



ΤΕΧΝΟΛΟΓΙΚΟ
ΕΚΠΑΙΔΕΥΤΙΚΟ
ΙΔΡΥΜΑ
ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ

ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ

ΤΜΗΜΑ ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΕΩΝ (ΠΑΤΡΑ)

ΠΡΩΗΝ ΤΜΗΜΑ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΥ ΣΧΕΔΙΑΣΜΟΥ ΚΑΙ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη εκπαιδευτικού υλικού για την υλοποίηση ενός Μαζικού Ανοικτού Διαδικτυακού Μαθήματος (Massive Open Online Course - MOOC) για την εκμάθηση της Γλώσσας Προγραμματισμού Python

Ένεμαρκ-Κρεμμύδα Ολυμπία - ΑΜ: 11743

Λάζαρη Αλεξάνδρα - ΑΜ: 11589

Επιβλέπων: Κουτσονίκος Ιωάννης

ΠΑΤΡΑ, 2017

ΠΡΟΛΟΓΟΣ

Για την ολοκλήρωση των σπουδών μας στο ΤΕΙ Δυτικής Ελλάδας ήταν απαραίτητο να συγγράψουμε μια πτυχιακή εργασία. Το θέμα μας τράβηξε κατευθείαν. Αρχικά μας άρεσε η ιδέα να μάθουμε Python, μια γλώσσα προγραμματισμού που δεν είχαμε διδαχθεί στο ΤΕΙ, και επιπλέον μας άρεσε η προοπτική της «διδασκαλίας» μέσω ΜΟΟC.

Στο πρώτο μέρος της εργασίας μας παρουσιάζουμε και εξηγούμε τα ΜΟΟCs. Στο δεύτερο μέρος παρουσιάζεται το ΜΟΟC που αναπτύξαμε με τίτλο «Βασικές αρχές προγραμματισμού χρησιμοποιώντας την Python». Το ΜΟΟC που αναπτύξαμε αποτελείται από έξι ενότητες οι οποίες αντιστοιχούν στο χρονικό διάστημα έξι εβδομάδων (μια ενότητα για κάθε εβδομάδα). Κάθε ενότητα αποτελείται από 3 υποενότητες, και κάθε υποενότητα αποτελείται από ένα (1) έως πέντε (5) μαθησιακά αντικείμενα. Στο τέλος κάθε ενότητας δίνονται ασκήσεις αξιολόγησης. Ο σκοπός είναι ο εκπαιδευόμενος να είναι σε θέση να κατανοεί και να δημιουργεί προγράμματα με τη γλώσσα Python όταν ολοκληρώσει το διαδικτυακό αυτό μάθημα.

Στην πρώτη ενότητα γίνεται μια εισαγωγή για την Python, ποια είναι η ιστορία και φιλοσοφία της, κάποια βασικά που πρέπει να γνωρίζει κανείς για τη συγγραφή και εξηγείται πώς γίνεται η εγκατάσταση. Στη δεύτερη και τρίτη ενότητα ο εκπαιδευόμενος έρχεται σε επαφή με βασικές αρχές του προγραμματισμού γενικότερα. Στην τέταρτη ενότητα παρουσιάζονται οι εντολές ελέγχου, ενώ στην πέμπτη οι εντολές επανάληψης. Στην έκτη ενότητα παρουσιάζονται οι δομές δεδομένων, οι κατηγορίες τους και πώς αυτές χρησιμοποιούνται και συντάσσονται.

Θέλουμε να ευχαριστήσουμε τον καθηγητή μας κ. Κουτσονίκο Ιωάννη για τη συνεργασία και τις συμβουλές του.

ΠΕΡΙΕΧΟΜΕΝΑ

1.	ΠΕΡΙΛΗΨΗ.....	1
2.	ABSTRACT.....	2
3.	ΕΙΣΑΓΩΓΗ.....	3
3.1.	Τι είναι ένα MOOC	3
3.2.	Τύποι των MOOC.....	4
3.3.	Δημοφιλέστερες Υπηρεσίες Παροχής MOOC	4
3.4.	Πλεονεκτήματα - Δυνατότητες που προσφέρουν τα MOOC.....	5
3.5.	Μειονεκτήματα - Σκεπτικισμός σχετικά με τα MOOC	7
3.6.	Περιγραφή Μαθήματος	8
4.	ΕΝΟΤΗΤΑ 1 - ΕΙΣΑΓΩΓΗ ΣΤΗ ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ PYTHON.....	11
4.1.	Ιστορία και Φιλοσοφία της Python.....	12
4.1.1.	Μαθησιακό Αντικείμενο 1 - Γιατί επιλέγουμε την Python	12
4.2.	Εγκατάσταση και Προγραμματιστικό Περιβάλλον	14
4.2.1.	Μαθησιακό Αντικείμενο 2 - Προγραμματιστικό Περιβάλλον	15
4.3.	Σύνταξη στο Προγραμματιστικό Περιβάλλον	16
4.3.1.	Μαθησιακό Αντικείμενο 3 - Σύνταξη και αποθήκευση στο προγραμματιστικό περιβάλλον.....	16
4.3.2.	Μαθησιακό Αντικείμενο 4 - Η Python ως αριθμομηχανή	18
4.4.	Ασκήσεις 1ης Ενότητας	19
5.	ΕΝΟΤΗΤΑ 2 - ΜΕΤΑΒΛΗΤΕΣ, ΣΤΑΘΕΡΕΣ, ΤΥΠΟΙ ΔΕΔΟΜΕΝΩΝ.....	21
5.1.	Τύποι Δεδομένων	22
5.1.1.	Μαθησιακό Αντικείμενο 5 - Τύποι Δεδομένων	23
5.2.	ΥΠΟΕΝΟΤΗΤΑ 2 - Σταθερές και Μεταβλητές.....	24
5.2.1.	Μαθησιακό Αντικείμενο 6 - Σταθερές και Μεταβλητές	25
5.3.	Εντολές Εισόδου/Εξόδου	27
5.3.1.	Μαθησιακό Αντικείμενο 7 - Εντολές Εισόδου/Εξόδου.....	28
5.4.	Ασκήσεις 2ης Ενότητας	30
6.	ΕΝΟΤΗΤΑ 3 - ΤΕΛΕΣΤΕΣ.....	32
6.1.	Αριθμητικοί Τελεστές.....	33
6.1.1.	Μαθησιακό Αντικείμενο 8 - Αριθμητικοί Τελεστές	34
6.2.	Συγκριτικοί Τελεστές	35
6.2.1.	Μαθησιακό Αντικείμενο 9 - Συγκριτικοί Τελεστές.....	36
6.3.	Λογικοί Τελεστές	37
6.3.1.	Μαθησιακό Αντικείμενο 10 - Λογικοί Τελεστές.....	37
6.4.	Ασκήσεις 3ης Ενότητας	39
7.	ΕΝΟΤΗΤΑ 4 - ΕΝΤΟΛΕΣ ΕΛΕΓΧΟΥ	42
7.1.	Εντολή Ελέγχου if	43
7.1.1.	Μαθησιακό Αντικείμενο 11 - Εντολή Ελέγχου if.....	43

7.2.	Εντολή Ελέγχου elif και Εμφωλευμένες Εντολές	45
7.2.1.	Μαθησιακό Αντικείμενο 12 - Εντολή Ελέγχου elif και Εμφωλευμένες εντολές	46
7.3.	Συναρτήσεις.....	49
7.3.1.	Μαθησιακό Αντικείμενο 13 - Συναρτήσεις.....	49
7.4.	Ασκήσεις 4ης Ενότητας	52
8.	ΕΝΟΤΗΤΑ 5 - ΕΝΤΟΛΕΣ ΕΠΑΝΑΛΗΨΗΣ.....	53
8.1.	Εντολή Επανάληψης for	54
8.1.1.	Μαθησιακό Αντικείμενο 14 - Εντολή Επανάληψης for	54
8.2.	Εντολή Επανάληψης while	57
8.2.1.	Μαθησιακό Αντικείμενο 15 - Εντολή Επανάληψης while	58
8.3.	Εντολές “αναγκαστικής εξόδου” break και continue	62
8.3.1.	Μαθησιακό Αντικείμενο 16 - Εντολές “αναγκαστικής εξόδου” break και continue.....	63
8.4.	Ασκήσεις 5ης Ενότητας	64
9.	ΕΝΟΤΗΤΑ 6 - ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ	66
9.1.	Λίστες	67
9.1.1.	Μαθησιακό Αντικείμενο 17 - Δημιουργία λίστας και προσπέλαση στοιχείων στη λίστα.....	67
9.1.2.	Μαθησιακό Αντικείμενο 18 - Πράξεις μέσα σε μια λίστα και φέτες λίστας.....	71
9.1.3.	Μαθησιακό Αντικείμενο 19 - Αφαίρεση, πρόσθεση και διαγραφή στοιχείων σε μια λίστα	72
9.1.4.	Μαθησιακό Αντικείμενο 20 - Αντικείμενα, ψευδώνυμα και λίστες κλώνοι.....	74
9.1.5.	Μαθησιακό Αντικείμενο 21 - Εμφωλευμένες λίστες, πίνακες και συμβολοσειρές.....	76
9.2.	Πλειάδες.....	79
9.2.1.	Μαθησιακό Αντικείμενο 22 - Πλειάδες	80
9.3.	Λεξικά	81
9.3.1.	Μαθησιακό Αντικείμενο 23 - Λεξικά.....	82
9.4.	Ασκήσεις 6ης Ενότητας	84
	ΠΑΡΑΡΤΗΜΑ 1. ΑΠΑΝΤΗΣΕΙΣ ΑΣΚΗΣΕΩΝ ΑΥΤΥΟΑΞΙΟΛΟΓΗΣΗΣ	86
	ΠΑΡΑΡΤΗΜΑ 2. ΠΡΟΤΥΠΟΙ ΠΙΝΑΚΕΣ	88
	Περιγραφή Μαθήματος.....	88
	Περιγραφή Ενοτήτων.....	88
	Περιγραφή Υποενοτήτων	89
	Περιγραφή Μαθησιακών Αντικειμένων.....	89
	ΒΙΒΛΙΟΓΡΑΦΙΑ - ΑΝΑΦΟΡΕΣ	90

1. ΠΕΡΙΛΗΨΗ

Για την παρούσα πτυχιακή εργασία αναπτύχθηκε εκπαιδευτικό υλικό (θεωρία, παραδείγματα και ασκήσεις αυτοαξιολόγησης) για τη γλώσσα προγραμματισμού Python. Ο σκοπός της ανάπτυξης του εκπαιδευτικού υλικού είναι να δημιουργηθεί ένα μαζικό ανοιχτό διαδικτυακό μάθημα (Massive Open Online Course, MOOC) για την εκμάθηση της γλώσσας προγραμματισμού Python μέσω του διαδικτύου, χωρίς την παρουσία εκπαιδευτή.

Το μάθημα αυτό απευθύνεται σε όποιον επιθυμεί να μάθει τη γλώσσα προγραμματισμού Python, χωρίς να είναι απαραίτητο να έχει γνώσεις προγραμματισμού. Μετά την εισαγωγική παρουσίαση των MOOC (τι είναι, ποια είναι τα πλεονεκτήματα και μειονεκτήματά τους) παρουσιάζονται οι έξι ενότητες του μαθήματος που δημιουργήσαμε, με τίτλο «Βασικές αρχές προγραμματισμού χρησιμοποιώντας την Python». Σε αυτές τις ενότητες αναφέρεται η ιστορία και φιλοσοφία της Python, εξηγείται ο τρόπος σύνταξής της, καθώς και έννοιες απαραίτητες για τη δημιουργία προγραμμάτων.

Η μελέτη κάθε ενότητας εκτιμάται να έχει διάρκεια μίας εβδομάδας. Κάθε ενότητα αποτελείται από τρεις υποενότητες. Μια υποενότητα μπορεί να έχει από ένα (1) έως πέντε (5) μαθησιακά αντικείμενα. Τα μαθησιακά αντικείμενα αποτελούνται από θεωρία και παραδείγματα για την καλύτερη εμπέδωση της θεωρίας. Στο τέλος κάθε ενότητας υπάρχουν ασκήσεις αυτοαξιολόγησης για τους εκπαιδευόμενους, σχεδιασμένες έτσι ώστε να μπορούν να αξιολογούν οι ίδιοι την πρόοδο της μελέτης τους, αφού ο όγκος των εκπαιδευομένων ενδέχεται να είναι πολύ μεγάλος.

Στο τέλος της εργασίας, δίνονται σε Παραρτήματα οι πίνακες που αποτυπώνουν τη μεθοδολογία που ακολουθήθηκε, καθώς και οι απαντήσεις στις ασκήσεις αυτοαξιολόγησης.

2. ABSTRACT

This thesis is concerned with the development of learning material for the Python programming language. The material contains theory, examples and self-evaluating exercises. The purpose of the development of the learning material is to create a Massive Open Online Course (MOOC) for the learning of Python via the Internet, without the presence of an instructor.

This course aims to anyone who wishes to learn Python, without being essential to have a background on programming. After the introductory presentation of MOOCs (content, advantages and disadvantages), the six sections of the created lesson are presented, entitled “Basic programming principles using Python”. In these sections the history and principles of Python are mentioned, its method of syntax and the necessary concepts for building programs are described.

The study of each section is approximated to last one week. Each section consists of three subsections and each subsection can include one to five learning objects. The learning objects consist of theory and examples for the better understanding of the theory. At the end of each subsection there are self-evaluating exercises, for the trainees, properly designed in order for them to be able to evaluate their personal study progress, since the number of trainees can be large.

At the end of this thesis, appendices and tables depicting the proposed methodology, as well as the answers to the self-evaluating exercises, are presented.

3. ΕΙΣΑΓΩΓΗ

3.1. Τι είναι ένα MOOC

Ο όρος MOOC (Massive Open Online Courses) ή «Μαζικά Ελεύθερα Διαδικτυακά Μαθήματα», όπως αποδίδεται στα ελληνικά, έκανε την εμφάνισή του για πρώτη φορά στο χώρο της εκπαίδευσης το 2008 από τους καθηγητές George Siemens και David Cormier του Πανεπιστημίου της Manitoba του Καναδά και από τότε έχει προσελκύσει την προσοχή των Μέσων Μαζικής Ενημέρωσης καθώς και διεθνώς αναγνωρισμένων εκπαιδευτικών ιδρυμάτων. Το ακρωνύμιο του ονόματος MOOC δείχνει την ουσιαστική σημασία των μαθημάτων αυτών: «massive» σημαίνει «μαζικά» (έχουν σχεδιαστεί ώστε να επιτρέπουν την εγγραφή δεκάδων χιλιάδων εκπαιδευομένων), «open» σημαίνει «ελεύθερα» (ο καθένας με μια σύνδεση στο διαδίκτυο μπορεί να εγγραφεί στην ελεύθερη σειρά μαθημάτων), «online» σημαίνει «διαδικτυακά» (ένα σημαντικό μέρος ή αν όχι όλη η αλληλεπίδραση λαμβάνει χώρα διαδικτυακά μέσω ομάδων συζητήσεων ή/και παρακολούθησης βιντεοδιαλέξεων) και, τέλος, «courses» σημαίνει «μαθήματα» (έχουν συγκεκριμένες ημερομηνίες έναρξης / λήξης και αξιολόγησης των εκπαιδευομένων).

Η διδασκαλία μέσω ενός MOOC γίνεται με διάφορους τύπους εκπαιδευτικού υλικού, κυρίως με τα εκπαιδευτικά βίντεο να αξιοποιούνται σε υψηλό ποσοστό. Παράλληλα, οι εκπαιδευόμενοι έχουν τη δυνατότητα αλληλεπίδρασης με τους συνεκπαιδευομένους τους μέσα από διαδικτυακές ομάδες συζητήσεων.

Παρακάτω παρουσιάζονται οι πιο γνωστοί τύποι εκπαιδευτικού υλικού που μπορούν να χρησιμοποιηθούν από την πλατφόρμα ενός MOOC:

- Εκπαιδευτικά Βίντεο (Educational Videos): Τα εκπαιδευτικά βίντεο παρουσιάζουν συγκεκριμένες ενότητες εκπαιδευτικού περιεχομένου με οπτικοακουστικό υλικό από τον διδάσκοντα. Διατίθενται σε μορφή η οποία μπορεί να αναπαραχθεί είτε σε CD είτε μέσω φυλλομετρητών στο διαδίκτυο.
- Παρουσίαση (Presentation): Μία παρουσίαση μπορεί να είναι οποιοσδήποτε τύπος μορφής διαφανειών για τους εκπαιδευομένους.
- Υπερκείμενο (Hypertext): Αυτός ο τύπος εκπαιδευτικού υλικού αποτελεί ηλεκτρονικό κείμενο που έχει διασυνδεθεί με άλλα κείμενα.
- Δοκίμιο (Document): Ένα δοκίμιο μπορεί να είναι οποιοσδήποτε τύπος εγγράφου PDF ή φυλλάδιο για εκπαιδευομένους.
- Ήχος (Audio): Ένας ήχος μπορεί να είναι οποιοσδήποτε τύπος ήχου για τους εκπαιδευομένους.
- Κουίζ: Αποτελεί οποιονδήποτε τύπο κουίζ που υλοποιεί ο εκπαιδευόμενος.
- MASHUP: Αυτός ο τύπος εκπαιδευτικού υλικού αποτελεί την εμφάνιση βίντεο και παρουσίασης ταυτόχρονα δίπλα- δίπλα στην οθόνη του εκπαιδευόμενου.

- Wiki: Ένα Wiki είναι συνήθως μία ιστοσελίδα που επιτρέπει στους χρήστες της να προσθέσουν, να αφαιρέσουν ή να επεξεργαστούν το περιεχόμενό της, γρήγορα και εύκολα.
- Εργασίες: Αποτελεί οποιαδήποτε εργασία (Project) που δίνεται στον εκπαιδευόμενο.

Τα ΜΟΟC θεωρούνται ως η πλέον πρόσφατη εξέλιξη στην εξ αποστάσεως εκπαίδευση και γίνονται όλο και πιο δημοφιλή εντός των κύκλων της τριτοβάθμιας εκπαίδευσης, αλλά και σε μεγάλα αμερικάνικα πανεπιστήμια, όπως το Στάνφορντ, το Πρίνστον, το πανεπιστήμιο του Μίσιγκαν καθώς και το Μπέρκλεϊ στην Καλιφόρνια, που προσφέρουν δωρεάν κύκλους σπουδών σε διάφορα γνωστικά αντικείμενα, παρέχοντας ένα πιστοποιητικό παρακολούθησης το οποίο δεν ισοδυναμεί με κανονικό πτυχίο. Όσον αφορά την απόκτηση της πιστοποίησης παρακολούθησης ο εκπαιδευόμενος μπορεί να χρεωθεί ένα χρηματικό ποσό, το οποίο όμως δεν συμπεριλαμβάνει το εκπαιδευτικό υλικό ή τη χρήση της εκπαιδευτικής πλατφόρμας.

3.2. Τύποι των ΜΟΟC

Παρά το σύντομο χρονικό διάστημα ύπαρξής τους, τα ΜΟΟC έχουν ήδη χωριστεί σε δυο διακριτούς τύπους:

- cΜοοc (connectivist ΜΟΟC): Τα ΜΟΟC αυτού του τύπου βασίζονται στις αρχές του Κονεκτιβισμού, όπως η αυτονομία, η ποικιλομορφία, η αλληλεπίδραση και η ανεκτικότητα. Τα cΜοοc δίνουν έμφαση στη συνεργατική μάθηση και τα μαθήματα είναι χτισμένα γύρω από μια ομάδα ατόμων της ίδιας φιλοσοφίας. Ο εκπαιδευόμενος έχει τη δυνατότητα να πλοηγηθεί σε μια πληθώρα διαδικτυακών πηγών γνώσης, επιλέγοντας μόνος του το περιεχόμενο, τον τρόπο, το χρόνο και τα εργαλεία που θα χρησιμοποιήσει για να επιτύχει τους προσωπικούς του εκπαιδευτικούς στόχους.
- xΜοοc (extended ΜΟΟC): Στα ΜΟΟC αυτού του τύπου δίνεται ιδιαίτερη έμφαση στο περιεχόμενο των μαθημάτων. Οι υποχρεώσεις των εκπαιδευομένων έχουν στόχο την καλή 'απορρόφηση' του περιεχομένου των μαθημάτων και με την ολοκλήρωσή τους, επικεντρώνονται στη διαδικασία αξιολόγησης του περιεχομένου.

3.3. Δημοφιλέστερες Υπηρεσίες Παροχής ΜΟΟC

Οι πιο γνωστές υπηρεσίες είναι οι εξής:

- Η edx (<http://www.edX.org/>) είναι μια μη κερδοσκοπική πλατφόρμα παροχής ΜΟΟC που δημιουργήθηκε από τα πανεπιστήμια MIT και Harvard, τα οποία τη χρηματοδότησαν με 60 εκατομμύρια δολάρια. Το 2013 προσφέρθηκαν μέσω της

πλατφόρμας 92 μαθήματα. Στη συγκεκριμένη πλατφόρμα παρέχονται 3 είδη διαφορετικών πιστοποιητικών (Honor Code, ID Verified και XSeries Certificates).

- Η Αμερικάνικη εταιρία Coursera (<http://www.coursera.org/>) είναι μια αφιλοκερδούς χαρακτήρα εκπαιδευτική εταιρία, η οποία ξεκίνησε με χρηματοδότηση 22 εκατομμυρίων δολαρίων από διάφορους επενδυτές επιχειρηματικών κεφαλαίων. Στη συγκεκριμένη πλατφόρμα υπάρχουν 4 πανεπιστημιακοί εταίροι, τα πανεπιστήμια Stanford, Princeton, Michigan και Pennsylvania.

Το έτος 2013 η Coursera παρείχε 540 μαθήματα σε 18 διαφορετικά γνωστικά αντικείμενα από 107 συνεργάτες, ενώ είχε 5,5 εκατομμύρια εγγεγραμμένους χρήστες. Ορισμένα από τα συνεργαζόμενα πανεπιστήμια προσφέρουν σε όσους εκπαιδευόμενους είναι διατεθειμένοι να καταβάλλουν μια συνδρομή, επιπλέον εκπαιδευτικό υλικό με εργασίες και δραστηριότητες καθώς και τη δυνατότητα απόκτησης επίσημου πιστοποιητικού.

- Η Udacity (<http://www.udacity.com/>) είναι μια κερδοσκοπική οργάνωση που ιδρύθηκε από τους Sebastian Thrun, David Stavens, και Mike Sokolsky και έχει χρηματοδοτηθεί από ιδιωτικές εταιρίες επενδύσεων. Το έτος 2013 η Udacity προσέφερε δωρεάν 28 μαθήματα από 5 διαφορετικούς γνωστικούς τομείς. Στην περίπτωση που κάποιος εκπαιδευόμενος επιθυμεί την απόκτηση πιστοποίησης μετά το πέρας των μαθημάτων και εφόσον καταβάλλει ένα χρηματικό ποσό θα λάβει πιστοποιητικό ολοκλήρωσης το οποίο θα παρουσιάζει τις επιδόσεις του, υπογεγραμμένο από τους εκπαιδευτές.

Εκτός από τις τρεις δημοφιλέστερες υπηρεσίες παροχής MOOC, υπάρχουν και άλλες λιγότερο γνωστές αλλά δυναμικές υπηρεσίες και είναι οι εξής:

- Udemy (<http://www.udemy.com/>)
- Khan Academy (<http://www.khanacademy.org/>)
- FutureLearn (<http://www.futurelearn.com/>)
- OpenupEd (<http://www.openuped.eu/>)

3.4. Πλεονεκτήματα - Δυνατότητες που προσφέρουν τα MOOC

Ο κύριος στόχος των MOOC είναι να παρέχουν ανοιχτή μάθηση και ελεύθερη πρόσβαση πανεπιστημιακού επιπέδου σε όσο το δυνατόν περισσότερους φοιτητές. Οι Chen, Barnett και Stephens στο άρθρο τους Fad of Future: The Advantages and Challenges of Massive Open Online Courses (2013) διακρίνουν τα εξής πλεονεκτήματα:

Μαζικότητα: Μπορούν να υποστηρίξουν μεγάλο πλήθος συμμετεχόντων καθώς παρέχουν πρόσβαση σε ανθρώπους οι οποίοι διαφορετικά θα είχαν αποκλειστεί για ποικίλους λόγους, όπως ο χρόνος και ο τόπος διεξαγωγής των μαθημάτων, οι τυπικές προϋποθέσεις και οι οικονομικές δυσκολίες. Τα MOOC αναδύουν μια αίσθηση

δημοκρατικής και ελεύθερης πρόσβασης στην εκπαίδευση, καθώς και τη φιλοσοφία ότι οι δυνατότητες εκπαίδευσης δεν θα πρέπει να περιορίζονται πλέον σε προνομιούχους. Οι συμμετέχοντες και οι εκπαιδευτές διαπιστώνουν τα σημαντικά οφέλη από τη βελτιωμένη προσβασιμότητα που προσφέρουν τα ΜΟΟC. Η μη ύπαρξη διδασκτρων και η online μορφή των ΜΟΟC προσφέρει πρόσβαση και ευελιξία, καθώς επίσης μειώνει την ανάγκη για προαπαιτούμενα. Παράλληλα, οι υποστηρικτές των ΜΟΟC υποστηρίζουν ότι ηλεκτρονικές πλατφόρμες μάθησης όπως η Coursera, edX και Udacity συμβάλλουν στον εκδημοκρατισμό της τριτοβάθμιας εκπαίδευσης σε χώρες που πριν είχαν μικρή πρόσβαση σε αυτήν.

Ανοικτότητα: Στα ΜΟΟC όλα είναι ανοικτά, από το λογισμικό που χρησιμοποιείται μέχρι την εγγραφή στα μαθήματα και τη διδακτέα ύλη, η οποία δεν είναι αυστηρά καθορισμένη αλλά μπορεί να μεταβάλλεται καθώς το μάθημα εξελίσσεται στην πορεία. Οι πηγές πληροφόρησης είναι ανοικτές και, τέλος, τα μαθησιακά περιβάλλοντα που θα επιλέξει ο χρήστης είναι και αυτά ανοικτά. Έτσι τα ΜΟΟC μέχρι σήμερα έχουν εγγράψει εκατομμύρια εκπαιδευόμενους από όλο τον κόσμο, με 5.5 εκατομμύρια χρήστες να εγγράφονται μόνο στα μαθήματα που παρέχει η Coursera. Άλλωστε, για τα ίδια τα πανεπιστημιακά ιδρύματα η παροχή ΜΟΟC μπορεί να επεκτείνει διεθνώς την πρόσβαση και τη φήμη τους. Επιπροσθέτως, τα πανεπιστήμια MIT και Harvard χρησιμοποιούν την πλατφόρμα edX προκειμένου να κατανοήσουν αφενός πώς μαθαίνουν οι φοιτητές τους και αφετέρου να εισάγουν τις καινοτομίες των ΜΟΟC όσον αφορά τη διδασκαλία και τη μάθηση εντός των πανεπιστημιακών αιθουσών.

Φιλοσοφία του Κονεκτιβισμού: Η πιο σύγχρονη θεωρία μάθησης στην ψηφιακή εποχή είναι ο Κονεκτιβισμός, θεμελιωτές του οποίου θεωρούνται οι George Siemens και Stephen Downes. Κύρια θέση του Κονεκτιβισμού είναι η αξιοποίηση δικτύων συνδέσμων, τα οποία αποτελούνται από κόμβους στους οποίους εμπεριέχεται η γνώση με αποτέλεσμα την κατασκευή από τους ίδιους τους εκπαιδευόμενους αντίστοιχων δικτύων. Τα ΜΟΟC βασίζονται στις αρχές αυτής της θεωρίας όπως: η αυτονομία, η ποικιλομορφία, η ανοικτότητα και η αλληλεπίδραση. Δηλαδή ο σύγχρονος εκπαιδευόμενος μαθαίνει πλοηγούμενος σε μια πληθώρα διαδικτυακών πηγών γνώσης, επιλέγοντας μόνος του το περιεχόμενο, τον τρόπο, το χρόνο και τα εργαλεία που θα χρησιμοποιήσει για να επιτύχει τους προσωπικούς του εκπαιδευτικούς στόχους. Οι στρατηγικές διδασκαλίας που βασίζονται στον Κονεκτιβισμό επιτρέπουν στο διδάσκοντα να αναλάβει ρόλο διευκολυντή προκειμένου οι εκπαιδευόμενοι να συμμετέχουν και να αλληλεπιδρούν ενεργητικά με βάση τους μαθησιακούς στόχους, την προϋπάρχουσα γνώση, τις δεξιότητες και τα κοινά τους ενδιαφέροντα.

Ορισμένες από τις αρχές του κονεκτιβισμού όπως τις διατυπώνει ο Siemens είναι οι ακόλουθες:

- Η μάθηση είναι η διεργασία σχηματισμού δικτύων.
- Η γνώση υπάρχει στο δίκτυο και όχι εσωτερικά μέσα μας.
- Η μάθηση πραγματοποιείται από τη σύνθεση διαφορετικών, πολλές φορές αντίθετων, προσεγγίσεων.

