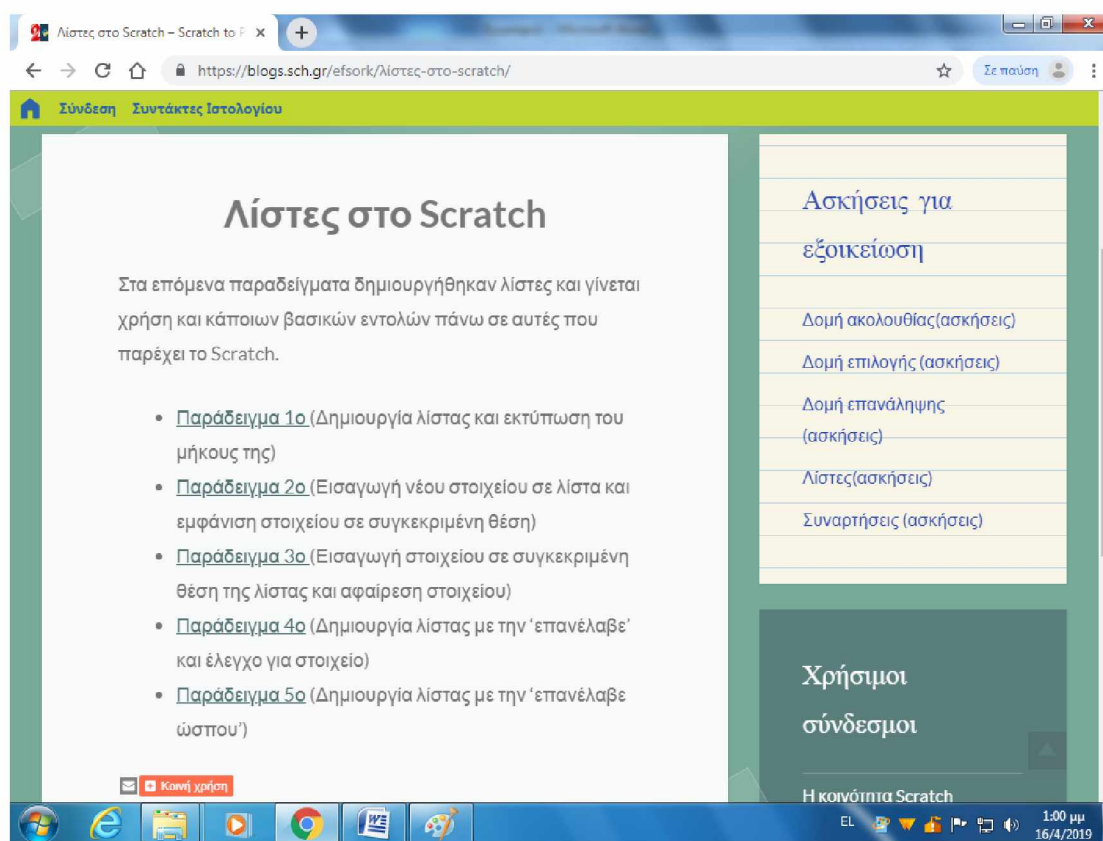


Εικόνα 5.66: Σενάριο στο Scratch, η εντολή ‘επανάλαβε ώσπου...’ (5^η)

Όπως παρατηρείται από την προηγούμενη εικόνα, οι μεταβλητές που χρησιμοποιήθηκαν ως μετρητές, αρχικοποιούνται κατά την έναρξη του σεναρίου στην τιμή 'μηδέν'. Επιπλέον κάθε φορά, ανάλογα με το αν ο αριθμός που εισάγεται είναι θετικός ή αρνητικός, οι αντίστοιχοι μετρητές αυξάνονται κατά ένα, ώστε τελικά να έχουμε το πλήθος των θετικών και των αρνητικών αριθμών που δόθηκαν από το χρήστη.

Το επόμενο στοιχείο του μενού **‘Σενάρια σε Scratch’** που θα αναλυθεί είναι οι **‘Λίστες’**.

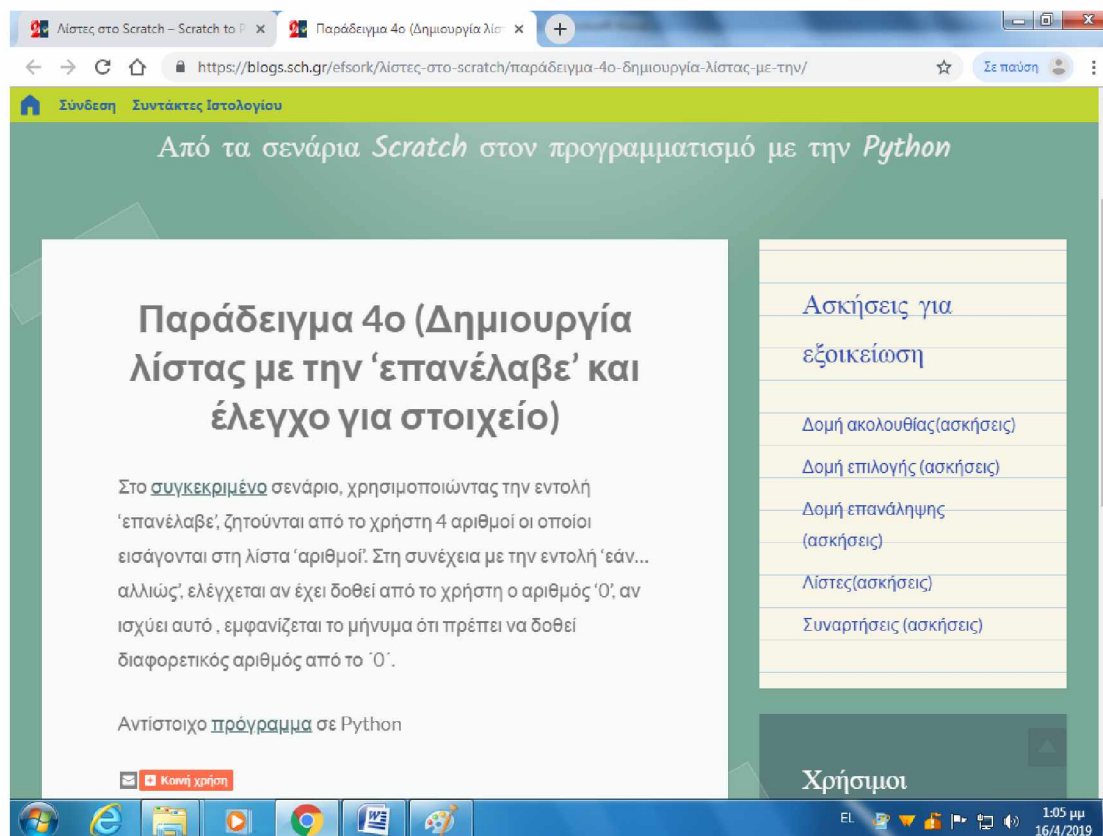
Στην επόμενη εικόνα παρουσιάζεται η οθόνη του ιστολογίου που περιέχει το συγκεκριμένο στοιχείο.



Εικόνα 5.67: Λίστες στο Scratch

Στη σελίδα υπάρχουν έξι παραδείγματα που αφορούν τη δομή της ‘Λίστας’ στον προγραμματισμό. Τα παραδείγματα παρουσιάζουν τον τρόπο με τον οποίο δημιουργούνται λίστες στο Scratch, χρησιμοποιώντας κάποιες από τις βασικές εντολές.

Αρχικά θα παρουσιαστεί και θα αναλυθεί το τέταρτο παράδειγμα της σελίδας, όπως απεικονίζεται στη συνέχεια.



Εικόνα 5.68: Δημιουργία λίστας με την 'επανάλαβε' (4^η)

Στο τέταρτο παράδειγμα δημιουργείται μια λίστα τεσσάρων αριθμών, χρησιμοποιώντας την εντολή 'επανάλαβε'. Οι αριθμοί ζητούνται και εισάγονται από το χρήστη. Σε περίπτωση που δοθεί ο αριθμός 'μηδέν' το σενάριο εμφανίζει το μήνυμα στο χρήστη ότι πρέπει να εισάγει διαφορετικό αριθμό από το 'μηδέν'. Στη συνέχεια γίνεται έλεγχος αν το μήκος της λίστας είναι μικρότερο του τέσσερα, αυτό μπορεί να συμβεί μόνο αν έχει προηγουμένως δοθεί το 'μηδέν' και αυτό δεν έχει εισαχθεί στη λίστα, σε περίπτωση που αυτό ισχύει ζητείται από το χρήστη επιπλέον αριθμός. Τελικά το σενάριο τυπώνει τη λίστα των αριθμών.

Στην εικόνα που ακολουθεί εμφανίζεται το περιβάλλον Scratch του σεναρίου.

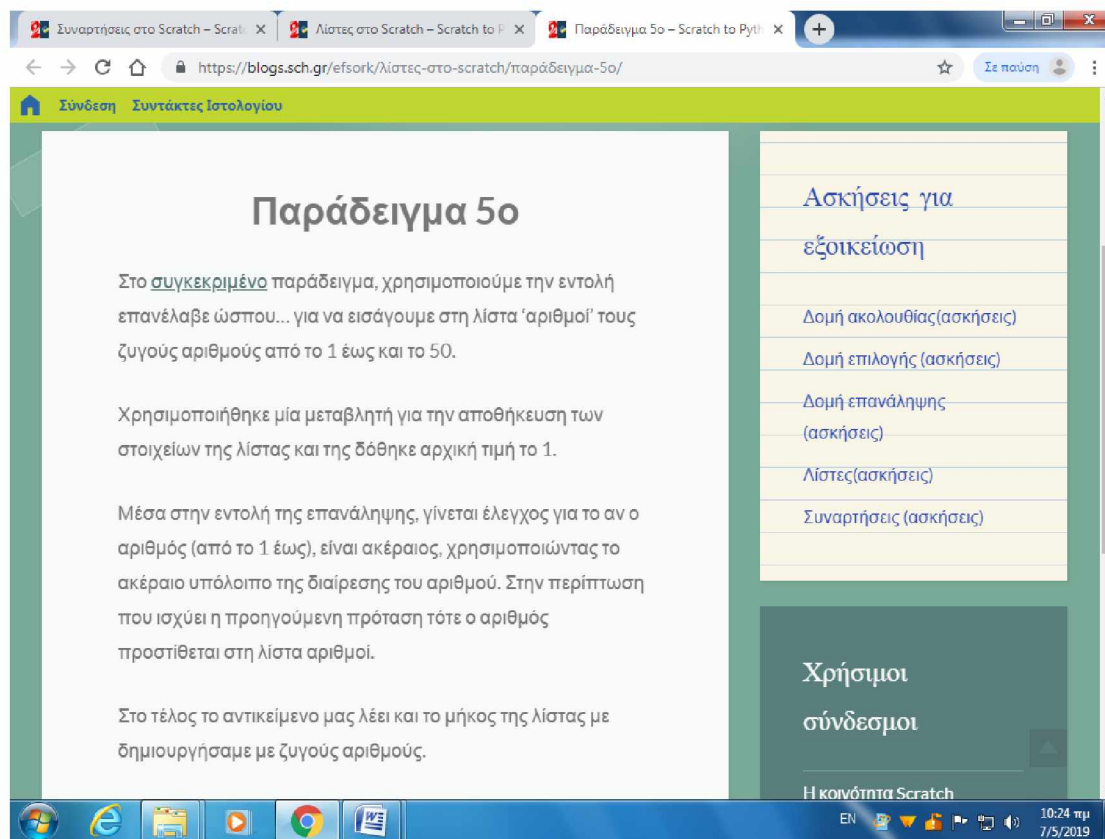


Εικόνα 5.69: Περιβάλλον Scratch, δημιουργία λίστας (4^η)



Εικόνα 5.70: Εκτέλεση σεναρίου, δημιουργία λίστας (4^η)

Στη συνέχεια ακολουθεί η παρουσίαση και η περιγραφή του πέμπτου παραδείγματος του στοιχείου ‘Λίστες’ του ιστολογίου, όπως εμφανίζεται στην επόμενη εικόνα.



Εικόνα 5.71: Δημιουργία λίστας (5^n)

Στο συγκεκριμένο παράδειγμα, το σενάριο δημιουργεί μια λίστα με τους ζυγούς αριθμούς οι οποίοι περιέχονται στο διάστημα των αριθμών από το 1 έως και το 50. Η εντολή που επιτυγχάνει την προαναφερόμενη λειτουργία είναι η ‘επανάλαβε ώσπου...’. Μέσα στο block της επανάληψης υπάρχει και η εντολή ‘εάν’ με την οποία ελεγχεται το ακέραιο υπόλοιπο της διαίρεσης του κάθε αριθμού στο διάστημα 1 έως 50, με το δύο είναι ίσο με το μηδέν (ζυγός αριθμός).

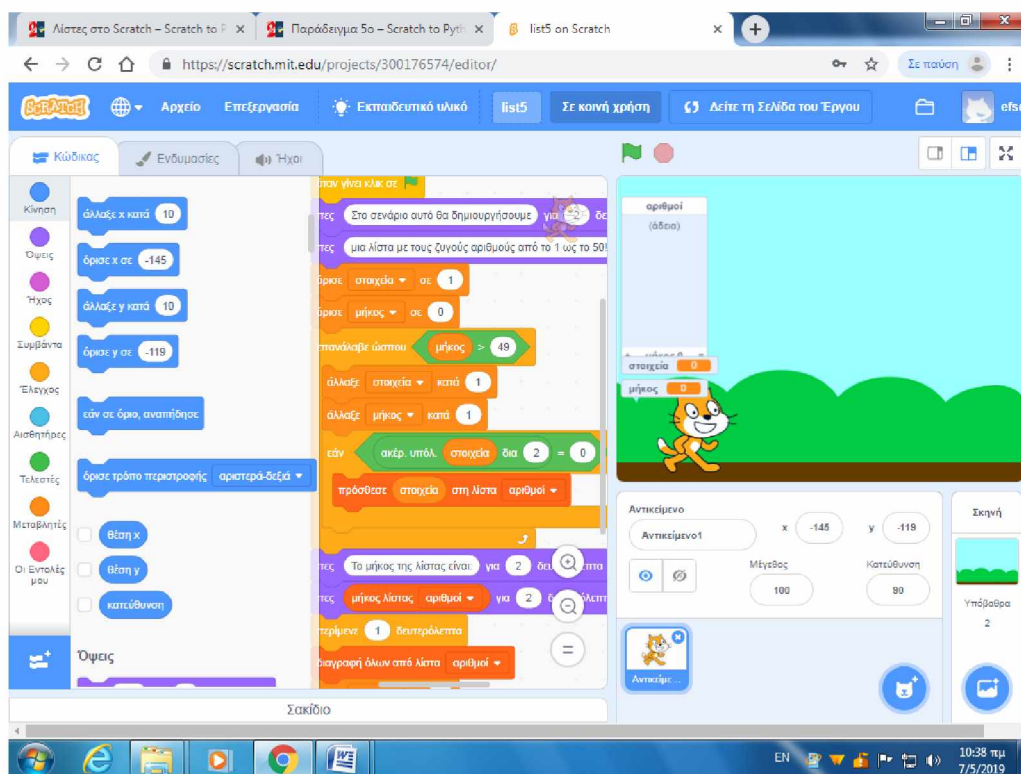
Αν ισχύει η συνθήκη τότε το σενάριο προσθέτει τον αριθμό στη λίστα. Τελικά εκτυπώνεται στην οθόνη η λίστα που δημιουργήθηκε, καθώς και το μήκος της (πλήθος στοιχείων).

Ακολουθώντας ο χρήστης τον υπερσύνδεσμο, μεταφέρεται στη σελίδα του Scratch στην οποία είναι αποθηκευμένο το σενάριο, όπως απεικονίζεται στη συνέχεια.



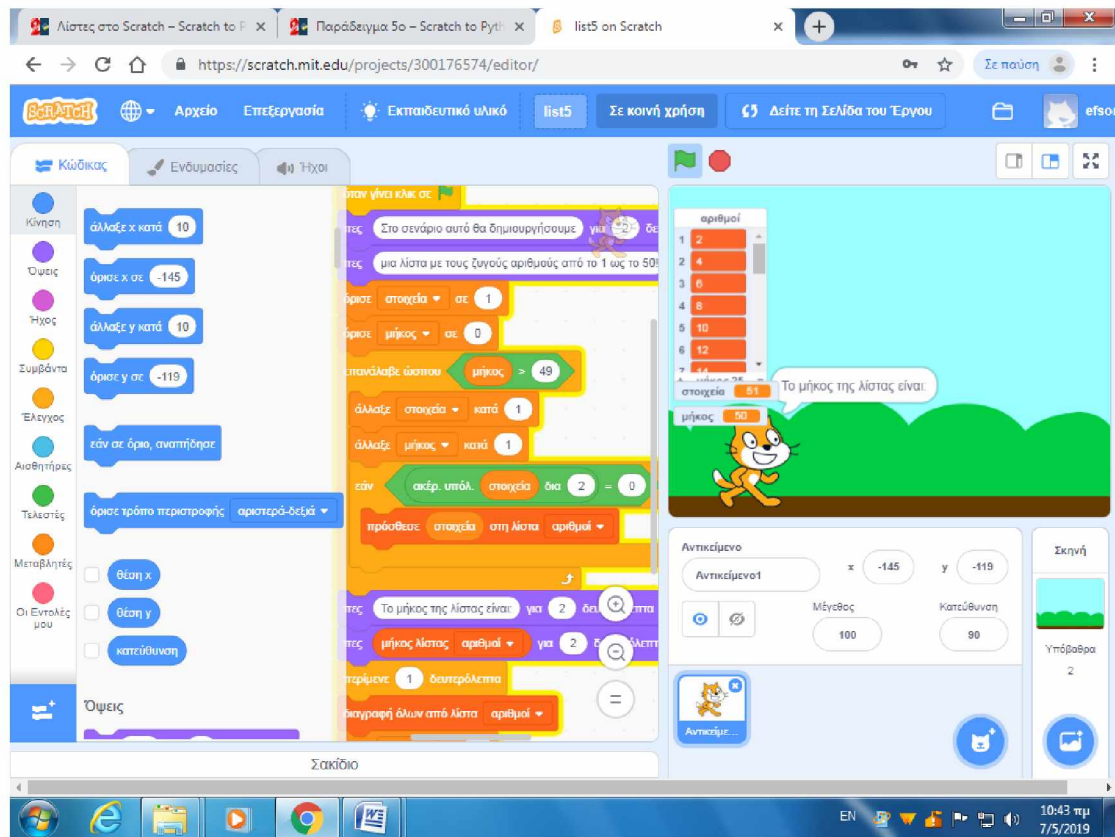
Εικόνα 5.72: Περιβάλλον Scratch, Λίστες (5^η)

Σε περίπτωση που ο χρήστης επιλέξει να εισέλθει στο περιβάλλον των εντολών του σεναρίου η οθόνη που προκύπτει είναι η επόμενη.



Εικόνα 5.73: Περιβάλλον Scratch, εντολές σεναρίου, Λίστες(5^η)

Εκτελώντας το σενάριο που αντιστοιχεί στο πέμπτο παράδειγμα η εικόνα που προκύπτει περιέχει τη λίστα που δημιουργήθηκε καθώς και το μήκος της, όπως ακολουθεί στην Εικόνα 5.74.

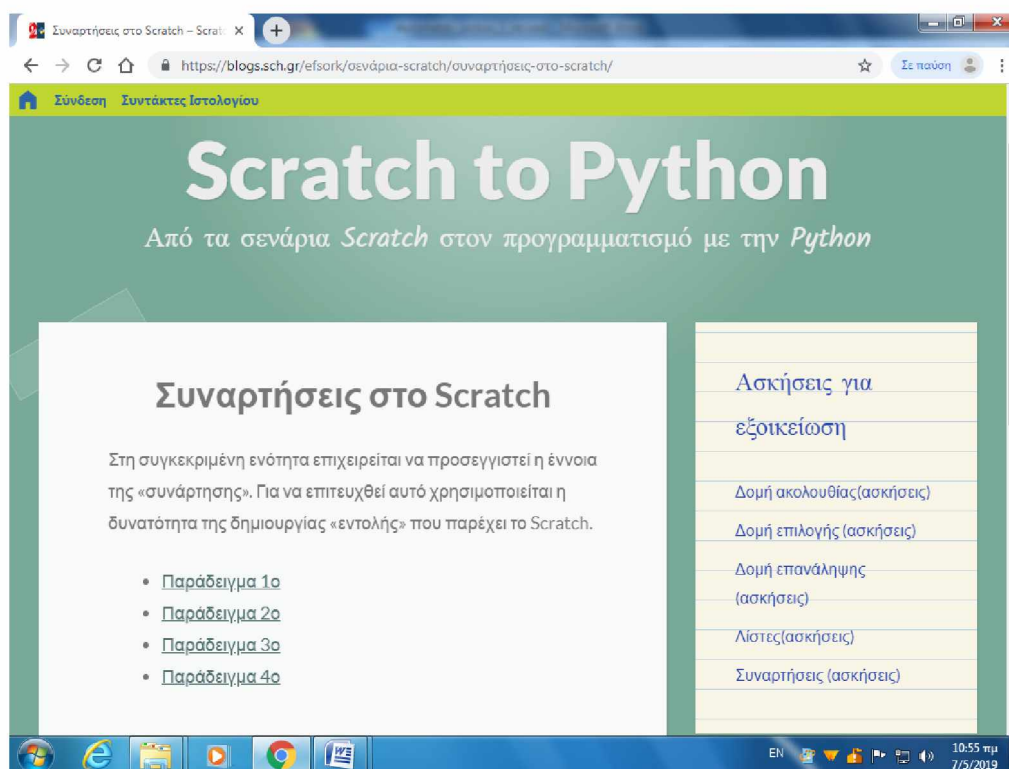


Εικόνα 5.74: Εκτέλεση σεναρίου, Λίστες (5^η)

Συνεχίζοντας, το στοιχείο που ακολουθεί για παρουσίαση και περιγραφή είναι οι ‘Συναρτήσεις’ με το Scratch.

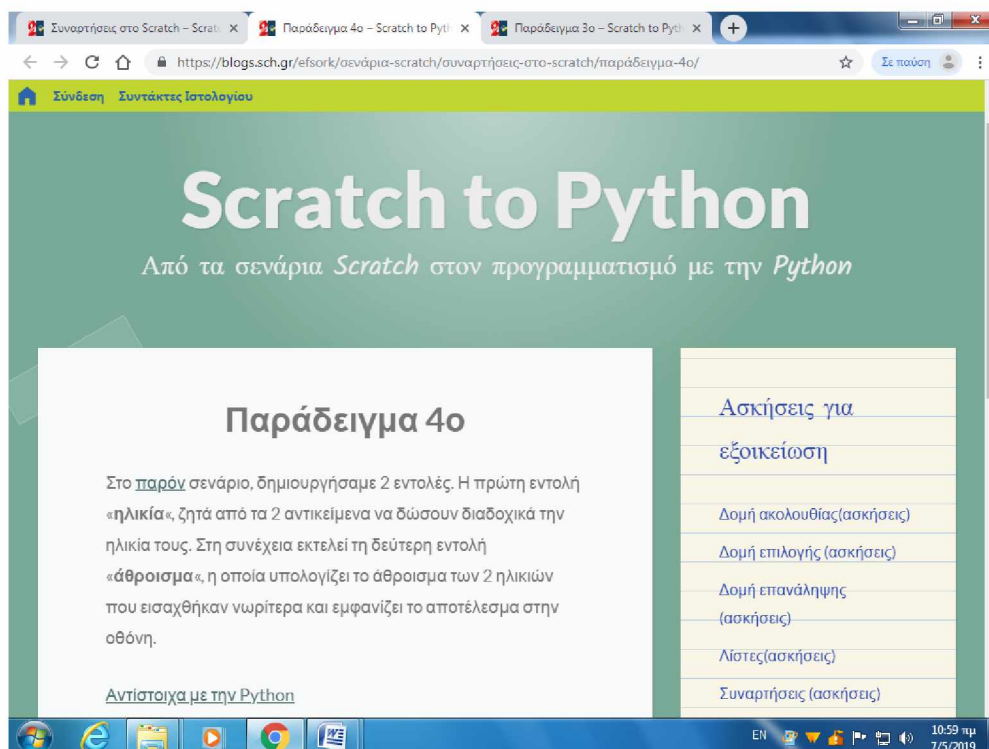
Για να επιτευχθεί η δημιουργία συναρτήσεων με τη χρήση του Scratch, η παροχή της δυνατότητας ‘Δημιουργία εντολής’ είναι η καταλληλότερη. Η σελίδα περιέχει τέσσερα παραδείγματα δημιουργίας συναρτήσεων - εντολών.

Στην επόμενη εικόνα εμφανίζεται η σελίδα του ιστολογίου η οποία περιέχει τα παραδείγματα που αφορούν στις συναρτήσεις.



Εικόνα 5.75: Συναρτήσεις στο Scratch

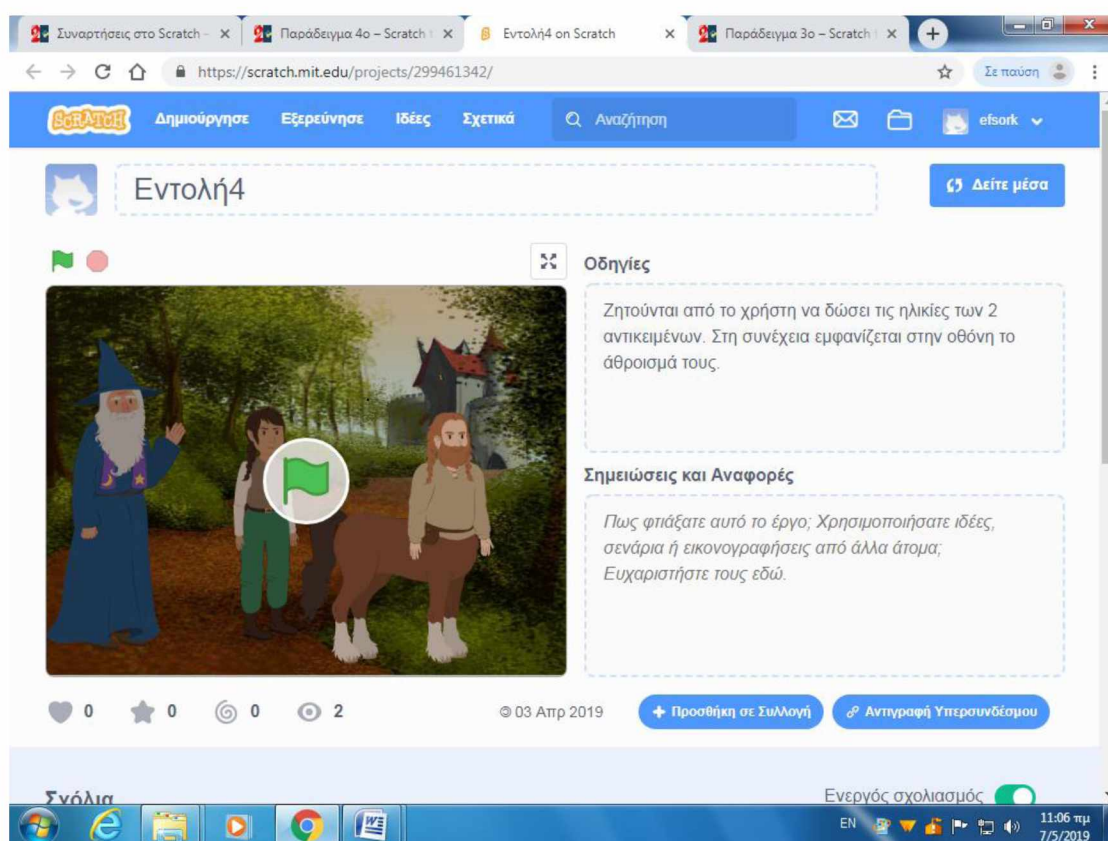
Το παράδειγμα που επιλέχθηκε για παρουσίαση είναι το τέταρτο και η οθόνη που προκύπτει επιλέγοντάς το είναι η επόμενη.



Εικόνα 5.76: Συναρτήσεις (4^η)

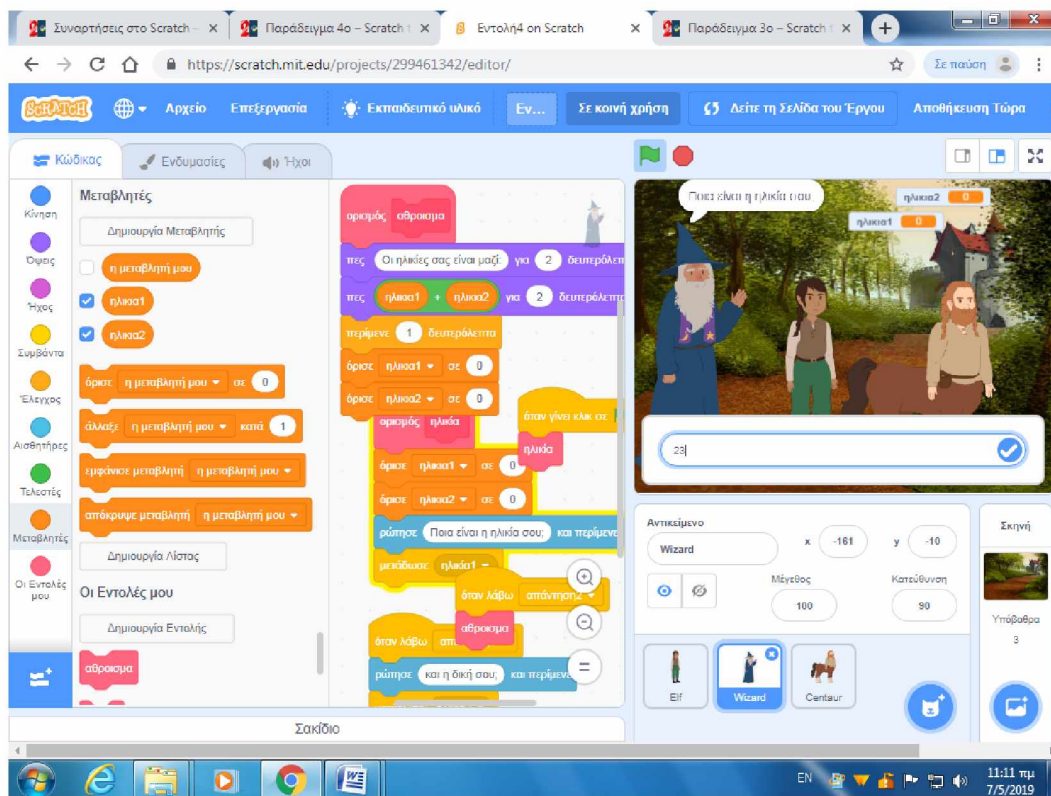
Στο εν λόγω παράδειγμα δημιουργηθήκαν με το Scratch δύο εντολές. Η πρώτη εντολή έχει το όνομα 'ηλικία' και ζητά από τα δύο αντικείμενα του σεναρίου να εισάγουν διαδοχικά την ηλικία τους.

