



**ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ**

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ  
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Υποστήριξη ομίλου Πληροφορικής Αξιοποιώντας το  
περιβάλλον του LAMS**

**Μαρία Ν. Καζαμία**

**Επιβλέπουσα : Αγορίτσα Γόγουλου**

**ΑΘΗΝΑ**

**ΟΚΤΩΒΡΙΟΣ 2021**

## **ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

Υποστήριξη Ομίλου Πληροφορικής αξιοποιώντας το περιβάλλον του LAMS

**Μαρία Ν. Καζαμία**

**A.M.: 1115201400278**

**ΕΠΙΒΛΕΠΟΝΤΕΣ: Αγορίτσα Γόγουλου**

## ΠΕΡΙΛΗΨΗ

Το LAMS αποτελεί ένα διαδικτυακό περιβάλλον που προσφέρει τη δυνατότητα σε έναν εκπαιδευτικό να δημιουργήσει ακολουθίες δραστηριοτήτων αξιοποιώντας πληθώρα εργαλείων που διαθέτει και στη συνέχεια να τις αναθέσει στους εκπαιδευόμενους για εκπόνηση. Το LAMS - Σύστημα Διαχείρισης Μαθησιακών Δραστηριοτήτων (Learning Activity Management System) - ως εργαλείο διαχείρισης μάθησης μπορεί να αξιοποιηθεί από εκπαιδευτικούς όλων των ειδικοτήτων στη σχεδίαση μαθημάτων παρέχοντας ευελιξία στα εργαλεία που θα αξιοποιηθούν και τα οποία θα υλοποιήσουν την επιθυμητή διδακτική προσέγγιση/εκπαιδευτική μέθοδο και δίνοντας τη δυνατότητα υποστήριξης διαφορετικών μαθησιακών αναγκών. Αντικείμενο της συγκεκριμένης εργασίας αποτελεί η αξιοποίηση του LAMS για τη δημιουργία εκπαιδευτικού υλικού και την υποστήριξη της εκπαιδευτικής διαδικασίας. Το εκπαιδευτικό υλικό αφορά την εκμάθηση προγραμματισμού στη γλώσσα Python και απευθύνεται σε μαθητές Λυκείου στο πλαίσιο διεξαγωγής μαθημάτων του Ομίλου Πληροφορικής. Καθ' όλη τη σχολική χρονιά 2020-21 υπήρχε διδασκαλία του αντικειμένου, εξ' αποστάσεως στα περισσότερα μαθήματα, γεγονός που οδήγησε στη σταδιακή δημιουργία του εκπαιδευτικού υλικού στο LAMS και σε συνεχείς διορθώσεις και βελτιώσεις ως προς τον σχεδιασμό των δραστηριοτήτων και την αξιοποίηση των εργαλείων του περιβάλλοντος. Μετά από κάθε μάθημα, λαμβάνοντας υπόψη τις αντιδράσεις και τα συναισθήματα των μαθητών, ακολουθούσε αναστοχασμός σχετικά με την καταλληλότητα και την αποτελεσματικότητα του σχεδιασμού και βελτίωση των επόμενων μαθημάτων στην κατεύθυνση αξιοποίησης εναλλακτικών ή συμπληρωματικών εργαλείων. Οι μαθητές εκφράστηκαν θετικά για τις δραστηριότητες και τις δυνατότητες του περιβάλλοντος και έδειξαν να μην αντιμετωπίζουν κάποια δυσκολία ως προς τη χρήση του περιβάλλοντος και την εκπόνηση δραστηριοτήτων όντας εξοικειωμένοι με την τεχνολογία. Βρήκαν ενδιαφέρουσα την ακολουθία δραστηριοτήτων και πολύ βοηθητική την οργάνωση των μαθημάτων. Για τα εξ' αποστάσεως μαθήματα, το LAMS αποτελεί εύχρηστο εργαλείο, διευκολύνει την εποπτεία του μαθήματος και δίνει κίνητρο στους μαθητές για ενασχόληση.

**ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ:** Διδακτική της Πληροφορικής

**ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ:** γλώσσα προγραμματισμού Python, LAMS, διαδικτυακό σύστημα διαχείρισης μάθησης, ακολουθία δραστηριοτήτων, σχέδιο μαθήματος

## **ABSTRACT**

LAMS is an online platform that enables a teacher to create sequences of learning activities using a variety of tools available to the system and then outsourcing them to learners. In the LAMS environment, the teacher can pass lesson plans with the material he wishes and at the same time easily modify it according to the needs of each learner. LAMS as a learning management tool can be used by teachers of all specialties as due to the tools and capabilities it offers can be used in lesson planning providing flexibility to the teacher to follow the educational method he desires and supports different learning requirements. This thesis is about the investigation of the usage of LAMS, (Learning Activity Management System), for the creation of lesson plans. The lesson plans are intended for learning programming in the Python language and is aimed to high school students in the context of conducting courses of the Informatics Group. Regarding the plan of courses for teaching Python to high school students in the context of the Informatics Group, throughout the school year 2020-21, as the teaching process includes distance courses in majority, that led to gradual creation of educational material in LAMS and in continuous corrections and improvements in the designing of learning activities and the use of the tools of the system. After each lesson, the students' reactions and feelings were taken into account, followed by a reflection on the appropriateness and effectiveness of the planning of that lesson and the following ones. The results from the performance of the activities by the students were positive in the majority. The students did not show any difficulty in using the system and doing activities as they are familiar with the technology. They found the sequence of activities interesting and very helpful as the material was gathered in this way. For e-learning, LAMS is an effective learning and supervision tool. A learning tool because it motivates students to participate and a supervision tool because it allows the teacher to know what level the students are on at each moment of the sequence of activities and to have access to their answers.

**SUBJECT AREA:** Teaching Informatics

**KEYWORDS:** Python programming language, LAMS, online learning management system, activities sequence, lesson plan

## ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ.....	3
ABSTRACT.....	4
ΠΕΡΙΕΧΟΜΕΝΑ.....	5
ΠΡΟΛΟΓΟΣ.....	7
1 ΕΙΣΑΓΩΓΗ.....	8
2 LAMS:ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΜΑΘΗΣΙΑΚΩΝ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ .....	10
2.1 Μια διαδικτυακή πλατφόρμα εκμάθησης .....	10
2.2 Τα περιβάλλοντα και οι δυνατότητες που προσφέρει στους χρήστες .....	10
2.3 Τα εργαλεία του LAMS .....	13
2.3.1 Εργαλεία Δραστηριοτήτων.....	13
2.3.2 Εργαλεία Ελέγχου Ροής .....	18
3 ΟΙ ΟΜΙΛΟΙ ΣΤΑ ΠΡΟΤΥΠΑ ΚΑΙ ΠΕΙΡΑΜΑΤΙΚΑ ΣΧΟΛΕΙΑ - Ο ΡΟΛΟΣ ΚΑΙ ΤΟ ΠΛΑΙΣΙΟ ΛΕΙΤΟΥΡΓΙΑΣ.....	20
4 Ο ΟΜΙΛΟΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΟ ΣΧΟΛΙΚΟ ΕΤΟΣ 2020-21 .....	21
4.1 Τα αντικείμενα μελέτης του Ομίλου .....	21
4.2 Σκοπός και στόχοι του Ομίλου Πληροφορικής.....	21
4.3 Οι συμμετέχοντες στα μαθήματα του Ομίλου Πληροφορικής .....	22
4.4 Το πλαίσιο λειτουργίας μαθημάτων του Ομίλου Πληροφορικής .....	22
5 ΣΧΕΔΙΑΣΗ ΜΑΘΗΜΑΤΩΝ ΔΙΔΑΣΚΑΛΙΑΣ ΡΥΘΜΩΝ ΓΙΑ ΤΟΝ ΟΜΙΛΟ ΠΛΗΡΟΦΟΡΙΚΗΣ .....	24
5.1 Συνθήκες και χαρακτηριστικά στα οποία που βασίστηκε η σχεδίαση μαθημάτων.....	24

5.2	Διδακτική Αρχή, Διδακτικές τεχνικές, Διδακτικές προσεγγίσεις που βασίστηκε η σχεδίαση μαθημάτων .....	25
5.3	Τα εργαλεία του LAMS που αξιοποιήθηκαν στη δημιουργία μαθημάτων.....	26
6	<b>ΣΧΕΔΙΑ ΜΑΘΗΜΑΤΩΝ ΣΤΟ LAMS ΚΑΙ ΑΝΑΛΥΣΗ ΠΟΡΕΙΑΣ ΜΑΘΗΜΑΤΟΣ ...</b>	<b>29</b>
6.1	Εισαγωγή στον προγραμματισμό με Python .....	29
6.2	Η Δομή Ακολουθίας στην Python .....	31
6.3	Η Δομή Επιλογής στην Python.....	34
6.4	Οι Συμβολοσειρές στην Python .....	39
6.5	Οι Λίστες στην Python.....	43
6.6	Η Δομή Επανάληψης στην Python.....	47
	<b>ΣΥΜΠΕΡΑΣΜΑΤΑ.....</b>	<b>53</b>
	<b>ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ .....</b>	<b>54</b>
	<b>ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ .....</b>	<b>55</b>
	<b>ΠΑΡΑΡΤΗΜΑ.....</b>	<b>56</b>
	<b>ΑΝΑΦΟΡΕΣ .....</b>	<b>134</b>

## ΠΡΟΛΟΓΟΣ

Αντικείμενο μελέτης για τη συγκεκριμένη εργασία αποτελεί η διδακτική της γλώσσας προγραμματισμού Python. Συγκεκριμένα αξιοποιήθηκε το σύστημα του LAMS για την δημιουργία μαθημάτων που προοριζόταν για μια ομάδα μαθητών Λυκείου και συγκεκριμένα για τον Όμιλο Πληροφορικής του 2ου Πρότυπου Γενικού Λυκείου Αθηνών. Καθώς είχα αναλάβει και τη διδασκαλία των μαθημάτων για το σχολικό έτος 2020-21, τα μαθήματα αρχικά πραγματοποιούνταν δια ζώσης στο εργαστήριο πληροφορικής του σχολείου και στη συνέχεια πραγματοποιούνταν με τη μορφή σύγχρονων εξ αποστάσεως μαθημάτων. Για την εκπόνηση της εργασίας θα ήθελα να ευχαριστήσω τη κ. Α. Γόγουλου, καθηγήτρια του Τμήματος και την κ. Α. Γασπαρινάτου, καθηγήτρια Πληροφορικής του σχολείου, για την πολύτιμη βοήθεια και στήριξή τους.

## 1 ΕΙΣΑΓΩΓΗ

Το Διαδίκτυο είναι ένα περιβάλλον δημιουργικότητας και ενεργούς μάθησης[1] το οποίο υποστηρίζει και ενθαρρύνει την πλοήγηση και εξερεύνηση προσφέροντας στους εκπαιδευόμενους πληθώρα εργαλείων μάθησης ενισχύοντας την διαδικασία της μάθησης. Ο παγκόσμιος ιστός (World Wide Web) είναι ένα αρκετά διαδεδομένο μέσο και εδώ και αρκετά χρόνια χρησιμοποιείται στην εκπαίδευση ως το κύριο μέσο ηλεκτρονικής (e-learning) αλλά και μικτής μάθησης (blended learning)[2] που δίνει την ευκαιρία στους μαθητές να ξεπεράσουν τους χρονικούς και χωρικούς περιορισμούς. Διαθέτει ποικιλία από σύγχρονα Συστήματα Διαχείρισης Μάθησης (Learning Management System, LMS) που επιτρέπουν την αναπαράσταση και διαχείριση οποιασδήποτε εκπαιδευτικής διαδικασίας[3]. Μάλιστα τα συστήματα αυτά είναι αρκετά δημοφιλή στην εκπαίδευση αφού διαθέτουν πληθώρα εργαλείων για αποθήκευση και διαμοιρασμό περιεχομένου καθώς και για άμεση επικοινωνία μεταξύ εκπαιδευτικού και εκπαιδευομένων. Επιπλέον, οι σύγχρονες πλατφόρμες ηλεκτρονικής μάθησης συμβάλλουν στην εφαρμογή της διαφοροποιημένης διδασκαλίας και την αποτελεσματικότερη σχεδίαση μαθημάτων με επίκεντρο τον μαθητή καθώς διαθέτουν το κατάλληλο περιβάλλον, μέσα και υπηρεσίες παρέχοντας στους εκπαιδευόμενους τη δυνατότητα να συμμετέχουν σε ομαδικές δραστηριότητες όπου μπορούν να συνεργάζονται, να παίρνουν ρόλους και να αλληλεπιδρούν μεταξύ τους[3]. Άλλωστε η χρήση της τεχνολογίας και η αξιοποίηση εργαλείων και μέσων ηλεκτρονικής μάθησης αποτελεί μία από τις στρατηγικές εφαρμογής της διαφοροποιημένης διδασκαλίας[4] καθώς οι μαθητές είναι πλέον εξοικειωμένοι με την τεχνολογία και ενδεχομένως να θεωρούν λιγότερο ελκυστικές τις παραδοσιακές προσεγγίσεις της διδασκαλίας.

Ο σχεδιασμός σεναρίου μαθήματος (lesson plan) από έναν εκπαιδευτικό είναι αναμφισβήτητα μία διαδικασία που απαιτεί πολύ προσπάθεια, χρόνο και ενέργεια εκτός του πλαισίου διδασκαλίας μέσα στην τάξη[5]. Ως σενάριο μαθήματος αποτελεί μία αναφορά που περιγράφει αναλυτικά τη διαδικασία της διδασκαλίας και της μάθησης που πρόκειται να διεξαχθεί σε πραγματικό χρόνο[6]. Περιγράφονται αναλυτικά τα αναμενόμενα μαθησιακά αποτελέσματα, οι μαθησιακές δραστηριότητες που θα κληθούν να εκπονήσουν οι μαθητές, οι ενέργειες που θα πραγματοποιήσει ο εκπαιδευτικός, η αλληλουχία των δραστηριοτήτων και ενεργειών και προσδιορίζονται οι εργασίες αξιολόγησης των μαθητών και τα κριτήρια αξιολόγησης μαθήματος[7]. Η δημιουργία σεναρίου μαθήματος είναι εξαιρετικά βοηθητική για όλους καθηγητές και περισσότερο για τους νέους σε εμπειρία καθηγητές οι οποίοι μέσα από τη δημιουργία σεναρίου μπορούν να είναι καλύτερα προετοιμασμένοι να είναι πιο ευέλικτοι κατά τη διάρκεια του μαθήματος και να αισθάνονται μεγαλύτερη αυτοπεποίθηση τους στη διδασκαλία τους [8].

Το LAMS ως ένα Σύστημα Διαχείρισης Μαθησιακών Δραστηριοτήτων μπορεί να αξιοποιηθεί για τη σχεδίαση σεναρίου μαθήματος αφού αποτελεί ένα πλήρως λειτουργικό εργαλείο σχεδίασης μαθημάτων και όχι απλά μία λεπτομερής περιγραφή [9]. Αποτελεί ένα εξαιρετικά βοηθητικό εργαλείο για έναν εκπαιδευτικό αφού χάρη στο γραφικό του περιβάλλον επιτρέπει και σε αυτόν και στους εκπαιδευομένους τους να έχουν μια πλήρη εικόνα του μαθήματος, σε αντίθεση με το κλασικό γραπτό σχέδιο μαθήματος [9]. Στο LAMS ένα σχέδιο μαθήματος έχει τη μορφή ενός διαγράμματος ροής αποτελούμενο από μια αλληλουχία μαθησιακών δραστηριοτήτων που διαθέτει η πλατφόρμα, επιτρέποντας στον εκπαιδευτικό να σχεδιάσει μαθήματα με εύκολο και εύχρηστο τρόπο. Επίσης το LAMS διαθέτει ποικιλία εργαλείων που μπορούν να αξιοποιηθούν από τον εκπαιδευτικό ανάλογα με την παιδαγωγική προσέγγιση που επιθυμεί να εφαρμόσει και τα διαφορετικά



επίπεδα γνώσεων των εκπαιδευομένων του [10]. Θεωρείται εξαιρετικά κατάλληλο και για το σχεδιασμό και την ανάπτυξη διαφοροποιημένων σεναρίων μάθησης σε ψηφιακή μορφή αφού τα εργαλεία διαθέτει επιτρέπουν την παροχή εναλλακτικού εκπαιδευτικού υλικού και δραστηριοτήτων σε διάφορες μορφές και συνθήκες [11]. Αυτό είναι ένα από τα πλεονεκτήματα που προσφέρει το LAMS πάνω στη σχεδίαση μαθήματος, το ότι επιτρέπει στον εκπαιδευτικό να δομήσει δραστηριότητες για διαφορετικές ομάδες εκπαιδευομένων με έναν πιο δημιουργικό τρόπο [12].

## **2 LAMS:Σύστημα Διαχείρισης Μαθησιακών Δραστηριοτήτων**

### **2.1 Μια διαδικτυακή πλατφόρμα εκμάθησης**

Το LAMS (Learning Activity Management System), Σύστημα Διαχείρισης Μαθησιακών Δραστηριοτήτων, είναι ένα δωρεάν και ανοιχτού κώδικα περιβάλλον σχεδίασης και μάθησης. Αποτελεί μία πλατφόρμα σχεδίασης, διαχείρισης, παροχής και εκτέλεσης διαδικτυακών μαθημάτων με τη μορφή ακολουθιών μαθησιακών δραστηριοτήτων. Το περιβάλλον του είναι σχεδιασμένο ώστε να απαιτείται μικρός χρόνος εκμάθησης, ενώ η δημιουργία ακολουθιών δραστηριοτήτων εκμάθησης μπορεί να γίνει με χειρισμό εικονικών αναπαραστάσεων σε ένα ιδιαίτερα φιλικό περιβάλλον συγγραφής. Πρόκειται για ένα από το πιο δημοφιλή σύγχρονα Συστήματα Διαχείρισης Μάθησης (ΣΔΜ) που υλοποιεί τις ιδέες του σχεδιασμού μάθησης (learning design)[10][12] και δίνει έμφαση τόσο στους εκπαιδευόμενους και την ενεργή συμμετοχή τους σε μαθησιακές δραστηριότητες όσο και στην υποστήριξη των νέων ρόλων του εκπαιδευτικού ως σχεδιαστή, συντονιστή και εμπυχωτή[13] της μαθησιακής διεργασίας. Η δημιουργία ακολουθιών δραστηριοτήτων είναι το κύριο χαρακτηριστικό του που το διακρίνει από άλλα Συστήματα Διαχείρισης Μάθησης και συμβάλει στην επίτευξη αποτελεσματικότερης μάθησης[10]. Ως μια διαδικτυακή πλατφόρμα εκμάθησης, προσφέρει ποικιλία εργαλείων και δραστηριοτήτων ώστε να μπορεί να αξιοποιηθεί εξίσου στη δια ζώσης, την εξ αποστάσεως αλλά και τη μικτή μάθηση, προσφέροντας τη δυνατότητα δημιουργίας, διαχείρισης και υποστήριξης σύγχρονων και ασύγχρονων ατομικών και ομαδικών δραστηριοτήτων [2].

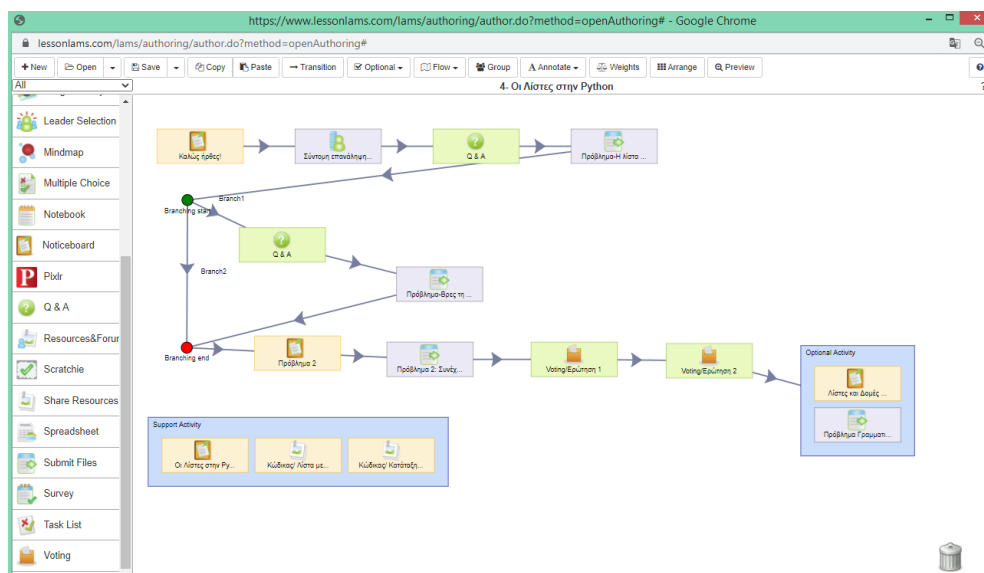
Η ανάπτυξη του LAMS γίνεται από το Macquarie E-learning Centre Of Excellence (MALCOE) στο Πανεπιστήμιο Macquarie, στο Σύδνεϋ της Αυστραλίας, με συντονιστή τον καθηγητή James Dalziel[14]. Έχει ήδη μεταφραστεί σε 30 γλώσσες μεταξύ αυτών και στα Ελληνικά. Το LAMS μπορεί να χρησιμοποιηθεί είτε ως αυτόνομο σύστημα, είτε διασυνδεδεμένο με άλλα Συστήματα Διαχείρισης Μαθημάτων όπως το Blackboard, Moodle, Sakai κ.α. Μαζί παρέχεται και ο ιστότοπος του LessonLAMS[15] όπου ένας καθηγητής μπορεί είτε να χρησιμοποιήσει τα διαθέσιμα έτοιμα υποδείγματα (templates) και τις ιδέες ψηφιακών σχεδίων μαθημάτων που διατίθενται ελεύθερα για χρήση και προσαρμογή ή να δημιουργήσει νέα μαθήματα ώστε να τα παραθέσει στους μαθητές του. Στο LessonLams το κάθε μάθημα έχει τη δική του url διεύθυνση και οι μαθητές αποκτούν πρόσβαση μέσω λογαριασμών που τους δημιουργεί ο εκπαιδευτικός. Σημαντικό ρόλο στην ανάπτυξη, υποστήριξη και εξέλιξή του διαδραματίζει η Διεθνής κοινότητα του LAMS[16]. Στην Ελλάδα, έχει δημιουργηθεί η κοινότητα μάθησης και πρακτικής Ελλήνων Εκπαιδευτικών του LAMS[17] της οποίας τα μέλη συμμετέχουν σε μια διεργασία συλλογικής μάθησης ώστε να μοιράζονται κοινά οράματα, αξίες και ενδιαφέρον για ανοιχτούς εκπαιδευτικούς πόρους. Μέσα από την κοινότητα αυτή μπορούν να ανταλλάξουν απόψεις μέσω των συζητήσεων, να προσφέρει βοήθεια ο ένας τον άλλον και μέσα από την αλληλεπίδραση να μαθαίνουν καλύτερα για αυτό.

### **2.2 Τα περιβάλλοντα και οι δυνατότητες που προσφέρει στους χρήστες**

Το LAMS ως ένα σύστημα υποστήριξης της διδασκαλίας και της μάθησης, προσφέρει στον χρήστη το περιβάλλον του συγγραφέα, του εκπαιδευομένου και του επόπτη[10]. Ένας καθηγητής έχει τη δυνατότητα να φτιάξει ολοκληρωμένα μαθήματα με τη μορφή ακολουθιακών δραστηριοτήτων (Εικόνα 1) τα οποία στη συνέχεια να παραθέσει στους μαθητές του. Ο καθηγητής, έχοντας τον ρόλο του συγγραφέα, δημιουργεί μία ακολουθία δραστηριοτήτων και στη συνέχεια το αναθέτει σε ομάδες εκπαιδευομένων, δηλαδή στους

μαθητές του. Οι μαθητές στη συνέχεια, έχοντας τον ρόλο των εκπαιδευομένων, εκπονούν τις δραστηριότητες από το προφίλ τους και ο καθηγητής, έχοντας τον ρόλο του επόπτη, εποπτεύει, υποστηρίζει, αξιολογεί και ανατροφοδοτεί τους εκπαιδευόμενους. Στο περιβάλλον του επόπτη, στο πεδίο των εκπαιδευομένων (Εικόνα 2), ο εκπαιδευτικός μπορεί να δει την πορεία του κάθε εκπαιδευομένου ξεχωριστά καθώς απεικονίζονται οι δραστηριότητες με μπλε αυτές που έχουν εκπονηθεί από τον εκάστοτε εκπαιδευόμενο, με κόκκινο η τρέχουσα δραστηριότητα στη οποία βρίσκεται, με πράσινο αυτές που δεν έχουν εκπονηθεί αλλά και στο πεδίο της ακολουθίας (Εικόνα 3) μπορεί να έχει μια συνολική εικόνα για το που βρίσκονται οι εκπαιδευόμενοι του στην ακολουθία δραστηριοτήτων. Εξαιρετικά βοηθητικό για τον εκπαιδευτικό είναι ότι στο περιβάλλον του επόπτη στο πεδίο της ακολουθίας έχει την δυνατότητα να μετακινήσει ο ίδιος τους μαθητές του σε επόμενες δραστηριότητες, χωρίς δηλαδή να έχουν εκπονηθεί οι προηγούμενες, σε περίπτωση που αυτό κριθεί αναγκαίο. Ο εκπαιδευτικός, λοιπόν, ως επόπτης μπορεί να ελέγχει τις απαντήσεις που δίνουν οι μαθητές του, να παρακολουθεί την πρόδό τους, σε ποιο σημείο της μαθησιακής πορείας βρίσκονται ακόμη και πόση ώρα χρειάστηκαν για την επίλυση των ασκήσεων. Ο εκπαιδευόμενος στο περιβάλλον του LAMS έχει τη δυνατότητα να επανέλθει σε δραστηριότητες μαθήματος που δεν έχει ολοκληρώσει ώστε να τις συνεχίσει αργότερα ή ακόμη και να ξανά μελετήσει τις ήδη ολοκληρωμένες δραστηριότητες.

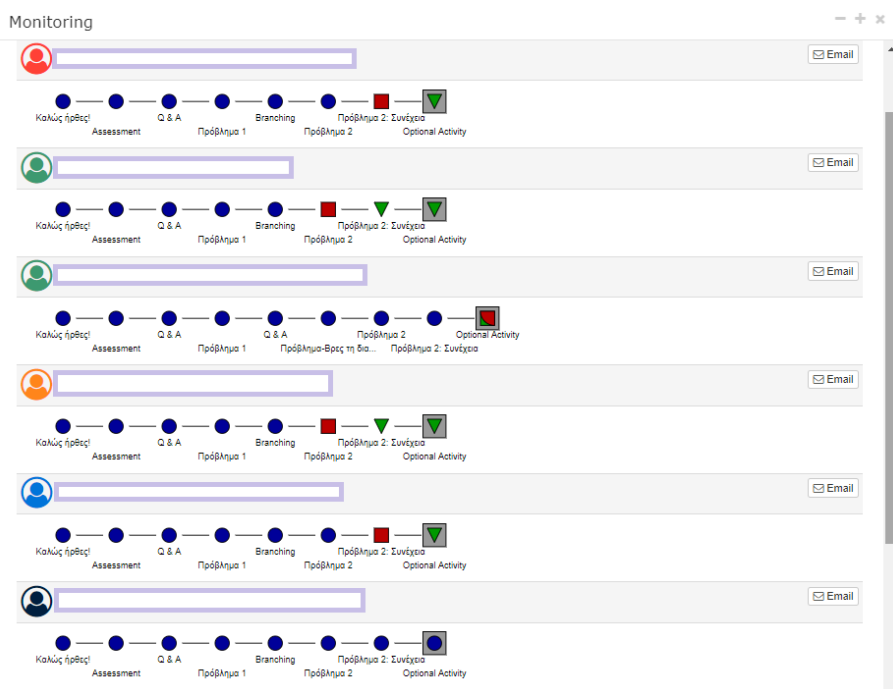
Τα μαθήματα που δημιουργεί ο καθηγητής στο LAMS μπορεί να τα αποθηκεύσει και να τα ξαναχρησιμοποιήσει στο μέλλον, για μία διαφορετική ομάδα μαθητών. Η ακολουθία μαθησιακών δραστηριοτήτων που υποστηρίζει το LAMS παρέχει ένα "παιδαγωγικό πρότυπο" στο οποίο ο εκπαιδευτικός μπορεί να βασιστεί και να διαμορφώσει εύκολα και γρήγορα το περιεχόμενο στο μέλλον ώστε να ταιριάζει για μία νέα ομάδα μαθητών. Έχει δηλαδή τη δυνατότητα να τροποποιεί και να προσαρμόζει τις ακολουθίες δραστηριοτήτων ανάλογα τη διδακτική προσέγγιση που θα επιλέξει και τις ανάγκες των μαθητών του [7].



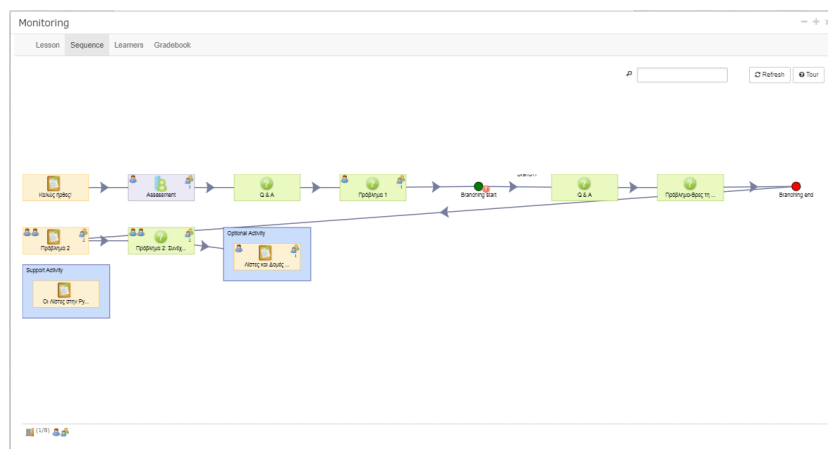
**Εικόνα 1: Το περιβάλλον συγγραφέα στο LAMS**

Το LAMS παρέχει ένα οπτικό περιβάλλον το οποίο επιτρέπει στον εκπαιδευτικό που θα σχεδιάσει ένα διαδικτυακό μάθημα να έχει πλήρη εικόνα για το πως θα φαίνεται στους εκπαιδευόμενους ώστε να προβεί σε τυχόν διορθώσεις και προσθήκες. Επίσης παρέχει ποικιλία εργαλείων προσφέροντας τη δυνατότητα σε έναν εκπαιδευτικό να εφαρμόσει ένα ολοκληρωμένο σχέδιο μαθήματος έχοντας συγκεντρωμένο όλο το εκπαιδευτικό υλικό

όπως φύλλα εργασίας, ασκήσεις, θεωρία, βίντεο, συνδέσμους διαδικτυακών πηγών και εφαρμογές. Παρέχει εργαλεία που συμβάλλουν στη σχεδίαση μιας αποτελεσματικότερης μάθησης για τους μαθητές εφαρμόζοντας τη μέθοδο της διαφοροποιημένης διδασκαλίας (differentiated learning) όπως τα εργαλεία συνεργασίας τα οποία ενθαρρύνουν τους μαθητές να σκεφτούν βαθύτερα αλληλεπιδρώντας μεταξύ τους, αλλά και τα εργαλεία διακλαδώσεων τα οποία δίνουν τη δυνατότητα στους εκπαιδευόμενους να ακολουθήσουν διαφορετικά μονοπάτια μάθησης ανάλογα των μαθησιακών τους αναγκών[18]. Επίσης λόγω των σύγχρονων και ασύγχρονων εργαλείων που προσφέρει, το LAMS μπορεί να χρησιμοποιηθεί ως το κύριο εργαλείο μάθησης εντός των σχολικών εργαστηρίων όπου την ώρα του μαθήματος οι μαθητές να εκπονοούν δραστηριότητες όλοι μαζί συντονισμένοι ή ως εργαλείο επανάληψης και ενισχυτικής διδασκαλίας στο σπίτι που ο κάθε μαθητής μπορεί να δουλεύει με τον δικό του ρυθμό. Έσω των δραστηριοτήτων του LAMS αυξάνεται η ενασχόληση των μαθητών με το μάθημα και προκαλείται το ενδιαφέρον τους[18].



**Εικόνα 2: Το περιβάλλον στο LAMS στο πεδίο των εκπαιδευομένων**



**Εικόνα 3: Το περιβάλλον του επόπτη στο LAMS στο πεδίο της ακολουθίας**

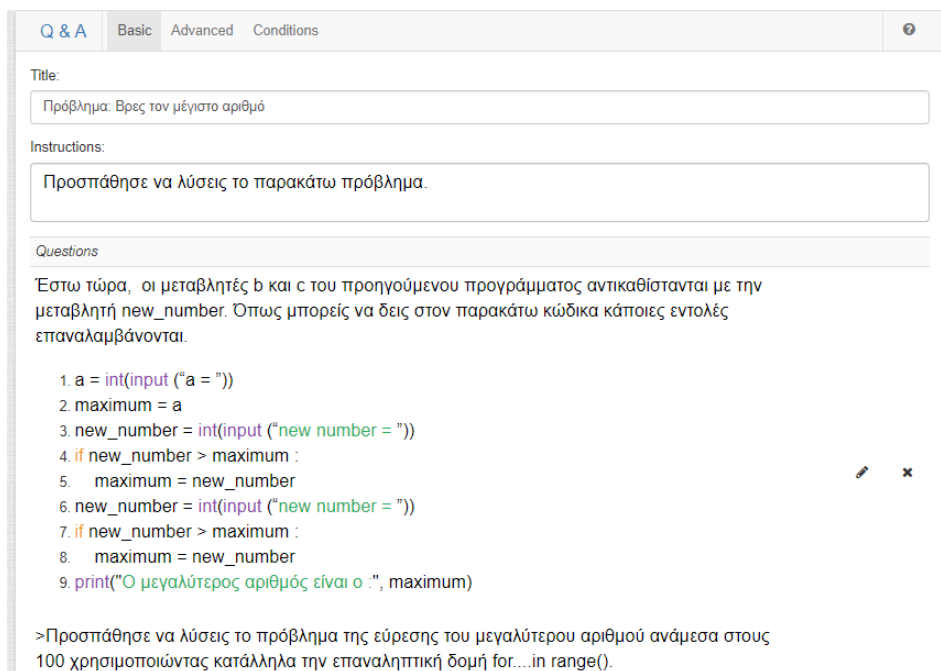
## 2.3 Τα εργαλεία του LAMS

Στο περιβάλλον του συγγραφέα είναι διαθέσιμα τα εργαλεία δραστηριοτήτων και τα εργαλεία ελέγχου ροής[19], κάθε ένα από τα οποία υποστηρίζει διαφορετικές λειτουργίες. Μέσω των εργαλείων αυτών μπορούν να δημιουργηθούν μαθήματα τα οποία εκτελούνται από τους μαθητές και εποπτεύονται από τον εκπαιδευτικό.

### 2.3.1 Εργαλεία Δραστηριοτήτων

Τα εργαλεία δραστηριοτήτων (Εικόνα 6) χρησιμοποιούνται για να δομήσουν την εκάστοτε δραστηριότητα που επιθυμεί ένας εκπαιδευτικός για το σχέδιο μαθήματος. Κάθε εργαλείο δραστηριότητας χαρακτηρίζεται από ιδιότητες όπως είναι ο τίτλος του εργαλείου και η έξοδος βαθμολογίου που αφορά στη βαθμολογία για την εκάστοτε δραστηριότητα. Τα εργαλεία των δραστηριοτήτων βρίσκονται τοποθετημένα το ένα κάτω από το άλλο αριστερά στο περιβάλλον συγγραφής. Ο εκπαιδευτικός μπορεί να προσθέσει εύκολα το μαθησιακό περιεχόμενο της δραστηριότητας που επιθυμεί στο οποιοδήποτε εργαλείο κάνοντας διπλό κλικ στο αντίστοιχο εργαλείο, κατευθύνοντάς τον στη συνέχεια σε ένα πλαίσιο με 2 ή 3 καρτέλες ανάλογα. Η καρτέλα «Βασικά» (Εικόνα 4) είναι η κύρια καρτέλα όπου μπορεί ένας καθηγητής να διαμορφώσει τη μαθησιακή δραστηριότητα που επιθυμεί εισάγοντας τον τίτλο και τις οδηγίες που θα εμφανίζονται στους εκπαιδευόμενους για την εκπόνηση της δραστηριότητας στα αντίστοιχα πεδία «Τίτλος» και «Οδηγίες» που διαθέτει η καρτέλα. Επίσης υπάρχει η καρτέλα «Προχωρημένα» (Εικόνα 5) όπου μπορεί να ορίσει τον τρόπο συμπεριφοράς του εργαλείου και η καρτέλα «Όροι» που υπάρχει σε ορισμένα εργαλεία και μπορεί να ορίσει συνθήκες σχετικά με τους εκπαιδευόμενους ανάλογα με τις απαντήσεις που έδωσαν.

Παρακάτω περιγράφονται συνοπτικά όλα τα εργαλεία δραστηριοτήτων[18] που διαθέτει το LAMS στο περιβάλλον του συγγραφέα ανάλογα με την κατηγορία στην οποία ανήκουν.



Εικόνα 4: Το πεδίο "Βασικά" στο περιβάλλον συγγραφής της δραστηριότητας "Q&A"

Εικόνα 5: Το πεδίο "Προχωρημένα" στο περιβάλλον συγγραφής της δραστηριότητας "Q&A"

### 2.3.1.1 Εργαλεία πληροφόρησης

Χρησιμοποιούνται για να παρέχουν πληροφορίες σε οποιαδήποτε μορφή όπως, κείμενο, εικόνα, ήχο, βίντεο, διεύθυνση URL, υπερμέσα και είναι τα: "Πίνακας ανακοινώσεων", "Διαμοίραση πόρων", "Λίστα εργασιών", "Χώρος έκθεσης εικόνων", "Επεξεργασία Εικόνας Richr" και "Υπολογιστικό φύλλο".

- Το εργαλείο "**Πίνακας Ανακοινώσεων**": Μέσω του οποίου μπορεί κάποιος εύκολα να παρέχει πληροφορίες στους εκπαιδευόμενους σε διάφορες μορφές όπως κείμενο, εικόνες, συνδέσμους στο διαδίκτυο όταν επιθυμεί για παράδειγμα να κάνει μια ανακοίνωση, να αναλύσει κάποιο θέμα θεωρίας ή να βάλει μια εισαγωγή ή σύνοψη του μαθήματος.
- Το εργαλείο "**Διαμοίραση Πόρων**": Μέσω του οποίου μπορεί κάποιος να διαμοιράσει πόρους στους εκπαιδευόμενους του για να μελετήσουν ως υποστηρικτικό υλικό και οι πόροι αυτοί μπορεί να είναι αρχεία κειμένων ή παρουσιάσεων καθώς και διευθύνσεις URL που θα τους παραπέμπουν να παρακολουθήσουν κάποιο βίντεο ή να επισκεφτούν κάποιο site.
- Το εργαλείο "**Λίστα Εργασιών**": Μέσω του οποίου μπορεί κάποιος να αναθέσει εργασίες στους εκπαιδευόμενους και οι οποίες μπορεί να είναι υποχρεωτικές ή προαιρετικές ανάλογα με τον σκοπό του καθηγητή. Η λίστα εργασιών μπορεί να αξιοποιηθεί από τον καθηγητή ώστε να ελέγξει αν έγιναν κατανοητά κομμάτια της ύλης που διδάχθηκαν οι εκπαιδευόμενοι.
- το εργαλείο "**Χώρος έκθεσης εικόνων**": Μέσω του οποίου ο εκπαιδευτικός μπορεί να προβάλλει εικόνες στους εκπαιδευόμενους ή να ζητήσει από τους ίδιους να τοποθετήσουν εικόνες στο εργαλείο αυτό.
- Το εργαλείο "**Υπολογιστικό φύλλο**": Πρόκειται για εργαλείο με παρόμοια λειτουργία με αυτή του προγράμματος επεξεργασίας υπολογιστικών φύλλων σαν το Excel. Το υπολογιστικό φύλλο μπορεί να αξιοποιηθεί από τον εκπαιδευτικό ώστε να δώσει δεδομένα σε μορφή πίνακα στους εκπαιδευόμενους για να τα μελετήσουν, να τα επεξεργαστούν ή ακόμη να ζητήσει από τους εκπαιδευόμενους να συμπληρώσουν έναν τέτοιο πίνακα.