- Το περιεχόμενο είναι συχνά το παραπροϊόν της διεργασίας μάθησης, όχι το σημείο εκκίνησης.
- Οι σύνδεσμοι, όχι το περιεχόμενο, αποτελούν το αρχικό σημείο εκκίνησης της διεργασίας μάθησης.
- Η μάθηση μπορεί να εντοπίζεται σε μη ανθρώπινα στοιχεία.
- Η ικανότητα να διακρίνει κανείς νέες μορφές (πρότυπα) συνδέσεων μεταξύ ιδεών και εννοιών είναι κρίσιμη για τη μάθηση.
- Η επικαιροποίηση της γνώσης είναι η πρόθεση όλων των δραστηριοτήτων μάθησης σύμφωνα με την οπτική του κονεκτιβισμού.
- Η λήψη αποφάσεων είναι από μόνη της μια διεργασία μάθησης

3.5. Μειονεκτήματα - Σκεπτικισμός σχετικά με τα MOOC

Η αλματώδης ανάπτυξη των MOOC, παρ' όλα αυτά, προκάλεσε και αρνητικές κριτικές με αποτέλεσμα να έχουν αρχίσει ήδη να αμφισβητούνται για το κατά πόσο προσφέρουν ουσιαστική εκπαίδευση.

Πνευματικά Δικαιώματα: Πώς διασφαλίζονται τα δικαιώματα πνευματικής ιδιοκτησίας; Είναι γεγονός ότι η πρόσβαση σε τεράστιες πηγές πληροφορίας είναι ένα από τα πλεονεκτήματα των MOOC (το υποστηρικτικό υλικό είναι βίντεο-διαλέξεις, παρουσιάσεις power point, διάφορα σχετικά άρθρα, links και ηλεκτρονικά βιβλία ή βιβλία σε έντυπη μορφή). Ως εκ τούτου πρέπει να βρεθούν μηχανισμοί ώστε το υλικό που διατίθεται να είναι νόμιμο και να εξασφαλίζει τα πνευματικά δικαιώματα των συγγραφέων.

Πιστοποίηση: Ένα ερώτημα που εγείρεται είναι εάν το απλό πιστοποιητικό που παρέχεται είναι αρκετό για όσους συμμετέχουν. Επιπλέον είναι σημαντικό να επιλέγεται σωστά η ενότητα, αφού όλο και περισσότεροι οργανισμοί (αλλά και επιχειρήσεις) συμμετέχουν στη διαδικασία, η εγκυρότητα των οποίων δεν είναι πάντα διασφαλισμένη. Σε όσα MOOC παρέχεται σήμερα κάποιας μορφής πιστοποίηση, δεν λείπουν οι ανησυχίες που σχετίζονται με προβλήματα λογοκλοπής και δικαιώματα πνευματικής ιδιοκτησίας.

Εκπαιδευτική Διαδικασία: Η εκπαιδευτική διαδικασία στα MOOC συντελείται αυτόνομα, πολλές φορές όμως μοιάζει να εξελίσσεται ανεξέλεγκτα από τον διδάσκοντα, γεγονός που οδηγεί σε υποβάθμιση του ρόλου του εκπαιδευτικού. Ο εκπαιδευτικός έρχεται αντιμέτωπος με προβλήματα που συνδέονται είτε άμεσα, είτε έμμεσα με το διδακτικό του έργο (Παγγέ, 2009) αφού οφείλει να διαχειρίζεται αποτελεσματικά τόσο τη διαδικασία μετάδοσης της γνώσης όσο και τη διαχείριση της τάξης, την κινητοποίηση των εκπαιδευομένων και τη διαμόρφωση της προσωπικότητάς τους (Παγγέ 2009, Villegas et al., 2002). Συχνά, τα κριτήρια επιλογής των καθηγητών διαφέρουν από αυτά στη συμβατική εκπαίδευση. Το ερευνητικό έργο λαμβάνεται ελάχιστα υπ' όψιν και εκτιμώνται περισσότερο οι διδακτικές ικανότητες του καθηγητή. Τέλος, αρκετοί ακαδημαϊκοί ανησυχούν ότι τα MOOC και τα συνδεδεμένα με αυτά διαδικτυακά ιδρύματα εν τέλει θα αντικαταστήσουν την

παραδοσιακή πρόσωπο με πρόσωπο επικοινωνία μεταξύ εκπαιδευτικού και εκπαιδευομένου.

Καθοδήγηση: Η πλειοψηφία των MOOC στηρίζεται στη θεωρία του κονεκτιβισμού, δημιουργώντας πολλούς προβληματισμούς σχετικά με τη διδακτική τους εφαρμογή. Σύμφωνα με τον Παγγέ (2009), η ανοιχτή εκπαίδευση ενισχύει την αυτονομία αλλά η αυτονομία δεν πρέπει να συγχέεται με την απομόνωση. Τα χαρακτηριστικά των μαθητών (ηλικία, κουλτούρα, μαθησιακή εμπειρία) καθορίζουν τις απαιτήσεις των εκπαιδευομένων, οι οποίοι επιλέγουν μαθήματα βάσει της διδακτέας ύλης και όχι του διδάσκοντα. «*Το απογοητευτικό στα MOOC είναι ότι ο διδάσκων δεν μπορεί να είναι διαθέσιμος για τους δεκάδες χιλιάδες μαθητές του. Πώς μπορείς να κάνεις ένα μαθητή να νιώσει οικειότητα σε ένα μαζικό μάθημα;*» (Pappano, 2012). Η κοινωνικότητα των συμμετεχόντων σε ένα ανοιχτό μάθημα πρέπει να υποστηρίζεται έντονα και αυτό είναι σπουδαία πρόκληση για τους υπευθύνους του μαθήματος, ειδικά αφού απευθύνονται σε ενήλικες που συχνά είναι περισσότερο επιφυλακτικοί ως προς τις συναναστροφές τους σε σχέση με τα παιδιά (MacKay, 2013). Η έλλειψη της αλληλεπίδρασης με ομότιμους, όμως, είναι βασικό σημείο προβληματισμού.

3.6. Περιγραφή Μαθήματος

Βασικές αρχές προγραμματισμού χρησιμοποιώντας την Python

Τίτλος Μαθήματος	Βασικές αρχές προγραμματισμού χρησιμοποιώντας την Python
Περιγραφή	Μέσα από αυτό το διαδικτυακό μάθημα, οι εκπαιδευόμενοι που θα το παρακολουθήσουν θα κατανοήσουν τις βασικές αρχές προγραμματισμού της Python. Στο τέλος των μαθημάτων θα είναι εξοικειωμένοι με το προγραμματιστικό περιβάλλον της γλώσσας και θα μπορούν να συνθέσουν δικά τους προγράμματα. Κατά τη διάρκεια αυτών των μαθημάτων θα κατανοήσουν έννοιες όπως εντολές ελέγχου, επανάληψης, δομές δεδομένων κλπ., καθώς και θα αποκτήσουν γνώσεις που είναι χρήσιμες και σε άλλες γλώσσες προγραμματισμού (μεταβλητές, σταθερές, τύποι δεδομένων, τελεστές, κ.ά.).
Γνωστικό Πεδίο	Το γνωστικό πεδίο αυτού του μαθήματος είναι η γλώσσα προγραμματισμού Python.
Τύπος Μαθήματος	Το μάθημα ορίζεται ως μη εντατικής φοίτησης και απευθύνεται σε φοιτητές, απόφοιτους ή εργαζόμενους.
Συνολικός Χρόνος Μαθήματος	Συνολική διάρκεια: 6 εβδομάδες (1 ενότητα/εβδομάδα)

Εκπαιδευτικό Πρόβλημα	Η Python είναι μια υψηλού επιπέδου γλώσσα διαδικασιακού προγραμματισμού, με βασικά χαρακτηριστικά την στην απλότητα, την αναγνωσιμότητα του κώδικά της και την ευκολία χρήσης της. Είναι μια γλώσσα ιδανική για αρχάριους, αλλά χρησιμοποιείται και από έμπειρους προγραμματιστές για ανάπτυξη ολοκληρωμένων εφαρμογών. Η σύνταξή της επιτρέπει μικρότερο κώδικα από άλλες γλώσσες προγραμματισμού, όπως η C++ ή η Java, αλλά συντηρείται και πιο εύκολα.
Χαρακτηριστικά και Ανάγκες Εκπαιδευόμενων	Η κύρια ανάγκη των εκπαιδευόμενων που θα παρακολουθήσουν το συγκεκριμένο μάθημα είναι η ευελιξία στο ωράριο, καθώς μπορεί να είναι φοιτητές ή εργαζόμενοι.
Έννοιες	Οι βασικές έννοιες που πρόκειται να διδαχθούν είναι οι εκφράσεις, οι εντολές, ο έλεγχος, η επανάληψη και οι δομές δεδομένων, δηλαδή ο εκπαιδευόμενος θα πρέπει να μελετήσει και να κατανοήσει πλήρως το συντακτικό της γλώσσας, χρησιμοποιώντας το κατάλληλο λεξιλόγιο, έτσι ώστε μέσα από απλές οδηγίες, βήμα-βήμα ο υπολογιστής να εκτελέσει μια σειρά εργασιών. Επίσης βασικές έννοιες, εξίσου σημαντικές, αποτελούν οι έννοιες της μεταβλητής και της σταθεράς, καθώς και τα μέρη που αποτελούν το προγραμματιστικό περιβάλλον.
Ενότητες Μάθησης	Οι ενότητες που θα διδαχθούν είναι: <ol style="list-style-type: none"> 1. Εισαγωγή στην Python 2. Μεταβλητές, Σταθερές, Τύποι δεδομένων 3. Τελεστές 4. Εντολές Ελέγχου 5. Εντολές Επανάληψης 6. Δομές Δεδομένων
Εκπαιδευτικοί Στόχοι	Στόχοι του μαθήματος είναι: <ul style="list-style-type: none"> • Η εκμάθηση της γλώσσας προγραμματισμού Python • Η εξοικείωση με το προγραμματιστικό περιβάλλον • Η εμπέδωση βασικών αρχών προγραμματισμού
Εργαλεία	Το μάθημα χρησιμοποιεί πλούσιο εκπαιδευτικό υλικό. Αναλυτικότερα θα δημιουργηθούν εκπαιδευτικά βίντεο καθώς και εκτενέστερο εκπαιδευτικό υλικό, όπως παρουσιάσεις, υπερκείμενο, δοκίμια και εργασίες.

Απαιτήσεις Περιβάλλοντος	Το μάθημα αυτό σαν μαζικό ανοιχτό διαδικτυακό μάθημα είναι σχεδιασμένο έτσι ώστε να παρέχει γνώση σε εκατοντάδες ή χιλιάδες εκπαιδευόμενους μέσω ανοικτής πρόσβασης. Στόχος λοιπόν είναι να δημιουργηθεί ένα περιβάλλον συνεργατικής μάθησης με τη δημιουργία κοινοτήτων μάθησης. Ένα τέτοιο περιβάλλον μάθησης απαιτεί πρόσθετο πολυμορφικό εκπαιδευτικό υλικό και μηχανισμό υποστήριξης για το ρόλο του διδάσκοντα με στόχο την ενεργητική εμπλοκή των διδασκόμενων στη διεργασία της μάθησης.
Εμπλεκόμενοι Ρόλοι	Κατά τη διεξαγωγή του μαθήματος εμπλέκονται οι ρόλοι του σχεδιαστή της ενότητας, των εκπαιδευτών και των εκπαιδευόμενων.
Χρονοπρογραμματισμός	Προτείνεται ο χρόνος ενασχόλησης των εκπαιδευομένων να είναι 3 έως 4 ώρες την πρώτη έως την τρίτη εβδομάδα του μαθήματος, 5 έως 6 ώρες την τέταρτη και πέμπτη εβδομάδα και περίπου 10 ώρες την έκτη εβδομάδα.

4. ΕΝΟΤΗΤΑ 1 - ΕΙΣΑΓΩΓΗ ΣΤΗ ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ PYTHON

Τίτλος Ενότητας	Εισαγωγή στη γλώσσα προγραμματισμού Python
Τίτλος Μαθήματος	Βασικές αρχές προγραμματισμού χρησιμοποιώντας την Python
Περιγραφή	Στην ενότητα αυτή θα παρουσιαστούν η ιστορία και η φιλοσοφία της Python, το προγραμματιστικό περιβάλλον της και βασικές έννοιες για την σύνταξη προγράμματος σε αυτό το περιβάλλον. Επιπλέον δίνεται και ο ιστότοπος για την εγκατάσταση του περιβάλλοντος που θα προγραμματίζουμε κατά την διάρκεια αυτών των μαθημάτων.
Εκπαιδευτικοί στόχοι	Στόχοι για την ενότητα αυτή είναι να μπορούν οι εκπαιδευόμενοι: <ul style="list-style-type: none"> • Να κατανοούν τη φιλοσοφία της Python • Να μπορούν να χειρίζονται την κονσόλα στην οποία θα γράφουν προγράμματα • Να γνωρίζουν πώς να γράφουν στην κονσόλα (γλώσσα, σημεία στίξης, εισαγωγή σχολίων κλπ.) • Να γνωρίζουν πώς να αποθηκεύουν και να ανακαλούν προγράμματα που ανέπτυξαν.
Μαθησιακά αποτελέσματα	Αφού μελετήσουν την ενότητα αυτή, οι εκπαιδευόμενοι: <ul style="list-style-type: none"> • Θα γνωρίζουν πώς να χειριστούν το προγραμματιστικό περιβάλλον • Θα είναι ικανοί να χρησιμοποιήσουν το περιβάλλον σαν αριθμομηχανή • Θα κατανοούν βασικές αρχές της γλώσσας
Εκπαιδευτικές δραστηριότητες	Οι εκπαιδευτικές δραστηριότητες της ενότητας είναι: <ul style="list-style-type: none"> • Ιστορία και Φιλοσοφία της Python • Εγκατάσταση και Περιβάλλοντα ανάπτυξης • Σύνταξη στο Προγραμματιστικό περιβάλλον
Εμπλεκόμενοι Ρόλοι	Στη διάρκεια αυτής της ενότητας εμπλέκονται οι ρόλοι των σχεδιαστών της ενότητας, των εκπαιδευόμενων.
Αξιολόγηση	Η ενότητα υποστηρίζεται από τρεις υποενότητες. Καθεμία περιλαμβάνει 1-2 μαθησιακά αντικείμενα, το πρώτο για τη θεωρία και τα υπόλοιπα ασκήσεις αυτοαξιολόγησης.
Συνολικός χρόνος ενότητας	1 εβδομάδα

4.1. Ιστορία και Φιλοσοφία της Python

Τίτλος Υποενότητας	Ιστορία και Φιλοσοφία της Python
Τίτλος Ενότητας	Εισαγωγή στη γλώσσα προγραμματισμού Python
Εκπαιδευτική Στρατηγική	Θα χρησιμοποιηθούν θεωρία, ασκήσεις, παραδείγματα και πρακτική εφαρμογή με ασκήσεις αυτοαξιολόγησης.
Περιγραφή	Η υποενότητα αυτή παρουσιάζει την Python για πρώτη φορά στον εκπαιδευόμενο. Διδάσκει τις βασικές αρχές της και γιατί είναι η καλύτερη επιλογή για γλώσσα εκμάθησης για αρχάριους.
Γλώσσα	Ελληνικά
Μαθησιακά Αποτελέσματα	Μέσα από αυτή την υποενότητα οι εκπαιδευόμενοι: <ul style="list-style-type: none">• Θα γνωρίζουν γιατί είναι τόσο ευρέως γνωστή• Θα αναγνωρίζουν τα πλεονεκτήματά της σε σχέση με άλλες γλώσσες• Θα κατανοούν πώς γίνεται η μεταγλώττιση στην Python
Μαθησιακά Αντικείμενα	Ιστορία και Φιλοσοφία της Python
Αξιολόγηση Εκπαιδευόμενων	Η αξιολόγηση των εκπαιδευόμενων θα γίνει με ασκήσεις αυτοαξιολόγησης στο τέλος της ενότητας οι οποίες θα βασίζονται στην θεωρία.
Λέξεις Κλειδιά	Γλώσσα προγραμματισμού, μεταγλωττιστής, συμβατότητα προς τα πίσω, ευκολία χρήσης

4.1.1. Μαθησιακό Αντικείμενο 1 - Γιατί επιλέγουμε την Python

Τίτλος	Γιατί επιλέγουμε την Python
Τίτλος Υποενότητας	Ιστορία και Φιλοσοφία της Python
Περιγραφή	Σε αυτό το μαθησιακό αντικείμενο παρουσιάζεται η ιστορία της γλώσσας που θα μελετήσουμε και τα βασικά χαρακτηριστικά της.
Γλώσσα	Ελληνικά
Μαθησιακός Τύπος	Θεωρία, ασκήσεις και παραδείγματα
Τεχνικός Τύπος	Βιντεοδιάλεξη, παρουσίαση
Λέξεις Κλειδιά	Python, bytecode, εκδόσεις

Μαθησιακά Αποτελέσματα	<p>Όταν ο εκπαιδευόμενος ολοκληρώσει αυτό το μαθησιακό αντικείμενο:</p> <ul style="list-style-type: none"> • Θα κατανοεί βασικές αρχές της γλώσσας • Θα κατανοεί γιατί επιλέγεται για αρχάριους • Θα γνωρίζει από ποιον και πότε δημιουργήθηκε
------------------------	---

ΙΣΤΟΡΙΑ ΚΑΙ ΦΙΛΟΣΟΦΙΑ ΤΗΣ PYTHON

Η **Python** είναι μια υψηλού επιπέδου γλώσσα προγραμματισμού η οποία δημιουργήθηκε από τον Ολλανδό προγραμματιστή Γκβίντο βαν Ρόσσουμ (Guido van Rossum) το 1990. Το όνομα της γλώσσας προέρχεται από το Monty Python's Flying Circus, μια Αγγλική σατυρική σειρά του '70 και όχι από ερπετό.

Τα κύρια χαρακτηριστικά της είναι η αναγνωσιμότητα του κώδικά της και η ευκολία χρήσης της. Για το λόγο αυτό είναι ιδανική για αρχάριους, αλλά χρησιμοποιείται και από έμπειρους προγραμματιστές για ανάπτυξη ολοκληρωμένων εφαρμογών. Ο τρόπος σύνταξής της επιτρέπει στους προγραμματιστές που τη χρησιμοποιούν να εκφράσουν δύσκολες έννοιες σε μικρότερο κώδικα απ' ό,τι θα γινόταν σε άλλες γλώσσες προγραμματισμού όπως η C++ ή η Java. Έτσι όχι μόνο είναι πιο εύκολη στην κατανόηση και σύνταξη, αλλά συντηρείται και πιο εύκολα.

Η φιλοσοφία της Python επικεντρώνεται κυρίως στην απλότητα και αναγνωσιμότητα του κώδικα. Αναπτύσσεται ως λογισμικό ανοικτού κώδικα με βάση το μοντέλο της κοινότητας προγραμματιστών που εργάζονται για την ανάπτυξή της και ο συντονισμός και η διαχείριση γίνεται από τον μη κερδοσκοπικό οργανισμό Python Software Foundation (PSF). Ο κώδικας διανέμεται με την άδεια Python Software Foundation License.

Η Python είναι μια εύκολη στην εκμάθηση γλώσσα προγραμματισμού. Η γλώσσα είναι ιδιαίτερα δημοφιλής για πολλούς λόγους:

- Μπορεί να χρησιμοποιηθεί διαδραστικά όπως και μια αριθμομηχανή. Είναι ένας εύκολος τρόπος να εξοικειωθεί κάποιος με τον προγραμματισμό.
- Αν και γλώσσα υψηλού επιπέδου, η σύνταξη της είναι πολύ απλή.
- Συνδυάζεται εύκολα με άλλες γλώσσες προγραμματισμού όπως Java, C, C++.
- Διατίθεται δωρεάν και υποστηρίζεται από όλα τα λειτουργικά συστήματα.

Ένα πρόγραμμα που γράφεται σε μεταγλωττιζόμενη γλώσσα όπως η C ή η C++ μετατρέπεται σε μια γλώσσα που μιλάει ο υπολογιστής μας (δυαδικός κώδικας) χρησιμοποιώντας έναν μεταγλωττιστή. Η Python, από την άλλη μεριά, δεν χρειάζεται μεταγλώττιση σε δυαδικό αρχείο. Απλά τρέχουμε το πρόγραμμα απευθείας από τον πηγαίο κώδικα. Εσωτερικά, η Python μετατρέπει τον πηγαίο κώδικα σε μια ενδιάμεση μορφή που ονομάζεται bytecode και μετά το μεταφράζει στη γλώσσα του υπολογιστή και το εκτελεί. Όλη αυτή η διαδικασία στην πραγματικότητα κάνει τη χρήση της Python πιο εύκολη, αφού δεν χρειάζεται να ανησυχούμε για τη μεταγλώττιση του προγράμματος, τη σύνδεση με τις κατάλληλες βιβλιοθήκες κλπ. Αυτό επίσης κάνει τα προγράμματα της Python εξαιρετικά

φορητά, αφού μπορούμε να αντιγράψουμε σε έναν άλλο υπολογιστή το πρόγραμμα της Python που δημιουργήσαμε και να δουλέψει έτσι απλά.

- ✓ Η **Python 2.0** κυκλοφόρησε στις 16 Οκτωβρίου 2000. Στις 3 Δεκεμβρίου 2008 κυκλοφόρησε η έκδοση 3.0, γνωστή ως **py3k** ή **python 3000**. Πολλά από τα νέα χαρακτηριστικά αυτής της έκδοσης έχουν μεταφερθεί στις εκδόσεις 2.6 και 2.7, οι οποίες είναι προς τα πίσω συμβατές. Η συμβατότητα προς τα πίσω είναι μια ιδιότητα που επιτρέπει συστήματα και τεχνολογίες να λειτουργούν και να συνδέονται με παλαιότερες εκδόσεις τους.
- ✓ Η **Python 3** είναι η πρώτη γλώσσα προγραμματισμού που σπάει τη συμβατότητα προς τα πίσω με όλες τις προηγούμενες εκδόσεις της ώστε να επιτρέπεται η διόρθωση σε λάθη που υπήρχαν σε προηγούμενες εκδόσεις και να γίνει ακόμα πιο εύκολη η χρήση της.

4.2. Εγκατάσταση και Προγραμματιστικό Περιβάλλον

Τίτλος Υποενότητας	Εγκατάσταση και Προγραμματιστικό Περιβάλλον
Τίτλος Ενότητας	Εισαγωγή στη γλώσσα προγραμματισμού Python
Εκπαιδευτική στρατηγική	Θα χρησιμοποιηθούν θεωρία, ασκήσεις, παραδείγματα και πρακτική εφαρμογή με ασκήσεις αυτοαξιολόγησης.
Περιγραφή	Σε αυτή την υποενότητα οι εκπαιδευόμενοι κατεβάζουν και χρησιμοποιούν για πρώτη φορά το προγραμματιστικό περιβάλλον με το οποίο θα ασχοληθούν κατά τη διάρκεια αυτών των μαθημάτων. Εξοικειώνονται με το περιβάλλον μαθαίνοντας πώς να ανοίγουν και να κλείνουν την κονσόλα.
Γλώσσα	Ελληνικά
Μαθησιακά Αποτελέσματα	Στο τέλος αυτής της υποενότητας οι εκπαιδευόμενοι: <ul style="list-style-type: none"> • Θα έχουν στον υπολογιστή τους το προγραμματιστικό περιβάλλον • Θα εξοικειωθούν με το περιβάλλον ανάπτυξης εφαρμογών
Μαθησιακά Αντικείμενα	Προγραμματιστικό περιβάλλον της Python
Αξιολόγηση Εκπαιδευόμενων	Η αξιολόγηση της υποενότητας γίνεται με ένα (1) μαθησιακό αντικείμενο περιλαμβάνοντας ασκήσεις αυτοαξιολόγησης στο τέλος της ενότητας.
Λέξεις Κλειδιά	IDLE, Περιβάλλοντα ανάπτυξης εφαρμογών

4.2.1. Μαθησιακό Αντικείμενο 2 - Προγραμματιστικό Περιβάλλον

Τίτλος	Προγραμματιστικό Περιβάλλον
Τίτλος Υποενότητας	Εγκατάσταση και Προγραμματιστικό Περιβάλλον
Περιγραφή	Σκοπός αυτού του μαθησιακού αντικειμένου είναι ο εκπαιδευόμενος να κατανοήσει γιατί η χρήση του IDLE είναι ιδανική για αρχάριους, τι προσφέρει το IDLE και τι είναι η χρωματική επισήμανση.
Γλώσσα	Ελληνικά
Μαθησιακός Τύπος	Θεωρία, ασκήσεις και παραδείγματα
Τεχνικός Τύπος	Παρουσίαση, βιντεοδιάλεξη, υπερσύνδεσμος
Λέξεις Κλειδιά	Χρωματική συντακτική επισήμανση, σφάλματα
Μαθησιακά Αποτελέσματα	Όταν ολοκληρώσουν τη μελέτη αυτού του μαθησιακού αντικειμένου οι εκπαιδευόμενοι θα είναι σε θέση: <ul style="list-style-type: none">• Να γνωρίζουν πού να γράψουν τον κώδικα και πού να τον εκτελέσουν• Να κατανοήσουν γιατί η χρήση της κονσόλας IDLE είναι τόσο εύκολη

ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΕΡΙΒΑΛΛΟΝΤΑ ΑΝΑΠΤΥΞΗΣ

Από έναν browser πληκτρολογούμε <https://www.python.org/>. Επιλέγουμε τον σύνδεσμο Downloads και κατεβάζουμε μια έκδοση της Python. Στη συγκεκριμένη περίπτωση επιλέχθηκε η 2.7.11. Όταν κατέβει το αρχείο το ανοίγουμε και το εκτελούμε ακολουθώντας τις οδηγίες ώστε να γίνει η εγκατάσταση.

Η Python προσφέρει ένα περιβάλλον ανάπτυξης εφαρμογών. Το περιβάλλον αυτό ονομάζεται IDLE (Integrated Development Environment). Συνήθως ένα IDLE περιλαμβάνει κάποιον επεξεργαστή πηγαίου κώδικα, έναν μεταγλωττιστή, αποσφαλματωτή, κ.ά. Το IDLE είναι ένα απλό περιβάλλον, ειδικά για αρχάριους προγραμματιστές, στο οποίο μπορούμε να γράψουμε και να εκτελέσουμε τα προγράμματά μας.

- **Κονσόλα του διερμηνευτή της Python (IDLE)** για διαδραστική και άμεση εκτέλεση εντολών, για πληκτρολόγηση εντολής και εκτέλεσής της από τον διερμηνευτή. Κατά τη σύνταξη των εντολών, αυτές χρωματίζονται κατάλληλα (χρωματική συντακτική επισήμανση). Η επισήμανση αυτή βοηθά στη σωστή σύνταξη των εντολών, αφού κάθε μία έχει διαφορετικό χρώμα και είναι εύκολο να ανιχνευθούν σφάλματα, όπως για παράδειγμα η λάθος σύνταξη των εντολών ή λάθη λόγω βιασύνης από την πληκτρολόγηση των εντολών. Το παράθυρο αυτό υποστηρίζει και επανάκληση των προηγούμενων εντολών που έχουμε δώσει. Αυτό γίνεται, αν κάνουμε κλικ με το ποντίκι σε προηγούμενη εντολή που έχουμε γράψει και πατήσουμε Enter.

Μπορούμε να ανοίξουμε την κονσόλα γράφοντας στη γραμμή εντολών των Windows και πληκτρολογώντας «IDLE» ή από το μενού όπου θα γράφει “Python 2.7” και επιλέγοντας το IDLE.

4.3. Σύνταξη στο Προγραμματιστικό Περιβάλλον

Τίτλος Υποενότητας	Σύνταξη στο Προγραμματιστικό Περιβάλλον
Τίτλος Ενότητας	Εισαγωγή στη γλώσσα προγραμματισμού Python
Εκπαιδευτική Στρατηγική	Θα χρησιμοποιηθούν θεωρία, ασκήσεις, παραδείγματα και πρακτική εφαρμογή με ασκήσεις αυτοαξιολόγησης.
Περιγραφή	Σκοπός αυτής της υποενότητας είναι να κατανοήσει ο εκπαιδευόμενος πώς και πού γίνεται η σύνταξη και η αποθήκευση του κώδικα για αυτή τη γλώσσα, καθώς και πώς μπορεί να τη χρησιμοποιήσει ως αριθμομηχανή.
Γλώσσα	Ελληνικά
Μαθησιακά Αποτελέσματα	Στο τέλος αυτής της υποενότητας ο εκπαιδευόμενος: <ul style="list-style-type: none"> • Θα γνωρίζει πώς να αποθηκεύει τον κώδικα που έγραψε. • Θα κατανοεί πώς χρησιμοποιείται η γλώσσα ως αριθμομηχανή • Θα γνωρίζει πώς να γράφει κώδικα (σύνταξη)
Μαθησιακά Αντικείμενα	Σύνταξη στο προγραμματιστικό περιβάλλον
Αξιολόγηση Εκπαιδευόμενων	Η αξιολόγηση της υποενότητας γίνεται με δύο (2) μαθησιακά αντικείμενα με ασκήσεις αυτοαξιολόγησης στο τέλος της ενότητας.
Λέξεις Κλειδιά	Αριθμομηχανή, σύνταξη κώδικα, αποθήκευση

4.3.1. Μαθησιακό Αντικείμενο 3 - Σύνταξη και αποθήκευση στο προγραμματιστικό περιβάλλον

Τίτλος	Σύνταξη και αποθήκευση στο προγραμματιστικό περιβάλλον
Τίτλος Υποενότητας	Σύνταξη στο προγραμματιστικό περιβάλλον
Περιγραφή	Σε αυτό το μαθησιακό αντικείμενο εξηγείται πώς και πού γίνεται η σύνταξη και αποθήκευση στο προγραμματιστικό περιβάλλον της Python, ποια γλώσσα χρησιμοποιείται ποια είναι η σύνταξη κ.ά.
Γλώσσα	Ελληνικά

Μαθησιακός Τύπος	Θεωρία, ασκήσεις και παραδείγματα
Τεχνικός Τύπος	Παρουσίαση, βιντεοδιάλεξη
Λέξεις Κλειδιά	Λατινικοί χαρακτήρες, ελληνικοί χαρακτήρες, σχόλια
Μαθησιακά Αποτελέσματα	Μετά τη μελέτη αυτού του μαθησιακού αντικειμένου, οι εκπαιδευόμενοι: <ul style="list-style-type: none"> • Θα είναι σε θέση να γράψουν κώδικα • Θα γνωρίζουν πώς να αποθηκεύσουν τον κώδικα που έγραψαν • Θα γνωρίζουν πώς να εμφανίζονται οι ελληνικοί χαρακτήρες

ΣΥΝΤΑΞΗ ΣΤΟ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ

Στο τρέχον μάθημα, θα συντάσσεται κώδικας στην Python Shell που ανοίγει γράφοντας IDLE στη γραμμή εντολών. Η κονσόλα κλείνει πατώντας Ctrl-Z και Enter ή γράφοντας την εντολή exit() είτε το παραδοσιακό x στην πάνω δεξιά μεριά του παραθύρου.