Η δεύτερη εντολή έχει το όνομα 'άθροισμα' και εκτελείται μέσα στο κυρίως μέρος της πρώτης εντολής. Η λειτουργία της δεύτερης εντολής είναι να υπολογίζει και να εκτυπώνει στην οθόνη το άθροισμα των ηλικιών των αντικειμένων. Στη εικόνα που ακολουθεί εμφανίζεται το περιβάλλον Scratch στο οποίο είναι αποθηκευμένο το παράδειγμα.

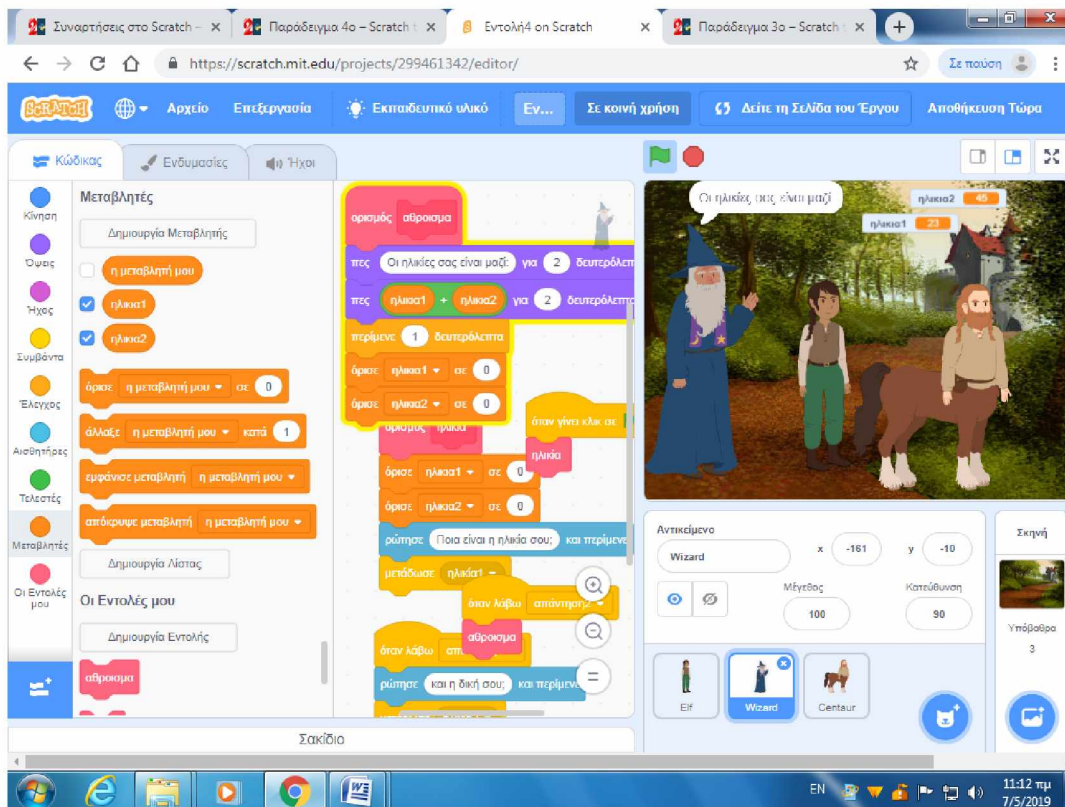


Εικόνα 5.77: Περιβάλλον Scratch-Συναρτήσεις (4^η)

Εισερχόμενος στο παράδειγμα ο χρήστης έχει τη δυνατότητα να δει και να μελετήσει ποιες εντολές χρησιμοποιήθηκαν για την δημιουργία του σεναρίου, όπως εμφανίζεται στη συνέχεια.



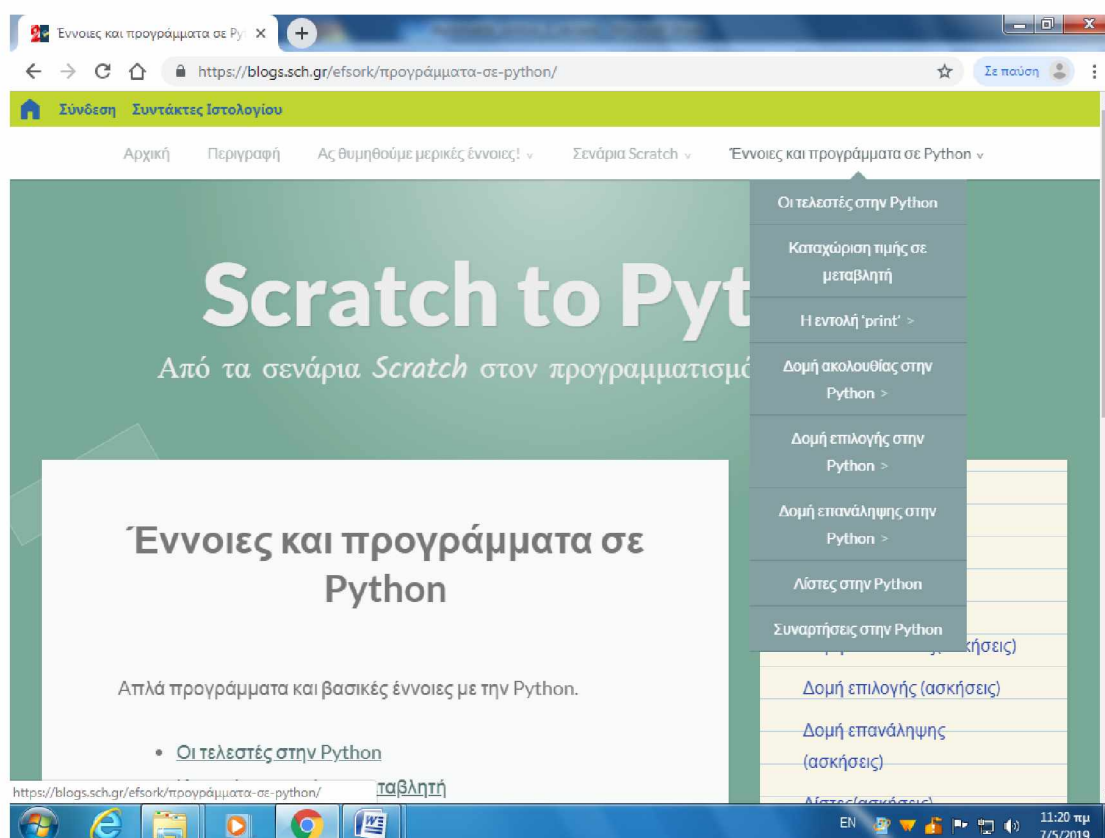
Εικόνα 5.78: Συνάρτηση-Ηλικία 1^{οβ} αντικειμένου (4^η)



Εικόνα 5.79: Συνάρτηση-Ηλικία 2^{οβ} αντικειμένου (4^η)

Στο σημείο αυτό της παρούσας εργασίας τελειώνει η παρουσίαση και η περιγραφή των παραδειγμάτων σε Scratch, των βασικών δομών.

Στη συνέχεια ακολουθεί παρουσίαση και περιγραφή των ίδιων λειτουργιών αλλά χρησιμοποιώντας την γλώσσα προγραμματισμού Python. Επομένως το στοιχείο του πρωταρχικού μενού του ιστολογίου που ακολουθεί είναι το ‘Έννοιες και προγράμματα σε Python’, όπως απεικονίζεται στη συνέχεια.



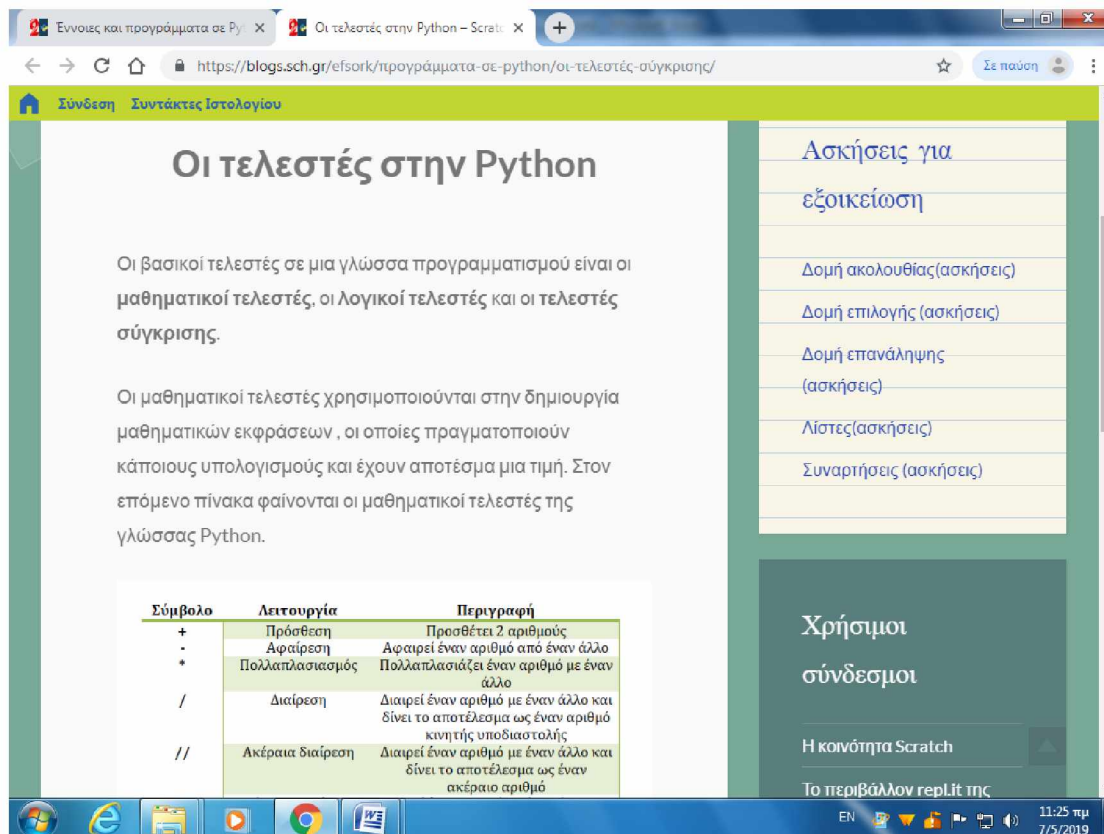
Εικόνα 5.80: Έννοιες και προγράμματα σε Python

Τα βασικά μέρη του συγκεκριμένου στοιχείου είναι:

- Οι τελεστές στην Python
- Καταχώριση τιμής σε μεταβλητή
- Η εντολή ‘print’
- Δομή ακολουθίας στην Python
- Δομή επιλογής στην Python
- Δομή επανάληψης στην Python

- Λίστες στην Python
- Συναρτήσεις στην Python

Το στοιχείο ‘Οι τελεστές στην Python’, περιέχει μια βασική θεωρία για τους τελεστές και τα είδη τους, όπως φαίνεται στη επόμενη εικόνα.

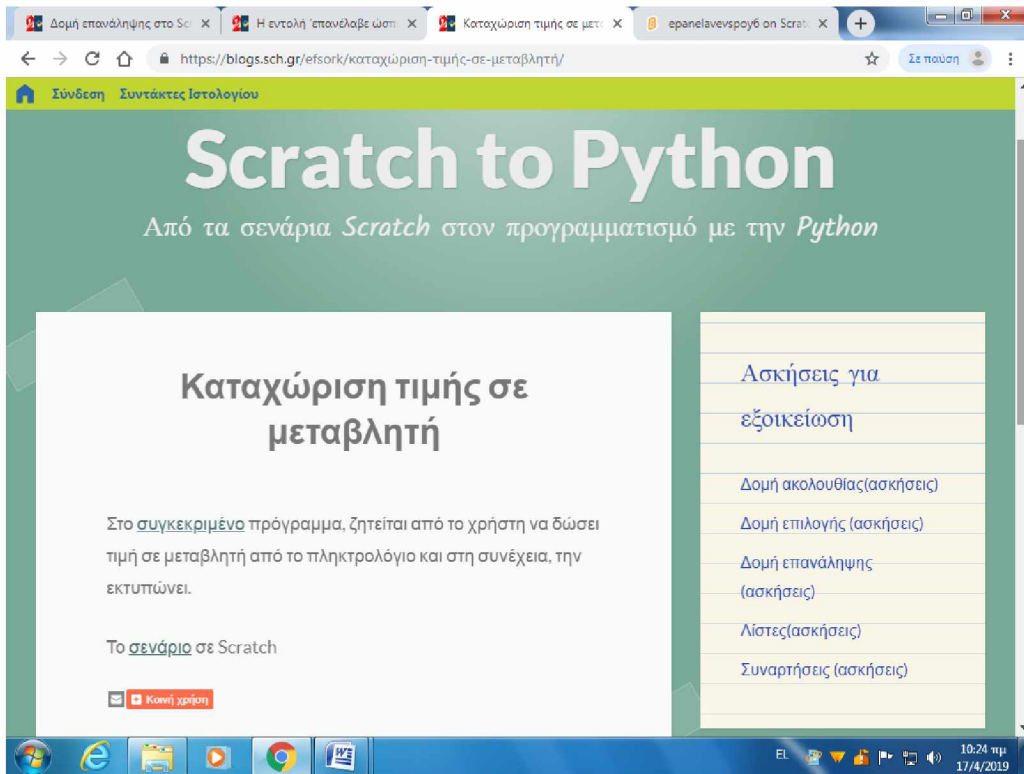


Εικόνα 5.81: Οι τελεστές στην Python

Συνεχίζοντας παρατηρείται ότι σε όλες τις σελίδες του στοιχείου ‘Έννοιες και προγράμματα σε Python’, υπάρχει και υπερσύνδεσμος για την υλοποίηση του παραδείγματος με το Scratch.

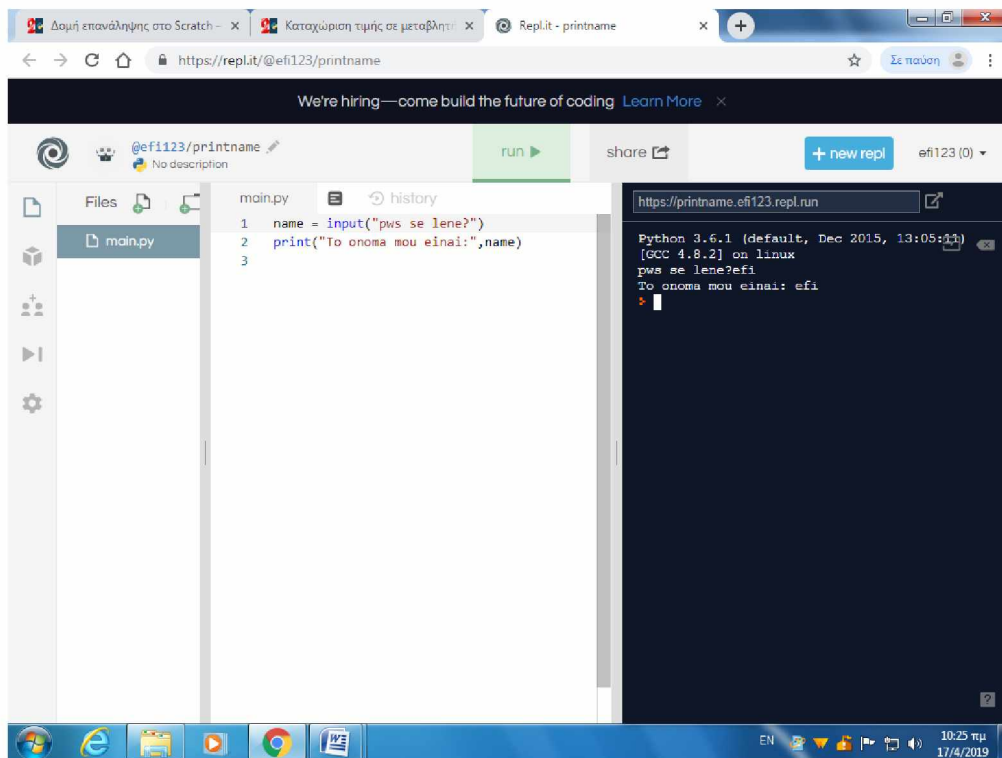
Η ύπαρξη του υπερσυνδέσμου για το Scratch, βοηθά τον χρήστη για άμεση υπενθύμιση του τρόπου υλοποίησης του παραδείγματος με τη χρήση του Scratch.

Η επόμενη σελίδα του ιστολογίου που θα παρουσιαστεί είναι η ‘Καταχώριση τιμής σε μεταβλητή’, όπως απεικονίζεται στην επόμενη εικόνα.



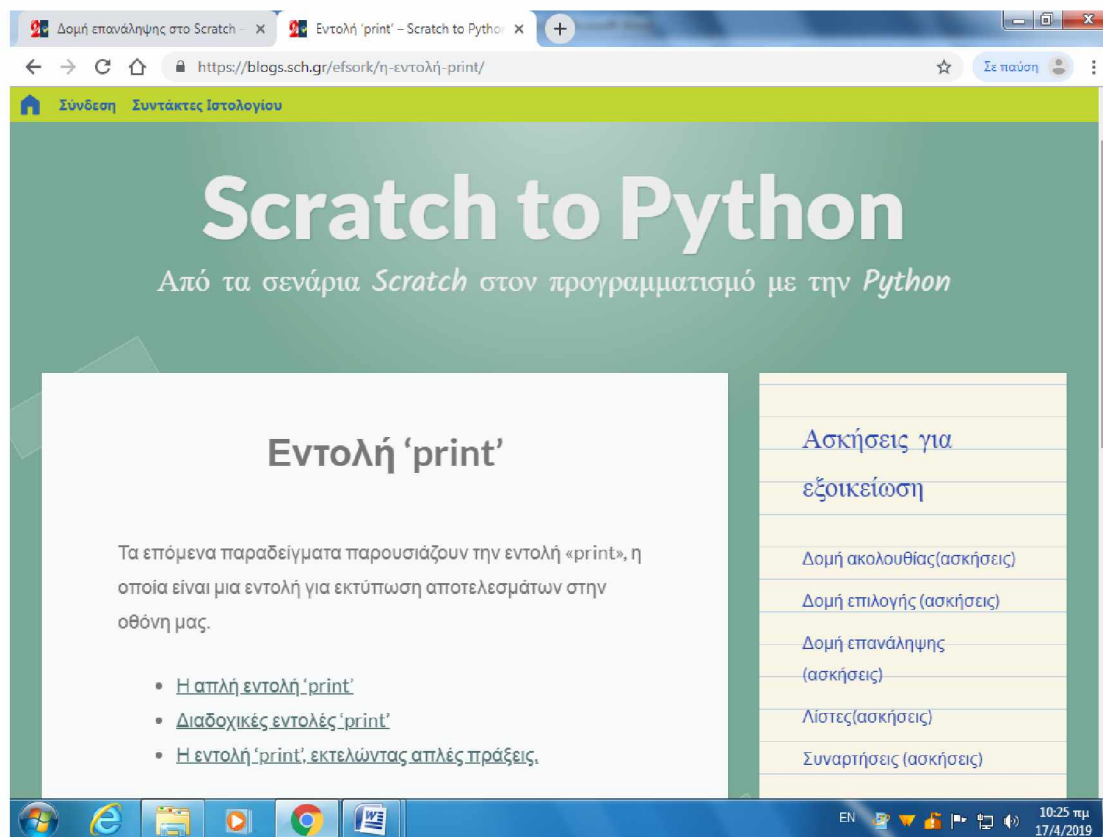
Εικόνα 5.82: Καταχώριση τιμής σε μεταβλητή με την Python

Σε αυτή τη σελίδα του ιστολογίου υπάρχει ένας υπερσύνδεσμος που οδηγεί στο περιβάλλον replit της Python (εικόνα 5.83).



Εικόνα 5.83: Καταχώριση τιμής σε μεταβλητή-περιβάλλον replit

Το στοιχείο του μενού που ακολουθεί είναι 'Η εντολή print'. Στη σελίδα αυτή περιέχονται τρεις υπερσύνδεσμοι οι οποίοι οδηγούν σε αντίστοιχα παραδείγματα με την Python (εικόνα 5.84).



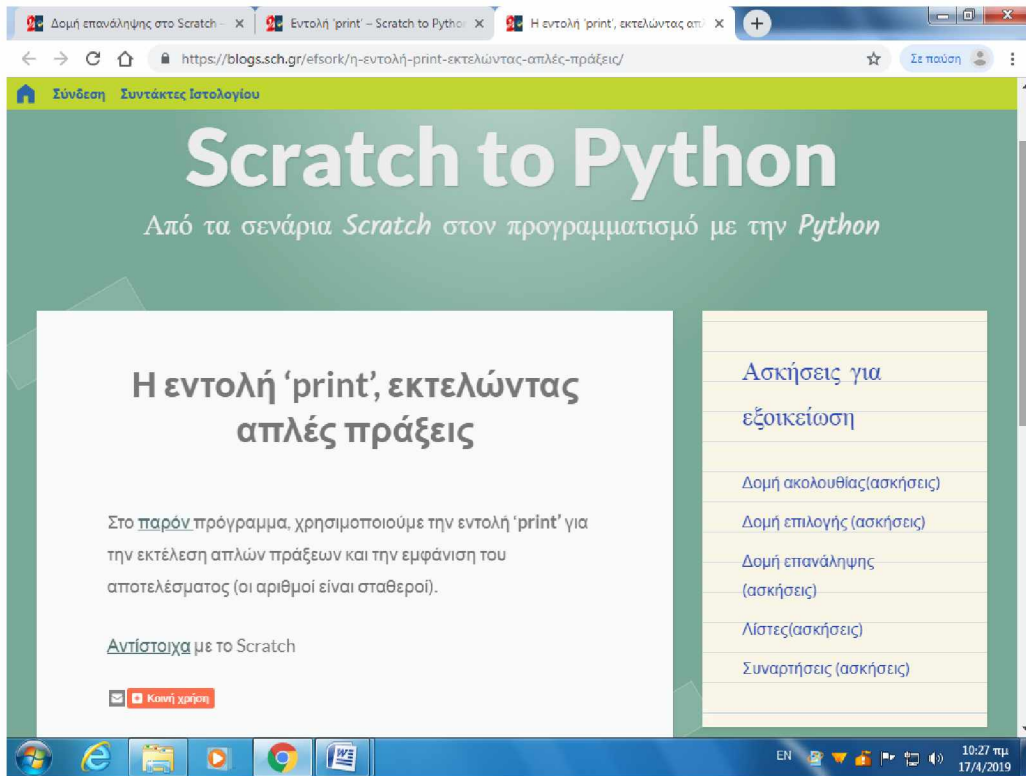
Εικόνα 5.84: Εντολή 'print'

Όπως παρατηρείται από την εικόνα 5.84, η σελίδα περιέχει τα εξής στοιχεία που αφορούν στην εντολή 'print'.

- Η απλή εντολή 'print'
- Διαδοχικές εντολές 'print'
- Η εντολή 'print' εκτελώντας απλές πράξεις

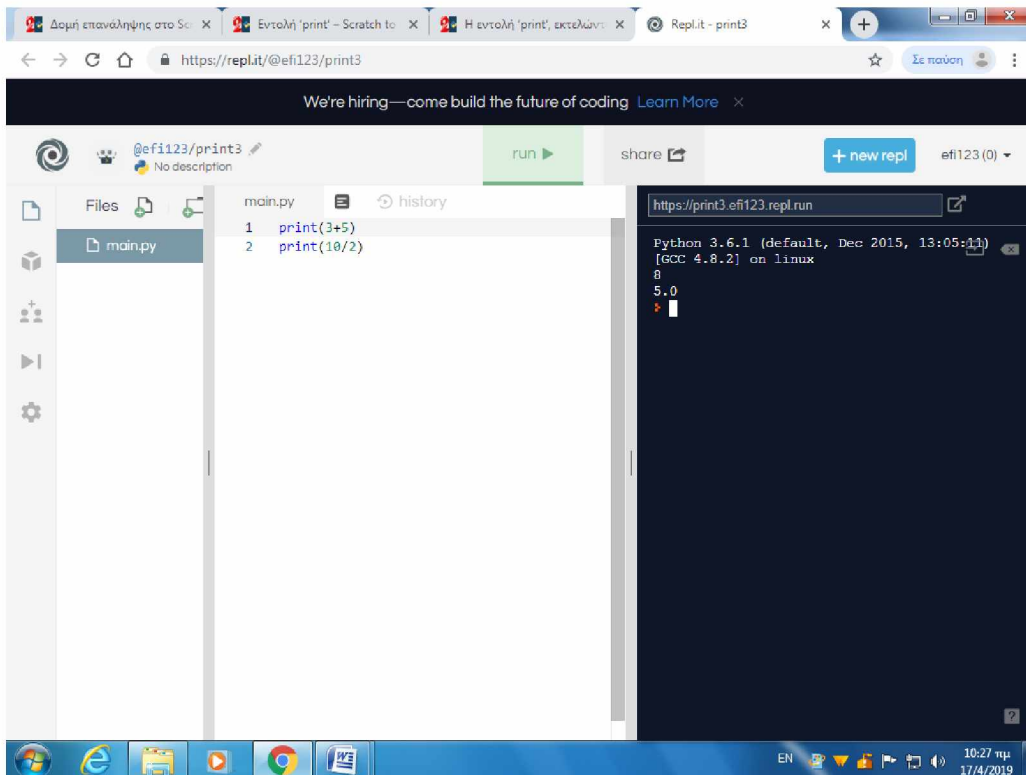
Το στοιχείο που θα παρουσιαστεί αναλυτικά από τα προηγούμενα είναι το τρίτο (εικόνα 5.85). Στο παράδειγμα αυτό χρησιμοποιείται η εντολή 'print', για την εκτέλεση απλών πράξεων και την εμφάνιση του αποτελέσματος.

Σημαντικό είναι να αναφερθεί ότι οι αριθμοί έχουν σταθερή τιμή.



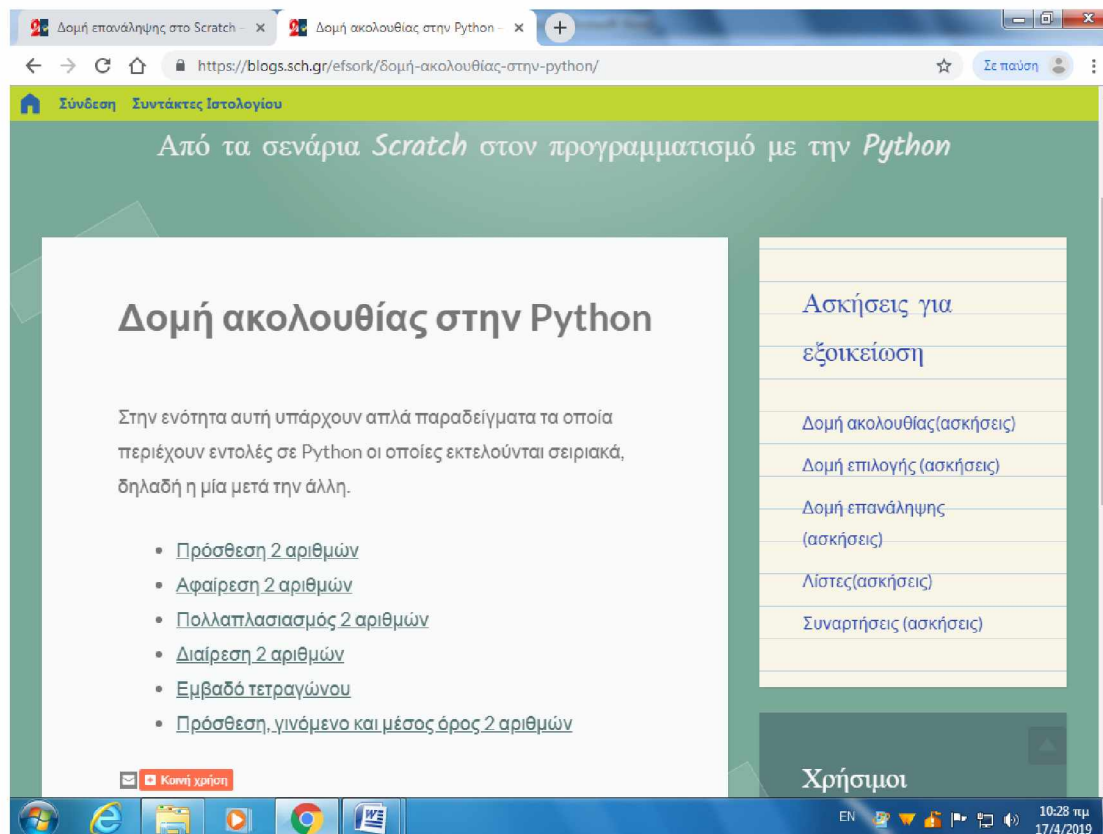
Εικόνα 5.85: Η εντολή 'print', εκτελώντας απλές πράξεις

Στην εικόνα 5.86 που ακολουθεί εμφανίζεται το παράδειγμα στο περιβάλλον replit.