- Το εργαλείο "**Επεξεργασία Εικόνας Pixlr**": Πρόκειται για ένα πρόγραμμα επεξεργασίας εικόνας το οποίο μοιάζει αρκετά με το πρόγραμμα ζωγραφικής των Windows διαθέτοντας ωστόσο πολλές παραπάνω δυνατότητες . Το εργαλείο αυτό μπορεί να αξιοποιηθεί από έναν εκπαιδευτικό στην περίπτωση που θέλει να δώσει μία εικόνα στους εκπαιδευόμενους του ώστε να επεξεργαστούν ή να την συμπληρώσουν.

### 2.3.1.2 Εργαλεία Αξιολόγησης

Τα εργαλεία αξιολόγησης χρησιμοποιούνται για να αξιολογήσουν τους εκπαιδευόμενους μέσα από διαγωνίσματα ή άλλες δοκιμασίες με βαθμολογία. Τα εργαλεία αυτά είναι: «Αξιολόγηση», «Ερωτήσεις πολλαπλών επιλογών», «Υποβολή αρχείου», «Ερωτηματολόγιο».

- Το εργαλείο "**Αξιολόγηση**": Μέσω του οποίου μπορεί να γίνει η αξιολόγηση των εκπαιδευόμενων καθώς υποστηρίζει σχεδόν όλων των ειδών των ερωτήσεων όπως πολλαπλή επιλογής, αντιστοίχισης, σύντομης απάντησης, σωστού/λάθους, έκθεσης και διάταξης με τα οποία μπορούν να δημιουργηθούν τεστ ή διαγωνίσματα παρέχοντας ανατροφοδότηση και υπολογίζοντας τελική βαθμολογία.
- Το εργαλείο "**Ερωτήσεις Πολλαπλής Επιλογής**": Μέσω του οποίου ένας καθηγητής μπορεί να δημιουργήσει τεστ με ερωτήσεις πολλαπλής επιλογής και σωστού/λάθους για τους εκπαιδευόμενους έχοντας τη δυνατότητα να παρέχει ανατροφοδότηση στις ερωτήσεις και εξαγωγή βαθμολογίας.
- Το εργαλείο "**Υποβολή αρχείου**": Μέσω του οποίου οι εκπαιδευόμενοι έχουν τη δυνατότητα να καταθέσουν ένα ή περισσότερα αρχεία τα οποία θα αξιολογήσει και ενδεχομένως να βαθμολογήσει ο καθηγητής.
- Το εργαλείο "**Ξυστό (Scratchie)**" ή "**Ερωτηματολόγιο**": Μέσω του οποίου ο καθηγητής μπορεί να δημιουργήσει ερωτήσεις πολλαπλής επιλογής και σύντομης ερώτησης για τους εκπαιδευόμενους του με την επιλογή εξαγωγής βαθμολογίας.

### 2.3.1.3 Εργαλεία Ανατροφοδότησης

Τα εργαλεία ανατροφοδότησης δίνουν τη δυνατότητα στους εκπαιδευτικούς να συλλέξουν διάφορες πληροφορίες από τους εκπαιδευόμενους με μορφή απαντήσεων, ψηφοφορίας κλπ. Τα εργαλεία ανατροφοδότησης είναι: "Ερωτήσεις και Απαντήσεις", "Ερευνα", "Ψηφοφορία", "Συλλογή Δεδομένων", "Σημειωματάριο" και "Νοητικός Χάρτης".

- Το εργαλείο "**Ερωτήσεις και Απαντήσεις**": Με το οποίο οι εκπαιδευτικοί έχουν τη δυνατότητα να θέτουν μια ή περισσότερες διαγνωστικές ερωτήσεις με σύντομο κείμενο στους εκπαιδευόμενους ξεχωριστά ώστε να τις απαντήσουν. Στη συνέχεια ο εκπαιδευτικός μπορεί να δει τις απαντήσεις των εκπαιδευόμενων αλλά υπάρχει η δυνατότητα και ο κάθε εκπαιδευόμενος να δει τις απαντήσεις των υπολοίπων.
- Το εργαλείο "**Ερευνα**": Με το οποίο ο εκπαιδευτικός μπορεί να θέσει μια σειρά ερωτήσεων στους εκπαιδευόμενους, όπως με το εργαλείο "Πολλαπλών επιλογών" με τη διαφορά ότι δεν υπάρχουν σωστές ή λάθος απαντήσεις. Το εργαλείο μπορεί να αξιοποιηθεί από τον εκπαιδευτικό με σκοπό να συλλέξει και να αναλύσει κάποια δεδομένα, με βάση τις απαντήσεις των εκπαιδευόμενων όπως για παράδειγμα στην περίπτωση δημιουργίας ενός ερωτηματολογίου για την αξιολόγηση του περιεχομένου ή της διδασκαλίας του μαθήματος.
- Το εργαλείο "**Ψηφοφορία**": Μέσω του οποίου ο εκπαιδευτικός μπορεί να δώσει έναν σύνολο επιλογών στους εκπαιδευόμενους για να ψηφίσουν, είτε ανώνυμα είτε επώνυμα και μέσα από τη διαδικασία αυτή να συλλέξει τις απόψεις τους πάνω σε κάποιο θέμα.

- Το εργαλείο "**Συλλογή Δεδομένων**": Μέσω του οποίου ο εκπαιδευτικός μπορεί να συλλέξει δεδομένα και στοιχεία από τους εκπαιδευόμενους τα οποία επιθυμεί να επεξεργαστεί, δημιουργώντας μία βάση δεδομένων.
- Το εργαλείο "**Σημειωματάριο**": Μέσω του οποίου οι εκπαιδευόμενοι μπορούν να καταγράψουν τις απόψεις και τις σκέψεις τους πάνω σε ένα θέμα ή ακόμη να κρατήσουν σημειώσεις για επανάληψη που χρειάζεται να κάνουν σε ένα μάθημα. Αυτά που καταγράφουν στο σημειωματάριο μπορούν να είναι ορατά από τον καθηγητή τους.
- Το εργαλείο "**Νοητικός Χάρτης**": Μέσω του οποίου ο καθηγητής μπορεί να προβάλει έναν νοητικό χάρτη στους εκπαιδευόμενους του αλλά και να ζητήσει από αυτούς είτε να τον επεξεργαστούν είτε να δημιουργήσουν έναν καινούριο ατομικά ή συνεργατικά. Οι νοητικοί χάρτες χρησιμοποιούνται συνήθως για να οργανώσουμε έννοιες και ιδέες ώστε στη συνέχεια να μελετήσουμε τις σχέσεις αλληλεπίδρασης που υπάρχουν ανάμεσα τους.

Εργαλεία Πληροφόρησης	Εργαλεία Αξιολόγησης	Εργαλεία Συνεργασίας	Εργαλεία Ανατροφοδότησης
Ανακ. Πίνακας	Αξιολόγηση	Επισκόπηση	Ερωτ. & Απαντ.
Διαμοίρ. Πόρων	Ερωτ. Πολ. Επ.	Ιστότ. Wiki	Ερευνα
Λίστα Εργασιών	Υποβολή Αρχείου	Συζήτηση (Forum)	Ψηφοφορία
Επεξ. Εικ. Pixlr	Ερωτηματολόγιο	Συνομιλία (Chat)	Συλ. Δεδομένων
Χώρος Έκθ. Εικόνων			Σημειωματάριο
Υπολογ. Φύλλο			Νοητικός χάρτης

Εικόνα 6: Τα εργαλεία Πληροφόρησης, Αξιολόγησης, Συνεργασίας, Ανατροφοδότησης στο LAMS[20]

### 2.3.1.4 Εργαλεία Συνεργασίας

Τα εργαλεία συνεργασίας παρέχουν τη δυνατότητα επικοινωνίας μεταξύ όλων των εκπαιδευομένων ή ομάδες αυτών. Η συζήτηση που προκύπτει από την επικοινωνία αυτή ελέγχεται φυσικά από τον εκπαιδευτικό. Τα εργαλεία συνεργασίας είναι: «Επισκόπηση», «Ιστότοπος Wiki», «Συζήτηση (Forum)» και «Συνομιλία (Chat)». Πιο συγκεκριμένα:

- Το εργαλείο "**Επισκόπηση**": Δίνει τη δυνατότητα σε μια ομάδα εκπαιδευομένων να συνεργάζεται και να καταγράφει τις απόψεις της δημιουργώντας μια γραπτή αναφορά. Ο πρώτος εκπαιδευόμενος που ξεκινάει τη δραστηριότητα της "Επισκόπησης" ορίζεται ως γραμματέας, δηλαδή αυτός που θα συνοψίσει τις απαντήσεις της ομάδας του. Οι εκπαιδευόμενοι μιας ομάδας έχουν επίσης τη δυνατότητα να ψηφίσουν επιμέρους απόψεις των μελών της ώστε να αποφασίσουν αν θέλουν να γίνει άποψη όλης της ομάδας. Η "Επισκόπηση" είναι αρκετά πιο αποτελεσματική όταν γίνεται σε συνδυασμό και με άλλη συνεργατική δραστηριότητα, όπως η συνομιλία ή η συζήτηση.
- Το εργαλείο "**Ιστότοπος Wiki**": Το εργαλείο αυτό κάνει δυνατή τη δημιουργία σελίδων Wiki συνεργατικά από τους συμμετέχοντες στην εκπαιδευτική διαδικασία. Μπορεί να



αξιοποιηθεί για οδηγίες, συχνές ερωτήσεις ή ακόμη για πληροφορίες με δομή για κάποιο θέμα. Φυσικά διατηρείται το ιστορικό της δημιουργίας των σελίδων

- Το εργαλείο "**Συζήτηση (Forum)**": Με το συγκεκριμένο εργαλείο γίνεται δυνατή η ασύγχρονη επικοινωνία μεταξύ των εκπαιδευόμενων και μεταξύ των εκπαιδευόμενων και των εκπαιδευτικών. Η συζήτηση μπορεί να είναι διαθέσιμη για όλο το χρόνο.
- Το εργαλείο «**Συνομιλία (Chat)**»: Το εργαλείο αυτό επιτρέπει μια ζωντανή (σύγχρονη) και άμεση συζήτηση μεταξύ των εκπαιδευόμενων. Ο εκπαιδευτικός μπορεί να αξιοποιήσει το εργαλείο της συζήτησης ώστε να βάλει τους εκπαιδευόμενους του να συνομιλήσουν για ένα μικρό χρονικό διάστημα πάνω σε ένα θέμα. Φυσικά ο εκπαιδευτικός έχει πρόσβαση πάνω σε αυτή τη συνομιλία από την οποία μπορεί να αντλήσει διάφορα στοιχεία.
- Το εργαλείο "**GMap**": Με το εργαλείο αυτό μπορεί να γίνει παρουσίαση ή επεξεργασία χαρτών με το google maps από τους εκπαιδευόμενους.

Σε αυτό το είδος των δραστηριοτήτων περιλαμβάνονται και οι **Ομαδικές Δραστηριότητες** (Εικόνα 7) οι οποίες αποτελούν συνδυασμό δύο δραστηριοτήτων που ήδη έχουν αναφερθεί παραπάνω. Αυτές είναι: «Πόροι και Συζήτηση», «Συζήτηση και Επισκόπηση» και «Συνομιλία και Επισκόπηση».

- "**Πόροι και Συζήτηση**": Είναι συνδυασμός των δραστηριοτήτων "Διαμοίραση Πόρων" και «Συζήτηση". Ο εκπαιδευτικός μπορεί να αξιοποιήσει αυτό το εργαλείο όταν θέλει οι μαθητές του να δουν κάποια αρχεία ή σελίδες στο διαδίκτυο σχετικά με ένα θέμα και στη συνέχεια να συζητήσουν μεταξύ τους αλλά και με τον ίδιο σχετικά με αυτά.
- "**Συζήτηση και Επισκόπηση**": Είναι συνδυασμός των δραστηριοτήτων "Συζήτηση" και "Επισκόπηση". Ο εκπαιδευτικός μπορεί να αξιοποιήσει το εργαλείο αυτό όταν επιθυμεί οι μαθητές του να συζητήσουν ασύγχρονα για κάποιο θέμα και στη συνέχεια κάποιος που έχει οριστεί να καταγράψει τα συμπεράσματα της συζήτησης τους.
- "**Συνομιλία και Επισκόπηση**": Είναι συνδυασμός των δραστηριοτήτων "Συνομιλία" και "Επισκόπηση". Ο εκπαιδευτικός μπορεί να αξιοποιήσει το εργαλείο αυτό όταν επιθυμεί οι μαθητές του να συζητήσουν σύγχρονα και στη συνέχεια κάποιος που έχει οριστεί να καταγράψει τα συμπεράσματα της συζήτησης.



Εικόνα 7: Τα εργαλεία Ομαδικών Δραστηριοτήτων και το εργαλείο Επιλογή Συντονιστή στο LAMS[20]

### 2.3.1.5 Εργαλείο "Επιλογή Συντονιστή"

Επιπλέον υπάρχει και το εργαλείο "**Επιλογή Συντονιστή**" (Εικόνα 7) το οποίο δεν ανήκει σε κάποια από τις παραπάνω κατηγορίες ωστόσο όταν προηγείται των δραστηριοτήτων

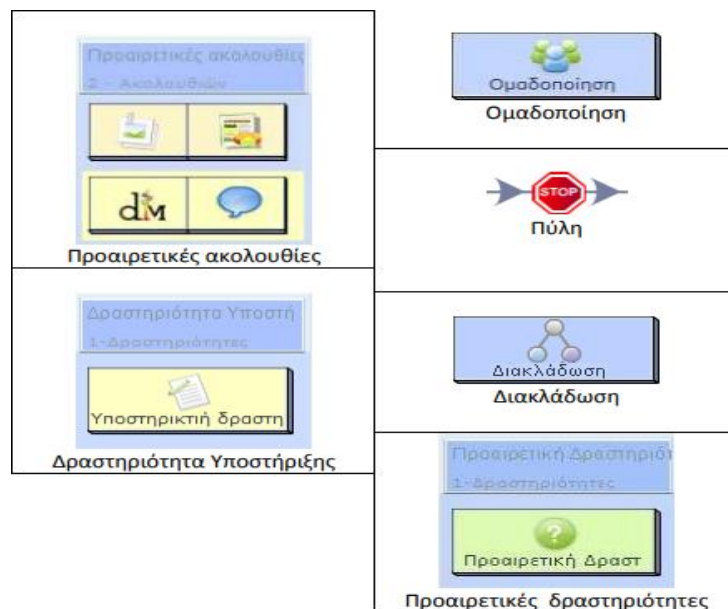
"Αξιολόγηση", "Υποβολή αρχείου", "Ερωτήσεις και απαντήσεις", "Ψηφοφορία" και "Ξυστό", δίνει τη δυνατότητα στους εκπαιδευόμενους όταν πραγματοποιούν τις συγκεκριμένες δραστηριότητες να επιλέξουν ποιος θα είναι ο συντονιστής (αρχηγός ή εκπρόσωπος) της ομάδας τους.

### 2.3.2 Εργαλεία Ελέγχου Ροής

Τα εργαλεία ελέγχου ροής (Εικόνα 8) μας επιτρέπουν να δημιουργούμε σημεία αναμονής – προσωρινής διακοπής της μαθησιακής πορείας, διακλαδώσεις στην ακολουθία, προαιρετικές μαθησιακές δραστηριότητες, προαιρετικές ακολουθίες μαθησιακών δραστηριοτήτων και υποστηρικτικές δραστηριότητες.

Παρακάτω περιγράφονται συνοπτικά όλα τα εργαλεία ελέγχου ροής που διαθέτει το LAMS στο περιβάλλον του συγγραφέα[19].

- **"Πύλη"**: Με το εργαλείο αυτό μπορούν να δημιουργηθούν σημεία αναμονής ή προσωρινής διακοπής (Stop Points) μέσα σε μια ακολουθία δραστηριοτήτων ώστε να μην έχουν πρόσβαση οι εκπαιδευόμενοι σε επόμενες δραστηριότητες μέχρις ότου μια συγκεκριμένη συνθήκη εκπληρωθεί, όπως για παράδειγμα να δώσει άδεια πρόσβασης ο εκπαιδευτικός. Αυτό το εργαλείο μπορεί να το αξιοποιήσει ο εκπαιδευτικός όταν ένα μάθημα επιλέγει να το κάνει σε παραπάνω από μία διδακτικές ώρες και χρησιμοποιώντας το εργαλείο αυτό να ανοίγει συγκεκριμένο μέρος δραστηριοτήτων πριν από το μάθημα.
- **"Διακλάδωση"**: Με το εργαλείο αυτό ο εκπαιδευτικός μπορεί να δημιουργήσει διαφορετικά μονοπάτια δραστηριοτήτων για να εκπονήσουν οι μαθητές με βάση αποτελέσματα προηγούμενων δραστηριοτήτων ή ομαδοποίησης που έχει επιλέξει ο ίδιος.

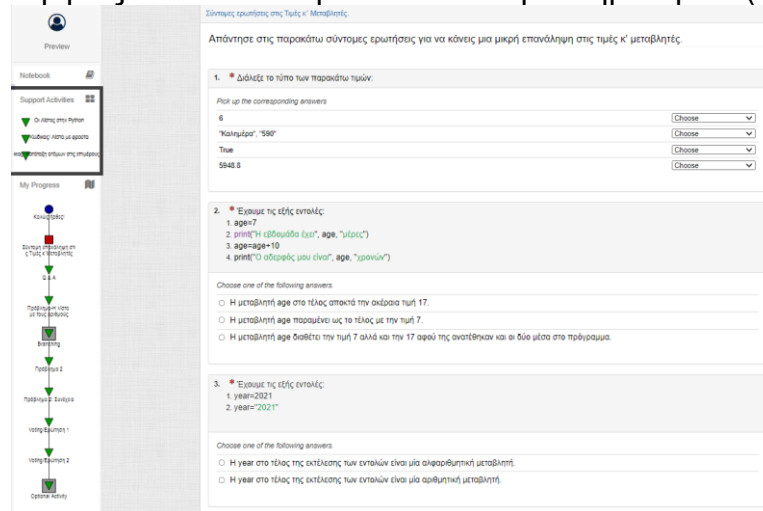


Εικόνα 8: Τα εργαλεία ελέγχου ροής στο LAMS[19]

- **"Ομαδοποίηση εκπαιδευομένων"**: Με το εργαλείο αυτό ο εκπαιδευτικός μπορεί να δημιουργήσει ομάδες εκπαιδευομένων και να ορίσει ποιες δραστηριότητες θα εκπονήσει η κάθε ομάδα στην συνέχεια.
- **"Προαιρετικές δραστηριότητες"**: Με το εργαλείο αυτό ο εκπαιδευτικός μπορεί να ορίσει ένα πλήθος δραστηριοτήτων από τις οποίες οι εκπαιδευόμενοι θα επιλέξουν

ποιες θα εκπονήσουν και ποιες όχι. Ο εκπαιδευτικός μπορεί να ορίσει το μέγιστο και τον ελάχιστο αριθμό δραστηριοτήτων που θα εκπονήσουν οι μαθητές του.

- **"Προαιρετικές ακολουθίες"**: Το εργαλείο αυτό έχει όμοια λειτουργία με τις προαιρετικές δραστηριότητες με τη διαφορά ότι εδώ ο εκπαιδευτικός μπορεί να ορίσει ολόκληρες ακολουθίες δραστηριοτήτων ως προαιρετικές.
- **"Υποστηρικτική Δραστηριότητα"**: Με το εργαλείο αυτό ο εκπαιδευτικός μπορεί να ορίσει ένα πλήθος δραστηριοτήτων με υποστηρικτικό υλικό για το μάθημα. Οι εκπαιδευόμενοι δεν είναι υποχρεωτικό να εκπονήσουν αυτού του είδους τις δραστηριότητες, μπορούν να έχουν πρόσβαση σε αυτές οποιαδήποτε στιγμή καθώς είναι ανεξάρτητες από την κύρια ακολουθία δραστηριοτήτων του μαθήματος. Γι αυτό το λόγο η δραστηριότητα αυτή εμφανίζεται στο πάνω αριστερά τμήμα στο περιβάλλον του εκπαιδευομένου, ακριβώς πάνω από την ακολουθία δραστηριοτήτων (Εικόνα 9).



**Εικόνα 9: Περιβάλλον εκπαιδευόμενου. Οι υποστηρικτικές δραστηριότητες, σε μαύρο πλαίσιο, είναι ανεξάρτητες από την ακολουθία δραστηριοτήτων**

### **3 ΟΙ ΟΜΙΛΟΙ ΣΤΑ ΠΡΟΤΥΠΑ ΚΑΙ ΠΕΙΡΑΜΑΤΙΚΑ ΣΧΟΛΕΙΑ - Ο ΡΟΛΟΣ ΚΑΙ ΤΟ ΠΛΑΙΣΙΟ ΛΕΙΤΟΥΡΓΙΑΣ**

Τα Πρότυπα και Πειραματικά Σχολεία συγκροτούν και λειτουργούν Ομίλους διαφόρων επιστημονικών πεδίων με σκοπό οι μαθητές να ασχοληθούν με τομείς που πιστεύουν ότι τους ενδιαφέρουν αλλά δεν τους δίνεται η ευκαιρία να το κάνουν στο πλαίσιο του αναλυτικού προγράμματος του σχολείου. Οι μαθητές συμμετέχοντας στους Ομίλους εκτός από την ευκαιρία που έχουν να εντρυφήσουν σε τομείς που τους ενδιαφέρει, ταυτόχρονα εμπλουτίζουν τις γνώσεις τους και αναπτύσσουν ικανότητες και κλίσεις στους τομείς αυτούς. Για κάθε Όμιλο που διεξάγεται υπάρχει συγκεκριμένο αναλυτικό πρόγραμμα και στόχοι που έχουν οριστεί από τους υπεύθυνους εκπαιδευτικούς.

Η λειτουργία των Ομίλων στηρίζεται συγκεκριμένους κανονισμούς που τίθενται. Σε κάθε Όμιλο έχουν δικαίωμα να οριστούν ως υπεύθυνοι το πολύ δύο εκπαιδευτικοί. Το εβδομαδιαίο ωράριο λειτουργίας των Ομίλων είναι 2 ή 4 διδακτικές ώρες, 1 ή 2 φορές την εβδομάδα αντίστοιχα και πραγματοποιούνται μετά το κανονικό ημερήσιο πρόγραμμα του εκάστοτε σχολείου, είναι δηλαδή απαραίτητο να έχουν ξεκινήσει μέχρι τις 14.30. Η επιλογή των μαθητών ολοκληρώνεται μέχρι το τέλος Σεπτεμβρίου κάθε σχολικού έτους και ταυτόχρονα γίνεται η έναρξη της λειτουργίας του κάθε ομίλου η οποία ολοκληρώνεται με το πέρας του διδακτικού έτους. Για τη λειτουργία του Ομίλου πρέπει οι μαθητές που συμμετέχουν να είναι το ελάχιστο 10 και το μέγιστο 20. Σε έναν Όμιλο, ανάλογα με το περιεχόμενο και τους στόχους που επιδιώκει, μπορεί να συμμετέχουν μαθητές από μία ή παραπάνω τάξεων. Επίσης, είναι δυνατόν ένας μαθητής να συμμετέχει το πολύ σε δύο διαφορετικούς Ομίλους. Σε έναν Όμιλο που διεξάγεται σε ένα Πρότυπο Σχολείο ή σε ένα Πειραματικό Σχολείο, μαζί με τους μαθητές των σχολείων αυτών μπορούν να συμμετέχουν και μαθητές από άλλα σχολεία της δημόσιας εκπαίδευσης. Οι μαθητές στο τέλος του σχολικού έτους λαμβάνουν πιστοποιητικό επιτυχούς συμμετοχής, εφόσον δεν έχουν υπερβεί τις προβλεπόμενες απουσίες και έχουν εκπονήσει τις εργασίες που τους έχει αναθέσει ο υπεύθυνος καθηγητής κατά τη διάρκεια διεξαγωγής του Ομίλου. Σημαντικό και αναγκαίο είναι οι δράσεις, τα αποτελέσματα και οι εκθέσεις των ομίλων να δημοσιοποιούνται είτε στο δικτυακό τόπο που ενδεχομένως να διαθέτει το σχολείο είτε με μία τελική παρουσίαση ώστε να παρουσιάζονται τα αποτελέσματα της λειτουργίας τους και να γίνονται γνωστά στη σχολική κοινότητα.

## 4 Ο ΟΜΙΛΟΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΟ ΣΧΟΛΙΚΟ ΕΤΟΣ 2020-21

### 4.1 Τα αντικείμενα μελέτης του Ομίλου

Ο Όμιλος Πληροφορικής του 2ου Πρότυπου Γενικού Λυκείου Αθηνών, λειτουργεί από 2015 και ασχολείται κυρίως με θέματα προγραμματισμού και ρομποτικής. Για τη χρονιά 2020-21 ο Όμιλος Πληροφορικής λειτούργησε σε συνεργασία με το τμήμα Πληροφορικής και Τηλ/νιών του ΕΚΠΑ και με το τμήμα Ηλεκτρολόγων Μηχανικών του ΕΜΠ σχετικά με τη δημιουργία εκπαιδευτικού υλικού και τη διδασκαλία των μαθημάτων που ανέλαβαν φοιτητές των αντίστοιχων τμημάτων. Για τη συγκεκριμένη χρονιά ασχολήθηκε με δύο αντικείμενα. Το πρώτο αντικείμενο ήταν η εκμάθηση βασικών εντολών και εργαλείων της γλώσσας προγραμματισμού Python: διαχείριση των μεταβλητών και αποθήκευση δεδομένων, δομές επιλογής και επανάληψης, λειτουργία πολλών συναρτήσεων που προσφέρει η γλώσσα, λίστες και συμβολοσειρές. Μέσα στη χρονιά κλήθηκαν να εκπονήσουν πολλές δραστηριότητες, να λύσουν πληθώρα προβλημάτων, να συντάξουν και να εκτελέσουν προγράμματα και στο τέλος να ετοιμάσουν φυσικά τις δικές τους εφαρμογές. Το δεύτερο αντικείμενο του Ομίλου ήταν η μελέτη και η εκμάθηση διαφόρων αλγορίθμων στον προγραμματισμό. Ξεκίνησαν με βασικούς αλγόριθμους, όπως της αναζήτησης και της ταξινόμησης και συνέχισαν αλγόριθμους προσπέλασης γραφών. Μέσα στη χρονιά δούλεψαν πάνω σε μία εφαρμογή το οποίο βασίστηκε στο γνωστό παιχνίδι `rackman` χρησιμοποιώντας τη γλώσσα προγραμματισμού Python που το παρουσίασαν στο τέλος.

### 4.2 Σκοπός και στόχοι του Ομίλου Πληροφορικής

Οι μαθητές παρακολουθώντας τα μαθήματα του Ομίλου Πληροφορικής είχαν την δυνατότητα να μάθουν προγραμματισμό με Python. Η Python αποτελεί την δημοφιλέστερη γλώσσα προγραμματισμού γενικού σκοπού που χρησιμοποιείται στην εκπαίδευση καθώς θεωρείται εξαιρετικά εύκολη στην εκμάθηση. Είναι κατάλληλη τόσο για αρχάριους στον προγραμματισμό[21] όσο και για έμπειρους αφού χρησιμοποιείται σε εξαιρετικά διαδεδομένους κλάδους της Πληροφορικής σήμερα όπως η Επιστήμη των Δεδομένων(Data Science ), η Τεχνητή Νοημοσύνη(Artificial Intelligence) και η Μηχανική Μάθηση(Machine Learning). Διαθέτει σύντομες εντολές που εύκολα θυμάται κάποιος καθώς προσφέρει και μεγάλο εύρος βιβλιοθηκών για τη ανάπτυξη διαφόρων εφαρμογών. Η Python ως γλώσσα προγραμματισμού θεωρείται αρκετά διαδεδομένη ως επιλογή στην εκπαίδευση, από το Γυμνάσιο μέχρι και το Πανεπιστήμιο. Μάλιστα παιδιά ηλικίας από 13 ετών μπορούν να ξεκινήσουν τα πρώτα τους βήματα στον προγραμματισμό με Python διευρύνοντας τις γνώσεις τους σταδιακά καθώς η ενασχόλησή τους συνεχίζεται μεγαλώνοντας. Φυσικά η εκμάθηση της γλώσσας αυτής ανά ηλικία πρέπει να συνοδεύεται από το κατάλληλο προγραμματιστικό περιβάλλον και τα κατάλληλα εκπαιδευτικά εργαλεία.

Γιατί όμως κάποιος να ασχοληθεί με τον προγραμματισμό; Είναι αλήθεια ότι τα οφέλη για τον εκπαιδευόμενο είναι πολλά ακόμη και αν δεν επιθυμεί να ασχοληθεί επαγγελματικά με τον προγραμματισμό στο μέλλον. Μαθαίνοντας κάποιος προγραμματισμό αναπτύσσει όχι μόνο την αλγοριθμική του σκέψη αλλά και τις ικανότητες του 21ου αιώνα. Αυτός είναι και ο κύριος σκοπός του Ομίλου Πληροφορικής, οι μαθητές να μάθουν να προγραμματίζουν σε Python αναπτύσσοντας ταυτόχρονα και ικανότητες που απαιτούνται σήμερα.

Συγκεκριμένα οι γενικοί στόχοι που τίθενται για τον Όμιλο Πληροφορικής είναι οι μαθητές:

- Να μπορούν αποφασίζουν ποιος είναι ο αποδοτικότερος αλγόριθμος για την επίλυση ενός συγκεκριμένου προβλήματος αναπτύσσοντας την κριτική και την υπολογιστική τους σκέψη.
- Να φτιάξουν τις δικές τους εφαρμογές πάνω στην Python καλλιεργώντας την δημιουργικότητά τους.
- Να είναι μέρος μιας ομάδας όπου μέσα από την συνεργασία και την συζήτηση να φτάσουν στην επίλυση ενός προβλήματος ή στην ολοκλήρωση ενός έργου εξασκώντας τις δεξιότητες επικοινωνίας και συνεργασίας.

Και οι ειδικοί στόχοι που αφορούν την εκμάθηση των βασικών στοιχείων της Python ως γλώσσα προγραμματισμού, για τα μαθήματα του Ομίλου Πληροφορικής, είναι οι μαθητές:

- Να εφαρμόζουν τις τρεις βασικές δομές του προγραμματισμού στα προγράμματά τους ανάλογα με τη φύση του προβλήματος, πιο συγκεκριμένα την δομή ακολουθίας, τη δομή επιλογής και τη δομή επανάληψης
- Να αποφασίζουν ποια δομή επιλογής (απλή, σύνθετη, πολλαπλή, εμφωλευμένη) και ποια δομή επανάληψης (while, for) είναι κατάλληλη για την επίλυση του εκάστοτε προβλήματος
- Να δημιουργούν τα δικά τους προγράμματα συντάσσοντας και εκτελώντας κώδικα κυρίως στο περιβάλλον IDLE που προσφέρει η Python
- Να αναγνωρίζουν και να ελέγχουν τυχόν λάθη στα προγράμματά τους και να τα διορθώνουν
- Να χρησιμοποιούν κατάλληλα τις εντολές εισόδου και εξόδου
- Να διαχειρίζονται κατάλληλα τις αριθμητικές, αλφαριθμητικές και λογικές μεταβλητές αποθηκεύοντας σωστά τα δεδομένα τους
- Να αξιοποιούν τους αριθμητικούς τελεστές +, -, /, //, \*, \*\*, % δημιουργώντας αριθμητικές εκφράσεις, αλλά και τους λογικούς τελεστές or και and συνθέτοντας λογικές εκφράσεις
- Να χρησιμοποιούν και να διαχειρίζονται σωστά τις λίστες και τις συμβολοσειρές, δύο από τις βασικές δομές δεδομένων που προσφέρει η Python

### 4.3 Οι συμμετέχοντες στα μαθήματα του Ομίλου Πληροφορικής

Υπεύθυνη για την επίβλεψη και διεξαγωγή των μαθημάτων του Ομίλου Πληροφορικής ήταν η κ. Αλεξάνδρα Γασπαρινάτου σε συνεργασία με τον κ. Ηλία Λίτινα. Και οι δύο είναι καθηγητές, Πληροφορικής και Χημείας αντίστοιχα του 2ου Πρότυπου Γενικού Λυκείου. Ήταν παρόντες στα μαθήματα, επέβλεπαν την εκπαιδευτική διαδικασία και συμμετείχαν κυρίως με διευκρινήσεις και ερωτήσεις. Παρέμβαιναν όπου θεωρούσαν ότι είναι απαραίτητο. Η διδασκαλία των μαθημάτων για την εκμάθηση των βασικών της Python γινόταν από εμένα, φοιτήτρια του τμήματος Πληροφορικής και Τηλ/νιών του ΕΚΠΑ και η διδασκαλία των μαθημάτων αλγορίθμων γινόταν αντίστοιχα από φοιτητή του τμήματος Ηλεκτρολόγων Μηχανικών του ΕΜΠ. Οι μαθητές που συμμετείχαν στα μαθήματα του Ομίλου ήταν 10 και προέρχονταν και από τις 3 τάξεις του Λυκείου.

### 4.4 Το πλαίσιο λειτουργίας μαθημάτων του Ομίλου Πληροφορικής

Τα μαθήματα του Ομίλου Πληροφορικής που είχαν ως αντικείμενο τον προγραμματισμό με Python γίνονταν μία φορά την εβδομάδα, συγκεκριμένα την ημέρα Πέμπτη και διαρκούσαν για δύο διδακτικές ώρες, δηλαδή 90 λεπτά. Ξεκινούσαν δηλαδή στις 14.15 και τελείωναν

στις 15.45. Τα μαθήματα που διεξήχθησαν ήταν συνολικά 25. Αρχικά τα μαθήματα πραγματοποιούνταν δια ζώσης στην αίθουσα εργαστηρίου πληροφορικής του σχολείου. Στις 29 Οκτωβρίου 2020 πραγματοποιήθηκε το πρώτο μάθημα που ήταν η παρουσίαση σχετικά με τα αντικείμενα που θα μελετούνταν στο πλαίσιο του Ομίλου ώστε να ενημερωθούν οι μαθητές. Στην συνέχεια, από τις 12 Νοεμβρίου 2020, λόγω του ολικού lockdown που καθιερώθηκε στη χώρα ως μέτρο για την εξάπλωση του ιού covid-19, τα μαθήματα συνεχίστηκαν διαδικτυακά μέχρι το τέλος της σχολικής χρονιάς. Αν και στις 10 Μαΐου τα Γυμνάσια και τα Λύκεια σχολεία άνοιξαν, τα μαθήματα του Ομίλου Πληροφορικής εξακολούθησαν να πραγματοποιούνται διαδικτυακά με τη μόνη αλλαγή το μάθημα να ξεκινά στις 15.00. Το τελευταίο μάθημα πραγματοποιήθηκε 2 Ιουνίου και ήταν η παρουσίαση των εφαρμογών που είχαν αναλάβει να δημιουργήσουν τα παιδιά. Κατά τη διάρκεια όλης της σχολικής χρονιάς υπήρχε διαδικτυακή επικοινωνία μεταξύ μαθητών και καθηγητών μέσω της πλατφόρμας του Edmodo.

## **5 ΣΧΕΔΙΑΣΗ ΜΑΘΗΜΑΤΩΝ ΔΙΔΑΣΚΑΛΙΑΣ ΡΥΘΜΟΝ ΓΙΑ ΤΟΝ ΟΜΙΛΟ ΠΛΗΡΟΦΟΡΙΚΗΣ**

### **5.1 Συνθήκες και χαρακτηριστικά στα οποία που βασίστηκε η σχεδίαση μαθημάτων**

Στο πλαίσιο της συγκεκριμένης εργασίας δημιουργείται εκπαιδευτικό υλικό για τα μαθήματα του Ομίλου Πληροφορικής που αφορούν την εκμάθηση των βασικών χαρακτηριστικών της γλώσσας προγραμματισμού Python αξιοποιώντας το περιβάλλον LAMS. Ένα μάθημα στο LAMS, δηλαδή μια σειρά από δραστηριότητες, διαμορφώνεται από τον καθηγητή ανάλογα με το αντικείμενο διδασκαλίας, το κοινό στο οποίο απευθύνεται και τη μέθοδο διδασκαλίας που επιλέγει να ακολουθήσει. Στη συγκεκριμένη περίπτωση, το αντικείμενο διδασκαλίας είναι η εκμάθηση προγραμματισμού σε γλώσσα Python, απευθύνεται σε μαθητές που προέρχονται και από τις τρεις τάξεις του Λυκείου και προορίζεται για μαθήματα του Ομίλου Πληροφορικής. Στόχος λοιπόν είναι να σχεδιαστούν μαθήματα αποδοτικά, δημιουργικά και ευχάριστα για τους μαθητές λαμβάνοντας υπόψη τις παραπάνω απαιτήσεις.

Μία συνθήκη που υπάρχει σχετικά με τη δημιουργία του συγκεκριμένου εκπαιδευτικού υλικού, είναι οι μαθητές που συμμετέχουν στα μαθήματα του Ομίλου να εκπονούν δραστηριότητες κυρίως την ώρα του μαθήματος. Πρέπει η προσδοκώμενη διδακτική ύλη να καλύπτεται την ώρα του μαθήματος, χωρίς να επιβαρύνονται οι μαθητές με επιπλέον υποχρεωτική μελέτη και εργασίες για το "σπίτι". Σχεδιάζονται λοιπόν μαθήματα με στόχο οι μαθητές να μαθαίνουν να προγραμματίζουν σε Python σταδιακά, εκπονώντας δραστηριότητες κατά τη διάρκεια του Ομίλου. Το κάθε μάθημα στο LAMS σχεδιάζεται έτσι ώστε να περιλαμβάνει 3-4 υποενότητες και να μπορεί να καλυφθεί σε 3-4 συνολικά διδακτικές ώρες του Ομίλου αντίστοιχα. Οι μαθητές σε κάθε μάθημα διάρκειας 90 λεπτών του Ομίλου, έχουν τη δυνατότητα εκπονώντας μια σειρά δραστηριοτήτων στο LAMS, αρχικά να μελετήσουν και να πειραματιστούν με τμήματα κώδικα, μέσα από τον πειραματισμό και τη διερεύνηση σε συνδυασμό με την υποστηρικτική διδασκαλία του καθηγητή, να μάθουν νέες εντολές και μεθόδους προγραμματισμού στην Python, να τις συνδυάσουν με προηγούμενες εντολές που έμαθαν και στη συνέχεια να μπουν στη διαδικασία να λύσουν προβλήματα και να δημιουργήσουν τα δικά τους έργα(projects). Φυσικά η διαρκής εξάσκηση με τον προγραμματισμό είναι απαραίτητη και αναγκαία για έναν εκπαιδευόμενο. Γι αυτό σε κάθε μάθημα στο LAMS παρέχονται από τον καθηγητή επιπλέον υποστηρικτικές και επιλογής δραστηριότητες για επιπλέον ατομική ενασχόληση του εκπαιδευόμενου με το αντικείμενο του προγραμματισμού.