Η αποθήκευση του προγράμματος γίνεται ως εξής:

Στην κονσόλα IDLE, επιλέγουμε το File και New File. Αυτό ανοίγει ένα άλλο παράθυρο. Εκεί θα ξεκινήσει η σύνταξη του προγράμματος και τα αποτελέσματα θα εμφανιστούν στην Python Shell (το πρώτο παράθυρο που εμφανίστηκε). Στο νέο παράθυρο αναφέρεται ο τίτλος untitled. Δίνουμε ένα όνομα για να μπορούμε να το βρούμε στη συνέχεια πατώντας File και έπειτα Save As. Δίνουμε ένα όνομα αρχείου, για παράδειγμα myfirstscript.py (πρέπει να υπάρχει πάντα η επέκταση .py), και αποθηκεύουμε σε φάκελο της αρεσκείας μας. Βλέπουμε πως τώρα ο τίτλος δεν είναι untitled, αλλά το όνομα που δώσαμε και η τοποθεσία στον υπολογιστή. Γράφουμε τον κώδικα και επιλέγουμε Run -> Run Module για να εκτελεστεί. Εμφανίζεται μήνυμα για να αποθηκεύσουμε τον κώδικα, οπότε επιλέγουμε OK. Αν είναι σωστός ο κώδικας ανοίγει η Python Shell και εμφανίζει το αποτέλεσμα. Αν υπάρχει συντακτικό λάθος στον κώδικα, η Python επισημαίνει με κόκκινο χρώμα την περιοχή του κώδικα όπου εντοπίστηκε το λάθος.

Το σύμβολο >>> ονομάζεται προτροπή και βοηθά στη σωστή σύνταξη για να είναι πιο εύκολα αναγνώσιμος ο κώδικας.

Η πρώτη εντολή των προγραμματιστών όταν μαθαίνουν μια νέα γλώσσα προγραμματισμού συνήθως είναι print 'Hello World!' όπου εκτυπώνεται η φράση Hello World!

Δοκιμάστε το στον υπολογιστή σας τώρα.

ΣΥΝΤΑΞΗ

Οι εντολές γράφονται με μικρούς λατινικούς χαρακτήρες. Οι υποδιαστολές συμβολίζονται με τελεία (.). Αυτά που γράφονται μέσα στα εισαγωγικά στην εντολή print μπορούν να είναι

οποιαδήποτε γλώσσα, αριθμοί ή σύμβολα, αφού θα τα εκτυπώσει ακριβώς όπως τα γράψαμε.

Για να εμφανίζονται τα μηνύματα που γράφουμε στα ελληνικά πρέπει στην αρχή του προγράμματος να γράψουμε την εντολή: **#-*- coding:utf-8 -*-**

Αντίστοιχα γράφουμε άλλες εντολές για άλλες γλώσσες εκτός της αγγλικής.

Η εισαγωγή σχολίων γίνεται με τη χρήση του συμβόλου της δίεσης (#). Το σχόλιο μπορεί να μπει οπουδήποτε στον κώδικα, αλλά μέσα στην εντολή print θα εμφανιστεί ακριβώς όπως γράφηκε.

Για παράδειγμα:

Η εντολή print '# example of comments' θα εκτυπώσει:

```
# example of comments
```

γιατί δεν είναι σχόλιο (η δίεση είναι μέσα στα εισαγωγικά).

Αν στην κονσόλα πληκτρολογήσετε την εντολή help() θα εμφανιστεί ένα κείμενο που θα παρέχει βοήθεια.

4.3.2. Μαθησιακό Αντικείμενο 4 - Η Python ως αριθμομηχανή

Τίτλος	Η Python ως αριθμομηχανή
Τίτλος Υποενότητας	Σύνταξη στο προγραμματιστικό περιβάλλον
Περιγραφή	Σε αυτό το μαθησιακό αντικείμενο εξηγείται πώς μπορεί η γλώσσα Python να χρησιμοποιηθεί διαδραστικά ως αριθμομηχανή. Αυτό σημαίνει ότι δεν χρειάζεται να έχει γνώσεις προγραμματισμού για να κάνει αριθμητικές πράξεις.
Γλώσσα	Ελληνικά
Μαθησιακός Τύπος	Θεωρία, ασκήσεις και παραδείγματα
Τεχνικός Τύπος	Παρουσίαση, βιντεοδιάλεξη
Λέξεις Κλειδιά	Αριθμομηχανή, σύμβολα, εντολή print, διαίρεση
Μαθησιακά Αποτελέσματα	Όταν ολοκληρώσει τη μελέτη αυτού του μαθησιακού αντικειμένου, ο εκπαιδευόμενος: <ul style="list-style-type: none">• Θα είναι ικανός να κάνει απλές και σύνθετες πράξεις με τη χρήση της Python• Θα κατανοεί την ιδιαιτερότητα της διαίρεσης ως αριθμητικής πράξης

ΑΡΙΘΜΟΜΗΧΑΝΗ

Η Python μπορεί να χρησιμοποιηθεί ως υπολογιστική μηχανή, όπου οι βασικές πράξεις γίνονται με τα σύμβολα +, -, *, /. Έτσι μπορούμε να εκτελέσουμε απλές πράξεις αλλά και πιο σύνθετες με τη χρήση παρενθέσεων. Κάθε φορά που θέλουμε το αποτέλεσμα μιας πράξης αρκεί να γράψουμε την εντολή "print" και την πράξη που θέλουμε. Επομένως, αν γράψουμε `print (5+5)` στην επόμενη γραμμή θα εμφανιστεί το αποτέλεσμα 10. Οι ακέραιοι αριθμοί στην Python έχουν απεριόριστο μέγεθος.

Μπορούμε επίσης να τυπώσουμε πολλά αποτελέσματα μαζί αρκεί να χωρίζονται με κόμματα οι πράξεις. Για παράδειγμα στην εντολή `print(7+3, 10-4, 4*2,3/3)` στην επόμενη γραμμή θα εμφανιστεί το αποτέλεσμα όλων των πράξεων: 10,6,8,1

Επειδή δεν έχουμε ορίσει ακόμα τους τύπους των δεδομένων, όταν κάνουμε μια πράξη, όπως για παράδειγμα διαίρεση, όπου μπορεί να μην έχουμε ακέραιο αποτέλεσμα, τότε:

Αν γράψουμε `print 6/5` θα δώσει αποτέλεσμα 1, δηλαδή το ακέραιο ηλίκο.

Αν όμως γράψουμε `print 6/5.0` θα εμφανιστεί 1.2.

Σε αυτή την περίπτωση επειδή έχουμε διαίρεση ακεραίου αριθμού με δεκαδικό αριθμό θα εμφανιστεί δεκαδικό ηλίκο.

4.4. Ασκήσεις 1ης Ενότητας

Επιλέξτε τη σωστή απάντηση (multiple choice)

1. Η Python δημιουργήθηκε από:
 - a) James Gosling
 - b) Bjarne Stroustrup
 - c) Guido van Rossum
2. Η προτροπή συμβολίζεται με:
 - a) >
 - b) >>
 - c) >>>
3. Τα σχόλια συμβολίζονται με:
 - a) #
 - b) * *
 - c) */
4. Με εντολή `print (5+5)-5` εκτυπώνεται:
 - a) 0
 - b) 5

c) Τίποτα, δεν μπορώ να κάνω πράξεις.

5. Με την εντολή `print 6*3, 9/5, 8+9` εκτυπώνεται:

a) 18, 1, 17

b) 18, 1.8, 17

c) error

Σωστό ή Λάθος

1. Υπάρχουν πολλές εκδόσεις της Python
2. Η Python δεν είναι γλώσσα υψηλού επιπέδου
3. Η Python δεν είναι συμβατή προς τα πίσω
4. Η Python δεν υποστηρίζεται από όλα τα λειτουργικά συστήματα
5. Η Python χρειάζεται μεταγλώττιση σε δυαδικό κώδικα
6. Η Python είναι γλώσσα ανοιχτού κώδικα
7. Για να κατεβάσουμε την Python πρέπει να πληρώσουμε
8. Τα αρχικά IDLE είναι από (Integrated DeveLopment Environment)
9. Το σύμβολο (`>>`) ονομάζεται προτροπή
10. Η υποδιαστολή συμβολίζεται με (`.`)
11. Μπορώ να γράψω εντολές και με ελληνικούς χαρακτήρες
12. Χρειάζεται εντολή στην αρχή του κώδικα για να εμφανίζονται ελληνικοί χαρακτήρες
13. Η εντολή `print 8/5` μου βγάζει 1.6
14. Η χρωματική συντακτική επισήμανση δυσκολεύει τον προγραμματιστή
15. Οι εντολές μπορούν να γράφονται και με κεφαλαία
16. Οι εντολές πρέπει να είναι πάντα στα αγγλικά
17. Η Python μπορεί να χρησιμοποιηθεί σαν αριθμομηχανή
18. Τα σχόλια μπορούν να μπουν παντού μέσα στον κώδικα
19. Μόλις ανοίξουμε την κονσόλα IDLE μπορούμε να αρχίσουμε να γράφουμε κώδικα
20. Η Python χρησιμοποιείται μόνο από αρχάριους.

5. ΕΝΟΤΗΤΑ 2 - ΜΕΤΑΒΛΗΤΕΣ, ΣΤΑΘΕΡΕΣ, ΤΥΠΟΙ

ΔΕΔΟΜΕΝΩΝ

Τίτλος Ενότητας	Μεταβλητές, Σταθερές, Τύποι Δεδομένων
Τίτλος Μαθήματος	Βασικές αρχές προγραμματισμού χρησιμοποιώντας την Python
Περιγραφή	Στην ενότητα αυτή θα παρουσιαστούν οι διαφορετικοί τύποι δεδομένων θα και εξηγηθούν οι έννοιες των μεταβλητών και των σταθερών συντελεστών. Αυτές οι γνώσεις μπορούν να εφαρμοστούν και σε άλλες γλώσσες προγραμματισμού. Ακόμα παρουσιάζονται και δίνονται παραδείγματα των εντολών εισόδου και εξόδου.
Εκπαιδευτικοί στόχοι	Στόχοι για την ενότητα αυτή είναι να μπορούν οι εκπαιδευόμενοι: <ul style="list-style-type: none">• Να κατανοούν τους διαφορετικούς τύπους δεδομένων• Να κατανέμουν κάθε στοιχείο στην κατάλληλη κατηγορία δεδομένων• Να γνωρίζουν τη διαφορά μεταξύ των μεταβλητών και των σταθερών συντελεστών, καθώς και τι είναι το καθένα• Να εισάγουν δεδομένα από τον χρήστη στον κώδικα• Να εκτυπώσουν τα αποτελέσματα που θέλουν στην οθόνη
Μαθησιακά αποτελέσματα	Αφού μελετήσουν την ενότητα αυτή, οι εκπαιδευόμενοι: <ul style="list-style-type: none">• Θα γνωρίζουν πώς να εισάγουν και να εξάγουν πληροφορίες από και προς τον χρήστη• Θα είναι ικανοί να διαφοροποιήσουν τους τύπους δεδομένων• Θα κατανοούν τις μεταβλητές και τους σταθερούς συντελεστές• Θα γνωρίζουν πώς να δώσουν το κατάλληλο όνομα στις μεταβλητές που χρησιμοποιούν

Εκπαιδευτικές δραστηριότητες	Οι εκπαιδευτικές δραστηριότητες της ενότητας είναι οι εξής: <ul style="list-style-type: none"> • Τύποι Δεδομένων • Σταθερές και Μεταβλητές • Εντολές Εισόδου/Εξόδου
Εμπλεκόμενοι Ρόλοι	Στη διάρκεια αυτής της ενότητας εμπλέκονται οι ρόλοι των σχεδιαστών της ενότητας και των εκπαιδευόμενων
Αξιολόγηση	Η αξιολόγηση αποτελεί αναπόσπαστο κομμάτι της μάθησης. Γι' αυτό και σε κάθε ενότητα χρησιμοποιείται και ως μέσο μάθησης. Οι ενότητες του μαθήματος περιλαμβάνουν τρεις (3) υποενότητες και κάθε μια από αυτές έχει από ένα (1) μαθησιακό αντικείμενο στο οποίο παρουσιάζονται ασκήσεις αυτοαξιολόγησης. Τέτοιες μπορεί να είναι ασκήσεις Multiple choice ή ασκήσεις Σωστό-Λάθος.
Συνολικός χρόνος ενότητας	1 εβδομάδα

5.1. Τύποι Δεδομένων

Τίτλος Υποενότητας	Τύποι Δεδομένων
Τίτλος Ενότητας	Μεταβλητές, Σταθερές, Τύποι Δεδομένων
Εκπαιδευτική Στρατηγική	Θα χρησιμοποιηθούν θεωρία, ασκήσεις, παραδείγματα και πρακτική εφαρμογή με ασκήσεις αυτοαξιολόγησης.
Περιγραφή	Η υποενότητα αυτή παρουσιάζει τους διαφορετικούς τύπους δεδομένων που χρησιμοποιούνται όχι μόνο στην Ρυθση αλλά και σε όλες τις γλώσσες προγραμματισμού.
Γλώσσα	Ελληνικά
Μαθησιακά Αποτελέσματα	Μέσα από αυτή την υποενότητα οι εκπαιδευόμενοι: <ul style="list-style-type: none"> • Θα γνωρίζουν πώς να ξεχωρίζουν τους διάφορους τύπους δεδομένων • Θα γνωρίζουν πώς να ζητήσουν βοήθεια σε περίπτωση που δεν γνωρίζουν τον τύπο
Μαθησιακά Αντικείμενα	Τύποι Δεδομένων
Αξιολόγηση Εκπαιδευόμενων	Η αξιολόγηση της υποενότητας γίνεται με ένα (1) μαθησιακό αντικείμενο με ασκήσεις αυτοαξιολόγησης στο τέλος της ενότητας οι οποίες θα βασίζονται στην θεωρία.
Λέξεις Κλειδιά	Ακέραιοι, δεκαδικοί, ακολουθίες χαρακτήρων

5.1.1. Μαθησιακό Αντικείμενο 5 - Τύποι Δεδομένων

Τίτλος	Τύποι Δεδομένων
Τίτλος Υποενότητας	Τύποι Δεδομένων
Περιγραφή	Σε αυτό το μαθησιακό αντικείμενο παρουσιάζονται οι διαφορετικοί τύποι δεδομένων που χρησιμοποιούνται σε όλες τις γλώσσες προγραμματισμού.
Γλώσσα	Ελληνικά
Μαθησιακός Τύπος	Θεωρία, ασκήσεις και παραδείγματα
Τεχνικός Τύπος	Παρουσίαση, βιντεοδιάλεξη
Λέξεις Κλειδιά	str, float, int
Μαθησιακά Αποτελέσματα	Όταν ο εκπαιδευόμενος ολοκληρώσει τη μελέτη αυτού του μαθησιακού αντικειμένου: <ul style="list-style-type: none">• Θα γνωρίζει τους διαφορετικούς τύπους δεδομένων

ΤΥΠΟΙ ΔΕΔΟΜΕΝΩΝ

Είναι πολύ σημαντικό για την εκμάθηση οποιασδήποτε γλώσσας προγραμματισμού να καταλάβουμε τους βασικούς τύπους δεδομένων, ώστε να μπορούμε να γράψουμε σωστά τον κώδικα. Οι τύποι δεδομένων στην Python χωρίζονται στους:

- Ακέραιους αριθμούς που συμβολίζονται με «**int**»
- Δεκαδικούς αριθμούς που συμβολίζονται με «**float**»
- Ακολουθίες χαρακτήρων που συμβολίζονται με «**str**»
- Λογικούς που συμβολίζονται με «**bool**» και έχουν μόνο δύο εκδοχές, αληθή (True) και ψευδή (False).

Παράδειγμα:

2

3.14

Hello World!

True

Οι τιμές αυτές ανήκουν σε διαφορετικό τύπο δεδομένων. Ο αριθμός 2 είναι ακέραιος, το 3.14 είναι δεκαδικός και το Hello World! μια συμβολοσειρά (ακολουθία από χαρακτήρες), ενώ το True συμβολίζει ότι η μεταβλητή είναι λογικού τύπου και είναι αληθής. Για να τυπώσουμε τις τιμές χρησιμοποιούμε την εντολή "print" με την τιμή που θέλουμε να εκτυπώσουμε μέσα σε παρένθεση. Αν η τιμή είναι συμβολοσειρά, τότε οι χαρακτήρες

περικλείονται από μονά ή διπλά εισαγωγικά. Αν βάλουμε αριθμούς μέσα στα εισαγωγικά, τότε εκλαμβάνονται ως συμβολοσειρές και όχι ως αριθμοί.

Αν θέλουμε να εκτυπώσουμε μια συμβολοσειρά σε δύο γραμμές αντί για μια, τότε μέσα στα εισαγωγικά που γράφουμε την τιμή χρησιμοποιούμε το \n στο σημείο που θέλουμε να αλλάξουμε γραμμή. Για παράδειγμα, έστω ότι θέλω να εκτυπώσω:

```
Hello
```

```
World!
```

Τότε η εντολή που θα γράψω θα είναι: `print 'Hello \n World!'`

Αν θέλω να εκτυπώνεται η έκφραση:

«Η επανάληψη είναι

η μητέρα της μαθήσεως»

θα χρησιμοποιήσω την εντολή `print 'Η επανάληψη είναι \n η μητέρα της μαθήσεως'`

Αν δεν είμαστε σίγουροι σε ποια κατηγορία ανήκει η τιμή που θέλουμε να εισάγουμε, τότε μπορούμε να ζητήσουμε βοήθεια. Πληκτρολογούμε την εντολή `"type"` και μέσα σε παρένθεση γράφουμε την τιμή για την οποία είμαστε αβέβαιοι. Ως αποτέλεσμα από κάτω ακριβώς θα εμφανιστεί το μήνυμα `<class 'ο τύπος που ανήκει η τιμή '>`.

Όταν γράφουμε στον κώδικα δεκαδικούς αριθμούς, η υποδιαστολή πρέπει να είναι με τελεία (.) και όχι με κόμμα (,) γιατί, για παράδειγμα με `print` θα εμφανιστεί `3 <κενό> 14` και όχι `3,14`.

Στην Python δεν είναι αναγκαίο να δηλώνουμε με ακρίβεια τους τύπους δεδομένων που χρησιμοποιούμε. Ο έλεγχος γίνεται καθώς εκτελείται το πρόγραμμα. Η ιδιότητα αυτή ονομάζεται δυναμική απόδοση τύπων (`dynamic typing`).

5.2. ΥΠΟΕΝΟΤΗΤΑ 2 - Σταθερές και Μεταβλητές

Τίτλος Υποενότητας	Σταθερές και Μεταβλητές
Τίτλος Ενότητας	Μεταβλητές, Σταθερές, Τύποι Δεδομένων
Εκπαιδευτική στρατηγική	Θα χρησιμοποιηθούν θεωρία, ασκήσεις, παραδείγματα και πρακτική εφαρμογή με ασκήσεις αυτοαξιολόγησης
Περιγραφή	Σε αυτή την υποενότητα οι εκπαιδευόμενοι κατανοούν τις έννοιες των σταθερών και των μεταβλητών συντελεστών, καθώς και πώς εκχωρούνται στον κώδικα. Επίσης δίνονται κάποιοι «κανόνες» για τη δημιουργία μεταβλητών.
Γλώσσα	Ελληνικά

Μαθησιακά Αποτελέσματα	Στο τέλος αυτής της υποενότητας οι εκπαιδευόμενοι: <ul style="list-style-type: none"> • Θα κατανοούν τις έννοιες των σταθερών και των μεταβλητών συντελεστών • Θα γνωρίζουν πώς να εκχωρούν μεταβλητές • Θα γνωρίζουν πώς να ονομάζουν μεταβλητές οι οποίες να είναι έγκυρες για το προγραμματιστικό περιβάλλον
Μαθησιακά Αντικείμενα	Σταθερές, μεταβλητές, εντολή εκχώρησης, κανόνες για τη δημιουργία μεταβλητών
Αξιολόγηση Εκπαιδευόμενων	Η αξιολόγηση της υποενότητας γίνεται με ένα (1) μαθησιακό αντικείμενο με ασκήσεις αυτοαξιολόγησης στο τέλος της ενότητας οι οποίες θα βασίζονται στην θεωρία.
Λέξεις Κλειδιά	Σταθερές, μεταβλητές, εντολές εκχώρησης, =, λέξεις κλειδιά

5.2.1. Μαθησιακό Αντικείμενο 6 - Σταθερές και Μεταβλητές

Τίτλος	Σταθερές και Μεταβλητές
Τίτλος Υποενότητας	Σταθερές και Μεταβλητές
Περιγραφή	Σκοπός αυτού του μαθησιακού αντικειμένου είναι να κατανοήσει ο εκπαιδευόμενος τις έννοιες των σταθερών και μεταβλητών και πώς θα εφαρμόσουν αυτές τις γνώσεις στη σύνταξη του κώδικα. Επίσης δίνονται κανόνες για τη δημιουργία έγκυρων μεταβλητών.
Γλώσσα	Ελληνικά
Μαθησιακός Τύπος	Θεωρία, ασκήσεις και παραδείγματα
Τεχνικός Τύπος	Κείμενο, παρουσίαση, βιντεοδιάλεξη
Λέξεις Κλειδιά	Σταθερές, μεταβλητές, εντολές εκχώρησης
Μαθησιακά Αποτελέσματα	Μόλις ολοκληρώσουν αυτό το μαθησιακό αντικείμενο οι εκπαιδευόμενοι θα είναι σε θέση: <ul style="list-style-type: none"> • Να χρησιμοποιούν σωστά τις μεταβλητές και τις σταθερές στα προγράμματά τους • Να γνωρίζουν τη σύνταξη μιας μεταβλητής • Να κατανοούν τα λάθη που πρέπει να αποφύγουν στη χρήση των μεταβλητών

ΣΤΑΘΕΡΕΣ ΚΑΙ ΜΕΤΑΒΛΗΤΕΣ

Σταθερά είναι ένα όνομα που δίνουμε στα μεγέθη που οι τιμές τους δεν αλλάζουν κατά τη διάρκεια εκτέλεσης ενός προγράμματος. Οι σταθερές είναι δυνατό να ανήκουν σε οποιονδήποτε από τους τέσσερις τύπων δεδομένων.

Μεταβλητή είναι ένα όνομα που αναφέρεται σε μία τιμή. Οι τιμές τους είναι δυνατό να αλλάζουν κατά τη διάρκεια εκτέλεσης ενός προγράμματος, εξ ου και η ονομασία “μεταβλητή”. Ενώ η τιμή της μεταβλητής μπορεί να αλλάζει κατά την εκτέλεση του προγράμματος, ο τύπος και το όνομα της μεταβλητής παραμένουν ίδια. Οι μεταβλητές μπορούν να ανήκουν σε οποιονδήποτε από τους τύπους δεδομένων. Οι μεταβλητές είναι τμήματα της μνήμης του υπολογιστή μας όπου μπορούμε να αποθηκεύσουμε πληροφορία.

Η **εντολή εκχώρησης** (assignment statement) δημιουργεί νέες μεταβλητές και τους δίνει τιμές. Ο τύπος που ορίζει μια μεταβλητή είναι ίδιος με τον τύπο της τιμής που θέλουμε να της εκχωρήσουμε στο πρόγραμμά μας.

Μια εντολή εκχώρησης αποτελείται από το αριστερό μέρος όπου είναι το όνομα της μεταβλητής, το ίσον (=) και το δεξί μέρος όπου είναι μια τιμή ή μια άλλη μεταβλητή ή μια έκφραση. Σε περίπτωση που είναι άλλη μεταβλητή, εκχωρείται η τιμή της μεταβλητής.

Θα ήταν συνετό να επιλέγουμε ονόματα μεταβλητών όπου καταλαβαίνουμε τι τιμή έχει εκχωρηθεί.

Για παράδειγμα, έστω ότι θέλω να εισαγάγω στο πρόγραμμα και να εκτυπώσω τις τιμές Όνομα, Αριθμός Μητρώου και Βαθμολογία μια φοιτήτριας.

Τότε:

```
name='Maria'  
am=1030  
grade=5.6  
print name, am, grade
```

ΠΡΟΣΟΧΗ: Στην εντολή print, θα πρέπει να γράψετε τις μεταβλητές όπως ακριβώς τις δηλώσατε. Αν δηλαδή προσπαθήσω να τρέξω την εντολή print Name με κεφαλαίο N, τότε θα βγάλει μήνυμα ότι δεν υπάρχει αυτή η μεταβλητή.

Επίσης, να σημειωθεί ότι οι τιμές των μεταβλητών θα εμφανιστούν με τη σειρά που θα τις γράψουμε στην εντολή print. Στο παράδειγμα που αναφέραμε, θα εμφανιστούν με τη σειρά το όνομα, ο αριθμός μητρώου και η βαθμολογία.

Υπάρχουν κάποιοι κανόνες για τη δημιουργία του ονόματος μιας μεταβλητής, που αν δεν τους ακολουθήσουμε θα εμφανιστεί μήνυμα ότι υπάρχει συντακτικό λάθος.

- Ένα όνομα μπορεί να έχει όσους χαρακτήρες θέλουμε, μπορεί να περιέχει αλφαριθμητικούς χαρακτήρες, αλλά πρέπει πάντα να ξεκινάει με ένα γράμμα ή τον χαρακτήρα _ (underscore / κάτω παύλα).

- Τα πεζά διακρίνονται από τα κεφαλαία γράμματα (case sensitive), για παράδειγμα όπως αναφέραμε παραπάνω το name και το Name είναι δύο διαφορετικές μεταβλητές.
- Δεν επιτρέπονται σύμβολα (εισαγωγικά, τελείες, κόμματα και άλλα) και κενά.
- Δεν πρέπει να χρησιμοποιούνται κάποιες από τις λέξεις - κλειδιά (keywords) της Python. Οι λέξεις - κλειδιά συνδέονται με τους κανόνες και τη δομή της γλώσσας και δεν μπορούν να χρησιμοποιηθούν ως ονόματα μεταβλητών.

Η Python 2.7 έχει τις παρακάτω λέξεις κλειδιά:

and, del, from, not, while, as, elif, global, or, with, assert, else, if, pass, yield, break, except, import, print, class, exec, in, raise, continue, finally, is, return, def, for, lambda, try.

Παράδειγμα

Το παρακάτω πρόγραμμα εμφανίζει ένα ονοματεπώνυμο, μία ηλικία και μία διεύθυνση.

```
name='Γιώργος'
age=23
address='Κορίνθου 220'
print name, age, address
```

5.3. Εντολές Εισόδου/Εξόδου

Τίτλος Υποενότητας	Εντολές Εισόδου/Εξόδου
Τίτλος Ενότητας	Σταθερές, Μεταβλητές, Τύποι Δεδομένων
Εκπαιδευτική Στρατηγική	Θα χρησιμοποιηθούν θεωρία, ασκήσεις, παραδείγματα και πρακτική εφαρμογή με ασκήσεις αυτοαξιολόγησης.
Περιγραφή	Σκοπός αυτής της υποενότητας είναι να κατανοήσει ο εκπαιδευόμενος πώς γίνεται η εισαγωγή δεδομένων από τον χρήστη στο πρόγραμμα και πώς γίνεται η εκτύπωση ώστε να βλέπει ο χρήστης το αποτέλεσμα.
Γλώσσα	Ελληνικά
Μαθησιακά Αποτελέσματα	Στο τέλος αυτής της υποενότητας ο εκπαιδευόμενος: <ul style="list-style-type: none"> • Θα γνωρίζει πώς να εισάγει στο πρόγραμμα δεδομένα από τον χρήστη. • Θα γνωρίζει πώς να εκτυπώνει τα αποτελέσματα μαζί με ένα μήνυμα.
Μαθησιακά Αντικείμενα	Εντολές Εισόδου/Εξόδου

Αξιολόγηση Εκπαιδευόμενων	Η αξιολόγηση της υποενότητας γίνεται με ένα (1) μαθησιακό αντικείμενο με ασκήσεις αυτοαξιολόγησης στο τέλος της ενότητας οι οποίες θα βασίζονται στην θεωρία.
Λέξεις Κλειδιά	Εντολές Εισόδου/Εξόδου, μεταβλητές, μηνύματα

5.3.1. Μαθησιακό Αντικείμενο 7 - Εντολές Εισόδου/Εξόδου

Τίτλος	Εντολές Εισόδου/Εξόδου
Τίτλος Υποενότητας	Σταθερές, Μεταβλητές, Τύποι Δεδομένων
Περιγραφή	Σε αυτό το μαθησιακό αντικείμενο εξηγούνται οι εντολές που χρησιμοποιούμε για την εισαγωγή δεδομένων στο πρόγραμμα από τον χρήστη, καθώς και η εκτύπωσή τους με τα κατάλληλα μηνύματα.
Γλώσσα	Ελληνικά
Μαθησιακός Τύπος	Θεωρία, ασκήσεις και παραδείγματα
Τεχνικός Τύπος	Κείμενο, παρουσίαση, βιντεοδιάλεξη
Λέξεις Κλειδιά	raw_input, print, μηνύματα
Μαθησιακά Αποτελέσματα	Μετά από τη μελέτη αυτού του μαθησιακού αντικειμένου, οι εκπαιδευόμενοι: <ul style="list-style-type: none"> • Θα είναι σε θέση να εισάγουν στο πρόγραμμα μεταβλητές από τον χρήστη • Θα γνωρίζουν πώς να εκτυπώσουν μηνύματα μαζί με τα αποτελέσματα ώστε να μπορεί ο χρήστης να καταλαβαίνει σε τι αναφέρεται το αποτέλεσμα.