Εικόνα 5.86: Εντολή 'print', εκτελώντας απλές πράξεις-replit

Το επόμενο στοιχείο του μενού των προγραμμάτων Python, είναι η ‘Δομή ακολουθίας στην Python’ (εικόνα 5.87).

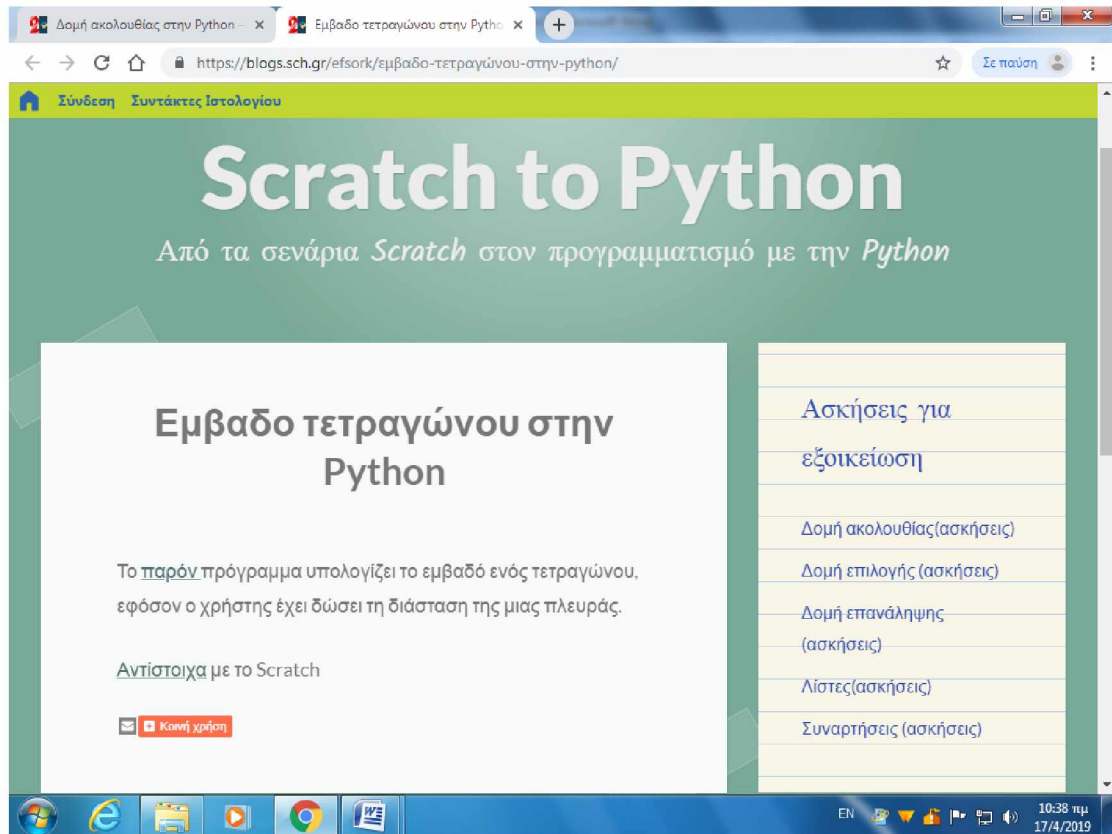


Εικόνα 5.87: Δομή ακολουθίας στην Python

Η εν λόγω σελίδα περιέχει διάφορα παραδείγματα, των οποίων η λειτουργία έχει υλοποιηθεί και με το Scratch, τα οποία είναι:

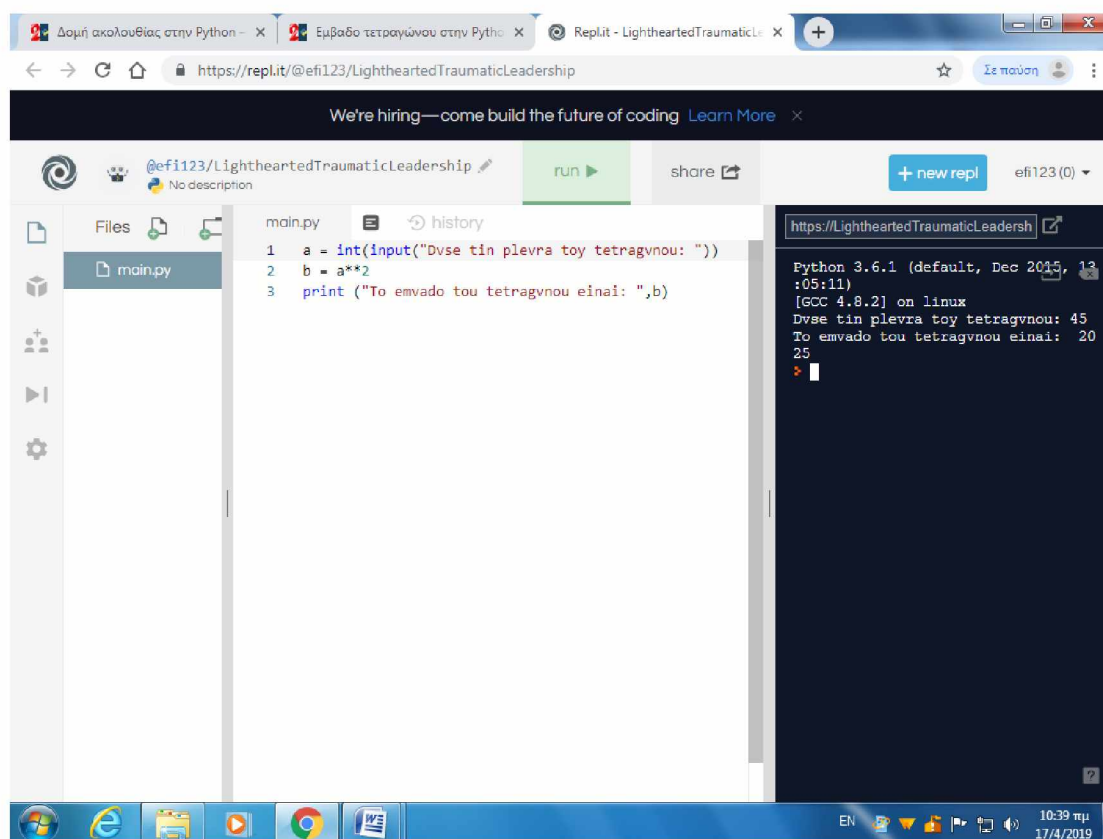
- Πρόσθεση 2 αριθμών
- Αφαίρεση 2 αριθμών
- Πολλαπλασιασμός 2 αριθμών
- Διαίρεση 2 αριθμών
- Εμβαδό τετραγώνου
- Πρόσθεση γινόμενο και μέσος όρος 2 αριθμών.

Στην παρούσα αναφορά θα γίνει μία σύντομη παρουσίαση του παραδείγματος με τον υπολογισμό του εμβαδού τετραγώνου, με την Python (εικόνα 5.88).



Εικόνα 5.88: Εμβαδό τετραγώνου στην Python

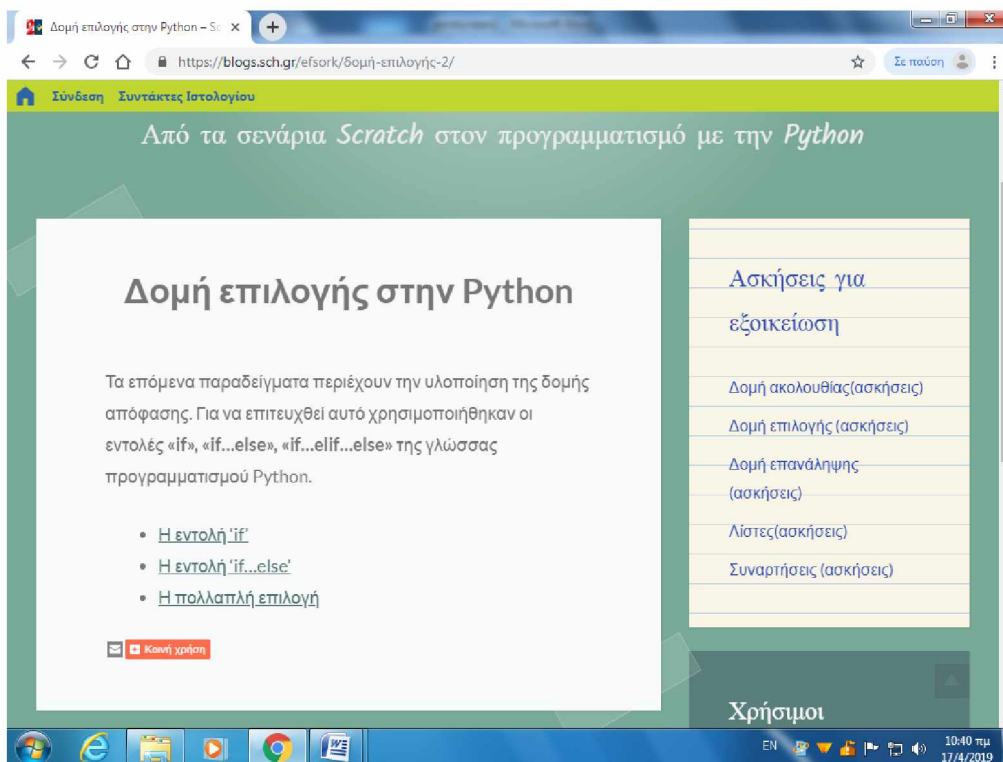
Στο συγκεκριμένο παράδειγμα ο χρήστης δίνει τη διάσταση της μιας πλευράς ενός τετραγώνου και το πρόγραμμα χρησιμοποιώντας τον μαθηματικό τύπο για τον υπολογισμό του εμβαδού, το υπολογίζει και εμφανίζει το αποτέλεσμα στην οθόνη, όπως απεικονίζεται στη συνέχεια στο περιβάλλον replit.



Εικόνα 5.89: Υπολογισμός εμβαδού τετραγώνου-Εκτέλεση προγράμματος, replit

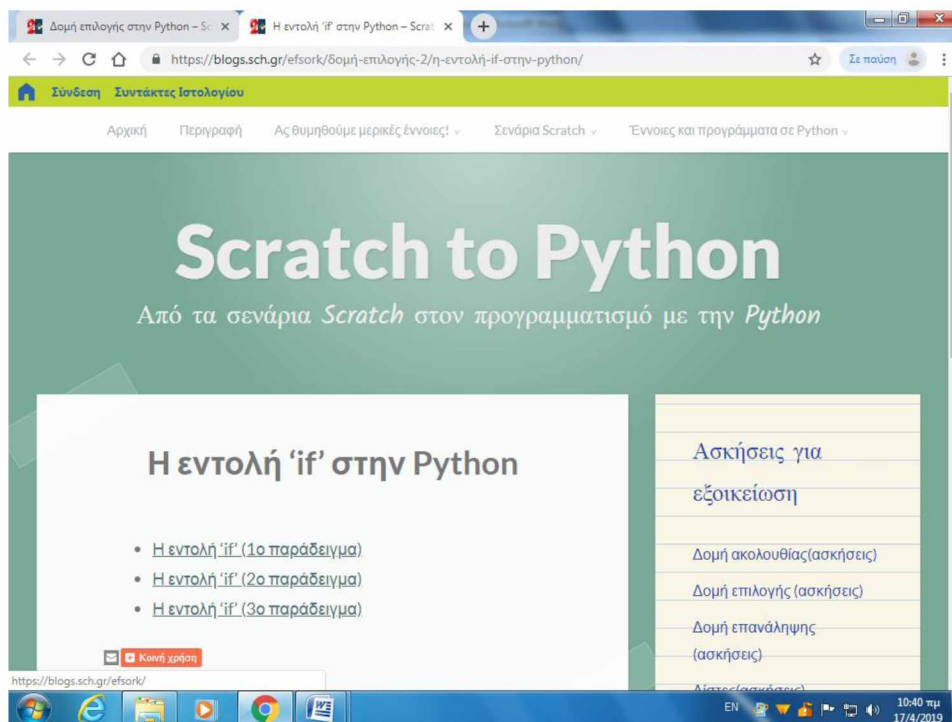
Στη συνέχεια ακολουθεί η σελίδα με τη Δομή επιλογής (εικόνα 5.90). Όπως παρατηρείται στην επόμενη εικόνα η σελίδα περιέχει τις τρεις βασικές εντολές που αφορούν στην δομή επιλογής στον προγραμματισμό.

- Η εντολή 'if'
- Η εντολή 'if...else'
- Η πολλαπλή επιλογή (η οποία υλοποιείται με την εντολή 'if...elif...else')



Εικόνα 5.90: Δομή επιλογής στην Python

Επιλέγοντας ο χρήστης την εντολή 'if', η σελίδα η οποία εμφανίζεται στην οθόνη είναι στην επόμενη εικόνα.

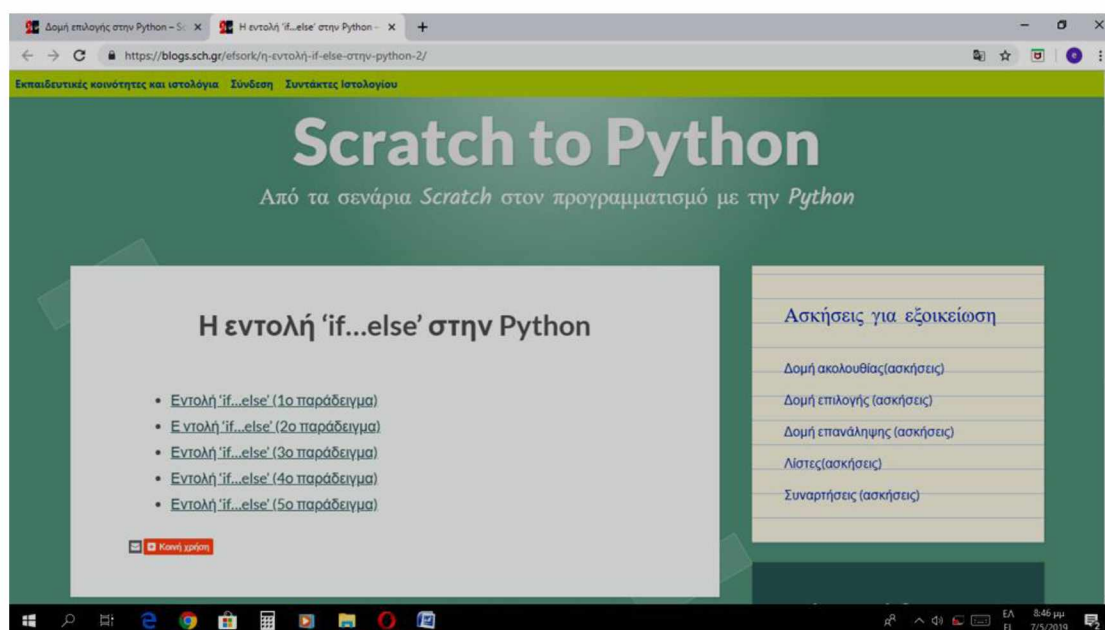


Εικόνα 5.91: Η εντολή 'if' στην Python

Στη σελίδα με την εντολή 'if' περιέχονται τρία παραδείγματα τα οποία υλοποιούνται με την Python στο περιβάλλον replit.

Στην παρούσα εργασία θα παρουσιαστούν αναλυτικά κάποια παραδείγματα από την εντολή 'if...else' και από την πολλαπλή επιλογή.

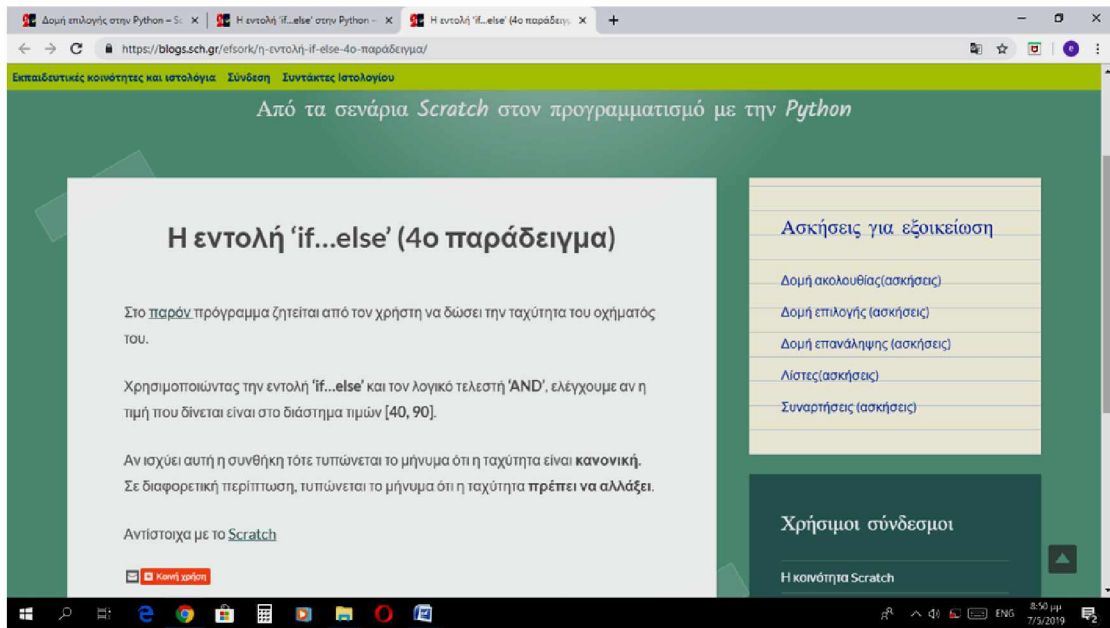
Η εικόνα που ακολουθεί εμφανίζει την οθόνη που προκύπτει όταν ο χρήστης εισέλθει στη σελίδα των παραδειγμάτων της εντολής 'if...else'.



Εικόνα 5.92: Η εντολή 'if...else' στην Python

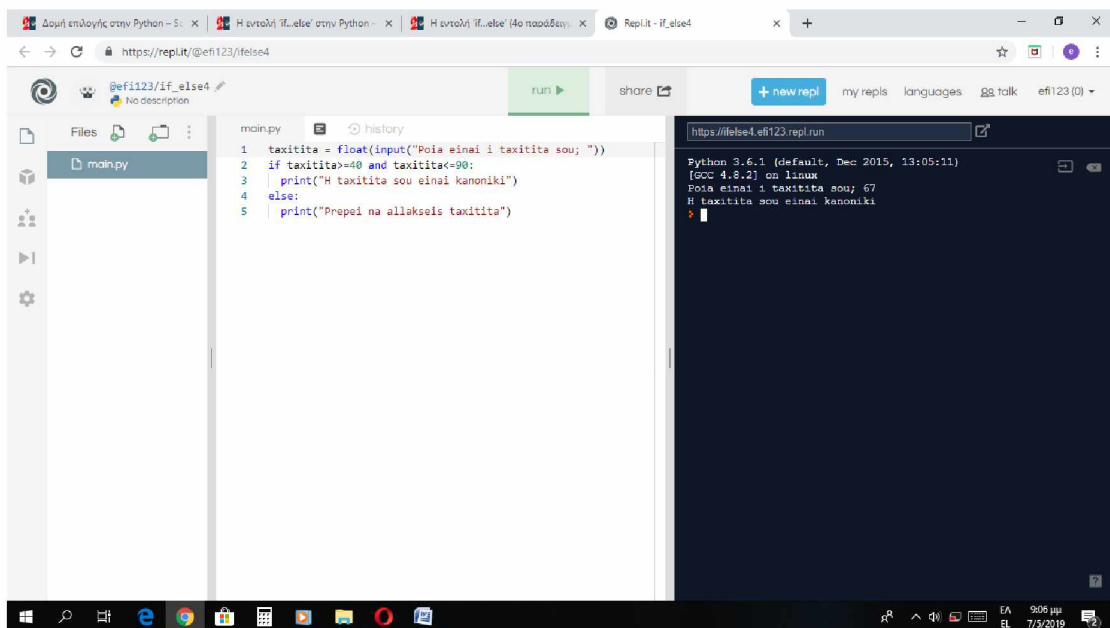
Όπως προκύπτει από την προηγούμενη εικόνα, η σελίδα της εντολής 'if...else', περιέχει πέντε παραδείγματα τα οποία έχουν υλοποιηθεί με τη χρήση της γλώσσας προγραμματισμού Python στο περιβάλλον replit.

Στην παρούσα αναφορά θα αναλυθεί το τέταρτο παράδειγμα της εντολής (εικόνα 5.93). Στο παράδειγμα αυτό το πρόγραμμα ζητά από το χρήστη να δώσει έναν αριθμό που να χαρακτηρίζει την ταχύτητα του οχήματός του. Σε περίπτωση που η δοθείσα τιμή είναι στο διάστημα 40 έως και 90, το πρόγραμμα εμφανίζει στον χρήστη το μήνυμα ότι έχει κανονική ταχύτητα. Εάν η τιμή που δοθεί είναι εκτός του προαναφερόμενου διαστήματος, τότε τυπώνεται στην οθόνη το μήνυμα ότι ο χρήστης πρέπει να αλλάξει ταχύτητα. Τα προηγούμενα που περιγράφηκαν υλοποιούνται με τη χρήση της εντολής 'if...else' και του λογικού τελεστή 'AND'.



Εικόνα 5.93: Η εντολή ‘if...else’ (4^η)

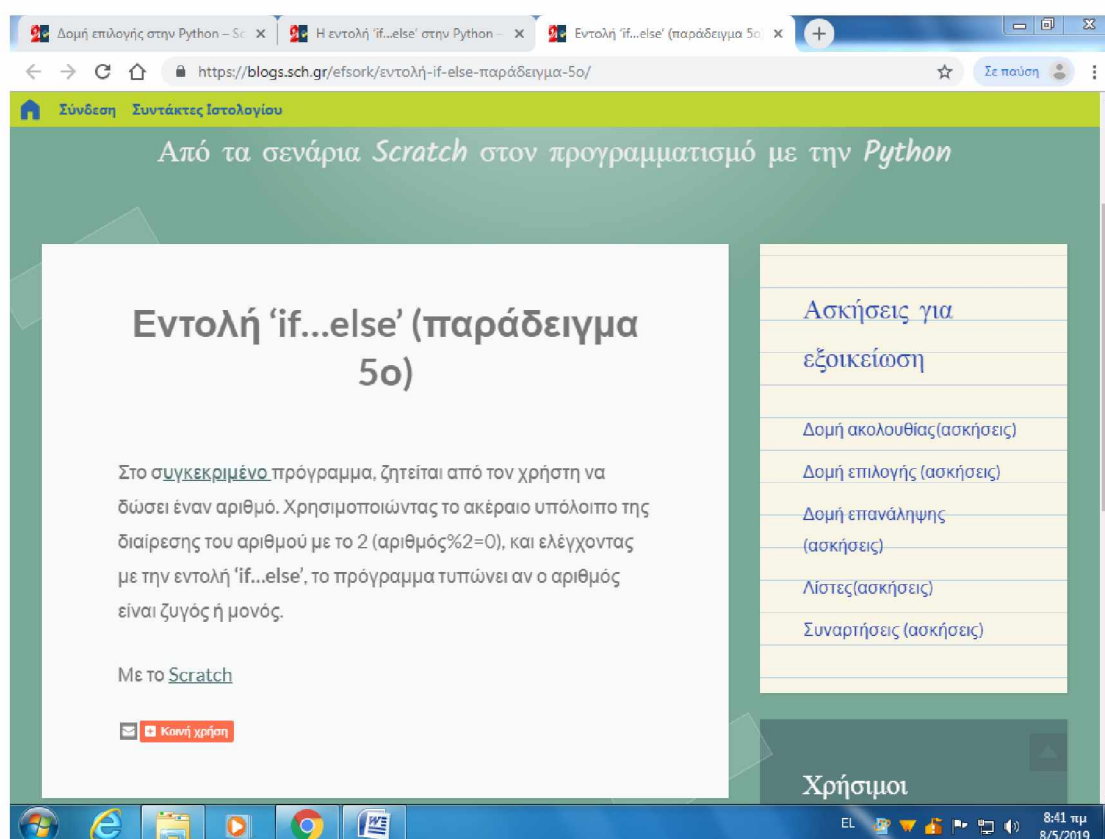
Στην εικόνα που ακολουθεί παρουσιάζεται το πρόγραμμα του παραδείγματος σε Python καθώς και η εκτέλεσή του.



Εικόνα 5.94: Εκτέλεση προγράμματος, εντολή ‘if...else’ (4^η)

Στο επόμενο (πέμπτο) παράδειγμα της εντολής 'if...else' που θα παρουσιαστεί και θα περιγραφεί το πρόγραμμα ζητά από το χρήστη να εισάγει έναν αριθμό. Στη συνέχεια χρησιμοποιώντας την εντολή 'if...else', καταλήγει εάν ο αριθμός που δόθηκε είναι ζυγός ή μονός.

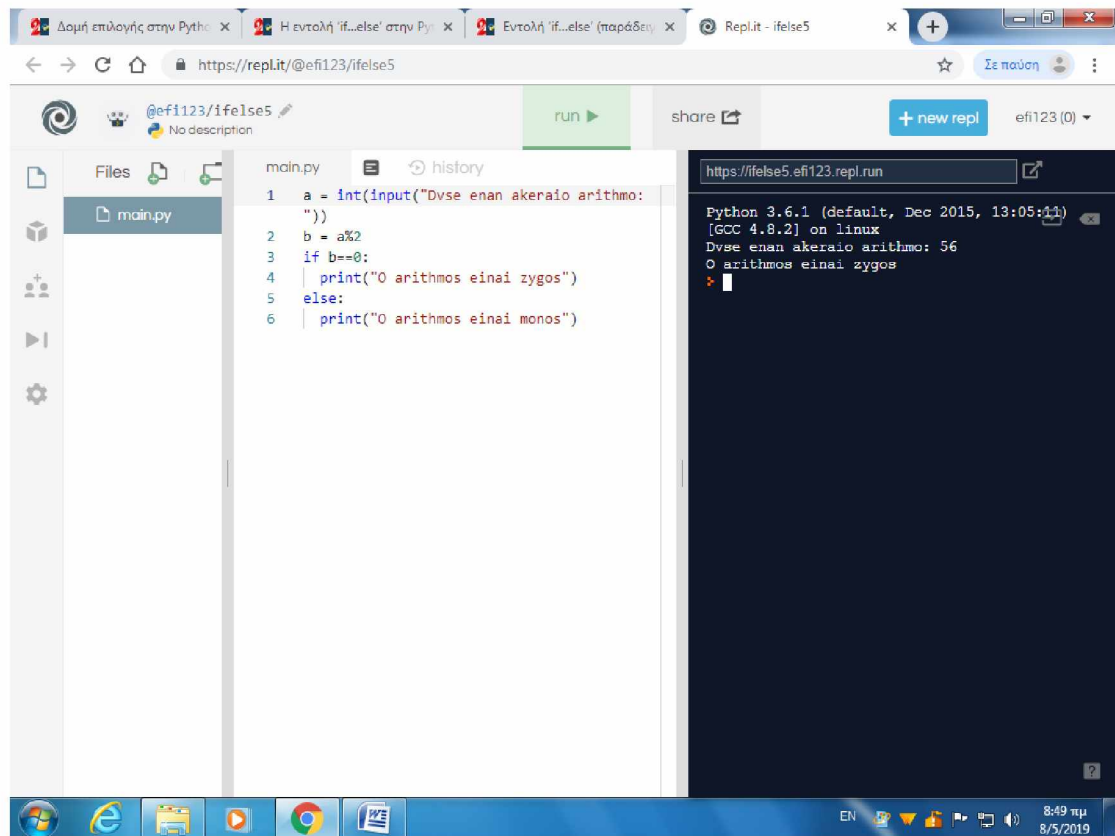
Στην επόμενη εικόνα παρουσιάζεται η οθόνη του πέμπτου παραδείγματος της εντολής.



Εικόνα 5.95: Εντολή 'if...else' (5^η)

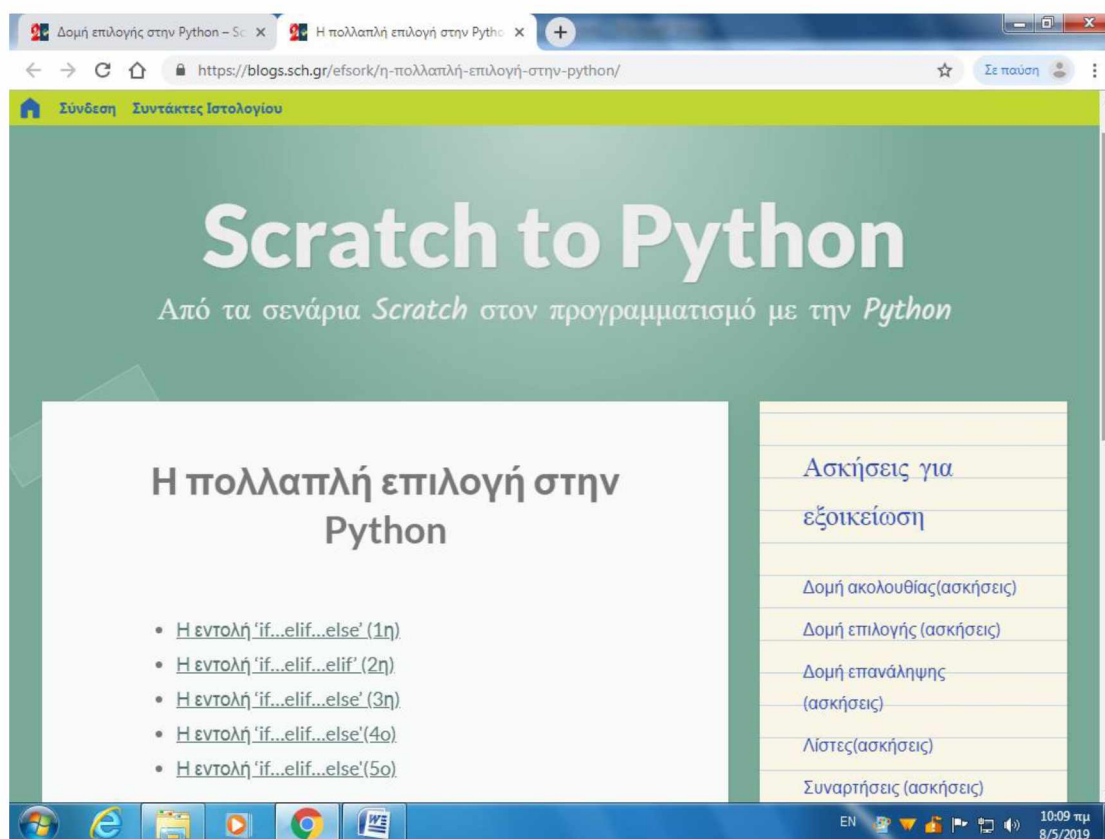
Όπως φαίνεται στην προηγούμενη εικόνα στη σελίδα του παραδείγματος υπάρχει υπερσύνδεσμος για το περιβάλλον replit, στο οποίο υπάρχουν αποθηκευμένα τα παραδείγματα σε Python.