Μία δεύτερη συνθήκη που αφορά τα μαθήματα του Ομίλου, το οποίο πρέπει να ισχύει γενικά στην εκπαίδευση, είναι η δημιουργία μαθημάτων βασισμένα στη μαθητοκεντρική σχεδίαση όπου ο μαθητής έχει ενεργό ρόλο και μπορεί να αλληλεπιδράσει με τους συμμαθητές του και τον καθηγητή του. Στόχος είναι μαθήματα σχεδιασμένα και δομημένα με επίκεντρο τον μαθητή που διευκολύνουν τη μάθησή του έναντι της παραδοσιακής διδασκαλίας και της απλής μεταφοράς γνώσεων. Γι αυτό το λόγο οι ακολουθίες δραστηριοτήτων που σχεδιάζονται στο LAMS, να μην είναι καθοδηγούμενες από τον καθηγητή ώστε ένας μαθητής να μπορεί να τις εκτελεί ανεξάρτητα και χωρίς την παρέμβαση του καθηγητή, ωστόσο το ύφος των μαθημάτων του Ομίλου είναι τέτοιο που καθιστά αναγκαία αλλά και εξαιρετικά ωφέλιμη την αλληλεπίδραση μεταξύ του μαθητή και



του καθηγητή αλλά και των μαθητών μεταξύ τους. Τα μαθήματα λοιπόν σχεδιάζονται με τέτοιο τρόπο ώστε να ακολουθούν ένα μοτίβο. Οι μαθητές στην αρχή να προσπαθούν ατομικά να εκπονήσουν τις δραστηριότητες, στην συνέχεια όλοι μαζί να σχολιάζουν τις απαντήσεις που δόθηκαν και έπειτα να προχωρούν στην επίλυση ασκήσεων και προβλημάτων όπου συλλογικά μέσα από τη ανταλλαγή απόψεων και τη συζήτηση που κατευθύνεται από τον καθηγητή να καταλήξουν στη λύση τους. Φυσικά ο καθηγητής είναι παρόν σε όλη τη διάρκεια του μαθήματος έχοντας υποστηρικτικό ρόλο, παρεμβαίνοντας όπου χρειάζεται και κατευθύνοντας την πορεία της μάθησης.

Άλλη μία συνθήκη που πρέπει να ληφθεί υπόψη για τη σχεδίαση μαθημάτων είναι ότι στα μαθήματα του Ομίλου συμμετέχουν μαθητές και από τις τρεις τάξεις του Λυκείου, δηλαδή μαθητές που βρίσκονται σε διαφορετικό επίπεδο σχετικά με τις γνώσεις και τις ικανότητές τους στον προγραμματισμό. Συγκεκριμένα, οι μαθητές της Γ' Λυκείου κατανοούν ευκολότερα και γρηγορότερα έννοιες και μεθόδους σχετικά με τον προγραμματισμό σε σχέση με τους μαθητές της Α' Λυκείου, καθώς ασχολούνται με το εισαγωγικό μάθημα στους αλγόριθμους και στον προγραμματισμό, το μάθημα της "Πληροφορικής". Έτσι λοιπόν σχεδιάζονται μαθήματα με τη μορφή κατευθυνόμενης ακολουθίας δραστηριοτήτων τα οποία περιλαμβάνουν δραστηριότητες διαβαθμισμένης δυσκολίας αλλά και διαφορετικά μονοπάτια μάθησης ανάλογα των δυνατοτήτων των μαθητών. Σε κάθε μάθημα, υπάρχουν απλές αλλά και πιο απαιτητικές δραστηριότητες που είναι κοινές για όλους τους μαθητές αλλά και μικρότερες ακολουθίες δραστηριοτήτων που ανατίθενται από τον καθηγητή ξεχωριστά σε μαθητές ανάλογα του γνωστικού τους επιπέδου τους.

## **5.2 Διδακτική Αρχή, Διδακτικές τεχνικές, Διδακτικές προσεγγίσεις που βασίστηκε η σχεδίαση μαθημάτων**

Οι αρχές μάθησης και διδασκαλίας στις οποίες βασίστηκε ο μαθησιακός σχεδιασμός είναι ο εποικοδομισμός[22]. Σύμφωνα με τον εποικοδομισμό η μάθηση είναι μία ενεργή διαδικασία στην οποία οι μαθητές κατασκευάζουν ενεργά την γνώση. Το περιεχόμενο της μάθησης δηλαδή, δεν ορίζεται εξ αρχής αλλά οι μαθητές την κατασκευάζουν σιγά-σιγά στηριζόμενοι στις πρότερες γνώσεις τους. Οι μαθητές συμμετέχουν σε μια διαδικασία επίλυσης προβλημάτων και μέσα από αυτήν τη διαδικασία πειραματισμού και ανακάλυψης, κατασκευάζουν τη γνώση και αναπτύσσουν μοναδικές για τον καθένα δεξιότητες. Η γνώση δεν είναι κάτι δεδομένο και απόλυτο αλλά συνθέτεται μέσα από τη μεταξύ τους επικοινωνία και την ανταλλαγή απόψεων. Γι αυτό το λόγο και ο καθηγητής έχει το ρόλο του βοηθού, του κατευθυντή, του υποστηρικτή στην διαδικασία μάθησης των μαθητών.

Στην αρχή κάθε μαθήματος οι μαθητές καλούνται να εκπονήσουν δραστηριότητες που σχεδιάστηκαν με βάση τη μέθοδο των "Διερευνήσεων", του "Μαύρου Κουτιού" και του "Παραδείγματος Αναφοράς". Οι δραστηριότητες που σχεδιάζονται βάση αυτών των τριών μεθόδων, αποτελούν μαθητοκεντρικές δραστηριότητες και έχουν ως στόχο την ενεργοποίηση και συμμετοχή των μαθητών στη μαθησιακή διαδικασία, την ανάδειξη και αξιοποίηση των πρότερων γνώσεων, την κατάκτηση της γνώσης μέσα από την επίλυση προβλημάτων και την αλληλεπίδρασή τους. Οι δραστηριότητες που σχεδιάζονται με βάση την προσέγγιση των "Διερευνήσεων" περιέχουν μικρά προγράμματα τα οποία οι μαθητές καλούνται να μελετήσουν και να απαντήσουν σε ερωτήσεις σχετικά με την λειτουργία τους, να προβλέψουν το αποτέλεσμα εκτέλεσης και στη συνέχεια εκτελώντας το πρόγραμμα, να παρατηρήσουν το αποτέλεσμα και να το συγκρίνουν με τις δικές τους απαντήσεις[22]. Οι δραστηριότητες που σχεδιάζονται με βάση την προσέγγιση του "Μαύρου Κουτιού" περιλαμβάνουν δύο βασικά τμήματα. Οι μαθητές αρχικά καλούνται να εκτελέσουν κάποιο

πρόγραμμα, χωρίς να έχουν πρόσβαση στον κώδικα, απαντώντας σε ερωτήσεις που σχετίζονται με το αποτέλεσμα της εκτέλεσης και στην συνέχεια να μελετήσουν τον κώδικα και να απαντήσουν σε ερωτήσεις που σχετίζονται με τις εντολές του προγράμματος, συνδυάζοντας με αυτό τον τρόπο τις προγραμματιστικές δομές που περιλαμβάνει ο κώδικας με την εκτέλεσή του[22]. Οι δραστηριότητες που σχεδιάζονται με βάση το “Παράδειγμα Αναφοράς” αναθέτουν στους μαθητές ένα πρόβλημα προς επίλυση και παράλληλα τους δίνετε ένα παράδειγμα που περιγράφουν την λύση ενός παρόμοιου προβλήματος. Οι μαθητές προσπαθούν να λύσουν το πρόβλημα μελετώντας προσεκτικά το παράδειγμα και προσαρμόζοντας τη λύση του παραδείγματος που τους έχει δοθεί[22].

Κατά την διάρκεια εκπόνησης των διερευνητικών δραστηριοτήτων ο καθηγητής έχει το ρόλο να καθοδηγεί, να συντονίζει και να βοηθάει τους μαθητές όποτε χρειαστεί παρέχοντάς τους κατάλληλες συμβουλές. Ωστόσο, υπάρχουν φορές που μερικές έννοιες απαιτούν παραπάνω ανάλυση η οποία δεν μπορεί να καλυφθεί πλήρως στο πλαίσιο των δραστηριοτήτων λόγω του περιορισμένου χρόνου που διαρκούν τα μαθήματα, καθιστώντας αναγκαία την “Εισήγηση” από τον καθηγητή. Με την τεχνική της εισήγησης, ο καθηγητής μπορεί να παρουσιάσει τις πληροφορίες που επιθυμεί σε συγκεκριμένο και σύντομο χρονικό διάστημα[22]. Η εισήγηση ως τεχνική, δεν αποτελεί μια άκρως αποτελεσματική τεχνική για τη μαθηση καθώς δεν υπάρχει συμμετοχή και αλληλεπίδραση των μαθητών. Για το λόγο αυτό, επιλέγεται η εισήγηση να συνοδεύεται πάντα από την τεχνική των “Ερωτοαποκρίσεων” ώστε να δημιουργηθεί μία μορφή διαλόγου μεταξύ του καθηγητή και των μαθητών και με αυτόν τον τρόπο να τους διεγείρει το ενδιαφέρον, να τους επιστήσει την προσοχή, να τους ενθαρρύνει την περιέργεια και να δημιουργήσει κλίμα συμμετοχής και επικοινωνίας μεταξύ τους[22]. Η ολιγόλεπτη εισήγηση σε συνδυασμό με τις ερωτοαποκρίσεις, εκτός από την ανάλυση νέων εννοιών, είναι αρκετά βοηθητική όταν πραγματοποιείται στο τέλος του μαθήματος με σκοπό την ανακεφαλαίωση των όσων διδάχθηκαν οι μαθητές αλλά και στην αρχή του μαθήματος με σκοπό την υπενθύμιση προηγούμενων εννοιών.

Μετά την εκπόνηση των διερευνητικών δραστηριοτήτων από τους μαθητές και της πιθανής εισήγησης από τον καθηγητή, ακολουθεί η τεχνική της “Πρακτικής Άσκησης” με την οποία οι μαθητές εφαρμόζουν τις γνώσεις τους[22]. Μέσω της πρακτικής άσκησης οι μαθητές δουλεύουν κυρίως ατομικά και προσπαθούν είτε να απαντήσουν σε ερωτήματα σχετικά με την ορθότητα δοσμένων προγραμμάτων είτε να επιλύσουν διάφορα προβλήματα που απαιτούν τη δημιουργία και εκτέλεση των δικών τους προγραμμάτων. Σε όλη την διάρκεια εκπόνησης τέτοιου είδους δραστηριοτήτων, ο καθηγητής καθοδηγεί τους μαθητές μέσω ερωτοαπαντήσεων δίνοντας τους συμβουλές και χωρίζοντας το αρχικό πρόβλημα σε μικρότερα ώστε να προχωρούν βήμα-βήμα προς την λύση.

### **5.3 Τα εργαλεία του LAMS που αξιοποιήθηκαν στη δημιουργία μαθημάτων**

Για το σχεδιασμό των μαθημάτων στο περιβάλλον του LAMS, αξιοποιήθηκαν μερικά από τα εργαλεία που διαθέτει συνολικά το περιβάλλον του συγγραφέα με σκοπό το σχεδιασμό κατάλληλων δραστηριοτήτων με βάση τις παραπάνω τεχνικές που αναφέρθηκαν. Έτσι τα εργαλεία του LAMS που κατά κύριο λόγο χρησιμοποιήθηκαν είναι:

- Το εργαλείο του “Πίνακα Ανακοινώσεων” με το οποίο παρατίθενται κυρίως οι στόχοι του μαθήματος στην αρχή και η θεωρία του μαθήματος ώστε να έχουν πρόσβαση οι μαθητές οποιαδήποτε στιγμή και μετά τη λήξη του μαθήματος.
- Το εργαλείο “Παράθεση Πόρων” με το οποίο ο καθηγητής παραθέτει αρχείο με τη θεωρία του μαθήματος (αν δεν έχει χρησιμοποιήσει το εργαλείο του “Πίνακα

Ανακοινώσεων” για το σκοπό αυτό) αλλά και χρήσιμους διαδικτυακούς πόρους που παρέχουν συμπληρωματικές πληροφορίες σχετικά με το μάθημα και οι μαθητές μπορούν να τους επισκεφτούν και να τους μελετήσουν όποτε και αν το επιθυμούν μετά τη λήξη του μαθήματος.

- Το εργαλείο “Ερώτηση/Απάντηση” με το οποίο παρατίθενται οι δραστηριότητες των “Διερευνήσεων”, του “Μαύρου Κουτιού” και του “Παραδείγματος Αναφοράς”. Όλες αυτές οι δραστηριότητες που έχουν δημιουργηθεί με τις παραπάνω τεχνικές συνοδεύονται από μια σειρά ερωτήσεων οι οποίες μπορούν να αποδοθούν με το συγκεκριμένο εργαλείο. Επίσης επειδή πρόκειται για διερευνητικές δραστηριότητες, το εργαλείο έχει ρυθμιστεί ώστε οι μαθητές μετά την υποβολή των απαντήσεών τους να μπορούν να δουν τις απαντήσεις των συμμαθητών τους ώστε να προβληματιστούν για τυχόν διαφορετικές απόψεις και να αναστοχαστούν για τη δική τους απάντηση. Το εργαλείο αυτό χρησιμοποιείται επίσης και για την παράθεση ερωτημάτων που απαιτούν από τους μαθητές να απαντήσουν με ανάπτυξη λόγου σχολιάζοντας την ορθότητα ενός προγράμματος ή συμπληρώνοντας τον κατάλληλα. Ακόμη και παράθεση προβλημάτων προς επίλυση που απαιτούν από τους μαθητές να δημιουργήσουν και να εκτελέσουν τα δικά τους προγράμματα και στη συνέχεια να παραθέσουν τη λύση τους ως απάντηση έχοντας ο καθηγητής άμεση πρόσβαση στην απάντηση-λύση του μαθητή. Το εργαλείο “Ερώτηση/Απάντηση”, για τις διερευνητικές δραστηριότητες, την παράθεση ερωτημάτων αλλά και για την παράθεση προβλημάτων προς επίλυση έχει ρυθμιστεί ώστε οι μαθητές να μπορούν να ξανά υποβάλουν την απάντησή τους, πράγμα πολύ βοηθητικό όταν θέλουν να διορθώσουν μια απάντησή τους έπειτα από κάποια επεξήγηση του καθηγητή ή δική τους αναθεώρηση.
- Το εργαλείο “Πολλαπλή Επιλογή” με το οποίο θέτονται ερωτήματα κλειστού τύπου στους μαθητές με σκοπό να διαλέξουν τη σωστή απάντηση ανάμεσα στις επιλογές που έχουν. Για να απαντήσουν στα ερωτήματα αυτά οι μαθητές πρέπει να εφαρμόσουν τις γνώσεις τους και έχουν σκοπό να εντοπιστούν τυχόν παρανοήσεις που μπορεί να έχουν προκύψει ώστε να αναλυθούν στην συνέχεια από τον καθηγητή. Τα ερωτήματα αυτά δεν έχουν το ρόλο αξιολόγησης, παρά μόνο ελέγχου κατανόησης νέων εννοιών από τους μαθητές. Για το λόγο αυτό δεν έχει ρυθμιστεί να εξάγεται βαθμολογία. Επιπλέον στις ερωτήσεις παρατίθεται γενική ανατροφοδότηση ώστε οι μαθητές να μπορούν να τη μελετήσουν μετά την υποβολή των απαντήσεών τους και να μπορούν να αναστοχαστούν.
- Το εργαλείο “Ψηφοφορία” με το οποίο τίθεται ένα ερώτημα στα παιδιά με σκοπό να διεγείρει το ενδιαφέρον τους και να αξιοποιήσει τις πρότερες γνώσεις τους πάνω σε ένα θέμα το οποίο θα διδαχθούν στην πορεία. Το εργαλείο αυτό εξάγει το ποσοστό απαντήσεων των μαθητών στο ερώτημα που του έχει τεθεί και γίνεται αφορμή για συζήτηση και ανταλλαγή απόψεων μεταξύ των μαθητών και του καθηγητή.
- Το εργαλείο της “Αξιολόγησης” με το οποίο παρατίθενται ερωτήσεις στους μαθητές πολλαπλής επιλογής, αντιστοίχισης, σωστού-λάθους και σύντομης απάντησης σχετικά με την κατανόηση εννοιών του μαθήματος. Η δραστηριότητα που δημιουργείται με αυτό το εργαλείο δεν έχει το ρόλο αυστηρής αξιολόγησης. Μπορεί να τοποθετεί στην αρχή του μαθήματος ώστε να σχηματίσει ο καθηγητής εικόνα σχετικά με κενά που πιθανόν να έχουν οι μαθητές πάνω σε προηγούμενες έννοιες που είχαν διδαχθεί είτε στο τέλος του μαθήματος ως ανατροφοδότηση για τον βαθμό που έγιναν κατανοητές οι νέες έννοιες του συγκεκριμένου μαθήματος. Σε κάθε περίπτωση στις ερωτήσεις που δημιουργούνται στο εργαλείο αυτό παρέχεται ανατροφοδότηση στους μαθητές για τον προσωπικό τους αναστοχασμό.
- Το εργαλείο “Δραστηριότητες Επιλογής” το οποίο χρησιμοποιείται ώστε να παρέχει στους μαθητές ένα πλήθος δραστηριοτήτων δίνοντάς τους την δυνατότητα να

επιλέξουν ποιες και πόσες από αυτές θα εκτελέσουν ή ακόμη και να μην εκτελέσουν καμία από αυτές.

- Το εργαλείο “Δραστηριότητα Υποστηρικτική” το οποίο χρησιμοποιείται κυρίως για να παρέχει στους μαθητές την θεωρία ή επιπλέον διαδικτυακές πηγές προς μελέτη ώστε να έχουν πρόσβαση σε αυτές οποιαδήποτε στιγμή ανεξάρτητα από την πορεία τους στην κατευθυνόμενη ακολουθία δραστηριοτήτων.
- Το εργαλείο “Διακλάδωση” το οποίο χρησιμοποιείται για τη δημιουργία διαφορετικών μονοπατιών μάθησης τα οποία ανταποκρίνονται στις διαφορετικές ικανότητες των μαθητών.

## 6 ΣΧΕΔΙΑ ΜΑΘΗΜΑΤΩΝ ΣΤΟ LAMS ΚΑΙ ΑΝΑΛΥΣΗ ΠΟΡΕΙΑΣ ΜΑΘΗΜΑΤΟΣ

Για τη διδασκαλία της Python στον Όμιλο Πληροφορικής είχε δημιουργηθεί εκπαιδευτικό υλικό την προηγούμενη χρονιά 2019-20 από φοιτητή του Τμήματος στο πλαίσιο διεξαγωγής πτυχιακής εργασίας. Το αντίστοιχο εκπαιδευτικό υλικό που δημιουργείται για τη διδασκαλία της Python τη χρονιά 2020-21 στηρίχθηκε σε αρκετά σημεία στο υπάρχον υλικό αλλά ενισχύθηκε και τροποποιήθηκε σε μεγάλο βαθμό.

Για τη διδασκαλία των βασικών χαρακτηριστικών της Python για τον Όμιλο Πληροφορικής σχεδιάζονται 6 μαθήματα στο σύστημα του LessonLAMS με το κάθε ένα να αντιστοιχεί και σε αντίστοιχη ενότητα. Τα μαθήματα είναι:

- Εισαγωγή στον προγραμματισμό με Python
- Η Δομή Ακολουθίας στην Python
- Η Δομή Επιλογής στην Python
- Οι Συμβολοσειρές στην Python
- Οι Λίστες στην Python
- Η Δομή Επανάληψης στην Python

### 6.1 Εισαγωγή στον προγραμματισμό με Python

Το πρώτο μάθημα είναι η "Εισαγωγή στον προγραμματισμό με Python" (Εικόνα 9), ένα εισαγωγικό μάθημα που βοηθάει τους μαθητές να μάθουν βασικές έννοιες του προγραμματισμού, βασικές ιδιότητες της γλώσσας προγραμματισμού Python και να αποκτήσουν μία πρώτη ιδέα σχετικά με το τι θα ασχοληθούν στη συνέχεια των μαθημάτων του Ομίλου Πληροφορικής. Καθώς στα πρώτα μαθήματα μπορεί να προσέλθουν μαθητές οι οποίοι δεν έχουν ιδιαίτερο ενδιαφέρον για τον προγραμματισμό αλλά περιέργεια για το περιεχόμενο, σκοπός του μαθήματος είναι να παρουσιάσει στους ενδιαφερόμενους το αντικείμενο μελέτης του Ομίλου.

#### Μαθησιακοί Στόχοι:

Οι μαθησιακοί στόχοι που τίθενται σε αυτό το μάθημα είναι οι μαθητές μετά το πέρας του μαθήματος να μπορούν:

- να εξηγούν τον ρόλο του υπολογιστή στην επίλυση προβλημάτων
- να διατυπώνουν τι είναι ένα πρόβλημα
- να ορίζουν τι είναι αλγόριθμος
- να επεξηγούν πώς ένα πρόγραμμα σχετίζεται με τις έννοιες του αλγορίθμου και την επίλυση προβλήματος με τον υπολογιστή
- να περιγράφουν τι είναι γλώσσα προγραμματισμού και να εξηγούν τον ρόλο της μεταξύ των προγραμματιστών και του υπολογιστή
- να εξηγούν γιατί η Python θεωρείται γλώσσα γενικού σκοπού
- να αναφέρουν 3 εφαρμογές που έχουν υλοποιηθεί με Python
- να εξηγούν γιατί η Python είναι μια διερμηνευόμενη γλώσσα προγραμματισμού και ποια είναι η διαφορά με μια μη διερμηνευόμενη γλώσσα προγραμματισμού
- να γράψουν και να εκτελέσουν το πρώτο τους πρόγραμμα Python στο IDLE.

## Δραστηριότητες και Εργαλεία του LAMS:

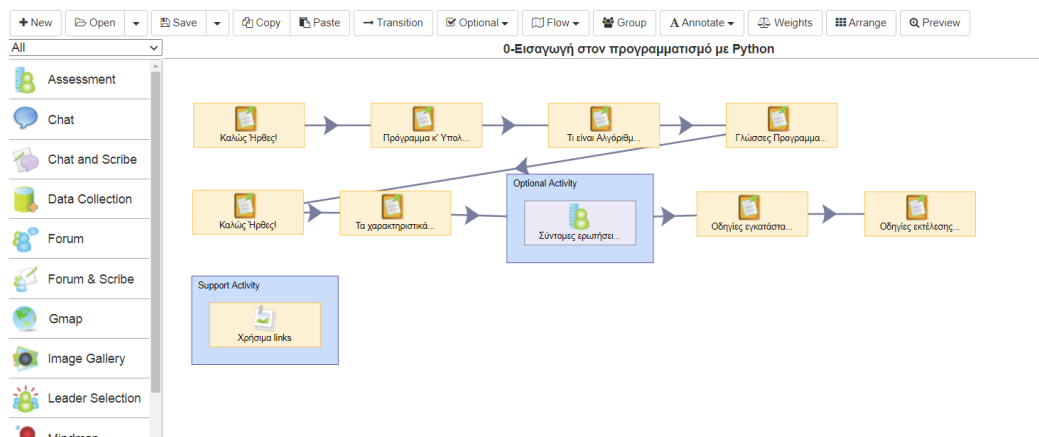
Το μάθημα είναι σχεδιασμένο ενδεικτικά για 1 διδακτική ώρα του Ομίλου 90 λεπτών και αποτελείται από μία σειριακή ακολουθία δραστηριοτήτων ίδια για κάθε εκπαιδευόμενο (Εικόνα 10) και πιο συγκεκριμένα αποτελείται από:

- 6 εργαλεία δραστηριοτήτων "Πίνακας Ανακοινώσεων" όπου παρατίθενται οι στόχοι του μαθήματος, θεωρία σχετικά με εισαγωγικές έννοιες του προγραμματισμού όπως τί είναι αλγόριθμος, πρόγραμμα, γλώσσα προγραμματισμού και θεωρία σχετικά με τα χαρακτηριστικά της Python όπως γιατί είναι μια διερμηνευόμενη, γενικού σκοπού γλώσσα προγραμματισμού και γιατί είναι αρκετά διαδεδομένη στις μέρες μας . Οι πληροφορίες έχουν κυρίως τη μορφή κειμένου.
- 1 δραστηριότητα επιλογής που περιλαμβάνει 1 εργαλείο δραστηριότητας "Αξιολόγηση" με ερωτήσεις κλειστού τύπου σχετικά με τις έννοιες που διδάχτηκαν προηγουμένως.
- 2 εργαλεία δραστηριοτήτων "Πίνακας Ανακοινώσεων" όπου παρατίθενται οδηγίες σχετικά με το πώς θα εγκαταστήσουν την Python στον υπολογιστή τους και πώς θα εκτελέσουν το πρώτο τους πρόγραμμα στο IDLE. Οι πληροφορίες έχουν τη μορφή κειμένου και υποστηρίζονται από εικόνες και βίντεο.
- 1 υποστηρικτική δραστηριότητα που περιλαμβάνει 1 εργαλείο δραστηριότητας "Παράθεση Πόρων" όπου παρατίθενται ιστοσελίδες με πηγές, βίντεο και ντοκιμαντέρ τα οποία περιλαμβάνουν περαιτέρω πληροφορίες σχετικά με τις έννοιες που διδάχτηκαν .

## Πορεία μαθήματος:

- Αρχικά στην ώρα του μαθήματος, ο καθηγητής κάνει μία εισήγηση σχετικά με έννοιες προγραμματισμού όπως τί είναι αλγόριθμος, πρόγραμμα, γλώσσα προγραμματισμού, ποιά είναι η συμβολή του υπολογιστή στην επίλυση προβλημάτων. Η εισήγηση αυτή συνοδεύεται από τις κατάλληλες ερωτοαπαντήσεις προς τους μαθητές ξεκινώντας ένα διάλογο ώστε να αξιοποιήσει την πρότερη γνώση που ενδεχομένως να έχουν. Άλλωστε έννοιες όπως αυτές που αναφέρθηκαν είναι πολύ πιθανό να τις ξέρουν αλλά είναι αναγκαίο να αναφερθούν ώστε να γίνει κατάλληλα η σύνδεση με τον προγραμματισμό σε Python. Η εισήγηση σε συνδυασμό ερωτοαπαντήσεων του καθηγητή προς τους μαθητές συνεχίζεται με ανάλυση των χαρακτηριστικών της Python ως γλώσσα προγραμματισμού, δηλαδή γιατί θεωρείται μία γλώσσα γενικού σκοπού, γιατί έχει διαδεδομένη χρήση στη δημιουργία σύγχρονων εφαρμογών, γιατί είναι μια διερμηνευμένη γλώσσα. Κατά τη διάρκεια της εισήγησης ο καθηγητής μπορεί να προβάλει και σχετικές διαφάνειες. Επίσης, επειδή πρόκειται για ένα εισαγωγικό μάθημα με σκοπό να διεγείρει το ενδιαφέρον των μαθητών για τον προγραμματισμό, στο τέλος εκτελεί κάποια από προγράμματα που θα ασχοληθούν στο μέλλον ώστε να να σχηματίσουν οι μαθητές μία γνώμη σχετικά με το αντικείμενο που θα ασχοληθούν στο πλαίσιο των μαθημάτων.
- Στη συνέχεια οι μαθητές θα προσπαθήσουν να απαντήσουν σε σύντομες ερωτήσεις σχετικά με τις έννοιες που διδάχτηκαν. Οι απαντήσεις τους στις ερωτήσεις αυτές δεν έχουν αυστηρό ύφος αξιολόγησης, έχουν πιο πολύ ρόλο να καταλάβει ο καθηγητής τι έχει γίνει κατανοητό και τι όχι. Επειδή είναι δραστηριότητα επιλογής οι μαθητές μπορούν να την προσπεράσουν προχωρώντας στην επόμενη δραστηριότητα και να την επισκεφτούν στο τέλος του μαθήματος ξανά αν υπάρχει διαθέσιμος χρόνος.

- Έπειτα ο καθηγητής δίνει στους μαθητές οδηγίες σχετικά με το πώς θα κατεβάσουν την Python στους υπολογιστές τους καθώς και πώς θα εκτελέσουν το πρώτο τους πρόγραμμα. Αν βρίσκονται στο εργαστήριο, η Python θα είναι ήδη κατεβασμένη, ωστόσο δίνει μια αναλυτική περιγραφή με τα βήματα που θα πρέπει να ακολουθήσουν οι μαθητές ώστε να το επιχειρήσουν στη συνέχεια στον υπολογιστή του σπιτιού τους. Οι οδηγίες και τα βήματα που αναφέρει ο καθηγητής υπάρχουν αναλυτικά με μορφή κειμένου και εικόνων που μπορούν να έχουν πρόσβαση οι μαθητές σε αυτές όποτε το επιθυμούν.



**Εικόνα 10: Το μάθημα 'Εισαγωγή στον προγραμματισμό με Python' στο LessonLAMS**

- Στη συνέχεια οι μαθητές καλούνται να εκτελέσουν το πρώτο τους πρόγραμμα στο IDLE. Οι μαθητές διαβάζοντας τις οδηγίες που υπάρχουν προσπαθούν να τις εφαρμόσουν στους υπολογιστές του εργαστηρίου και ο καθηγητής ελέγχει την πρόοδό τους βοηθώντας όπου υπάρχει πρόβλημα. Το πρώτο τους πρόβλημα αποτελείται ουσιαστικά από μία εντολή εκτύπωσης μηνύματος. Παράλληλα με την προσπάθεια των μαθητών να εκτελέσουν το πρώτο τους πρόγραμμα μαθαίνουν και την πρώτη εντολή στην Python, την εκτύπωση συγκεκριμένου μηνύματος. Όταν τα καταφέρουν, ο καθηγητής τους προτρέπει να πειραματιστούν ώστε να εμφανίσουν ό,τι μηνύματα επιθυμούν.
- Αφού τελειώσουν με την εκτέλεση των δραστηριοτήτων, ο καθηγητής τους προτρέπει στο σπίτι να ανοίξουν την υποστηρικτική δραστηριότητα ώστε να επισκεφτούν μία ή περισσότερες από τις ιστοσελίδες που διατίθενται καθώς επίσης και να δοκιμάσουν να εκτελέσουν και στο σπίτι το πρόγραμμα που είδαν στο μάθημα.

## 6.2 Η Δομή Ακολουθίας στην Python

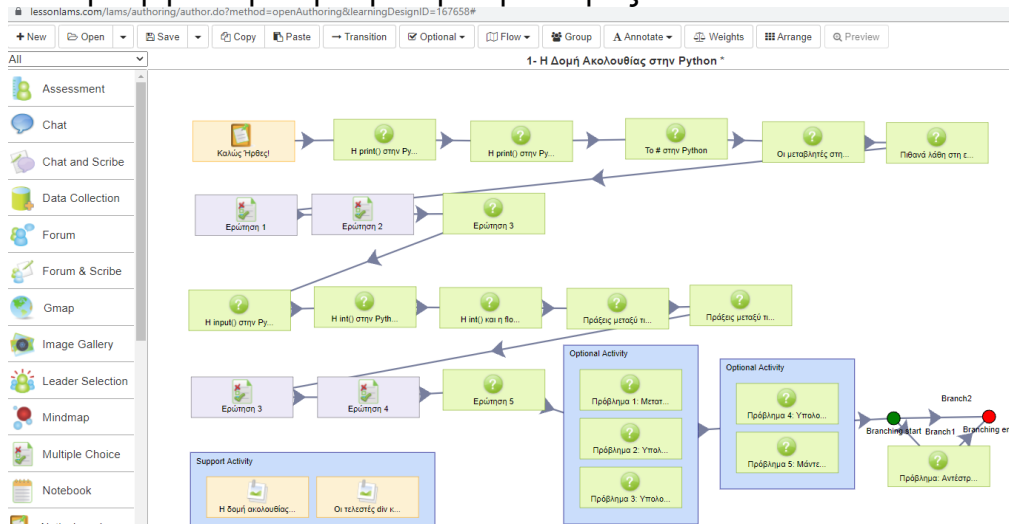
Το δεύτερο μάθημα είναι "Η Δομή Ακολουθίας στην Python" το οποίο έχει σκοπό να εισάγει τους μαθητές στη διαδικασία του προγραμματισμού και τη δημιουργία δικών τους προγραμμάτων πάνω στην δομή ακολουθίας.

### Μαθησιακοί Στόχοι:

Οι μαθησιακοί στόχοι που τίθενται σε αυτό το μάθημα είναι οι μαθητές μετά το πέρας του μαθήματος να μπορούν:

- Να αναγνωρίζουν την εντολή εξόδου, προσδιορίζουν το μήνυμα εκτύπωσης και να την χρησιμοποιούν για να εμφανίζουν στην οθόνη το επιθυμητό μήνυμα.

- Να ορίζουν και αναθέτουν τιμές σε μεταβλητές, χρησιμοποιώντας μνημονικές και έγκυρες ονομασίες.
- Να διακρίνουν και να κατονομάζουν τους διαφορετικούς τύπους μεταβλητών.
- Να αναγνωρίζουν την εντολή εισόδου και να εξηγούν τον τρόπο με τον οποίο χειρίζεται το πρόγραμμα τα δεδομένα που δίνονται ως είσοδο.
- Να αναγνωρίζουν τους αριθμητικούς τελεστές και να συντάσσουν μια αριθμητική παράσταση σύμφωνα με τη σειρά προτεραιότητας.



Εικόνα 11: Μάθημα "Η Δομή Ακολουθίας στην Python" στο LessonLAMS

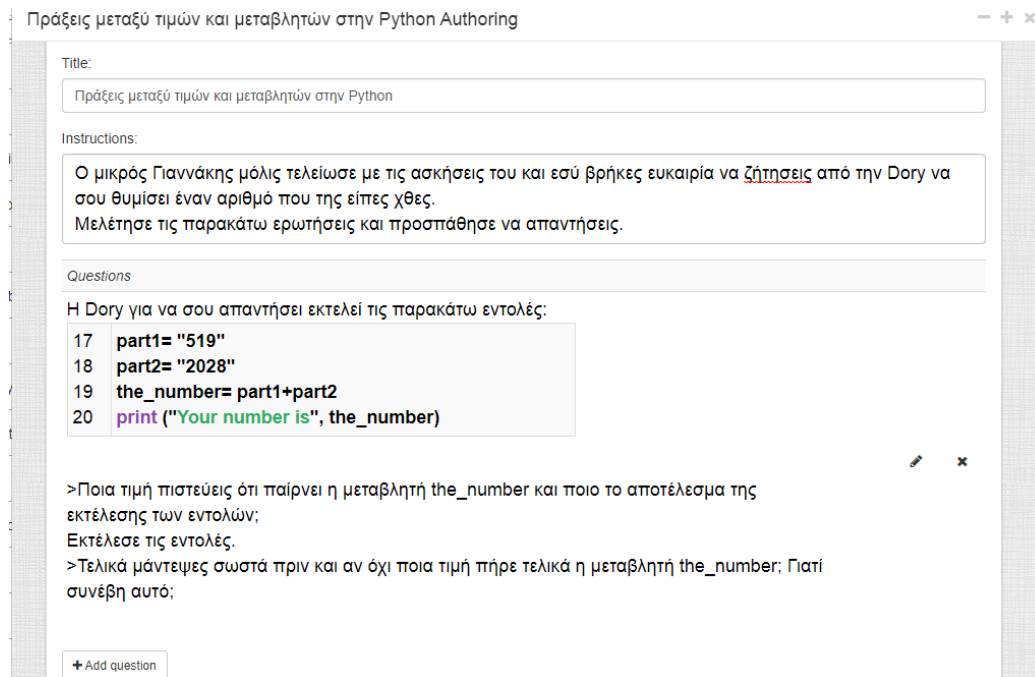
## Δραστηριότητες και Εργαλεία του LAMS:

Το μάθημα είναι σχεδιασμένο ενδεικτικά για 5 διδακτικές ώρες του Ομίλου 90 λεπτών και αποτελείται κυρίως από μία σειριακή ακολουθία δραστηριοτήτων ίδια για όλους τους μαθητές (Εικόνα 11). Πιο συγκεκριμένα αποτελείται από:

- 1 εργαλείο δραστηριοτήτων "Πίνακας Ανακοινώσεων" όπου παρατίθεται μία περιγραφή σε μορφή κειμένου για το αντικείμενο με το οποίο θα ασχοληθούν στο συγκεκριμένο μάθημα στο Lams όπως επίσης και οι μαθησιακοί στόχοι.
- 10 εργαλεία δραστηριοτήτων "Ερώτηση-Απάντηση" όπου υπάρχουν δραστηριότητες με βάση τη διδακτική προσέγγιση των "Διερευνήσεων". Το περιεχόμενο και οι ερωτήσεις έχουν καταναμηθεί στις 10 αυτές δραστηριότητες ανάλογα με τη θεματική ενότητα που ανήκει το περιεχόμενό τους, δηλαδή σχετικά με τη σύνταξη και τη λειτουργία της εντολής print() στη μία δραστηριότητα, σχετικά με τις πράξεις μεταξύ αλφαριθμητικών τιμών στην άλλη δραστηριότητα (Εικόνα 12) κ.ο.κ.
- 4 εργαλεία δραστηριοτήτων "Ερωτήσεις Πολλαπλής Επιλογής" στις οποίες τίθενται μικρά προβλήματα προς τους εκπαιδευόμενους ώστε να διαλέξουν τη σωστή απάντηση.
- 2 εργαλεία δραστηριοτήτων "Ερώτηση-Απάντηση" όπου υπάρχουν μικρά προβλήματα-ερωτήματα στα οποία οι εκπαιδευόμενοι πρέπει να απαντήσουν με ανάπτυξη λόγου.
- 2 δραστηριότητες επιλογής στις οποίες υπάρχουν 3 και 2 εργαλεία δραστηριοτήτων "Ερώτηση-Απάντηση" αντίστοιχα στις οποίες υπάρχουν προβλήματα προς επίλυση ανάπτυξης κώδικα διαβαθμισμένης δυσκολίας, πιο απλές στην πρώτη και πιο δύσκολες στη δεύτερη αντίστοιχα.
- 1 διακλάδωση η οποία περιέχει 1 εργαλείο δραστηριότητας "Ερώτηση-Απάντηση" στο οποίο υπάρχει ένα προγραμματιστικό πρόβλημα προς επίλυση.



- 1 υποστηρικτική δραστηριότητα η οποία περιέχει 2 εργαλεία δραστηριοτήτων "Παράθεση Πόρων" όπου παρατίθενται pdf αρχεία με θεωρία και παραδείγματα σχετικά με το μάθημα.