ΕΝΤΟΛΕΣ ΕΙΣΟΔΟΥ/ΕΞΟΔΟΥ

Εντολές Εισόδου

Ένα πρόγραμμα μπορεί να πάρει τιμές για τις μεταβλητές της από τον χρήστη. Τα δεδομένα εισάγονται κατά τη διάρκεια της εκτέλεσης του προγράμματος από μια μονάδα εισόδου (π.χ. το πληκτρολόγιο).

Για την είσοδο στοιχείων που δεν **θέλουμε να τροποποιήσουμε** (θα εκτυπωθούν στον χρήστη δηλαδή όπως δόθηκαν) θα χρησιμοποιήσουμε την εντολή `raw_input()`. Στην παρένθεση της εντολής μπορούμε μέσα σε εισαγωγικά να βάλουμε ένα μήνυμα που θα εμφανίζεται στον χρήστη πριν την εισαγωγή των στοιχείων.

Για την είσοδο στοιχείων που παρακάτω στον κώδικα θα **τροποποιηθούν**, τότε χρησιμοποιούμε την εντολή `input()`.

Για παράδειγμα, για να εισάγω στη μεταβλητή name το όνομα του φοιτητή θα γράψω το εξής:

```
name=raw_input('Δώσε το όνομα σου')
print name
```

Σε αυτό το σημείο, το πρόγραμμα που τρέχει θα σταματήσει, θα εμφανίσει το μήνυμα και θα περιμένει από τον χρήστη να εισάγει την τιμή της μεταβλητής και να πατήσει Enter. Μετά θα καταχωρήσει στη μεταβλητή name την τιμή που έδωσε ο χρήστης και θα την εκτυπώσει.

Έστω ότι θέλω ο χρήστης να μου δώσει βαθμολογίες για να υπολογίσω τον μέσο όρο. Επειδή τα δεδομένα που θα εκχωρήσουμε στις μεταβλητές θα τροποποιηθούν χρησιμοποιούμε την εντολή input:

```
grade=input('Δώσε πρώτη βαθμολογία')
grade2=input('Δώσε δεύτερη βαθμολογία')
mesos_oros=(grade1+grade2)/2.0
print 'Ο μέσος όρος των βαθμών είναι: ', mesos_oros
```

Εντολές Εξόδου

Τα δεδομένα που εισήχθησαν στο πρόγραμμα επεξεργάζονται και εξάγουν τα αποτελέσματα από μία μονάδα εξόδου (π.χ. την οθόνη).

Μία εντολή εξόδου είναι η “print μεταβλητή ή μήνυμα”. Έχουμε αναφερθεί παραπάνω για το πώς χρησιμοποιούμε την εντολή εξόδου. Μπορούμε να συνδυάσουμε εκτύπωση μεταβλητών και μηνυμάτων. Για παράδειγμα αν γράψουμε:

```
print 'Το όνομα του φοιτητή είναι: ',name, ', με αριθμό μητρώου ',am, ' και βαθμό ', grade)
```

θα εμφανιστεί στην οθόνη:

Το όνομα του φοιτητή είναι: Μαρία, με αριθμό μητρώου 1030 και βαθμό 5.6

Παράδειγμα

Με το παρακάτω πρόγραμμα ζητάμε από τον χρήστη 2 αριθμούς και εκτυπώνουμε το μέσο όρο τους.

```
number1=input('Δώσε τον πρώτο αριθμό')
number2=input('Δώσε τον δεύτερο αριθμό')
athroisma=number1+number2
mesos_oros=athroisma/2.0
print mesos_oros
```

5.4. Ασκήσεις 2ης Ενότητας

Multiple Choice

1. Οι ακολουθίες χαρακτήρων συμβολίζονται με:
 - a) int
 - b) float
 - c) str

2. Αν θέλω να αλλάξω γραμμή στην εκτύπωση θα προσθέσω:
 - a) /n
 - b) \n
 - c) \r

3. Σε μια μεταβλητή μπορούμε να εκχωρήσουμε:
 - a) Τιμή
 - b) Άλλη μεταβλητή
 - c) Συμβολοσειρά
 - d) Όλα τα παραπάνω

4. Τι θα εκτυπωθεί με την εντολή `print 'abc'+123'`;
 - a) abc123
 - b) abc+123
 - c) syntax error

5. Η εντολή εκχώρησης συμβολίζεται με:
 - a) =
 - b) ==
 - c) <-

Σωστό ή Λάθος

1. Ο αριθμός 3.14 συμβολίζεται με int
2. Η εντολή `print Hello World!` είναι σωστή
3. Dynamic typing ονομάζεται η ιδιότητα να δίνουμε τύπους στις μεταβλητές
4. Οι τιμές των μεταβλητών μπορούν να αλλάζουν στον κώδικα
5. Στο παρακάτω πρόγραμμα πρώτα θα εκτυπωθεί το όνομα και μετά ο AM.

```
name='Δημήτρης'  
am=1177  
print name, am
```
6. Η μεταβλητή με όνομα 2016 είναι έγκυρη

7. Η μεταβλητή με όνομα `_2016` είναι έγκυρη
8. Οι μεταβλητές `name` και `Name` είναι ίδιες
9. Η εντολή εισόδου από τον χρήστη είναι `input()`
10. Ο τύπος μιας μεταβλητής μπορεί να αλλάξει
11. Ο τύπος και το όνομα μιας μεταβλητής δεν αλλάζει ποτέ
12. Οι εντολές εκχώρησης `input()` και `raw_input()` είναι ίδιες
13. Οι μεταβλητές είναι τμήματα της μνήμης του υπολογιστή για αποθήκευση πληροφοριών
14. Η εντολή εκχώρησης χρησιμοποιείται για να δοθεί όνομα σε μια μεταβλητή
15. Μια μεταβλητή μπορεί να ονομαστεί όπως μια λέξη κλειδί

6. ΕΝΟΤΗΤΑ 3 - ΤΕΛΕΣΤΕΣ

Τίτλος Ενότητας	Τελεστές
Τίτλος Μαθήματος	Βασικές αρχές προγραμματισμού χρησιμοποιώντας την Python
Περιγραφή	Στην ενότητα αυτή θα παρουσιαστούν οι διαφορετικοί τύποι τελεστών. Ο στόχος είναι ο εκπαιδευόμενος να κατανοήσει τις έννοιες των τελεστών και των τελεστών που είναι απαραίτητοι για την εκτέλεση πράξεων. Ακόμα θα παρουσιαστεί ο τρόπος εκτέλεσής τους (προτεραιότητα πράξεων)
Εκπαιδευτικοί στόχοι	<p>Στόχοι για την ενότητα αυτή είναι να μπορούν οι εκπαιδευόμενοι:</p> <ul style="list-style-type: none"> • Να κατανοούν τους διαφορετικούς τύπους τελεστών • Να κατανέμουν κάθε τελεστή στην κατάλληλη κατηγορία • Να γνωρίζουν τη διαφορά μεταξύ των τελεστών • Να γνωρίζουν ποια είναι η προτεραιότητα των πράξεων • Να γνωρίζουν τα σύμβολα για κάθε τελεστή
Μαθησιακά αποτελέσματα	<p>Αφού μελετήσουν την ενότητα αυτή, οι εκπαιδευόμενοι:</p> <ul style="list-style-type: none"> • Θα γνωρίζουν ποια είναι τα σύμβολα για κάθε τελεστή και τη σημασία τους • Θα κατανοούν την προτεραιότητα πράξεων • Θα ξεχωρίζουν τους διαφορετικούς τύπους τελεστών
Εκπαιδευτικές δραστηριότητες	<p>Οι εκπαιδευτικές δραστηριότητες της ενότητας είναι οι εξής:</p> <ul style="list-style-type: none"> • Αριθμητικοί Τελεστές • Συγκριτικοί Τελεστές • Λογικοί Τελεστές
Εμπλεκόμενοι Ρόλοι	Στη διάρκεια αυτής της ενότητας εμπλέκονται οι ρόλοι των σχεδιαστών της ενότητας και των εκπαιδευόμενων

Αξιολόγηση	Η ενότητα του μαθήματος αυτού χρησιμοποιεί σε κάθε υποενότητα και σε κάθε μαθησιακό αντικείμενο ασκήσεις αυτοαξιολόγησης. Τέτοιες μπορεί να είναι ασκήσεις Multiple choice ή Σωστό-Λάθος καθώς και ασκήσεις επίλυσης προβλημάτων.
Συνολικός χρόνος ενότητας	1 εβδομάδα

6.1. Αριθμητικοί Τελεστές

Τίτλος Υποενότητας	Αριθμητικοί Τελεστές
Τίτλος Ενότητας	Τελεστές
Εκπαιδευτική Στρατηγική	Θα χρησιμοποιηθούν θεωρία, ασκήσεις, παραδείγματα και πρακτική εφαρμογή με ασκήσεις αυτοαξιολόγησης.
Περιγραφή	Η υποενότητα αυτή παρουσιάζει τις έννοιες των τελεστών και των τελεστέων. Πιο συγκεκριμένα, αναφέρεται στους αριθμητικούς τελεστές και πώς αυτοί συμβολίζονται στην Python. Υπάρχουν λεπτομερείς πίνακες όπου αναφέρεται η ονομασία, το σύμβολο που χρησιμοποιείται, επεξήγηση και παράδειγμα για τον καθένα από τους αριθμητικούς τελεστές
Γλώσσα	Ελληνικά
Μαθησιακά Αποτελέσματα	Μέσα από αυτή την υποενότητα οι εκπαιδευόμενοι: <ul style="list-style-type: none"> • Θα γνωρίζουν ποια είναι τα σύμβολα για κάθε αριθμητικό τελεστή • Θα γνωρίζουν τη σημασία κάθε αριθμητικού τελεστή • Θα είναι ικανοί να δώσουν παραδείγματα χρήσης των τελεστών που αναφέρθηκαν
Μαθησιακά Αντικείμενα	Αριθμητικοί Τελεστές
Αξιολόγηση Εκπαιδευόμενων	Η αξιολόγηση της υποενότητας γίνεται με ένα (1) μαθησιακό αντικείμενο με ασκήσεις αυτοαξιολόγησης στο τέλος της ενότητας οι οποίες θα βασίζονται στην θεωρία.
Λέξεις Κλειδιά	Αριθμητικοί τελεστές, ονομασία, επεξήγηση, παράδειγμα

6.1.1. Μαθησιακό Αντικείμενο 8 - Αριθμητικοί Τελεστές

Τίτλος	Αριθμητικοί Τελεστές
Τίτλος Υποενότητας	Αριθμητικοί Τελεστές
Περιγραφή	Σκοπός αυτού του μαθησιακού αντικειμένου είναι να παρουσιάσει τους αριθμητικούς τελεστές και να δώσει παραδείγματα για το πώς χρησιμοποιούνται σε ένα πρόγραμμα
Γλώσσα	Ελληνικά
Μαθησιακός Τύπος	Θεωρία, ασκήσεις και παραδείγματα
Τεχνικός Τύπος	Παρουσίαση, βιντεοδιάλεξη
Λέξεις Κλειδιά	Συν, πλην, επί, διά
Μαθησιακά Αποτελέσματα	Όταν οι εκπαιδευόμενοι ολοκληρώσουν τη μελέτη αυτού του μαθησιακού αντικειμένου: <ul style="list-style-type: none">• Θα γνωρίζουν τους αριθμητικούς τελεστές και τα σύμβολά τους• Θα κατανοούν πώς να τους χρησιμοποιήσουν σε ένα πρόγραμμα

ΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕΣΤΕΣ

Ένας τελεστής είναι ένα σύμβολο που αναπαριστά μια λειτουργία. Για παράδειγμα, στην έκφραση $5+2$ το $+$ είναι ένας τελεστής. Οι αριθμοί 5 και 2 ονομάζονται τελεστέοι. Οι τελεστές (operators) είναι λειτουργίες που εκτελούν κάποια ενέργεια και μπορούν να αναπαρασταθούν με σύμβολα όπως το $+$ ή με λέξεις - κλειδιά όπως το and.

Οι αριθμητικοί τελεστές είναι τα σύμβολα που χρησιμοποιούνται για να εκτελεστούν αριθμητικές πράξεις. Παραδείγματα αριθμητικών τελεστών φαίνονται στον παρακάτω πίνακα.

ΤΕΛΕΣΤΗΣ	ΟΝΟΜΑΣΙΑ	ΕΠΕΞΗΓΗΣΗ	ΠΑΡΑΔΕΙΓΜΑΤΑ
+	Συν	Πρόσθεση αριθμών ή συμβολοσειρών	$6+5$ $\alpha+\beta$
-	Μείον	Αφαίρεση αριθμών ή συμβολοσειρών	$5-2$ $\alpha-\beta$
*	Επί	Πολλαπλασιασμός αριθμών ή επανάληψη συμβολοσειρών τόσες φορές	$4*3$ $\alpha*3=\alpha\alpha$

/	Διά	Διαίρεση αριθμών Η διαίρεση μεταξύ ακεραίων αριθμών έχει ως αποτέλεσμα ακέραιο πηλίκο. Η διαίρεση μεταξύ ακεραίων και δεκαδικών αριθμών έχει ως αποτέλεσμα δεκαδικό πηλίκο.	$6/5=1$ $6/5.0=1.2$
**	Δύναμη	Ύψωση αριθμού σε μια δύναμη	$2^{**}3=8$
%	Υπόλοιπο	Το υπόλοιπο της ακέραιας διαίρεσης	$10\%3=1$

6.2. Συγκριτικοί Τελεστές

Τίτλος Υποενότητας	Συγκριτικοί Τελεστές
Τίτλος Ενότητας	Τελεστές
Εκπαιδευτική στρατηγική	Θα χρησιμοποιηθούν θεωρία, ασκήσεις, παραδείγματα και πρακτική εφαρμογή με ασκήσεις αυτοαξιολόγησης.
Περιγραφή	Η υποενότητα αυτή παρουσιάζει την έννοια των συγκριτικών τελεστών, τη χρησιμότητά τους και πώς συμβολίζονται στην Python. Υπάρχουν λεπτομερείς πίνακες όπου αναφέρεται η ονομασία, το σύμβολο που χρησιμοποιείται, επεξήγηση και παράδειγμα για τον καθένα από τους συγκριτικούς τελεστές.
Γλώσσα	Ελληνικά
Μαθησιακά Αποτελέσματα	Μέσα από αυτή την υποενότητα οι εκπαιδευόμενοι: <ul style="list-style-type: none"> • Θα γνωρίζουν ποια είναι τα σύμβολα για κάθε συγκριτικό τελεστή • Θα γνωρίζουν τη σημασία κάθε συγκριτικού τελεστή • Θα είναι ικανοί να δώσουν παραδείγματα χρήσης των τελεστών που αναφέρθηκαν
Μαθησιακά Αντικείμενα	Συγκριτικοί Τελεστές
Αξιολόγηση Εκπαιδευόμενων	Η αξιολόγηση της υποενότητας γίνεται με ένα (1) μαθησιακό αντικείμενο με ασκήσεις αυτοαξιολόγησης στο τέλος της ενότητας οι οποίες θα βασίζονται στην θεωρία.
Λέξεις Κλειδιά	Συγκριτικοί τελεστές, ονομασία, επεξήγηση, παράδειγμα

6.2.1. Μαθησιακό Αντικείμενο 9 - Συγκριτικοί Τελεστές

Τίτλος	Συγκριτικοί Τελεστές
Τίτλος Υποενότητας	Τελεστές
Περιγραφή	Σκοπός αυτού του μαθησιακού αντικειμένου είναι να παρουσιάσει τους συγκριτικούς τελεστές και να δώσει παραδείγματα για το πώς χρησιμοποιούνται σε ένα πρόγραμμα.
Γλώσσα	Ελληνικά
Μαθησιακός Τύπος	Θεωρία, ασκήσεις και παραδείγματα
Τεχνικός Τύπος	Παρουσίαση, βιντεοδιάλεξη
Λέξεις Κλειδιά	Μεγαλύτερο, μικρότερο, ίσο, διάφορο
Μαθησιακά Αποτελέσματα	Όταν ολοκληρώσουν τη μελέτη αυτού του μαθησιακού αντικειμένου, οι εκπαιδευόμενοι θα είναι σε θέση: <ul style="list-style-type: none"> • Να γνωρίζουν τους συγκριτικούς τελεστές και τα σύμβολά τους • Να κατανοούν πώς να τους χρησιμοποιούν σε ένα πρόγραμμα

ΣΥΓΚΡΙΤΙΚΟΙ ΤΕΛΕΣΤΕΣ

Οι συγκριτικοί τελεστές είναι τα σύμβολα που χρησιμοποιούνται για να συγκριθούν μεταξύ τους δύο ή παραπάνω αριθμοί ή μεταβλητές. Παραδείγματα συγκριτικών τελεστών φαίνονται στον παρακάτω πίνακα.

ΤΕΛΕΣΤΗΣ	ΟΝΟΜΑΣΙΑ	ΕΠΕΞΗΓΗΣΗ	ΠΑΡΑΔΕΙΓΜΑ
\geq	Μεγαλύτερο ή ίσο	Το πρώτο μέλος είναι μεγαλύτερο ή ίσο του δεύτερου	$5 \geq 5$
$>$	Μεγαλύτερο	Το πρώτο μέλος είναι μεγαλύτερο του δεύτερου	$5 > 2$
\leq	Μικρότερο ή ίσο	Το πρώτο μέλος είναι μικρότερο ή ίσο του δεύτερου	$1 \leq 1$
$<$	Μικρότερο	Το πρώτο μέλος είναι μικρότερο του δεύτερου	$1 < 3$
\neq	Διάφορο	Το πρώτο μέλος είναι διαφορετικό του δεύτερου	$\alpha=3, \beta=4$ άρα $\alpha \neq \beta$
$==$	Ίσο	Το πρώτο μέλος είναι ίσο με το δεύτερο	$\alpha=3, \beta=3$ άρα $\alpha == \beta$

6.3. Λογικοί Τελεστές

Τίτλος Υποενότητας	Λογικοί Τελεστές
Τίτλος Ενότητας	Τελεστές
Εκπαιδευτική Στρατηγική	Θα χρησιμοποιηθούν θεωρία, ασκήσεις, παραδείγματα και πρακτική εφαρμογή με ασκήσεις αυτοαξιολόγησης.
Περιγραφή	Η υποενότητα αυτή παρουσιάζει την έννοια των λογικών τελεστών, τη χρησιμότητά τους και πώς συμβολίζονται στην Ρυθση. Υπάρχουν λεπτομερείς πίνακες όπου αναφέρεται η ονομασία, το σύμβολο που χρησιμοποιείται, επεξήγηση και παράδειγμα για τον καθένα από τους λογικούς τελεστές. Επίσης, θα αναφερθούμε στην προτεραιότητα πράξεων μεταξύ των τελεστών.
Γλώσσα	Ελληνικά
Μαθησιακά Αποτελέσματα	Μέσα από αυτή την υποενότητα οι εκπαιδευόμενοι: <ul style="list-style-type: none"> • Θα γνωρίζουν ποια είναι τα σύμβολα για κάθε λογικό τελεστή • Θα γνωρίζουν τη σημασία κάθε λογικού τελεστή • Θα είναι ικανοί να δώσουν παραδείγματα χρήσης των τελεστών που αναφέρθηκαν • Θα γνωρίζουν την προτεραιότητα με την οποία εκτελούνται οι πράξεις
Μαθησιακά Αντικείμενα	Λογικοί Τελεστές, προτεραιότητα πράξεων
Αξιολόγηση Εκπαιδευόμενων	Η αξιολόγηση της υποενότητας γίνεται με ένα (1) μαθησιακό αντικείμενο με ασκήσεις αυτοαξιολόγησης στο τέλος της ενότητας οι οποίες θα βασίζονται στην θεωρία.
Λέξεις Κλειδιά	Λογικοί τελεστές, ονομασία, επεξήγηση, παράδειγμα, προτεραιότητα πράξεων

6.3.1. Μαθησιακό Αντικείμενο 10 - Λογικοί Τελεστές

Τίτλος	Λογικοί Τελεστές
Τίτλος Υποενότητας	Τελεστές

Περιγραφή	Σκοπός αυτού του μαθησιακού αντικειμένου είναι να παρουσιάσει τους λογικούς τελεστές και να δώσει παραδείγματα για το πώς χρησιμοποιούνται σε ένα πρόγραμμα.
Γλώσσα	Ελληνικά
Μαθησιακός Τύπος	Θεωρία, ασκήσεις και παραδείγματα
Τεχνικός Τύπος	Παρουσίαση, βιντεοδιάλεξη
Λέξεις Κλειδιά	NOT, AND, OR, αληθής, ψευδής
Μαθησιακά Αποτελέσματα	Όταν οι εκπαιδευόμενοι ολοκληρώσουν τη μελέτη αυτού του μαθησιακού αντικειμένου: <ul style="list-style-type: none"> • Θα γνωρίζουν τους λογικούς τελεστές και τα σύμβολά τους • Θα κατανοούν πώς να τους χρησιμοποιήσουν σε ένα πρόγραμμα • Θα γνωρίζουν την προτεραιότητα με την οποία εκτελούνται οι πράξεις

ΛΟΓΙΚΟΙ ΤΕΛΕΣΤΕΣ

Οι λογικοί τελεστές χρησιμοποιούνται σε λογικές εκφράσεις (δηλαδή το αποτέλεσμα τους είναι πάντα ΑΛΗΘΕΣ ή ΨΕΥΔΕΣ). Οι λογικοί τελεστές είναι:

ΤΕΛΕΣΤΗΣ	ΟΝΟΜΑΣΙΑ	ΕΠΕΞΗΓΗΣΗ	ΠΑΡΑΔΕΙΓΜΑ
NOT	NOT X	Αν το X είναι αληθές, τότε επιστρέφει ως αποτέλεσμα ψευδές.	X = αληθές, άρα not X = ψευδές
AND	X AND Y	Αν το X είναι ψευδές τότε δεν θα ελέγξει αν το Y είναι ψευδές ή αληθές, γιατί το αποτέλεσμα θα είναι πάντα ψευδές. Αν το X είναι αληθές, ελέγχει το Y. Αν το Y είναι ψευδές τότε βγάζει αποτέλεσμα ψευδές, αν είναι αληθές βγάζει αληθές.	X = αληθές, Y = ψευδές, τότε X and Y είναι ψευδές X = αληθές, Y = αληθές, τότε X and Y είναι αληθές
OR	X OR Y	Αρκεί ένα από τα δύο να είναι αληθές για να βγάλει αποτέλεσμα αληθές.	X = αληθές, Y = ψευδές, επομένως X or Y είναι αληθές

Σημείωση: Οι μεταβλητές (X, Y) μπορούν να γραφούν είτε με κεφαλαία είτε με πεζά γράμματα. Έχει σημασία όμως αν εκχωρηθεί τιμή σε μεταβλητή με κεφαλαίο, οι πράξεις να γίνουν επίσης με κεφαλαίο. Παράδειγμα: Έστω $X = 5$, τότε `not X = False`.

Επίσης, για να γίνει η πράξη στην Python, στο IDLE πρέπει να γραφεί: `print not X`

Γράφουμε τον τελεστή με πεζά γράμματα, κενό και έπειτα το X όπως το έχουμε δηλώσει στην αρχή. Αν γραφεί NOT παρατηρούμε ότι δεν αλλάζει χρώμα η λέξη, επομένως είναι λάθος, αφού οι δεσμευμένες λέξεις (μεταξύ των οποίων είναι και οι λογικοί τελεστές) αλλάζουν χρώμα.

Παράδειγμα:

```
x=False
```

```
print not x
```

Εδώ θα εμφανιστεί η έκφραση True όταν εκτελεστεί το πρόγραμμα. Βλέπουμε ότι το x το ορίζω με την δεσμευμένη λέξη False. Πρέπει να γραφεί με κεφαλαίο F για να θεωρείται σωστό. Αντίστοιχα το True.

ΠΡΟΤΕΡΑΙΟΤΗΤΑ ΠΡΑΞΕΩΝ

Όταν σε μία έκφραση εμφανίζονται περισσότερα από ένα σύμβολα τελεστών, η σειρά που θα υπολογιστούν εξαρτάται από τους κανόνες προτεραιότητας. Οι κανόνες αυτοί είναι σύμφωνα με τη σειρά εκτέλεσης των αριθμητικών πράξεων που ξέρουμε από τα Μαθηματικά.

Οι παρενθέσεις έχουν τη μεγαλύτερη προτεραιότητα και έτσι χρησιμοποιούνται για να «αναγκάσουν» την Python να αποτιμήσει μία έκφραση σύμφωνα με τη σειρά που θέλουμε. Εκφράσεις σε παρενθέσεις αποτιμώνται πρώτες. Επίσης, παρενθέσεις χρησιμοποιούνται για να κάνουν τις εκφράσεις πιο αναγνώσιμες, χωρίς να αλλάξουν το τελικό αποτέλεσμα. Η ύψωση σε δύναμη έχει την επόμενη μεγαλύτερη προτεραιότητα. Ο πολλαπλασιασμός, η διαίρεση και το υπόλοιπο της ακεραίας διαίρεσης έχουν την ίδια προτεραιότητα, η οποία είναι μεγαλύτερη από την πρόσθεση και την αφαίρεση που μεταξύ τους έχουν επίσης την ίδια προτεραιότητα. Αν οι πράξεις που θέλουμε να κάνουμε έχουν την ίδια προτεραιότητα, τότε εκτελούμε τις πράξεις από τα αριστερά προς τα δεξιά.

Η προτεραιότητα στους λογικούς τελεστές είναι από τα αριστερά προς τα δεξιά.