Ακολουθώντας τον συγκεκριμένο υπερσύνδεσμο ο χρήστης η οθόνη που προκύπτει εμφανίζεται στη συνέχεια.



Εικόνα 5.96: Εντολή 'if...else', εκτέλεση προγράμματος (5^η)

Στη συνέχεια της αναφοράς, ακολουθεί το παράδειγμα που υλοποιήθηκε σε Python για την εκμάθηση της πολλαπλής επιλογής. Η σελίδα που περιέχει παραδείγματα με την συγκεκριμένη δομή ακολουθεί στην επόμενη εικόνα.

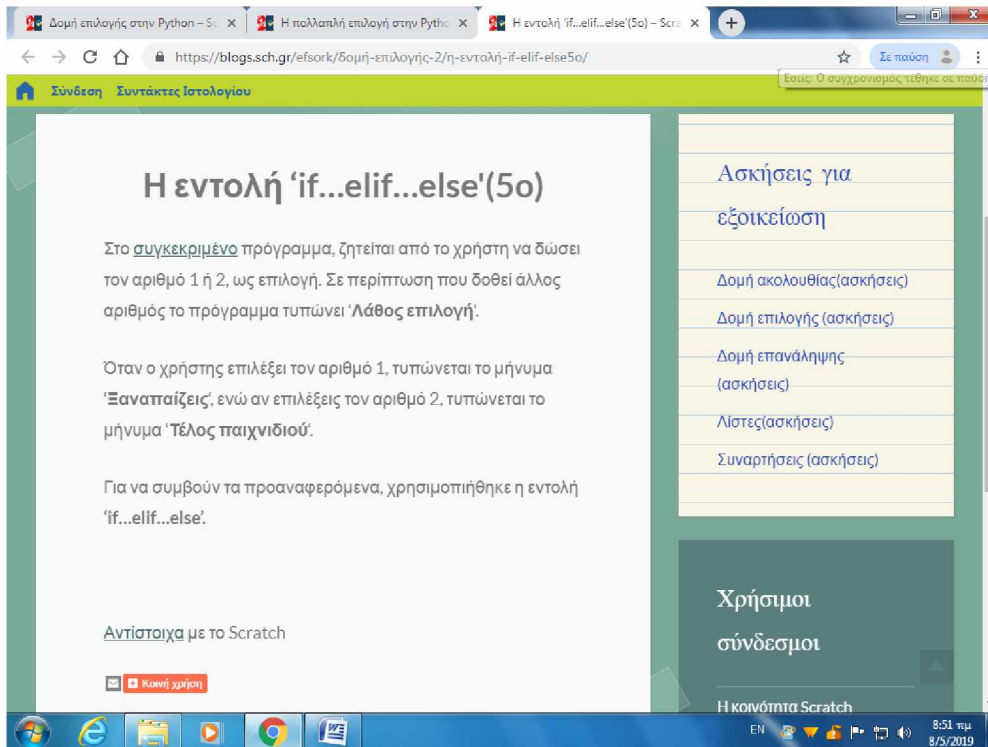


Εικόνα 5.97: Η πολλαπλή επιλογή στην Python

Όπως παρατηρείται η σελίδα περιέχει πέντε παραδείγματα τα οποία αφορούν στην εντολή 'if...elif...else', της Python.

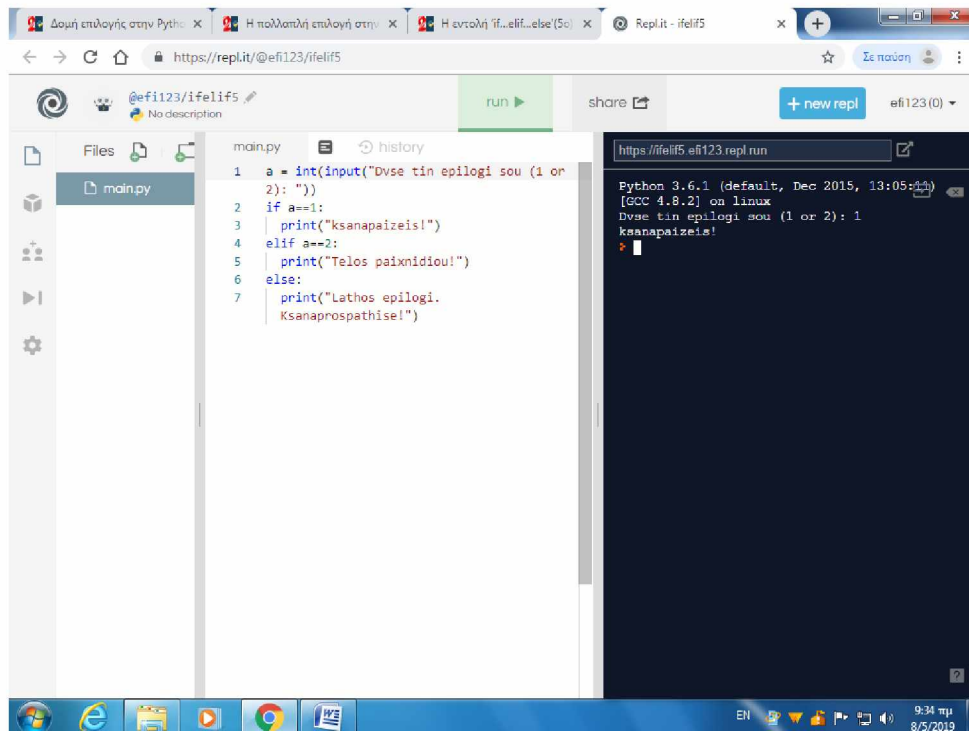
Στο πέμπτο παράδειγμα, που εμφανίζεται στην εικόνα που ακολουθεί, το πρόγραμμα ζητά από το χρήστη να δώσει σαν επιλογή τον αριθμό '1' ή '2'. Σε περίπτωση που δοθεί διαφορετική τιμή το πρόγραμμα τυπώνει το μήνυμα 'Λάθος επιλογή'.

Όταν ο χρήστης επιλέξει τον αριθμό '1' τότε το πρόγραμμα τυπώνει το μήνυμα 'Ξαναπαίζεις' ενώ αν επιλέξει τον αριθμό '2' τυπώνει το μήνυμα 'Τέλος παιχνιδιου'. Τα προαναφερόμενα για να υλοποιηθούν χρησιμοποιείται η εντολή 'if...elif...else', της Python.



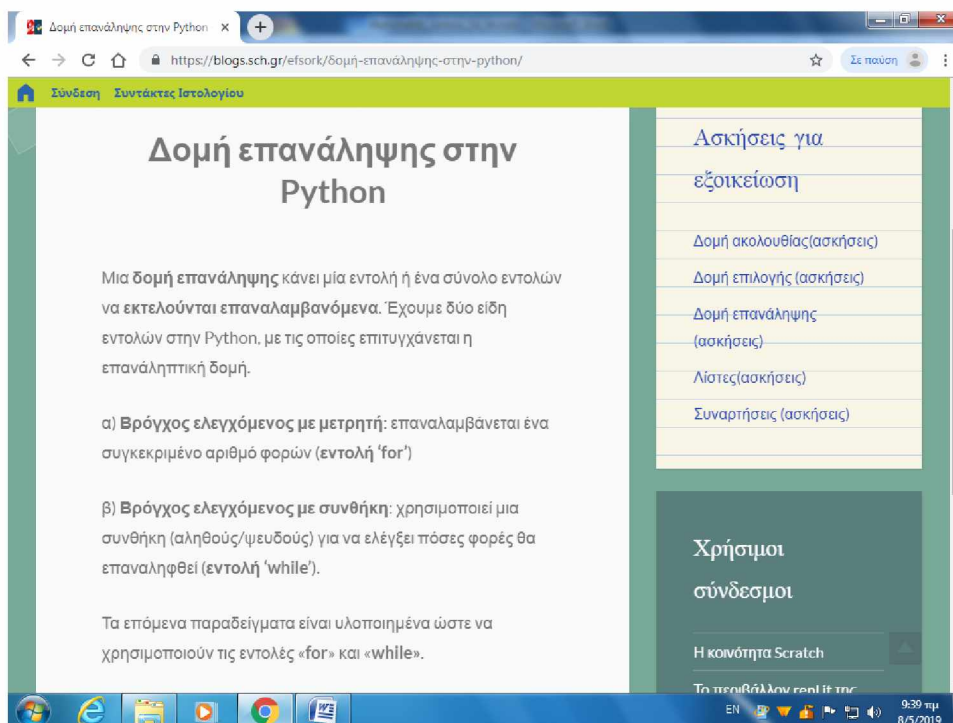
Εικόνα 5.98: Η εντολή ‘if...elif...else’ (5^η)

Όταν ο χρήστης επιλέξει τον υπερσύνδεσμο για το πρόγραμμα που είναι υλοποιημένο σε Python η οθόνη που μεταφέρεται ακολουθεί στη συνέχεια.



Εικόνα 5.99: Η εντολή ‘if...elif...else’-replit (5^η)

Στη συνέχεια ακολουθεί το στοιχείο ‘Δομή επανάληψης στην Python’ και σχετικά παραδείγματα. Η σελίδα της συγκεκριμένης εντολής απεικονίζεται στη συνέχεια.

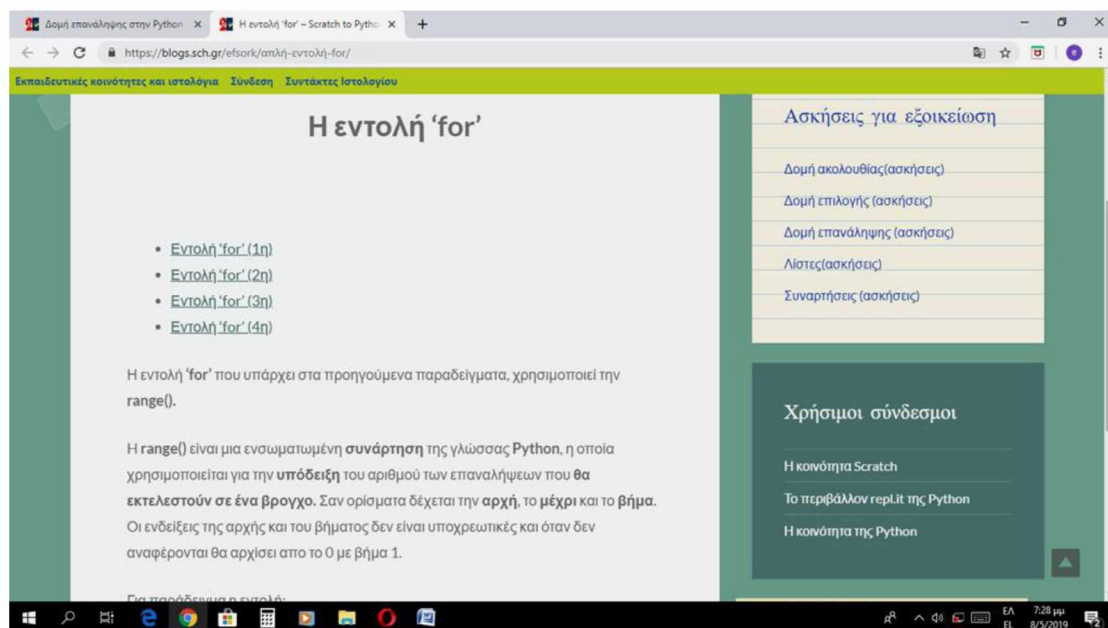


Εικόνα 5.100: Δομή επανάληψης στην Python

Στην εικονιζόμενη σελίδα περιέχεται η βασική θεωρία για τη δομή της επανάληψης στον προγραμματισμό και στη συνέχεια οι τρεις βασικές κατηγορίες με τις οποίες υλοποιείται η επανάληψη.

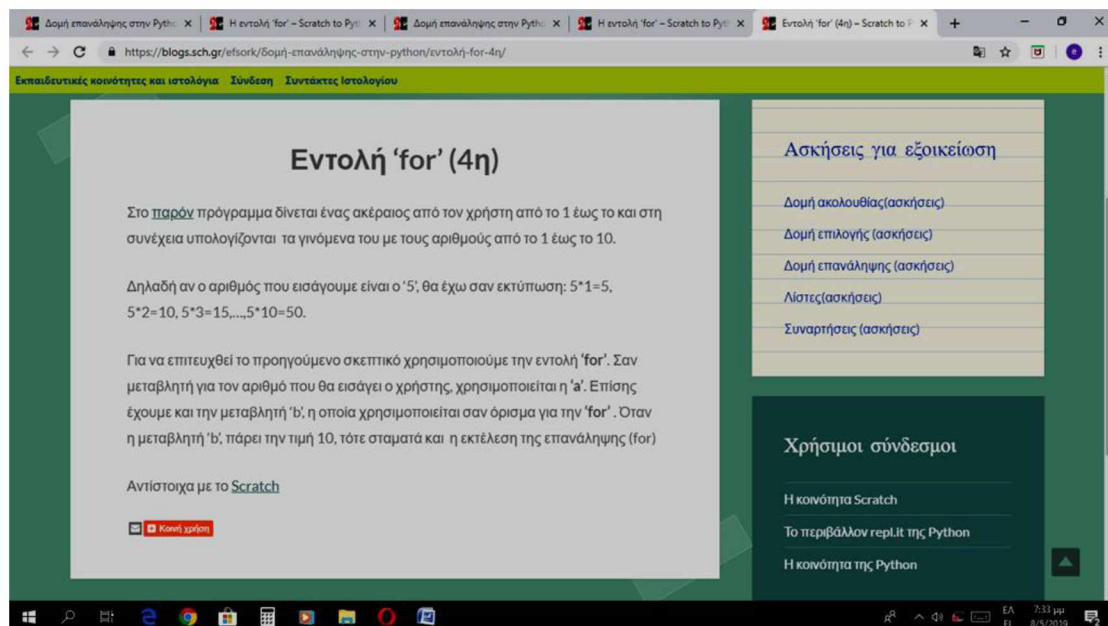
- Η εντολή ‘for’ στην Python
- Η εντολή ‘for’ με συνθήκη ‘if’ στην Python
- Η εντολή ‘while’ στην Python

Αρχικά παρουσιάζεται η απλή εντολή ‘for’ σε μια σελίδα του ιστολογίου και αναλύεται ο τρόπος που συνήθως χρησιμοποιείται για να υλοποιηθεί. Αυτό έχει ως συνέπεια την περιγραφή της συνάρτησης range() και την υλοποίηση τεσσάρων παραδειγμάτων με την εν λόγω εντολή, όπως απεικονίζεται στη συνέχεια.



Εικόνα 5.101: Η εντολή ‘for’

Στη συνέχεια θα παρουσιαστεί και θα περιγραφεί αναλυτικά το τέταρτο παράδειγμα της εντολής ‘for’ της Python (εικόνα 5.102).

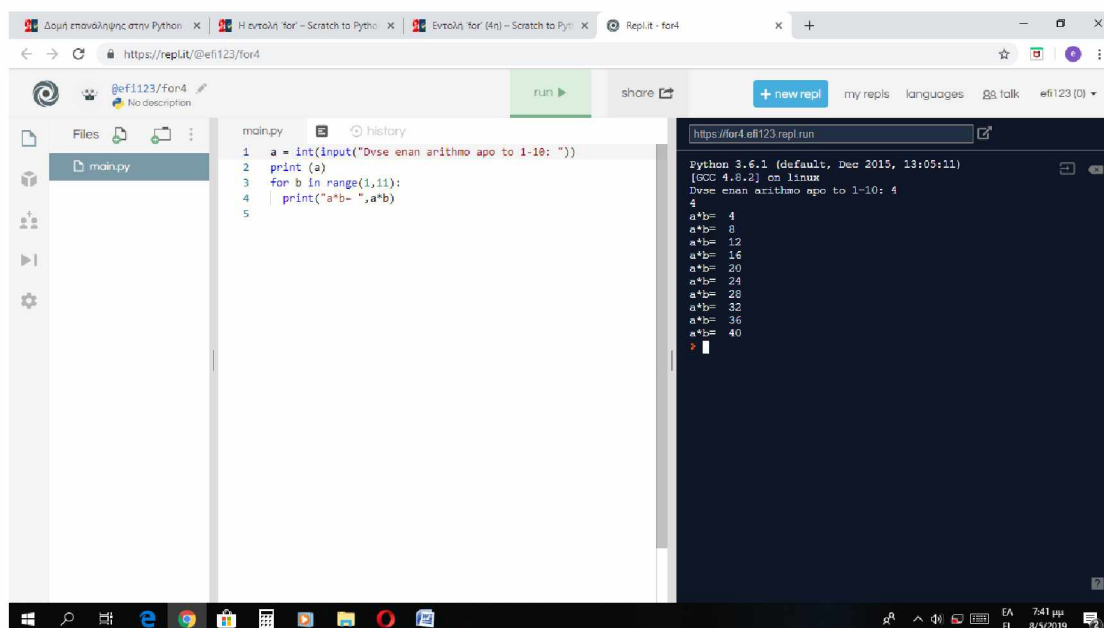


Εικόνα 5.102: Εντολή ‘for’ (4^η)

Στο συγκεκριμένο παράδειγμα, το πρόγραμμα ζητά έναν ακέραιο αριθμό από το χρήστη από το ‘1’ έως το ‘10’. Στη συνέχεια υπολογίζονται τα γινόμενα του αριθμού που δόθηκε με τους αριθμούς από το ‘1’ έως το ‘10’. Για να υλοποιηθεί η

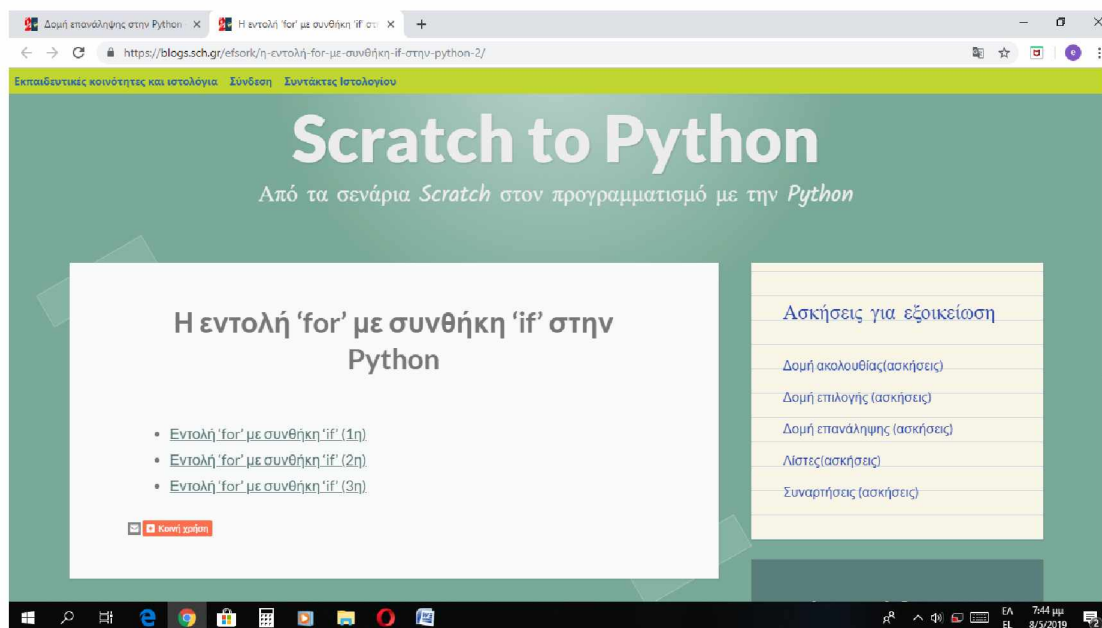
περιγραφόμενη λειτουργία χρησιμοποιήθηκε η εντολή της Python ‘for’. Στο πρόγραμμα του παραδείγματος έχουν δηλωθεί και δύο μεταβλητές, η μία χρησιμοποιείται για τον ακέραιο αριθμό που δίνεται από το χρήστη και η δεύτερη ως όρισμα για την εντολή επανάληψης ‘for’, μειώνοντάς τη κατά ένα σε κάθε επανάληψη. Η εκτέλεση της εντολής ‘for’ σταματά όταν η δεύτερη μεταβλητή πάρει τη τιμή μηδέν.

Στην εικόνα που ακολουθεί εμφανίζεται το πρόγραμμα σε Python, στο περιβάλλον replit και η εκτέλεσή του.



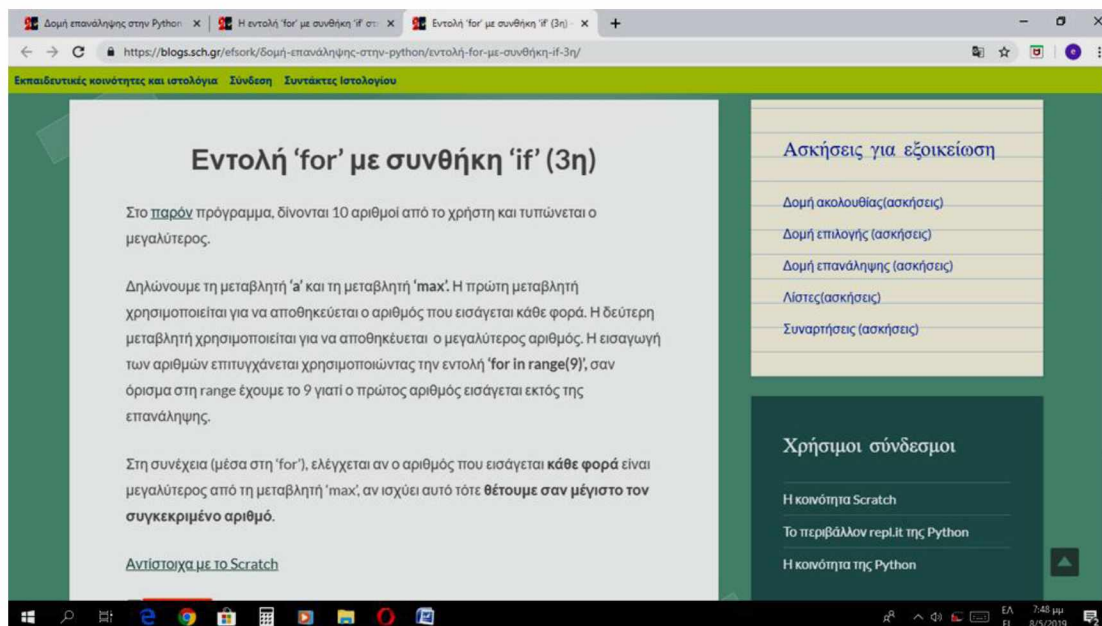
Εικόνα 5.103: Η εντολή ‘for’ –εκτέλεση του προγράμματος (4^η)

Η σελίδα του ιστολογίου που ακολουθεί είναι η εντολή ‘for’ με συνθήκη ‘if’, η οποία περιέχει τρία παραδείγματα υλοποιημένα με τις προαναφερόμενες εντολές, όπως στην εικόνα 5.104.



Εικόνα 5.104: Η εντολή 'for' με συνθήκη 'if' στην Python

Το παράδειγμα που έχει επιλεγεί για παρουσίαση είναι το τρίτο και η οθόνη που προκύπτει εισερχόμενος ο χρήστης σε αυτό, απεικονίζεται στη συνέχεια.



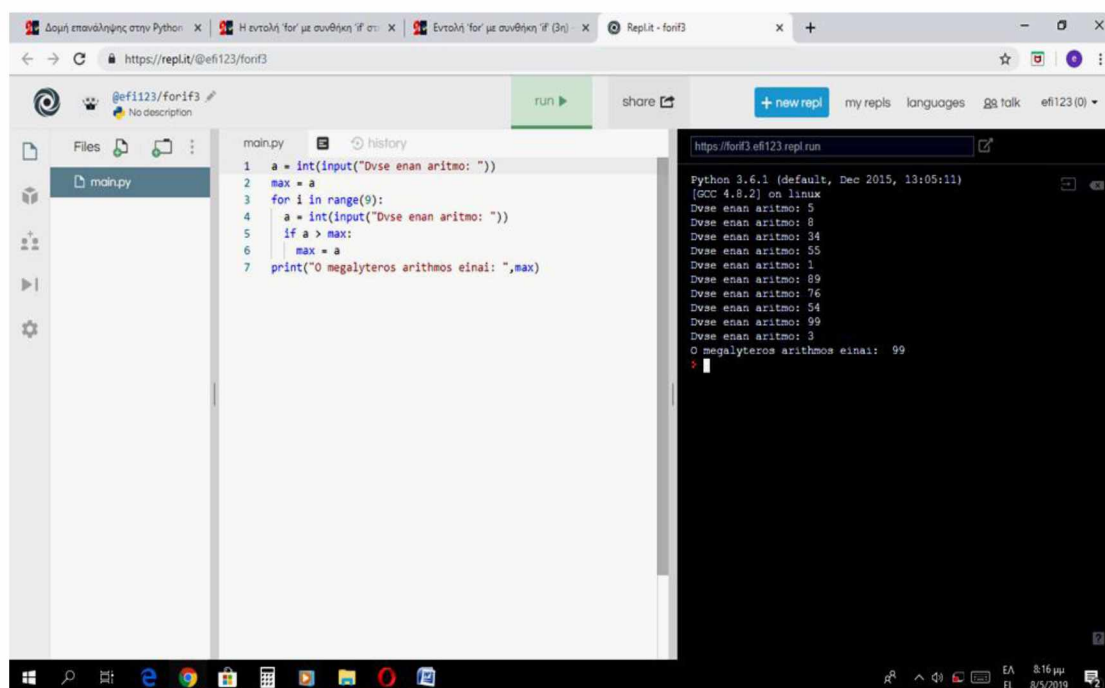
Εικόνα 5.105: Εντολή 'for' με συνθήκη 'if' στην Python

Στο συγκεκριμένο παράδειγμα το πρόγραμμα ζητά από το χρήστη να εισάγει δέκα αριθμούς. Στη συνέχεια το πρόγραμμα βρίσκει τον μεγαλύτερο αριθμό και τον τυπώνει στην οθόνη. Το πρόγραμμα χρησιμοποιεί δύο μεταβλητές την 'a' και την

‘max’. Η πρώτη μεταβλητή χρησιμοποιείται για να αποθηκεύεται ο αριθμός που εισάγει ο χρήστης κάθε φορά. Η δεύτερη μεταβλητή χρησιμοποιείται για να αποθηκεύεται ο μεγαλύτερος αριθμός.

Η διαδοχική εισαγωγή των αριθμών επιτυγχάνεται με την εντολή ‘for in range(9)’. Σαν όρισμα της συνάρτησης range() είναι ο αριθμός ‘9’. Μέσα στο βρόγχο της εντολής επανάληψης ‘for’ γίνεται έλεγχος εάν ο αριθμός που εισάγεται από το χρήστη κάθε φορά είναι μεγαλύτερος από τη μεταβλητή ‘max’ η οποία έχει αρχικοποιηθεί στην μεταβλητή ‘a’.