**Εικόνα 12: Δραστηριότητα του μαθήματος "Η Δομή Ακολουθίας στην Python" με το εργαλείο "Ερώτηση-Απάντηση" στο LessonLAMS σχετικά με τις πράξεις ανάμεσα σε αλφαριθμητικές τιμές.**

### Πορεία Μαθήματος:

- Αρχικά ο καθηγητής κάνει μία εισαγωγή στους μαθητές σχετικά με το αντικείμενο το οποίο θα ασχοληθούν στο συγκεκριμένο μάθημα στο LAMS, δηλαδή τη δομή ακολουθίας.
- Έπειτα οι μαθητές ξεκινούν να λύσουν 4 δραστηριότητες "Διερευνήσεων".. Τα τμήματα κώδικα που δίνονται αποτελούνται από εντολές εκτύπωσης μηνυμάτων print(), εισαγωγή σχολίων και αναθέσεις τιμών σε αριθμητικές και αλφαριθμητικές μεταβλητές. Στο διάστημα ενασχόλησης των μαθητών με τις συγκεκριμένες δραστηριότητες ο καθηγητής απαντά σε ερωτήσεις των μαθητών και τους καθοδηγεί. Σκοπός των διερευνητικών δραστηριοτήτων είναι να ενεργοποιήσουν τους μαθητές και με κατάλληλη καθοδήγηση να φτάσουν στη γνώση.
- Με αφορμή τις δραστηριότητες αυτές και τις ερωτήσεις των μαθητών, ο καθηγητής προχωράει σε μια ολιγόλεπτη εισήγηση που αναλύει περαιτέρω τη λειτουργία των νέων εντολών που είδαν οι μαθητές στις προηγούμενες δραστηριότητες. Για την εισήγηση χρησιμοποιούνται διαφάνειες οι οποίες είναι διαθέσιμες στους μαθητές μέσω της υποστηρικτικής δραστηριότητας παράθεσης πόρων.
- Στη συνέχεια οι μαθητές καλούνται να εφαρμόσουν τις γνώσεις απαντώντας σε 3 ερωτήματα, 2 πολλαπλής επιλογής και 1 ανάπτυξης. Μόλις δοθούν οι απαντήσεις, όλοι μαζί με τον καθηγητή συζητούν τις απαντήσεις που δόθηκαν. Ο καθηγητής έχοντας εικόνα των απαντήσεων των μαθητών μέσω του περιβάλλοντος LAMS, προτρέπει τους μαθητές να εκφραστούν, εστιάζοντας σε απαντήσεις-παρανοήσεις, ώστε μέσα από τη συζήτηση και την ανταλλαγή απόψεων να αντιμετωπιστούν.

- Έπειτα οι μαθητές καλούνται πάλι να μελετήσουν 5 δραστηριότητες "Διερευνήσεων", με τον τρόπο που αναφέρθηκε παραπάνω, οι οποίες σχετίζονται με εντολές εισχώρησης τιμής `input()`, συναρτήσεις μετατροπής τιμών σε ακέραιες και πραγματικές τιμές, `int()` και `float()` αντίστοιχα, και πράξεις μεταξύ τιμών και μεταβλητών με τη βοήθεια αριθμητικών τελεστών.
- Στην συνέχεια ο καθηγητής κάνει μία ολιγόλεπτη εισήγηση ώστε να αναλύσει τη λειτουργία των εντολών που είδαν οι μαθητές. Η εισήγηση πάλι θα συνοδεύεται από προβολή διαφανειών οι οποίες είναι διαθέσιμες μέσω υποστηρικτικής δραστηριότητας παράθεσης πόρων.
- Μετά το πέρας της εισήγησης οι μαθητές προσπαθούν να εφαρμόσουν τις νέες γνώσεις τους απαντώντας σε 3 ερωτήματα, 2 πολλαπλής επιλογής και 1 ανάπτυξης πρώτα μόνοι τους και στη συνέχεια συζητώντας τις απαντήσεις όλοι μαζί με τον καθηγητή.
- Ακολουθεί η επίλυση προβλημάτων για τα οποία οι μαθητές θα πρέπει να υλοποιήσουν και να εκτελέσουν τα δικά τους προγράμματα. Τα προβλήματα αυτά έχουν ταξινομηθεί σε δύο ομάδες (μία ομάδα με πιο απλά προβλήματα και μία με προβλήματα αυξημένης δυσκολίας) και διατίθενται στους μαθητές με τη μορφή 2 δραστηριοτήτων επιλογής. Οι μαθητές μελετούν τα προβλήματα αυτά, προσπαθούν να τα λύσουν και παραθέτουν στη συνέχεια το αντίστοιχο πρόγραμμά τους ως λύση. Ο καθηγητής ανάλογα με τον διαθέσιμο χρόνο που υπάρχει καθορίζει ποια από αυτά τα προβλήματα θα συζητηθούν και θα παρουσιαστούν στη διάρκεια του μαθήματος και ποια θα αποτελέσουν ως προαιρετική εργασία για το σπίτι. Σκοπός είναι να συζητηθεί ένα πρόβλημα από την κάθε ομάδα. Για το εκάστοτε πρόβλημα αναθέτει σε έναν μαθητή να παρουσιάσει τη λύση του προς τους υπόλοιπους αιτιολογώντας τις επιλογές του. Ο καθηγητής έχοντας συνολική εικόνα από τις απαντήσεις των μαθητών, κατευθύνει τη διαδικασία κάνοντας ερωτήσεις και παροτρύνοντας τους υπόλοιπους μαθητές να συμμετέχουν ώστε να λύσει τυχόν παρανοήσεις που έχουν προκύψει.
- Στη διακλάδωση υπάρχει μία δραστηριότητα που παραθέτει ένα ακόμη πρόβλημα το οποίο για να λυθεί απαιτεί συγκεκριμένη μέθοδο κατάλληλης χρήσης των τελεστών της ακέραιας διαίρεσης και του υπολοίπου της ακέραιας διαίρεσης. Για να λυθεί το πρόβλημα αυτό χρειάζεται ο καθηγητής να κάνει μία εισαγωγή με τη μέθοδο ερωτοαπαντήσεων δείχνοντας παραδείγματα προς τους μαθητές. Η λογική της μεθόδου και τα παραδείγματα παρατίθενται και σε μορφή αρχείου στους μαθητές στην υποστηρικτική δραστηριότητα ώστε να έχουν πρόσβαση σε αυτό ανεξάρτητα της κύριας ακολουθίας. Επειδή πρόκειται για ένα απαιτητικό πρόβλημα που η λύση του πρέπει να ακολουθήσει συγκεκριμένη μεθοδολογία ο καθηγητής ανάλογα με το επίπεδο των μαθητών και το περιθώριο χρόνου που έχει στη διάθεσή του επιλέγει σε ποιους μαθητές θα την αναθέσει.

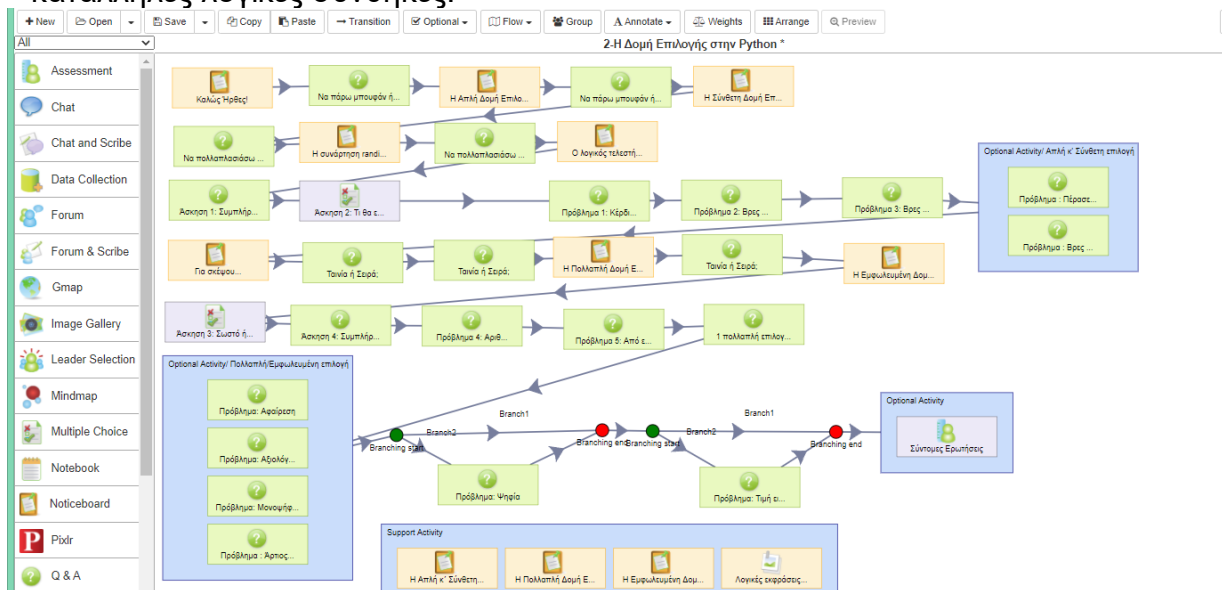
### 6.3 Η Δομή Επιλογής στην Python

Το τρίτο μάθημα είναι "Η Δομή Επιλογής στην Python" το οποίο έχει σκοπό τη μελέτη των τεσσάρων μορφών της δομής επιλογής, απλή-σύνθετη-πολλαπλή-εμφωλευμένη και την επίλυση αντίστοιχων προβλημάτων που καθιστούν τη χρήση της δομής επιλογής απαραίτητη για τη λύση τους.

#### Μαθησιακοί Στόχοι:

Οι μαθησιακοί στόχοι που τίθενται σε αυτό το μάθημα είναι οι μαθητές μετά το πέρας του μαθήματος να μπορούν:

- Να κατονομάζουν τις τέσσερις μορφές της δομής επιλογής.
- Να εξηγούν τις διαφορές που έχουν οι δομές επιλογής ως προς την σύνταξη και ως προς τη λειτουργία τους.
- Να αναγνωρίζουν σε δοσμένο κώδικα σε ποια μορφή ανήκει η εκάστοτε δομή επιλογής.
- Να υπολογίζουν τις διαφορετικές τιμές που μπορεί να πάρει μια λογική συνθήκη (Αληθής Ψευδής) και να εντοπίζουν ποια ή ποιες εντολές θα εκτελεστούν ανάλογα με την τιμή της συνθήκης.
- Να αποφασίζουν ποια είναι η κατάλληλη δομή επιλογής που πρέπει να χρησιμοποιήσουν στο πρόγραμμά τους για την επίλυση δοσμένου προβλήματος.
- Να συντάξουν προγράμματα που απαιτούν χρήση δομών επιλογής δημιουργώντας κατάλληλες λογικές συνθήκες.



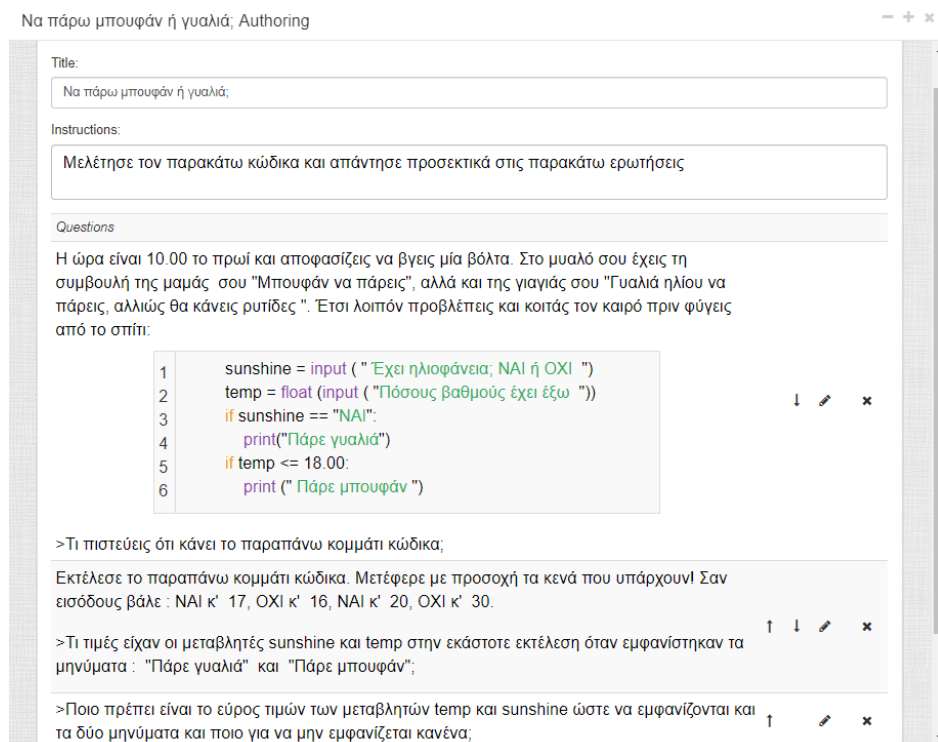
Εικόνα 13: Το μάθημα "Η Δομή Επιλογής στην Python" στο LessonLAMS

### Δραστηριότητες και Εργαλεία του LAMS:

Το μάθημα είναι σχεδιασμένο ενδεικτικά για 5 διδακτικές ώρες του Ομίλου 90 λεπτών. Αποτελείται από μία ακολουθία δραστηριοτήτων, κυρίως σειριακή κοινή για όλους τους μαθητές (Εικόνα 13) μαζί με δύο διακλαδώσεις που περιλαμβάνουν από ένα πιο απαιτητικό πρόβλημα η κάθε μια. Πιο συγκεκριμένα αποτελείται από:

- 2 εργαλεία δραστηριοτήτων "Πίνακας Ανακοινώσεων", ένα στην αρχή όπου παρατίθεται μία περιγραφή σε μορφή κειμένου για το αντικείμενο με το οποίο θα ασχοληθούν γενικά οι μαθητές στο συγκεκριμένο μάθημα στο Lams και οι μαθησιακοί στόχοι και ένα στη μέση του μαθήματος για να συνδέσει το περιεχόμενο των μαθησιακών δραστηριοτήτων που προηγήθηκαν με αυτές που θα ακολουθήσουν.
- 7 εργαλεία δραστηριοτήτων "Ερώτηση-Απάντηση" όπου υπάρχουν δραστηριότητες με βάση τη διδακτική προσέγγιση των "Διερευνήσεων" (Εικόνα 14). Έχουν δημιουργηθεί 3 δραστηριότητες διερευνήσεων, η πρώτη σχετίζεται με τη μορφή της απλής και της σύνθετης δομής επιλογής, η δεύτερη με τη χρήση της συναρτησης randint() και τη δημιουργία σύνθετων λογικών συνθηκών με χρήση λογικών τελεστών και η τρίτη σχετίζεται με τη μορφή της πολλαπλής και της εμφωλευμένης δομής επιλογής. Οι 3 αυτές δραστηριότητες των διερευνήσεων έχουν αποδοθεί σε αυτά τα 7 εργαλεία

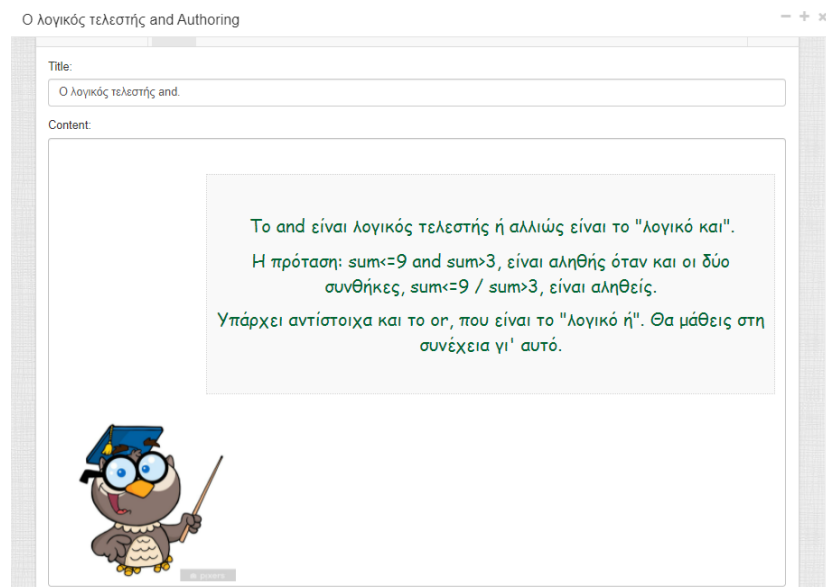
- δραστηριοτήτων "Ερώτηση-Απάντηση" ώστε οι μαθητές να προχωράνε βήμα-βήμα και σε κάθε εργαλείο να παρουσιάζεται μία νέα έννοια την οποία θα μελετήσουν.
- 6 εργαλεία δραστηριοτήτων "Πίνακας Ανακοινώσεων" όπου υπάρχουν μικρά κομμάτια θεωρίας σε μορφή κειμένου (Εικόνα 15) και έχουν τοποθετηθεί ανάμεσα στις διερευνητικές δραστηριότητες σαν υποστηρικτικές συμβουλές.
- 2 εργαλεία δραστηριοτήτων "Πολλαπλή Επιλογή" στις οποίες υπάρχουν ερωτήματα πολλαπλής επιλογής ώστε να επιλέξουν οι μαθητές τη σωστή απάντηση. Οι δραστηριότητες αυτές δεν έχουν ρόλο αξιολόγησης και επίσης έχει προστεθεί ανατροφοδότηση που μπορούν να δουν οι μαθητές μετά την υποβολή της απάντησης (Εικόνα 16).
- 8 δραστηριότητες "Ερώτηση-Απάντηση" στις οποίες υπάρχουν ερωτήματα που πρέπει να απαντηθούν με ανάπτυξη λόγου και προβλήματα προς επίλυση που απαιτούν την δημιουργία και την εκτέλεση προγραμμάτων από τους μαθητές. Για τα προβλήματα που απαιτούν τη δημιουργία και εκτέλεση προγραμμάτων στο συγκεκριμένο μάθημα αλλά και στα υπόλοιπα, χρησιμοποιώντας το εργαλείο δραστηριοτήτων "Ερώτηση-Απάντηση" οι μαθητές μπορούν να υποβάλουν τον κώδικά τους ως απάντηση. Για τον σκοπό αυτό θα μπορούσε να χρησιμοποιηθεί και το εργαλείο δραστηριοτήτων "Υποβολή Αρχείου" υποβάλλοντας οι μαθητές κατευθείαν το αρχείο με το πρόγραμμά τους. Ωστόσο για τα προβλήματα που υπάρχουν στο συγκεκριμένο μάθημα προτιμήθηκε το εργαλείο "Ερώτηση-Απάντηση" επειδή τα προγράμματα που απαιτείται να δημιουργηθούν είναι σχετικά μικρά σε έκταση και ο καθηγητής μπορεί να έχει πιο άμεση πρόσβαση στην λύση.



**Εικόνα 14: Εργαλείο δραστηριότητας "Ερώτηση-Απάντηση" για παράθεση διερευνητικής δραστηριότητας στο μάθημα "Η Δομή Επιλογής στην Python".**

- 2 δραστηριότητες επιλογής στις οποίες υπάρχουν 2 και 4 αντίστοιχα δραστηριότητες "Ερώτηση-Απάντηση" με προβλήματα προς επίλυση. Τα προβλήματα αυτά έχουν χωριστεί σε δύο ομάδες ανάλογα με την ενότητα στην οποία ανήκουν. Στην πρώτη

ομάδα των δραστηριοτήτων επιλογής υπάρχουν προβλήματα που απαιτούν τη χρήση απλής-σύνθετης δομής επιλογής και στη δεύτερη ομάδα δραστηριοτήτων προβλήματα που απαιτούν πολλαπλή-εμφωλευμένη δομή επιλογής. Οι μαθητές μπορούν να επιλέξουν να εκτελέσουν όσες από αυτές επιθυμούν.



Εικόνα 15: Εργαλείο δραστηριότητας "Πίνακας Ανακοινώσεων" στο μάθημα "Η Δομή Επιλογής στην Python".

- 2 διακλαδώσεις που περιλαμβάνουν από μία δραστηριότητα "Ερώτηση-Απάντηση" η κάθε μία. Στις δραστηριότητες αυτές υπάρχουν 2 προβλήματα πιο απαιτητικά από όσα έχουν προηγηθεί και είναι στην επιλογή του καθηγητή αν θα τα αναθέσει σε κάποιους από τους μαθητές, σε όλους ή σε κανέναν ανάλογα του επιπέδου τους.
- 1 δραστηριότητα επιλογής η οποία περιλαμβάνει 1 δραστηριότητα "Αξιολόγηση" με ερωτήσεις κλειστού τύπου, πολλαπλής επιλογής, σωστό-λάθος, αντιστοίχισης, για να απαντήσουν οι μαθητές.
- 1 υποστηρικτική δραστηριότητα η οποία περιλαμβάνει 3 δραστηριότητες "Πίνακας Ανακοινώσεων" και 1 "Παράθεση πόρων" οι οποίες περιλαμβάνουν όλη τη θεωρία σχετικά με τη δομή επιλογής στην Python. Στις δραστηριότητες "Πίνακας Ανακοινώσεων" παρέχονται σε μορφή κειμένου ενώ στην "Παράθεση Πόρων" σε μορφή αρχείου pdf.

### Πορεία Μαθήματος:

- Αρχικά ο καθηγητής κάνει μία υπενθύμιση στους μαθητές σχετικά με το τι έχουν μάθει μέχρι στιγμής στα μαθήματα, ποια ανάγκη προκύπτει στην επίλυση προβλημάτων όταν πρέπει να εκτελεστούν συγκεκριμένες εντολές υπό συγκεκριμένες συνθήκες και αμέσως μετά τους εισάγει στο νέο αντικείμενο με το οποίο θα ασχοληθούν στο συγκεκριμένο μάθημα στο LAMS, δηλαδή τη δομή επιλογής. Ο καθηγητής για να κάνει αυτή τη μικρή εισαγωγή χρησιμοποιεί ερωτοαποκρίσεις ώστε να ενεργοποιήσει τους μαθητές και να διεγείρει το ενδιαφέρον τους για το νέο αντικείμενο που θα μελετήσουν.
- Έπειτα οι μαθητές ξεκινούν να λύσουν 4 δραστηριότητες "Διερευνήσεων". Οι δραστηριότητες των "Διερευνήσεων" περιέχουν μικρά τμήματα κώδικα τα οποία οι μαθητές καλούνται να μελετήσουν και να απαντήσουν σε ερωτήσεις σχετικά με τη

λειτουργία τους και στη συνέχεια εκτελώντας τον κώδικα να παρατηρήσουν το αποτέλεσμα και να το συγκρίνουν με τις δικές τους απαντήσεις. Στις 4 αυτές δραστηριότητες τα τμήματα κώδικα που δίνονται περιλαμβάνουν την απλή και τη σύνθετη δομή επιλογής, σύνθετες λογικές συνθήκες με χρήση λογικών τελεστών, τη συνάρτηση `randint()` για δημιουργία ψευδοτυχαίων αριθμών και εντολές φυσικά που έμαθαν στα προηγούμενα μαθήματα. Πρώτα καλούνται να μελετήσουν τον κώδικα και να απαντήσουν στις ερωτήσεις μόνοι τους και μετά όλοι μαζί με τον καθηγητή να συζητήσουν για την ορθότητα των απαντήσεων που έδωσαν συνολικά. Μετά από κάθε δραστηριότητα, επειδή είναι λογικό να προκληθούν απορίες στους μαθητές, υπάρχουν τμήματα θεωρίας σχετικά με τις εντολές που είδαν προηγουμένως ώστε να μπορούν να προχωρήσουν μόνοι τους στις επόμενες δραστηριότητες, χωρίς την παρέμβαση καθηγητή. Ωστόσο στο διάστημα αυτό, ο καθηγητής είναι παρόν ώστε να απαντήσει σε οποιοσδήποτε ερωτήσεις μαθητών προκύψουν.

Ασκηση 2: Τι θα εκτυπωθεί Authoring

```

Τι θα εκτυπώσει το παρακάτω πρόγραμμα,
a=1
b=2
c=3
if a+b==c or a>b and not c>a:
    print ("A")
else:
    print ("B")

```

Answers: Correct?

1  A ⌵ ×

2  B ⌵ ×

[Add answer](#)

This question is worth  mark(s)

[Question feedback](#)

A διότι η συνθήκη: `a+b==c or a>b and not c>a` είναι True.  
 Η προτεραιότητα στους λογικούς τελεστές είναι η εξής: **not, and, or**.  
 Οπότε έχουμε: `a+b==c or (a>b and (not c>a))`->  
`a+b==c or (False and False)`->  
`a+b==c or False`->  
 True or False->  
 True

**Εικόνα 16: Εργαλείο δραστηριότητας "Πολλαπλή Επιλογή" για παράθεση ερώτησης πολλαπλής επιλογής στο μάθημα "Η Δομή Επιλογής στην Python".**

- Με αφορμή τις δραστηριότητες και τις ερωτήσεις των μαθητών, ο καθηγητής προχωράει σε μια ολιγόλεπτη εισήγηση που αναλύει περαιτέρω τη λειτουργία της απλής και της σύνθετης δομής επιλογής. Για την εισήγηση θα χρησιμοποιήσει διαφάνειες που θα προβάλλονται στον πίνακα.
- Στη συνέχεια οι μαθητές καλούνται να εφαρμόσουν τις γνώσεις στις ασκήσεις που ακολουθούν. Μία άσκηση συμπλήρωσης κενών σε δοσμένο κώδικα, μία ερώτηση πολλαπλής επιλογής και 3 προβλήματα στα οποία καλούνται να δημιουργήσουν και να εκτελέσουν τον δικό τους κώδικα. Ο καθηγητής δίνει χρόνο στους μαθητές ώστε να μελετήσουν τις ασκήσεις και να υποβάλλουν τις απαντήσεις τους. Μόλις δοθούν οι απαντήσεις, όλοι μαζί με τον καθηγητή συζητούν τις απαντήσεις που δόθηκαν. Ο καθηγητής έχοντας εικόνα των απαντήσεων των μαθητών δίνει τον λόγο σε αυτούς και τους προτρέπει να παρουσιάσουν την άποψή τους στους υπόλοιπους και μέσα από τη συζήτηση και την ανταλλαγή απόψεων να αντιμετωπιστούν και τυχόν παρανοήσεις που έχουν προκύψει.

- Μετά τη μελέτη των προηγούμενων ασκήσεων ακολουθούν 2 προβλήματα επιλογής σχετικά με την απλή και τη σύνθετη δομή επιλογής. Ο καθηγητής, ανάλογα με τον διαθέσιμο χρόνο που υπάρχει, αφήνει τους μαθητές να ασχοληθούν με όσες από αυτές επιθυμούν είτε στην ώρα του μαθήματος είτε ως εργασία στο σπίτι.
- Αφού λοιπόν οι μαθητές εκπονήσουν δραστηριότητες χρησιμοποιώντας την απλή και τη σύνθετη δομή επιλογής, ο καθηγητής κάνει μία μικρή αναφορά σχετικά με την ανάγκη που προκύπτει όταν οι περιπτώσεις για την επίλυση ενός προβλήματος είναι παραπάνω από δύο. Με αυτό τον τρόπο προϊδεάζει τους μαθητές για την πολλαπλή επιλογή που θα ασχοληθούν στη συνέχεια.
- Στη συνέχεια οι μαθητές εκπονούν τρεις διερευνητικές δραστηριότητες πάνω στην πολλαπλή και στην εμφωλευμένη δομή επιλογής. Ο τρόπος διεξαγωγής των δραστηριοτήτων αυτών είναι ίδιος όπως παραπάνω, αντίστοιχα δηλαδή με τις διερευνητικές δραστηριότητες που προηγήθηκαν.
- Αφού εκπονήσουν οι μαθητές τις διερευνητικές δραστηριότητες και σχολιαστούν οι απαντήσεις ο καθηγητής, όπως παραπάνω, προχωράει σε μια ολιγόλεπτη εισήγηση που θα αναλύσει τη λειτουργία της πολλαπλής και της εμφωλευμένης δομής επιλογής. Για την εισήγηση θα χρησιμοποιήσει διαφάνειες που θα προβάλλονται στον πίνακα.
- Σειρά έχουν οι ασκήσεις. Μία άσκηση πολλαπλής επιλογής, μία συμπλήρωση κενών, δύο προβλήματα προς επίλυση και μία άσκηση σχολιασμού δοσμένου κώδικα. Όπως παραπάνω, οι μαθητές προσπαθούν να τις επιλύσουν αρχικά μόνοι τους και στην συνέχεια όλοι μαζί σχολιάζουν τις απαντήσεις παροτρύνοντας τους μαθητές να παρουσιάσουν τις απαντήσεις τους στους υπόλοιπους.
- Ακολουθούν 4 προβλήματα επιλογής σχετικά με την πολλαπλή και την εμφωλευμένη δομή επιλογής. Ο καθηγητής, ανάλογα με τον διαθέσιμο χρόνο που υπάρχει, αφήνει τους μαθητές να ασχοληθούν με όσες από αυτές επιθυμούν στην ώρα του μαθήματος ή παροτρύνοντάς τους στο σπίτι.
- Ύστερα υπάρχουν δύο πιο απαιτητικά προβλήματα τα οποία βρίσκονται σε διακλάδωση. Ανάλογα με το επίπεδο των μαθητών και τον διαθέσιμο χρόνο, ο καθηγητής αναθέτει τις δραστηριότητες αυτές στους μαθητές.
- Τέλος υπάρχει μία δραστηριότητα επιλογής με ερωτήσεις κλειστού τύπου. Η δραστηριότητα αυτή δεν έχει ρόλο αξιολόγησης αλλά να καταλάβει ο καθηγητής τι δεν έχει γίνει κατανοητό από τους μαθητές και τι τους δυσκολεύει. Η δραστηριότητα μπορεί να ανατεθεί στους μαθητές την ώρα του μαθήματος αν υπάρχει χρόνος ή προτρέποντάς τους να την εκτελέσουν στο σπίτι.
- Η θεωρία σχετικά με τις δομές επιλογής καθώς και των λογικών τελεστών για τη σωστή δημιουργία συνθηκών παρέχονται στην υποστηρικτική δραστηριότητα. Οι μαθητές μπορούν να έχουν πρόσβαση σε αυτή όποτε το επιθυμούν.

#### 6.4 Οι Συμβολοσειρές στην Python

Το τέταρτο μάθημα σχετίζεται με τις συμβολοσειρές και έχει σκοπό την εκμάθηση διαχείρισης των συμβολοσειρών αξιοποιώντας κατάλληλα βασικές συναρτήσεις και μεθόδους που προσφέρει η Python.

##### **Μαθησιακοί Στόχοι :**

Οι μαθησιακοί στόχοι που τίθενται σε αυτό το μάθημα είναι οι μαθητές μετά το πέρας του μαθήματος να μπορούν να:

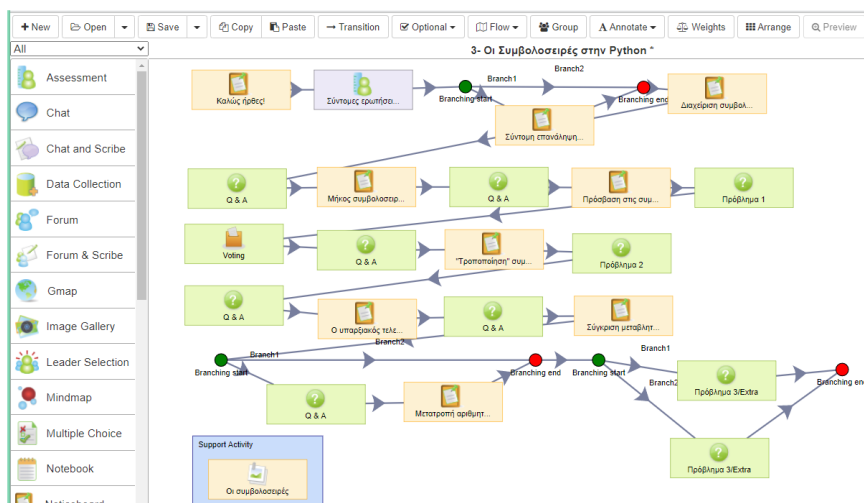
- Να αποσπούν συγκεκριμένο μέρος μιας συμβολοσειράς.
- Να δημιουργούν και να αποθηκεύουν μια νέα συμβολοσειρά χρησιμοποιώντας μέρη άλλων συμβολοσειρών .

- Να ελέγχουν αν μια συμβολοσειρά είναι μέρος μιας μεγαλύτερης.
- Να βρίσκουν το μέγεθος μιας συμβολοσειράς.
- Να συγκρίνουν 2 συμβολοσειρές μεταξύ τους.

### Δραστηριότητες και Εργαλεία του LAMS:

Το μάθημα είναι σχεδιασμένο ενδεικτικά για 3 διδακτικές ώρες του Ομίλου 90 λεπτών. Αποτελείται από μία ακολουθία δραστηριοτήτων, σειριακή στο μεγαλύτερο μέρος της κοινή για όλους τους μαθητές (Εικόνα 17) μαζί με τρεις διακλαδώσεις. Πιο συγκεκριμένα αποτελείται από:

- 8 εργαλεία δραστηριοτήτων “Πίνακας Ανακοινώσεων” στα οποία παρατίθενται οι στόχοι του μαθήματος και τμήματα θεωρίας σχετικά με τον τρόπο διαχείρισης των συμβολοσειρών και τις δυνατότητες που παρέχει η Python. Οι πληροφορίες αυτές έχουν τη μορφή κειμένου.
- 1 εργαλείο “Αξιολόγηση” στο οποίο υπάρχουν ερωτήσεις πολλαπλής επιλογής και σωστού-λάθους σχετικά με τη διαχείριση συμβολοσειρών που έχουν διδαχθεί οι μαθητές σε προηγούμενα μαθήματα.
- 5 εργαλεία “Ερώτηση-Απάντηση” στα οποία υπάρχουν δραστηριότητες με την μέθοδο των “Διερευνήσεων”.
- 2 εργαλεία “Ερώτηση-Απάντηση” στα οποία παρατίθενται προβλήματα που απαιτούν δημιουργία και εκτέλεση προγραμμάτων από τους μαθητές για την επίλυσή τους .



Εικόνα 17: Το μάθημα “Οι Συμβολοσειρές στην Python” στο LAMS

- 1 εργαλείο “Ψηφοφορία” στο οποίο θέτει ένα ερώτημα στους μαθητές ώστε να ψηφίσουν την απάντηση που πιστεύουν ότι ισχύει.
- 1 εργαλείο “Διακλάδωση” στο οποίο υπάρχει ένας “Πίνακας Ανακοινώσεων” με μια σύντομη επανάληψη σχετικά με όσα ήδη έχουν διδαχθεί σχετικά με τις συμβολοσειρές στα προηγούμενα μαθήματα και χρειάζονται για την εκτέλεση των επόμενων δραστηριοτήτων στο συγκεκριμένο μάθημα. Η διακλάδωση αυτή βρίσκεται μετά από μία αξιολόγηση και έχει ρυθμιστεί ώστε να οδηγούνται οι μαθητές με βάση το βαθμό που πήραν στην προηγούμενη δραστηριότητας της αξιολόγησης.
- 1 εργαλείο “Διακλάδωση” το οποίο περιέχει δύο εργαλεία δραστηριοτήτων, “Ερώτηση-Απάντηση” και “Πίνακας Ανακοινώσεων”. Στα δύο αυτά εργαλεία περιέχεται μία



δραστηριότητα διερεύνησης και το αντίστοιχο τμήμα θεωρίας που σχετίζεται με την νέα γνώση που υπάρχει στην δραστηριότητα διερεύνησης.

- 1 εργαλείο “Διακλάδωση” το οποίο περιέχει 2 δραστηριότητες “Ερώτηση-Απάντηση”. Κάθε μία από αυτές τις δραστηριότητες βρίσκεται αντίστοιχα στα δύο διαφορετικά μονοπάτια που δημιουργεί η διακλάδωση (Εικόνα 17) και περιέχουν ένα πρόβλημα προς επίλυση από τους μαθητές εφαρμόζοντας τις γνώσεις που πήραν από το συγκεκριμένο μάθημα. Στο ένα μονοπάτι βρίσκεται το πρόβλημα στην απλοποιημένη μορφή του ενώ στο άλλο μονοπάτι βρίσκεται το ίδιο πρόβλημα με περισσότερα βήματα και λίγο πιο απαιτητικό. Η διακλάδωση έχει ρυθμιστεί ώστε ο καθηγητής να ρυθμίσει ποια δραστηριότητα θα εκτελεστεί από ποιους μαθητές.
- 1 “Υποστηρικτική” δραστηριότητα η οποία περιλαμβάνει ένα εργαλείο δραστηριότητας “Παράθεση Πόρων” στην οποία παρατίθενται δύο διαδικτυακοί σύνδεσμοι με επιπλέον πληροφορίες σχετικά με τις συμβολοσειρές. Ένα βίντεο σχετικά με την αποθήκευση των συμβολοσειρών στην μνήμη του υπολογιστή και ένας ιστότοπος που παρέχει όλες τις συναρτήσεις, μεθόδους και δυνατότητες διαχείρισης που παρέχει η Python για τις συμβολοσειρές.

### Πορεία Μαθήματος:

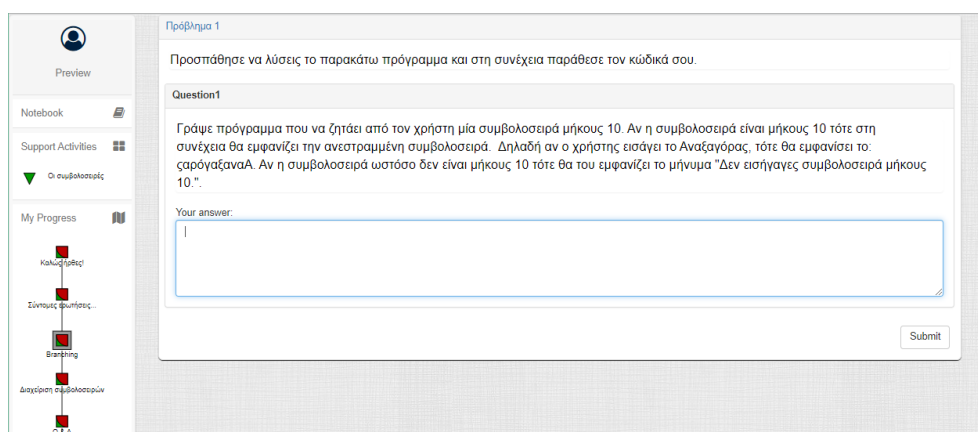
- Αρχικά ο καθηγητής κάνει μια εισαγωγή στους μαθητές λέγοντάς τους ότι στα προηγούμενα μαθήματα έμαθαν πώς να ορίζουν και πώς να αναθέτουν συμβολοσειρές στις μεταβλητές καθώς και πώς να ενώνουν δύο ή παραπάνω συμβολοσειρές. Στα επόμενα μαθήματα που ακολουθούν θα μάθουν πώς μπορούν να έχουν πρόσβαση σε ένα συγκεκριμένο μέρος μιας συμβολοσειρές και πώς να τις διαχειρίζονται μαθαίνοντας παράλληλα μερικές από τις πιο βασικές και βοηθητικές συναρτήσεις και μεθόδους που προσφέρει η Python.

Εικόνα 18: Τμήμα θεωρίας σχετικά με τη συνάρτηση len() στο μάθημα “Οι Συμβολοσειρές στην Python” στο περιβάλλον του εκπαιδευμένου στο LAMS

- Ωστόσο, επειδή οι προηγούμενες γνώσεις πάνω στις συμβολοσειρές είναι απαραίτητες για να προχωρήσουν με τις δραστηριότητες του συγκεκριμένου μαθήματος, οι μαθητές καλούνται να απαντήσουν σε μερικές ερωτήσεις κατανόησης κλειστού τύπου. Οι ερωτήσεις αυτές αναφέρονται κυρίως σε έννοιες και εντολές σχετικές με τις συμβολοσειρές που διδάχθηκαν σε προηγούμενα μαθήματα. Για τη δραστηριότητα εξάγεται ενδεικτικός βαθμός αξιολόγησης σχετικά με τις απαντήσεις που δίνουν οι μαθητές. Όσοι δεν απαντήσουν σωστά σε όλες τις ερωτήσεις, που σημαίνει ότι

μάλλον έχουν κάποια κενά με όσα έχουν διδαχθεί περί συμβολοσειρών στα προηγούμενα μαθήματα, οδηγούνται σε μια μικρή περιγραφή που τους υπενθυμίζει τα όσα έχουν διδαχθεί και χρειάζονται για να προχωρήσουν με τις δραστηριότητες που ακολουθούν. Όσοι απαντήσουν σωστά σε όλες τις ερωτήσεις προχωρούν στην επόμενη δραστηριότητα.