6.4. Ασκήσεις 3ης Ενότητας

Multiple Choice:

1. Στην έκφραση $4+6$ οι αριθμοί 4 και 6 ονομάζονται:
 - a) τελεστές
 - b) τελεστέοι

- c) ακέραιοι
- 2. Ο τελεστής % τι αποτέλεσμα εμφανίζει:
 - a) Υπόλοιπο ακέραιας διαίρεσης
 - b) Ακέραια διαίρεση
 - c) Ύψωση σε δύναμη
- 3. Το αποτέλεσμα της διαίρεσης $8/5$ είναι:
 - a) 1.6
 - b) 2
 - c) 1
- 4. Ποια από τις παραπάνω συγκρίσεις είναι αληθής αν $x=5$;
 - a) $x \geq 5$
 - b) $x < 4$
 - c) $x != 5$
- 5. Ποιο είναι το αποτέλεσμα της έκφρασης $5 \% 2 * 10$;
 - a) 20
 - b) 10
 - c) 5

Ερωτήσεις Σωστό ή Λάθος

1. Αν $x = \Psi\text{EY}\Delta\text{H}\Sigma$ και $y = \text{A}\text{L}\text{H}\Theta\text{H}\Sigma$ τότε x or y αληθής
2. Αν $x = \text{A}\text{L}\text{H}\Theta\text{H}\Sigma$ και $y = \Psi\text{EY}\Delta\text{H}\Sigma$ τότε x and y αληθής
3. Οι τελεστές μπορούν να αναπαρασταθούν μόνο με σύμβολα
4. Η προτεραιότητα πράξεων ανάμεσα σε λογικούς τελεστές γίνεται από τα δεξιά στα αριστερά
5. Στην προτεραιότητα πράξεων ανάμεσα στους αριθμητικούς τελεστές πρώτα εκτελείται η διαίρεση
6. Οι παρενθέσεις κάνουν πιο δυσανάγνωστες τις εκφράσεις
7. Αν $x=10$ τότε η έκφραση $0 < x \leq 10$ είναι σωστή
8. Ο πολλαπλασιασμός, η διαίρεση και το υπόλοιπο ακέραιας διαίρεσης έχουν την ίδια προτεραιότητα πράξεων
9. Όταν υπάρχει ίδια προτεραιότητα, τότε η έκφραση υπολογίζεται από τα δεξιά προς τα αριστερά
10. Η εντολή `print x NOT` είναι σωστή
11. Αν $x=6$ τότε η έκφραση $x \geq 5$ είναι σωστή
12. Η έκφραση $x = \text{false}$ είναι σωστή
13. Οι λογικοί τελεστές γράφονται με κεφαλαία
14. Πρέπει να υπάρχει κενό ανάμεσα στη μεταβλητή και τον τελεστή

Ασκήσεις:

Αν $X = \Psi\text{EY}\Delta\text{H}\Sigma$, $Y = \text{A}\text{L}\text{H}\Theta\text{H}\Sigma$ και $Z = \text{A}\text{L}\text{H}\Theta\text{H}\Sigma$ τότε να υπολογίσετε:

1. X and Y or Z
2. X and Z
3. Y or Z
4. not Z
5. not X and Y or Z
6. not Z or Z and Y

7. ΕΝΟΤΗΤΑ 4 - ΕΝΤΟΛΕΣ ΕΛΕΓΧΟΥ

Τίτλος Ενότητας	Εντολές Ελέγχου
Τίτλος Μαθήματος	Βασικές αρχές προγραμματισμού χρησιμοποιώντας την Python
Περιγραφή	Στην ενότητα αυτή θα παρουσιαστούν οι εντολές ελέγχου που χρησιμοποιούνται στην Python και θα αναφερθεί η έννοια της συνάρτησης. Στις εντολές ελέγχου περιγράφονται οι κανόνες για τη σύνταξή τους και δίνονται παραδείγματα. Αντίστοιχα παρουσιάζονται τα οφέλη της χρήσης συναρτήσεων και πώς είναι η σύνταξή τους, ενώ δίνονται και σχετικά παραδείγματα.
Εκπαιδευτικοί στόχοι	<p>Στόχοι για την ενότητα αυτή είναι να μπορούν οι εκπαιδευόμενοι:</p> <ul style="list-style-type: none"> • Να γνωρίζουν τους κανόνες σύνταξης των συνθηκών ελέγχου • Να γνωρίζουν τους κανόνες σύνταξης των συναρτήσεων • Να κατανοούν τον τρόπο σύνταξης των εμφωλευμένων εντολών • Να κατανοούν πώς τους βοηθά η χρήση των συναρτήσεων στη λύση προβλημάτων και καθιστά πιο εύκολη τη σύνταξη κώδικα
Μαθησιακά αποτελέσματα	<p>Αφού μελετήσουν την ενότητα αυτή, οι εκπαιδευόμενοι:</p> <ul style="list-style-type: none"> • Θα μπορούν να δώσουν παραδείγματα χρήσης των εντολών ελέγχου • Θα μπορούν να αντιμετωπίσουν προβλήματα που στη λύση τους απαιτείται η χρήση εντολών ελέγχου • Θα μπορούν να κάνουν τη διαδικασία σύνταξης του κώδικα πιο απλή μέσω των συναρτήσεων
Εκπαιδευτικές δραστηριότητες	<p>Οι εκπαιδευτικές δραστηριότητες της ενότητας είναι οι εξής:</p> <ul style="list-style-type: none"> • Εντολή Ελέγχου if • Εντολή Ελέγχου elif και Εμφωλευμένες Εντολές • Συναρτήσεις
Εμπλεκόμενοι Ρόλοι	Στη διάρκεια αυτής της ενότητας εμπλέκονται οι ρόλοι των σχεδιαστών της ενότητας και των εκπαιδευόμενων.

Αξιολόγηση	Η ενότητα του μαθήματος αυτού χρησιμοποιεί σε κάθε υποενότητα και σε κάθε μαθησιακό αντικείμενο ασκήσεις αυτοαξιολόγησης. Τέτοιες μπορεί να είναι ασκήσεις Multiple choice.
Συνολικός χρόνος ενότητας	1 εβδομάδα

7.1. Εντολή Ελέγχου if

Τίτλος Υποενότητας	Εντολή Ελέγχου if
Τίτλος Ενότητας	Εντολές Ελέγχου
Εκπαιδευτική Στρατηγική	Θα χρησιμοποιηθούν θεωρία, ασκήσεις, παραδείγματα και πρακτική εφαρμογή με ασκήσεις αυτοαξιολόγησης.
Περιγραφή	Στην υποενότητα αυτή παρουσιάζεται η εντολή ελέγχου if. Αυτή χρησιμοποιείται για τον έλεγχο συνθηκών και ανάλογα με αυτό τον έλεγχο (αληθής ή ψευδής) εκτελούνται ή όχι κάποιες εντολές. Αναφέρεται ποιος είναι ο σωστός τρόπος σύνταξης και δίνεται παράδειγμα χρήσης.
Γλώσσα	Ελληνικά
Μαθησιακά Αποτελέσματα	Μέσα από αυτή την υποενότητα οι εκπαιδευόμενοι: <ul style="list-style-type: none"> • Θα κατανοήσουν τη σημασία της εντολής if • Θα γνωρίζουν πώς να συντάξουν την εντολή για να χρησιμοποιηθεί σε πρόγραμμα
Μαθησιακά Αντικείμενα	Εντολή if
Αξιολόγηση Εκπαιδευόμενων	Η αξιολόγηση της υποενότητας γίνεται με ένα (1) μαθησιακό αντικείμενο με ασκήσεις αυτοαξιολόγησης στο τέλος της ενότητας οι οποίες θα βασίζονται στην θεωρία με την βοήθεια παραδειγμάτων.
Λέξεις Κλειδιά	Δεσμευμένη λέξη, εντολές, συνθήκες

7.1.1. Μαθησιακό Αντικείμενο 11 - Εντολή Ελέγχου if

Τίτλος	Εντολή Ελέγχου if
Τίτλος Υποενότητας	Εντολή Ελέγχου if

Περιγραφή	Σκοπός αυτού του μαθησιακού αντικειμένου είναι να παρουσιάσει τις εντολές ελέγχου, συγκεκριμένα την εντολή if. Μέσα από αυτή την παρουσίαση γίνεται κατανοητή η έννοια της εντολής ελέγχου, αλλά και πως συντάσσεται.
Γλώσσα	Ελληνικά
Μαθησιακός Τύπος	Θεωρία, ασκήσεις και παραδείγματα
Τεχνικός Τύπος	Παρουσίαση, βιντεοδιάλεξη
Λέξεις Κλειδιά	if, else, true, false
Μαθησιακά Αποτελέσματα	Όταν οι εκπαιδευόμενοι ολοκληρώσουν τη μελέτη αυτού του μαθησιακού αντικειμένου: <ul style="list-style-type: none"> • Θα αναγνωρίζουν πότε εκτελούνται εντολές μέσα σε μια εντολή ελέγχου • Θα κατανοούν τις έννοιες True, False • Θα γνωρίζουν πώς να συντάξουν την εντολή if

ΕΝΤΟΛΗ ΕΛΕΓΧΟΥ IF

Χρησιμοποιούμε την εντολή if για να ελέγξουμε τη ροή της εκτέλεσης ενός προγράμματος. Ελέγχεται μία συνθήκη και ανάλογα με το αποτέλεσμα (Αληθής ή Ψευδής) εκτελείται ή δεν εκτελείται μία ή κάποια άλλη ομάδα (μπλοκ) εντολών. Συντάσσουμε την εντολή if ως εξής:

if (συνθήκη) :

 Εντολή / εντολές

else (διαφορετικό της συνθήκης):

 Εντολή / εντολές

Στην πρώτη γραμμή βλέπουμε μία επικεφαλίδα με τη δεσμευμένη λέξη if, μία συνθήκη και στο τέλος μία άνω κάτω τελεία (:).

Η συνθήκη μπορεί να είναι μία οποιαδήποτε λογική έκφραση που παίρνει τιμή True ή False. Αν η συνθήκη είναι αληθής (True), τότε εκτελούνται οι εντολές. Αν η συνθήκη είναι ψευδής προχωράει στην else, όπου εκτελούνται οι εντολές που έχουμε ορίσει σε αυτή.

Μέσα στην εντολή if μπορούμε να βάλουμε όσες εντολές θέλουμε, αρκεί να υπάρχει τουλάχιστον μία.

Οι εντολές πρέπει να είναι μετατοπισμένες προς τα δεξιά. Η τυπική μετατόπιση ή εσοχή (indentation) των εντολών είναι τέσσερα κενά. Η Pythοn μάς βοηθάει σε αυτό ρυθμίζοντας αυτόματα τις εσοχές για μας, απλά πατώντας Enter μετά την πληκτρολόγηση της άνω κάτω τελείας. Αν λείπει ένα κενό ή υπάρχουν παραπάνω από 4 τότε μπορεί να προκύψει λάθος στη συνέχεια του προγράμματος.

ΠΑΡΑΔΕΙΓΜΑΤΑ

```
x=input('Δώσε βαθμολογία');
```

```
if x>=5 :
```

```
    print 'Πέρασες'
```

```
else :
```

```
    print 'Κόπηκες'
```

Το πρόγραμμα που ακολουθεί διαβάζει έναν αριθμό από τον χρήστη και εκτυπώνει την απόλυτη τιμή του.

```
num=input('Δώσε αριθμό για τον υπολογισμό απόλυτης τιμής')
```

```
if num<0:
```

```
    num=num*(-1)
```

```
    print num
```

```
else:
```

```
    print num
```

Εδώ το πρόγραμμα διαβάζει έναν αριθμό από τον χρήστη και ελέγχει αν είναι θετικός ή αρνητικός. Αν είναι αρνητικός, υπολογίζει την απόλυτη τιμή του. Αν είναι θετικός, απλά τυπώνει τον αριθμό.

7.2. Εντολή Ελέγχου elif και Εμφωλευμένες Εντολές

Τίτλος Υποενότητας	Εντολή Ελέγχου elif και Εμφωλευμένες Εντολές
Τίτλος Ενότητας	Εντολές Ελέγχου
Εκπαιδευτική στρατηγική	Θα χρησιμοποιηθούν θεωρία, ασκήσεις, παραδείγματα και πρακτική εφαρμογή με ασκήσεις αυτοαξιολόγησης.
Περιγραφή	Η υποενότητα αυτή παρουσιάζει την εντολή ελέγχου elif και τις εμφωλευμένες συνθήκες. Μέσα από παραδείγματα εξηγείται η σύνταξη της εντολής, καθώς και σε ποιες περιπτώσεις χρησιμοποιείται. Έπειτα εξηγείται η έννοια των εμφωλευμένων συνθηκών και πώς χρησιμοποιούνται.
Γλώσσα	Ελληνικά

Μαθησιακά Αποτελέσματα	Μέσα από αυτή την υποενότητα οι εκπαιδευόμενοι: <ul style="list-style-type: none"> • Θα κατανοήσουν τη σημασία της εντολής elif • Θα είναι σε θέση να χρησιμοποιήσουν την εντολή elif σε δικό τους πρόγραμμα • Θα γνωρίζουν τι είναι οι εμφωλευμένες συνθήκες • Θα είναι ικανοί να δώσουν παραδείγματα χρήσης των αντικειμένων που αναφέρθηκαν
Μαθησιακά Αντικείμενα	Εντολή elif, Εμφωλευμένες συνθήκες
Αξιολόγηση Εκπαιδευόμενων	Η αξιολόγηση της υποενότητας γίνεται με ένα (1) μαθησιακό αντικείμενο με ασκήσεις αυτοαξιολόγησης στο τέλος της ενότητας οι οποίες θα βασίζονται στην θεωρία με την βοήθεια παραδειγμάτων.
Λέξεις Κλειδιά	Δεσμευμένη λέξη, elif, εμφωλευμένη, συνθήκη

7.2.1. Μαθησιακό Αντικείμενο 12 - Εντολή Ελέγχου elif και Εμφωλευμένες εντολές

Τίτλος	Εντολή Ελέγχου elif και Εμφωλευμένες εντολές
Τίτλος Υποενότητας	Εντολή Ελέγχου elif και Εμφωλευμένες εντολές
Περιγραφή	Σκοπός αυτού του μαθησιακού αντικείμενου είναι να παρουσιάσει τις εντολές ελέγχου, συγκεκριμένα την εντολή elif. Μέσα από αυτή την παρουσίαση γίνεται κατανοητή η έννοια της εντολής ελέγχου αλλά και πώς συντάσσεται. Ακόμη παρουσιάζεται η έννοια των εμφωλευμένων συνθηκών και δίνονται παραδείγματα χρήσης.
Γλώσσα	Ελληνικά
Μαθησιακός Τύπος	Θεωρία, ασκήσεις και παραδείγματα
Τεχνικός Τύπος	Παρουσίαση, βιντεοδιάλεξη
Λέξεις Κλειδιά	Elif, συνθήκη, αληθής, ψευδής
Μαθησιακά Αποτελέσματα	Μόλις ολοκληρώσουν τη μελέτη αυτού του μαθησιακού αντικείμενου οι εκπαιδευόμενοι θα είναι σε θέση: <ul style="list-style-type: none"> • Να γνωρίζουν πότε μέσα σε μια εντολή ελέγχου εκτελούνται εντολές • Να κατανοήσουν τις έννοιες True, False • Να γνωρίζουν πώς να συντάξουν την εντολή elif • Να γράψουν κώδικα με εμφωλευμένες συνθήκες

ΕΝΤΟΛΗ ΕΛΕΓΧΟΥ ELIF ΚΑΙ ΕΜΦΩΛΕΥΜΕΝΕΣ ΕΝΤΟΛΕΣ

Στην Python μπορούμε να έχουμε αλυσιδωτές συνθήκες (με πολλές συνθήκες και εντολές). Εκτελούνται μόνο οι εντολές που υπακούουν σε μία συνθήκη. Κάθε συνθήκη ελέγχεται με τη σειρά, αν η πρώτη είναι ψευδής, πάμε στη δεύτερη κ.ο.κ. Αν κάποια συνθήκη είναι αληθής, τότε εκτελείται η αντίστοιχη ομάδα εντολών και η εντολή if ολοκληρώνεται, ακόμη και αν ακολουθούν και άλλες συνθήκες που αληθεύουν. Για αλυσιδωτές συνθήκες χρησιμοποιούμε τη δεσμευμένη λέξη elif (συντόμευση του else if).

ΠΑΡΑΔΕΙΓΜΑΤΑ

```
x=input('Δώσε βαθμολογία')
```

```
if x>=0 AND x<5 :
```

```
    print 'Κόπηκες'
```

```
elif x>=5 AND x <8.5:
```

```
    print 'Πέρασες με 'καλώς''
```

```
else:
```

```
    print 'Πέρασες με 'άριστα''
```

Το πρόγραμμα που ακολουθεί παίρνει από τον χρήστη δύο αριθμούς και την επιλογή για το αν θα τους προσθέσει ή θα τους πολλαπλασιάσει και εκτυπώνει το αντίστοιχο αποτέλεσμα.

```
x=input('Αν θέλεις να προσθέσεις τους αριθμούς πληκτρολόγησε 1, αν θέλεις να τους πολλαπλασιάσεις πληκτρολόγησε 2\n')
```

```
num1=input('Δώσε τον πρώτο αριθμό:\n')
```

```
num2=input('Δώσε τον δεύτερο αριθμό:\n')
```

```
if x==1:
```

```
    result=num1+num2
```

```
    print 'Το αποτέλεσμα της πρόσθεσης των δύο αριθμών είναι: ', result
```

```
elif x==2:
```

```
    result=num1*num2
```

```
    print 'Το αποτέλεσμα του πολλαπλασιασμού των δύο αριθμών είναι: ', result
```

```
else:
```

```
    print 'Μη έγκυρη επιλογή'
```

Σε αυτό το πρόγραμμα δίνουμε στον χρήστη δυο επιλογές, είτε να προσθέσει είτε να πολλαπλασιάσει. Για να "διαβάσουμε" αυτή την επιλογή εκχωρούμε σε μια μεταβλητή την απάντηση του χρήστη. Έπειτα ζητάμε από τον χρήστη να μας δώσει τους δυο αριθμούς που θέλει να προσθέσει ή να πολλαπλασιάσει. Στην πρώτη περίπτωση, αν το x είναι 1, το

πρόγραμμα προχωράει στο να προσθέσει τους δύο αριθμούς. Αν το x δεν είναι 1, ελέγχει αν είναι 2. Αν η συνθήκη είναι αληθής, προχωράει στην εκτέλεση των εντολών μέσα στο σώμα της συνθήκης. Τέλος, επειδή δεν μπορούμε να είμαστε σίγουροι ότι ο χρήστης δεν θα βάλει οτιδήποτε άλλο εκτός από 1 και 2, βάζουμε μια τρίτη εντολή ελέγχου για να εμφανιστεί στον χρήστη μήνυμα για την μη έγκυρη επιλογή αριθμού.

Η Pythοn μας επιτρέπει να έχουμε εμφωλευμένες συνθήκες, δηλαδή μια συνθήκη μέσα σε μια άλλη. Οι εμφωλευμένες συνθήκες δεν χρησιμοποιούνται τόσο συχνά γιατί κάνουν τα προγράμματα δυσανάγνωστα. Οι εμφωλευμένες συνθήκες συντάσσονται όπως φαίνεται παρακάτω.

if (συνθήκη):

 Εντολή/εντολές

else:

if (συνθήκη):

 Εντολή/εντολές

else:

 εντολές

ΠΑΡΑΔΕΙΓΜΑΤΑ

Το πρόγραμμα που ακολουθεί διαβάζει από τον χρήστη έναν βαθμό και εμφανίζει το κατάλληλο μήνυμα για κάθε περίπτωση.

```
x=input('Δώσε βαθμολογία')
```

```
if x<5 :
```

```
    print 'Κόπηκες'
```

```
else:
```

```
    if x>=5 AND x<=8.5:
```

```
        print 'Πέρασες με 'καλώς''
```

```
    else:
```

```
        print 'Πέρασες με 'άριστα''
```

Το πρόγραμμα που ακολουθεί διαβάζει την ηλικία του χρήστη και πόσα χρόνια έχει δίπλωμα αυτοκινήτου (αν έχει) και εμφανίζει αν μπορεί ο χρήστης να ενοικιάσει αυτοκίνητο. Το ελάχιστο όριο ηλικίας είναι 23 έτη και το μέγιστο 70 έτη. Η άδεια οδήγησης θα πρέπει να έχει εκδοθεί 1 έτος πριν την ημερομηνία ενοικίασης.

```
age=input('Δώσε την ηλικία σου\n')
```

```
if (age<=18):
```

```
    print 'Δεν έχεις δίπλωμα'
```

else:

```
years=input('Πόσα χρόνια έχεις δίπλωμα οδήγησης;')
```

```
if (years>=1) and (age>=23) and (age<=70):
```

```
    print 'Μπορείς να ενοικιάσεις αμάξι'
```

```
else:
```

```
    print 'Δεν μπορείς να ενοικιάσεις αμάξι'
```

7.3. Συναρτήσεις

Τίτλος Υποενότητας	Συναρτήσεις
Τίτλος Ενότητας	Εντολές ελέγχου
Εκπαιδευτική Στρατηγική	Θα χρησιμοποιηθούν θεωρία, ασκήσεις, παραδείγματα και πρακτική εφαρμογή με ασκήσεις αυτοαξιολόγησης.
Περιγραφή	Η υποενότητα αυτή παρουσιάζει την έννοια των συναρτήσεων. Αναφέρεται το πώς ορίζονται οι συναρτήσεις και οι παράμετροί τους. Ακόμα δίνονται δύο τρόποι με τους οποίους μπορούμε να τρέξουμε τις συναρτήσεις.
Γλώσσα	Ελληνικά
Μαθησιακά Αποτελέσματα	Μέσα από αυτή την υποενότητα οι εκπαιδευόμενοι: <ul style="list-style-type: none">• Θα κατανοήσουν την έννοια της συνάρτησης• Θα γνωρίζουν πώς να ορίζουν συναρτήσεις• Θα γνωρίζουν πώς να δίνουν παραμέτρους στις συναρτήσεις• Θα γνωρίζουν πώς να τρέχουν τις συναρτήσεις στο πρόγραμμά τους
Μαθησιακά Αντικείμενα	Συναρτήσεις
Αξιολόγηση Εκπαιδευόμενων	Η αξιολόγηση της υποενότητας γίνεται με ένα (1) μαθησιακό αντικείμενο με ασκήσεις αυτοαξιολόγησης στο τέλος της ενότητας οι οποίες θα βασίζονται στην θεωρία με την βοήθεια παραδειγμάτων.
Λέξεις Κλειδιά	Συναρτήσεις, κλήση συνάρτησης, παράμετροι

7.3.1. Μαθησιακό Αντικείμενο 13 - Συναρτήσεις

Τίτλος	Συναρτήσεις
Τίτλος Υποενότητας	Συναρτήσεις

Περιγραφή	Σε αυτό το μαθησιακό αντικείμενο παρουσιάζεται η έννοια της συνάρτησης, καθώς και το πώς ορίζεται μέσα από μία δεσμευμένη λέξη. Έπειτα παρουσιάζεται το πώς ορίζονται οι παράμετροι για πιο σύνθετες συναρτήσεις και οι τρόποι με τους οποίους τρέχουμε τη συνάρτηση.
Γλώσσα	Ελληνικά
Μαθησιακός Τύπος	Θεωρία, ασκήσεις και παραδείγματα
Τεχνικός Τύπος	Παρουσίαση, βιντεοδιάλεξη
Λέξεις Κλειδιά	NOT, AND, OR, αληθής, ψευδής
Μαθησιακά Αποτελέσματα	Όταν οι εκπαιδευόμενοι ολοκληρώσουν τη μελέτη αυτού του μαθησιακού αντικειμένου: <ul style="list-style-type: none"> • Θα γνωρίζουν τι είναι μία συνάρτηση • Θα γνωρίζουν πώς να ορίζουν συναρτήσεις • Θα γνωρίζουν πώς να δίνουν παραμέτρους στη συνάρτηση • Θα γνωρίζουν πώς να τρέχουν τις συναρτήσεις στο πρόγραμμά τους

ΣΥΝΑΡΤΗΣΕΙΣ

Οι συναρτήσεις είναι επαναχρησιμοποιήσιμα μέρη προγραμμάτων. Μας επιτρέπουν να δίνουμε ένα όνομα σε ένα σύνολο εντολών και να τρέχουμε το σύνολο εντολών χρησιμοποιώντας το όνομά τους, οπουδήποτε στο πρόγραμμα και όσες φορές θέλουμε. Αυτό είναι γνωστό σαν κλήση (calling) της συνάρτησης.

Οι συναρτήσεις ορίζονται χρησιμοποιώντας τη λέξη κλειδί **def**, μετά την οποία ακολουθεί ένα όνομα που ταυτοποιεί τη συνάρτηση και κατόπιν ακολουθεί ένα ζευγάρι παρενθέσεων που μπορούν να περικλείουν μερικά ονόματα παραμέτρων και η γραμμή τελειώνει με διπλή τελεία (:).

Ας δώσουμε ένα παράδειγμα:

```
def sayHello():
    print 'Hello World!'
```

Τώρα αν σε οποιοδήποτε μέρος του προγράμματος γράψουμε τη συνάρτηση sayHello θα εμφανιστεί το μήνυμα Hello World!. Μπορούμε να καλέσουμε τη συνάρτηση όσες φορές θέλουμε μέσα στο πρόγραμμα.

Μια συνάρτηση μπορεί να δεχθεί παραμέτρους, οι οποίες είναι τιμές που δίνονται στη συνάρτηση, έτσι ώστε αυτή να μπορεί να κάνει κάτι αξιοποιώντας αυτές τις τιμές. Αυτές οι παράμετροι μοιάζουν με τις μεταβλητές, διαφέροντας ως προς το ότι οι τιμές αυτών των μεταβλητών ορίζονται όταν καλούμε τη συνάρτηση και τους έχουν ήδη εκχωρηθεί τιμές όταν τρέχει η συνάρτηση. Οι παράμετροι καθορίζονται μέσα στο ζευγάρι των παρενθέσεων

στον ορισμό της συνάρτησης και διαχωρίζονται με κόμμα. Όταν καλούμε τη συνάρτηση δίνουμε και τις τιμές με τον ίδιο τρόπο. Σημείωση για την ορολογία που χρησιμοποιείται: οι ονομασίες που δίνονται στον ορισμό της συνάρτησης ονομάζονται παράμετροι, ενώ οι τιμές που δίνονται όταν καλείται η συνάρτηση ονομάζονται ορίσματα.

Παράδειγμα:

```
def printMax(a, b):  
    if a > b:  
        print a, 'είναι το μέγιστο'  
    elif a == b:  
        print a, 'είναι ίσο με το', b  
    else:  
        print b, 'είναι το μέγιστο'
```

Μπορούμε λοιπόν να τρέξουμε με δύο τρόπους τη συνάρτηση.

Ο πρώτος είναι να καλέσουμε τη συνάρτηση έχοντας δώσει ήδη ορίσματα.

```
printMax(3, 4) # δίνουμε απ' ευθείας κυριολεκτικές τιμές
```

Έξοδος: 4 είναι το μέγιστο

Ο δεύτερος είναι να δώσουμε τιμές σε δύο μεταβλητές και να καλέσουμε τη συνάρτηση με τις μεταβλητές. Αυτός ο τρόπος χρησιμοποιείται όταν τα ορίσματα είναι μεταβλητά, αλλάζουν δηλαδή όσο τρέχει το πρόγραμμα.

```
x = 5
```

```
y = 7
```

```
printMax(x, y) # δίνουμε μεταβλητές σαν ορίσματα
```

Έξοδος: 7 είναι το μέγιστο

Πώς δουλεύει:

Εδώ ορίζουμε μια συνάρτηση που ονομάζεται printMax με δύο παραμέτρους, τις a και b. Εντοπίζουμε τη μεγαλύτερη τιμή χρησιμοποιώντας μια απλή εντολή if ... else και μετά τυπώνουμε τη μεγαλύτερη τιμή ή ένα μήνυμα ότι οι δύο τιμές είναι ίσες.