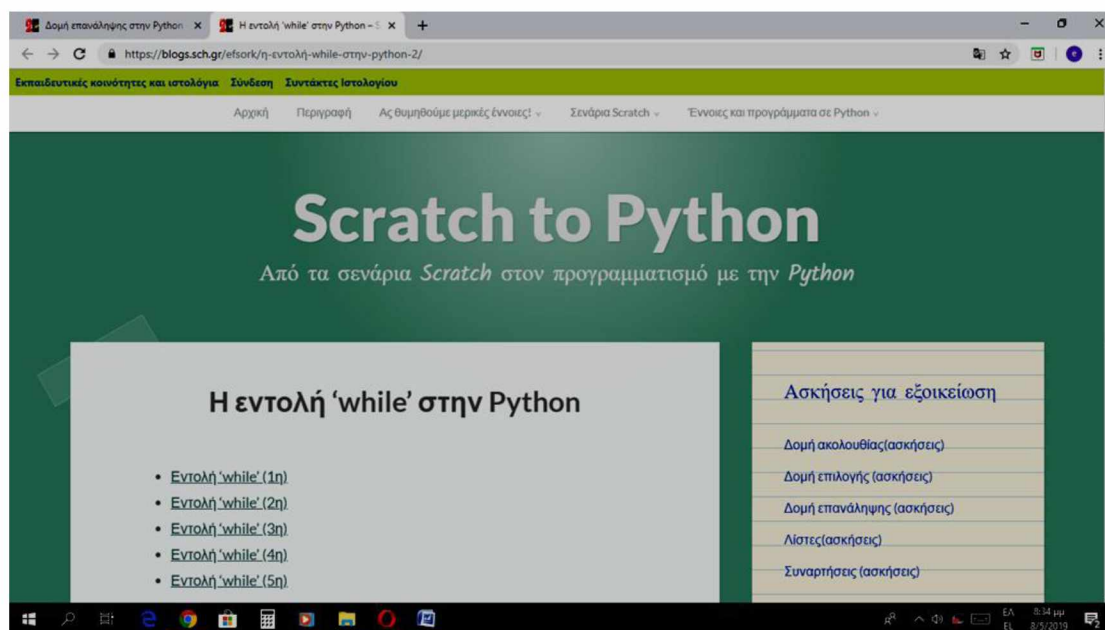
Σε περίπτωση που ισχύει η προηγούμενη συνθήκη το πρόγραμμα θέτει σαν μέγιστο τον αριθμό για τον οποίο ισχύει η συνθήκη. Το πρόγραμμα σε Python που υλοποιεί τα προαναφερόμενα ακολουθεί στην εικόνα 5.106.



```
main.py | history | run | share | + new repl | my repls | languages | talk | efi123 (0) |  
https://replit/@efi123/forif3  
@efi123/forif3 | No description |  
Files | main.py |  
1 a = int(input("Dvse enan aritmo: "))  
2 max = a  
3 for i in range(9):  
4     a = int(input("Dvse enan aritmo: "))  
5     if a > max:  
6         max = a  
7 print("O megalyteros arithmos einai: ",max)  
Python 3.6.1 (default, Dec 2015, 13:05:11)  
[GCC 4.8.2] on linux  
Dvse enan aritmo: 5  
Dvse enan aritmo: 8  
Dvse enan aritmo: 34  
Dvse enan aritmo: 55  
Dvse enan aritmo: 1  
Dvse enan aritmo: 89  
Dvse enan aritmo: 76  
Dvse enan aritmo: 54  
Dvse enan aritmo: 99  
Dvse enan aritmo: 3  
O megalyteros arithmos einai: 99  
> |
```

Εικόνα 5.106: Η εντολή ‘for’ με την ‘if’ (3^η)-εκτέλεση προγράμματος στο replit

Στη συνέχεια θα παρουσιαστεί η τρίτη και τελευταία εντολή η οποία χρησιμοποιείται για την υλοποίηση της επανάληψης στον προγραμματισμό και συγκεκριμένα στη γλώσσα προγραμματισμού Python. Αυτή είναι η εντολή ‘while’, της οποίας τα παραδείγματα προγραμμάτων περιέχονται στην σελίδα του ιστολογίου που απεικονίζεται στη συνέχεια.



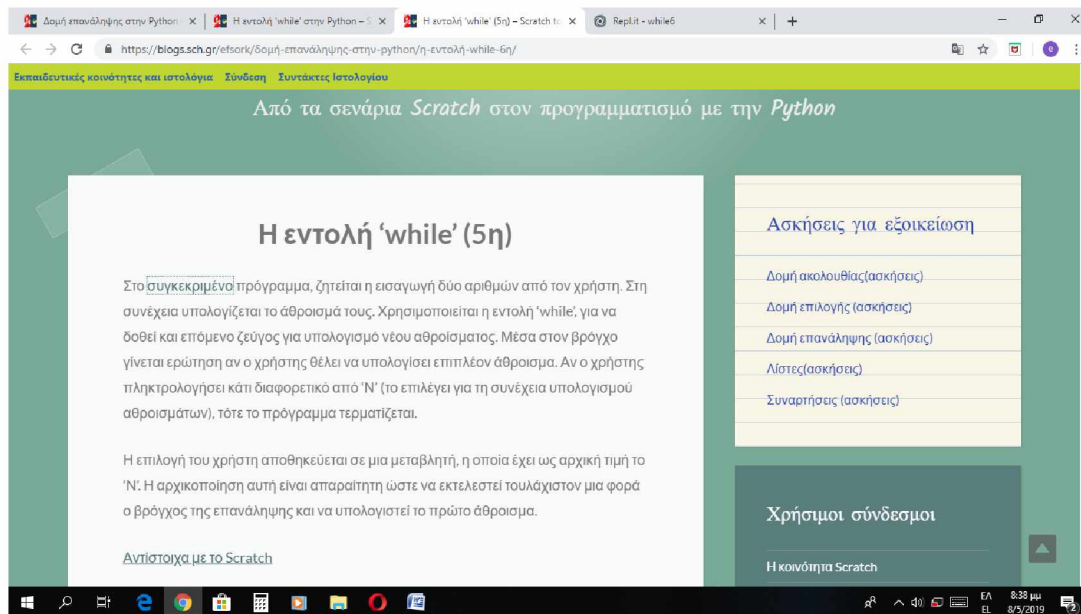
Εικόνα 5.107: Η εντολή ‘while’ στην Python

Όπως φαίνεται στην εικόνα 5.107, υπάρχουν πέντε παραδείγματα υλοποιημένα σε Python με την εντολή ‘while’. Στην παρούσα αναφορά επιλέχθηκε να αναλυθεί το πέμπτο και τελευταίο παράδειγμα της εντολής επανάληψης.

Στο συγκεκριμένο παράδειγμα το πρόγραμμα σε Python ζητά ως είσοδο ζεύγη αριθμών από το χρήστη ώστε να υπολογίσει το άθροισμά τους. Η διαδοχική εισαγωγή ζεύγους αριθμών πραγματοποιείται με την εντολή ‘while’. Στο εσωτερικό του βρόγχου της επανάληψης γίνεται ερώτηση στο χρήστη εάν επιθυμεί να εισάγει και νέο ζεύγος αριθμών. Η θετική απάντηση εκφράζεται με την πληκτρολόγηση του χαρακτήρα ‘N’ και αποθηκεύεται σε μία μεταβλητή, της οποίας η τιμή ελέγχεται για την συνέχεια της εισαγωγής αριθμών από το χρήστη.

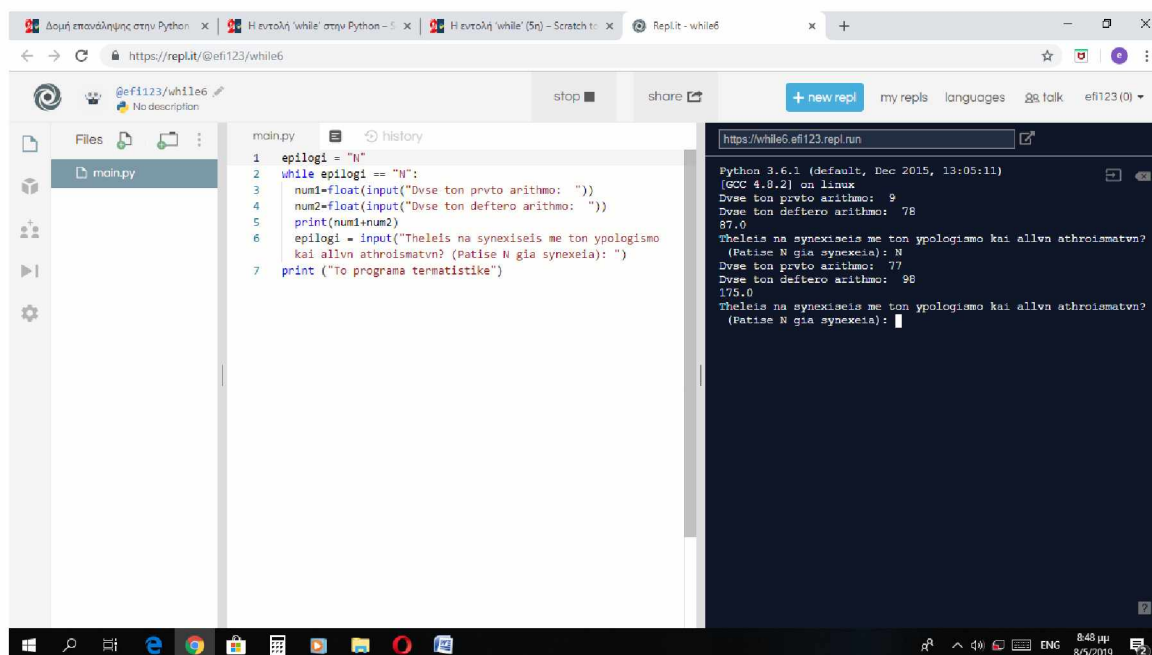
Σημαντικό είναι να σημειωθεί ότι η μεταβλητή η οποία αποθηκεύει την επιλογή του χρήστη αρχικοποιείται στην τιμή 'N', ώστε το πρόγραμμα να εκτελέσει τουλάχιστον μία φορά τον βρόγχο της επανάληψης.

Η οθόνη του παραδείγματος απεικονίζεται στη συνέχεια.



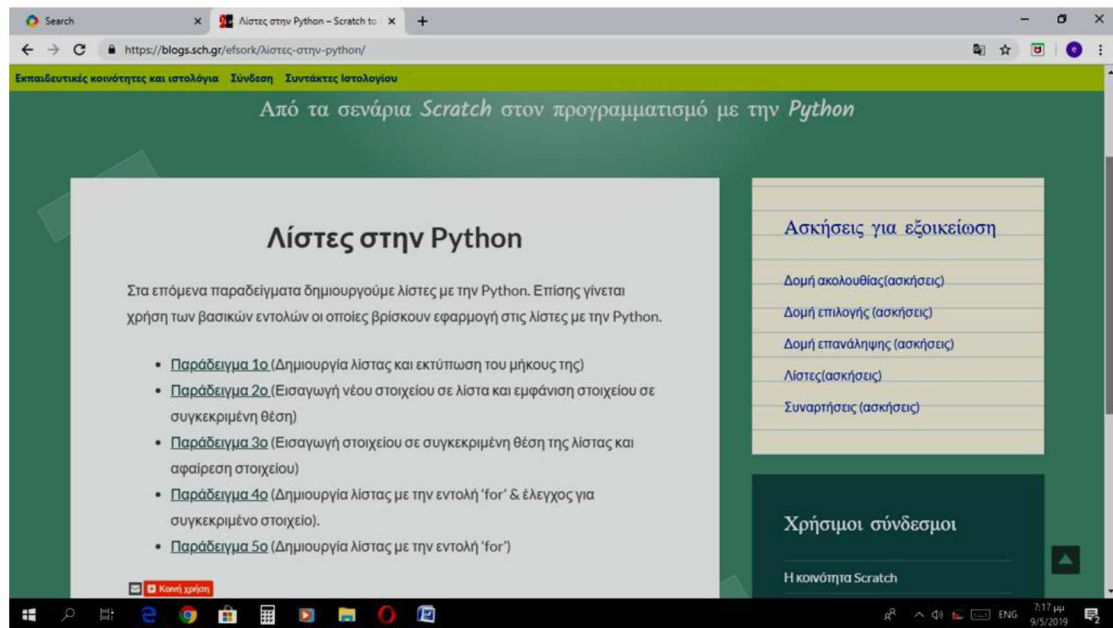
Εικόνα 5.108: Η εντολή 'while' (5^η)

Στην επόμενη εικόνα υπάρχει το πρόγραμμα στην Python και η εκτέλεσή του.



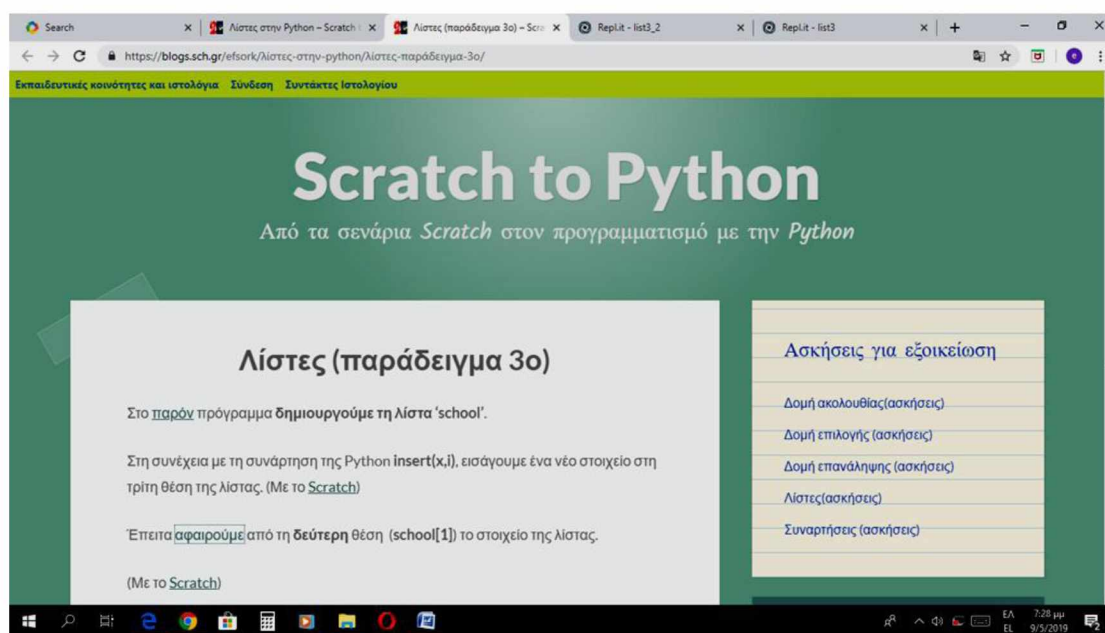
Εικόνα 5.109: Η εντολή 'while'-εκτέλεση προγράμματος (5^η)

Το επόμενο στοιχείο του μενού του ιστολογίου που θα αναλυθεί στην παρούσα αναφορά είναι οι ‘Λίστες στην Python’. Όπως φαίνεται στην εικόνα 5.110, η συγκεκριμένη σελίδα περιέχει πέντε παραδείγματα δημιουργίας λίστας με την γλώσσα προγραμματισμού Python.



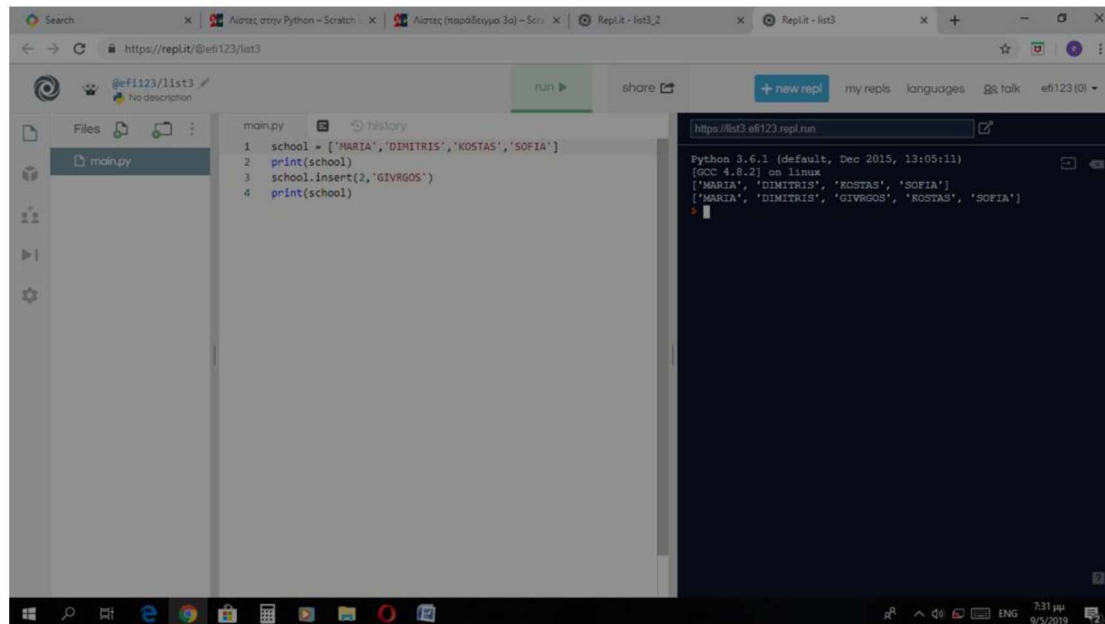
Εικόνα 5.110: Λίστες στην Python

Στη συνέχεια θα παρουσιαστεί και θα αναλυθεί το τρίτο παράδειγμα δημιουργίας λίστας. Στο συγκεκριμένο παράδειγμα ο χρήστης δημιουργεί τη λίστα με όνομα 'school' η οποία περιέχει τέσσερα ονόματα (στοιχεία). Έπειτα το πρόγραμμα προσθέτει ένα νέο στοιχείο στην τρίτη θέση της λίστας και με επιπλέον πρόγραμμα σε Python αφαιρείται το δεύτερο στοιχείο της λίστας. Στην επόμενη εικόνα εμφανίζεται η σελίδα του τρίτου παραδείγματος.



Εικόνα 5.111: Listas (παράδειγμα 3^ο)

Έπειτα στην εικόνα 5.112, εμφανίζεται το πρόγραμμα σε Python στο περιβάλλον replit, που εισάγει το νέο στοιχείο στη λίστα με τη συνάρτηση `insert()`.



Εικόνα 5.112: Listas, εισαγωγή στοιχείου –εκτέλεση προγράμματος (3^η)

Στην επόμενη εικόνα εμφανίζεται το πρόγραμμα σε Python που αφαιρεί στοιχείο της λίστας από συγκεκριμένη θέση με τη συνάρτηση `del()`.


```

main.py
1 school = ['MARIA', 'DIMITRIS', 'KOSTAS', 'SOFIA']
2 print(school)
3 school.insert(2, 'GIVRGOS')
4 print(school)
5 del school[1]
6 print(school)

Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
['MARIA', 'DIMITRIS', 'KOSTAS', 'SOFIA']
['MARIA', 'DIMITRIS', 'GIVRGOS', 'KOSTAS', 'SOFIA']
['MARIA', 'GIVRGOS', 'KOSTAS', 'SOFIA']

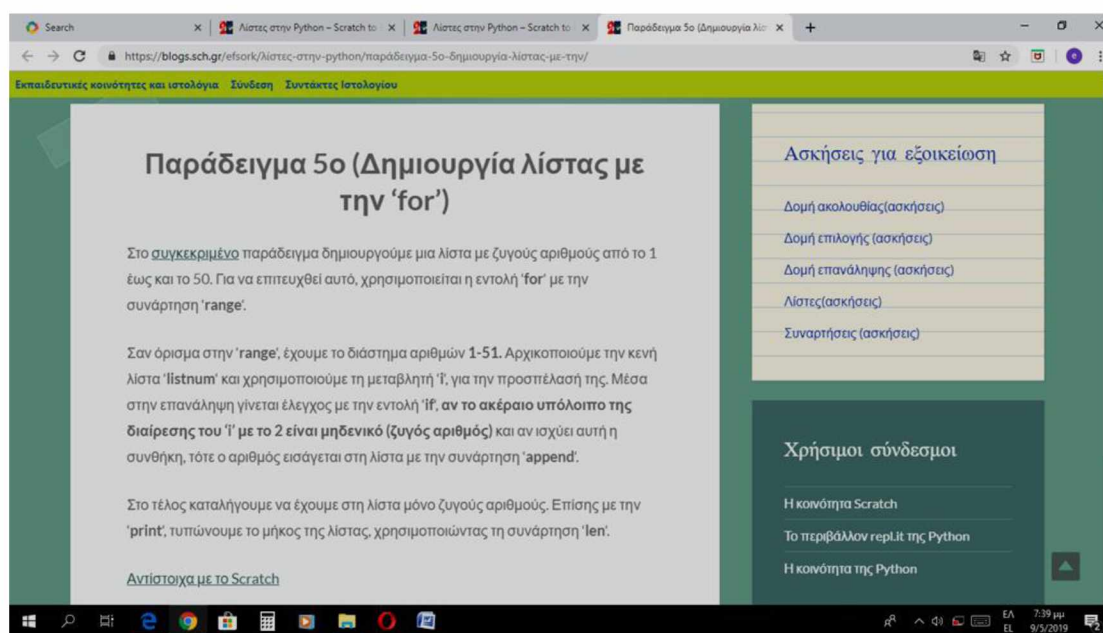
```

Εικόνα 5.113: Λίστες, διαγραφή στοιχείου-εκτέλεση προγράμματος (3^η)

Το επόμενο παράδειγμα που θα παρουσιαστεί στην κατηγορία 'Λίστες', είναι το πέμπτο. Σε αυτό το παράδειγμα δημιουργούμε μια λίστα με ζυγούς αριθμούς από το 1 έως και το 50. Για να επιτευχθεί αυτό, χρησιμοποιείται η εντολή 'for' με την συνάρτηση 'range'. Σαν όρισμα στην 'range', έχουμε το διάστημα αριθμών 1-51. Αρχικοποιούμε την κενή λίστα 'listnum' και χρησιμοποιούμε τη μεταβλητή 'i', για την προσπέλασή της λίστας.

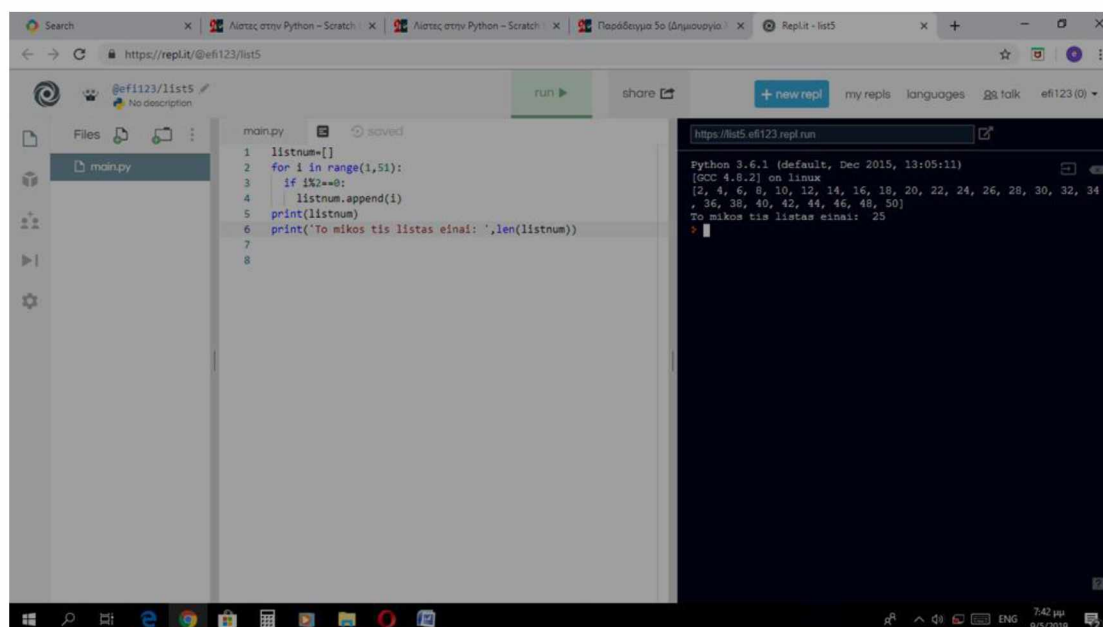
Μέσα στην επανάληψη γίνεται έλεγχος με την εντολή 'if', αν το ακέραιο υπόλοιπο της διαίρεσης του 'i' με το 2 είναι μηδενικό (=ζυγός αριθμός) και αν ισχύει αυτή η συνθήκη, τότε ο αριθμός εισάγεται στη λίστα με την συνάρτηση 'append'.

Στο τέλος καταλήγουμε να έχουμε στη λίστα μόνο ζυγούς αριθμούς. Επίσης με την 'print', τυπώνουμε το μήκος της λίστας, χρησιμοποιώντας τη συνάρτηση 'len'. Η σελίδα του παραδείγματος απεικονίζεται στην επόμενη εικόνα.



Εικόνα 5.114: Παράδειγμα 5^ο (δημιουργία λίστας με την 'for')

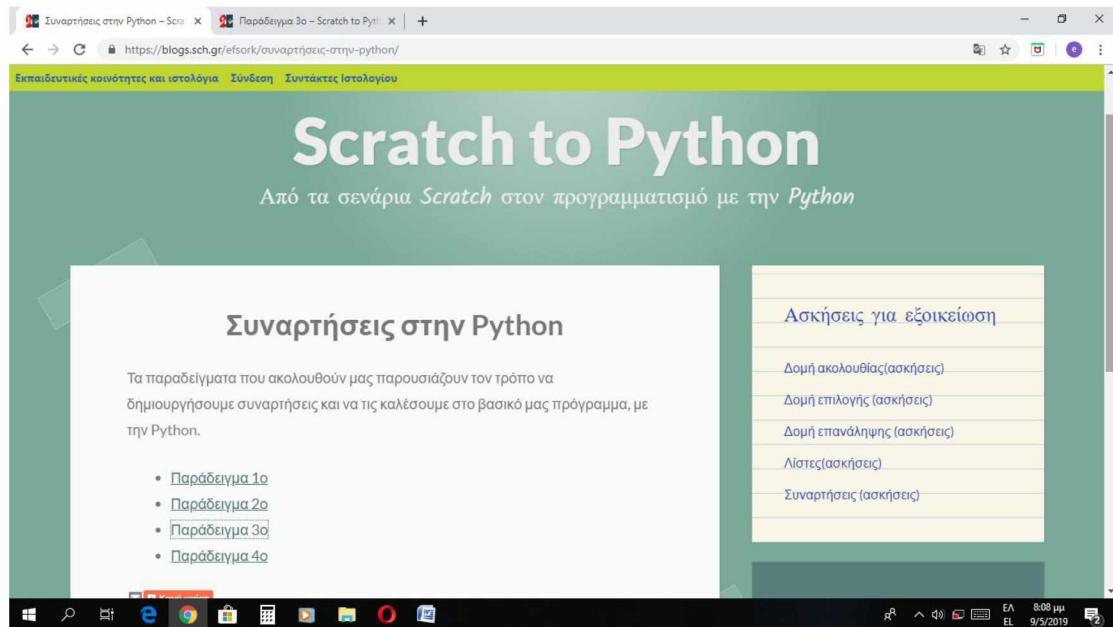
Επίσης στη συνέχεια παρουσιάζεται το πρόγραμμα σε Python καθώς και η εκτέλεσή του στο περιβάλλον repl.it.



Εικόνα 5.115: Παράδειγμα 5^ο (δημιουργία λίστας με την 'for')-εκτέλεση προγράμματος

Το επόμενο στοιχείο του μενού 'Έννοιες και προγράμματα σε Python', είναι οι 'Συνάρτησεις στην Python'. Η σελίδα που παρουσιάζει το προαναφερόμενο στοιχείο

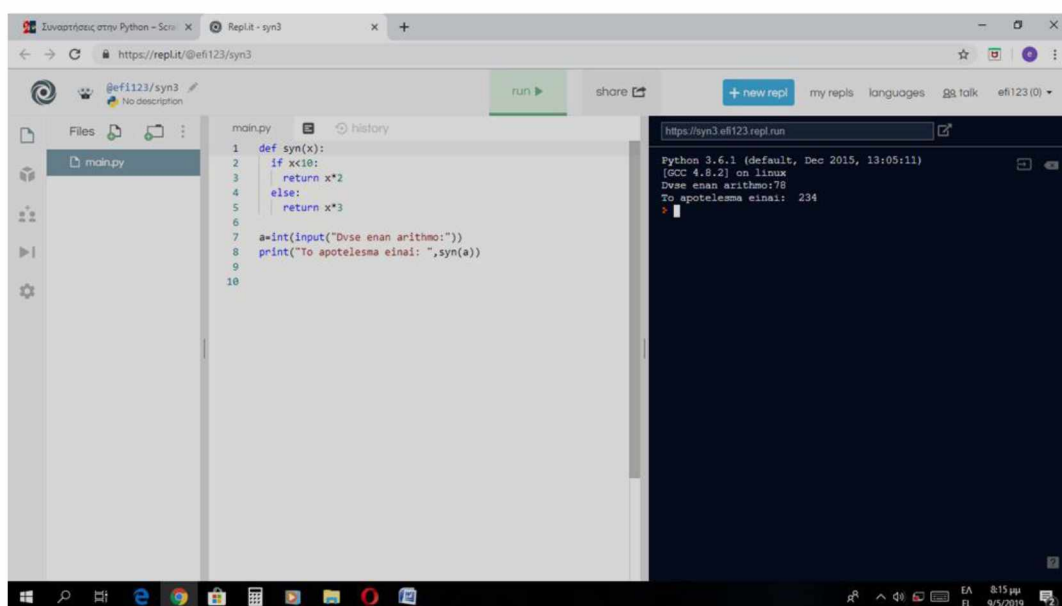
απεικονίζεται στη συνέχεια και περιέχει τέσσερα παραδείγματα δημιουργίας συναρτήσεων με την Python.



Εικόνα 5.116: Συναρτήσεις στην Python

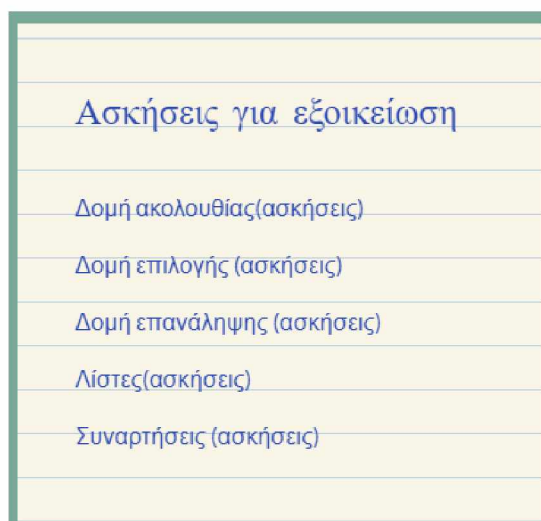
Αρχικά θα αναλυθεί το τρίτο παράδειγμα της δημιουργίας συναρτήσεων. Στο εν λόγω παράδειγμα, δημιουργούμε μια συνάρτηση η οποία έχει σαν όρισμα τη μεταβλητή 'x'. Η συνάρτηση ελέγχει αν η μεταβλητή, η οποία είναι ακέραιος αριθμός και δίνεται από το χρήστη, είναι μικρότερη από το 10. Αν ισχύει αυτό τότε επιστρέφει το διπλάσιο του 'x', διαφορετικά το τριπλάσιό του. Στο κυριώς πρόγραμμα, δίνεται από το χρήστη ένας ακέραιος και γίνεται η κλήση της συνάρτησης.