- Στην συνέχεια, οι μαθητές εκπονοούν 2 δραστηριότητες Διερευνήσεων. Οι δραστηριότητες αυτές περιλαμβάνουν τμήματα κώδικα που περιέχουν εντολές σχετικά με τον υπολογισμό μήκους μιας συμβολοσειράς και τον τρόπο πρόσβασης που μπορεί να αποκτήσει κάποιος σε συγκεκριμένο τμήμα της συμβολοσειράς. Ανάμεσα στις δραστηριότητες αυτές, υπάρχουν αντίστοιχα τμήματα θεωρίας που αναλύουν το ρόλο και τη λειτουργία των καινούριων εντολών που υπήρχαν στις δραστηριότητες. Τα τμήματα θεωρίας (Εικόνα 18) υπάρχουν ανάμεσα από τις δραστηριότητες ώστε οι μαθητές να μπορούν ανεξάρτητοι να εκτελούν τις δραστηριότητες και προχωρούν στις επόμενες. Ωστόσο ο καθηγητής είναι παρόν και έχει το ρόλο να τους βοηθάει και να λύνει οποιαδήποτε απορία προκύψει.
- Μετά την εκπόνηση των δραστηριοτήτων, δημιουργείται διάλογος μεταξύ του καθηγητή και των μαθητών ώστε να συζητήσουν τις απαντήσεις που έδωσαν και μέσα από αυτόν το διάλογο να αποσαφηνιστούν πιθανές απορίες που έχουν προκύψει και να αναλυθούν οι νέες εντολές που υπήρχαν στις διερευνητικές δραστηριότητες.
- Έπειτα οι μαθητές καλούνται να λύσουν ένα πρόβλημα (Εικόνα 19) στο οποίο πρέπει να δημιουργήσουν και να εκτελέσουν το δικό τους πρόγραμμα εφαρμόζοντας στην πράξη τις γνώσεις που πήραν. Οι μαθητές προσπαθούν να λύσουν το πρόβλημα αρχικά ατομικά και στη συνέχεια παρουσιάζεται η λύση από έναν μαθητή εξηγώντας τα βήματα που ακολούθησε και στη συνέχεια, στο πλαίσιο διαλόγου που δημιουργείται ανάμεσα σε όλους τους μαθητές με τον καθηγητή, η λύση αυτή αναλύεται περαιτέρω και διορθώνεται αν τυχόν χρειάζεται διορθώσεις.
- Μετά οι μαθητές καλούνται να απαντήσουν σε μία ερώτηση ψηφίζοντας την απάντηση την οποία πιστεύουν. Η ερώτηση τίθεται στους μαθητές ώστε να εκφράσουν την άποψη τους πάνω σε μία πρόταση σχετικά με την “τροποποίηση” που μπορεί να υποστεί μία συμβολοσειρά, έννοια που δεν έχουν διδαχθεί ακόμη. Ωστόσο σκοπός αυτή της ερώτησης είναι να τους ενεργοποιήσει και θα διεγείρει το ενδιαφέρον τους για την δραστηριότητα που ακολουθεί.



**Εικόνα 19: Πρόβλημα που τίθεται στους μαθητές στο μάθημα “Οι Συμβολοσειρές στην Python” στο περιβάλλον του εκπαιδευομένου στο LAMS**

- Η επόμενη δραστηριότητα πρόκειται για δραστηριότητα διερευνήσεων και σχετίζεται με τη δημιουργία νέας συμβολοσειράς αποτελούμενη από μέρη άλλων αλλά και τη χρήση των μεθόδων `replace()`, `upper()`, `lower()` που προσφέρει η Python. Οι μαθητές

μελετούν αρχικά την δραστηριότητα ατομικά απαντώντας στις ερωτήσεις και στην συνέχεια οι απαντήσεις σχολιάζονται στο πλαίσιο διαλόγου μεταξύ καθηγητή και μαθητών.

- Σειρά έχει ένα ακόμη πρόβλημα προς επίλυση ώστε να εφαρμόσουν οι μαθητές τις γνώσεις που έχουν αποκτήσει. Προσπαθούν στην αρχή ατομικά και στην συνέχεια παρουσιάζεται και σχολιάζεται μία από τις λύσεις των μαθητών στην ολομέλεια.
- Έπειτα, οι μαθητές συνεχίζουν με την εκπόνηση δύο ακόμη δραστηριοτήτων διερεύνησης σχετικά με την χρήση του υπαρξιακού τελεστή `in` και τη σύγκριση συμβολοσειρών. Οι δραστηριότητες υποστηρίζονται με τα αντίστοιχα τμήματα θεωρίας ώστε οι μαθητές να μπορούν να τις εκτελέσουν χωρίς την παρέμβαση του καθηγητή. Οι απαντήσεις που δίνονται σχολιάζονται αμέσως μετά στο πλαίσιο διαλόγου μεταξύ καθηγητή και μαθητών.
- Ακολουθεί άλλη μία δραστηριότητα διερευνήσεων σχετική με τη χρήση της μεθόδου `format()` που διαθέτει η Python. Η δραστηριότητα αυτή βρίσκεται σε διακλάδωση μαζί με το αντίστοιχο τμήμα θεωρίας και ο καθηγητής επιλέγει σε ποιους από τους μαθητές θα την αναθέσει καθώς η χρήση και η λειτουργία της είναι λίγο πιο απαιτητική από τις υπόλοιπες.
- Τέλος, οι μαθητές καλούνται να λύσουν ένα πρόβλημα που απαιτεί τη δημιουργία προγράμματος και εκτέλεση από τους ίδιους εφαρμόζοντας όσα έμαθαν για τις συμβολοσειρές. Το πρόβλημα αυτό είναι λίγο πιο απαιτητικό από τα προηγούμενα που έλυσαν και γι αυτό βρίσκεται σε διακλάδωση διαθέτοντας στους μαθητές δύο εκδοχές του προβλήματος αυτού, μία απλή και μία πιο δύσκολη, που υπάρχουν αντίστοιχα στους δύο κλάδους της διακλάδωσης. Ο καθηγητής ανάλογα με το διαθέσιμο χρόνο αποφασίζει αν θα την αναθέσει στους μαθητές να την εκπονήσουν ή όχι. Σε περίπτωση που τους την αναθέσει, καθορίζει ποιοι μαθητές θα εκτελέσουν ποια εκδοχή του προβλήματος ανάλογα των δυνατοτήτων τους. Ο καθηγητής φυσικά βοηθά τους μαθητές καθοδηγώντας τους για να την λύση τους προβλήματος. Όταν οι μαθητές λύσουν το πρόβλημα, κάποιος από τους μαθητές παρουσιάζει την λύση του ώστε να σχολιαστεί από τους υπόλοιπους και τον καθηγητή. Η λύση που αντιστοιχεί στην πιο δύσκολη εκδοχή του προβλήματος παρουσιάζεται και αυτή στη συνέχεια ώστε να την δουν και οι υπόλοιποι μαθητές που έλυσαν την πιο απλή εκδοχή του, μελετώντας τις διαφορές που έχει από το πρόβλημα που μελέτησαν οι ίδιοι.
- Επίσης στην υποστηρικτική δραστηριότητα διατίθενται 2 διαδικτυακοί σύνδεσμοι και ο καθηγητής προτρέπει τους μαθητές να τους επισκεφτούν στο σπίτι. Πρόκειται για ένα επεξηγηματικό βίντεο σχετικά με την αποθήκευση των συμβολοσειρών στην μνήμη και έναν ιστότοπο στον οποίο υπάρχουν όλες οι πληροφορίες σχετικά με τη διαχείριση συμβολοσειρών στην Python.

## 6.5 Οι Λίστες στην Python

Το πέμπτο μάθημα σχετίζεται με τις λίστες και έχει σκοπό την εκμάθηση δημιουργίας, προσπέλασης και διαχείρισης των λιστών στην Python.

### Μαθησιακοί Στόχοι:

Οι μαθησιακοί στόχοι που τίθενται σε αυτό το μάθημα είναι οι μαθητές μετά το πέρας του μαθήματος να μπορούν:

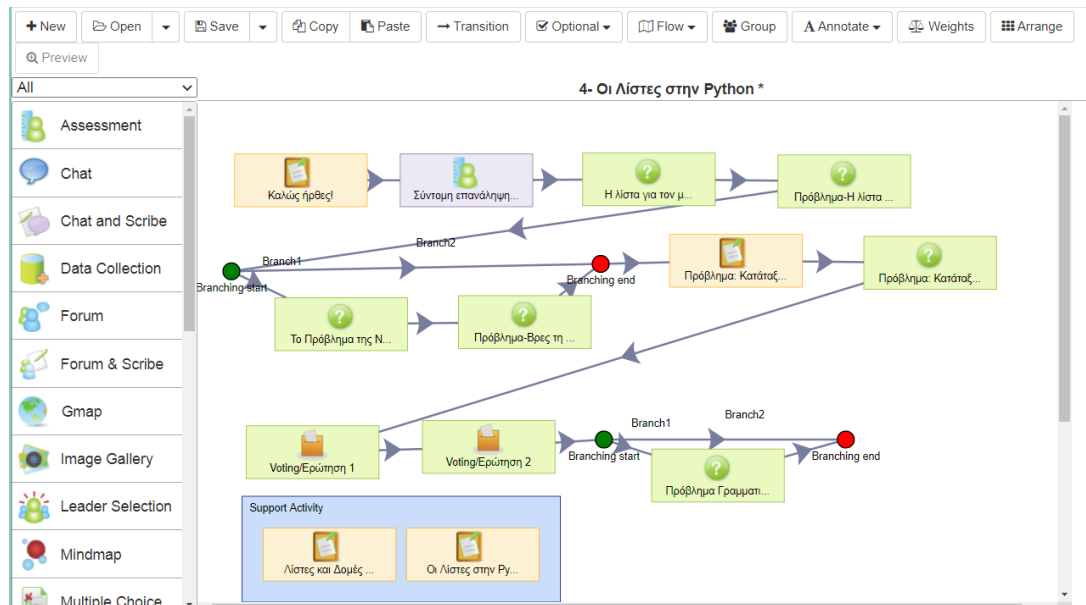
- Να δημιουργούν τις δικές τους λίστες.
- Να έχουν πρόσβαση στις τιμές μιας λίστας.
- Να τροποποιούν τις τιμές που περιέχει μία λίστα.

- Να χρησιμοποιούν συναρτήσεις και μεθόδους που προσφέρει η Python ώστε να υπολογίζουν το μέγεθος μιας λίστας, να προσθέτουν και να αφαιρούν τιμές σε αυτή.

### **Δραστηριότητες και Εργαλεία του LAMS:**

Το μάθημα είναι σχεδιασμένο ενδεικτικά για 3 διδακτικές ώρες του Ομίλου 90 λεπτών. Αποτελείται από μία ακολουθία δραστηριοτήτων, σειριακή στο μεγαλύτερο μέρος της κοινή για όλους τους μαθητές (Εικόνα 20) μαζί με δύο διακλαδώσεις. Πιο συγκεκριμένα αποτελείται από:

- 1 εργαλείο δραστηριότητας “Πίνακας Ανακοινώσεων” στο οποίο παρατίθενται οι στόχοι του μαθήματος.
- 1 εργαλείο “Αξιολόγηση” στο οποίο υπάρχουν ερωτήσεις πολλαπλής επιλογής σχετικά με την ανάθεση τιμών και του τύπους των μεταβλητών, έννοιες που έχουν διδαχθεί σε προηγούμενα μαθήματα.
- 1 εργαλείο δραστηριότητας “Ερώτηση-Απάντηση” στο οποίο υπάρχει μία δραστηριότητα βασισμένη στη μέθοδο των “Διερευνήσεων”.
- 1 εργαλείο δραστηριότητας “Ερώτηση-Απάντηση” στο οποίο παρατίθεται πρόβλημα που απαιτεί τη δημιουργία και εκτέλεση προγράμματος από τους μαθητές για την επίλυσή του.
- 1 εργαλείο δραστηριότητας “Πίνακας Ανακοινώσεων” και ένα “Ερώτηση-Απάντηση” όπου υπάρχει μία δραστηριότητα με τη μέθοδο “Παράδειγμα Αναφοράς”. Στον “Πίνακα Ανακοινώσεων” περιγράφεται το πρόβλημα που πρέπει να λύσουν οι μαθητές και στην “Ερώτηση-Απάντηση” δίνεται το παράδειγμα αναφοράς με βάση το οποίο πρέπει να λύσουν το πρόβλημα και να παραδώσουν την λύση τους στη συνέχεια.
- 2 εργαλεία “Ψηφοφορία” στα οποία τίθενται δύο ερωτήματα στους μαθητές ώστε να ψηφίσουν την απάντηση που πιστεύουν ότι ισχύει.
- 1 εργαλείο “Διακλάδωση” το οποίο περιέχει δύο εργαλεία δραστηριοτήτων “Ερώτηση-Απάντηση”. Στο ένα εργαλείο υπάρχει μια δραστηριότητα με τη μέθοδο των “Διερευνήσεων” και στο άλλο εργαλείο ένα πρόβλημα προς επίλυση το οποίο στηρίζεται στην νέα γνώση που υπάρχει στην προηγούμενη δραστηριότητα διερευνήσεων.
- 1 εργαλείο “Διακλάδωση” το οποίο περιέχει 1 εργαλείο δραστηριότητας “Ερώτηση-Απάντηση”. Στο εργαλείο αυτό τίθεται ένα πρόβλημα αυξημένης δυσκολίας για επίλυση από τους μαθητές και απαιτεί τη δημιουργία και εκτέλεση προγράμματος εφαρμόζοντας τις γνώσεις που πήραν από το συγκεκριμένο μάθημα.
- 1 “Υποστηρικτική” δραστηριότητα η οποία περιλαμβάνει δύο εργαλεία δραστηριοτήτων “Πίνακας Ανακοινώσεων”. Στον έναν πίνακα ανακοινώσεων υπάρχει όλη η θεωρία σχετικά με τις λίστες, έννοιες και εντολές που υπήρχαν στις δραστηριότητες του μαθήματος. Οι πληροφορίες έχουν τη μορφή κειμένου και εικόνων. Στο δεύτερο πίνακα ανακοινώσεων υπάρχουν πληροφορίες σχετικά με τις δομές δεδομένων. Οι πληροφορίες έχουν κυρίως την μορφή κειμένου και παρέχεται και σχετικό βίντεο.



Εικόνα 20: Το μάθημα “Οι Λίστες στην Python” στο LAMS

### Πορεία Μαθήματος:

- Αρχικά ο καθηγητής κάνει μία εισαγωγή στους μαθητές σχετικά με τις λίστες (Εικόνα 21). Στα προηγούμενα μαθήματα διαχειριζόνταν μεταβλητές που περιείχαν μία τιμή, ενώ υπάρχουν δομές, όπως οι λίστες, που περιέχουν παραπάνω από μία τιμές. Αυτή η εισαγωγή γίνεται ώστε ο καθηγητής να κάνει γνωστό το αντικείμενο με το οποίο θα ασχοληθούν στα επόμενα μαθήματα και να διεγείρει το ενδιαφέρον των μαθητών.
- Μάλιστα επειδή η λίστα είναι μια δομή αποθήκευσης τιμών, τίθενται στους μαθητές μερικές ερωτήσεις πολλαπλής επιλογής σχετικά με την ανάθεση τιμών και τους τύπους στους οποίους ανήκουν. Οι ερωτήσεις αυτές γίνονται ώστε ο καθηγητής να διαπιστώσει αν υπάρχουν κενά και να προχωρήσει σε μια σύντομη επανάληψη σχετικά με τις τιμές και τις μεταβλητές ώστε οι μαθητές να μπορέσουν να προχωρήσουν στις επόμενες δραστηριότητες.
- Στην συνέχεια, οι μαθητές καλούνται να εκπονήσουν μια δραστηριότητα βασισμένη στη μέθοδο των διερευνήσεων. Το πρόγραμμα που περιέχεται στη δραστηριότητα περιέχει εντολές σχετικά με τη δημιουργία μιας λίστας, την εισαγωγή, τη διαγραφή, την αντικατάσταση τιμών από αυτή, τον έλεγχο ύπαρξης τιμής στη λίστα, την προσπέλαση της λίστας. Οι μαθητές δουλεύουν αρχικά ατομικά απαντώντας στις ερωτήσεις και στη συνέχεια στο πλαίσιο συζήτησης όλων των μαθητών με τον καθηγητή σχολιάζονται οι απαντήσεις που δόθηκαν. Παράλληλα με το σχολιασμό των απαντήσεων ο καθηγητής με αφορμή τη δραστηριότητα, παρεμβαίνει εξηγώντας με τη μέθοδο ολιγόλεπτης εισήγησης την λειτουργία των συγκεκριμένων νέων εντολών που υπήρχαν στον κώδικα της δραστηριότητας.
- Μετά οι μαθητές καλούνται να εφαρμόσουν τις γνώσεις που μόλις έλαβαν ώστε να δημιουργήσουν και να εκτελέσουν το δικό τους πρόγραμμα με σκοπό να επιλύσουν ένα πρόβλημα (Εικόνα 22). Καθώς προσπαθούν να επιλύσουν το πρόβλημα, ο καθηγητής βοηθά τους μαθητές προσφέροντας βοήθεια και λύνοντας απορίες. Αφού οι μαθητές υποβάλουν τη λύση τους, παρουσιάζεται και σχολιάζεται μία από τις λύσεις των μαθητών.

Καλώς ήρθες!

Preview

Notebook

Support Activities

▼ Λίστες και Δομές δεδομένων

▼ Οι λίστες στην Python

My Progress

Καλώς ήρθες!

Συντάξι επανάληψη στις Τιμές κλειστών μεταβλητών

Η λίστα για τον μονόδρομο

Πρόβλημα με λίστα με τους αριθμούς

Επιτήρηση

Πρόβλημα με κλειστές δομές δεδομένων

Πρόβλημα με κλειστές μεταβλητές

Καλώς ήρθες!

Στα προηγούμενα μαθήματα... έμαθες να ορίζεις και να διαχειρίζεσαι αριθμητικές, αλφαριθμητικές και λογικές μεταβλητές. Οι παραπάνω μεταβλητές περιέχουν μία συγκεκριμένη τιμή.

Ωστόσο υπάρχουν και μεταβλητές που περιέχουν παραπάνω από μια τιμές. Μια λίστα είναι ένας τέτοιος τύπος μεταβλητής!

**Είσαι έτοιμος λοιπόν να μάθεις πως:**

- >να δημιουργείς τις δικές σου λίστες
- >να διαχειρίζεσαι τις τιμές μιας λίστας
- >να τροποποιείς μία λίστα ως προς τις τιμές της
- >να χρησιμοποιείς συναρτήσεις και μεθόδους λιστών μέσα στα προγράμματά σου

Next Activity

**Εικόνα 21: Το εισαγωγικό σημείωμα του μαθήματος “Οι λίστες στην Python” στο LAMS με το εργαλείο “Πίνακας Ανακοινώσεων” στο περιβάλλον του εκπαιδευομένου.**

- Ακολουθεί μία δραστηριότητα διερευνήσεων η οποία παρουσιάζει τη χρήση και τη λειτουργία της συνάρτησης `index()` στις λίστες και στη συνέχεια ένα πρόβλημα προς επίλυση το οποίο απαιτεί τη χρήση της συνάρτησης αυτής για τη λύση του. Η δραστηριότητα διερεύνησης και το πρόβλημα δεν είναι αυξημένης δυσκολίας, ωστόσο βρίσκονται σε μία διακλάδωση ώστε ο καθηγητής να αποφασίσει αν θα τις αναθέσει και σε ποιους μαθητές ανάλογα με τον διαθέσιμο χρόνο και το στάδιο που βρίσκονται οι μαθητές στη σειρά δραστηριοτήτων. Αφού ο καθηγητής καθορίσει ποιοι μαθητές θα ασχοληθούν με τις συγκεκριμένες δραστηριότητες, συζητά μαζί τους τις απαντήσεις που έδωσαν σχολιάζοντας την ορθότητά τους και εξηγώντας την λειτουργία της καινούριας συνάρτησης που έπρεπε να χρησιμοποιήσουν.
- Έπειτα οι μαθητές καλούνται να ασχοληθούν με μία δραστηριότητα βασισμένη στη μέθοδο του “Παραδείγματος Αναφοράς”. Ο καθηγητής παρουσιάζει το πρόβλημα το οποίο οι μαθητές πρέπει να λύσουν. Έπειτα ο καθηγητής παρουσιάζει ένα άλλο πρόβλημα, ως παράδειγμα αναφοράς, και τη λύση του και μέσω ερωτοαπαντήσεων σχολιάζει και αναλύει τα βήματα και τη μέθοδο που οδήγησαν στη λύση του. Στο πρόβλημα που παρουσιάζεται και που καλούνται να λύσουν οι μαθητές, η νέα γνώση την οποία πρέπει να κατανοήσουν και να εφαρμόσουν είναι η χρήση και η διαχείριση λίστας λιστών. Οι μαθητές καλούνται να λύσουν το αρχικό πρόβλημα στηριζόμενοι στο παράδειγμα αναφοράς. Προσπαθούν πρώτα ατομικά με τον καθηγητή να προσφέρει την απαιτούμενη βοήθεια και καθοδήγηση. Στο τέλος ένας από τους μαθητές παρουσιάζει τη λύση που έχει δώσει εξηγώντας για την επιλογή των εντολών που χρησιμοποίησε. Η λύση σχολιάζεται για την ορθότητα της μέσα από το διάλογο που αναπτύσσεται μεταξύ των μαθητών και του καθηγητή, προτείνοντας διορθώσεις ή βελτιώσεις αν χρειάζεται.

Πρόβλημα-Η λίστα με τους αριθμούς

Μελέτησε το πρόβλημα και παράθεσε στη συνέχεια τη λύση σου παρακάτω.

**Question1**

Ο κώδικας που είδες νωρίτερα αποτελεί ένα παράδειγμα διαμόρφωσης μία λίστας με ψώνια για τον μανάβη. Υπήρχαν κάποια προϊόντα στην αρχή, κάποια προστέθηκαν στην συνέχεια, κάποια διαγράφηκαν καθώς και κάποια τροποποιήθηκαν. Προσπάθησε λοιπόν να λύσεις αντίστοιχα το παρακάτω πρόβλημα που περιλαμβάνει μία λίστα με αριθμούς.

Φτιάξε πρόγραμμα το οποίο:

1. Στην αρχική λίστα `numbers_list [20, 14.5, 8.1, -3.9, 67.7]` να **προστίθενται** άλλοι **3 τυχαίοι αριθμοί** οι οποίοι βρίσκονται στο διάστημα `[80,200]`. Επίσης στην ίδια λίστα θα προστίθενται άλλες **δύο τιμές** που θα δίνονται ως **είσοδο από το πληκτρολόγιο** αφού ελεγχθεί ότι δεν υπάρχουν ήδη στη λίστα.
2. Αφού δημιουργηθεί η τελική λίστα, να **εκτυπώνονται** οι τιμές που περιέχει.
3. Στη συνέχεια να ορίζεται μία νέα λίστα `new_list`. Στη λίστα αυτή θα αποθηκεύονται οι τιμές της αρχικής λίστας **αυξημένες όμως κατά τον πρώτο της αριθμό**. Δηλαδή αν `numbers_list [20, 14.5, 8.1, -3.9, 67.7, 199, 90, 4, 7]` τότε όλες οι τιμές θα αυξηθούν κατά 20 και θα προκύψει η `new_list[40, 34.5,28.1, 16.1, 87.7, 219, 110, 24, 27]`.
4. Τέλος πρέπει να **εκτυπώνονται** οι τιμές της καινούριας λίστας.

**Extra:** Να τροποποιεί την `numbers_list[]` αυξάνοντας τις τιμές της κατά την πρώτη της τιμή. Δηλαδή οι νέες τιμές να μην αποθηκεύονται σε νέα λίστα αλλά να τροποποιείται η αρχική λίστα `numbers_list`.

Your answer:

Submit

**Εικόνα 22: Πρόβλημα προς επίλυση στο μάθημα “Οι λίστες στην Python” στο LAMS με το εργαλείο “Ερώτηση-Απάντηση” στο περιβάλλον του εκπαιδευομένου.**

- Για τη συγκεκριμένη δραστηριότητα, για το παράδειγμα αναφοράς δόθηκε ένα συγκεκριμένο πρόγραμμα ως λύση στο οποίο οι μαθητές βασίστηκαν για τη λύση του αρχικού προβλήματος. Η λύση ωστόσο που δόθηκε δεν ήταν η βέλτιστη που θα μπορούσε να δοθεί. Με αφορμή το γεγονός αυτό, ο καθηγητής κάνει δύο ερωτήσεις προς τους μαθητές οι οποίοι πρέπει να ψηφίσουν ποια από τις επιλογές θεωρούν ως βέλτιστη λύση. Βασιζόμενος στις απαντήσεις που δίνουν οι μαθητές, αλλά και με βάση τα προβλήματα τα οποία έλυσαν πάνω στις λίστες, κάνει μία σύντομη εξήγηση σχετικά με τη χρησιμότητα των δομών δεδομένων στον προγραμματισμό.
- Μετά υπάρχει ένα ακόμη πρόβλημα προς επίλυση. Το πρόβλημα αυτό έχει παρουσιαστεί σε προηγούμενο μάθημα συμβολοσειρών αλλά τώρα η λύση του απαιτεί τη δημιουργία και χρήση λίστας λιστών. Για το λόγο αυτό, το πρόβλημα θεωρείται αυξημένης δυσκολίας και βρίσκεται σε διακλάδωση ώστε να επιλέξει ο καθηγητής σε ποιους μαθητές θα το αναθέσει. Οι μαθητές κατά τη διάρκεια επίλυσης του συγκεκριμένου προβλήματος λαμβάνουν βοήθεια και καθοδήγηση από τον καθηγητή.
- Στην υποστηρικτική δραστηριότητα υπάρχουν δύο δραστηριότητες. Η μία περιλαμβάνει όλη τη θεωρία και τη λειτουργία των εντολών σχετικά με τις λίστες που χρειάστηκαν για την εκπόνηση των δραστηριοτήτων και η δεύτερη περιέχει πληροφορίες σε μορφή κειμένου και βίντεο σχετικά με τις δομές δεδομένων στον προγραμματισμό. Ο καθηγητής προτρέπει τους μαθητές μετά το πέρας των μαθημάτων να επισκεφτούν τις συγκεκριμένες δραστηριότητες στο σπίτι ώστε να θυμηθούν όλα όσα ειπώθηκαν σχετικά με τις λίστες αλλά και να εμπλουτίσουν τις γνώσεις τους σχετικά με τις δομές δεδομένων.

## 6.6 Η Δομή Επανάληψης στην Python

Το έκτο μάθημα σχετίζεται με τη δομή επανάληψης. Το συγκεκριμένο μάθημα έχει σκοπό την εκμάθηση και αξιοποίηση των επαναληπτικών δομών `for` και `while` στην Python για την επίλυση προβλημάτων που απαιτούν την χρήση επαναληπτικών δομών.

### Μαθησιακοί Στόχοι:

Οι μαθησιακοί στόχοι που τίθενται σε αυτό το μάθημα είναι οι μαθητές μετά το πέρας του μαθήματος να μπορούν:

- Να αναλύουν τη λειτουργία της επαναληπτικής δομής for και while.
- Να προσδιορίζουν πόσες φορές εκτελείται μία δομή επανάληψης.
- Να εξηγούν πως μεταβάλλονται οι τιμές μεταβλητών σε κάθε διαδοχική επανάληψη.
- Να εντοπίζουν τη συνθήκη τερματισμού μιας επαναληπτικής δομής.

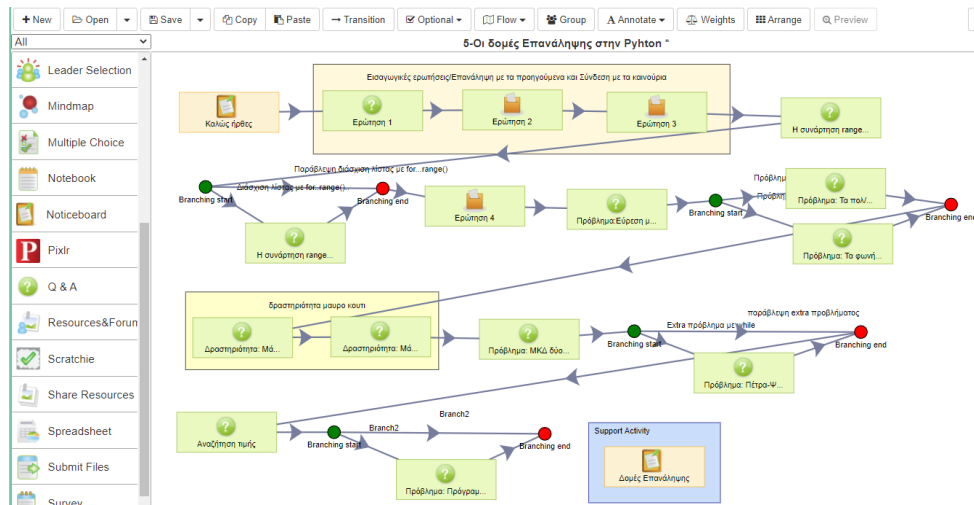
### **Δραστηριότητες και Εργαλεία του LAMS:**

Το μάθημα είναι σχεδιασμένο ενδεικτικά για 4 διδακτικές ώρες του Ομίλου 90 λεπτών. Αποτελείται από μία ακολουθία δραστηριοτήτων, σειριακή με τέσσερις διακλαδώσεις (Εικόνα 23). Πιο συγκεκριμένα αποτελείται από:

- 1 εργαλείο δραστηριότητας “Πίνακας Ανακοινώσεων” στο οποίο παρατίθενται οι στόχοι του μαθήματος.
- 1 εργαλείο “Ερώτηση-Απάντηση” στο οποίο υπάρχει μία επαναληπτική ερώτηση σχετικά με τη χρήση της επαναληπτικής δομής for στην προσπέλαση τιμών που περιέχει μία λίστα.
- 3 εργαλεία “Ψηφοφορία” στις οποίες παρατίθενται 3 ερωτήσεις προς τους μαθητές, μία σε κάθε εργαλείο, ώστε να ψηφίσουν την απάντηση την οποία πιστεύουν ότι ισχύει.
- 2 εργαλεία δραστηριοτήτων “Ερώτηση-Απάντηση” στα οποία υπάρχουν δύο δραστηριότητες βασισμένες στη μέθοδο των “Διερευνήσεων”, μία δραστηριότητα στο εκάστοτε εργαλείο.
- 2 εργαλεία δραστηριοτήτων “Ερώτηση-Απάντηση” στα οποία υπάρχει μία δραστηριότητα βασισμένη στη μέθοδο του “Μαύρου Κουτιού”. Στο πρώτο εργαλείο κατευθύνει τους μαθητές στην εκτέλεση ενός προγράμματος και υπάρχουν ερωτήσεις σχετικές με την εκτέλεση, στο δεύτερο εργαλείο παρουσιάζεται ο κώδικας του προγράμματος και υπάρχουν ερωτήσεις σχετικές με τις εντολές του προγράμματος.
- 2 εργαλεία δραστηριοτήτων “Ερώτηση-Απάντηση” στα οποία παρατίθενται προβλήματα που απαιτούν τη δημιουργία και εκτέλεση προγράμματος από τους μαθητές για την επίλυσή τους.
- 1 εργαλείο “Διακλάδωση” το οποίο περιέχει ένα εργαλείο δραστηριότητας “Ερώτηση-Απάντηση” όπου παρέχεται μία δραστηριότητα με τη μέθοδο των “Διερευνήσεων”. Η διακλάδωση καθώς περιλαμβάνει μια πιο απαιτητική δραστηριότητα έχει ρυθμιστεί ώστε ο καθηγητής να επιλέξει ποιοι μαθητές θα την εκτελέσουν.
- 2 εργαλεία “Διακλάδωση”, σε διαφορετικά σημεία της ακολουθίας δραστηριοτήτων, τα οποία περιέχουν από 1 εργαλείο δραστηριότητας “Ερώτηση-Απάντηση” στο οποίο τίθεται ένα πρόβλημα για επίλυση από τους μαθητές δημιουργώντας και εκτελώντας το δικό τους πρόγραμμα. Η διακλάδωση έχει ρυθμιστεί ώστε ο καθηγητής να επιλέξει ποιοι μαθητές θα εκτελέσουν το εκάστοτε πρόβλημα καθώς είναι αυξημένης δυσκολίας.
- 1 εργαλείο “Διακλάδωση” το οποίο περιέχει 2 δραστηριότητες “Ερώτηση-Απάντηση”. Κάθε μία από αυτές τις δραστηριότητες βρίσκεται αντίστοιχα στα δύο διαφορετικά μονοπάτια που δημιουργεί η διακλάδωση και περιέχουν ένα πρόβλημα προς επίλυση από τους μαθητές. Τα δυο προβλήματα που βρίσκονται στους δύο κλάδους ακολουθούν την ίδια λογική με τη διαφορά ότι το ένα πρόβλημα απαιτεί τη διαχείριση αλφαριθμητικών τιμών ενώ το άλλο πρόβλημα τη διαχείριση αριθμητικών τιμών. Η διακλάδωση έχει ρυθμιστεί ώστε οι μαθητές να μπορούν να επιλέξουν ποιον κλάδο να ακολουθήσουν ανάλογα με αν προτιμούν να χειριστούν αλφαριθμητικές ή αριθμητικές τιμές.
- 1 “Υποστηρικτική” δραστηριότητα η οποία περιλαμβάνει ένα εργαλείο δραστηριότητας “Πίνακας Ανακοινώσεων”. Στον πίνακα ανακοινώσεων υπάρχει όλη η θεωρία σχετικά με τις επαναληπτικές δομές που διδάχτηκαν στο συγκεκριμένο μάθημα, έννοιες και



εντολές που υπήρχαν στις δραστηριότητες του μαθήματος. Οι πληροφορίες έχουν τη μορφή κειμένου.



Εικόνα 23: Το μάθημα “Οι Δομές Επανάληψης στην Python” στο LAMS

### Πορεία Μαθήματος:

- Στα περασμένα μαθήματα, οι μαθητές μέσα από τη διδασκαλία των λιστών στην Python είδαν και τη δομή επανάληψης for. Ωστόσο μέσα από τα μαθήματα αυτά, η δομή της for δεν αναλύθηκε στο έπακρο καθώς είδαν μόνο τη λειτουργία της σχετικά με την προσπέλαση τιμών μιας λίστας. Έτσι λοιπόν, με αφορμή τις γνώσεις που ήδη έχουν αποκτήσει οι μαθητές πάνω στη δομή της for, τους αναθέτει να απαντήσουν σε 2 επαναληπτικές ερωτήσεις, ώστε να τους υπενθυμίσει την λειτουργία την επαναληπτικής δομής for στην διαπέραση μιας λίστας. Αφού γίνει αυτή η σύντομη επανάληψη, ο καθηγητής θέτει στους μαθητές άλλη μία ερώτηση σχετικά με την αναγκαιότητα της χρήσης των δομών επανάληψης στον προγραμματισμό ώστε να τους εισάγει στις νέες έννοιες που θα διδαχθούν και τους καλεί να απαντήσουν ανάλογα με το τι πιστεύουν. Μόλις απαντήσουν στο ερώτημα, ακολουθεί ανταλλαγή απόψεων των μαθητών σχετικά με την απάντηση που έδωσαν και το λόγο που τους οδήγησε σε αυτή. Μέσω της διαδικασίας αυτής, προκαλείται το ενδιαφέρον των μαθητών σχετικά με τη λειτουργία των δομών επανάληψης στον προγραμματισμό και τους βοηθά να κατανοήσουν τη χρήση τους στην επίλυση προβλημάτων.
- Στην συνέχεια οι μαθητές εκπονούν μία δραστηριότητα βασισμένη στη μέθοδο των “Διερευνήσεων” (Εικόνα 24). Τους δίνεται τμήμα κώδικα στο οποίο παρουσιάζεται η χρήση της επαναληπτικής δομής for με τη χρήση της συνάρτησης range(). Οι μαθητές καλούνται αρχικά να μελετήσουν τον κώδικα, να απαντήσουν σε ερωτήσεις και μετά να συγκρίνουν τις απαντήσεις που έδωσαν με το αποτέλεσμα που έφερε η εκτέλεση του προγράμματος. Για την εκτέλεση των προγραμμάτων σε αυτή και στις επόμενες δραστηριότητες, προτείνεται στους μαθητές η εφαρμογή οπτικοποίησης Python Tutor η οποία τους βοηθά να οπτικοποιήσουν τη ροή εκτέλεσης των προγραμμάτων και το περιεχόμενο των μεταβλητών. Οι απαντήσεις των μαθητών σχολιάζονται μέσα από το διάλογο που αναπτύσσεται ανάμεσα στον καθηγητή και στους μαθητές και με αφορμή τις απαντήσεις και τις απορίες που πιθανόν να έχουν προκύψει, ο καθηγητής μέσω ολιγόλεπτης εισήγησης αναλύει τη μορφή και τη ροή εκτέλεσης της επαναληπτικής δομής for.

Η συνάρτηση range( A, M, B ) επιστρέφει μια λίστα αριθμών ξεκινώντας με τον αριθμό A μέχρι το M με βήμα B. Το M δεν συμπεριλαμβάνεται στη λίστα. Αν το B =1 ή αν A=0 μπορούν να παραληφθούν σαν πεδία.

Παρακάτω υπάρχουν τα εξής 2 προγράμματα:

**Πρόγραμμα 1**

```
1. for item in [1,2,3,4,5,6,7,8] :
2.   print (item)
3.   print("Hi!")
```

**Πρόγραμμα 2**

```
1. for item in ..... :
2.   print (item)
3.   print("Hi!")
```

>Το πρόγραμμα 1 εκτυπώνει τους αριθμούς 1,2,3,4,5,6,7,8 οι οποίοι αποτελούν τιμές μιας λίστας όπως γνωρίζεις από προηγούμενο μάθημα. Συμπλήρωσε κατάλληλα το πρόγραμμα 2 ώστε να εκτυπώνονται οι ίδιοι αριθμοί χρησιμοποιώντας την συνάρτηση **range()**.

>Πόσες φορές πιστεύεις ότι εκτελούνται οι εντολές 2 και 3 στο πρόγραμμα 1 και πόσες στο πρόγραμμα 2;

Εκτέλεσε τα δύο προγράμματα στο περιβάλλον οπτικοποίησης <https://pythontutor.com/visualize.html#mode=edit>

>Τελικά πόσες φορές εκτελούνται; Γιατί συμβαίνει αυτό;

Έστω οι εξής εντολές:

```
1. for i in range(12,23):
2.   print("Hello!")
```

>Ποιο πιστεύεις είναι το αποτέλεσμα της εκτέλεσης των εντολών και πόσες φορές θα εκτελεστεί η εντολή 2;

Εκτέλεσε τις εντολές στην εφαρμογή οπτικοποίησης.