ΠΑΡΑΔΕΙΓΜΑ

Η συνάρτηση που ακολουθεί έχει ως παραμέτρους δύο αριθμούς και υπολογίζει ποιος από τους δύο είναι μικρότερος.

```
def printMin(a, b)  
    if a < b:  
        print a, 'είναι το μικρότερο'  
    elif a == b:  
        print a, 'είναι ίσο με το', b  
    else:
```

```
print b, ' είναι το μικρότερο'
```

Στο κυρίως πρόγραμμα, ο χρήστης δίνει δύο αριθμούς και εκτυπώνεται ο μικρότερος, με κλήση της συνάρτησης printMin.

```
num1=input('Δώσε τον 1ο αριθμό')  
num2=input('Δώσε τον 2ο αριθμό')  
printMin(num1,num2)
```

7.4. Ασκήσεις 4ης Ενότητας

Ερωτήσεις Σωστό ή Λάθος

1. Η λέξη if στην Python είναι δεσμευμένη λέξη
2. Μια συνθήκη τελειώνει με τη λέξη if
3. Αν η συνθήκη είναι αληθής τότε προχωράει στην else
4. Υπάρχει συγκεκριμένος αριθμός εντολών που μπορούμε να ορίσουμε σε μια εντολή if
5. Οι εμφωλευμένες συνθήκες κάνουν το πρόγραμμα πιο ευανάγνωστο
6. Οι συναρτήσεις δεν επαναχρησιμοποιούνται
7. Οι συναρτήσεις ορίζονται με τη λέξη κλειδί def
8. Οι συναρτήσεις δέχονται παραμέτρους
9. Οι παράμετροι μιας συνάρτησης δεν μπορούν να είναι μεταβλητές
10. Υπάρχει μόνο ένας τρόπος να τρέξουμε μια συνάρτηση
11. Οι έννοιες παράμετροι και ορίσματα είναι ταυτόσημες
12. Μπορούμε να έχουμε μόνο μια elif σε κάθε εντολή ελέγχου
13. Οι συναρτήσεις διευκολύνουν τον προγραμματιστή
14. Αν δεν μπει η άνω κάτω τελεία μετά την δεσμευμένη λέξη if το πρόγραμμα δεν τρέχει
15. Με την εντολή if - else υπάρχουν μόνο δύο επιλογές

8. ΕΝΟΤΗΤΑ 5 - ΕΝΤΟΛΕΣ ΕΠΑΝΑΛΗΨΗΣ

Τίτλος Ενότητας	Εντολές Επανάληψης
Τίτλος Μαθήματος	Βασικές αρχές προγραμματισμού χρησιμοποιώντας την Python
Περιγραφή	Στην ενότητα αυτή θα παρουσιαστούν οι εντολές επανάληψης που χρησιμοποιούνται στην Python, καθώς και οι εντολές “αναγκαστικής εξόδου” break και continue. Σε κάθε υποενότητα ορίζεται ο τρόπος σύνταξής τους, και παρουσιάζονται παραδείγματα για καθένα από αυτά.
Εκπαιδευτικοί στόχοι	Στόχοι για την ενότητα αυτή είναι να μπορούν οι εκπαιδευόμενοι με την ολοκλήρωση της μελέτης της: <ul style="list-style-type: none"> • Να γνωρίζουν ποιες είναι οι εντολές επανάληψης • Να μπορούν να συντάξουν αυτές τις εντολές • Να κατανοούν τη χρησιμότητα των εντολών “αναγκαστικής εξόδου” break και continue καθώς και το πώς συντάσσονται
Μαθησιακά αποτελέσματα	Αφού μελετήσουν την ενότητα αυτή, οι εκπαιδευόμενοι: <ul style="list-style-type: none"> • Θα μπορούν να δώσουν παραδείγματα χρήσης των εντολών επανάληψης • Θα κατανοούν τη χρησιμότητά της • Θα μπορούν να μειώσουν το μέγεθος του κώδικα χρησιμοποιώντας τις εντολές επανάληψης • Θα γνωρίζουν τη σύνταξη των εντολών
Εκπαιδευτικές δραστηριότητες	Οι εκπαιδευτικές δραστηριότητες της ενότητας είναι οι εξής: <ul style="list-style-type: none"> • Εντολή Επανάληψης for • Εντολή Επανάληψης while • Εντολές break και continue
Εμπλεκόμενοι Ρόλοι	Στη διάρκεια αυτής της ενότητας εμπλέκονται οι ρόλοι των σχεδιαστών της ενότητας και των εκπαιδευόμενων
Αξιολόγηση	Η ενότητα του μαθήματος αυτού χρησιμοποιεί σε κάθε υποενότητα και σε κάθε μαθησιακό αντικείμενο ασκήσεις αυτοαξιολόγησης. Τέτοιες μπορεί να είναι ασκήσεις Multiple choice ή Σωστό-Λάθος.
Συνολικός χρόνος ενότητας	1 εβδομάδα

8.1. Εντολή Επανάληψης for

Τίτλος Υποενότητας	Εντολή Επανάληψης for
Τίτλος Ενότητας	Εντολές Επανάληψης
Εκπαιδευτική Στρατηγική	Θα χρησιμοποιηθούν θεωρία, ασκήσεις, παραδείγματα και πρακτική εφαρμογή με ασκήσεις αυτοαξιολόγησης
Περιγραφή	Στην υποενότητα αυτή παρουσιάζεται η εντολή επανάληψης for. Αναφέρεται στην χρησιμότητά της όσον αφορά στο μέγεθος του κώδικα, αφού απλοποιεί πολύ κάποιες διαδικασίες. Δίνεται επίσης ο τρόπος σύνταξης αυτής της εντολής.
Γλώσσα	Ελληνικά
Μαθησιακά Αποτελέσματα	Μέσα από αυτή την υποενότητα οι εκπαιδευόμενοι: <ul style="list-style-type: none"> • Θα κατανοήσουν τη σημασία της εντολής επανάληψης for • Θα γνωρίζουν πώς να συντάξουν την εντολή για να χρησιμοποιηθεί σε ένα πρόγραμμα
Μαθησιακά Αντικείμενα	Εντολή επανάληψης for
Αξιολόγηση Εκπαιδευόμενων	Η αξιολόγηση της υποενότητας γίνεται με ένα (1) μαθησιακό αντικείμενο με ασκήσεις αυτοαξιολόγησης στο τέλος της ενότητας οι οποίες θα βασίζονται στην θεωρία με την βοήθεια παραδειγμάτων.
Λέξεις Κλειδιά	Εντολή επανάληψης, βρόχοι, εντολές, επαναλήψεις

8.1.1. Μαθησιακό Αντικείμενο 14 - Εντολή Επανάληψης for

Τίτλος	Εντολή επανάληψης for
Τίτλος Υποενότητας	Εντολή επανάληψης for
Περιγραφή	Σκοπός αυτού του μαθησιακού αντικειμένου είναι να παρουσιάσει τις εντολές επανάληψης, συγκεκριμένα την εντολή for. Μέσα από αυτή την παρουσίαση γίνεται κατανοητή η έννοια της εντολής επανάληψης αλλά και πώς αυτή συντάσσεται. Επίσης θα κατανοούν την έννοια «βήμα» και πώς να το αλλάζουν σύμφωνα με τις απαιτήσεις του προγράμματος.
Γλώσσα	Ελληνικά
Μαθησιακός Τύπος	Θεωρία, ασκήσεις και παραδείγματα
Τεχνικός Τύπος	Παρουσίαση, βιντεοδιάλεξη

Λέξεις Κλειδιά	for, in, range
Μαθησιακά Αποτελέσματα	<p>Όταν οι εκπαιδευόμενοι ολοκληρώσουν τη μελέτη αυτού του μαθησιακού αντικειμένου:</p> <ul style="list-style-type: none"> • Θα γνωρίζουν πώς να συντάσσουν την εντολή επανάληψης for • Θα κατανοούν τις έννοιες που περιλαμβάνονται στη δομή της εντολής • Θα γνωρίζουν πώς να ορίζουν το «βήμα» της επανάληψης

ΕΝΤΟΛΗ ΕΠΑΝΑΛΗΨΗΣ FOR

Ένας κώδικας μπορεί να γίνει πιο απλός και πιο εύκολος στη σύνταξή του αν χρησιμοποιήσουμε εντολές επανάληψης. Αυτό σημαίνει ότι κατά την εκτέλεση του προγράμματος μπορούμε να επαναλάβουμε πολλές φορές την ίδια εντολή / εντολές. Μια ομάδα εντολών που επαναλαμβάνονται ονομάζεται βρόχος (loop).

Η εντολή for επαναλαμβάνει ένα μπλοκ εντολών για συγκεκριμένο πλήθος φορών.

Για παράδειγμα, ο παρακάτω κώδικας τυπώνει τους ακέραιους αριθμούς από το 0 έως και το 9, με τη βοήθεια και της ενσωματωμένης λίστας range:

```
for i in range(10):
```

```
    print i
```

Αν εκτελέσουμε αυτόν τον κώδικα, θα μας εκτυπώσει τους αριθμούς από 0 έως 9 τον έναν κάτω από τον άλλο.

Βλέπουμε ότι μετά τη δεσμευμένη λέξη for ακολουθεί μια μεταβλητή i, η λέξη κλειδί in και στη συνέχεια η λίστα range, ενώ η εντολή ολοκληρώνεται με άνω κάτω τελεία (:). Όπως και στην εντολή if, έτσι και στην εντολή for αφήνουμε τέσσερα κενά για την εντολή εκτύπωσης. Η μεταβλητή i παίρνει την τιμή 0 και στην τελευταία επανάληψη την τιμή 9. Προσοχή! Ο αριθμός 10 δεν θα τυπωθεί. Αν θέλουμε να τυπωθεί θα πρέπει μέσα στην συνάρτηση range να βάλουμε τον αριθμό 11. Σε κάθε επανάληψη η τιμή της μεταβλητής i αυξάνει κατά 1.

Σε περίπτωση που δεν θέλουμε η μέτρηση των επαναλήψεων να ξεκινάει από μηδέν αλλά να ξεκινάει από έναν άλλο αριθμό τότε:

```
for i in range(5, 10):
```

```
    print i
```

Η μεταβλητή i θα πάρει την πρώτη τιμή, σε αυτή την περίπτωση 5, και θα αρχίσει την εκτύπωση των αριθμών από το 5 μέχρι και το 9.

Σε περίπτωση που θέλουμε να ξεκινάει από έναν άλλο αριθμό διάφορο του 0 και θέλουμε το i να μην αυξάνεται κατά 1 κάθε φορά, τότε:

```
for i in range(1, 10, 2):
```

```
    print i
```

Σε αυτό τον κώδικα η αρχική τιμή που θα πάρει το i είναι 1 και θα εμφανίσει τα αποτελέσματα 1,3,5,7,9

Στη συνάρτηση range δηλαδή θα έχω:

range (αρχική τιμή, τελική τιμή, βήμα επανάληψης)

Βλέπουμε ότι το βήμα επανάληψης είναι προαιρετικό. Αν δεν το ορίσουμε, από την Python είναι 1. Ακόμη, το βήμα μπορεί να είναι αρνητικός αριθμός, αλλά όχι δεκαδικός ή μηδέν. Στην περίπτωση που είναι μηδέν τότε ο βρόχος εκτελείται επ' άπειρον.

Αν το βήμα είναι αρνητικό θα πρέπει η αρχική τιμή να είναι μεγαλύτερη από την τελική τιμή. Για παράδειγμα:

```
for i in range(100,1,-10):  
    print i
```

Εδώ θα εκτυπωθούν οι αριθμοί 100, $(100-10)=90$, $(90-10)=80$ κ.ο.κ. μέχρι να φτάσει στην πιο κοντινή στο 1 τιμή χωρίς όμως να είναι μικρότερο από την απόλυτη τιμή του βήματος, στην περίπτωση αυτή το 10.

Το πρόγραμμα που ακολουθεί βρίσκει και εκτυπώνει όλους τους άρτιους αριθμούς από το 1 μέχρι το 100.

```
for i in range(2,101,2):  
    print i
```

Στο range βρίσκονται οι αριθμοί 2, 101 και 2. Το 2 ορίζει από πού θα ξεκινήσει η μέτρηση, αφού είναι ο πρώτος άρτιος μετά το 1. Το 101 ορίζει πού θα σταματήσει, αλλά δεν θα συμπεριλάβει στην μέτρηση και το 101, και το 2 ορίζει το βήμα επανάληψης.

Το πρόγραμμα που ακολουθεί βρίσκει τον μέσο όρο 5 αριθμών που θα δώσει ο χρήστης, χρησιμοποιώντας την εντολή επανάληψης for.

```
athroisma=0  
for i in range (5):  
    num=input('Δώσε αριθμό \n')  
    athroisma=athroisma+num  
mesos_oros=athroisma/5.0  
print 'Ο μέσος όρος των αριθμών που δόθηκαν: ',mesos_oros
```

ΕΜΦΩΛΕΥΜΕΝΟΣ ΒΡΟΧΟΣ

Μπορούμε να έχουμε και έναν βρόχο μέσα σε έναν άλλο βρόχο (εμφωλευμένος βρόχος). Για παράδειγμα, έστω ότι θέλουμε να τυπώσουμε την προπαίδεια του 2 και του 3:

```
for i in range(2,4):  
    print 'Η προπαίδεια ',i
```

```
for j in range (1,11)
    print i,'*',j,'=', i*j
```

Σε αυτό το παράδειγμα θα τυπωθούν οι προπαίδειες του 2 και του 3.

Στην πρώτη εντολή επανάληψης καθορίζω ποιες προπαίδειες θέλω να εκτυπωθούν. Βάζω 2,4 για να εκτυπωθούν οι προπαίδειες του 2 και του 3.

Στη δεύτερη εντολή επανάληψης θα καθορίσω τον πολλαπλασιασμό δηλαδή από το 1 έως το 10 (γι' αυτό και γράφω 1,11). Στο δεύτερο print υπολογίζονται και εκτυπώνονται οι αριθμοί που πολλαπλασιάζονται και το γινόμενό τους.

Να γραφεί πρόγραμμα το οποίο να εκτυπώνει την προπαίδεια από το 1 έως το 10.

```
for i in range (1,11):
    print ' Η προπαίδεια του ',i,' είναι: '
    for j in range(1,11):
        print i,'*',j,'=',i*j
```

8.2. Εντολή Επανάληψης while

Τίτλος Υποενότητας	Εντολή Επανάληψης while
Τίτλος Ενότητας	Εντολές Επανάληψης
Εκπαιδευτική στρατηγική	Θα χρησιμοποιηθούν θεωρία, ασκήσεις, παραδείγματα και πρακτική εφαρμογή με ασκήσεις αυτοαξιολόγησης.
Περιγραφή	Η υποενότητα αυτή παρουσιάζει την εντολή επανάληψης while και τις διαφορές που υπάρχουν ανάμεσα σε αυτή και την εντολή επανάληψης for. Μέσα από παραδείγματα εξηγείται η σύνταξη της εντολής, καθώς και σε ποιες περιπτώσεις χρησιμοποιείται.
Γλώσσα	Ελληνικά
Μαθησιακά Αποτελέσματα	Μέσα από αυτή την υποενότητα οι εκπαιδευόμενοι: <ul style="list-style-type: none"> • Θα κατανοήσουν τη σημασία της εντολής while • Θα είναι σε θέση να χρησιμοποιήσουν την εντολή while σε δικό τους πρόγραμμα • Θα γνωρίζουν τις διαφορές ανάμεσα στην εντολή for και στη while • Θα είναι ικανοί να δώσουν παραδείγματα χρήσης της εντολής while • Θα είναι ικανοί να μετατρέψουν την εντολή for στην εντολή while και αντίστροφα

Μαθησιακά Αντικείμενα	Εντολή επανάληψης while
Αξιολόγηση Εκπαιδευόμενων	Η αξιολόγηση της υποεπάρκειας γίνεται με ένα (1) μαθησιακό αντικείμενο με ασκήσεις αυτοαξιολόγησης στο τέλος της ενότητας οι οποίες θα βασίζονται στην θεωρία με την βοήθεια παραδειγμάτων.
Λέξεις Κλειδιά	Εντολή επανάληψης, συνθήκη, εντολή, αληθής, ψευδής

8.2.1. Μαθησιακό Αντικείμενο 15 - Εντολή Επανάληψης while

Τίτλος	Εντολή επανάληψης while
Τίτλος Υποεπάρκειας	Εντολές επανάληψης
Περιγραφή	Σκοπός αυτού του μαθησιακού αντικειμένου είναι να παρουσιάσει μία από τις εντολές επανάληψης, συγκεκριμένα την εντολή while. Μέσα από αυτή την παρουσίαση γίνεται κατανοητή η έννοια της εντολής, αλλά και πώς αυτή συντάσσεται. Επίσης περιγράφεται η έννοια «βήμα» και πώς αυτό αλλάζει σύμφωνα με τις απαιτήσεις του προγράμματος. Ακόμη, αναφέρονται οι διαφορές μεταξύ των εντολών for και while.
Γλώσσα	Ελληνικά
Μαθησιακός Τύπος	Θεωρία, ασκήσεις και παραδείγματα
Τεχνικός Τύπος	Παρουσίαση, βιντεοδιάλεξη
Λέξεις Κλειδιά	While, λογική έκφραση, δεσμευμένη λέξη, βήμα
Μαθησιακά Αποτελέσματα	Μόλις ολοκληρώσουν τη μελέτη αυτού του μαθησιακού αντικειμένου, οι εκπαιδευόμενοι θα είναι σε θέση: <ul style="list-style-type: none"> • Να γνωρίζουν πώς να συντάξουν την εντολή while • Να γνωρίζουν σε ποιες περιπτώσεις να χρησιμοποιούν τις εντολές επανάληψης που αναφέρθηκαν • Να γνωρίζουν πώς να ορίζουν το «βήμα» της επανάληψης

Η ΕΝΤΟΛΗ ΕΠΑΝΑΛΗΨΗΣ WHILE

Η εντολή επανάληψης while εκτελεί επαναλαμβανόμενα μια ομάδα εντολών, όσο μια συνθήκη που έχουμε ορίσει παραμένει αληθής. Στην εντολή while, αν χρησιμοποιούμε κάποια μεταβλητή i για έλεγχο της συνθήκης, θα πρέπει να ορίσουμε την τιμή της πριν ξεκινήσουμε την εντολή επανάληψης.

Για παράδειγμα, για την εκτύπωση αριθμών από το 0 έως το 9 γράφουμε:

```
i=0
while i<10:
    print i
    i=i+1
```

Η εντολή επανάληψης ξεκινάει με τη δεσμευμένη λέξη `while`, μία συνθήκη (οποιαδήποτε λογική έκφραση) και τελειώνει με άνω κάτω τελεία. Στη συνέχεια ακολουθεί σε εσοχή τεσσάρων κενών μια ομάδα εντολών που εκτελούνται επαναλαμβανόμενα όσο η συνθήκη είναι αληθής. Όταν η συνθήκη γίνει ψευδής, το «σώμα» δεν εκτελείται και η ροή εκτέλεσης των εντολών μεταφέρεται στην πρώτη εντολή μετά το βρόχο `while`. Αν η συνθήκη είναι ψευδής από την πρώτη φορά που ελέγχεται, το «σώμα» της επανάληψης δεν εκτελείται ποτέ.

Το «σώμα» του βρόχου θα πρέπει να αλλάζει τις τιμές της μεταβλητής κάθε φορά που γίνεται η επανάληψη, ώστε η συνθήκη να γίνεται κάποια στιγμή ψευδής και ο βρόχος να τερματίζεται. Αυτό γίνεται στην εντολή `"i=i+1"`. Διαφορετικά, ο βρόχος θα επαναλαμβάνεται επ' άπειρο, γι' αυτό και στην περίπτωση αυτή ονομάζεται ατέρμων βρόχος (`infinite loop`).

Η μεταβλητή που αναφέρεται στη συνθήκη θα πρέπει να έχει πάρει νωρίτερα στον κώδικα αρχική τιμή με τις λεγόμενες εντολές αρχικοποίησης.

Αν θέλουμε να τυπωθούν οι αριθμοί 5 έως 9, τότε πρέπει να αρχικοποιήσουμε αντίστοιχα τη μεταβλητή `i` σε 5.

```
i=5
while i<10:
    print i
    i=i+1
```

Εδώ θα αρχίσει η εκτύπωση των αριθμών από το 5.

Τι γίνεται όμως αν θέλουμε το βήμα να μην είναι 1;

Σε αυτή την περίπτωση αλλάζει στον κώδικα η γραμμή `i=i+1`. Έτσι γίνεται:

```
i=1
while i<10
    print i
    i=i+2
```

Εδώ θα εμφανιστούν οι τιμές από το 1 και θα αυξάνονται κατά δύο μέχρι το 9, δηλαδή θα εμφανιστούν οι τιμές 1, 3, 5, 7, 9.

Ισχύει και για την εντολή επανάληψης `while` ότι μπορούμε να βάλουμε αρνητικό βήμα.

ΠΑΡΑΔΕΙΓΜΑΤΑ

```
i=100
while i>1
```

```
print i
i=i-10
```

Εδώ η αρχική τιμή είναι το 100, το βήμα -10 και η τελική τιμή 1. Οι τιμές που θα εμφανιστούν είναι 100,90 κ.ο.κ. μέχρι και το 10.

Το πρόγραμμα που ακολουθεί υπολογίζει τον μέσο όρο 5 αριθμών που θα δώσει ο χρήστης, χρησιμοποιώντας την εντολή επανάληψης while.

```
i=0
athroisma=0
while i<5:
    num=input('Δώσε αριθμό\n')
    i=i+1
    athroisma=athroisma+num
mesos_oros=athroisma/5.0
print 'Ο μέσος όρος των αριθμών που δόθηκαν: ',mesos_oros
```

Σημείωση: Βάζουμε 5.0 και όχι 5 για να καλύψουμε την πιθανότητα να είναι δεκαδικό το αποτέλεσμα. Δεν έχει σημασία αν θα βάλουμε τον υπολογισμό του μέσου όρου μέσα στο σώμα της επανάληψης, το αποτέλεσμα θα είναι το ίδιο.

Το πρόγραμμα που ακολουθεί προσθέτει τους αριθμούς που θα δώσει ο χρήστης μέχρι το άθροισμα να ξεπεράσει το 100 και εμφανίζει το πλήθος των αριθμών που αθροίστηκαν.

```
i=0
count=0
while i<=100:
    num=input('Δώσε αριθμό')
    i=i+num
    count=count+1
print ' Πρόσθεσες', count,'αριθμούς να για να βγάλεις άθροισμα πάνω από 100'
```

Στο πρόγραμμα που ακολουθεί δίνεται ένας αριθμός στο διάστημα από 1 ως 100 και ο χρήστης προσπαθεί να μαντέψει ποιος είναι αυτός. Στο τέλος εκτυπώνεται πόσες προσπάθειες χρειάστηκαν μέχρι ο χρήστης να μαντέψει σωστά τον αριθμό.

```
number=0
count=0
num=78
limit1=0
```



```

limit2=100
while (num!=number):
    number=input('Μάντεψε τον αριθμό \n')
    if (number>num):
        count=count+1
        limit2=number-1
        print "Έδωσες μεγαλύτερο αριθμό, ο αριθμός που ψάχνεις είναι μεταξύ του ', limit1, '
και του ',limit2
    elif (number<num):
        count=count+1
        limit1=number+1
        print "Έδωσες μικρότερο αριθμό, ο αριθμός που ψάχνεις είναι μεταξύ του ', limit1, ' και
του ', limit2
    else :
        print 'Μάντεψες σωστά! Ο αριθμός είναι: ',number
        print 'Προσπάθησες ',count,' φορές'

```

Στο παραπάνω παράδειγμα βλέπουμε συνδυασμό εντολών επανάληψης και ελέγχου. Καθορίζουμε ποιος είναι ο αριθμός που θέλουμε να μαντέψει ο χρήστης με τη μεταβλητή num. Καθορίζουμε τα όρια στα οποία είναι το limit1 και limit2. Στην εντολή επανάληψης while βάζουμε συνθήκη όσο ο αριθμός που έβαλε ο χρήστης είναι διαφορετικός από αυτόν που ορίσαμε να συνεχίζεται η επανάληψη. Αν ο αριθμός που έδωσε ο χρήστης είναι μεγαλύτερος από τον αριθμό που έχουμε ορίσει τότε αλλάζει το πάνω όριο και γίνεται ο αριθμός που έδωσε ο χρήστης το νέο όριο. Αντίστοιχα αν είναι μικρότερος. Αυτό συνεχίζεται μέχρι είτε ο χρήστης να μαντέψει τον αριθμό, είτε με τα όρια να είναι 77, 79, όπου μένει πλέον μόνο μια επιλογή, το 78.

Αμυντικός Προγραμματισμός

Ο Αμυντικός Προγραμματισμός είναι μια προγραμματιστική τεχνική με την οποία ελέγχεται η εγκυρότητα των δεδομένων εισόδου. Δηλαδή, όταν ο χρήστης πληκτρολογεί την είσοδο, ελέγχεται αν αυτή είναι έγκυρη. Αν δεν είναι έγκυρη, τότε ζητείται από τον χρήστη να την πληκτρολογήσει ξανά. Αν είναι ορθή, τότε προχωράμε στην επεξεργασία των δεδομένων εισόδου.

Παράδειγμα:

```

#ζητάμε από τον χρήστη να εισάγει τον μήνα που επιθυμεί
m=input('Εισάγετε μήνα')
#ελέγχουμε αν αυτό που εισήγαγε δεν είναι μεταξύ του 1 και του 12

```

```

while (m<1 or m>12) :
    print 'Λάθος είσοδος'
    print ' Εισάγετε σωστό μήνα'
    m=input('Εισάγετε μήνα')

```

Σύγκριση for και while

- Στην εντολή for ο αριθμός των επαναλήψεων είναι προκαθορισμένος, ενώ στην εντολή while δεν είναι πάντα.
- Στην εντολή while υπάρχουν περιπτώσεις που δεν μπορούμε να αποδείξουμε διαβάζοντας απλά τον κώδικα αν θα έχουμε ή όχι ατέρμονα βρόχο κατά την εκτέλεση.
- Συνήθως η while μας δίνει πιο πολύπλοκο κώδικα.

8.3. Εντολές “αναγκαστικής εξόδου” break και continue

Τίτλος Υποενότητας	Εντολές “αναγκαστικής εξόδου” break και continue
Τίτλος Ενότητας	Εντολές επανάληψης
Εκπαιδευτική Στρατηγική	Θα χρησιμοποιηθούν θεωρία, ασκήσεις, παραδείγματα και πρακτική εφαρμογή με ασκήσεις αυτοαξιολόγησης.
Περιγραφή	Η υποενότητα αυτή παρουσιάζει τις εντολές “αναγκαστικής εξόδου” break και continue. Αναφέρεται πώς συντάσσονται και μέσα από παραδείγματα φαίνεται η χρησιμότητά τους.
Γλώσσα	Ελληνικά
Μαθησιακά Αποτελέσματα	Μέσα από αυτή την υποενότητα οι εκπαιδευόμενοι: <ul style="list-style-type: none"> • Θα κατανοήσουν τη χρησιμότητα των εντολών break και continue • Θα γνωρίζουν πώς να χρησιμοποιούν την εντολή break • Θα γνωρίζουν πώς να χρησιμοποιούν την εντολή continue
Μαθησιακά Αντικείμενα	Εντολές break και continue
Αξιολόγηση Εκπαιδευόμενων	Η αξιολόγηση της υποενότητας γίνεται με ένα (1) μαθησιακό αντικείμενο με ασκήσεις αυτοαξιολόγησης στο τέλος της ενότητας οι οποίες θα βασίζονται στην θεωρία με την βοήθεια παραδειγμάτων.
Λέξεις Κλειδιά	Εντολές, βρόχος, έξοδος, συνθήκη

8.3.1. Μαθησιακό Αντικείμενο 16 - Εντολές “αναγκαστικής εξόδου” break και continue

Τίτλος	Εντολές “αναγκαστικής εξόδου” break και continue
Τίτλος Υποενότητας	Εντολές break και continue
Περιγραφή	Σε αυτό το μαθησιακό αντικείμενο, παρουσιάζονται οι εντολές break και continue. Πιο συγκεκριμένα, αναφέρονται τα οφέλη κάθε εντολής και πώς αυτές συντάσσονται.
Γλώσσα	Ελληνικά
Μαθησιακός Τύπος	Θεωρία, ασκήσεις και παραδείγματα
Τεχνικός Τύπος	Παρουσίαση, βιντεοδιάλεξη
Λέξεις Κλειδιά	Break, continue, for, while, true
Μαθησιακά Αποτελέσματα	Όταν οι εκπαιδευόμενοι ολοκληρώσουν τη μελέτη αυτού του μαθησιακού αντικειμένου: <ul style="list-style-type: none"> • Θα γνωρίζουν ποια από τις δυο εντολές να χρησιμοποιήσουν σύμφωνα με τις απαιτήσεις του προγράμματος • Θα γνωρίζουν πώς να χρησιμοποιήσουν την εντολή break • Θα γνωρίζουν πώς να χρησιμοποιήσουν την εντολή continue

ΕΝΤΟΛΕΣ BREAK ΚΑΙ CONTINUE

Χρησιμοποιούμε την εντολή **break** όταν θέλουμε να “σπάσει” ο κώδικας και να έχουμε άμεση έξοδο από τον βρόχο.

Παράδειγμα: Έστω ότι θέλουμε να αθροίσουμε ένα άγνωστο πλήθος αριθμών.

```
athroisma=0
```

```
while True:
```

```
    num=input ('Δώσε έναν θετικό αριθμό, για έξοδο πληκτρολόγησε -1')
```

```
    if num== -1:
```

```
        break
```

```
    athroisma=athroisma+num
```

```
print 'Το άθροισμα των θετικών αριθμών που έδωσες είναι: ',sum
```

Στο παράδειγμα αυτό, η συνθήκη του βρόχου while είναι True και αυτό σημαίνει ότι είναι πάντα αληθής και ο βρόχος θα επαναλαμβάνεται διαρκώς εκτός αν εκτελεστεί η εντολή break. Ο μόνος τρόπος να εκτελεστεί η break είναι να δώσουμε ως είσοδο τον αριθμό -1.

Η Python διαθέτει και μία άλλη σχετική εντολή, την **continue**.

Παράδειγμα: έστω ότι θέλουμε να βρούμε τους περιττούς και άρτιους αριθμούς από το 0 έως το 9.

```
for num in range (0,10):
```

```
    if num%2==0
```

```
        print 'Ο αριθμός ',num,' είναι άρτιος
```

```
        continue
```

```
    print 'Ο αριθμός ', num,' είναι περιττός
```

Εδώ βλέπουμε ότι χρησιμοποιήσαμε την εντολή επανάληψης for και τη συνθήκη if. Αν ο αριθμός που δίνεται από την επανάληψη ικανοποιεί τη συνθήκη (δηλαδή το υπόλοιπο της διαίρεσής του με το 2 είναι 0, άρα ο αριθμός είναι άρτιος) τότε προχωράει κανονικά, εμφανίζει το ανάλογο μήνυμα και συνεχίζει την επανάληψη, δηλαδή παίρνει τον αμέσως επόμενο αριθμό. Αν δεν ικανοποιεί τη συνθήκη τότε προχωράει και εκτυπώνει το επόμενο μήνυμα αφού υπάρχουν μόνο δύο συνθήκες.