Η οθόνη που προκύπτει σε περίπτωση που ο χρήστης εισέλθει στο πρόγραμμα της Python του παραδείγματος είναι στην εικόνα 5.117.



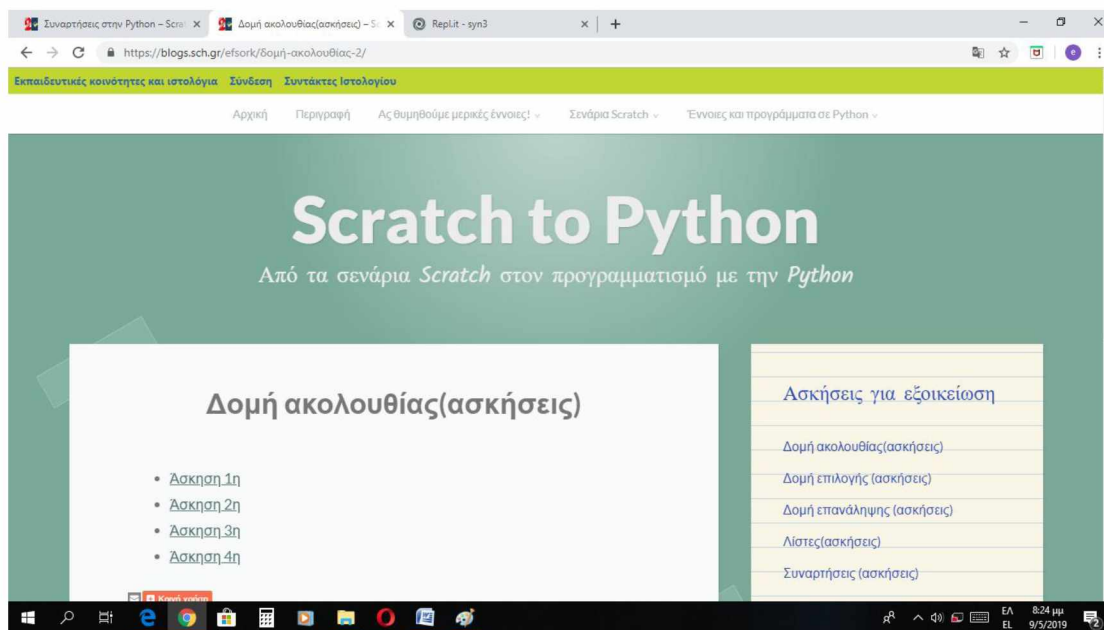
Εικόνα 5.117: Συναρτήσεις –εκτέλεση προγράμματος (3^η)

Στη συνέχεια της αναφοράς θα γίνει μια παρουσίαση του πλευρικού μενού του ιστολογίου, το οποίο αφορά στις ασκήσεις που προτείνονται στον εκπαιδευόμενο για εμπέδωση των βασικών εννοιών που παρουσιάστηκαν μέσω παραδειγμάτων. Σημαντικό είναι να αναφερθεί το γεγονός ότι σε όλες τις ασκήσεις υπάρχει υπερσύνδεσμος που οδηγεί τον εκπαιδευόμενο σε προτεινόμενη λύση του προβλήματος που καλείται να επιλύσει με τη χρήση της γλώσσας προγραμματισμού Python. Στην εικόνα που ακολουθεί υπάρχει το πλευρικό μενού με τις βασικές προς εκμάθηση ενότητες.



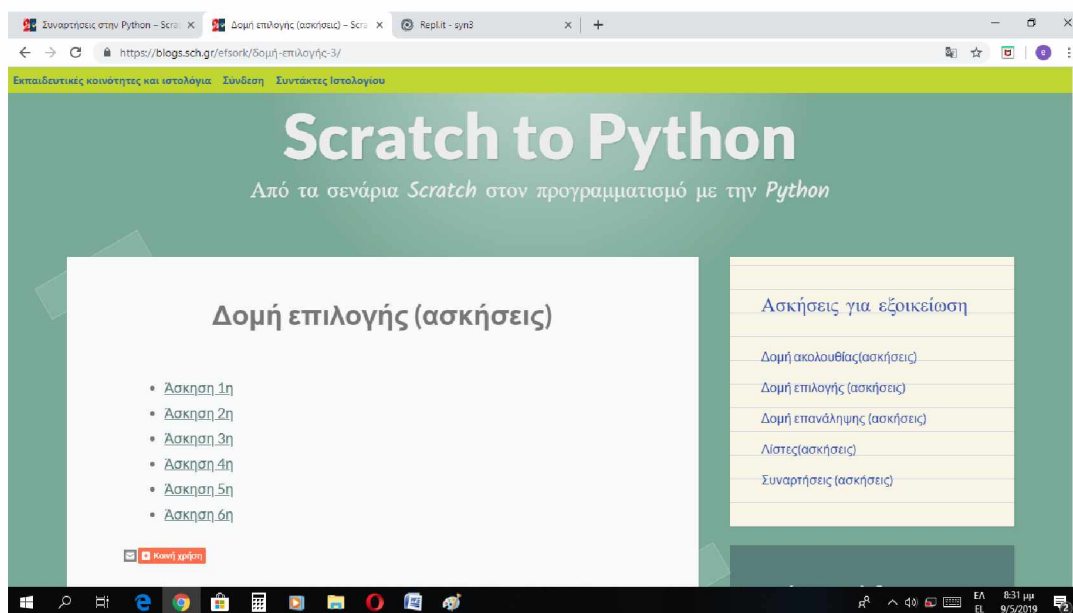
Εικόνα 5.118: Ασκήσεις για εξοικείωση

Όταν ο χρήστης επιλέξει τις ασκήσεις που αφορούν στη Δομή ακολουθίας, η οθόνη που προκύπτει είναι στην εικόνα 5.119.



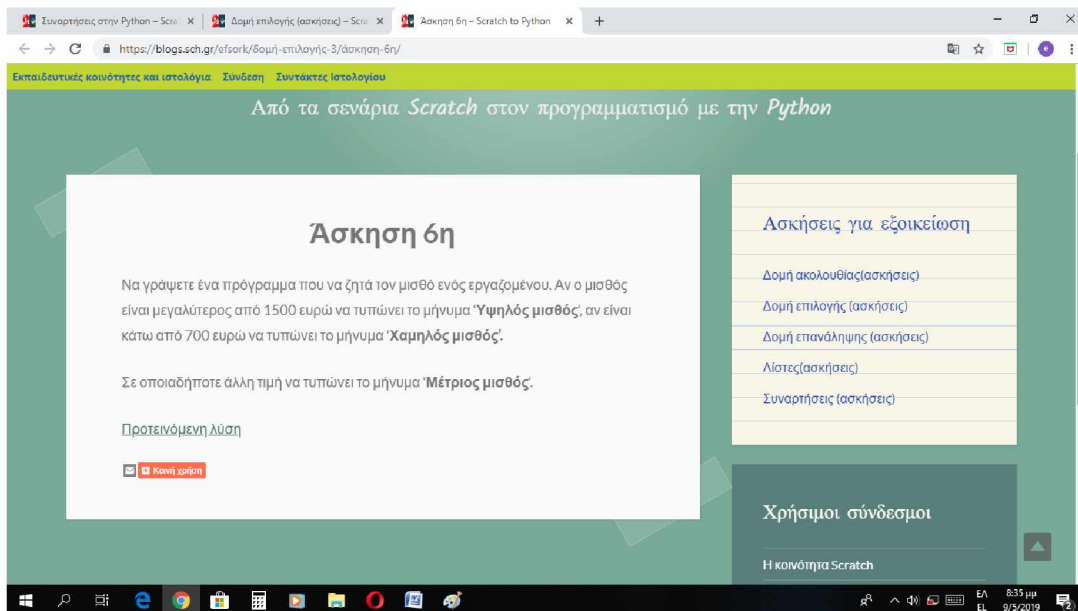
Εικόνα 5.119: Δομή ακολουθίας(ασκήσεις)

Οι προτεινόμενες ασκήσεις για τη Δομή ακολουθίας που υπάρχουν στη σελίδα είναι τέσσερις. Στην επόμενη εικόνα εμφανίζεται η σελίδα που περιέχει τις ασκήσεις της Δομής επιλογής.



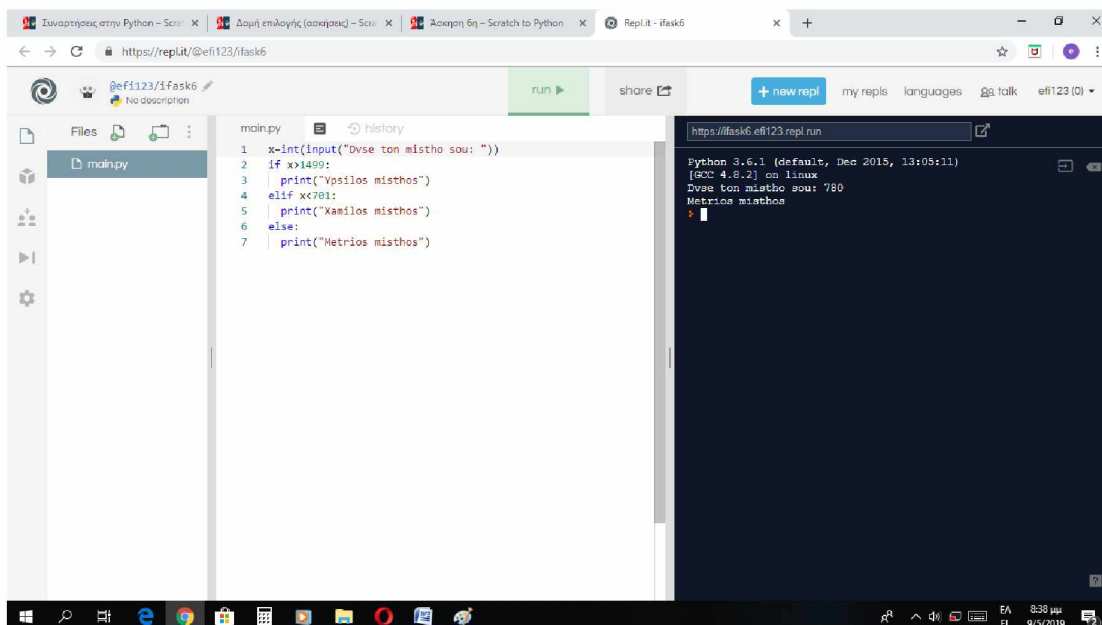
Εικόνα 5.120: Δομή επιλογής (ασκήσεις)

Στην περίπτωση που ο χρήστης επιλέξει την έκτη άσκηση η οθόνη που προκύπτει ακολουθεί στη συνέχεια.



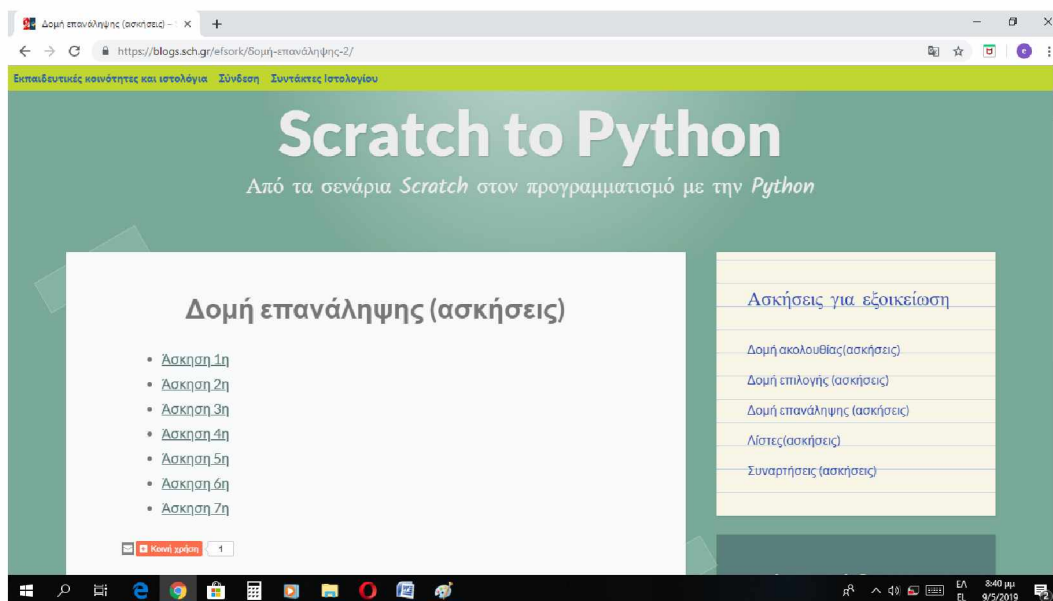
Εικόνα 5.121: Άσκηση 6^η (Δομή επιλογής)

Επιλέγοντας ο χρήστης τον υπερσύνδεσμο για την προτεινόμενη λύση μεταφέρεται στο περιβάλλον replit και στο πρόγραμμα της Python, όπως στην εικόνα που ακολουθεί.



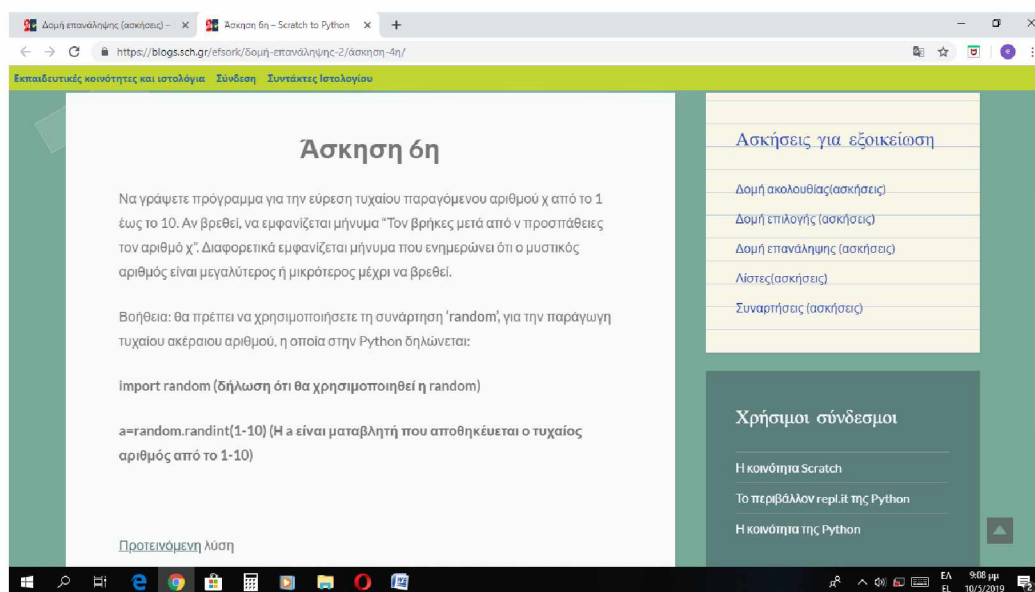
Εικόνα 5.122: Άσκηση 6^η, δομή επιλογής-προτεινόμενη λύση

Συνεχίζοντας την περιήγηση ο χρήστης στο ιστολόγιο και στις ασκήσεις για εξοικείωση βρίσκεται στις ασκήσεις που αφορούν την Δομή επανάληψης και η οθόνη που προκύπτει εμφανίζεται στη συνέχεια. Η σελίδα των ασκήσεων που προτείνονται, περιέχει έξι ασκήσεις.



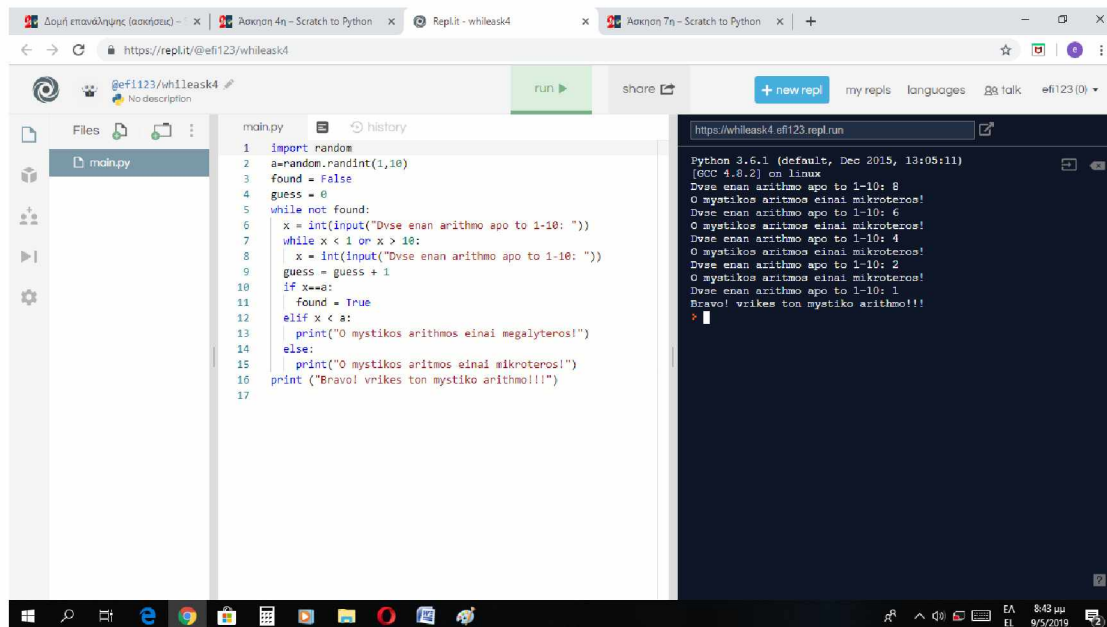
Εικόνα 5.123: Δομή επανάληψης (ασκήσεις)

Η άσκηση που επιλέγεται να παρουσιαστεί είναι η έκτη, της οποίας η οθόνη υπάρχει στην εικόνα 5.124. Στην άσκηση αυτή δίνεται και μία επιπλέον βοήθεια για την επίλυσή της για την συνάρτηση random().



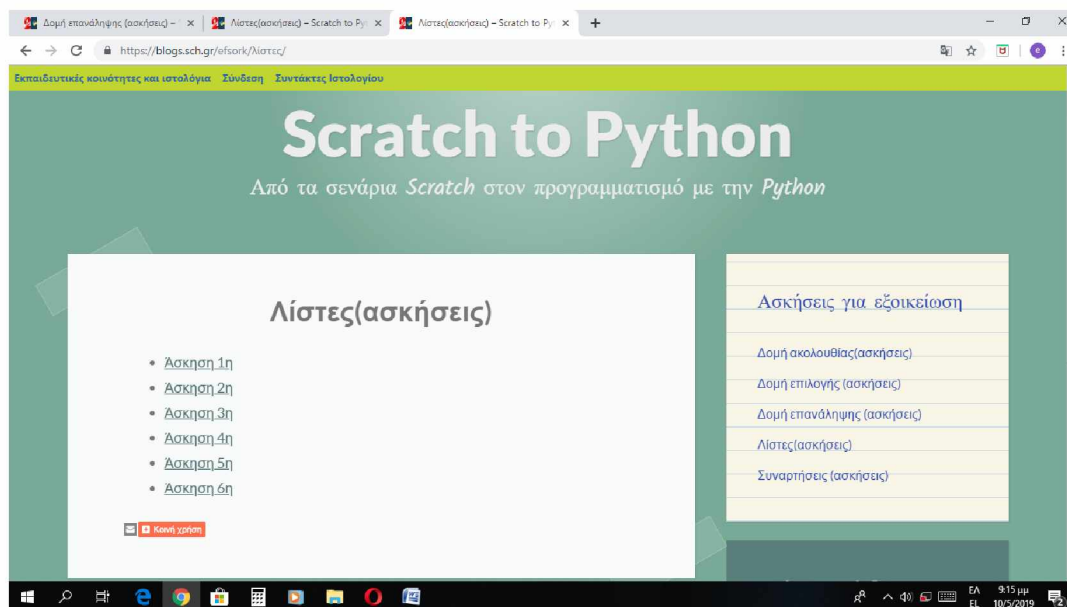
Εικόνα 5.124: Άσκηση 6^η (Δομή επανάληψης)

Η προτεινόμενη λύση της άσκησης και η εκτέλεση του προγράμματος στο περιβάλλον replit, απεικονίζεται στη συνέχεια.



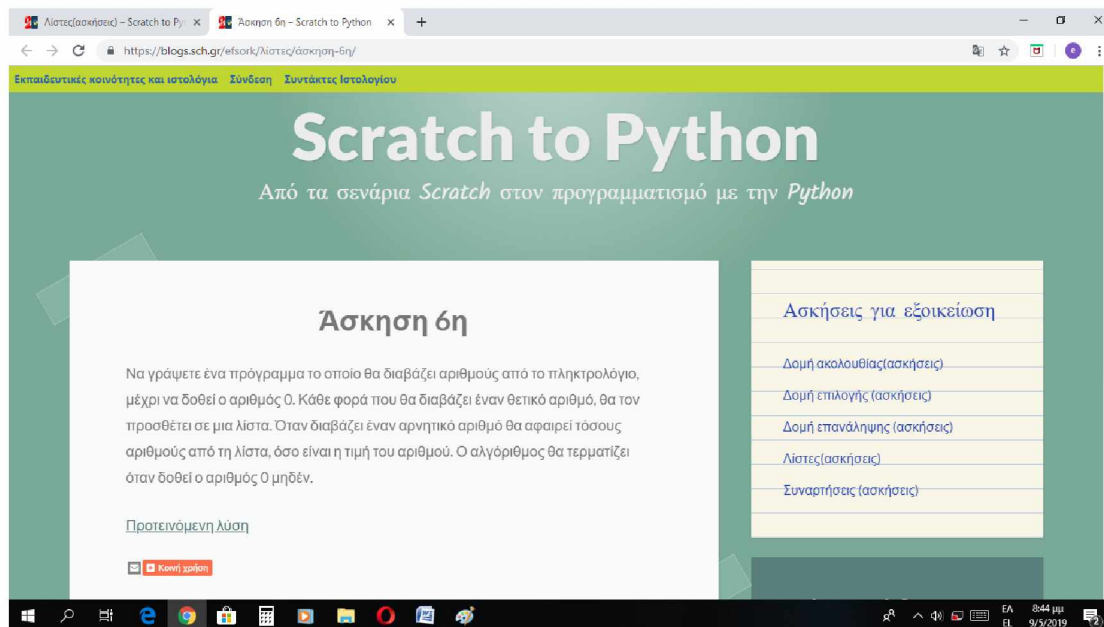
Εικόνα 5.125: Άσκηση 6^η-εκτέλεση προγράμματος (Δομή επανάληψης)

Στη συνέχεια σε περίπτωση που ο χρήστης επιλέξει τον υπερσύνδεσμο για την εξοικείωση με τη δημιουργία λίστας με την Python, η οθόνη που προκύπτει εμφανίζεται στην εικόνα 5.126.



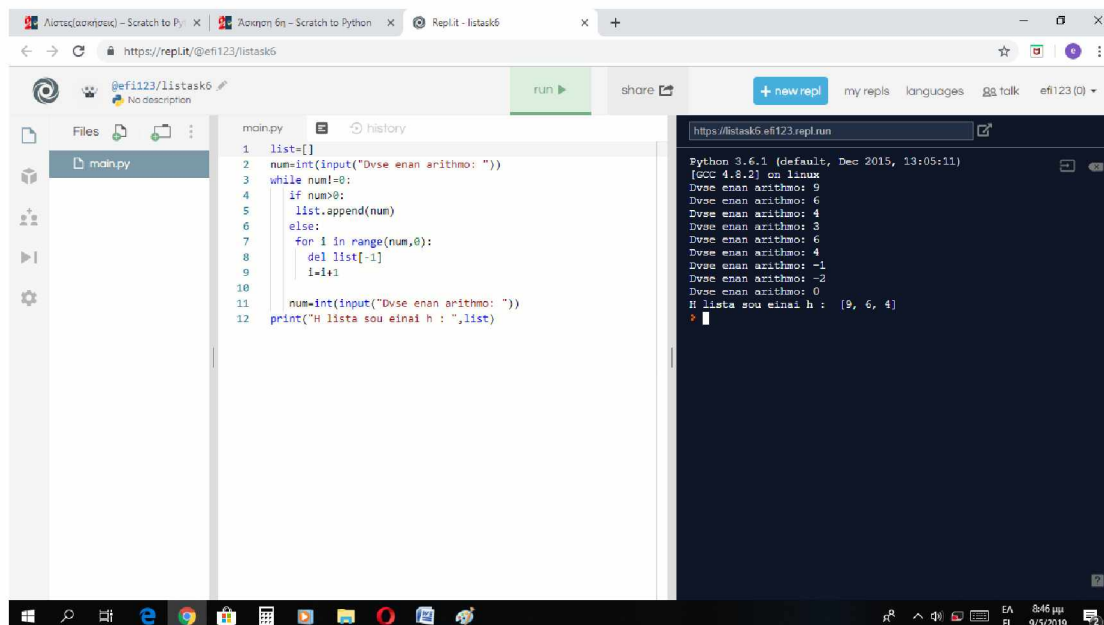
Εικόνα 5.126: Λίστες (ασκήσεις)

Στην παρούσα αναφορά επιλέγεται να παρουσιαστεί η έκτη άσκηση στην εικόνα 5.127.



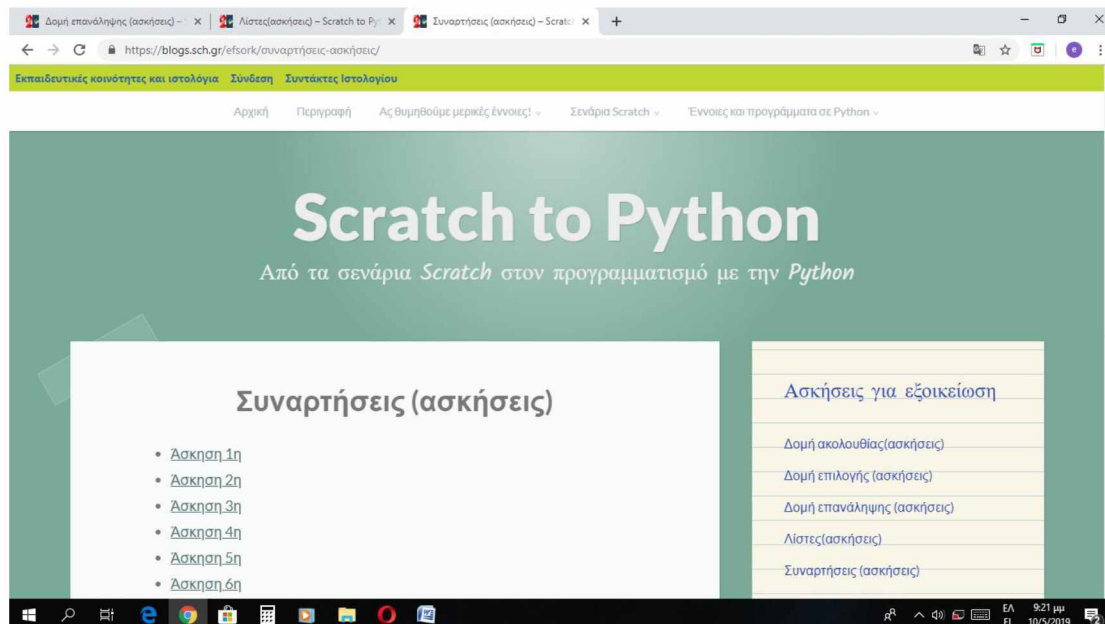
Εικόνα 5.127: Άσκηση 6^η (Λίστες)

Το πρόγραμμα που προτείνεται ως λύση της άσκηση καθώς και η εκτέλεσή του στο replit, είναι στην εικόνα 5.128.



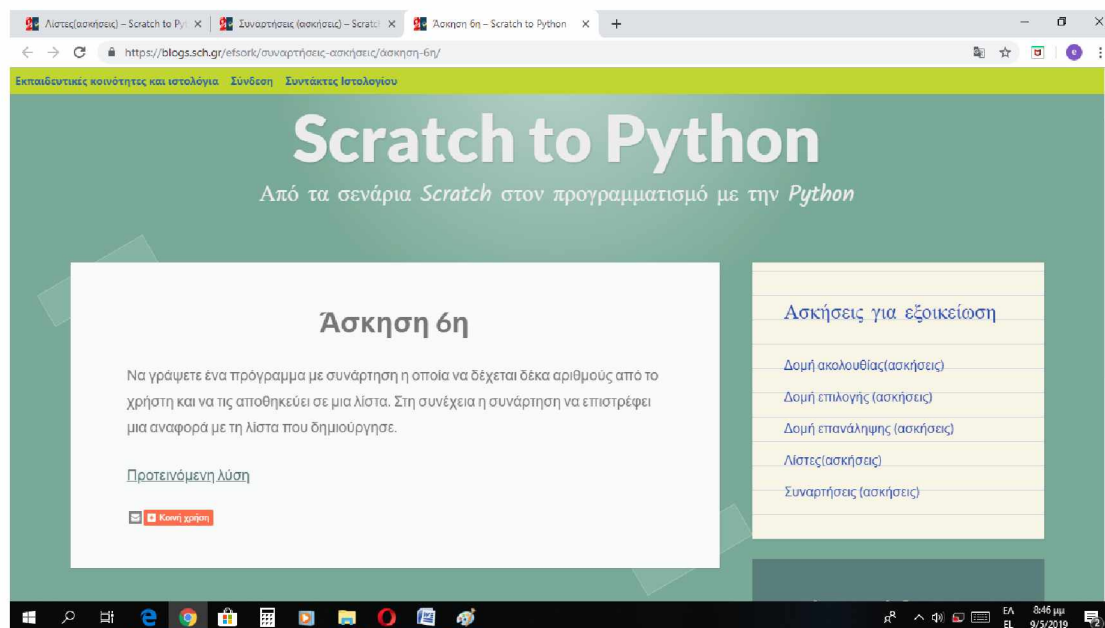
Εικόνα 5.128: Άσκηση 6^η – εκτέλεση προγράμματος (Λίστες)

Στο τέλος του μενού των ασκήσεων υπάρχουν ασκήσεις που αφορούν στην δημιουργία συναρτήσεων με την γλώσσα προγραμματισμού Python, εικόνα 5.129.