>Τελικά ποιο ήταν το αποτέλεσμα και πόσες φορές εκτελέστηκε η εντολή 2, γιατί;

**Εικόνα 24: Δραστηριότητα διερευνήσεων στο μάθημα “Οι Δομές Επανάληψης στην Python” στο LAMS**

- Ακολουθεί άλλη μία δραστηριότητα βασισμένη στη μέθοδο των “Διερευνήσεων” η οποία βρίσκεται σε διακλάδωση. Στη δραστηριότητα παρουσιάζεται ένας τρόπος προσπέλασης μιας λίστας και προσβασης των τιμών που περιλαμβάνει με χρήση της επαναληπτικής δομής for και της συνάρτησης range(). Ο καθηγητής επιλέγει σε ποιους μαθητές θα αναθέσει τη δραστηριότητα ανάλογα των δυνατοτήτων τους, καθώς αποτελεί μία δραστηριότητα πιο απαιτητική σε σχέση με την προηγούμενη.
- Ακολουθεί άλλη μία ερώτηση προς τους μαθητές όπου τους παρουσιάζεται μία κατάσταση προς επίλυση και καλούνται να απαντήσουν ανάλογα με το τι πιστεύουν ότι είναι καλύτερο για τη λύση της. Μόλις απαντήσουν, ακολουθεί ανταλλαγή απόψεων από τους μαθητές σχετικά με τις απαντήσεις που έδωσαν αιτιολογώντας για την επιλογή τους. Η ερώτηση έχει το ρόλο να προβληματίσει τους μαθητές και να τους ενεργοποιήσει καθώς η επόμενη δραστηριότητα που ακολουθεί απαιτεί τη λύση του συγκεκριμένου προβλήματος.
- Έτσι λοιπόν οι μαθητές μετά από τη συζήτηση που είχαν μαζί με τον καθηγητή σχετικά με το συγκεκριμένο πρόβλημα, καλούνται να εφαρμόσουν τη γνώση που έλαβαν προηγουμένως για τη δομή επανάληψης for και να επιλύσουν το πρόβλημα δημιουργώντας και εκτελώντας το πρόγραμμά τους. Ο καθηγητής κατά τη διάρκεια της επίλυσης του προβλήματος από τους μαθητές προσφέρει καθοδήγηση και βοήθεια σε αυτούς.
- Ακολουθεί άλλο ένα πρόβλημα προς επίλυση το οποίο απαιτεί τη χρήση της δομής for. Στους μαθητές παρέχονται δύο παρόμοια προβλήματα με τη διαφορά ότι το ένα πρόβλημα σχετίζεται με διαχείριση αριθμητικών τιμών και το άλλο με διαχείριση αλφαριθμητικών τιμών. Ο καθηγητής αφού τους παρουσιάσει τα δύο προβλήματα και τονίσει τη διαφορά που έχουν, τους αφήνει να επιλέξουν το πρόβλημα που προτιμούν να λύσουν. Ο καθηγητής βοηθά και καθοδηγεί τους μαθητές για την επίλυση του

προβλήματος. Αφού τελειώσουν, αναθέτει σε δύο μαθητές οι οποίοι ασχολήθηκαν με διαφορετικό πρόβλημα ο καθένας, να παρουσιάσουν τη λύση τους και να αιτιολογήσουν γι αυτή ώστε να την δουν και οι υπόλοιποι μαθητές που δεν μελέτησαν το συγκεκριμένο πρόβλημα.

- Στη συνέχεια οι μαθητές καλούνται να εκπονήσουν άλλη μία διερευνητική δραστηριότητα η οποία βασίζεται στη μέθοδο του “Μαύρου Κουτιού” και το πρόβλημα που παρουσιάζεται έχει λυθεί με την επαναληπτική δομή while. Για τη δραστηριότητα αυτή χρησιμοποιείται η εφαρμογή Your Python Trinket η οποία προσφέρει την δυνατότητα εκτέλεσης ενός προγράμματος από έναν χρήστη χωρίς να έχει πρόσβαση στον κώδικα. Έτσι λοιπόν οι μαθητές κατευθύνονται μέσω συνδέσμου που τους έχει δοθεί στο περιβάλλον της εφαρμογής ώστε να εκτελέσουν ένα πρόγραμμα και με βάση το αποτέλεσμα της εκτέλεσης και της αλληλεπίδρασης που είχαν με το πρόγραμμα να απαντήσουν σε ερωτήσεις. Στη συνέχεια τους δίνεται και ο κώδικας του προγράμματος και μελετώντας τον σε συνδυασμό με την εκτέλεση που είδαν προηγουμένως καλούνται να απαντήσουν σε νέες ερωτήσεις. Οι απαντήσεις των μαθητών σχολιάζονται από τους ίδιους και τον καθηγητή και μέσα από τη συζήτηση για τη συγκεκριμένη δραστηριότητα ο καθηγητής αναλύει τη μορφή και τη λειτουργία της επαναληπτικής δομής while.
- Στη συνέχεια οι μαθητές πρέπει να εφαρμόσουν τις γνώσεις που μόλις έλαβαν σχετικά με την καινούρια δομή επανάληψης που διδάχθηκαν και να επιλύσουν ένα ακόμη πρόβλημα που απαιτεί χρήση της while δημιουργώντας και εκτελώντας το δικό τους πρόγραμμα (Εικόνα 25). Ο καθηγητής καθόλη τη διάρκεια επίλυσης του προβλήματος, καθοδηγεί τους μαθητές δίνοντας συμβουλές και λύνοντας τυχόν απορίες.
- Ακολουθεί ένα ακόμη πρόβλημα προς επίλυση, λίγο πιο απαιτητικό. Ο καθηγητής αποφασίζει σε ποιους θα το αναθέσει ανάλογα των δυνατοτήτων τους και ανάλογα το στάδιο στο οποίο βρίσκονται στην ακολουθία δραστηριοτήτων.
- Έπειτα υπάρχει μία δραστηριότητα σχεδιασμένη με τη μέθοδο των “Διερευνήσεων” σχετική με τη μέθοδο αναζήτησης τιμής καθώς και πώς ένα πρόβλημα που λύνεται με την επαναληπτική δομή for μπορεί να λυθεί και με την επαναληπτική δομή while(). Οι μαθητές προσπαθούν να απαντήσουν στις ερωτήσεις μελετώντας αρχικά τον κώδικα του προγράμματος και στη συνέχεια επαληθεύουν την ορθότητα των απαντήσεών τους με βάση την εκτέλεση του προγράμματος. Όταν υποβάλουν τις απαντήσεις τους, αυτές σχολιάζονται στο πλαίσιο συζήτησης που αναπτύσσεται μεταξύ των μαθητών και του καθηγητή.
- Μετά τη διερευνητική δραστηριότητα υπάρχει ένα πρόβλημα το οποίο η λύση του απαιτεί την εφαρμογή της μεθόδου αναζήτησης τιμής που αναλύθηκε στην προηγούμενη δραστηριότητα. Το πρόβλημα θεωρείται αυξημένης δυσκολίας γι’ αυτό ο καθηγητής καθορίζει σε ποιους μαθητές θα την αναθέσει.
- Τέλος, στην υποστηρικτική δραστηριότητα παρέχεται όλη η θεωρία σχετικά με τη σύνταξη και τη λειτουργία των δομών επανάληψης for και while. Ο καθηγητής προτρέπει τους μαθητές να την επισκεφτούν όποτε το επιθυμούν για να τη μελετήσουν και να ξαναθυμηθούν έννοιες που διδάχτηκαν στο μαθημα.

Πρόβλημα: ΜΚΔ δύο αριθμών

Προσπάθησε να λύσεις το παραπάνω πρόβλημα και παράθεσε την λύση σου παρακάτω.

**Question 1**

Ο Ευκλείδης, στο έβδομο βιβλίο των Στοιχείων του, περιγράφει μια μέθοδο για την εύρεση του μέγιστου κοινού διαιρέτη δύο ακεραίων. Μπορεί κανείς να βρει πολλές διαφορετικές εκδοχές του αλγορίθμου, αλλά βασικά ο αλγόριθμος μπορεί να συνοψιστεί ως εξής:

Αφαιρέσε τον μικρότερο αριθμό από τον μεγαλύτερο  
και επανάλαβε τη διαδικασία μέχρι οι δύο αριθμοί να γίνουν ίσοι.  
Η τιμή των δύο αριθμών όταν τελειώσει η διαδικασία είναι  
ο μέγιστος κοινός διαιρέτης τους.

Γράψτε ένα πρόγραμμα το οποίο ζητάει από το χρήστη δύο ακεραίους αριθμούς και υπολογίζει τον μέγιστο κοινό διαιρέτη τους, χρησιμοποιώντας τη μέθοδο του Ευκλείδη.

✦ Τσέκαρε την ορθότητα της λύσης σου χρησιμοποιώντας τα εξής ζευγάρια αριθμών:

- 6,35 -> ΜΚΔ=1
- 105,252 -> ΜΚΔ=21
- 90,25 -> ΜΚΔ=5

Your answer:

**Εικόνα 25: Πρόβλημα προς επίλυση στο μάθημα “Οι Δομές Επανάληψης στην Python” στο LAMS στο περιβάλλον εκπαιδευομένου.**

## ΣΥΜΠΕΡΑΣΜΑΤΑ

Τα αποτελέσματα αξιοποίησης του συστήματος εκμάθησης LAMS στη δημιουργία μαθημάτων για την υποστήριξη του ομίλου Πληροφορικής του 2ου Γενικού Πρότυπου Λυκείου Αθηνών ήταν πολύ θετικά και από την πλευρά των εκπαιδευομένων και από την πλευρά του εκπαιδευτικού.

Από την πλευρά των εκπαιδευομένων, οι μαθητές έδειξαν να εξοικειώνονται εύκολα με το περιβάλλον, χωρίς να αντιμετωπίζουν ιδιαίτερες δυσκολίες στην εκπόνηση των δραστηριοτήτων. Σε κάθε μάθημα που είχε σχεδιαστεί στο LAMS υπήρχε συγκεντρωμένο όλο το εκπαιδευτικό υλικό, αρκετά βοηθητικό για τους μαθητές καθώς μπορούσαν εύκολα να ανατρέξουν στη θεωρία του μαθήματος ή σε προηγούμενες δραστηριότητες χωρίς να χρειάζεται να ανοίγουν επιπρόσθετα αρχεία εκτός του συγκεκριμένου περιβάλλοντος. Επίσης είχαν άμεση πρόσβαση σε εκπαιδευτικά βίντεο και διαδικτυακούς ιστοτόπους για επιπλέον προσωπική μελέτη.

Η χρήση του περιβάλλοντος απαιτεί ευχέρεια στη χρήση του υπολογιστή και ενδεχομένως αυξάνει το χρόνο που χρειάζονται για να υποβάλλουν την απάντησή τους (μερικοί μαθητές χρειάζονταν παραπάνω χρόνο για να πληκτρολογήσουν και να υποβάλλουν τις απαντήσεις τους στις διερευνητικές δραστηριότητες και στα προβλήματα). Όμως είναι σημαντικό οι μαθητές να έχουν το χρόνο να σκεφτούν και να δομήσουν τη σκέψη τους σχετικά με τις απαντήσεις τους. Ένας καθηγητής άλλωστε επιβλέπει τη διαδικασία της μάθησης και μπορεί να παρέμβει σε όποιο σημείο θεωρεί απαραίτητο προσφέροντας υποστήριξη και βοήθεια τους μαθητές. Βέβαια στη συγκεκριμένη περίπτωση το LAMS αποτελεί ένα πολύ χρήσιμο εργαλείο για τον εκπαιδευτικό καθώς προσφέρει τη δυνατότητα δημιουργίας διαφορετικών μονοπατιών μάθησης. Πιο απαιτητικές δραστηριότητες βρίσκονταν σε διακλαδώσεις και αναθέτονταν σε μαθητές που προχωρούσαν πιο γρήγορα στην ακολουθία δραστηριοτήτων σε σχέση με άλλους. Με αυτόν τον τρόπο, οι μαθητές είχαν τη δυνατότητα να εκπονοούν δραστηριότητες ανάλογες των δυνατοτήτων τους αλλά και να μην αποκλίνουν σε μεγάλο βαθμό ο ένας από τον άλλον στη μαθησιακή πορεία.

Από την πλευρά του εκπαιδευτικού, ένας εκπαιδευτικός στο περιβάλλον του LAMS έχει τη δυνατότητα δημιουργίας ολοκληρωμένων σχεδίων μαθημάτων αξιοποιώντας πολλά από τα εργαλεία για ανάθεση επαναληπτικών ασκήσεων, διερευνητικών δραστηριοτήτων, προβλημάτων προς επίλυση, ερωτήσεων αξιολόγησης και παράθεση διαδικτυακών πόρων ενισχύοντας τη μάθηση των εκπαιδευομένων. Επίσης ο εκπαιδευτικός έχει άμεση πρόσβαση στις απαντήσεις των εκπαιδευομένων και μπορεί να ελέγχει την πορεία κάθε εκπαιδευόμενου στην ακολουθία δραστηριοτήτων παρεμβαίνοντας και προσφέροντας βοήθεια όπου χρειάζεται. Το τελευταίο μάλιστα φάνηκε εξαιρετικά βοηθητικό στην διεξαγωγή των εξ αποστάσεων μαθημάτων καθώς στη συγκεκριμένη περίπτωση ο καθηγητής έχει μεγαλύτερη ανάγκη για έλεγχο της πορείας των εκπαιδευομένων του.

Αν και το αποτέλεσμα από την αξιοποίηση του LAMS για τη δημιουργία μαθημάτων εκμάθησης βασικών στοιχείων της Python στο πλαίσιο υποστήριξης του ομίλου Πληροφορικής είναι αρκετά θετικό, σίγουρα τα σχέδια μαθημάτων επιδέχονται βελτιώσεις για το μέλλον. Είναι άξιο μελέτης η αξιοποίηση του LAMS στη δια ζώσης μάθηση καθώς και η δημιουργία συνεργατικών δραστηριοτήτων που προτρέπουν τους εκπαιδευόμενους για παραπάνω αλληλεπίδραση και επικοινωνία μεταξύ τους.

## ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

<b>Ξενογλωσσος όρος</b>	<b>Ελληνικός Όρος</b>
World Wide Web	Παγκόσμιος Ιστός
e-learning	Ηλεκτρονική μάθηση
Blended learning	Μεικτή μάθηση
Lesson plan	Σενάριο μαθήματος
Learning design	Σχεδιασμός μάθησης
Template	Υπόδειγμα
Differentiated learning	Διαφοροποιημένη μάθηση
Data Science	Επιστήμη Δεδομένων
Artificial Intelligence	Τεχνητή Νοημοσύνη
Machine Learning	Μηχανική Μάθηση
Forum	Συζήτηση
Chat	Συνομιλία

## ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ

LAMS	Learning Activities Management System
LMS	Learning Management System
ΣΔΜ	Σύστημα Διαχείρισης Μάθησης
MALCOE	Macquarie E-learning Centre Of Excellence
ΕΚΠΑ	Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών
ΕΜΠ	Εθνικό Μετσόβιο Πολυτεχνείο

## ΠΑΡΑΡΤΗΜΑ

### Μάθημα 1: Εισαγωγή στον προγραμματισμό με Python

#### Εισαγωγή

Ο υπολογιστής είναι μια μηχανή που επεξεργάζεται δεδομένα και μπορεί να επιλύσει διάφορα υπολογιστικά προβλήματα. Ωστόσο, για να επιτελέσει τις παραπάνω λειτουργίες σωστά πρέπει εμείς να του δώσουμε σαφείς εντολές με τη μορφή προγραμμάτων. Ουσιαστικά ένα πρόγραμμα αποτελεί το μέσο μεταξύ του προγραμματιστή και του υπολογιστή. Ή αλλιώς θα μπορούσαμε να πούμε ότι ένας υπολογιστής χρειάζεται ένα πρόγραμμα κατανοητό σε αυτόν ώστε να εκτελέσει μία λειτουργία. Η διαδικασία λοιπόν, **συγγραφής προγραμμάτων ονομάζεται προγραμματισμός και περιλαμβάνει τη διατύπωση των κατάλληλων εντολών προς τον υπολογιστή με τη χρήση τεχνητών γλωσσών, δηλαδή των γλωσσών προγραμματισμού.**



Αυτά και άλλα πολλά θα προσπαθήσουμε να αναλύσουμε στη συνέχεια, προσεγγίζοντας όσο γίνεται καλύτερα το θέμα του προγραμματισμού και απαντώντας τελικά στα ερωτήματα:

- Τι είναι ένα υπολογιστικό πρόβλημα;
- Τι ορίζεται ως αλγόριθμος;
- Τι είναι μία γλώσσα προγραμματισμού;

#### Προσδοκώμενα αποτελέσματα:

Όταν ολοκληρώσεις το μάθημα αυτό θα είσαι ικανός να:

- εξηγείς το ρόλο του υπολογιστή στην επίλυση προβλημάτων
- διατυπώνεις τι είναι ένα πρόβλημα
- ορίζεις τι είναι αλγόριθμος
- επεξηγείς πως ένα πρόγραμμα σχετίζεται με τις έννοιες του αλγορίθμου και την επίλυση προβλήματος από τον υπολογιστή
- περιγράφεις τι είναι μια γλώσσα προγραμματισμού και να εξηγείς ποιος είναι ο ρόλος της μεταξύ των προγραμματιστών και του υπολογιστή
- να εξηγείς γιατί η Python θεωρείται γλώσσα γενικού σκοπού
- να αναφέρεις 3 εφαρμογές στις οποίες μπορείς να χρησιμοποιήσεις την Python για την υλοποίησή τους
- να εξηγείς γιατί Python είναι μια διερμηνευόμενη γλώσσα προγραμματισμού και ποια είναι η διαφορά με μια μη διερμηνευόμενη γλώσσα προγραμματισμού
- να γράψεις και να εκτελέσεις το πρώτο σου πρόγραμμα Python στο IDLE

#### Τι είναι πρόβλημα;

Καθημερινά λέμε ότι έχουμε να αντιμετωπίσουμε ή να επιλύσουμε διάφορα προβλήματα. Τι είναι όμως ένα πρόβλημα και πως μπορούμε να ορίσουμε ένα πρόβλημα; **Ως πρόβλημα θεωρούμε κάθε ζήτημα που τίθεται προς επίλυση ή αλλιώς κάθε κατάσταση που μας απασχολεί**





**και πρέπει να αντιμετωπιστεί. Σε κάποια προβλήματα βρίσκουμε εύκολα την λύση, σε κάποια άλλα δυσκολευόμαστε επειδή είναι πιο σύνθετα και σε άλλα δεν μπορούμε να διατυπώσουμε καθόλου μια πιθανή λύση.**

Η αλήθεια είναι ότι καθημερινά χρησιμοποιούμε τον υπολογιστή (επιτραπέζιο, φορητό, ταμπλέτα, έξυπνο κινητό, έξυπνη τηλεόραση, κ.ο.κ.) για να εργαστούμε, να επικοινωνήσουμε με άλλους, να ενημερωθούμε, να ψυχαγωγηθούμε κ.α. Για κάθε δραστηριότητα μας στον υπολογιστή λοιπόν, αυτός συμβάλει στην επίλυση διάφορων επιμέρους προβλημάτων.

### **Η συμβολή του υπολογιστή στην επίλυση προβλημάτων:**

Γιατί όμως η χρήση υπολογιστή για επίλυση προβλημάτων είναι τόσο διαδεδομένη αλλά και απαραίτητη; Η αλήθεια είναι ο υπολογιστής σε ορισμένες περιπτώσεις υπερβαίνει τις δυνατότητες του ανθρώπου. Ένας υπολογιστής προσφέρει μεγάλο αποθηκευτικό χώρο δεδομένων (εικόνες, κείμενα, αριθμοί, βίντεο) και άρα μπορεί να επεξεργαστεί μεγάλο πλήθος δεδομένων και ζητούμενων, την επεξεργασία αυτή την κάνει με αρκετά μεγαλύτερη ταχύτητα από ότι ο άνθρωπος και επίσης μπορεί να εκτελέσει πολλές φορές κατά επανάληψη μια συγκεκριμένη διαδικασία, ή αλλιώς μια σειρά εντολών. Έτσι λοιπόν ο υπολογιστής είναι αναγκαίος όταν θέλουμε να επιλύσουμε προβλήματα που περιέχουν τις παραπάνω απαιτήσεις.



Για να επιλύσει ο υπολογιστής λοιπόν ένα πρόβλημα, όπως αναφέρθηκε και παραπάνω, χρειάζεται συγκεκριμένες εντολές και οι εντολές αυτές έχουν την μορφή ενός προγράμματος. Παράλληλα, κατά την εκτέλεση των προγραμμάτων, μπορούμε ως χρήστες να εισάγουμε διάφορα δεδομένα μέσω συσκευών εισόδου, όπως είναι το πληκτρολόγιο, αλλά και να λάβουμε απαντήσεις σχετικά με τη λύση του προβλήματος μέσω των συσκευών εξόδου, όπως είναι η οθόνη.

### **Στάδια επίλυσης ενός προβλήματος:**

Ακόμη και στην περίπτωση που ο υπολογιστής λύνει το πρόβλημα, ο άνθρωπος είναι αυτός που θα πρέπει να κατασκευάσει κατάλληλα προγράμματα ώστε να τα εκτελέσει ο υπολογιστής. Έτσι λοιπόν σε κάθε περίπτωση για να επιλύσουμε με αποτελεσματικότητα ένα δύσκολο και σύνθετο πρόβλημα, με ή χωρίς τη βοήθεια υπολογιστή πρέπει:

- να διακρίνουμε τα δεδομένα και τα ζητούμενα που απαιτούνται
- να το αναλύουμε σε επιμέρους απλούστερα υπό-προβλήματα που επιλύονται ευκολότερα
- να εκτελούμε οργανωμένα βήματα επίλυσης

Μπορείς να δεις το παρακάτω βίντεο εισάγοντας αγγλικούς υπότιτλους από τις ρυθμίσεις:

[https://www.youtube.com/embed/kM9ASKAni\\_s](https://www.youtube.com/embed/kM9ASKAni_s)

### **Τι είναι αλγόριθμος:**

Πολλές φορές μαζί με τις έννοιες πρόγραμμα και προγραμματισμός συναντάμε και την έννοια του αλγορίθμου. Τι είναι λοιπόν ένας αλγόριθμος και πότε λέμε ότι ακολουθούμε έναν αλγόριθμο;

Ως αλγόριθμος ορίζεται ως μια πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, που στοχεύουν στην επίλυση ενός προβλήματος. Πιο απλά (αλγόριθμο) ονομάζουμε μία σειρά από εντολές που έχουν αρχή και τέλος, είναι σαφείς και έχουν ως σκοπό την επίλυση κάποιου προβλήματος.

Η αλήθεια είναι ότι αλγορίθμους ακολουθούμε αρκετά συχνά στην καθημερινότητά μας. Ένα παράδειγμα είναι η εκτέλεση μιας μαγειρικής συνταγής ή η επίλυση ενός μαθηματικού προβλήματος.

Η διατύπωση ενός αλγορίθμου μπορεί να γίνει:

- με χρήση της φυσικής γλώσσας
- με διαγραμματικές τεχνικές
- ή με κωδικοποίηση σε ειδική (τεχνητή) γλώσσα.

Ένα πρόγραμμα αποτελεί έναν αλγόριθμο γραμμένο σε γλώσσα κατανοητή για τον υπολογιστή και περιέχει εντολές (οδηγίες) που κατευθύνουν με κάθε λεπτομέρεια τον υπολογιστή, για να εκτελέσει μια συγκεκριμένη εργασία και να επιλύσει τελικά ένα πρόβλημα. Οι υπολογιστές δεν μπορούν να κάνουν υποθέσεις ή να μαντέψουν τι θέλει ο άνθρωπος. Αν πείτε στον υπολογιστή: τηγάνισε τις πατάτες το πιο πιθανό είναι να πάρει τις πατάτες ολόκληρες και να τις τηγανίσει έτσι όπως είναι. Ο υπολογιστής εκτελεί τις εντολές που του δίνουμε κυριολεκτικά γι αυτό πρέπει να σκεφτόμαστε πιθανά **λογικά λάθη** στις οδηγίες μας! Λάθη δηλαδή που προκύπτουν από δική μας απερισκεψία και φέρουν ένα μη επιθυμητό αποτέλεσμα.

Κάντε ένα πείραμα:

- Γράψτε τα βήματα για να φτιάξετε ένα φλιτζάνι καφέ ή τσάι.
- Γράψτε όλα όσα χρειάζεται να γίνουν. Αν θέλετε γάλα ή κρέμα καθώς και που μπορεί κάποιος να το βρει.
- Μόλις τελειώσετε δοκιμάστε να δώσετε τις οδηγίες σε έναν φίλο σας και πείτε του να τις εκτελέσει.



**Μήπως σου ξέφυγε κάτι;**

Οι υπολογιστές λοιπόν μπορούν να εκτελούν δισεκατομμύρια λειτουργίες ανά δευτερόλεπτο ωστόσο η δουλειά του προγραμματιστή είναι να βρίσκει λύσεις και αυτό το κάνει αναλύοντας ένα πρόβλημα σε καθορισμένα βήματα που μπορεί να εκτελέσει ένας υπολογιστής.

Μπορείς να δεις το παρακάτω βίντεο εισάγοντας αγγλικούς ή ελληνικούς υπότιτλους από τις ρυθμίσεις.

<https://www.youtube.com/embed/6hfOvs8pY1k>

**Τι είναι γλώσσα προγραμματισμού:**

Μια γλώσσα προγραμματισμού είναι μια τεχνητή γλώσσα που χρησιμοποιεί ο προγραμματιστής για να γράψει τα προγράμματα. Μία φυσική γλώσσα περιέχει συγκεκριμένο συντακτικό σύμφωνα με το οποίο εμείς οι άνθρωποι δημιουργούμε τις προτάσεις και μπορούμε να επικοινωνήσουμε. Το ίδιο συμβαίνει και στις



γλώσσες προγραμματισμού. Οι εντολές που διαθέτει μια γλώσσα προγραμματισμού πρέπει να συντάσσονται σύμφωνα με συγκεκριμένους κανόνες από εμάς τους προγραμματιστές. Αν για κάποιο λόγο συντάξουμε λανθασμένα μία εντολή τότε το πρόγραμμά μας περιέχει λάθος, συγκεκριμένα ένα **συντακτικό** λάθος, και δεν μπορεί να εκτελεστεί μέχρι να διορθωθεί.



Τα προγράμματά μας, και οι εντολές με τις οποίες συντάσσονται, πρέπει να είναι κατανοητά και από εμάς τους προγραμματιστές που τα δημιουργούμε αλλά και από τον υπολογιστή που τα εκτελεί. Ο υπολογιστής καταλαβαίνει την γλώσσα μηχανής, ακολουθίες δηλαδή από 0 και 1. Οι άνθρωποι από την άλλη καταλαβαίνουμε και επικοινωνούμε με την φυσική γλώσσα. Στα πρώτα χρόνια του προγραμματισμού λοιπόν υπήρχαν οι γλώσσες χαμηλού επιπέδου, γλώσσες προγραμματισμού δηλαδή που ήταν πολύ κοντά στην γλώσσα μηχανής. Μια τέτοια γλώσσα περιείχε εντολές που δύσκολα μάθαινε και τις θυμόταν ένας προγραμματιστής. Με το πέρασμα του χρόνου λοιπόν καταλήξαμε στις γλώσσες προγραμματισμού υψηλού επιπέδου που περιέχουν εντολές και συμβολισμούς κατανοητούς από τους ανθρώπους που μπορούν να μάθουν πολύ πιο εύκολα και γρήγορα. Μια γλώσσα προγραμματισμού υψηλού επιπέδου μοιάζει με τη φυσική γλώσσα, περιέχει δικό της συντακτικό, αλφάβητο και λεξιλόγιο και περιέχει ειδικό πρόγραμμα ώστε να το μετατρέψει με γλώσσα κατανοητή από τον υπολογιστή, δηλαδή σε γλώσσα μηχανής.

Μερικές από τις πιο δημοφιλείς γλώσσες είναι η **C**, **C++**, **Java**, **PHP**, **Javascript**, **Basic**, **Ruby** και φυσικά η **Python**. Ανάλογα με το είδος του προγράμματος που θέλουμε να αναπτύξουμε (για γενική χρήση, για επιστημονική χρήση, για χρήση στον Παγκόσμιο Ιστό, κ.λπ.) επιλέγουμε και την κατάλληλη γλώσσα προγραμματισμού.

Μπορείς να δεις το παρακάτω βίντεο βάζοντας αγγλικούς υπότιτλους από τις ρυθμίσεις:

<https://www.youtube.com/embed/XASY30EfGAc>

## Εισαγωγή στην Python

Η **Python** είναι μία γλώσσα προγραμματισμού υψηλού επιπέδου, γενικού σκοπού, **διερμηνευόμενη και είναι κατάλληλη και για αρχάριους και για έμπειρους προγραμματιστές**. Περιέχει εντολές εύκολες ως προς τη συγγραφή σε σχέση με άλλες γλώσσες προγραμματισμού που γίνονται εύκολα κατανοητές. Επιπλέον τα προγράμματα σε αυτή την γλώσσα μπορούν να τρέξουν σε διάφορα λειτουργικά συστήματα όπως Windows, Mac, Linux ακόμα και σε iOS/Android συσκευές. Διαθέτει πληθώρα έτοιμων βιβλιοθηκών που μπορούν να χρησιμοποιηθούν εύκολα και άμεσα. Τα προγράμματα σε Python είναι συμπαγή, ευανάγνωστα, και γράφονται και συντηρούνται γρηγορότερα σε σχέση με άλλες δημοφιλείς γλώσσες προγραμματισμού όπως οι C, C++ και Java.



Στη συνέχεια του μαθήματος λοιπόν θα αναλύσουμε γιατί η Python θεωρείται μία γλώσσα γενικού σκοπού και λέμε ότι είναι διερμηνευόμενη, πώς κάποιος μπορεί

να εγκαταστήσει την Python στον υπολογιστή του καθώς και πώς μπορεί να εκτελέσει εύκολα τα προγράμματά του στο περιβάλλον IDLE.

### Πως ορίζουμε ότι μια γλώσσα προγραμματισμού είναι γενικού σκοπού;

Μια γλώσσα προγραμματισμού γενικού σκοπού είναι μια γλώσσα σχεδιασμένη ώστε να χρησιμοποιείται για την υλοποίηση εφαρμογών διαφόρων τομέων. Μερικές από αυτές είναι η C, C++, C#, Java, Ruby, η Python φυσικά και πολλές άλλες. Η Python συγκεκριμένα μπορεί να χρησιμοποιηθεί για ανάπτυξη εφαρμογών που αφορούν τη διαχείριση συστήματος υπολογιστή, την ανάπτυξη εφαρμογών Διαδικτύου, την επεξεργασία αρχείων κειμένου, διάφορες επιστημονικές εφαρμογές, την εκπαίδευση, την ανάπτυξη παιχνιδιών, τη μουσική βιομηχανία (Spotify), τις πλατφόρμες βίντεο (Youtube/Netflix), διαστημικές εφαρμογές κ.λπ. Αντιστρόφως, υπάρχουν και γλώσσες προγραμματισμού ειδικού σκοπού, που είναι σχεδιασμένες για χρήση σε συγκεκριμένο τομέα εφαρμογής. Παραδείγματα περιλαμβάνουν γλώσσες ερωτημάτων βάσης δεδομένων όπως η SQL ή γλώσσες για ανάπτυξη ιστοσελίδων όπως η HTML.



### Γιατί η Python είναι μια διερμηνυόμενη γλώσσα προγραμματισμού:

Όπως αναφέρθηκε και σε προηγούμενο μάθημα οι υπολογιστές καταλαβαίνουν μόνο συμβολοσειρές από 0 και 1, ή αλλιώς την γλώσσα μηχανής. Ωστόσο οι προγραμματιστές χρησιμοποιούν πια γλώσσες προγραμματισμού υψηλού επιπέδου που είναι πολύ κοντά στη φυσική γλώσσα και είναι εύκολα κατανοητές από τον άνθρωπο. Τα προγράμματα λοιπόν αν και είναι γραμμένα σε κάποια γλώσσα προγραμματισμού ωστόσο με τη βοήθεια ειδικών εργαλείων μπορούν να μετατραπούν σε γλώσσα μηχανής για να γίνουν κατανοητά από τον υπολογιστή και να εκτελεστούν. Ένα από αυτά τα εργαλεία είναι ο μεταγλωττιστής. Ο **μεταγλωττιστής** μετατρέπει όλον τον πηγαίο κώδικα, κώδικα γραμμένο σε κάποια γλώσσα προγραμματισμού υψηλού επιπέδου, σε αντικειμενικό κώδικα, κώδικα δηλαδή γραμμένο γλώσσα μηχανής. Την ίδια λειτουργία έχει και ο διερμηνέας/διερμηνευτής ωστόσο έχει την ικανότητα να μετατρέπει μία-μία τις εντολές του πηγαίου κώδικα σε γλώσσα μηχανής και να τις εκτελεί άμεσα χωρίς να απαιτείται προηγουμένως να έχει γίνει η μεταγλώττιση όλου του προγράμματος ή αλλιώς η μετατροπή του σε γλώσσας μηχανής.



Η Python λοιπόν είναι μια διερμηνυόμενη γλώσσα προγραμματισμού αφού η μετατροπή του πηγαίου κώδικα σε γλώσσα μηχανής γίνεται από τον διερμηνέα.

### Δραστηριότητα Επιλογής

Απάντησε στις παρακάτω ερωτήσεις κατανόησης:

**Ερώτηση 1:** Πρόβλημα είναι ...

- μία κατάσταση που μας απασχολεί και πρέπει να λυθεί με κάποιον τρόπο
- ένα ζήτημα που τίθεται απλά προς σκέψη
- ένα θέμα που απλά αναλύουμε σε υπό-προβλήματα χωρίς να μας νοιάζει η λύση του

**Ερώτηση 2:** Η χρήση του υπολογιστή είναι πολύ διαδεδομένη για την επίλυση προβλημάτων διότι..

- είναι πιο έξυπνος από τον άνθρωπο
- μπορεί να διαχειριστεί μεγάλο πλήθος δεδομένων και να εκτελέσει πολλές πράξεις σε λιγότερο χρόνο από τον άνθρωπο, πολύπλοκους υπολογισμούς και επαναληπτικές διαδικασίες.
- μπορεί να εκτελέσει πολλές εντολές χωρίς να μπερδευτεί για την σειρά εκτέλεσης σε μεγάλο χρόνο

**Ερώτηση 3:** Τα προγράμματα που είναι γραμμένα σε μία γλώσσα προγραμματισμού εκτελούνται κατευθείαν από τον υπολογιστή χωρίς να χρειαστεί να περάσουν από ένα στάδιο μετάφρασης σε γλώσσα μηχανής.

- Σωστό
- Λάθος

**Ερώτηση 4:** Διερμηνευτής είναι...

- Ένα πρόγραμμα που μετατρέπει μία-μία εντολή αντικειμενικού κώδικα σε πηγαίο κώδικα.
- Ένα πρόγραμμα που μεταφράζει μία εντολή κάθε φορά από γλώσσα προγραμματισμού σε γλώσσα μηχανής και στην συνέχεια την εκτελεί.
- Ένα πρόγραμμα που μεταφράζει όλον τον κώδικα σε γλώσσα προγραμματισμού συνολικά σε γλώσσα μηχανής και στην συνέχεια το εκτελεί.

**Ερώτηση 5:** Ένας αλγόριθμος έχει αυστηρά τη μορφή μιας σειράς εντολών σε μια γλώσσα προγραμματισμού.

- Σωστό
- Λάθος

**Ερώτηση 6:** Η Python είναι μια γλώσσα προγραμματισμού γενικού σκοπού αφού χρησιμοποιείται μόνο σε εφαρμογές διαχείρισης δεδομένων και δημιουργία παιχνιδιών.

- Σωστό
- Λάθος

**Ερώτηση 7:** Από τα παρακάτω ισχύει ότι...

- Οι γλώσσες προγραμματισμού υψηλού επιπέδου προσφέρουν εντολές εύκολα κατανοητές από τον άνθρωπο και από τον υπολογιστή και διαθέτουν δικό τους συντακτικό και λεξιλόγιο.
- Οι γλώσσες προγραμματισμού είναι πολύ κοντά στην γλώσσα μηχανής γι' αυτό και οι προγραμματιστές τις χρησιμοποιούν για να φτιάξουν κατάλληλους αλγόριθμους κατανοητούς από τους υπολογιστές.
- Ένα πρόγραμμα γραμμένο σε μία γλώσσα προγραμματισμού υψηλού επιπέδου πρέπει να μετατραπεί σε ακολουθίες 0 και 1 ώστε να εκτελεστεί από τον υπολογιστή.

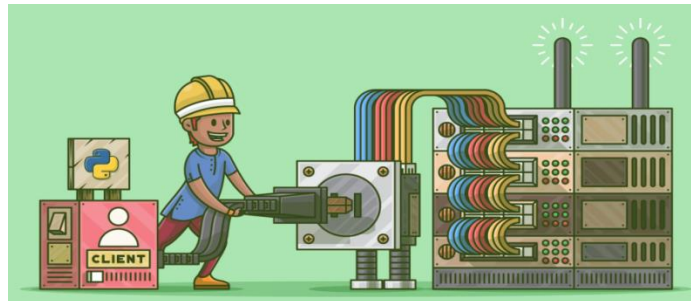
**Ερώτηση 8:** Αντικειμενικός κώδικας είναι ο κώδικας γραμμένος σε μία γλώσσα προγραμματισμού.

- Σωστό
- Λάθος

## Οδηγίες εγκατάστασης της Python:

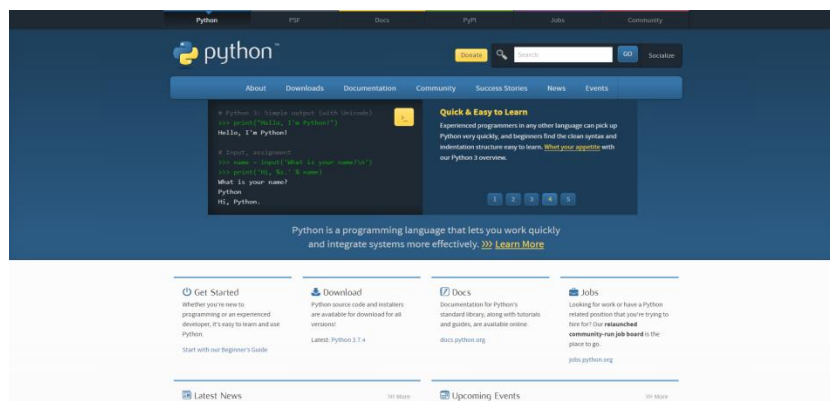
Η Python μπορεί να χρησιμοποιηθεί σε 21 διαφορετικά Λειτουργικά Συστήματα και περιβάλλοντα. Η διαδικασία εγκατάστασης είναι απλή:

- Windows: Μπορείς να την κατεβάσεις δωρεάν από την ιστοσελίδα της Python : <https://www.python.org>.
- Linux: Η Python είναι ήδη εγκατεστημένη, όμως μια παλαιότερη έκδοσή της. Μπορείς να εγκαταστήσεις μια νεότερη έκδοση, κατεβάζοντάς την αντίστοιχα από τον ιστοσελίδα της Python.
- Mac OS X: Πρέπει να υπάρχει ήδη εγκατεστημένη μια έκδοση της Python αλλά το πιθανότερο είναι να πρόκειται για μια παλαιότερη. Μπορείς να εγκαταστήσεις μια νεότερη έκδοση, κατεβάζοντάς την αντίστοιχα από τον ιστοσελίδα της Python: <http://www.python.org/downloads/mac-osx>

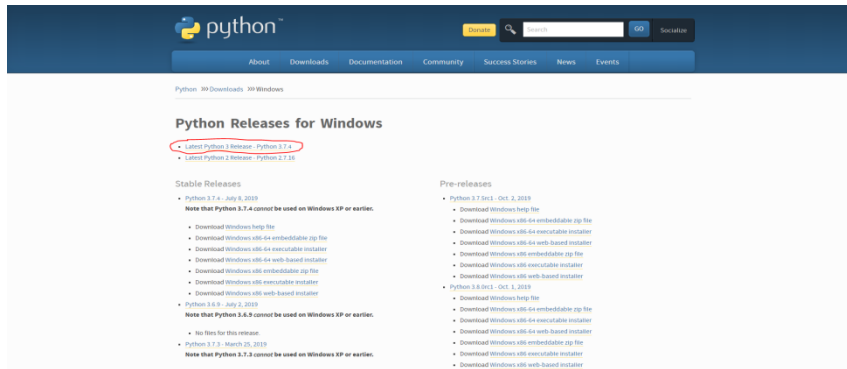


## Εγκατάσταση της Python:

- Άνοιξε τον περιηγητή ιστού (web browser) και πήγαινε στην επίσημη σελίδα της Python : <https://www.python.org>



- Στην καρτέλα Downloads επέλεξε το κατάλληλο λειτουργικό σύστημα του υπολογιστή σου (π.χ. Windows) και επέλεξε την τελευταία έκδοση της Python 3.