8.4. Ασκήσεις 5ης Ενότητας

Ερωτήσεις Πολλαπλής Επιλογής (Multiple Choice)

1. Οι εντολές

```
for i in range(10)
```

```
    print i
```

- a) Θα εκτυπώσουν όλους τους αριθμούς από το 1 έως 10
- b) Θα εκτυπώσουν όλους τους αριθμούς από το 1 έως 9
- c) Δεν θα εκτυπώσουν τίποτα γιατί υπάρχει συντακτικό λάθος

2. Μια ομάδα εντολών που επαναλαμβάνονται ονομάζεται:

- a) Βρόχος
- b) Εντολές
- c) Δεν έχει ονομασία

3. Όταν θέλω η μέτρηση των επαναλήψεων να ξεκινά από το 10 και να καταλήγει στο 20, τότε χρησιμοποιώ την εντολή for ως εξής:

- a) for i in range (10,20)
- b) for i in range (10,21)
- c) for i in range (20,10)

4. Όταν θέλω η μέτρηση επαναλήψεων να ξεκινά από το 10, να καταλήγει στο 20 και να έχει βήμα 2, τότε χρησιμοποιώ την εντολή for ως εξής:
- a) for i in range (10,21,2)
 - b) for i in range (2,10,20)
 - c) for i in range (10,20,2)
5. Το βήμα για την εντολή επανάληψης while
- a) Δεν υπάρχει απαραίτητα
 - b) Είναι όπως στην εντολή επανάληψης for.
 - c) $i=i+\text{βήμα}$

Ερωτήσεις Σωστό - Λάθος:

- 1. Η for δεν είναι δεσμευμένη λέξη
- 2. Η range είναι λίστα
- 3. Το βήμα δεν μπορεί να είναι αρνητικός αριθμός
- 4. Το βήμα δεν μπορεί να είναι δεκαδικός αριθμός
- 5. Το βήμα είναι προαιρετικό
- 6. Η εντολή επανάληψης while εκτελεί τις εντολές όσο η συνθήκη είναι αληθής
- 7. Ένας βρόχος μπορεί να εκτελείται επ' άπειρον
- 8. Στην εντολή επανάληψης while πρέπει να αρχικοποιήσω τη μεταβλητή i που καθορίζει την τιμή της συνθήκης
- 9. Για να αλλάξει το βήμα με την εντολή επανάληψης while πρέπει να γράψω $i=i+\text{βήμα}$
- 10. Στην εντολή while ο αριθμός των επαναλήψεων είναι προκαθορισμένος
- 11. Στην εντολή for ο αριθμός των επαναλήψεων είναι προκαθορισμένος
- 12. Χρησιμοποιούμε την εντολή break για να σπάσει ο κώδικας
- 13. Αν το βήμα είναι αρνητικό τότε η τελική τιμή πρέπει να είναι μεγαλύτερη από την αρχική
- 14. Το βήμα για την εντολή επανάληψης while είναι προαιρετικό
- 15. Αν το βήμα σε μια εντολή επανάληψης είναι μηδέν, τότε εκτελείται επ' άπειρον

9. ΕΝΟΤΗΤΑ 6 - ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Τίτλος Ενότητας	Δομές δεδομένων
Τίτλος Μαθήματος	Βασικές αρχές προγραμματισμού στη γλώσσα Python
Περιγραφή	Στην ενότητα αυτή θα παρουσιαστεί η «έννοια δομές δεδομένων» και τρεις κατηγορίες δομών δεδομένων που χρησιμοποιούνται στην Python. Στις υποενότητες που ακολουθούν θα εξηγηθούν χαρακτηριστικά αυτών των δομών, καθώς και η χρησιμότητά τους. Αναλύεται επίσης πώς γίνεται η σύνταξη για κάθε ιδιότητα που εμφανίζουν αυτές οι δομές.
Εκπαιδευτικοί στόχοι	<p>Στόχοι για την ενότητα αυτή είναι να μπορούν οι εκπαιδευόμενοι στο τέλος της:</p> <ul style="list-style-type: none"> • Να κατανοήσουν την έννοια «δομές δεδομένων» • Να ξεχωρίζουν κάθε κατηγορία δομών δεδομένων • Να είναι ικανοί να εφαρμόσουν κάθε ιδιότητα στα προγράμματά τους
Μαθησιακά αποτελέσματα	<p>Αφού μελετήσουν την ενότητα αυτή, οι εκπαιδευόμενοι:</p> <ul style="list-style-type: none"> • Θα μπορούν να δώσουν τον ορισμό των «δομών δεδομένων» και να ξεχωρίσουν κάθε κατηγορία • Θα είναι ικανοί να χρησιμοποιήσουν κάθε συνάρτηση που εφαρμόζεται σε κάθε κατηγορία δομών
Εκπαιδευτικές δραστηριότητες	<p>Οι εκπαιδευτικές δραστηριότητες της ενότητας είναι οι εξής:</p> <ul style="list-style-type: none"> • Λίστες • Πλειάδες • Λεξικά
Εμπλεκόμενοι Ρόλοι	Στη διάρκεια αυτής της ενότητας εμπλέκονται οι ρόλοι των σχεδιαστών της ενότητας και των εκπαιδευόμενων
Αξιολόγηση	Η ενότητα του μαθήματος αυτού χρησιμοποιεί σε κάθε υποενότητα και σε κάθε μαθησιακό αντικείμενο ασκήσεις αυτοαξιολόγησης. Τέτοιες μπορεί να είναι ασκήσεις Multiple choice ή Σωστό-Λάθος.
Συνολικός χρόνος ενότητας	1 εβδομάδα

9.1. Λίστες

Τίτλος Υποενότητας	Λίστες
Τίτλος Ενότητας	Δομές Δεδομένων
Εκπαιδευτική Στρατηγική	Θα χρησιμοποιηθούν θεωρία, ασκήσεις, παραδείγματα και πρακτική εφαρμογή με ασκήσεις αυτοαξιολόγησης.
Περιγραφή	Σε αυτή την υποενότητα παρουσιάζεται η έννοια «δομές δεδομένων» και μία από τις κατηγορίες της, οι λίστες. Αναφέρεται στις ιδιότητες που έχουν οι λίστες και πώς αυτές συντάσσονται.
Γλώσσα	Ελληνικά
Μαθησιακά Αποτελέσματα	Μέσα από αυτή την υποενότητα οι εκπαιδευόμενοι: <ul style="list-style-type: none">• Θα κατανοήσουν τη σημασία των δομών δεδομένων• Θα κατανοήσουν την έννοια της λίστας• Θα είναι ικανοί να συντάσσουν λίστες• Θα είναι ικανοί να κάνουν προσπέλαση στοιχείων σε μια λίστα• Θα είναι ικανοί να κάνουν πράξεις στις λίστες• Θα είναι ικανοί να χωρίζουν τις λίστες σε φέτες• Θα είναι ικανοί να κάνουν αφαίρεση, πρόσθεση καθώς και διαγραφή στοιχείων της λίστας• Θα είναι ικανοί να φτιάχνουν αντικείμενα και κλώνους από τις λίστες• Θα κατανοούν τις εμφωλευμένες λίστες και τους πίνακες μιας λίστας
Μαθησιακά Αντικείμενα	Λίστες
Αξιολόγηση Εκπαιδευόμενων	Η αξιολόγηση της υποενότητας γίνεται με πέντε (5) μαθησιακά αντικείμενα με ασκήσεις αυτοαξιολόγησης στο τέλος της ενότητας οι οποίες θα βασίζονται στην θεωρία με την βοήθεια παραδειγμάτων.
Λέξεις Κλειδιά	Λίστες, στοιχεία, πίνακες, συμβολοσειρές

9.1.1. Μαθησιακό Αντικείμενο 17 - Δημιουργία λίστας και προσπέλαση στοιχείων στη λίστα

Τίτλος	Δημιουργία λίστας και προσπέλαση στοιχείων στη λίστα
Τίτλος Υποενότητας	Λίστες

Περιγραφή	Σκοπός αυτού του μαθησιακού αντικειμένου είναι να παρουσιάσει την έννοια των δομών δεδομένων και μια από τις κατηγορίες της. Δίνεται η διαδικασία δημιουργίας μιας λίστας και επεξηγείται η διαδικασία προσπέλασης στοιχείων σε αυτή.
Γλώσσα	Ελληνικά
Μαθησιακός Τύπος	Θεωρία, ασκήσεις και παραδείγματα
Τεχνικός Τύπος	Παρουσίαση, βιντεοδιάλεξη
Λέξεις Κλειδιά	Λίστα, προσπέλαση στοιχείων, βήμα, range, δείκτες
Μαθησιακά Αποτελέσματα	Όταν οι εκπαιδευόμενοι ολοκληρώσουν τη μελέτη αυτού του μαθησιακού αντικειμένου: <ul style="list-style-type: none"> • Θα κατανοήσουν την έννοια «δομές δεδομένων» • Θα γνωρίζουν πώς να δημιουργήσουν μια λίστα • Θα γνωρίζουν πώς να ορίζουν δείκτες σε μια λίστα • Θα γνωρίζουν πώς να κάνουν προσπέλαση στοιχείων σε μια λίστα

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Μία δομή δεδομένων (data structure) είναι ένα σύνολο δεδομένων (τιμών) μαζί με ένα σύνολο επιτρεπτών λειτουργιών (πράξεων) επί αυτών. Οι πιο βασικές λειτουργίες είναι οι ακόλουθες: προσπέλαση, εισαγωγή, διαγραφή, αναζήτηση, ταξινόμηση, συγχώνευση και διαχωρισμός. Τα δεδομένα που χειρίζονται τα διάφορα προγράμματα που αναπτύσσουμε θα πρέπει να είναι οργανωμένα σε δομές δεδομένων. Ουσιαστικά ένα πρόγραμμα είναι ένα σύνολο αλγορίθμων και δομών δεδομένων. Η Python διαθέτει αρκετές δομές δεδομένων και παρέχει επαρκή υποστήριξη αυτών. Οι συμβολοσειρές που είδαμε στο προηγούμενο κεφάλαιο μπορούν να θεωρηθούν ως δομές δεδομένων που αποθηκεύουν αυστηρά μόνο χαρακτήρες. Άλλες χρήσιμες δομές δεδομένων που μπορούν να αποθηκεύουν δεδομένα όλων των τύπων είναι οι λίστες, οι πλειάδες και τα λεξικά.

ΛΙΣΤΕΣ

Μία λίστα (list) είναι μία συλλογή τιμών σε σειρά, οι οποίες αντιστοιχίζονται σε δείκτες. Οι τιμές που περιλαμβάνονται στη λίστα ονομάζονται στοιχεία (elements). Τα στοιχεία μιας λίστας δεν χρειάζεται να είναι ίδιου τύπου και ένα στοιχείο σε μία λίστα μπορεί να υπάρχει περισσότερες από μία φορές. Μία λίστα μέσα σε μία άλλη λίστα ονομάζεται εμφωλευμένη λίστα (nested list).

Ακόμα, τόσο οι λίστες όσο και οι συμβολοσειρές, που συμπεριφέρονται ως συλλογές τιμών, ονομάζονται ακολουθίες (sequences).

Τα στοιχεία μιας λίστας διαχωρίζονται με κόμμα και περικλείονται σε αγκύλες [και].

Μία λίστα που δεν περιέχει στοιχεία ονομάζεται κενή λίστα και συμβολίζεται με [].

Αν γράψουμε για παράδειγμα:

```
mathimata=['μαθηματικά','φυσική','έκθεση']
```

```
bathmoi=[5, 8.5, 7]
```

```
emfwleumeni=[5.2, 'χημεία', [5,6,7]]
```

```
adeialista=[]
```

Προσοχή, όταν δεν γράφουμε αριθμούς αλλά συμβολοσειρές τότε η λέξη/φράση μπαίνει πάντα μέσα σε μονά εισαγωγικά ('_')

Έτσι, αν εκτελέσουμε την εντολή:

```
print mathimata
```

θα εμφανιστεί η λίστα mathimata με όλα τα στοιχεία της και με τη σειρά που γράφηκαν.

Έτσι γίνεται και με τις υπόλοιπες λίστες.

Η Python μας προσφέρει έναν εύκολο τρόπο να δημιουργήσουμε λίστες που περιέχουν **συνεχόμενους ακέραιους αριθμούς**:

```
numbers=list(range(1,10))
```

```
print numbers
```

Εδώ αν τυπωθεί η λίστα numbers θα τυπωθούν όλοι οι αριθμοί από το 1 μέχρι το 9. Η δημιουργία της λίστας numbers μπορεί να γίνει και πιο απλά, με την εντολή:

```
numbers=range(1,10)
```

καθώς, όπως έχουμε ήδη αναφέρει στην εντολή for, η range είναι εξ ορισμού μία λίστα ακεραίων αριθμών.

Αντίστοιχα, θα μπορούσαμε να βάλουμε στη λίστα range άλλους αριθμούς. Αν βάζαμε το range(10,25) θα τυπώνονταν οι αριθμοί από το 10 μέχρι το 24. Η λίστα range έχει δύο ακεραίους ως ορίσματα και επιστρέφει την ακολουθία των ακεραίων αριθμών στο διάστημα από το πρώτο έως το δεύτερό της όρισμα, συμπεριλαμβάνοντας το πρώτο αλλά αποκλείοντας το δεύτερο.

Με ένα μόνο όρισμα, η range επιστρέφει μία ακολουθία που ξεκινάει από το μηδέν. Αν δηλαδή γράψουμε την εντολή numbers=list(range(10)) ή, απλούστερα, numbers=range(10), θα τυπωθούν όλοι οι ακέραιοι αριθμοί από το 0 έως το 9.

Αν υπάρχει και τρίτο όρισμα, τότε αυτό ορίζει το «βήμα» της ακολουθίας τιμών. Αν παραλειφθεί το τρίτο όρισμα, τότε εξ ορισμού το βήμα είναι 1.

Για παράδειγμα, αν γράψουμε την εντολή numbers=list(range(1,10,2)) ή, απλούστερα, numbers=range(1,10,2), τότε θα εμφανιστούν όλοι οι περιττοί αριθμοί από το 1 έως το 10.

Χωρίς όρισμα η list επιστρέφει άδεια λίστα.

ΠΑΡΑΔΕΙΓΜΑ

Το πρόγραμμα που ακολουθεί εκχωρεί σε μια λίστα τους αριθμούς από το 1 έως το 20. Στη συνέχεια εκτυπώνει μόνο τους περιττούς αριθμούς αυτής της λίστας.

```
numbers=list(range(1,21))
print numbers
print 'Εκτυπώνονται μόνο οι περιττοί'
i=0
while i<len(numbers):
    if i%2==1:
        print numbers[i]
    i=i+1
```

Προσπέλαση στοιχείων σε μια λίστα

Κάθε ακέραια έκφραση μπορεί να χρησιμοποιηθεί ως δείκτης.

Έστω μια λίστα `numbers=[14, 30, 45, 53]`. Θέτουμε τον δείκτη του πρώτου αριθμού στην λίστα 0 (Προσοχή! ΟΧΙ 1) και συνεχίζουμε μέχρι να αριθμήσουμε όλους τους δείκτες. Στον πίνακα βλέπουμε στην πάνω γραμμή τους αριθμούς της λίστας και στην κάτω γραμμή τους δείκτες. Έτσι εκπροσωπείται ο κάθε αριθμός χωρίς να ξέρουμε απαραίτητα ποιος είναι. Σημειώνεται ότι δεν χρειάζεται να εκτελεστεί κάποια εντολή για να θέσουμε δείκτες στο πρόγραμμα γιατί αυτό γίνεται αυτόματα. Εδώ γίνεται μόνο για να μην υπάρχει σύγχυση.

14	30	45	53
0	1	2	3

Έστω λοιπόν ότι θέλουμε να εκτυπώσουμε τον αριθμό που βρίσκεται 3^{ος} στη λίστα. Ο δείκτης του είναι το 2. Έτσι:

```
print numbers[2]
```

Αν προσπαθήσουμε να προσπελάσουμε στοιχείο που δεν υπάρχει, π.χ. στη θέση 5, τότε θα εμφανιστεί μήνυμα λάθους γιατί δεν υπάρχει τέτοιος δείκτης.

Συχνά μία μεταβλητή βρόχου χρησιμοποιείται για την προσπέλαση όλων των στοιχείων μιας λίστας (πέρασμα λίστας). Για παράδειγμα:

```
katoikidia=['σκύλος', 'γάτα', 'παπαγάλος']
i=0
while i<len(katoikidia):
    print katoikidia[i]
    i=i+1
```

Όταν εκτελεστεί αυτό το πρόγραμμα θα εμφανίσει όλες τις τιμές της λίστας, τη μία κάτω από την άλλη, σε αυτή την περίπτωση τα κατοικίδια.

Η συνάρτηση `len` χρησιμοποιείται για να μετράει το σύνολο των τιμών που έχει η λίστα (πλήθος των τιμών).

Εναλλακτικά, για το πέρασμα μιας λίστας, μπορούμε να χρησιμοποιήσουμε βρόχο `for`. Για παράδειγμα:

```
numbers= [14, 30, 45, 53]
```

```
for numbers in numbers:
```

```
    print numbers
```

Με τον λογικό τελεστή `in` μπορούμε να ελέγξουμε αν μια τιμή ανήκει σε μια λίστα. Ο τελεστής αυτός λειτουργεί όπως και στις συμβολοσειρές. Μπορούμε, επίσης, να χρησιμοποιήσουμε την έκφραση `not in`. Για παράδειγμα:

```
numbers= [14, 30, 45, 53]
```

```
print 15 in numbers
```

```
print 16 not in numbers.
```

Η πρώτη εκτύπωση θα εμφανιστεί `False`, γιατί δεν υπάρχει ο αριθμός 15 στην λίστα. Στην δεύτερη εκτύπωση θα εμφανιστεί `True`, γιατί όντως το 16 δεν είναι στην λίστα.

9.1.2. Μαθησιακό Αντικείμενο 18 - Πράξεις μέσα σε μια λίστα και φέτες λίστας

Τίτλος	Πράξεις μέσα σε μια λίστα και φέτες λίστας
Τίτλος Υποενότητας	Λίστες
Περιγραφή	Σκοπός αυτού του μαθησιακού αντικειμένου είναι να εξηγήσει πώς γίνονται πράξεις μέσα σε μια λίστα και να παρουσιάσει την έννοια της «φέτας» από λίστα και πώς υλοποιείται.
Γλώσσα	Ελληνικά
Μαθησιακός Τύπος	Θεωρία, ασκήσεις και παραδείγματα
Τεχνικός Τύπος	Παρουσίαση, βιντεοδιάλεξη
Λέξεις Κλειδιά	Λίστα, πράξεις, τελεστές, φέτες, δείκτης
Μαθησιακά Αποτελέσματα	Όταν οι εκπαιδευόμενοι ολοκληρώσουν τη μελέτη αυτού του μαθησιακού αντικειμένου: <ul style="list-style-type: none">• Θα γνωρίζουν πώς να κάνουν πράξεις σε μια λίστα• Θα γνωρίζουν πώς να χωρίζουν μια λίστα σε φέτες

Πράξεις σε λίστες

Με τον τελεστή + μπορούμε να συνενώσουμε λίστες (αλληλουχία) και με τον τελεστή * να επαναλάβουμε μια λίστα. Για παράδειγμα:

```
a=[1,2,3]
```

```
b=[4,5]
```

```
c=a+b
```

```
print c
```

Θα εκτυπωθεί [1,2,3,4,5]

```
5*[1]
```

Θα εκτυπωθεί [1,1,1,1,1]

Φέτες από λίστες

```
mylist=[a,b,c,d,e,f]
```

```
mylist[1:3]
```

Εδώ θα εκτυπωθεί ['b','c']

```
mylist[:4]
```

Εδώ θα εκτυπωθούν όλες οι τιμές μέχρι τον δείκτη 4 (χωρίς αυτόν), δηλαδή ['a','b','c','d']

```
mylist[3:]
```

Εδώ θα εκτυπωθούν όλες οι τιμές μετά τον δείκτη 3 δηλαδή ['d','e','f']

```
mylist[:]
```

Εδώ θα εκτυπωθούν όλες οι τιμές της λίστας.

Οι λίστες είναι μετατρέψιμες (mutable). Μπορούμε δηλαδή εύκολα να αλλάξουμε στοιχεία μιας λίστας. Για παράδειγμα:

```
katokidia=[‘σκύλος’, ‘γάτα’, ‘παπαγάλος’]
```

```
katokidia[1]=‘χρυσόψαρο’
```

```
print katokidia
```

Εδώ θα εκτυπωθεί η λίστα katokidia: σκύλος, χρυσόψαρο, παπαγάλος

9.1.3. Μαθησιακό Αντικείμενο 19 - Αφαίρεση, πρόσθεση και διαγραφή στοιχείων σε μια λίστα

Τίτλος	Αφαίρεση, πρόσθεση και διαγραφή στοιχείων σε μια λίστα
Τίτλος Υποενότητας	Λίστες

Περιγραφή	Σκοπός αυτού του μαθησιακού αντικειμένου είναι να εξηγήσει πώς γίνεται η αφαίρεση στοιχείων δημιουργώντας μια άδεια λίστα μέσα στην ήδη υπάρχουσα, πρόσθεση επιπλέον στοιχείων και διαγραφή των στοιχείων μιας λίστας, χρησιμοποιώντας μια λέξη κλειδί της Python.
Γλώσσα	Ελληνικά
Μαθησιακός Τύπος	Θεωρία, ασκήσεις και παραδείγματα
Τεχνικός Τύπος	Παρουσίαση, βιντεοδιάλεξη
Λέξεις Κλειδιά	Αφαίρεση, πρόσθεση, del, άδεια λίστα
Μαθησιακά Αποτελέσματα	Όταν οι εκπαιδευόμενοι ολοκληρώσουν τη μελέτη αυτού του μαθησιακού αντικειμένου: <ul style="list-style-type: none"> • Θα γνωρίζουν πώς να αφαιρέσουν ένα στοιχείο από μια λίστα • Θα γνωρίζουν πώς να προσθέσουν ένα στοιχείο στη λίστα • Θα γνωρίζουν πώς να διαγράψουν ένα στοιχείο από τη λίστα

Αφαίρεση και προσθήκη στοιχείων σε μια λίστα

Μπορούμε να αφαιρέσουμε στοιχεία από μια λίστα εκχωρώντας τους μια άδεια λίστα. Για παράδειγμα:

```
mylist = ['a', 'b', 'c', 'd', 'e', 'f']
```

```
mylist [1:3]= []
```

```
print mylist
```

Σε αυτή την εκτύπωση τα αποτελέσματα που θα βγάλει, είναι το a,d,e,f, αφού στη λίστα οι χαρακτήρες με δείκτες τα 1 έως 3 αντικαταστάθηκαν με κενό.

Παρομοίως, μπορούμε και να προσθέσουμε στοιχεία σε μια λίστα με φέτες από λίστες όπως είδαμε προηγουμένως. Παράδειγμα:

```
mylist = ['a', 'd', 'e', 'f']
```

```
mylist[1:1]= ['b', 'c']
```

```
print mylist
```

Στην εκτύπωση θα εμφανιστούν με την σειρά τα: a, b, c, d, e, f, αφού βάλαμε στον δείκτη 1 δύο χαρακτήρες, επομένως οι επόμενοι δείκτες άλλαξαν.

Διαγραφή στοιχείων από τη λίστα

Για να διαγράψουμε στοιχεία από μια λίστα χρησιμοποιούμε τη λέξη κλειδί `del`. Για παράδειγμα:

```
mylist = ['a', 'b', 'c', 'd', 'e', 'f']  
del mylist[5]  
print mylist
```

Στα αποτελέσματα της εκτύπωσης, παρατηρούμε ότι ο χαρακτήρας με δείκτη το 5, δηλαδή το 'f', δεν εμφανίζεται.

Αν από κάτω γράψουμε:

```
del mylist[1:3]  
print mylist
```

παρατηρούμε ότι οι χαρακτήρες με δείκτες 1 και 2 δεν εμφανίζονται και μένουν μόνο τα 'a', 'd' και 'e'.

Στο πρόγραμμα που ακολουθεί εκχωρούνται σε μια λίστα 10 αριθμοί και εκτυπώνονται:

α) Όλοι οι αριθμοί, β) Όλοι οι αριθμοί μετά τον 5^ο δείκτη, γ) Όλοι οι αριθμοί πριν τον 3^ο δείκτη, δ) Ο αριθμός που βρίσκεται στον 10^ο δείκτη.

Τέλος, διαγράφονται όλα τα στοιχεία της λίστας εκτός από τα τρία τελευταία.

```
numbers=list(range(1,11))  
print numbers          #(α)  
print numbers[5:]     #(β)  
print numbers[:2]     #(γ)  
print numbers[9]      #(δ)  
del numbers[0:7]  
print numbers
```

9.1.4. Μαθησιακό Αντικείμενο 20 - Αντικείμενα, ψευδώνυμα και λίστες κλώνοι

Τίτλος	Αντικείμενα, ψευδώνυμα και λίστες κλώνοι
Τίτλος Υποενότητας	Λίστες
Περιγραφή	Σκοπός αυτού του μαθησιακού αντικειμένου είναι να εξηγήσει την έννοια του αντικειμένου και των ψευδωνύμων στην Python μέσα από παραδείγματα. Επίσης, δίνεται μια τεχνική επαλήθευσης για να ξεχωρίζουν ταυτόσημα αντικείμενα. Ακόμα δίνεται ο τρόπος δημιουργίας «κλώνων» μιας λίστας.

Γλώσσα	Ελληνικά
Μαθησιακός Τύπος	Θεωρία, ασκήσεις και παραδείγματα
Τεχνικός Τύπος	Παρουσίαση, βιντεοδιάλεξη
Λέξεις Κλειδιά	Αντικείμενα, ψευδώνυμα, κλώνοι
Μαθησιακά Αποτελέσματα	Όταν οι εκπαιδευόμενοι ολοκληρώσουν τη μελέτη αυτού του μαθησιακού αντικειμένου: <ul style="list-style-type: none"> • Θα κατανοούν την έννοια του αντικειμένου • Θα γνωρίζουν πώς να ξεχωρίζουν δύο αντικείμενα • Θα γνωρίζουν πώς να κλωνοποιούν μια λίστα ώστε να κάνουν αλλαγές στον κλώνο

Αντικείμενα και ψευδώνυμα

Όλες οι τιμές στην Python αποθηκεύονται στη μνήμη του υπολογιστή με τη μορφή αντικειμένων. Είναι πολύ πιθανό δύο μεταβλητές να έχουν τις ίδιες τιμές, αλλά να αναφέρονται σε διαφορετικά αντικείμενα. Είναι δηλαδή ισότιμες αλλά όχι ταυτόσημες. Ας δούμε ένα παράδειγμα:

```
list1=[a, b, c]
```

```
list2=[a, b, c]
```

Αυτές οι δύο λίστες, αν και έχουν τις ίδιες τιμές, είναι διαφορετικά αντικείμενα. Αυτό θα άλλαζε αν εκχωρούσαμε την list2 ως list1. Ας δούμε το παράδειγμα:

```
list1=[a, b, c]
```

```
list2=list1
```

Ξέρουμε ότι όταν εκχωρούμε μια μεταβλητή μέσα σε μια άλλη αυτομάτως παίρνουν τις ίδιες τιμές. Άρα, σε αυτή την περίπτωση είναι ίδια αντικείμενα.

Υπάρχει ένας τρόπος επαλήθευσης για να δούμε αν είναι ένα ή δύο αντικείμενα σε περίπτωση που έχουμε αμφιβολίες. Ο τελεστής is μας λέει αν δύο μεταβλητές αναφέρονται στο ίδιο αντικείμενο. Για παράδειγμα:

```
list1=[a, b, c]
```

```
list2=[a, b, c]
```

```
list1 is list2
```

Στην εκτύπωση θα μας βγάλει το μήνυμα: "False" γιατί όντως το list1 δεν είναι το list2.

```
list1=[a, b, c]
```

```
list2=list1
```

```
list1 is list 2
```

Σε αυτό το παράδειγμα είναι διαφορετικά τα πράγματα. Εκεί όντως το list1 είναι το list2 και στην εκτύπωση θα εμφανίσει "True".

Εάν μία λίστα έχει δύο ονόματα (όπως στο δεύτερο παράδειγμα όπου υπάρχει το list1, list2), λέμε ότι έχει ψευδώνυμο. Αλλαγές που γίνονται στο ένα ψευδώνυμο επηρεάζουν το άλλο. Δηλαδή αν αλλάξει μια τιμή στην list1 τότε αυτομάτως θα αλλάξει και στην list2, αφού είναι το ίδιο αντικείμενο. Παράδειγμα:

```
list1=[a, b, c]
```

```
list2=list1
```

```
list2[1]=4
```

```
print list1
```

Στην εκτύπωση θα βγει το αποτέλεσμα: a, 4, c.

Λίστες - κλώνοι

Για να κλωνοποιήσουμε μια λίστα χρησιμοποιούμε τον τελεστή φέτας. Η κλωνοποίηση δημιουργεί μια καινούρια λίστα, στην οποία μπορούμε να κάνουμε αλλαγές χωρίς να επηρεάζεται η αρχική. Για παράδειγμα:

```
list1=[a, b, c]
```

```
list2=list1[:]
```

```
print list2
```

Εδώ θα εκτυπωθεί η list2 με τις τιμές τις, οι οποίες είναι ίδιες με της list1. Συνεχίζοντας:

```
list2[0]=5
```

```
print list2
```

```
print list1
```

Στην πρώτη εκτύπωση θα εμφανιστούν: 5, b, c ενώ στη δεύτερη a, b, c.

9.1.5. Μαθησιακό Αντικείμενο 21 - Εμφωλευμένες λίστες, πίνακες και συμβολοσειρές

Τίτλος	Εμφωλευμένες λίστες, πίνακες και συμβολοσειρές
Τίτλος Υποενότητας	Λίστες
Περιγραφή	Σκοπός αυτού του μαθησιακού αντικειμένου είναι να εξηγήσει πώς δημιουργούνται πίνακες μέσα από εμφωλευμένες λίστες και πώς γίνεται η εκτύπωση ενός στοιχείου από αυτόν τον πίνακα. Επίσης, παρουσιάζονται τρόποι με τους οποίους συμβολοσειρές μετατρέπονται σε στοιχεία μιας λίστας. Τέλος, αναφέρεται και στη μέθοδο ένωσης στοιχείων μιας λίστας για τη δημιουργία συμβολοσειρών.
Γλώσσα	Ελληνικά

Μαθησιακός Τύπος	Θεωρία, ασκήσεις και παραδείγματα
Τεχνικός Τύπος	Παρουσίαση, βιντεοδιάλεξη
Λέξεις Κλειδιά	Λίστες, πίνακες, list, split, join
Μαθησιακά Αποτελέσματα	<p>Όταν οι εκπαιδευόμενοι ολοκληρώσουν τη μελέτη αυτού του μαθησιακού αντικειμένου:</p> <ul style="list-style-type: none"> • Θα γνωρίζουν πώς να δημιουργήσουν πίνακες από εμφωλευμένες λίστες • Θα γνωρίζουν τους τρόπους με τους οποίους σπάει μια συμβολοσειρά και γίνεται στοιχείο μιας λίστας • Θα γνωρίζουν πώς να δημιουργούν συμβολοσειρές, χρησιμοποιώντας τα στοιχεία μιας λίστας

Εμφωλευμένες λίστες, πίνακες

Όπως έχουμε δει, μια εμφωλευμένη λίστα είναι μια λίστα που εμφανίζεται ως στοιχείο άλλης λίστας. Μέσω μια εμφωλευμένης λίστας, μπορούμε να αναπαραστήσουμε πίνακες στην Python. Για παράδειγμα, ο παρακάτω πίνακας μπορεί να αναπαρασταθεί ως εξής:

1	2	3
4	5	6
7	8	9

```
table=[[1,2,3], [4,5,6], [7,8,9]]
```

```
print table[0]
```

```
print table[1]
```

```
print table[2]
```

Στην εκτύπωση θα εμφανιστεί ο παραπάνω πίνακας.