Εικόνα 5.129: Συναρτήσεις (ασκήσεις)

Η εικόνα που ακολουθεί είναι η οθόνη που προκύπτει όταν ο χρήστης επιλέξει τον υπερσύνδεσμο της έκτης άσκησης.



Εικόνα 5.130: Άσκηση 6^η (Συναρτήσεις)

Όταν ο χρήστης επιλέξει τον υπερσύνδεσμο της προτεινόμενης λύσης της άσκησης, προκύπτει η επόμενη εικόνα.

```

main.py
1 def getvalues():
2     values=[]
3     count=0
4     while count<10:
5         num=int(input("Dose enan arithmo: "))
6         values.append(num)
7         count=count+1
8     return values
9 #call getvalues
10 numbers=getvalues()
11 print("oi arithmoi tis listas einai: ")
12 print(numbers)
    
```

```

Python 3.6.1 (default, Dec 2015, 13:05:11)
[ESC 4.0.2] on Linux
Dose enan arithmo: 6
Dose enan arithmo: 5
Dose enan arithmo: 3
Dose enan arithmo: 56
Dose enan arithmo: 77
Dose enan arithmo: 34
Dose enan arithmo: 12
Dose enan arithmo: 88
Dose enan arithmo: 56
Dose enan arithmo: 67
Oi arithmoi tis listas einai:
[5, 5, 3, 56, 77, 34, 12, 88, 56, 67]
    
```

Εικόνα 5.131: Άσκηση 6^η –εκτέλεση προγράμματος (συναρτήσεις)

6. Αξιολόγηση

6.1 Στόχος της έρευνας

Στόχος της παρούσας έρευνας είναι η αξιολόγηση του εκπαιδευτικού υλικού που δημιουργήθηκε, το οποίο βασίζεται στη μάθηση μέσω παραδειγμάτων πάνω στις βασικές προγραμματιστικές έννοιες και δομές της γλώσσας προγραμματισμού Python. Δηλαδή η αξιολόγηση του εκπαιδευτικού ιστολογίου blogs.sch.gr/efsork, το οποίο κατασκευάστηκε στα πλαίσια της παρούσας διπλωματικής εργασίας.

6.2 Μεταβλητές προς έρευνα

Οι προς μελέτη μεταβλητές ορίζονται σύμφωνα με τον [42] και περιγράφονται στη συνέχεια.

Κατά τη δημιουργία του ερωτηματολογίου από τον [42], πρωτίστης σημασίας ήταν να καθοριστεί ο τύπος των ερωτήσεων που θα μπορούσε αυτό να περιέχει. Για να καταστεί αυτό εφικτό χρησιμοποιήθηκαν προϋπάρχουσες μελέτες, βιβλιογραφία και η τεχνική του καταγισμού ιδεών. Η βασική ιδέα για την επιλογή των ερωτήσεων περιείχε την απλότητα και την γενικότητα που τις χαρακτήριζαν.

Γενικά, οι ερωτήσεις σε κάθε κατηγορία του ερωτηματολογίου ήταν περίπου ισοβαρείς.

Στη συνέχεια κατά την ανάλυση των παραγόντων σύμφωνα με τον [42], κατέληξε ότι οι χρήστες αξιολόγησαν τα διάφορα προϊόντα κάνοντας χρήση τριών διαστάσεων, την **χρησιμότητα**, την **ικανοποίηση** και την **ευχρηστία**.

Υπολογίζοντας επιμέρους συσχετίσεις χρησιμοποιώντας τις προηγούμενες διαστάσεις, βρέθηκε ότι η ευχρηστία και η χρησιμότητα αλληλοεπηρεάζονται, ώστε βελτιώνοντας την ευχρηστία να βελτιώνονται και τα αποτελέσματα της χρησιμότητας.

Η σημασία της χρησιμότητας μειώνεται όταν τα συστήματα είναι εσωτερικά και οι χρήστες είναι υποχρεωμένοι να τα χρησιμοποιήσουν. Αρκετά μεγάλες διακυμάνσεις στη βαθμολογία της χρησιμότητας όταν οι χρήστες έχουν μικρής διάρκειας έκθεση σε ένα αξιολογούμενο προϊόν.

Σύμφωνα με τον [42], η ικανοποίηση έχει ισχυρή σχέση με τη χρήση και τα ερωτήματα της κατηγορίας ευχρηστία μπορούν να διαχωριστούν σε δύο κατηγορίες, την **ευκολία στη μάθηση** και την **ευχρηστία** (με υψηλή συσχέτιση μεταξύ τους).

Πιο συγκεκριμένα, η μελέτη κινήθηκε στους εξής άξονες:

- Χρησιμότητα
- Ευχρηστία
- Ευκολία στη μάθηση
- Ικανοποίηση

6.3 Περιγραφή δείγματος

Το δείγμα της έρευνας αποτέλεσαν 25 φοιτητές του τμήματος Πληροφορικής με Εφαρμογές στην Βιοϊατρική του Πανεπιστημίου Θεσσαλίας.

6.4 Μέσο συλλογής δεδομένων

Αρχικά ζητήθηκε από τους φοιτητές του τμήματος Πληροφορικής με εφαρμογές στην Βιοϊατρική του Πανεπιστημίου Θεσσαλίας, να περιηγηθούν στο ιστολόγιο blogs.sch.gr/efsork, το οποίο δημιουργήθηκε σαν πρόταση μετάβασης από τη γλώσσα προγραμματισμού Scratch στη γλώσσα προγραμματισμού Python.

Στη συνέχεια διανεμήθηκε ένα ηλεκτρονικό ερωτηματολόγιο το οποίο δημιουργήθηκε με GoogleForms, ώστε οι φοιτητές να αξιολογήσουν τον προτεινόμενο τρόπο μετάβασης από τη μία γλώσσα προγραμματισμού σε μία επόμενη υψηλότερου επιπέδου.

Χρησιμοποιήθηκε ως βάση το ερωτηματολόγιο USE (Usefulness –Satisfaction – Ease of Use) του [42], το οποίο δίνεται στο παράρτημα. Από το ερωτηματολόγιο χρησιμοποιήθηκαν οι 23 ερωτήσεις του διότι οι υπόλοιπες 7 που περιέχει δεν έχουν εφαρμογή στην παρούσα έρευνα.

Οι φοιτητές έπρεπε για κάθε μία ερώτηση να κυκλώσουν έναν αριθμό από το 1 έως το 7, όπου το 1 σημαίνει ότι ο φοιτητής διαφωνεί πλήρως με αυτό που παρουσιάζεται στην ερώτηση, το 4 σημαίνει ότι ο φοιτητής έχει μια ουδέτερη στάση σε σχέση με την

ερώτηση (ούτε συμφωνεί ούτε διαφωνεί) και το 7 σημαίνει ότι ο φοιτητής συμφωνεί εντελώς.

Έχοντας ως σκοπό τη στατιστική επεξεργασία των ερωτηματολογίων, οι ερωτήσεις ομαδοποιήθηκαν σε τέσσερις κατηγορίες σύμφωνα με τον [42].

- 1^η Κατηγορία: Χρησιμότητα (Usefulness)
- 2^η Κατηγορία: Ευχρηστία (Ease of Use)
- 3^η Κατηγορία: Ευκολία στη Μάθηση (Ease of Learning)
- 4^η Κατηγορία: Ικανοποίηση (Satisfaction)

Στον επόμενο πίνακα παρουσιάζονται οι ερωτήσεις που περιέχονται στο ερωτηματολόγιο, ομαδοποιημένες στις προαναφερόμενες τέσσερις κατηγορίες.

Πίνακας 5: Ομαδοποίηση ερωτήσεων στις 4 κατηγορίες

Δηλώσεις	
Χρησιμότητα (usefulness)	<ol style="list-style-type: none"> 1. Είναι μια χρήσιμη μέθοδος (ερ. 4) 2. Κάνει ότι θα περίμενα από μια τέτοια μέθοδο (ερ. 7) 3. Θα με βοηθήσει να πετύχω ευκολότερα το στόχο μου (ερ. 12) 4. Ταιριάζει με τις ανάγκες μου (ερ.16) 5. Εξοικονομώ χρόνο με τη χρήση της μεθόδου (ερ. 18)
Ευκολία χρήσης (ease of use)	<ol style="list-style-type: none"> 6. Είναι απλή μέθοδος στη χρήση (ερ.2) 7. Δεν παρατήρησα ασυνέπειες-προβλήματα κατά τη χρήση της μεθόδου (ερ. 5) 8. Απαιτεί τα λιγότερα δυνατά βήματα για να πετύχω το στόχο μου (ερ. 8) 9. Το ιστολόγιο είναι φιλικό στη χρήση του (ερ. 9) 10. Μπορώ με τη μέθοδο αυτή να

Ευκολία μάθησης (ease of learning)

διορθώσω ευκολότερα και γρηγορότερα πιθανά λάθη (ερ. 13)

11. Η χρήση της μεθόδου δεν απαιτεί ιδιαίτερη προσπάθεια από εμένα (ερ. 14)

12. Είναι μια μέθοδος που μπορώ να χρησιμοποιήσω χωρίς γραπτές οδηγίες (ερ. 15)

13. Είναι εύκολη μέθοδος στη χρήση της (ερ. 21)

14. Έμαθα να χρησιμοποιώ γρήγορα τη μέθοδο αυτή (ερ. 11)

15. Μπορώ εύκολα να θυμάμαι πως λειτουργεί η μέθοδος (ερ. 19)

16. Μπορώ εύκολα να προγραμματίσω σε Python χρησιμοποιώντας τη μέθοδο αυτή (ερ. 20)

17. Είναι εύκολο να μάθεις και να χρησιμοποιήσεις τη μέθοδο (ερ. 23)

Ικανοποίηση (satisfaction)

18. Είμαι ικανοποιημένος/η από τη χρήση του ιστολογίου (ερ. 1)

19. Είναι μια καταπληκτική μέθοδος (ερ. 3)

20. Είναι μια ευχάριστη μέθοδος (ερ. 6)

21. Νομίζω ότι πρέπει να γίνεται χρήση αυτής της μεθόδου (ερ. 10)

22. Λειτουργεί με τον τρόπο που επιθυμώ (ερ. 17)

23. Θα σύστηνα το ιστολόγιο σε ένα φίλο/φίλη μου (ερ. 22)

Για κάθε ερωτηματολόγιο, για κάθε μία από τις προαναφερόμενες τέσσερις κατηγορίες που είναι προς μελέτη, υπολογίσθηκε η μέση τιμή όλων των δηλώσεων της κατηγορίας και αυτή η μέση τιμή χρησιμοποιήθηκε ως τιμή κατηγορίας στην στατιστική ανάλυση που παραγματοποιήθηκε. Αυτό μετατρέπει τα σύνολα σε βαθμολογία που κυμαίνεται από 1 έως 7 και διευκολύνει την σύγκριση των επιδόσεων σε κάθε μία υποκλίμακα. Η κατηγοριοποίηση των δηλώσεων σε κάθε κατηγορία έγινε σύμφωνα με τον προηγούμενο πίνακα.

6.5 Ερωτήματα της έρευνας

Στη συνέχεια παρουσιάζονται και αναλύονται τα ερευνητικά ερωτήματα της παρούσας έρευνας.

➤ 1^ο Ερευνητικό ερώτημα

Ο προτεινόμενος τρόπος μετάβασης από το Scratch στη γλώσσα προγραμματισμού Python, μέσω του ιστολογίου που δημιουργήθηκε στη παρούσα εργασία, χαρακτηρίζεται από **χρησιμότητα**.

➤ 2^ο Ερευνητικό ερώτημα

Ο προτεινόμενος τρόπος μετάβασης από το Scratch στη γλώσσα προγραμματισμού Python, μέσω του ιστολογίου που δημιουργήθηκε στη παρούσα εργασία, χαρακτηρίζεται από **ευκολία στη χρήση**.

➤ 3^ο Ερευνητικό ερώτημα

Ο προτεινόμενος τρόπος μετάβασης από το Scratch στη γλώσσα προγραμματισμού Python, μέσω του ιστολογίου που δημιουργήθηκε στη παρούσα εργασία, χαρακτηρίζεται από **ευκολία στη μάθηση**.

➤ 4^ο Ερευνητικό ερώτημα

Υπάρχει στατιστικά σημαντική επίδραση στην **ικανοποίηση** των εκπαιδευομένων κατά τη χρήση του προτεινόμενου τρόπου μετάβασης από το Scratch στη γλώσσα προγραμματισμού Python, μέσω του ιστολογίου που δημιουργήθηκε στη παρούσα εργασία.

➤ 5^ο Ερευνητικό ερώτημα

Η **χρησιμότητα** και η **ευχρηστία**, κατά τη χρήση του προτεινόμενου ιστολογίου, είναι εξαρτημένες μεταβλητές.

6.6 Ερευνητικές υποθέσεις

Οι ερευνητικές υποθέσεις για την απάντηση των ερωτημάτων που έχουν τεθεί είναι:

➤ 1^η Ερευνητική υπόθεση

H₀: Ο προτεινόμενος τρόπος μετάβασης από το Scratch στη γλώσσα προγραμματισμού Python, μέσω του ιστολογίου που δημιουργήθηκε στη παρούσα εργασία, **δεν χαρακτηρίζεται από χρησιμότητα.**

H₁: Ο προτεινόμενος τρόπος μετάβασης από το Scratch στη γλώσσα προγραμματισμού Python, μέσω του ιστολογίου που δημιουργήθηκε στη παρούσα εργασία, **χαρακτηρίζεται από χρησιμότητα.**

➤ 2^η Ερευνητική υπόθεση

H₀: Ο προτεινόμενος τρόπος μετάβασης από το Scratch στη γλώσσα προγραμματισμού Python, μέσω του ιστολογίου που δημιουργήθηκε στη παρούσα εργασία, **δεν χαρακτηρίζεται από ευκολία στη χρήση.**

H₁: Ο προτεινόμενος τρόπος μετάβασης από το Scratch στη γλώσσα προγραμματισμού Python, μέσω του ιστολογίου που δημιουργήθηκε στη παρούσα εργασία, **χαρακτηρίζεται από ευκολία στη χρήση.**

➤ 3^η Ερευνητική υπόθεση

H₀: Ο προτεινόμενος τρόπος μετάβασης από το Scratch στη γλώσσα προγραμματισμού Python, μέσω του ιστολογίου που δημιουργήθηκε στη παρούσα εργασία, **δεν χαρακτηρίζεται από ευκολία στη μάθηση.**

H₁: Ο προτεινόμενος τρόπος μετάβασης από το Scratch στη γλώσσα προγραμματισμού Python, μέσω του ιστολογίου που δημιουργήθηκε στη παρούσα εργασία, **χαρακτηρίζεται από ευκολία στη μάθηση.**

➤ 4^η Ερευνητική υπόθεση

H0: Δεν υπάρχει στατιστικά σημαντική επίδραση στην **ικανοποίηση** των εκπαιδευομένων κατά τη χρήση του προτεινόμενου τρόπου μετάβασης από το Scratch στη γλώσσα προγραμματισμού Python, μέσω του ιστολογίου που δημιουργήθηκε στη παρούσα εργασία.

H1: Υπάρχει στατιστικά σημαντική επίδραση στην **ικανοποίηση** των εκπαιδευομένων κατά τη χρήση του προτεινόμενου τρόπου μετάβασης από το Scratch στη γλώσσα προγραμματισμού Python, μέσω του ιστολογίου που δημιουργήθηκε στη παρούσα εργασία.

➤ **5^η Ερευνητική υπόθεση**

H0: Η χρησιμότητα και η ευχρηστία του προτεινόμενου ιστολογίου, είναι **ανεξάρτητες** μεταβλητές.

H1: Η χρησιμότητα και η ευχρηστία του προτεινόμενου ιστολογίου, είναι **εξαρτημένες** μεταβλητές.

6.7 Στατιστική Ανάλυση Δεδομένων

Για τη στατιστική ανάλυση των δεδομένων χρησιμοποιήθηκε το στατιστικό πακέτο SPSS 22.0, σύμφωνα με [43].

6.7.1 Περιγραφική Στατιστική

Για κάθε ερωτηματολόγιο, για κάθε μία από τις προαναφερόμενες τέσσερις κατηγορίες που είναι προς μελέτη, υπολογίστηκε η μέση τιμή όλων των δηλώσεων της κατηγορίας και αυτή η μέση τιμή χρησιμοποιήθηκε ως τιμή κατηγορίας στην στατιστική ανάλυση που παραγματοποιήθηκε. Αυτό μετατρέπει τα σύνολα σε βαθμολογία που κυμαίνεται από 1 έως 7 και διευκολύνει την σύγκριση των επιδόσεων σε κάθε μία υποκλίμακα. Η κατηγοριοποίηση των δηλώσεων σε κάθε κατηγορία έγινε σύμφωνα με τον πίνακα 5. Στον πίνακα 6.1, παρουσιάζονται τα αποτελέσματα της περιγραφικής στατιστικής ανάλυσης σχετικά με τις τέσσερις κατηγορίες. Τα αποτελέσματα αφορούν το μέγεθος του δείγματος, την ελάχιστη τιμή, την μέγιστη τιμή, την μέση τιμή και τέλος την τυπική απόκλιση.

Πίνακας 6.1: Περιγραφική στατιστική των 4 κατηγοριών

Descriptive Statistics					
	N	Minimum	Maximum	Mean	Std. Deviation
usefulness	25	3,40	7,00	5,9520	,88934
ease_of_use	25	3,88	7,00	5,9200	,79622
ease_of_learn	25	1,75	7,00	5,9500	1,14337
satisfaction	25	2,67	7,00	5,9933	,95073
Valid N (listwise)	25				

Από τις μέσες τιμές του πίνακα, φαίνεται ότι το μέσο σκορ για την κατηγορία χρησιμότητα είναι 5,95, το μέσο σκορ για την κατηγορία ευκολία χρήσης είναι 5,92, το μέσο σκορ για την κατηγορία ευκολία στη μάθηση 5,95 και το μέσο σκορ για την κατηγορία ικανοποίηση είναι 5,99. Από τις μέσες τιμές του πίνακα 6.1 (και δεδομένου ότι οι μέσες τιμές παίρνουν τιμές στο διάστημα από 1 έως 7) προκύπτει ο πίνακας 6.2 με τα ποσοστά αποδοχής σε κάθε κατηγορία.

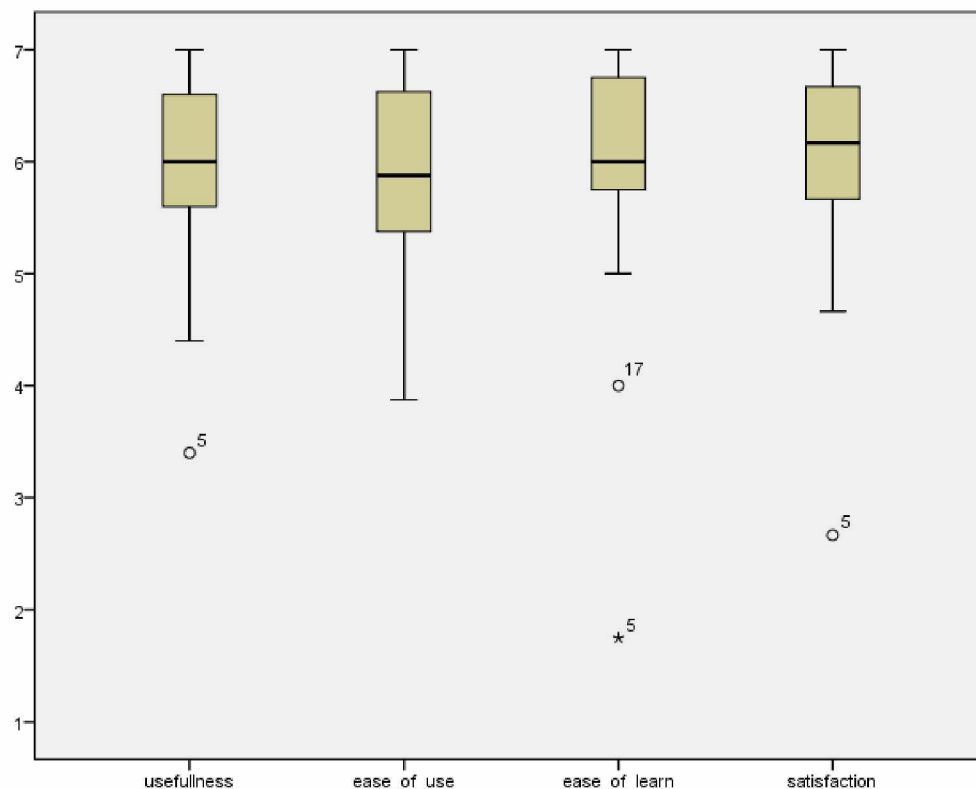
Πίνακας 6.2: Ποσοστά αποδοχής ανά κατηγορία

Χρησιμότητα	82,53%
Ευκολία χρήσης	82%
Ευκολία μάθησης	82,5%
Ικανοποίηση	83,22%

Σύμφωνα με τον πίνακα 6.2, φαίνεται ότι το δείγμα που χρησιμοποιήθηκε για την παρούσα έρευνα, αποδέχεται την προτεινόμενη μέθοδο για την εκμάθηση της γλώσσας προγραμματισμού Python χρησιμοποιώντας σενάρια Scratch, σε ποσοστά μεγαλύτερα από 82% για οποιαδήποτε από τις τέσσερις κατηγορίες.

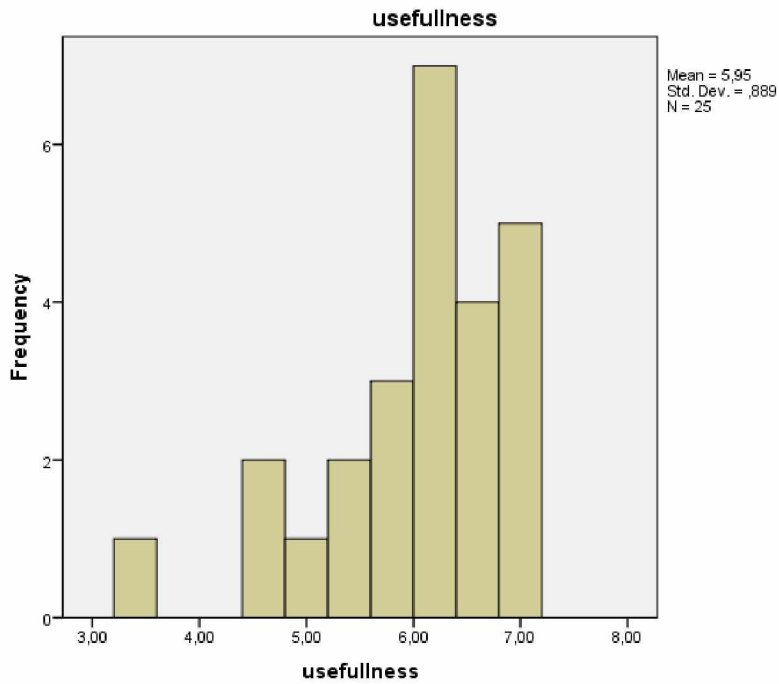
Πιο συγκεκριμένα, οι φοιτητές θεωρούν την προτεινόμενη μέθοδο χρήσιμη σε ποσοστό 82,53%, εύκολη στη χρήση σε ποσοστό 82%, εύκολη στη μάθηση σε ποσοστό 82,5% και ικανοποιητική σε ποσοστό 83,22%

Στην εικόνα 6.1 δίνεται το θηκόγραμμα για τις τέσσερις κατηγορίες του ερωτηματολογίου, στο οποίο εμφανίζονται η διάμεσος, το ενδοτεταρτομοριακό πλάτος και οι προεξέχουσες τιμές.

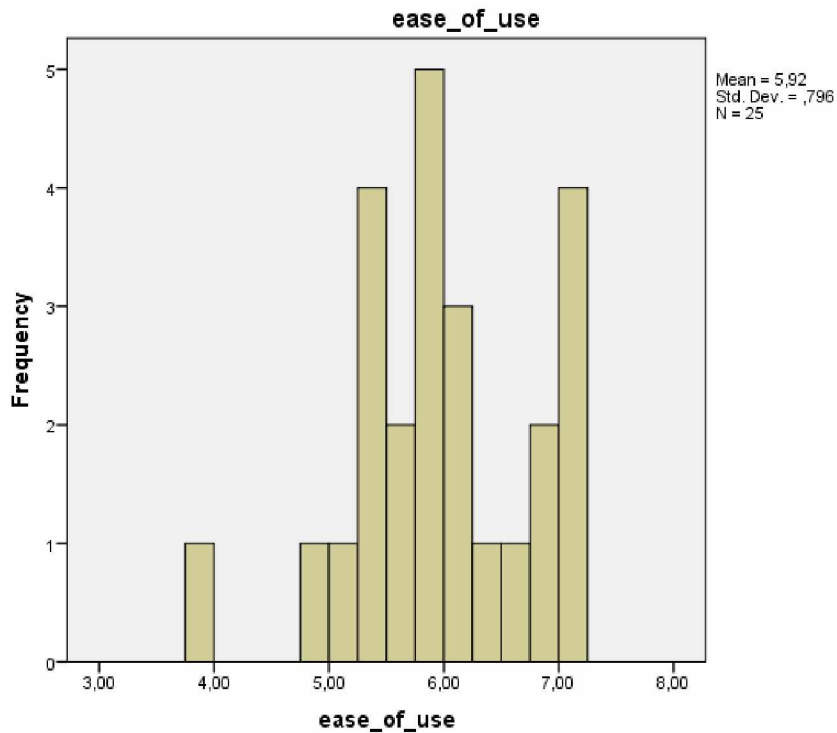


Εικόνα 6.1: Θηκόγραμμα των 4 κατηγοριών

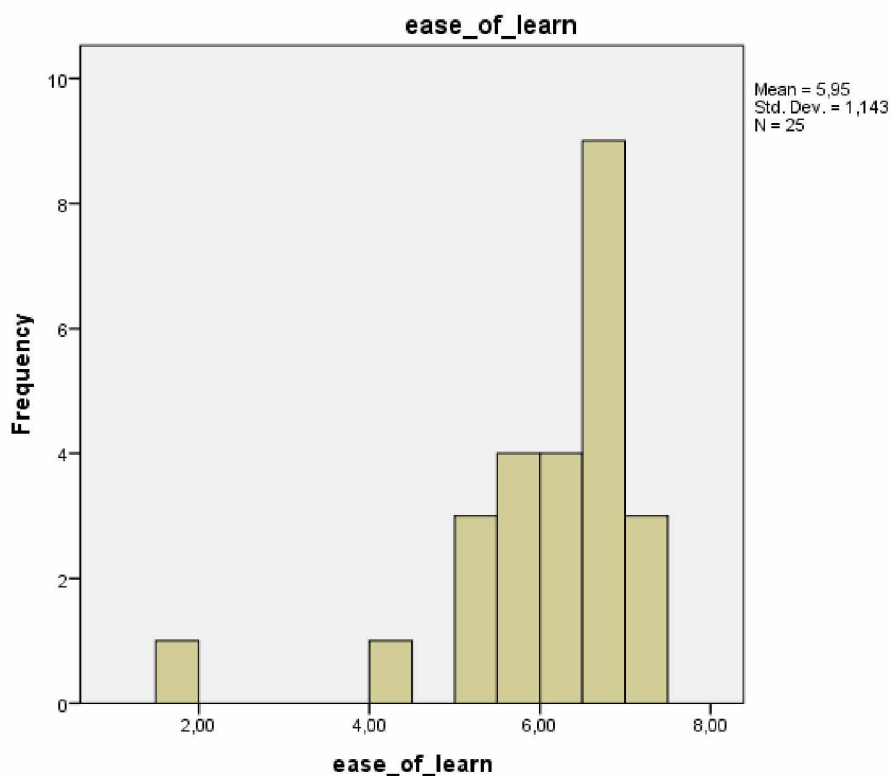
Στη συνέχεια στις εικόνες 6.2, 6.3, 6.4 και 6.5 εμφανίζονται τα ιστογράμματα των κατηγοριών: χρησιμότητα (usefulness), ευχρηστία (ease_of_use), ευκολία στη μάθηση (ease_of_learn), ικανοποίηση (satisfaction).