- Κατεβείτε χαμηλά μέχρι να βρείτε την κατηγορία Files και κατεβάστε το αρχείο που είναι της μορφής executable installer (ανάλογα με την έκδοση των Windows σας)

**macOS users**

- **Changed in 3.7.4** OpenSSL has been updated from 1.1.0 to 1.1.1 and SQLite updated to 3.28.0.
- For Python 3.7.4, we provide two binary installer options for download. The default variant is 64-bit-only and works on macOS 10.9 (Mavericks) and later systems. **Changed in 3.7.4** The 64-bit/32-bit variant that also works on very old versions of macOS, from 10.6 (Snow Leopard) on, is now deprecated and will no longer be provided in future releases; see the macOS installer `README` file for more info. Both variants come with batteries-included versions of Tcl/Tk 8.6 for users of IDLE and other skinker-based GUI applications, third-party and system versions of Tcl/Tk are no longer used.
- Both python.org installer variants include private copies of OpenSSL. Please carefully read the `Important Information` displayed during installation for information about SSL/TLS certificate validation and the `Install Certificates.command`.

[Full Changelog](#)

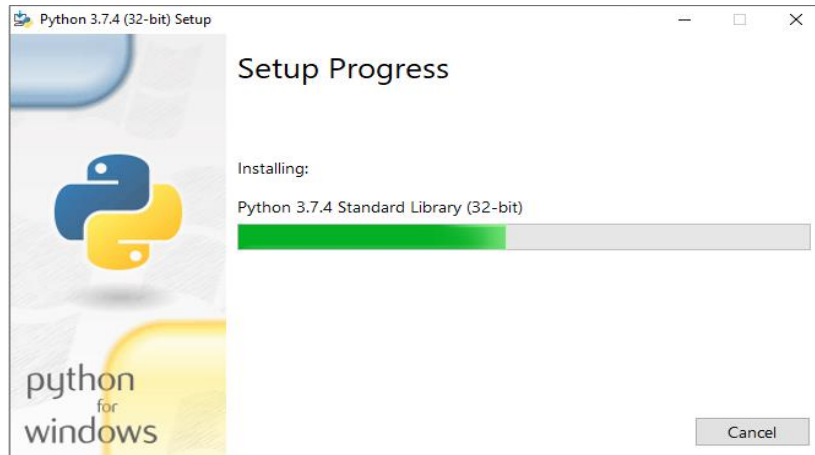
**Files**

Version	Operating System	Description	MDS Sum	File Size	GPG
gzipped source tarball	Source release		6811671e502b04ae7795ab010f090e	23017663	SIG
XZ compressed source tarball	Source release		d33e4aa66097051c2cca45ee3604803	17131432	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6428b4fa7583daf1a442c2ba8c0e086	34898416	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	5d4605c38217a457738f5ea036b241f	28082845	SIG
Windows help file	Windows		d63999573a2c01b2ac56cade6b4f7cd2	8131161	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	9b00c8cf6d9ec0b9ab683184a40729a2	7504391	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	a702b4bdad76debdb3043a5836563400	26680368	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	28c1c088bd73a9e53a3bd351b4bd2	1362904	SIG
Windows x86 embeddable zip file	Windows		9fab3b81f8841879fa9413357413968	6741626	SIG
<b>Windows x86 executable installer</b>	Windows		33cc602942a54446a3d6e451476394789	25663848	SIG
Windows x86 web-based installer	Windows		1b670cf45d317d82c30983ea371d87c	1324608	SIG

- Τρέξτε το αρχείο που μόλις κατεβάσατε χωρίς να πειράξετε τις προτεινόμενες ρυθμίσεις. Ελέγξτε όμως αν το “Add Python to PATH” είναι ενεργοποιημένο και αν επιθυμείτε αλλάξτε και την τοποθεσία εγκατάστασης.



- Πατήστε το “Install Now” που εμφανίζεται στον διάλογο για να ξεκινήσει η εγκατάσταση.



- Η εγκατάσταση έχει τερματίσει επιτυχώς, μόλις εμφανιστεί ο διάλογος που φαίνεται παρακάτω. Πάτησε Close για να κλείσει ο διάλογος



Μπορείς να κοιτάξεις το παρακάτω video tutorial που δείχνει βήμα-βήμα την εγκατάσταση της Python σε Windows. Μπορείς να ενεργοποιήσεις τους αγγλικούς υπότιτλους ή ακόμη να βάλεις μετάφραση στα Ελληνικά.

<https://www.youtube.com/embed/O5nHFBhCfFo>

### Οδηγίες εκτέλεσης προγράμματος Python:

Ως νέος προγραμματιστής που είσαι πρέπει να ξέρεις ότι χρειάζεσαι ένα περιβάλλον για να γράφεις τα αρχεία κώδικά σου και ένα περιβάλλον για να τα εκτελείς.

Πλέον υπάρχει πληθώρα από τέτοια εργαλεία αλλά για τη δική μας διευκόλυνση ως προγραμματιστές, η Python μας προσφέρει ένα ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών (IDE) το οποίο ονομάζεται IDLE (Integrated DeveLopment Environment). Το **IDLE εγκαθίσταται εξαρχής με την εγκατάσταση της Python στα Windows και το βρίσκουμε στη λίστα των προγραμμάτων (εφαρμογών) του υπολογιστή μας.**



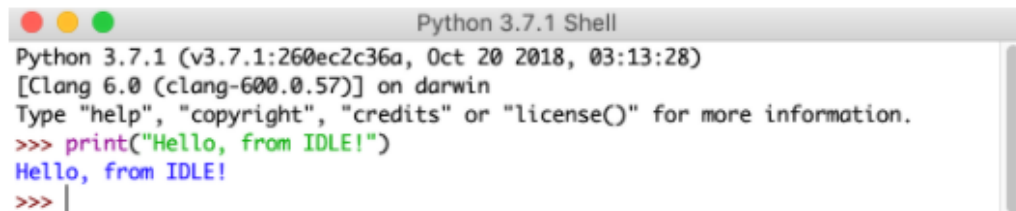


Το IDLE είναι ένα απλό περιβάλλον στο οποίο μπορούμε να γράψουμε και να εκτελέσουμε τα προγράμματά μας. Διαθέτει:

- Παράθυρο Python Shell (κονσόλα του διερμηνευτή της Python) για άμεση εκτέλεση εντολών.
- Παράθυρο συντάκτη προγράμματος (editor), το οποίο υποστηρίζει πολλές δυνατότητες, όπως χρωματική επισήμανση των εντολών, πολλαπλές αναιρέσεις, κατάλληλη στοίχιση των εντολών. Σε αυτό το παράθυρο γράφουμε πρώτα ολόκληρο το πρόγραμμά μας , στη συνέχεια το αποθηκεύουμε σε αρχείο και στο τέλος το εκτελούμε. Τα αποτελέσματα της εκτέλεσης τα βλέπουμε στο Python Shell.
- Αποσφαλματωτή (debugger) για παρακολούθηση της εκτέλεσης των προγραμμάτων μας με σκοπό την εύρεση σφαλμάτων.

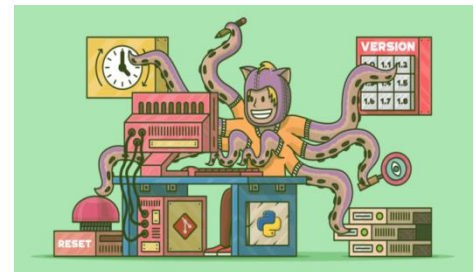
### Εκτέλεση πρώτου προγράμματος στο IDLE:

- Άνοιξε το IDLE
- Στην προτροπή `>>>` γράψε την εντολή: `print('Hello, from IDLE !')`
- Πάτα το πλήκτρο Enter



```
Python 3.7.1 Shell
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 03:13:28)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Hello, from IDLE!")
Hello, from IDLE!
>>> |
```

Είδες. Τόσο εύκολο ήταν! Καθώς το πρόγραμμα εκτελέστηκε εμφανίστηκε και το αντίστοιχο μήνυμα αφού η εντολή `print()` είναι μια εντολή εξόδου. Η προτροπή `>>>` εμφανίζεται ξανά δηλώνοντάς σου ότι η Python περιμένει την επόμενη εντολή σου.



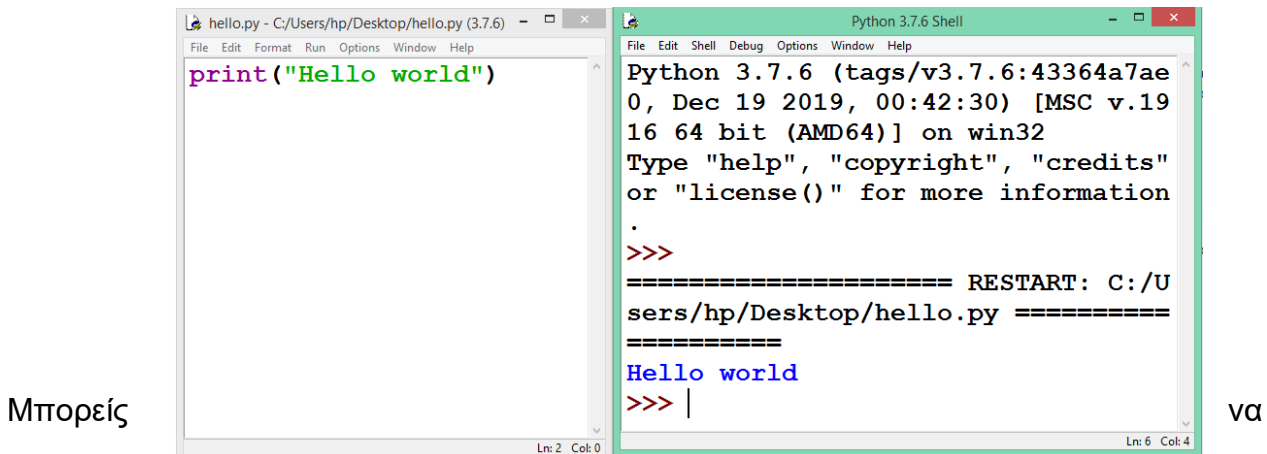
Όμως τα προγράμματα σε Python δεν θα μας ήταν χρήσιμα, εάν έπρεπε να τα ξαναγράφουμε κάθε φορά που τα χρειαζόμασταν. Κάποιες εφαρμογές μπορεί να αποτελούνται από κώδικα χιλιάδων γραμμών εντολών. Έτσι λοιπόν συνετό είναι να δημιουργείς τα δικά σου αρχεία με κώδικα ώστε να τα αποθηκεύεις, να τα εκτελείς και να τα μετατρέπεις όποτε το επιθυμείς.

### Αποθήκευση και εκτέλεση ολοκληρωμένου προγράμματος στην Python:

- Επίλεξε στη γραμμή μενού του IDLE: File->New File.
- Ένα άδειο παράθυρο εμφανίζεται με τίτλο Untitled. Το παράθυρο αυτό είναι ένας συντάκτης προγράμματος (editor) στον οποίο μπορείς να γράψεις όλο σου το πρόγραμμα, να το αποθηκεύσεις σε αρχείο πηγαίου κώδικα και στο τέλος να το εκτελέσεις. Για εξάσκηση μπορείς να γράψεις παραδοσιακή εντολή: `print('Hello, World!')`
- Αποθήκευσε το αρχείο σου το επιλέγοντας: File->Save, δίνοντας όποιο όνομα θέλεις, για παράδειγμα `helloworld.py`, και σε όποιο φάκελο επιθυμείς. Πρόσεξε μοναχά

η κατάληξη που θα δώσεις στο αρχείο σου να είναι.py αφού είναι αρχείο με κώδικα Python.

- Για να το εκτελέσεις επιλέγεις: Run->Run Module. Το αποτέλεσμα της εκτέλεσης εμφανίζεται στο Python Shell.



παρακολουθήσεις το παρακάτω [video-tutorial](https://www.youtube.com/embed/da8YU52iPrE) ώστε να δεις πως να δημιουργείς και να εκτελείς τα δικά σου προγράμματα Python. Στο βίντεο μπορείς να εισάγεις αγγλικούς υπότιτλους ή και ελληνικούς.

<https://www.youtube.com/embed/da8YU52iPrE>

### Υποστηρικτική Δραστηριότητα:

Παρακάτω παρατίθενται μερικά links με βίντεο και πηγές σχετικά με την έννοια του αλγορίθμου και των γλωσσών προγραμματισμού. Μπορείς να τους επισκεφτείς για να μάθεις περεταίρω πληροφορίες. Στα βίντεο μπορείς να εισάγεις αγγλικούς υπότιτλους από τις ρυθμίσεις.

- [Αλγόριθμος: Πως προέκυψε η έννοια αυτή; Ποια είναι τα βήματα δημιουργίας; Ποια είναι τα κριτήρια που πρέπει να τηρεί;](#)
- [Γλώσσες προγραμματισμού χαμηλού επιπέδου](#)
- [Why algorithms are called algorithms?](#)
- <https://www.youtube.com/watch?v=kiFfp-HAu64>
- [Machine Code and High level Languages Using Interpreters and Compilers](#)

## Μάθημα 2: Η Δομή Ακολουθίας στην Python

### Στα προηγούμενα μαθήματα...

Στο προηγούμενο μάθημα έμαθες βασικά πράγματα σχετικά με το τι είναι ένα πρόγραμμα, ένας αλγόριθμος, μία γλώσσα προγραμματισμού καθώς επίσης και πως μπορείς να γράφεις και να εκτελείς το δικό σου πρόγραμμα στην Python.

Είσαι έτοιμος λοιπόν να δημιουργήσεις πιο περίπλοκα προγράμματα και να



ανακαλύψεις τις δυνατότητες που προσφέρει η Python;

Στο μάθημα αυτό θα μάθεις την πρώτη από τις **τρεις δομές** που θα μάθεις συνολικά και θα αναλυθούν στην συνέχεια, τη **δομή ακολουθίας**. Η δομή ακολουθίας περιέχει μια σειρά εντολών οι οποίες εκτελούνται όλες η μία μετά την άλλη, δηλαδή ακολουθιακά!

### Μαθησιακά Αποτελέσματα

Εκτελώντας τις επόμενες δραστηριότητες και προβλήματα θα είσαι σε θέση :

- Να αναγνωρίζεις την εντολή εξόδου, προσδιορίζεις το μήνυμα εκτύπωσης και να τη χρησιμοποιείς για να εμφανίσεις στην οθόνη το μήνυμα που επιθυμείς.
- Να ορίζεις και αναθέτεις τιμή σε μεταβλητές, χρησιμοποιώντας μνημονικές και έγκυρες ονομασίες.
- Να διακρίνεις και να κατονομάζεις τους διαφορετικούς τύπους μεταβλητών.
- Να αναγνωρίζεις την εντολή εισόδου και να εξηγείς τον τρόπο με τον οποίο χειρίζεται το πρόγραμμα τα δεδομένα που δίνονται ως είσοδο.
- Να αναγνωρίζεις τους αριθμητικούς τελεστές και να συντάσσεις μια αριθμητική παράσταση σύμφωνα με τη σειρά προτεραιότητας.



### Δραστηριότητα 1

Έστω στο δωμάτιο σου έχεις έναν έξυπνο υπολογιστή, τη Dory, που καθημερινά σε καλημερίζει και σου δίνει χρήσιμες συμβουλές για τη μέρα σου. Είναι λοιπόν Πέμπτη πρωί και μόλις έχεις ξυπνήσει.

Έστω ότι στον υπολογιστή αυτόν έχουν δοθεί οι παρακάτω εντολές και η Dory τις εκτελεί:

1	<code>print("Καλημέρα!")</code>
2	<code>print ("Πως είσαι σήμερα;")</code>
3	<code>print("Να φορέσεις μπουφάν γιατί θα κάνει κρύο!")</code>

**Ερώτηση 1:** Ποιο πιστεύεις είναι το αποτέλεσμα των παραπάνω εντολών;

---

Μετέφερε τις εντολές αυτές σε ένα αρχείο και εκτέλεσέ το.

**Ερώτηση 2:** Το αποτέλεσμα της εκτέλεσης των παραπάνω εντολών ήταν ίδιο με αυτό που υπέθεσες πριν;

---

Αφού σε καλημέρισε η Dory, θυμήθηκες ξαφνικά ότι δεν έχεις βρει απάντηση στην ερώτηση των μαθηματικών που έχεις για σήμερα: "Πόσα δευτερόλεπτα έχουν οι 3 ώρες".

Επειδή όμως δεν προλαβαίνεις να κάνεις τους υπολογισμούς αυτή τη στιγμή ζητάς τη βοήθεια της Dory ώστε να το υπολογίσει εκείνη για εσένα.

Η Dory αναγνωρίζει την ερώτηση που της κάνεις και για να σου απαντήσει εκτελεί τις εξής εντολές:

```
4 print("Οι 3 ώρες έχουν:")
5 print(3*60*60)
6 print("δευτερόλεπτα")
```

**Ερώτηση 3:** Τι πιστεύεις ότι θα εμφανιστεί με την εκτέλεση αυτών των εντολών;

---

**Ερώτηση 4:** Εκτέλεσε τις καινούριες εντολές. Ποιο ήταν το αποτέλεσμα; Ήταν το ίδιο με αυτό που μάντεψες;

---

**Ερώτηση 5:** Η αριθμητική τιμή που εμφανίστηκε στην οθόνη, σε ποιο κομμάτι του κώδικα πιστεύεις ότι αντιστοιχεί;

---

Τέλος, λίγο πριν φύγεις για το σχολείο η Dory θυμάται να σου εμφανίσει την χρήσιμη πληροφορία της ημέρας και εκτελεί την παρακάτω εντολή:

```
7 #Ωχ! Να μην ξεχάσω να πω την χρήσιμη πληροφορία της ημέρας!
8 print("Να θυμάσαι! Μία μέρα έχει ", 24*60*60, " δευτερόλεπτα")
```

**Ερώτηση 6:** Ποιο πιστεύεις ότι θα είναι το αποτέλεσμα στην οθόνη της Dory αφού εκτελεστούν οι παραπάνω εντολές;

---

**Ερώτηση 7:** Εκτέλεσε τις εντολές. Ποιο ήταν τελικά το αποτέλεσμα; Ποια είναι η λειτουργία της εντολής 8;

---

**Ερώτηση 8:** Γράψε κατάλληλη εντολή που θα πρέπει να εκτελέσει η Dory ώστε να εμφανίσει το μήνυμα: "Καλό μάθημα! Να έχεις μια όμορφη μέρα!!!"

---

Έστω ότι η Dory αντί για τις προηγούμενες εντολές, εκτελεί τις παρακάτω για τον υπολογισμό των δευτερολέπτων:

```
4 3_hours_to_sec=3*60*60
5 unit_of_time="δευτερόλεπτα"
6 print("Οι 3 ώρες έχουν:", 3_hours_to_sec, unit_of_time)
7 #Ωχ! Να μην ξεχάσω να πω την χρήσιμη πληροφορία της ημέρας!
8 1_day_to_sec=24*60*60
9 print("Να θυμάσαι! Μία μέρα έχει ", 1_day_to_sec, unit_of_time)
```

**Ερώτηση 9:** Τι διαφορετικό παρατηρείς σε σχέση με τις προηγούμενες εντολές;

---

**Ερώτηση 10:** Ποιο πιστεύεις ότι θα είναι το αποτέλεσμα της εκτέλεσης των παραπάνω εντολών;

---

**Ερώτηση 11:** Εκτέλεσε τις καινούριες εντολές και αν το αποτέλεσμα είναι διαφορετικό από την υπόθεσή σου κατάγραψέ το.

---

**Ερώτηση 12:** Ποιος πιστεύεις ότι είναι ο ρόλος των μεταβλητών `3_hours_to_sec`, `1_day_to_sec` και `unit_of_time`;

---

Τελικά 1 δευτερόλεπτο πριν φύγεις θυμάται ότι θέλει να σου πει και μια δεύτερη χρήσιμη πληροφορία.

Μελέτησε τις παρακάτω εντολές:

```
4 3_hours_to_sec=3*60*60
5 unit_of_time="δευτερόλεπτα"
6 print("Οι 3 ώρες έχουν:", 3_hours_to_sec, unit_of_time)
7 #Ωχ! Να μην ξεχάσω να πω την χρήσιμη πληροφορία της ημέρας!
8 1_day_to_sec=24*60*60
9 print("Να θυμάσαι! Μία μέρα έχει ", 1_day_to_sec, unit_of_time)
10 2_days_to_min=2*24*60
11 unit_of_time="λεπτά"
12 print("Και... Δύο μέρες έχουν ", 2_days_to_min, unit_of_time)
```

**Ερώτηση 13:** Ποιο είναι το περιεχόμενο της μεταβλητής `unit_of_time` στην εντολή 5 και ποιο στην εντολή 11;

---

**Ερώτηση 14:** Ποιο πιστεύεις θα είναι το μήνυμα που θα εκτυπωθεί όταν εκτελεστεί η εντολή 12;

Εκτέλεσε ξανά τον κώδικα με τις καινούριες εντολές.

---

**Ερώτηση 15:** Μάντεψες σωστά πριν; Αν όχι ποιο μήνυμα εκτυπώθηκε στην εντολή 12; Γιατί συνέβη αυτό;

---

Παρατήρησε τις εξής εντολές:

```
3_hours_to_sec= 3*60*60
1_day_to_sec= 24*60*60
unit_of_time= "δευτερόλεπτα"
print ("Οι 3 ώρες έχουν: 3_hours_to_sec unit_of_time")
```

```
# Ωχ! Να μην ξεχάσω να πω την χρήσιμη πληροφορία της ημέρας!  
print ("Να θυμάσαι! Μία μέρα έχει 1_day_to_sec unit_of_time")
```

```
3_hours_to_sec= 3*60*60  
unit_of_time= "δευτερόλεπτα"  
print ("Οι 3 ώρες έχουν:", 3_hours_to_sec, unit_of_time)  
# Ωχ! Να μην ξεχάσω να πω την χρήσιμη πληροφορία της ημέρας!  
print ("Να θυμάσαι! Μία μέρα έχει", 1_day_to_sec, unit_of_time)  
1_day_to_sec= 24*60*60
```

**Ερώτηση 16:** Πιστεύεις ότι η εκτέλεσή τους θα φέρει το ίδιο αποτέλεσμα με πριν, ή μήπως υπάρχει κάποιο λάθος; Εκτέλεσε τις εντολές για να επαλυθεύσεις το αποτέλεσμα! Ποιο πρόβλημα υπήρχε;

## Δραστηριότητα 2

Καθώς λοιπόν τελείωσε μία κουραστική μέρα στο σχολείο, επιστρέφοντας στο σπίτι φέρνεις μαζί σου το ξαδερφάκι σου, το μικρό Γιαννάκη, για να τον βοηθήσεις με τα μαθηματικά του. Η Dory ακούει μία άγνωστη φωνή και θέλει να μάθει ποιο είναι αυτό το καινούριο άτομο.

```
1 cousin_name=input("Πως είναι το όνομά σου;")  
2 cousin_surname=input("Πως είναι το επίθετό σου;")  
3 print("Γεια σου", cousin_name, cousin_surname, " Εγώ είμαι η Dory")
```

**Ερώτηση 1:** Ποιο πιστεύεις ότι θα είναι το αποτέλεσμα της εκτέλεσης των παραπάνω εντολών;

**Ερώτηση 2:** Αφού εκτελέσεις τις εντολές ποια πιστεύεις ότι είναι η λειτουργία της input();

Με την εκτέλεση της εντολής input() συμβαίνουν 3 γεγονότα:

1. Εκχώρηση τιμής εισόδου σε μια μεταβλητή.
2. Είσοδος τιμής από το πληκτρολόγιο.
3. Εμφάνιση μηνύματος στην οθόνη (αν περιέχεται κάποιο μήνυμα μέσα στην input()).

**Ερώτηση 3:** Με ποια σειρά πιστεύεις ότι συμβαίνουν τα παραπάνω γεγονότα.

Η Dory συνεχίζει με τις ερωτήσεις σχετικά με το μικρό Γιαννάκη. Τα καταφέρνει πιστεύεις;

4	<code>year_of_birth= input("Ποια χρονιά γεννήθηκες;")</code>
5	<code>age= 2020-year_of_birth</code>
6	<code>print ("Α! Ωστε είσαι", age, "χρονών")</code>

**Ερώτηση 3:** Ποιο πιστεύεις ότι είναι το αποτέλεσμα της εκτέλεσης των παραπάνω εντολών;

---

**Ερώτηση 4:** Τι τύπου μεταβλητή είναι η `year_of_birth`; Υπάρχει κάποιο πρόβλημα στην εκτέλεση των παραπάνω εντολών;

---

Η Dory κατάλαβε ότι κάτι πάει λάθος και ξαναρωτάει.

4	<code>year_of_birth= int(input("Ποια χρονιά γεννήθηκες;"))</code>
5	<code>age= 2020-year_of_birth</code>
6	<code>print ("Α! Ωστε είσαι", age, "χρονών")</code>

**Ερώτηση 5:** Ποια διαφορά παρατηρείς σε σχέση με πριν;

---

Ξανά εκτέλεσε τις καινούριες εντολές.

**Ερώτηση 6:** Ποιο είναι το αποτέλεσμα της εκτέλεσης των παραπάνω εντολών;

---

**Ερώτηση 7:** Τι τύπου μεταβλητή είναι τώρα η `year_of_birth` και ποιος πιστεύεις ότι είναι ο ρόλος της `int()`;

---

Ο μικρός Γιαννάκης πρέπει να λύσει ασκήσεις πάνω στην εύρεση ακέραιου μέρους δεκαδικών αριθμών και ζητάς φυσικά τη βοήθεια της Dory.

7	<code>print ( "Η ακέραια τιμή που αντιστοιχεί στο 123 είναι η:", int(123) )</code>
8	<code>print ( "Η ακέραια τιμή που αντιστοιχεί στο 123.23 είναι η:", int(123.23) )</code>
9	<code>print ( "Η ακέραια τιμή που αντιστοιχεί στο '123' είναι η:", int("123") )</code>

**Ερώτηση 8:** Γνωρίζοντας τη λειτουργία της `int()`, ποιο πιστεύεις είναι το αποτέλεσμα της εκτέλεσης των εντολών;

---

**Ερώτηση 9:** Εκτέλεσε τις παραπάνω εντολές και παρατήρησε το αποτέλεσμα της εκτέλεσης. Μάντεψε σωστά πριν; Ποιο είναι τελικά το αποτέλεσμα της εκτέλεσης;

---

10	<code>int_value= int (input ("Δώσε μου τον επόμενο δεκαδικό αριθμό:"))</code>
----	---

11	<code>print ("Το ακέραιο μέρος είναι:", int_value)</code>
----	---

**Ερώτηση 10:** Ποιο πιστεύεις είναι το αποτέλεσμα της εκτέλεσης των παραπάνω εντολών;

**Ερώτηση 11:** Εκτέλεσε τις εντολές. Υπήρχε κάποιο λάθος και αν πως διορθώνεται;

12	<code>print ( "Η δεκαδική τιμή που αντιστοιχεί στο 123 είναι η:", float(123) )</code>
13	<code>print ( "Η δεκαδική τιμή που αντιστοιχεί στο 123.23 είναι η:", float(123.23) )</code>
14	<code>print ( "Η δεκαδική τιμή που αντιστοιχεί στο '123' είναι η:", float("123") )</code>
15	<code>print ( "Η δεκαδική τιμή που αντιστοιχεί στο '123.4' είναι η:", float("123.4") )</code>

**Ερώτηση 12:** Αν η float() έχει αντίστοιχη λειτουργία με αυτή της int(), δηλαδή επιστρέφει την αντίστοιχη δεκαδική τιμή, ποιο πιστεύεις είναι το αποτέλεσμα της εκτέλεσης των παραπάνω εντολών;

**Ερώτηση 13:** Εκτέλεσε τις παραπάνω εντολές και παρατήρησε το αποτέλεσμα της εκτέλεσης. Αν το αποτέλεσμα ήταν διαφορετικό από την υπόθεσή σου κατάγραφέ το.

Αφού ο μικρός Γιαννάκης βρήκε με επιτυχία τα ακέραια μέρη δεκαδικών αριθμών, σειρά έχουν μερικά περίπλοκα προβλήματα. Μάλλον θα ξανά χρειαστείς την βοήθεια της Dory.

Ο μικρός Γιαννάκης περιέγραψε το πρόβλημα και η Dory για να του δώσει την λύση εκτελεί τις παρακάτω εντολές:

16	<code>hours= 35.0</code>
17	<code>rate= 12.50</code>
18	<code>payment= hours * rate</code>
19	<code>print ("The payment is", payment, "euros")</code>

**Ερώτηση 14 :** Σύμφωνα με τις εντολές τι πρόβλημα πιστεύεις έπρεπε να λύσει ο μικρός Γιαννάκης;

**Ερώτηση 15 :** Πόσο βοηθητικά πιστεύεις ότι είναι τα ονόματα που δίνει κάποιος στις μεταβλητές του;

Ο μικρός Γιαννάκης συνεχίζει με τα προβλήματα και η Dory για να το λύσει αυτή τη φορά πρέπει να υπολογίσει μία περίπλοκη αριθμητική παράσταση.

20	<code>x= 1+2**3/4*5</code>
21	<code>print (x)</code>



**Ερώτηση 16:** Ποια αριθμητική πράξη αντιστοιχεί σε κάθε αριθμητικό τελεστή;

---

**Ερώτηση 17:** Ποια τιμή πιστεύεις θα εκτυπωθεί αν εκτελεστούν οι παραπάνω εντολές;

---

**Ερώτηση 18:** Εκτέλεσε τις εντολές. Τελικά μάντεψες σωστά πριν; Αν όχι τι τιμή πήρε τελικά η μεταβλητή  $x$  και γιατί πιστεύεις συνέβη αυτό;

---

22	<code>y= 1+2**3//4*5</code>
23	<code>print (y)</code>

**Ερώτηση 19:** Τι διαφορά παρατηρείς στις παραπάνω εντολές σε σχέση με τις προηγούμενες; Η εκτέλεση των παραπάνω εντολών πιστεύεις θα επιφέρει το ίδιο αποτέλεσμα με πριν;

---

**Ερώτηση 20:** Εκτέλεσε τις εντολές. Ποια τιμή παίρνει τελικά η μεταβλητή  $y$  και γιατί;

---

Ο μικρός Γιαννάκης μόλις τελείωσε με τις ασκήσεις του και εσύ μόλις βρήκες ευκαιρία ζήτησες από την Dory να σου θυμίσει έναν αριθμό που της είπες χθες.

24	<code>part1= "519"</code>
25	<code>part2= "2028"</code>
26	<code>the_number= part1+part2</code>
27	<code>print ("Your number is", the_number)</code>

**Ερώτηση 21:** Ποια τιμή πιστεύεις ότι παίρνει η μεταβλητή `the_number` και ποιο το αποτέλεσμα της εκτέλεσης των εντολών;

---

**Ερώτηση 22:** Εκτέλεσε τις εντολές. Τελικά μάντεψες σωστά πριν; Αν όχι ποια τιμή πήρε τελικά η μεταβλητή `the_number`; Γιατί συνέβη αυτό;

---

## Ασκήσεις Κατανόησης:

**Άσκηση 1:** Σε ποιον τύπο δεδομένων ανήκουν οι παρακάτω τιμές:

- 17
- 'Integer'
- 2.5
- '17'
- '2.5'

**Άσκηση 2:** Τι θα εκτυπωθεί στην οθόνη μετά την εκτέλεση των παρακάτω εντολών:

```
print ("Hello my name is 'John' ")
```

```
print (1, 000. 000)
```

**Άσκηση 3:** Είναι αποδεκτά τα παρακάτω ονόματα μεταβλητών; Αν όχι, γιατί;

1. 76trombones
2. more@
3. global

**Άσκηση 4:** Τι θα εκτυπωθεί στην οθόνη μετά την εκτέλεση των παρακάτω εντολών

- first = '100'
- second = '150'
- third = first + second
- print (third)

**Άσκηση 5:** Έστω ότι έχουμε κάνει τις εξής αναθέσεις μεταβλητών:

- width = 17
- height = 12.0

Τι τιμές θα λάβουν οι παρακάτω μεταβλητές;

- a=width//2
- b=width/2
- c=height//3
- d=height/3

**Άσκηση 6:** Έστω πρόγραμμα το οποίο ζητάει από τον χρήστη το ύψος και την ακτίνα της βάσης ενός κώνου και υπολογίζει τον όγκο του βάσει του τύπου  $V=(\pi r^2 h)/3$ . Το πρόγραμμα εκτυπώνει κατάλληλο μήνυμα με το αποτέλεσμα.

Κάποιος στο πρόβλημα αυτό έδωσε της εξής λύση:

```
1 height=input('Enter height of the cone in cm: ')
2 radius=input('Enter radius of the cone in cm: ')
3 pi=3.14
4 volume=(pi*radius**2*height)/3
5 print('Volume of cone is', volume, 'cm3')
```

Αξιολόγησε τον κώδικα ως προς

1. την ορθότητα των μεταβλητών που έχουν χρησιμοποιηθεί.
2. τα μηνύματα εκτύπωσης που περιέχουν οι εντολές εισόδου-εξόδου .
3. την ορθότητα του κώδικα.

## Προβλήματα Επιλογής Ομάδα Α

**Πρόβλημα 1:** Γράψε ένα πρόγραμμα που ζητάει σαν είσοδο από τον χρήστη μία θερμοκρασία σε βαθμούς Κελσίου και τη μετατρέπει σε βαθμούς Φαρενάιτ. Το πρόγραμμα εκτυπώνει κατάλληλο μήνυμα με το αποτέλεσμα.

**!Tip:**

Τύπος μετατροπής θερμοκρασίας: Fahrenheit Temperature = Celsius Temperature x 1.8 + 32

**Πρόβλημα 2:** Γράψε ένα πρόγραμμα που ζητάει σαν είσοδο από τον χρήστη πόσες ώρες δούλεψε, πόσα λεφτά πληρώνεται την ώρα και να υπολογίζει την αμοιβή του. Το πρόγραμμα θα εκτυπώνει κατάλληλο μήνυμα με το αποτέλεσμα.

**Πρόβλημα 3:** Να γράψετε πρόγραμμα που θα ζητάει από το χρήστη το βάρος του στη Γη και θα εμφανίζει το βάρος του στη Σελήνη, την Αφροδίτη και τον Ήλιο.

Tip!

Στη Σελήνη το βάρος ενός αντικειμένου είναι το 1/6 του βάρους του στη Γη. Στην Αφροδίτη το βάρος ενός αντικειμένου είναι 0,9 φορές το βάρος του στη Γη. Στον Ήλιο το βάρος ενός αντικειμένου είναι 27,07 φορές το βάρος του στη Γη, αλλά όταν βρίσκεται κανείς εκεί η αύξηση του βάρους δεν είναι το βασικότερο πρόβλημα.

## Προβλήματα Επιλογής Ομάδα Β

**Πρόβλημα 4:** Γράψε ένα πρόγραμμα που ζητάει σαν είσοδο από τον χρήστη το κόστος του γεύματος του χωρίς ΦΠΑ. Το ΦΠΑ είναι 23% της αξίας του γεύματος. Ο χρήστης μένοντας ευχαριστημένος από το γεύμα του, θέλει να αφήσει και φιλοδώρημα το 5% της αξίας του γεύματος (με ΦΠΑ). Το πρόγραμμα λοιπόν κάνει τους κατάλληλους υπολογισμούς και εκτυπώνει κατάλληλο μήνυμα με το τελικό πόσο που πρέπει να πληρώσει ο χρήστης, δηλαδή κόστος γεύματος και φιλοδώρημα.

**Πρόβλημα 5:** Υπάρχουν πολλά παιχνίδια στα οποία κάποιος σας ζητά να σκεφτείτε έναν μυστικό αριθμό και, μετά από μερικές ερωτήσεις και υπολογισμούς, «μαντεύει» τον αριθμό που είχατε σκεφτεί. Ένα πολύ παλιό παράδειγμα βασίζεται σ' ένα θεώρημα του κινέζου Sun Tzu, που έζησε κάπου ανάμεσα στον 3ο και τον 5ο αιώνα. Ας υποθέσουμε ότι ο άγνωστος αριθμός είναι ο  $x$ . Αν ονομάσουμε  $\alpha$ ,  $\beta$  και  $\gamma$  τα υπόλοιπα της διαίρεσης του  $x$  με το 3, το 5 και το 7 αντίστοιχα, τότε για να βρούμε τον άγνωστο αριθμό θα πρέπει να υπολογίσουμε την τιμή της παράστασης  $70\alpha + 21\beta + 15\gamma$  και στη συνέχεια να διαιρέσουμε με το 105. Το υπόλοιπο της διαίρεσης θα είναι ο μυστικός αριθμός.

Να αναπτύξετε ένα πρόγραμμα το οποίο ζητάει από το χρήστη να σκεφτεί έναν μυστικό αριθμό από το 1 μέχρι και το 100. Στη συνέχεια, το πρόγραμμα ρωτάει το χρήστη ποιο είναι το υπόλοιπο της διαίρεσης αυτού του αριθμού με το 3, το 5 και το 7 και ανακοινώνει τον μυστικό αριθμό, αφήνοντας το χρήστη έκπληκτο με τις υπερφυσικές του ικανότητες!

## Πρόβλημα Διακλάδωσης:

Γράψε ένα πρόγραμμα που ζητάει ως είσοδο από τον χρήστη έναν ακέραιο τετραψήφιο αριθμό και στην συνέχεια να εκτυπώνει τον αντεστραμμένο του αριθμό με κατάλληλο μήνυμα. Δηλαδή αν ο χρήστης εισάγει τον αριθμό 1739 το πρόγραμμα πρέπει να εκτυπώσει τον 9371.

! Θεωρούμε ότι η είσοδος του χρήστη είναι πάντα τετραψήφιος αριθμός.