Αν θέλουμε να εμφανίσουμε έναν συγκεκριμένο αριθμό από τον πίνακα τότε η εντολή εκτύπωσης θα είναι η παρακάτω. Έστω ότι θέλουμε να εκτυπώσουμε τον αριθμό 8.

```
table=[[1,2,3], [4,5,6], [7,8,9]]
```

```
print table[2] [1]
```

Έτσι το [2] δηλαδή ο δείκτης 2 συμβολίζει τη λίστα 7,8,9 και ο δείκτης 1 το 8. Δείχνω δηλαδή πρώτα σε ποια εμφωλευμένη λίστα ανήκει ο αριθμός και μετά μέσα από αυτή τη λίστα τον δείκτη του χαρακτήρα που θέλουμε.

ΠΑΡΑΔΕΙΓΜΑ

Το πρόγραμμα που ακολουθεί εκτυπώνει τον παρακάτω πίνακα.

x	o	o
o	x	o
o	o	x

```
table=[['x','o','o'],['o','x','o'],['o','o','x']]
print table[0]
print table[1]
print table[2]
```

Λίστες και συμβολοσειρές

Στην Python, μπορούμε να χωρίσουμε λέξεις ή προτάσεις και να τις κάνουμε λίστες. «Σπάει» δηλαδή τις συμβολοσειρές και τις κάνει χαρακτήρες.

- Για να «σπάσουν» οι λέξεις, χρησιμοποιείται η συνάρτηση “list” όπως φαίνεται παρακάτω:

```
x='word'
list_of_characters = list(x)
print list_of_characters
```

Σαν αποτέλεσμα εκτύπωσης θα εμφανιστεί η λίστα: w, o, r, d

- Για να «σπάσουν» οι προτάσεις χρησιμοποιείται η μέθοδος “split” που η μετάφραση στα ελληνικά σημαίνει «χωρισμός». Οι χαρακτήρες space θεωρούνται όρια λέξεων.

```
phrase='Hello world'
phrase_list=phrase.split()
print phrase_list
```

Εδώ στο αποτέλεσμα της εκτύπωσης εμφανίζεται η λίστα: ‘Hello’, ‘world’

Μπορούμε επίσης να χωρίσουμε την πρόταση με ένα άλλο όρισμα όπως για παράδειγμα μια συλλαβή. Ας δούμε παράδειγμα:

```
phrase='Hello world'
phrase_list=phrase.split('wo')
print phrase_list
```

Στο αποτέλεσμα εκτύπωσης θα εμφανιστεί το ‘Hello ‘, ‘rld’ (σημειώστε ότι στην πρώτη τιμή συμπεριλαμβάνεται το κενό).

- Η μέθοδος “join” είναι η αντίστροφη της “split”. Δέχεται μία λίστα από συμβολοσειρές και συνενώνει τα στοιχεία της. Πρόκειται για μέθοδο για συμβολοσειρές, την επικαλούμαστε σε έναν οριοθέτη (delimiter) και περνάμε τη λίστα ως όρισμα. Για παράδειγμα:

```
phrase_list=['Hello', 'world']
delimiter='- '
phrase=delimiter.join(phrase_list)
```

Σαν αποτέλεσμα εκτύπωσης θα εμφανιστεί: Hello-world

ΠΑΡΑΔΕΙΓΜΑ

Το πρόγραμμα που ακολουθεί εκχωρεί μία λέξη σε μια λίστα και μετράει πόσα στοιχεία υπάρχουν στη λίστα.

```
x='python'  
list_of_characters=list(x)  
i=0  
while i<len(list_of_characters):  
    i=i+1  
print 'Η λίστα έχει ', i, ' στοιχεία '
```

Εναλλακτικά:

```
x='python'  
list_of_characters=list(x)  
print 'Η λίστα έχει ', len(list_of_characters), ' στοιχεία '
```

9.2. Πλειάδες

Τίτλος Υποενότητας	Πλειάδες
Τίτλος Ενότητας	Δομές Δεδομένων
Εκπαιδευτική στρατηγική	Θα χρησιμοποιηθούν θεωρία, ασκήσεις, παραδείγματα και πρακτική εφαρμογή με ασκήσεις αυτοαξιολόγησης.
Περιγραφή	Η υποενότητα αυτή παρουσιάζει τη δεύτερη κατηγορία των δομών δεδομένων, τις πλειάδες. Μέσα σε αυτήν εξηγείται η έννοια της πλειάδας, η διαφορά της από μια λίστα, καθώς και διάφορα χαρακτηριστικά της και πώς αυτά γράφονται στην Python.
Γλώσσα	Ελληνικά
Μαθησιακά Αποτελέσματα	Μέσα από αυτή την υποενότητα οι εκπαιδευόμενοι: <ul style="list-style-type: none">• Θα κατανοούν την έννοια της πλειάδας• Θα γνωρίζουν πώς να δημιουργήσουν μια πλειάδα• Θα κατανοούν τις διαφορές της από μια λίστα• Θα γνωρίζουν πότε να τη χρησιμοποιούν
Μαθησιακά Αντικείμενα	Πλειάδες

Αξιολόγηση Εκπαιδευόμενων	Η αξιολόγηση της υποενότητας γίνεται με ένα (1) μαθησιακό αντικείμενο με ασκήσεις αυτοαξιολόγησης στο τέλος της ενότητας οι οποίες θα βασίζονται στην θεωρία με την βοήθεια παραδειγμάτων.
Λέξεις Κλειδιά	Πλειάδα, λίστα, στοιχεία, συμβολοσειρές

9.2.1. Μαθησιακό Αντικείμενο 22 - Πλειάδες

Τίτλος	Πλειάδες
Τίτλος Υποενότητας	Δομές Δεδομένων
Περιγραφή	Σε αυτό το μαθησιακό αντικείμενο παρουσιάζεται η κατηγορία «πλειάδες» των δομών δεδομένων. Αναφέρεται στον ορισμό της και ορίζει τις διαφορές μιας πλειάδας από μια λίστα. Ακόμα εξηγεί πώς δημιουργείται μια πλειάδα και περιγράφει τα χαρακτηριστικά της.
Γλώσσα	Ελληνικά
Μαθησιακός Τύπος	Θεωρία, ασκήσεις και παραδείγματα
Τεχνικός Τύπος	Παρουσίαση, βιντεοδιάλεξη
Λέξεις Κλειδιά	Tuple, συμβολοσειρά, ακολουθία τιμών
Μαθησιακά Αποτελέσματα	Όταν ολοκληρώσουν τη μελέτη αυτού του μαθησιακού αντικείμενου, οι εκπαιδευόμενοι θα είναι σε θέση: <ul style="list-style-type: none"> • Να κατανοούν την έννοια της πλειάδας • Να δημιουργούν μια πλειάδα • Να κατανοούν τις διαφορές της από μια λίστα • Να γνωρίζουν πώς να κάνουν πράξεις μέσα σε μια πλειάδα • Να γνωρίζουν τις συναρτήσεις in, len • Να γνωρίζουν την συνάρτηση tuple

ΠΛΕΙΑΔΕΣ

Μία πλειάδα (tuple) είναι μια ακολουθία τιμών με συγκεκριμένη σειρά, οι οποίες αντιστοιχίζονται σε δείκτες. Οι τιμές που είναι μέλη μιας πλειάδας ονομάζονται στοιχεία (elements) και μπορεί να είναι οποιουδήποτε τύπου (αριθμοί, συμβολοσειρές, λίστες, πλειάδες). Οι πλειάδες μοιάζουν με τις λίστες στη χρήση δεικτών, στον τρόπο με τον οποίο διατρέχονται και στη χρήση του τελεστή φέτας. Όμως οι πλειάδες, όπως και οι συμβολοσειρές, είναι αμετάβλητες. Οι πλειάδες αξιοποιούνται συνήθως στις περιπτώσεις όπου πρόκειται να χρησιμοποιηθεί μια ακολουθία τιμών (πλειάδα) που δεν πρόκειται να αλλάξει.

Συντακτικά, μια πλειάδα είναι μια λίστα τιμών που χωρίζονται με κόμμα. Παρ' όλο που δεν είναι αναγκαίο, είναι σύνηθες να περικλείουμε τις πλειάδες με παρενθέσεις. Για παράδειγμα:

```
example=(1,2,3,4,5)
```

```
print example
```

```
1,2,3,4,5
```

Για να δημιουργήσουμε μια άδεια πλειάδα, χρησιμοποιούμε άδειες παρενθέσεις. Για να δημιουργήσουμε πλειάδα με ένα μόνο στοιχείο, πρέπει να προσθέσουμε ένα κόμμα δίπλα στο στοιχείο. Χωρίς κόμμα, η Python θεωρεί ότι πρόκειται για συμβολοσειρά μέσα σε παρενθέσεις.

Οι πράξεις πάνω σε πλειάδες είναι παρόμοιες με τις πράξεις πάνω σε λίστες. Ο τελεστής `[]` επιλέγει ένα στοιχείο από μια πλειάδα. Ο τελεστής «φέτα» επιλέγει διάστημα τιμών, όπως ακριβώς και στις λίστες. Ο τελεστής `in` ελέγχει εάν μια τιμή ανήκει σε μια πλειάδα. Η συνάρτηση `len` επιστρέφει το μήκος μιας πλειάδας (τον αριθμό των στοιχείων που περιέχει).

Οι πλειάδες είναι αμετάβλητες. Αν αλλάζαμε ένα από τα στοιχεία μιας πλειάδας, θα εμφανιζόταν μήνυμα λάθους. Για να αλλάξουμε ένα στοιχείο μπορούμε να αντικαταστήσουμε μια πλειάδα με μια άλλη, όπου θα έχουμε αλλάξει ένα στοιχείο, αλλά τα υπόλοιπα στοιχεία παραμένουν ίδια. Για παράδειγμα:

```
example= ('a','b','c')
```

```
example=('A',)+example[1:]
```

```
print example
```

Το αποτέλεσμα είναι: A,b,c

Η συνάρτηση **tuple** δέχεται ως όρισμα μια συμβολοσειρά ή μια λίστα και επιστρέφει μια πλειάδα. Με κενό όρισμα επιστρέφει μια άδεια πλειάδα. Για παράδειγμα:

```
a=tuple('example')
```

```
print a
```

```
('e', 'x', 'a', 'm', 'p', 'l', 'e')
```

9.3. Λεξικά

Τίτλος Υποενότητας	Λεξικά
Τίτλος Ενότητας	Δομές Δεδομένων
Εκπαιδευτική Στρατηγική	Θα χρησιμοποιηθούν θεωρία, ασκήσεις, παραδείγματα και πρακτική εφαρμογή με ασκήσεις αυτοαξιολόγησης.

Περιγραφή	Η υποενότητα αυτή παρουσιάζει την τρίτη κατηγορία δομών δεδομένων, τα λεξικά. Στην υποενότητα ορίζεται η έννοια «λεξικά» και αναφέρονται οι τρόποι δημιουργίας λεξικών. Εξηγείται η βασική διαφορά του λεξικού με τις λίστες και τις πλειάδες και πώς προστίθενται και αφαιρούνται στοιχεία από ένα λεξικό.
Γλώσσα	Ελληνικά
Μαθησιακά Αποτελέσματα	Μέσα από αυτή την υποενότητα οι εκπαιδευόμενοι: <ul style="list-style-type: none"> • Θα κατανοήσουν την έννοια του «λεξικού» • Θα γνωρίζουν πώς να ορίσουν ένα λεξικό • Θα γνωρίζουν δύο τρόπους δημιουργίας λεξικού • Θα γνωρίζουν πώς να προσθέτουν και να αφαιρούν στοιχεία από ένα λεξικό
Μαθησιακά Αντικείμενα	Λεξικά
Αξιολόγηση Εκπαιδευόμενων	Η αξιολόγηση της υποενότητας γίνεται με ένα (1) μαθησιακό αντικείμενο με ασκήσεις αυτοαξιολόγησης στο τέλος της ενότητας οι οποίες θα βασίζονται στην θεωρία με την βοήθεια παραδειγμάτων.
Λέξεις Κλειδιά	Λεξικά, στοιχεία, ζεύγη κλειδιών - τιμών

9.3.1. Μαθησιακό Αντικείμενο 23 - Λεξικά

Τίτλος	Λεξικά
Τίτλος Υποενότητας	Λεξικά
Περιγραφή	Σε αυτό το μαθησιακό αντικείμενο παρουσιάζεται η έννοια «λεξικά» καθώς και πώς ορίζονται. Δίνονται τρόποι δημιουργίας λεξικών και εξηγείται ο βασικός διαχωρισμός των λεξικών από τις λίστες και τις πλειάδες. Ακόμα αναφέρεται στον τρόπο με τον οποίο γίνεται η προσθήκη και αφαίρεση στοιχείων από ένα λεξικό.
Γλώσσα	Ελληνικά
Μαθησιακός Τύπος	Θεωρία, ασκήσεις και παραδείγματα
Τεχνικός Τύπος	Παρουσίαση, βιντεοδιάλεξη
Λέξεις Κλειδιά	λεξικά, κλειδί, τιμή

Μαθησιακά Αποτελέσματα	<p>Όταν οι εκπαιδευόμενοι ολοκληρώσουν τη μελέτη αυτού του μαθησιακού αντικειμένου:</p> <ul style="list-style-type: none"> • Θα γνωρίζουν πως ορίζεται ένα λεξικό • Θα γνωρίζουν τι είναι τα κλειδιά και οι τιμές • Θα γνωρίζουν πώς να προσθέτουν και να αφαιρούν στοιχεία από ένα λεξικό
------------------------	---

ΛΕΞΙΚΑ

Το λεξικό (dictionary) είναι μια δομή δεδομένων στην οποία εκχωρούμε δεδομένα, τα οποία μπορούμε να ανακτήσουμε με το όνομά τους. Κάθε στοιχείο αποτελείται από ένα “ζεύγος” κλειδί : τιμή (key : value). Το κλειδί είναι το όνομα και χωρίζεται από την τιμή με το σύμβολο ‘:’.

Κάθε κλειδί αντιστοιχίζεται σε μια τιμή και είναι μοναδικό σε ένα λεξικό. Μπορούμε να χρησιμοποιούμε μόνο αμετάβλητα αντικείμενα (όπως ακέραιους αριθμούς, συμβολοσειρές) για κλειδιά ενός λεξικού, αλλά μπορούμε να έχουμε είτε αμετάβλητα ή μετατρέψιμα αντικείμενα για τις τιμές του. Τα λεξικά είναι μετατρέψιμα και μπορούμε εύκολα να προσθέσουμε και να διαγράψουμε στοιχεία. Επιπλέον, ένα λεξικό αποτελεί μια συλλογή από στοιχεία (ζεύγη κλειδιών - τιμών), τα οποία δεν ταξινομούνται με κανέναν τρόπο (απροσδιόριστη σειρά). Δεν υπάρχει η έννοια της θέσης δείκτη και έτσι σε ένα λεξικό δεν μπορούμε να κάνουμε πέρασμα ή να χρησιμοποιήσουμε φέτες.

Για να ορίσουμε ένα λεξικό στην Python, ορίζουμε ένα όνομα για τη μεταβλητή και έπειτα μέσα σε αγκύλες βάζουμε τα κλειδιά (μέσα σε εισαγωγικά) και τις τιμές. Τα ζεύγη τα χωρίζουμε με κόμματα. Για παράδειγμα:

```
bathmoi={'μαθηματικά':5,'φυσική':8,'έκθεση':7}
```

Αν ζητήσουμε να εκτυπωθεί το λεξικό (με την εντολή `print bathmoi`) θα εμφανιστεί ό,τι έχουμε μέσα στις αγκύλες. Από την άλλη, αν θέλουμε να εκτυπωθεί μόνο ένα στοιχείο στο λεξικό, το μόνο που έχουμε να γράψουμε είναι:

```
print bathmoi['μαθηματικά'],
```

το όνομα δηλαδή της μεταβλητής και το κλειδί που θέλω από το λεξικό.

Ένας άλλος τρόπος για να ξεκινήσουμε είναι να φτιάξουμε ένα άδειο λεξικό και μετά να προσθέσουμε όσα στοιχεία θέλουμε. Παράδειγμα:

```
bathmoi={}
```

```
bathmoi['μαθηματικά']=5
```

Μπορούμε να δημιουργήσουμε ένα λεξικό και με χρήση της ενσωματωμένης συνάρτησης **dict**.

```
bathmoi=dict(μαθηματικά = 5, φυσική= 8, έκθεση= 7)
```

Ο τελεστής **del** αφαιρεί ένα ζευγάρι κλειδιού - τιμής από ένα λεξικό. Για παράδειγμα:

```
bathmoi={'μαθηματικά':5,'φυσική':8,'έκθεση':7}
```

```
del bathmoi['φυσική']
```

```
print bathmoi
```

Σε αυτή την εκτύπωση θα εμφανιστούν μόνο τα ζεύγη: 'μαθηματικά':5,'έκθεση':7

9.4. Ασκήσεις 6ης Ενότητας

Ερωτήσεις πολλαπλής επιλογής (Multiple Choice):

1. Οι τιμές που περιλαμβάνονται μέσα σε μια λίστα ονομάζονται:
 - a) Στοιχεία
 - b) Μεταβλητές
 - c) Αριθμοί
2. Η σωστή δομή μιας λίστας είναι:
 - a) `example{1,2,3}`
 - b) `example[1,2,3]`
 - c) `example=[1,2,3]`
3. Η συνάρτηση `len` χρησιμοποιείται:
 - a) Για να μετρήσει το πλήθος των τιμών μιας λίστας
 - b) Για να μετρήσει το άθροισμα των τιμών μιας λίστας
 - c) Για να ελέγξει αν μια τιμή βρίσκεται στην λίστα
4. Αν εκτελέσω `2*[6]` το αποτέλεσμα θα είναι:
 - a) Δεν θα εκτελεστεί
 - b) `[6,6]`
 - c) `[12]`
5. Αν εκτελέσω την εντολή `example[3:]` ενώ η λίστα είναι `example[5,10,15,20,25,30]` το αποτέλεσμα θα είναι:
 - a) `['15','20','25','30']`
 - b) `['5','15','20']`
 - c) `['20','25','30']`
6. Η σωστή δομή ενός λεξικού είναι:
 - a) `lexiko{bathmos1:5,bathmos2:6}`
 - b) `lexiko={bathmos1:5,bathmos2:6}`
 - c) `lexiko={'bathmos1':5,'bathmos2':6}`

Ερωτήσεις τύπου Σωστό - Λάθος:

1. Οι συμβολοσειρές θεωρούνται δομές δεδομένων
2. Τα στοιχεία της λίστας μπορούν να είναι μόνο ίδιου τύπου
3. Ένα στοιχείο σε μια λίστα μπορεί να υπάρχει μόνο μία φορά
4. Για να εκτυπωθούν τα στοιχεία μιας λίστας εκτελώ την εντολή `print onoma_tis_listas`
5. Η αρίθμηση των δεικτών ξεκινάει από το 1
6. Αν $x=[1,2]$, $y=[1,2]$ τότε το $z=x+y$ είναι $[1,2,1,2]$
7. Με την εντολή `mylist[:]` θα εκτυπωθούν όλα τα στοιχεία της λίστας
8. Οι λίστες είναι μη μετατρέψιμες
9. Είναι πιθανό δύο μεταβλητές να έχουν ίδιες τιμές
10. Για να σπάσουν οι λέξεις και κάθε γράμμα να γίνει στοιχείο μιας λίστας χρησιμοποιούμε τη συνάρτηση `split`
11. Οι πλειάδες είναι αμετάβλητες
12. Η συνάρτηση `tuple` δέχεται ως όρισμα μια συμβολοσειρά και επιστρέφει λίστα
13. Σε ένα λεξικό το κλειδί είναι το όνομα του στοιχείου και είναι μοναδικό
14. Ο τελεστής `del` αφαιρεί ένα ζευγάρι κλειδιού - τιμής από ένα λεξικό
15. Τα λεξικά είναι μια ακολουθία τιμών με συγκεκριμένη σειρά, οι οποίες αντιστοιχίζονται σε δείκτες

ΠΑΡΑΡΤΗΜΑ 1. ΑΠΑΝΤΗΣΕΙΣ ΑΣΚΗΣΕΩΝ

ΑΥΤΥΟΑΞΙΟΛΟΓΗΣΗΣ

ΕΝΟΤΗΤΑ 1

Multiple Choice:

1-c 2-b 3-a 4-b 5-a

Ερωτήσεις Σωστό - Λάθος:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Σ	Λ	Λ	Λ	Λ	Σ	Λ	Σ	Σ	Σ	Λ	Σ	Λ	Λ	Λ	Σ	Σ	Λ	Λ	Λ

ΕΝΟΤΗΤΑ 2

Multiple Choice:

1-c 2-b 3-d 4-a 5-a

Ερωτήσεις Σωστό - Λάθος:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Λ	Λ	Λ	Σ	Σ	Λ	Σ	Λ	Λ	Λ	Σ	Λ	Σ	Λ	Λ

ΕΝΟΤΗΤΑ 3

Multiple Choice:

1-b 2-a 3-a 4-a 5-b

Ερωτήσεις Σωστό - Λάθος:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Σ	Λ	Λ	Λ	Λ	Λ	Σ	Σ	Λ	Λ	Λ	Λ	Λ	Λ	Σ

Λύσεις ασκήσεων:

1. ΑΛΗΘΗΣ
2. ΨΕΥΔΗΣ
3. ΑΛΗΘΗΣ
4. ΨΕΥΔΗΣ

5. ΑΛΗΘΗΣ

6. ΑΛΗΘΗΣ

Ένας τρόπος επαλήθευσης είναι να τα τρέξετε στην Python. Για παράδειγμα, το πρώτο θα συντασσόταν ως εξής:

```
x=False
```

```
y=True
```

```
z=True
```

```
print x and y or z
```

ΕΝΟΤΗΤΑ 4

Ερωτήσεις Σωστό - Λάθος:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Σ	Λ	Λ	Λ	Λ	Λ	Σ	Σ	Λ	Λ	Λ	Λ	Σ	Σ	Λ

ΕΝΟΤΗΤΑ 5

Multiple Choice:

1-c 2-a 3-b 4-a 5-a

Ερωτήσεις Σωστό - Λάθος:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Λ	Σ	Λ	Σ	Σ	Σ	Σ	Σ	Σ	Λ	Σ	Σ	Λ	Λ	Σ

ΕΝΟΤΗΤΑ 6

Multiple Choice:

1-a 2-c 3-a 4-b 5-c 6-c

Ερωτήσεις Σωστό - Λάθος:

1	2	3	4	5	6	7	8	9	10	11	12	13	14
Σ	Λ	Λ	Σ	Λ	Σ	Σ	Λ	Σ	Λ	Σ	Λ	Λ	Σ

ΠΑΡΑΡΤΗΜΑ 2. ΠΡΟΤΥΠΟΙ ΠΙΝΑΚΕΣ

Περιγραφή Μαθήματος

Τίτλος Μαθήματος	Τίτλος Μαθήματος
Γνωστικό πεδίο	Συνοπτική περιγραφή του μαθήματος
Τύπος Μαθήματος	Γνωστικό πεδίο στο οποίο αναφέρεται το μάθημα
Συνολικός Χρόνος Μαθήματος	Καταγραφή του προβλεπόμενου συνολικού χρόνου του μαθήματος
Εκπαιδευτικό Πρόβλημα	Αναφορά στο εκπαιδευτικό πρόβλημα που αντιμετωπίζεται
Χαρακτηριστικά και Ανάγκες Εκπαιδευόμενων	Περιγραφή των χαρακτηριστικών και αναγκών των εκπαιδευόμενων
Έννοιες	Περιγραφή των βασικών εννοιών που θα διδαχθούν
Ενότητες Μάθησης	Καταγραφή των ενοτήτων μάθησης που θα διδαχθούν
Εκπαιδευτικοί Στόχοι	Περιγραφή των κύριων εκπαιδευτικών στόχων
Εργαλεία	Περιγραφή των εργαλείων που θα χρησιμοποιηθούν στο μάθημα
Απαιτήσεις Περιβάλλοντος	Αναφορά σε τυχόν απαιτήσεις του περιβάλλοντος
Εμπλεκόμενοι Ρόλοι	Περιγραφή των εμπλεκόμενων ρόλων και των ενεργειών τους
Χρονοπρογραμματισμός	Περιγραφή του χρονοπρογραμματισμού

Περιγραφή Ενοτήτων

Τίτλος Ενότητας	Τίτλος Ενότητας
Τίτλος Μαθήματος	Τίτλος Μαθήματος
Περιγραφή	Περιγραφή της ενότητας
Εκπαιδευτικοί στόχοι	Περιγραφή των εκπαιδευτικών στόχων της ενότητας
Μαθησιακά Αποτελέσματα	Περιγραφή Μαθησιακών Αποτελεσμάτων
Εκπαιδευτικές δραστηριότητες	Περιγραφή των εκπαιδευτικών δραστηριοτήτων

	που θα πλαισιώνουν τα Μαθησιακά Αποτελέσματα
Εμπλεκόμενοι Ρόλοι	Περιγραφή των εμπλεκόμενων ρόλων και των ενεργειών τους
Αξιολόγηση	Περιγραφή των μέσων και εργαλείων αξιολόγησης των εκπαιδευομένων
Συνολικός χρόνος ενότητας	Αναφορά στον χρόνο απασχόλησης με την ενότητα

Περιγραφή Υποενότητων

Τίτλος Υποενότητας	Τίτλος υποενότητας Μάθησης
Τίτλος Ενότητας	Τίτλος ενότητας που ανήκει
Εκπαιδευτική Στρατηγική	Περιγραφή υποενότητας
Περιγραφή	Περιγραφή εκπαιδευτικών στόχων της υποενότητας
Μαθησιακά Αποτελέσματα	Περιγραφή των Μαθησιακών Αποτελεσμάτων
Μαθησιακά Αντικείμενα	Περιγραφή των Μαθησιακών Αντικειμένων
Αξιολόγηση Εκπαιδευομένων	Περιγραφή των μέσων και εργαλείων αξιολόγησης εκπαιδευομένων
Λέξεις Κλειδιά	Λέξεις - Κλειδιά

Περιγραφή Μαθησιακών Αντικειμένων

Τίτλος	Τίτλος
Τίτλος Υποενότητας	Τίτλος Υποενότητας
Περιγραφή	Περιγραφή Μαθησιακού Αντικειμένου
Γλώσσα	Αναφορά της γλώσσας του Μαθησιακού Αντικειμένου
Μαθησιακός Τύπος	Αναφορά του Μαθησιακού Τύπου
Τεχνικός Τύπος	Αναφορά του Τεχνικού Τύπου
Λέξεις Κλειδιά	Λέξεις - κλειδιά
Μαθησιακά Αποτελέσματα	Αναφορά στα μαθησιακά αποτελέσματα

ΒΙΒΛΙΟΓΡΑΦΙΑ - ΑΝΑΦΟΡΕΣ

1. Wikipedia, (2015), "Massive Open Online Course", Ανακτήθηκε από http://en.wikipedia.org/wiki/Massive_open_online_course
2. <http://europestartsmooc.weebly.com/>
3. Καλογιαννάκης Μιχαήλ & Παπαδάκης Σταμάτιος: "MOOC (Massive Open Online Courses): μια νέα πρόκληση στη σύγχρονη διαδικτυακή εκπαίδευση", Ανακτήθηκε από <http://dide-peiraia.att.sch.gr/plinetp/images/stories/files/newsletter/131/MOOC.pdf>
4. <http://desarrolloweb.dlsi.ua.es/moocs/>
5. Ν. Αβούρης, Κ. Σγάρμπας, Β. Παλιουράς, Μ. Κουκιάς: "Εισαγωγή στους υπολογιστές με τη γλώσσα Python"
6. Alex Martelli: "Python in a Nutshell", Ανακτήθηκε από <http://droppdf.com/v/gNp6j>
7. <https://www.learnpython.org/>
8. <https://wiki.python.org/moin/BeginnersGuide>
9. Νικόλαος Α. Αγγελιδάκης: "Εισαγωγή στον προγραμματισμό με την Python", Ανακτήθηκε από http://aggelid.mysch.gr/pythonbook/INTRODUCTION_TO_COMPUTER_PROGRAMMING_WITH_PYTHON.pdf
10. Δημήτρης Λεβαντέας: "Οδηγός Python Μέσω Παραδειγμάτων", Ανακτήθηκε από http://python.org.gr/phocadownload/Tutorials/tutorial_by_example.pdf
11. Γεώργιος Μανής: "Εισαγωγή στον Προγραμματισμό με Αρωγό την Γλώσσα Python", Ανακτήθηκε από <https://repository.kallipos.gr/bitstream/11419/2745/2/ALL-KOY.pdf>