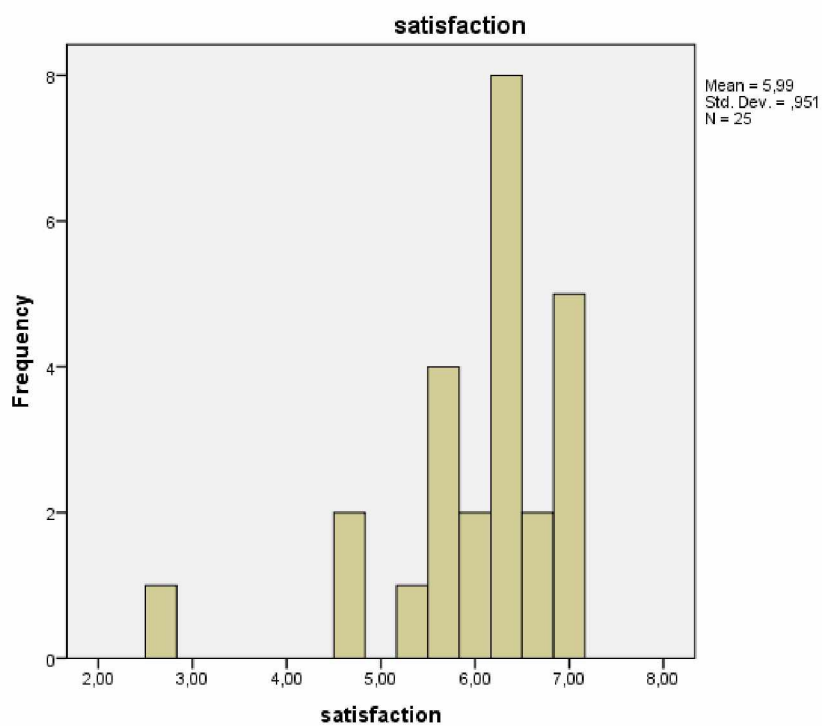
Εικόνα 6.2: Ιστόγραμμα της κατηγορίας Χρησιμότητα



Εικόνα 6.3: Ιστόγραμμα της κατηγορίας Ευχρηστία



Εικόνα 6.4: Ιστόγραμμα της κατηγορίας Ευκολία στη μάθηση



Εικόνα 6.5: Ιστόγραμμα της κατηγορίας Ικανοποίηση

6.7.2 Στατιστικός έλεγχος υποθέσεων

Σύμφωνα με το [43]:

- ❖ Θα ελεγχθεί ο συντελεστής εσωτερικής συνέπειας Cronbach's alpha για να μετρηθεί η αξιοπιστία του χρησιμοποιούμενου ερωτηματολογίου και να καταλήξουμε αν τελικά οι διαφορετικές δηλώσεις μετρούν την ίδια μεταβλητή (κατηγορία). Για να θεωρηθεί αποδεκτή η τιμή του Cronbach's alpha θα πρέπει να είναι μεγαλύτερη από το 0.7 (>0.7). Όσο μεγαλύτερος είναι ο συντελεστής του Cronbach's alpha τόσο μεγαλύτερη είναι η αξιοπιστία.

- ❖ Έλεγχος κανονικότητας των δεδομένων του δείγματος, ώστε να εξεταστεί αν αυτά προσαρμόζονται στην κανονική κατανομή. Δήλωση υποθέσεων:
H₀: Τα δεδομένα προσαρμόζονται ικανοποιητικά στην κανονική κατανομή
H₁: Τα δεδομένα δεν προσαρμόζονται ικανοποιητικά στην κανονική κατανομή

- ❖ Έλεγχος για τη μέση τιμή του δείγματος (one sample T-test), ως προς μια καθορισμένη τιμή από τον ερευνητή, την μ_0 , εδώ είναι το 4 (η μέση τιμή από το 1 έως το 7).

- ❖ Έλεγχος για την εξάρτηση των μεταβλητών χρησιμότητα και ευχρηστία, με τον στατιστικό έλεγχο Pearson (εφόσον ισχύει η κανονικότητα).
H₀: Η χρησιμότητα και η ευχρηστία του προτεινόμενου ιστολογίου, είναι **ανεξάρτητες** μεταβλητές.
H₁: Η χρησιμότητα και η ευχρηστία του προτεινόμενου ιστολογίου, είναι **εξαρτημένες** μεταβλητές.

6.7.3 Στατιστική Ανάλυση Ερευνητικών Ερωτημάτων

1^ο Ερευνητικό Ερώτημα

Πίνακας 6.3: Υπολογισμός αξιοπιστίας της μεταβλητής **Χρησιμότητα** με τον δείκτη Cronbach's alpha

Reliability Statistics	
Cronbach's Alpha	N of Items
,889	5

Σύμφωνα με τον παραπάνω πίνακα, ο συντελεστής Cronbach's alpha είναι 0.889, τιμή μεγαλύτερη από το 0.7. Άρα υπάρχει αξιοπιστία στην εσωτερική συνοχή της κλίμακας.

Πίνακας 6.4: Κανονικότητα δείγματος της μεταβλητής **Χρησιμότητα**

	Tests of Normality					
	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
usefulness	,162	25	,091	,905	25	,024

a. Lilliefors Significance Correction

Από τον πίνακα 6.4, προκύπτει ότι το p-value με τον έλεγχο Kolmogorov-Smirnov, είναι $0.091 > 0.05$, επομένως αποδεχόμαστε την H_0 . Άρα με πιθανότητα λανθασμένης απόρριψης 5% τα δεδομένα μας προσαρμόζονται στην κανονική κατανομή.

Πίνακας 6.5: One Sample T-test για τη μεταβλητή **Χρησιμότητα**

	One-Sample Test					
	Test Value = 4					
	t	df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
usefulness	10,974	24	,000	1,95200	1,5849	2,3191

Σύμφωνα με τον πίνακα 6.5, το p-value του ελέγχου One Sample T-test είναι 0.000, δηλαδή μικρότερο από το 0.05. Άρα απορρίπτουμε την μηδενική υπόθεση. Σαν αποτέλεσμα προκύπτει ότι ο προτεινόμενος τρόπος μετάβασης από το Scratch στην Python μέσω του ιστολογίου χαρακτηρίζεται από **Χρησιμότητα**.

2° Ερευνητικό ερώτημα

Πίνακας 6.6: Υπολογισμός αξιοπιστίας της μεταβλητής **Ευκολία στη χρήση** με τον δείκτη Cronbach's alpha

Reliability Statistics	
Cronbach's Alpha	N of Items
,876	8

Σύμφωνα με τον πίνακα 6.6, ο δείκτης Cronbach's alpha είναι $0.884 > 0.7$. Η τιμή αυτή του συντελεστή δείχνει την αξιοπιστία της εσωτερικής συνοχής της κλίμακας.

Πίνακας 6.7: Κανονικότητα δείγματος της μεταβλητής **Ευκολία στη χρήση**

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Ease of use	,125	25	,200*	,936	25	,119

*. This is a lower bound of the true significance.

a. Lilliefors Significance Correction

Όπως φαίνεται στον πίνακα 6.7, το p-value για τον έλεγχο της κανονικότητας των δεδομένων με τον έλεγχο Kolmogorov-Smirnov, είναι 0.200 και άρα μεγαλύτερο από το 0.05. Σαν αποτέλεσμα έχουμε ότι: Με πιθανότητα λανθασμένης απόρριψης 5%, τα δεδομένα προσαρμόζονται στην κανονική κατανομή.

Πίνακας 6.8: One Sample T-test για τη μεταβλητή **Ευκολία στη χρήση**

One-Sample Test						
	Test Value = 4					
	t	df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
ease_of_use	12,057	24	,000	1,92000	1,5913	2,2487

Σύμφωνα με τον πίνακα 6.8, το p-value του ελέγχου One Sample T-test, είναι $0.000 < 0.05$. Άρα απορρίπτουμε την H_0 . Επομένως με πιθανότητα λανθασμένης απόρριψης 5%, εκτιμάται ότι ο προτεινόμενος τρόπος μετάβασης από το Scratch στην Python με τη χρήση του ιστολογίου που δημιουργήθηκε για αυτό το σκοπό, χαρακτηρίζεται από **ευχρηστία**.

3^ο Ερευνητικό ερώτημα

Πίνακας 6.9: Υπολογισμός αξιοπιστίας της μεταβλητής **Ευκολία στη μάθηση** με τον δείκτη Cronbach's alpha

Reliability Statistics	
Cronbach's Alpha	N of Items
,933	4

Σύμφωνα με τον πίνακα 6.9, ο συντελεστής Cronbach's alpha έχει τιμή $0.933 > 0.7$, γεγονός που δείχνει την αξιοπιστία της εσωτερικής συνοχής της κλίμακας.

Πίνακας 6.10: Κανονικότητα δείγματος της μεταβλητής **Ευκολία στη μάθηση**

	Tests of Normality					
	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
ease_of_learn	,191	25	,020	,767	25	,000

a. Lilliefors Significance Correction

Σύμφωνα με τον πίνακα 6.10, το p-value με τον έλεγχο κανονικότητας των δεδομένων του Kolmogorov-Smirnov, είναι $0.020 < 0.05$ και απορρίπτουμε την H_0 .

Επομένως με πιθανότητα λανθασμένης απόρριψης 5%, τα δεδομένα δεν προσαρμόζονται ικανοποιητικά στην κανονική κατανομή.

Παρόλο που τα δεδομένα δεν προσαρμόζονται στην κανονική κατανομή, θα συνεχίσουμε με τον έλεγχο T-test.

Πίνακας 6.11: One Sample T-test για τη μεταβλητή **Ευκολία στη μάθηση**

One-Sample Test						
	Test Value = 4					
	t	df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
ease_of_learn	8,527	24	,000	1,95000	1,4780	2,4220

Σύμφωνα με τον πίνακα 6.11, το p-value του ελέγχου One Sample T-test, είναι $0.000 < 0.05$. Αυτό έχει σαν αποτέλεσμα την απόρριψη της μηδενικής υπόθεσης. Άρα, με πιθανότητα λανθασμένης απόρριψης 5%, υπάρχει η εκτίμηση ότι το προτεινόμενο ιστολόγιο ως τρόπος μετάβασης από το Scratch στην Python, χαρακτηρίζεται από **ευκολία στη μάθηση**.

4^ο Ερευνητικό ερώτημα

Πίνακας 6.12: Υπολογισμός αξιοπιστίας της μεταβλητής **Ικανοποίηση** με τον δείκτη Cronbach's alpha

Reliability Statistics	
Cronbach's Alpha	N of Items
,910	6

Σύμφωνα με τον πίνακα 6.12, ο συντελεστής Cronbach's alpha έχει τιμή $0.910 > 0.7$, γεγονός που δείχνει την αξιοπιστία της εσωτερικής συνοχής της κλίμακας.

Πίνακας 6.13: Υπολογισμός κανονικότητας δεδομένων της μεταβλητής **Ικανοποίηση**

Tests of Normality						
	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
satisfaction	,172	25	,054	,834	25	,001

a. Lilliefors Significance Correction

Σύμφωνα με τον πίνακα 6.13, το p-value για τον έλεγχο της κανονικότητας του δείγματος με τον έλεγχο Kolmogorov-Wilk, είναι $0.054 > 0.05$ και επομένως αποδεχόμαστε τη μηδενική υπόθεση. Με πιθανότητα λανθασμένης απόρριψης 5%, το δείγμα προσαρμόζεται ικανοποιητικά στην κανονική κατανομή.

Πίνακας 6.14: One Sample T-test για τη μεταβλητή **Ικανοποίηση**

One-Sample Test						
	Test Value = 4					
	t	df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
satisfaction	10,483	24	,000	1,99333	1,6009	2,3858

Σύμφωνα με τον πίνακα 6.14, το p-value του στατιστικού ελέγχου One Sample T-test, είναι $0.000 < 0.05$, άρα απορρίπτουμε την H_0 . Με πιθανότητα λανθασμένης απόρριψης 5% υπάρχει η εκτίμηση ότι ο προτεινόμενος τρόπος μετάβασης από το Scratch στην Python, μέσω του ιστολογίου επιδρά στατιστικά σημαντικά στην ικανοποίηση των εκπαιδευόμενων.

5ο Ερευνητικό ερώτημα

Για να ελεγχθεί η εξάρτηση των μεταβλητών χρησιμότητα και ευχρηστία, θα χρησιμοποιηθεί ο συντελεστής συσχέτισης Correlation Coefficient.

Πίνακας 6.15: Έλεγχος ανεξαρτησίας των μεταβλητών **χρησιμότητα** και **ευχρηστία**

Correlations			
		usefulness	ease_of_use
usefulness	Pearson Correlation	1	,815**
	Sig. (2-tailed)		,000
	N	25	25
ease_of_use	Pearson Correlation	,815**	1
	Sig. (2-tailed)	,000	
	N	25	25

** . Correlation is significant at the 0.01 level (2-tailed).

Σύμφωνα με τον πίνακα 6.15 (Correlations), το p-value στον στατιστικό έλεγχο Pearson, είναι $0.000 < 0.05$, επομένως απορρίπτουμε τη μηδενική υπόθεση και συμπεραίνουμε πως σε επίπεδο σημαντικότητας 5%, οι μεταβλητές ευχρηστία και χρησιμότητα είναι εξαρτημένες.

Επίσης αξίζει να σημειωθεί ότι ο συντελεστής συσχέτισης Pearson είναι ίσος με 0.815. Γεγονός που σημαίνει ότι οι φοιτητές βρήκαν τον προτεινόμενο τόπο μετάβασης από το Scratch στην Python, χρήσιμο, στον ίδιο βαθμό που τον βρήκαν και εύχρηστο.

7. Συμπέρασμα

Η παρούσα έρευνα προσπάθησε να μελετήσει τις τέσσερις κατηγορίες που προκύπτουν από τη χρήση του ερωτηματολογίου USE. Πιο συγκεκριμένα, στην παρούσα εργασία έγινε μια πρόταση εκμάθησης της γλώσσας προγραμματισμού Python χρησιμοποιώντας παραδείγματα υλοποιημένα σε Scratch. Η πρόταση της μετάβασης από μια γλώσσα προγραμματισμού χαμηλότερου επιπέδου σε μια γλώσσα προγραμματισμού υψηλότερου επιπέδου, παρουσιάστηκε σε ένα ιστολόγιο που δημιουργήθηκε για αυτό το σκοπό.

Οι φοιτητές του τμήματος Πληροφορικής με Εφαρμογές στην Βιοϊατρική, αποτέλεσαν το δείγμα της έρευνας και αφού αρχικά περιηγήθηκαν στο ιστολόγιο, συμπλήρωσαν ένα ερωτηματολόγιο 23 ερωτήσεων. Οι ερωτήσεις ομαδοποιήθηκαν σε τέσσερις κατηγορίες, οι οποίες αποτέλεσαν το βασικό αντικείμενο της παρούσας έρευνας.

Αφού προηγήθηκε η συλλογή, η επεξεργασία και η ανάλυση των δεδομένων προέκυψαν τα ακόλουθα συμπεράσματα.

- ❖ Η προτεινόμενη μέθοδος για την εκμάθηση Python χρησιμοποιώντας το Scratch, χαρακτηρίζεται στατιστικά σημαντικά από χρησιμότητα.
- ❖ Χρησιμοποιώντας οι εκπαιδευόμενοι το ιστολόγιο που χρησιμοποιήθηκε για την εκμάθηση της γλώσσας προγραμματισμού Python, θεώρησαν ότι αυτό είναι εύκολο στη χρήση.
- ❖ Το ιστολόγιο blogs.sch.gr/efsork, είναι μια μέθοδος για την εκμάθηση Python, η οποία χαρακτηρίζεται από ευκολία στη μάθηση, σύμφωνα με το δείγμα της παρούσα έρευνας.
- ❖ Η μέθοδος που προτείνεται στην παρούσα διπλωματική εργασία για εκμάθηση της γλώσσας προγραμματισμού Python, κρίθηκε ικανοποιητική από τους φοιτητές που αποτέλεσαν το δείγμα της έρευνας.

Συνοψίζοντας, οι φοιτητές που συμμετείχαν στην παρούσα έρευνα, είχαν θετική στάση απέναντι στην προτεινόμενη μέθοδο μετάβασης από τα σεναρία Scratch στη γλώσσα προγραμματισμού Python, μέσω του ιστολογίου blogs.sch.gr/efsork, σε ποσοστό μεγαλύτερο του 82%. Πιο συγκεκριμένα, οι φοιτητές θεωρούν το

εκπαιδευτικό ιστολόγιο χρήσιμο (82,55%), εύκολο στη χρήση (82%), εύκολο στη μάθηση (82,5%) και ικανοποιητικό (83,22%).

Παράρτημα

Ερωτηματολόγιο USE

Μαθαίνοντας Python από το Scratch, χρησιμοποιώντας το ιστολόγιο παραδειγμάτων: blogs.sch.gr/efsork

Το παρόν ερωτηματολόγιο δημιουργήθηκε για την αξιολόγηση της μεθόδου που προτείνεται για μετάβαση από το Scratch στην Python, η οποία αναπτύχθηκε στα πλαίσια της μεταπτυχιακής μου εργασίας με τίτλο: "Ανάπτυξη εκπαιδευτικού υλικού για τη γλώσσα προγραμματισμού Python με αξιοποίηση σεναρίων σε Scratch", στην κατεύθυνση "Πληροφορική και Τεχνολογία Πληροφορίας και Επικοινωνιών στην Εκπαίδευση" του Διατμηματικού Μεταπτυχιακού Προγράμματος του τμήματος "Πληροφορική και Υπολογιστική Βιοϊατρική".

Βασισμένοι στην εμπειρία σας στη χρήση του Scratch, παρακαλούμε βαθμολογείτε το πώς αισθανθήκατε κατά την περιήγησή σας στο ιστολόγιο blogs.sch.gr/efsork, το οποίο αποτελεί μια πρόταση μεθόδου για την μετάβασή σας από το Scratch στη γλώσσα προγραμματισμού Python. Απλά κυκλώστε το επίπεδο συμφωνίας (αριθμό) που ταιριάζει περισσότερο στη γνώμη σας (όπου 1 σημαίνει ότι διαφωνείτε πλήρως, το 4 σημαίνει ότι έχετε ουδέτερη στάση και το 7 σημαίνει ότι συμφωνείτε πλήρως). Η συνεργασία μέσω της συμμετοχής σας στην έρευνα είναι εθελοντική και ανώνυμη. Οι πληροφορίες που θα συγκεντρωθούν θα χρησιμοποιηθούν αποκλειστικά και μόνο για το σκοπό της συγκεκριμένης έρευνας. Ωστόσο η συμμετοχή σας κρίνεται απαραίτητη για τη σωστή και έγκυρη διεκπεραίωση της. Οι απαντήσεις και οι απόψεις σας πάνω στο συγκεκριμένο θέμα είναι εξαιρετικά σημαντικές. Σας ευχαριστώ πολύ εκ των προτέρων για το χρόνο, τη συνεργασία και τη συμμετοχή σας και παραμένω στη διάθεσή σας για οποιαδήποτε περαιτέρω πληροφορία ή διευκρίνιση (email: efsork65@gmail.com).

Με
Σόρκου
Καθηγήτρια Πληροφορικής, α/θμιας εκπ/σης

εκτίμηση
Ευτυχία

* Απαιτείται

1. Είμαι ικανοποιημένος/η από τη χρήση του ιστολογίου. *

Διαφωνώ πλήρως 1 2 3 4 5 6 7 Συμφωνώ πλήρως

2. Είναι απλή μέθοδος στη χρήση *

Διαφωνώ πλήρως 1 2 3 4 5 6 7 Συμφωνώ πλήρως

3. Είναι μια καταπληκτική μέθοδος *

Διαφωνώ πλήρως 1 2 3 4 5 6 7 Συμφωνώ πλήρως

4. Είναι μια χρήσιμη μέθοδος *

Διαφωνώ πλήρως 1 2 3 4 5 6 7 Συμφωνώ πλήρως

5. Δεν παρατήρησα ασυνέπειες-προβλήματα κατά τη χρήση της μεθόδου *

Διαφωνώ πλήρως 1 2 3 4 5 6 7 Συμφωνώ πλήρως

6. Είναι μια ευχάριστη μέθοδος *

Διαφωνώ πλήρως 1 2 3 4 5 6 7 Συμφωνώ πλήρως

7. Κάνει ότι θα περίμενα από μια τέτοια μέθοδο *

Διαφωνώ πλήρως 1 2 3 4 5 6 7 Συμφωνώ πλήρως

8. Απαιτεί τα λιγότερα δυνατά βήματα για να πετύχω το στόχο μου *

Διαφωνώ πλήρως 1 2 3 4 5 6 7 Συμφωνώ πλήρως

9. Το ιστολόγιο είναι φιλικό στη χρήση του *

Διαφωνώ πλήρως 1 2 3 4 5 6 7 Συμφωνώ πλήρως

10. Νομίζω ότι πρέπει να γίνεται χρήση αυτής της μεθόδου *

Διαφωνώ πλήρως 1 2 3 4 5 6 7 Συμφωνώ πλήρως

11. Έμαθα να χρησιμοποιώ γρήγορα τη μέθοδο αυτή *

Διαφωνώ πλήρως 1 2 3 4 5 6 7 Συμφωνώ πλήρως

12. Θα με βοηθήσει να πετύχω ευκολότερα το στόχο μου *

Διαφωνώ πλήρως 1 2 3 4 5 6 7 Συμφωνώ πλήρως

13. Μπορώ με τη μέθοδο αυτή να διορθώσω ευκολότερα και γρηγορότερα πιθανά λάθη *

Διαφωνώ πλήρως 1 2 3 4 5 6 7 Συμφωνώ πλήρως

14. Η χρήση της μεθόδου δεν απαιτεί ιδιαίτερη προσπάθεια από εμένα *

Διαφωνώ πλήρως 1 2 3 4 5 6 7 Συμφωνώ πλήρως

15. Είναι μια μέθοδος που μπορώ να χρησιμοποιήσω χωρίς γραπτές οδηγίες *

Διαφωνώ πλήρως 1 2 3 4 5 6 7 Συμφωνώ πλήρως

16. Ταιριάζει με τις ανάγκες μου *

Διαφωνώ πλήρως 1 2 3 4 5 6 7 Συμφωνώ πλήρως

17. Λειτουργεί με τον τρόπο που επιθυμώ *

Διαφωνώ πλήρως 1 2 3 4 5 6 7 Συμφωνώ πλήρως

18. Εξοικονομώ χρόνο με τη χρήση της μεθόδου *

Διαφωνώ πλήρως 1 2 3 4 5 6 7 Συμφωνώ πλήρως

19. Μπορώ εύκολα να θυμάμαι πως λειτουργεί η μέθοδος *

Διαφωνώ πλήρως 1 2 3 4 5 6 7 Συμφωνώ πλήρως

20. Μπορώ εύκολα να προγραμματίσω σε Python
χρησιμοποιώντας τη μέθοδο αυτή *

Διαφωνώ πλήρως 1 2 3 4 5 6 7 Συμφωνώ πλήρως

21. Είναι εύκολη μέθοδος στη χρήση της *

Διαφωνώ πλήρως 1 2 3 4 5 6 7 Συμφωνώ πλήρως

22. Θα σύστηνα το ιστολόγιο σε ένα φίλο/φίλη μου *

Διαφωνώ πλήρως 1 2 3 4 5 6 7 Συμφωνώ πλήρως

23. Είναι εύκολο να μάθεις και να χρησιμοποιήσεις τη
μέθοδο *

Διαφωνώ πλήρως 1 2 3 4 5 6 7 Συμφωνώ πλήρως

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] J. Maloney, M. Resnick, N. Rusk, B. Silverman και E. Eastmod, «The Scratch Programming Language and Environment,» *ACM Transactions on Computing Education*, τόμ. 10, αρ. 4, 2010.
- [2] Y. Jimenez, A. Kapoor και C. Gardner-McCune, «Usability Challenges that Novice Programmers Experience when Using Scratch for the First Time,» σε *IEEE Symposium on Visual Languages and Human-Centric Computing*, Lisbon, 2018.
- [3] D. Jonassen, «Evaluating Constructivist Learning,» *Educational Technology*, τόμ. 31, αρ. 9, pp. 28-33, 1991.
- [4] M. Van Gorp και S. Grissom, «An empirical evaluation of using constructive classroom activities to teach introductory programming,» *Computer Science Education*, pp. 247-260, 2001.
- [5] S. Garner, «Learning to Program from Scratch,» σε *Ninth IEEE International Conference on Advanced Learning Technologies*, Riga, 2009.
- [6] M. Ben-Ari, «Principles of Concurrent and Distributed Programming,» 2006.
- [7] M. Ben-Ari και Y. B. Kolikant, «Thinking Parallel: The Process of Learning Concurrency,» pp. 13-16, 1998.
- [8] K. Kahn, «Drawing on napkins, video game animation and other ways to program computers,» *ACM*, τόμ. 39, αρ. 8, pp. 49-59, 1996.
- [9] M. J. Wing, «Computational Thinking,» 2012.
- [10] Β. Κόμης, Εισαγωγή στη διδακτική της πληροφορικής, Αθήνα: Κλειδάριθμος, 2005.
- [11] Α. Τζιμογιάννης, «Προς ένα Παιδαγωγικό Πλαίσιο Διδασκαλίας του Προγραμματισμού στη Δευτεροβάθμια Εκπαίδευση,» σε *3ου Πανελλήνιο Συνεδρίο «Διδακτική της Πληροφορικής»*, Κόρινθος, 2005.
- [12] Δ. Νικολός και Β. Κόμης, «Μια διδακτική πρόταση για τη γλώσσα προγραμματισμού Scratch,» σε *5ο Πανελλήνιο συνέδριο: ΤΠΕ στην εκπαίδευση*, Ερμούπολη, 2010.
- [13] Γ. Φεσάκης, Τ. Καρακίτσα, Ε. Γούλη, Κ. Γλέζου και Α. Γόγουλου, *Εφαρμογές του SCRATCH στη διδασκαλία της Πληροφορικής*.
- [14] D. Gioia και D. Brass, «Teaching the TV generation: The case for observational learning,» αρ. 10, pp. 11-18, 1985.
- [15] Α. Αράπογλου, Ε. Βραχνός, Ε. Κανίδης, Δ. Λέκκα, Π. Μακρυγιάννης, Β. Μπελεσιώτης, Σ. Παπαδάκης και Δ. Τζήμας, Προγραμματισμός Υπολογιστών, 2 επιμ., ΑΘΗΝΑ: ΙΤΥΕ-ΔΙΟΦΑΝΤΟΣ, 2016.

- [16] A. N., «Εισαγωγή στον προγραμματισμό με την Python,» 2015.
- [17] K. D. Lee, *Foundations of Programming Languages*, Springer International Publishing, 2017, pp. 7-10,14.
- [18] C. Swaroopch, «A Byte of Python,» [Ηλεκτρονικό]. Available: <https://python.swaroopch.com/>. [Πρόσβαση 2018].
- [19] M. Lutz, *Learning Python*, O' Reilly Media, 2013.
- [20] P. Guo, «Why Python is a great language for teaching beginners in introductory programming classes,» 2007.
- [21] B. Putano, «Most Popular and Influential Programming Languages of 2018,» 2017.
- [22] S. Cass, «The 2018 Top Programming Languages,» 2018. [Ηλεκτρονικό]. Available: <https://spectrum.ieee.org/at-work/innovation/the-2018-top-programming-languages>.
- [23] Β. Εφόπουλος, Γ. Ευαγγελίδης, Β. Δαγδιλέλης και Α. Κλεφτοδήμος, «Οι δυσκολίες των Αρχάριων Προγραμματιστών,» σε *3ο Πανελλήνιο Συνέδριο "Διδακτική της Πληροφορικής"*, Κόρινθος, 2005.
- [24] B. Du Boulay, «Some difficulties of learning program,» 1986.
- [25] Σ. Ξυνόγαλος, Μ. Σατρατζέμη και Β. Δαγδιλέλης, «Η εισαγωγή στον προγραμματισμό: Διδακτικές προσεγγίσεις και εκπαιδευτικά εργαλεία,» σε *2ο Πανελλήνιο συνέδριο με διεθνή συμμετοχή: Οι τεχνολογίες της πληροφορίας και της επικοινωνίας στην εκπαίδευση*, Πάτρα, 2000.
- [26] Χ. Παναγιωτακόπουλος, Χ. Πιερρακέας και Π. Πιντέλας, «Σχεδίαση εκπαιδευτικού Λογισμικού,» 2005.
- [27] Β. Κόμης και Α. Μικρόπουλος, *Πληροφορική και Εκπαίδευση*, Ελληνικό Ανοικτό Πανεπιστήμιο, 2001.
- [28] L. Nielsen, «10 Reasons Students Say They Prefer Learning Online,» 2010.
- [29] R. Gagnè και L. Briggs, *Principles of instructional Design*, 2 επιμ., New York: Holt, Rinehart and Winston, 1979.
- [30] R. Gagne, *Conditions of Learning*, 3rd επιμ., New York: Holt, Rinehart & Winston, 1977.
- [31] J. Lambrecht, *Developing Employment-Related Office Technology Skills*, University of California, 1999.
- [32] Ε. Κολιάδης, *Θεωρίες Μάθησης και Εκπαιδευτική Πράξη. Συμπεριφοριστικές Θεωρίες*, Αθήνα: Αυτοέκδοση, 1996.
- [33] R. Gagnè, *The conditions of learning and theory instruction*, New York: Holt, Rinehart

and Winston, 1985.

- [34] R. Gagne, «Learning Outcomes and Their Effects: Useful Categories of Human Performance,» *American Psychologist*, τόμ. 4, αρ. 39, 1984.
- [35] K. Davidson, *Education in the internet linking theory to reality*, 1998.
- [36] A. Mathrani, S. Christian και A. Ponder-Sutton, «PlayIt: Game Based Learning Approach for Teaching Programming Concepts,» *ERIC*, p. 13, 2016.
- [37] Γ. Σιδέρης, *Ανάπτυξη παιχνιδιού σοβαρού σκοπού για την εκμάθηση εννοιών προγραμματισμού με τη γλώσσα Python*, Θεσσαλονίκη, 2019.
- [38] Ν. Μαγκιτούκα και Β. Μουτζούρη, «Συγκριτική Μελέτη Περιβαλλόντων και Πλατφορμών Για τη Διδασκαλία του Προγραμματισμού,» Βόλος, 2010.
- [39] Τ. Θεοφανέλλης, «Η ιστοσελίδα codeacademy ως εργαλείο για την εκμάθηση προγραμματισμού,» *Open Education*, τόμ. 12, αρ. 1, pp. 95-105, 2016.
- [40] Έ. κ. Θ. Υπ. Παιδείας, *Οδηγίες διδασκαλίας για τις Τεχνολογίες Πληροφορίας και Επικοινωνιών (ΤΠΕ) όλων των τάξεων του Δημοτικού σχολείου*, 2018.
- [41] G. Halfacree, «Python,» σε *The Official BBC micro:bit® User Guide*, John Wiley & Sons, Inc, 2017, p. 36.
- [42] A. Lund, *Measuring Usability with the USE Questionnaire.*, 2001.
- [43] Ι. Τριανταφύλλου, «Στατιστική Ανάλυση Δεδομένων με χρήση του πακέτου IBM SPSS Statistics,» 2017.