## &gt;Tips:

- Σκέψου πως μπορείς να αποσπάσεις τα ψηφία ένα-ένα από έναν αριθμό. Δηλαδή αν έστω έχει τον 23, με ποια πράξη μπορείς να πάρεις το 2 ως ψηφίο δεκάδων και με ποια το 3 ως ψηφίο μονάδων;
- Σκέψου πως μπορείς να συνθέσεις έναν αριθμό από την αρχή γνωρίζοντας τα ψηφία από τα οποία θα αποτελείται. Δηλαδή αν ξέρεις ότι το ψηφίο δεκάδων είναι το 3 και το ψηφίο μονάδων το 2, με ποια πράξη θα δημιουργήσεις το 32;

**Υποστηρικτική Δραστηριότητα****Θεωρία:****Τιμή και Τύποι:**

Μία τιμή είναι από τα βασικά πράγματα που χειρίζεται ένα πρόγραμμα. Οι τιμές διακρίνονται σε:

- Ακέραιες τιμές ( int από το integer) π.χ. 100,4,7,4055
- Δεκαδικές τιμές που ανήκουν στους πραγματικούς αριθμούς (float από την έκφραση floating point) π.χ. 55.8, 2.3, 300.1
- Συμβολοσειρές, δηλαδή μια σειρά από σύμβολα/χαρακτήρες (str από το string) π.χ. "Πέμπτη", "a", :Hello world", "20088", "\$"
- Λογικές τιμές ( bool από το boolean) που είναι η True και η False

```
>>> print(123)
123
>>> print(98.6)
98.6
>>> print('Hello world')
Hello world
```

```
>>> type(3.2)
<class 'float'>
```

Ένα string ξεχωρίζει καθώς κλείνεται μέσα σε εισαγωγικά. Μπορούμε να χρησιμοποιήσουμε είτε μονά είτε διπλά εισαγωγικά, αλλά προσοχή με όποια εισαγωγικά «ανοίγουμε» με αυτά πρέπει και να «κλείνουμε».

Αν δεν είσαι σίγουρος ποιος τύπος ανήκει σε μία τιμή, τότε ο διερμηνευτής μπορεί να σου πει με τη χρήση της συνάρτησης type().

**Μεταβλητές:**

- Ένα από τα πιο ισχυρά χαρακτηριστικά μιας γλώσσας προγραμματισμού είναι η δυνατότητα της να χειρίζεται μεταβλητές.
- Μια μεταβλητή είναι μία θέση στη μνήμη του υπολογιστή και έχει ένα μοναδικό όνομα. Σε αυτήν μπορούμε να εκχωρούμε τιμές, δηλαδή να αποθηκεύουμε προσωρινά κάποια τιμή.
- Μια εντολή εκχώρησης δημιουργεί μια νέα μεταβλητή και της δίνει κάποια τιμή.
- Δεν μπορούμε να έχουμε δύο μεταβλητές με το ίδιο όνομα. Αλλά μπορούν δύο μεταβλητές να έχουν την ίδια τιμή. Μια μεταβλητή μπορεί να έχει μόνο μια τιμή κάθε φορά, αλλά μπορούμε να αλλάζουμε / αντικαθιστούμε την τιμή αυτή όποτε θέλουμε, σβήνοντας έτσι την παλιά της τιμή.

```
x = 12.2    x 12.2
y = 14      y 14
```

```
x = 12.2    x 12.2
y = 14      y 14
x = 100     x 100
```

Ανάλογα με την τιμή που τους ανατίθεται διακρίνονται σε :

- Αριθμητικές: Αν στην μεταβλητή καταχωρείται μια ακέραια ή πραγματική τιμή
- Αλφαριθμητικές: Αν στην μεταβλητή καταχωρείται μια συμβολοσειρά
- Λογικές: Αν στην μεταβλητή καταχωρείται η λογική τιμή True ή False.
- (Λίστες, Πλειάδες, Λεξικά)

Παρατηρήσεις:

- Δεν χρειάζεται να δηλωθούν σε κανένα σημείο του προγράμματος
- Έχουν τύπο, αλλά ο τύπος αποκτάται όταν τους πρωτοεκχωρηθεί τιμή
- Ο τύπος τους μπορεί να αλλάξει κατά την εκτέλεση του προγράμματος

```
>>> my_var = 10
>>> type ( my_var )
<class 'int' >
>>> my_var = "Πέμπτη"
>>> type ( my_var )
<class 'str' >
```

## Ονομασία Μεταβλητών

1. Τα ονόματα μεταβλητών μπορούν να περιέχουν γράμματα και αριθμούς, αλλά πρέπει να ξεκινάνε με ένα γράμμα. (Δεν μπορούν να ξεκινάνε με αριθμό, αλλά μπορούν να έχουν αριθμούς ενδιάμεσα ή στο τέλος.)
2. Τα πεζά γράμματα διακρίνονται από τα κεφαλαία, π.χ. Course και το course είναι διαφορετικές μεταβλητές. Είναι επιτρεπτό τα ονόματα των μεταβλητών να ξεκινάνε με κεφαλαίο γράμμα, αλλά καλό είναι να ξεκινάνε με ένα μικρό γράμμα.
3. Ο χαρακτήρας 'underscore' ( \_ ) μπορεί να εμφανιστεί σε ένα όνομα. Συχνά χρησιμοποιείται σε ονόματα με πολλές λέξεις, όπως my\_name ή the\_first\_lesson. Γενικά καλό είναι οι μεταβλητές να μην ξεκινάνε με τον χαρακτήρα αυτό, αν και δεν είναι λάθος.
4. Δεν μπορεί να γίνει χρήση κάποιου άλλου χαρακτήρα.
5. Όπως σε κάθε γλώσσα προγραμματισμού, έτσι και στη Python υπάρχουν κάποιες δεσμευμένες λέξεις ή αλλιώς λέξεις κλειδιά. Ο διερμηνευτής χρησιμοποιεί αυτές τις λέξεις για να αναγνωρίσει τη δομή του προγράμματος.

```
False   class   return
None    if       for
True    def      from
and     del      global
as      elif     try
assert  else     import
break   except  in
```

- Εάν δεν ονομάσεις σωστά μια μεταβλητή, σύμφωνα με τους παραπάνω κανόνες, ή αν χρησιμοποιήσεις κάποια λέξη-κλειδί της Python θα λάβεις ένα σφάλμα σύνταξης.
- Οι προγραμματιστές γενικά επιλέγουν ονόματα για τις μεταβλητές τους που έχουν νόημα και χαρακτηρίζουν τη χρησιμότητα της μεταβλητής. Αυτό συμβάλει στο να είναι το πρόγραμμα ευανάγνωστο!

```
is      finally
lambda  continue
while   nonlocal
not     with
or      yield
pass
raise
```

## Εκτέλεση και Ανάθεση

Κατά την εκτέλεση των προγραμμάτων ο κώδικας εκτελείται από πάνω προς τα κάτω και οι εντολές εκτελούνται μία-μία με τη σειρά. Η ανάθεση τιμής σε μια μεταβλητή γίνεται με τον τελεστή ανάθεσης ( = ). Αριστερά βρίσκεται το όνομα της μεταβλητής και δεξιά βρίσκεται μια έκφραση. Προσοχή, δεν πρόκειται για ισοδυναμία. Αλλά το αποτέλεσμα της έκφρασης στα δεξιά

```
x = 2          ← Assignment statement
x = x + 2     ← Assignment with expression
print(x)      ← Print statement
```

εκχωρείται στην μεταβλητή αριστερά από τον τελεστή ανάθεσης.

- Μία έκφραση είναι ένας συνδυασμός από τιμές, μεταβλητές και αριθμητικούς τελεστές.

```
x = 3.9 * x * ( 1 - x )
```

## Αριθμητικοί Τελεστές

Οι αριθμητικοί τελεστές είναι ειδικά σύμβολα, τα οποία αντιπροσωπεύουν αριθμητικές πράξεις όπως η πρόσθεση ή ο πολλαπλασιασμός.

Το πηλίκο της διαίρεσης (x/y) στην Python έκδοση 3 έχει σαν αποτέλεσμα τον πραγματικό αριθμό που αντιστοιχεί στην διαίρεση. Αν θέλεις το αποτέλεσμα της πράξης να είναι ακέραιος αριθμός τότε χρησιμοποιείς τον τελεστή: //. Το αποτέλεσμα της πράξης αυτής είναι ο στρογγυλοποιημένος προς τα κάτω ακέραιος αριθμός που αντιστοιχεί στο πηλίκο της διαίρεσης .

Operator	Operation
+	Addition
-	Subtraction
*	Multiplication
/	Division
**	Power
%	Remainder

```
>>> xx = 2
>>> xx = xx + 2
>>> print(xx)
4
>>> yy = 440 * 12
>>> print(yy)
5280
>>> zz = yy / 1000
>>> print(zz)
5.28
```

```
>>> jj = 23
>>> kk = jj % 5
>>> print(kk)
3
>>> print(4 ** 3)
64
```

## Σειρά Προτεραιότητας Πράξεων

- Όταν σε μία έκφραση εμφανίζονται πάνω από ένας αριθμητικός τελεστής τότε τηρείται η παρακάτω σειρά προτεραιότητας.
- Παρενθέσεις > Δυνάμεις > Πολλαπλασιασμός > Πρόσθεση > Αριστερά προς τα Δεξιά
- Γενικά καλό είναι να χρησιμοποιείς παρενθέσεις όταν γράφεις κώδικα, για να γίνει η έκφραση πιο ξεκάθαρη. Επίσης βοηθάει να σπας μεγάλες εκφράσεις σε μικρότερες, ώστε να είναι πιο απλές και εύκολα κατανοητές.
- Εκτός από πρόσθεση αριθμών ο τελεστής της πρόσθεσης μπορεί να χρησιμοποιηθεί και για να ενώσει δύο strings σε ένα.

```
>>> ddd = 1 + 4
>>> print(ddd)
5
>>> eee = 'hello ' + 'there'
>>> print(eee)
hello there
```

## Είσοδος και Αλλαγή Τύπου

Μερικές φορές χρειάζεται να ζητήσουμε είσοδο κάποιου δεδομένου από τον χρήστη μέσω του πληκτρολογίου και να το χρησιμοποιήσουμε στην συνέχεια. Η Python διαθέτει την συνάρτηση input() για αυτόν τον σκοπό.

- Όταν γίνει η κλήση της συνάρτησης αυτής το πρόγραμμα σταματάει την εκτέλεση του κώδικα και περιμένει από τον χρήστη να πληκτρολογήσει κάτι.
- Όταν ο χρήστης τελειώσει με την πληκτρολόγηση, πατώντας Enter, η εκτέλεση του προγράμματος συνεχίζεται

```
name = input('Who are you? ')
print('Welcome', name)
```

- Η συνάρτηση `input()` επιστρέφει ό,τι πληκτρολόγησε ο χρήστης με τη μορφή `string` και το εκχωρεί στην αντίστοιχη μεταβλητή.
- Πριν ζητήσεις από τον χρήστη κάποια είσοδο, είναι καλή ιδέα να εκτυπώσεις αντίστοιχο μήνυμα εξηγώντας τι περιμένεις να σου δώσει ως είσοδο.

Πολλές φορές θέλουμε ο χρήστης να εισάγει έναν αριθμό και αυτόν τον αριθμό μετά να τον επεξεργαστούμε! Θα πρέπει να μετατρέψουμε τον τύπο της εισόδου από `string` σε `integer` ή `float` ανάλογα τι ζητάμε από τον χρήστη. Αν ο χρήστης δώσει σαν είσοδο έναν πραγματικό αριθμό και θέλουμε να πάρουμε τον αντίστοιχο ακέραιο τότε πρώτα μετατρέπουμε την συμβολοσειρά σε πραγματικό αριθμό και μετά σε ακέραιο αριθμό.

```
inp = input('Europe floor?')
usf = int(inp) + 1
print('US floor', usf)
```

## Έξοδος

Καθώς εκτελούμε το πρόγραμμά για να εμφανίσουμε στην οθόνη διάφορα μηνύματα χρησιμοποιούμε την συνάρτηση `print()`. Μπορεί να θέλουμε να τυπώσουμε ολόκληρες συμβολοσειρές, ακέραιες και πραγματικές τιμές, αλλά και μεταβλητές.

- Οτιδήποτε εγκλείετε σε εισαγωγικά, μονά ή διπλά, τυπώνεται αυτούσιο αφού αντιστοιχεί σε μία συμβολοσειρά.

```
>>> print ("Hello world!")
Hello world!
```

- Αντίστοιχα μπορούμε να βάλουμε οποιουδήποτε τύπου τιμές: ακέραιες, πραγματικές, λογικές, ακόμη και ολόκληρες εκφράσεις ή συναρτήσεις που εκφέρουν σαν αποτέλεσμα μία τιμή.

- Όταν γράφουμε ονόματα μεταβλητών, οποιουδήποτε τύπου, τυπώνεται η αντίστοιχη τιμή της μεταβλητής.

```
>>> name= " Ελένη "
>>> print ("Γεια σου", name, "! Τι
κάνεις;")
Γεια σου Ελένη ! Τι κάνεις;
```

- ❖ Για να γίνει η αλλαγή γραμμής ανάμεσα στα μηνυμάτά μας χρησιμοποιούμε τον χαρακτήρα αλλαγής γραμμής: `"\n"` ο οποίος πρέπει να εγκλείεται από εισαγωγικά.

```
>>> print (" Καλημέρα!")
>>> print ("Πως σε λένε;")
Καλημέρα!
Πως σε λένε;
>>> print ("Γεια σου.\nΕίμαι η Αλίκη!")
Γεια σου.
Είμαι η Αλίκη;
```

- ❖ Στην Python στην συνάρτηση `print()` υπάρχει το όρισμα `end` όπου αν δεν καταχωρίσουμε κάτι, καταχωρείται αυτόματα ο χαρακτήρας αλλαγής γραμμής. Αν πάλι θέλουμε να μην γίνει αλλαγή γραμμής ανάμεσα σε δύο εντολές `print()`, στο όρισμα αυτό καταχωρούμε τον χαρακτήρα που επιθυμούμε π.χ. `" "` ή `"-"`.

- ❖ Η `print()` είναι μια πολύ χρήσιμη συνάρτηση καθώς μπορούμε να ελέγχουμε τις διάφορες τιμές που παίρνει μια μεταβλητή κατά την εκτέλεση του προγράμματος.

```
# Get the name of the file and open it
name = input('Enter file:')
handle = open(name, 'r')

# Count word frequency
counts = dict()
for line in handle:
    words = line.split()
    for word in words:
        counts[word] = counts.get(word,0) + 1

# Find the most common word
bigcount = None
bigword = None
for word,count in counts.items():
    if bigcount is None or count > bigcount:
        bigword = word
        bigcount = count

# All done
print(bigword, bigcount)
```

## Σχόλια

Όσο ο κώδικας ενός προγράμματος γίνεται μεγαλύτερος και πιο περίπλοκος είναι δύσκολο κανείς να καταλάβει τι κάνει ή γιατί. Είναι λοιπόν καλή ιδέα να προσθέτεις σχόλια στον κώδικα

εξηγώντας σε φυσική γλώσσα τι κάνει ο κώδικας σου ή γιατί.

Τα σχόλια στην Python ξεκινάνε με το σύμβολο #. Οτιδήποτε μετά το σύμβολο αυτό και μέχρι το τέλος της γραμμής θα αγνοηθεί κατά την εκτέλεση του προγράμματος.

### Μάθημα 3: Η Δομή Επιλογής στην Python

#### Στα προηγούμενα μαθήματα...

μελέτησες και εκτέλεσες προγράμματα που περιείχαν αποκλειστικά την δομή ακολουθίας. Η δομή ακολουθίας είναι μία από τις τρεις βασικές δομές στον προγραμματισμό και είναι η πιο απλή. Μία δομή ακολουθίας περιέχει εντολές οι οποίες εκτελούνται μία μετά την άλλη.



Τι γίνεται όμως στην περίπτωση που θέλουμε ένα σύνολο εντολών να εκτελεστεί κάτω από συγκεκριμένες συνθήκες;

Στην περίπτωση αυτή χρειάζεσαι τη **δομή επιλογής**.

Η δομή επιλογής έχει τέσσερις μορφές:

1. απλή
2. σύνθετη
3. πολλαπλή
4. εμφωλευμένη

#### Μαθησιακά Αποτελέσματα

Εκτελώντας τις επόμενες δραστηριότητες θα είσαι σε θέση :

- Να κατονομάζεις τις 4 μορφές της δομής επιλογής
- Να εξηγείς τις διαφορές που έχουν οι δομές επιλογής ως προς την σύνταξη και ως προς την λειτουργία τους
- Να αναγνωρίζεις σε δοσμένο κώδικα σε ποια μορφή ανήκει η εκάστοτε δομή επιλογής
- Να υπολογίζουν τις διαφορετικές τιμές που μπορεί να πάρει μια λογική συνθήκη (Αληθής Ψευδής) και να εντοπίζουν ποια ή ποιες εντολές θα εκτελεστούν ανάλογα με την τιμή της συνθήκης.
- Να αποφασίζεις ποια είναι η κατάλληλη δομή επιλογής που πρέπει να χρησιμοποιήσεις στο πρόγραμμά σου, για να λύσεις ένα δοσμένο πρόβλημα
- Να συντάξουν προγράμματα που απαιτούν χρήση δομών επιλογής δημιουργώντας κατάλληλες λογικές συνθήκες



#### Δραστηριότητα 1: Να πάρω μπουφάν ή γυαλιά;

Η ώρα είναι 10.00 το πρωί και αποφασίζεις να βγεις μία βόλτα. Στο μυαλό σου έχεις τη συμβουλή της μαμάς σου "Μπουφάν να πάρεις", αλλά και της γιαγιάς σου "Γυαλιά ηλίου να πάρεις, αλλιώς θα κάνεις ρυτίδες ". Έτσι λοιπόν προβλέπεις και κοιτάς τον καιρό πριν φύγεις από το σπίτι:



```
1. sunshine = input ( " Έχει ηλιοφάνεια; ΝΑΙ ή ΟΧΙ " )
2. temp = float (input ( "Πόσους βαθμούς έχει έξω " ))
3. if sunshine == "ΝΑΙ":
4.     print("Πάρε γυαλιά")
5. if temp <= 18.00:
6.     print ( " Πάρε μπουφάν " )
```

**Ερώτηση 1:** Τι πιστεύεις ότι κάνει το παραπάνω κομμάτι κώδικα;

Εκτέλεσε το παραπάνω κομμάτι κώδικα. Μετέφερε με προσοχή τα κενά που υπάρχουν! Σαν εισόδους βάλε : ΝΑΙ κ' 17, ΟΧΙ κ' 16, ΝΑΙ κ' 20, ΟΧΙ κ' 30.

**Ερώτηση 2:** Τι τιμές είχαν οι μεταβλητές sunshine και temp στην εκάστοτε εκτέλεση όταν εμφανίστηκαν τα μηνύματα : "Πάρε γυαλιά" και "Πάρε μπουφάν";

**Ερώτηση 3:** Ποιο πρέπει είναι το εύρος τιμών των μεταβλητών temp και sunshine ώστε να εμφανίζονται και τα δύο μηνύματα και ποιο για να μην εμφανίζεται κανένα;



*Η if συνοδεύεται από μία συνθήκη η οποία ελέγχεται κατά την εκτέλεση του προγράμματος αν είναι αληθής (True) ή ψευδής (False). Οι εντολές που ακολουθούν την if είναι στοιχισμένες δεξιά από αυτή. Για να το πετύχεις αυτό απλά εισάγεις κενά πριν από τις εντολές. Αν η συνθήκη είναι αληθής τότε θα εκτελεστούν οι εντολές κάτω και δεξιά από την if!*



Το == χρησιμοποιείται για να ελέγξει αν δυο τιμές είναι ίσες μεταξύ τους και το <= αν μία τιμή είναι μικρότερη ή ίση από μία δεύτερη. Είναι δηλαδή τελεστές σύγκρισης τιμών. Στην συνέχεια θα μάθεις και άλλους τέτοιους τελεστές. Το sunshine == "ΝΑΙ" και temp<=18 είναι μια λογική έκφραση και το αποτέλεσμα είναι True ή False.

"Και αν δεν χρειάζομαι μπουφάν μήπως να πάρω μία ζακετούλα; "

```
1. sunshine = input ( " Έχει ηλιοφάνεια; ΝΑΙ ή ΟΧΙ " )
2. temp = float (input ( "Πόσους βαθμούς έχει έξω " ))
3. if sunshine == "ΝΑΙ":
4.     print ( "Πάρε γυαλιά" )
5. if temp <= 18.00:
6.     print ( " Πάρε μπουφάν " )
7. else:
8.     print ( "Πάρε ζακέτα" )
```

Μελέτησε τον παραπάνω κώδικα.

**Ερώτηση 4:** Σε ποια περίπτωση πιστεύεις θα εμφανιστεί το μήνυμα "Πάρε ζακέτα" ;

Εκτέλεσε τον κώδικα 4 φορές δίνοντας τις παρακάτω τιμές: ΝΑΙ κ' 17, ΟΧΙ κ' 16, ΝΑΙ κ' 20, ΟΧΙ κ' 30.

**Ερώτηση 5:** Τι τιμή είχε η μεταβλητή temp στην εκάστοτε εκτέλεση όταν εμφανίστηκε το μήνυμα "Πάρε ζακέτα";

**Ερώτηση 6:** Πειραματίσου και εισήγαγε διάφορες τιμές σχετικά με τη θερμοκρασία. Ποιο πιστεύεις είναι το εύρος τιμών που αν πάρει η μεταβλητή temp, θα εμφανιστεί το μήνυμα "Πάρε ζακέτα", και ποιο για το "Πάρε μπουφάν";



Οι εντολές που ακολουθούν την else είναι επίσης στοιχισμένες δεξιότερα και θα εκτελεστούν μόνο εφόσον η προηγούμενη συνθήκη του if είναι ψευδής. Η else δεν είναι υποχρεωτική σε μια δομή επιλογής αν δεν έχεις να εκτελέσεις συγκεκριμένες εντολές

## Δραστηριότητα 2: Να πολλαπλασιάσω ή να προσθέσω;

Θέλεις να δοκιμάσεις το καινούριο παιχνίδι τύχης 'Ρίξε και Κέρδισε'. Οι κανόνες είναι απλοί. Ανάλογα με το άθροισμα των δύο ζαριών παίρνεις και την αντίστοιχη αμοιβή σε χρήματα.

```
1. import random
2. # τυχαίες τιμές για τα δύο ζάρια
3. dice1 = random.randint(1,6)
4. dice2 = random.randint(1,6)
5. print ( " Έφερεις: ", dice1," ", dice2)
```

**Ερώτηση 1:** Εκτέλεσε τον παραπάνω κώδικα 2-3 φορές. Τι ζαριά ήρθε κάθε φορά;

**Ερώτηση 2:** Το αποτέλεσμα που θα φέρει η ρίψη ζαριού είναι κάτι το τυχαίο. Ποια εντολή ανέθεσε τυχαίες τιμές στις μεταβλητές dice1 και dice2;



Η συνάρτηση randint(low,high) παράγει τυχαίους ακέραιους εντός συγκεκριμένων ορίων που καθορίζεις και ανήκει στην βιβλιοθήκη random την οποία πρέπει να την εισάγεις με το import random.

Στο παραπάνω κομμάτι κώδικα πρόσθεσε τις παρακάτω εντολές:

```
6. sum=dice1+dice2
7. if sum<=9:
8.     money=sum+2
9. else:
10.    money=sum*2
11.print ( " Άθροισμα ζαριών: ", sum, "Αμοιβή: ", money)
```

**Ερώτηση 3:** Σύμφωνα με τις εντολές 1-10 πως πιστεύεις ότι υπολογίζεται η αμοιβή/χρήματα που θα πάρει ο παίχτης;

---

Εκτέλεσε το πρόγραμμα 2-3 φορές και παρατήρησε τις ζαριές και την τελική αμοιβή.

**Ερώτηση 4:** Ποιο είναι το εύρος τιμών της μεταβλητής `sum` ώστε η αμοιβή να είναι `sum+2` και ποιο αντίστοιχα για να είναι `sum*2`;

---

"Και αν αλλάξουν οι κανόνες;"

```
6. sum=dice1+dice2
7. if sum<=9 and sum>3:
8.     money=sum+2
9. else:
10.    money=sum*2
11.print ( " Άθροισμα ζαριών: ", sum, "Αμοιβή: ", money)
```

**Ερώτηση 5:** Ποια εντολή άλλαξε σε σχέση με τον προηγούμενο κώδικα;

---

Εκτέλεσε το καινούριο πρόγραμμα 2-3 φορές και παρατήρησε τις ζαριές και τις αντίστοιχες αμοιβές.

**Ερώτηση 6:** Ποιο είναι το εύρος των τιμών που πρέπει να πάρει η μεταβλητή `sum` ώστε να εκτελεστεί η εντολή: `money=sum+2`;

---



Το `and` είναι λογικός τελεστής ή αλλιώς είναι το "λογικό και". Η πρόταση: `sum<=9 and sum>3`, είναι αληθής όταν και οι δύο συνθήκες, `sum<=9` και `sum>3`, είναι αληθείς. Υπάρχει αντίστοιχα και το `or`, που είναι το "λογικό ή". Θα μάθεις στη συνέχεια γι αυτό.

Μέχρι στιγμής μελέτησες προβλήματα στα οποία **υπό κάποια συνθήκη έπρεπε να προχωρήσεις σε συγκεκριμένη εκτέλεση ενεργειών-εντολών**. Ή ακόμη έμπαινες στο σκεπτικό: **αν ισχύει αυτή η συνθήκη θα κάνω αυτό, αλλιώς θα κάνω το άλλο**.

Τι γίνεται όμως όταν οι **επιλογές είναι παραπάνω από 2**;

**Για παράδειγμα:**

Έστω ότι σκέφτεσαι έναν αριθμό και το λες σε έναν φίλο σου. Ο φίλος σου περιμένεις να σου απαντήσει αν ο αριθμός αυτός είναι αρνητικός, μηδέν ή θετικός.

**Για σκέψου λοιπόν:**

- Ο αριθμός αυτός γίνεται να ανήκει σε παραπάνω από μία κατηγορίες (αρνητικός, μηδέν, θετικός);
- Πόσες είναι οι κατηγορίες που πρέπει ελέγξεις για να βρεις που ανήκει ο αριθμός αυτός και άρα πόσες συνθήκες ελέγχου που πρέπει να σκεφτείς;
- Η σύνθετη επιλογή που έμαθες (if...else...) θα σε εξυπηρετούσε ώστε να φτιάξεις ένα πρόγραμμα που θα σου απαντούσε σε ποια κατηγορία ανήκει ο αριθμός που σκέφτηκες εξ αρχής;
- Μήπως θα σε εξυπηρετούσε μία δομή επιλογής που θα σου πρόσφερε την δυνατότητα παραπάνω από δύο πολλαπλών επιλογών και δημιουργίας συνθηκών ελέγχου;

### Δραστηριότητα 3: Ταινία ή Σειρά;

Οι διακοπές τελείωσαν και το καθημερινό διάβασμα ξεκίνησε ξανά. Είναι Πέμπτη και αν και έχεις πολύ διάβασμα για αύριο, θες να δεις την αγαπημένη σου ταινία που προβάλλεται στην τηλεόραση στις 20.00. Όμως στις 21.00 προβάλλεται το επεισόδιο της σειράς που παρακολουθείς αυτό τον καιρό. Για να ανοίξεις όμως στην τηλεόραση πρέπει να πάρεις άδεια από την μηχανή ελέγχου που είναι εγκατεστημένη.

```
1.hour=float( input ( "Τι ώρα είναι;" ))
2.if hour <= 20.00:
3.    print ("Μπορείς να δεις την ταινία ή την σειρά που επιθυμείς")
4.    print ( "Ευτυχώς τελείωσες νωρίς! " )
5.else:
6.    print ( "Δεν πειράζει...Σκέψου ότι αύριο είναι Παρασκευή..." )
7.print("Είσαι σίγουρος ότι έβαλες τη σωστή ώρα; ")
```

**Ερώτηση 7:** Τι πιστεύεις ότι κάνει το παραπάνω κομμάτι κώδικα;

**Ερώτηση 8:** Εκτέλεσε τον κώδικα 2 φορές. Μετέφερε με προσοχή τα κενά που υπάρχουν. Ως είσοδο από το πληκτρολόγιο για την ώρα βάλε το 18.10 και μετά το 21.00. Ποια μηνύματα εμφανίστηκαν κάθε φορά;

**Ερώτηση 9:** Πειραμάτισου με το παραπάνω κώδικα εισάγοντας διαφορετικές τιμές εισόδου κάθε φορά. Ποιο είναι το εύρος των τιμών που πρέπει να πάρει η μεταβλητή hour για να εκτυπωθεί το μήνυμα "Μπορείς να δεις την ταινία ή την σειρά που επιθυμείς" και ποιο για να εκτυπωθεί αντίστοιχα το "Δεν πειράζει...Σκέψου ότι αύριο είναι Παρασκευή...";

**Ερώτηση 9:** Αν τελειώσεις με το διάβασμα στις 8.30, προλαβαίνεις να δεις την σειρά στις 21.00 σε περίπτωση που δεν χρειάζεσαι έγκριση από το μηχάνημα ελέγχου; Αν το μηχάνημα ελέγχου εκτελέσει τον παραπάνω κώδικα θα στο επιτρέψει;

Η μηχανή ελέγχου λοιπόν αναβαθμίζεται ώστε να σου επιτρέπει να δεις την σειρά αν χάσεις την αρχή της ταινίας.

```
1.hour=float( input ( "Τι ώρα είναι;" ))
2.if hour <= 20.00:
3.  print("Μπορείς να δεις την ταινία ή την σειρά που επιθυμείς")
4.  print("Ευτυχώς τελείωσες νωρίς! ")
5.elif hour <= 21.00:
6.  print("Μπορείς να δεις μόνο την σειρά!")
7.else:
8.  print ( "Δεν πειράζει...Σκέψου ότι αύριο είναι Παρασκευή..." )
9.print("Είσαι σίγουρος ότι έβαλες τη σωστή ώρα; ")
```

**Ερώτηση 10:** Αν ως είσοδος για την ώρα δοθεί το 20.40, ποια μηνύματα πιστεύεις θα εμφανιστούν στην οθόνη ;

---

**Ερώτηση 11:** Εκτέλεσε τον κώδικα 3 φορές και εισήγαγε ως τιμές για την ώρα το 18.20, το 22.00 και το 20.40. Ποιο ήταν το εκάστοτε αποτέλεσμα/μηνύματα που εμφανίστηκαν;

---

**Ερώτηση 12:** Ποια πιστεύεις ότι είναι η λειτουργία της εντολής elif;

---

**Ερώτηση 13:** Πειραματίσου βάζοντας διάφορες τιμές εισόδου για την ώρα. Ποιο είναι το εύρος των τιμών που πρέπει να πάρει η μεταβλητή hour ώστε να τυπωθεί το μήνυμα "Μπορείς να δεις μόνο την σειρά";

---

**Ερώτηση 14:** Όταν hour=18.20, αφού το 18.20 είναι μικρότερο του 21.00, δηλαδή η συνθήκη  $18.20 \leq 21.00$  είναι αληθής, γιατί πιστεύεις στην αντίστοιχη εκτέλεση εμφανίστηκαν τα μηνύματα κάτω από το if και όχι τα μηνύματα κάτω από το elif;

---



*Το elif είναι το else if. Μετά την if μπορείς να προσθέσεις όσες elif θες. Η elif συνοδεύεται πάντα από μια συνθήκη. Κάτω από το elif οι εντολές πρέπει να είναι στοιχισμένες δεξιάτερα. Αυτές και μόνο αυτές οι εντολές θα εκτελεστούν αν η αντίστοιχη συνθήκη είναι αληθής.*

Επειδή όμως είναι πιθανόν να μπεις στον πειρασμό να ανοίξεις οποιαδήποτε ώρα την τηλεόραση, η μηχανή ελέγχου προσθέτει έναν ακόμη αυστηρό έλεγχο.

```
1.hour_now = float ( input ( "Τι ώρα είναι τώρα που τελείωσες το διάβασμα; " ))
2.if hour_now>=19.00 and hour_now<=23.00:
3.    print( "Μπορείς να δεις τηλεόραση μόνο 19.00-23.00" )
4.    if hour_now <= 20.00:
5.        print("Μπορείς να δεις την ταινία ή την σειρά που επιθυμείς")
6.        print("Ευτυχώς τελείωσες νωρίς! ")
7.    elif hour_now <= 21.00:
8.        print("Μπορείς να δεις μόνο την σειρά!")
9.    else:
10.        print ( " Έχασες και ταινία και σειρά. Τσέκαρε το blueChannel, έχει ντοκιμαντέρ . " )
11.else:
12.    print ( " Δεν μπορείς να δεις τηλεόραση αυτήν την ώρα!" )
```

**Ερώτηση 15:** Εκτέλεσε τον παραπάνω κώδικα και δώσε ως τιμές εισόδου για την ώρα το 18.10, 19.50, 20.40, 22.00 και το 23.01. Ποια μηνύματα εμφανίστηκαν κάθε φορά;

**Ερώτηση 16:** Ποια πιστεύεις είναι λειτουργία των εντολών 2 και 11; Παρατηρείς κάτι διαφορετικό στη δομή αυτή;

**Ερώτηση 17:** Ποιο είναι το εύρος των τιμών που πρέπει να δοθούν ως είσοδος για την ώρα για να εκτελεστεί η εντολή 12 και ποιο αντίστοιχα για να εκτελεστούν οι εντολές 3-10;

**Ερώτηση 18:** Ποιο είναι το εύρος των τιμών που πρέπει να δοθούν ως είσοδος για την ώρα για να εκτελεστούν οι εντολές 5-6, 8 και 10 αντίστοιχα;



Μία δομή επιλογής μπορεί να περιέχεται "μέσα" σε μία άλλη δομή επιλογής. Αυτή η δομή λέγεται εμφωλευμένη. Μπορείς να έχεις όσες δομές επιλογής θέλεις, απλές, σύνθετες ή πολλαπλές μέσα σε άλλες.

## Ασκήσεις:

**Άσκηση 1:** Συμπληρώστε κατάλληλα τα κενά στον παρακάτω κώδικα

- a = ..... ("Δώσε ακέραιο αριθμό: ")
- if ..... :
- print ("άρτιος")
- if ..... :
- print ( "περιττός" )

**Άσκηση 2:** Τι θα τυπώσει το παρακάτω πρόγραμμα;

- a=1
- b=2
- c=3
- if a+b==c or a>b and not c>a:
- print ("A")
- else:
- print ("B")

**Άσκηση 3:** Έστω πρόγραμμα το οποίο να ζητάει από τον χρήστη έναν αριθμό και στην συνέχεια να του εμφανίζει αν ο αριθμός αυτός είναι θετικός ή αρνητικός καθώς επίσης και αν ο αριθμός αυτός είναι διψήφιος ή όχι;

Κάποιος έγραψε τη παρακάτω λύση :

```
number=int(input( "Δώσε έναν ακέραιο αριθμό"))
if number<0:
    print("Ο αριθμός είναι αρνητικός")
elif number>0:
    print("Ο αριθμός είναι θετικός")
elif number>=10 and number <100:
    print("Ο αριθμός είναι διψήφιος")
elif number<10 or number>=100:
    print("Ο αριθμός δεν είναι διψήφιος")
```

Η παραπάνω λύση είναι....

- Σωστή
- Λανθασμένη

**Άσκηση 4:** Δίνεται το παρακάτω κομμάτι κώδικα :

1. fever=False
2. body\_temp=float(input("You use a body thermometer. What does it show?" ))
3. if .....
4.     fever=.....
5.     print("You have fever.")
6. elif.....:
7.     print("You have low-grade fever.")
8.     print("Drink some juice.")
9. elif.....:
10.    print("You have normel body temperature.")
11. else:
12.    print("You have hypothermia.")
13.    print("Get some rest.")
14. if.....:
15.    print("You have to take medication.")

Γράψε ξανά τον κώδικα συμπληρώνοντας κατάλληλα ότι λείπει στα κενά. Ο κώδικας πρέπει να εκτελεί την παρακάτω λειτουργία:

- Διαβάζει αρχικά την θερμοκρασία σώματος ενός ανθρώπου. Αν είναι μεγαλύτερη ή ίση του 38, τότε έχει πυρετό. Αν είναι ανάμεσα σε 37 και 38, έχει χαμηλό επιπέδου πυρετό. Αν είναι ανάμεσα σε 36.6 και 37, τότε είναι φυσιολογική.

## Προβλήματα

**Πρόβλημα 1:** Γράψε πρόγραμμα το οποίο να μετράει το άθροισμα τυχαίας ρίψης δύο ζαριών. **Αν** το άθροισμα είναι ίσο με 10 να εμφανίζει το μήνυμα "Κέρδισες". **Αν** το άθροισμα είναι ίσο με 7 να εμφανίζει το μήνυμα "Ξανά παίξε".

**Πρόβλημα 2:** Γράψε πρόγραμμα το οποίο θα ζητάει από τον χρήστη να εισάγει έναν ακέραιο αριθμό και θα υπολογίζει την απόλυτη τιμή του αριθμού αυτού. Στο τέλος θα εμφανίζει μήνυμα της μορφής "Η απόλυτη τιμή του -8.5 είναι το 8.5".

! Η απόλυτη τιμή ενός αριθμού δείχνει την απόσταση του αριθμού από το 0. Ο κανόνας για τον υπολογισμό της είναι ο εξής:

$$|α| = \begin{cases} α & \text{για } α \geq 0 \\ -α & \text{για } α < 0. \end{cases}$$

>Σκέψου προσεκτικά τι δομή επιλογής πρέπει να χρησιμοποιήσεις για να το λύσεις. Σύνθετη ή απλή; Μήπως γίνεται και με τις δύο;

**Πρόβλημα 3:** Γράψε πρόγραμμα που θα ζητάει από τον χρήστη να εισάγει 3 αριθμούς και στη συνέχεια θα υπολογίζει και θα εμφανίζει το μέγιστο αριθμό από αυτούς τους 3.

Τip! Διάβασε τους αριθμούς έναν – έναν και κράτα τον μεγαλύτερο κάθε φορά. Δηλαδή, έλεγξε αν ο νέος αριθμός που ήρθε είναι μεγαλύτερος και αν είναι, κράτησε τον ως τον max.

**Πρόβλημα 4:** Να γραφεί πρόγραμμα το οποίο διαβάζει δύο αριθμούς κινητής υποδιαστολής και έναν αριθμητικό τελεστή ('+' ή '-' ή '\*' ή '/') και να υπολογίζει και να εμφανίζει το αποτέλεσμα της αντίστοιχης πράξης.

**Πρόβλημα 5:** Να ξαναγράψεις τον παρακάτω κώδικα, έτσι ώστε να μην έχει εμφωλευμένη δομή επιλογής.

(Υπενθύμιση : οι λογικοί τελεστές βοηθούν στην απλοποίηση εμφωλευμένων δομών επιλογής).

```

1 x=int(input("Insert an integer"))
2 if x<=20:
3     if x>=16:
4         print("Between 16 and 20")
5     elif x>=10:
6         print("Between 10 and 15")
7     else:
8         print("Bellow 10")
9 else:
10    print("Number above 20")

```



**Πρόβλημα 6:** Έχουμε το εξής πρόβλημα το οποίο ζητάει από τον χρήστη να εισάγει έναν ακέραιο αριθμό και να ελέγχει αν ο αριθμός αυτός είναι αρνητικός, θετικός μονοψήφιος, θετικός διψήφιος ή θετικός πολυψήφιος εμφανίζοντας το κατάλληλο μήνυμα.

Κάποιος λοιπόν έγραψε τις εξής δύο λύσεις:

#### Λύση με Πολλαπλή Επιλογή

```
number = input ("Δώσε έναν ακέραιο αριθμό: ")
if number < 0 :
    print ("Αρνητικός ")
elif number < 10 :
    print ("Θετικός Μονοψήφιος ")
elif number < 100:
    print ("Θετικό Διψήφιος ")
elif number >= 100:
    print ("Θετικός Πολυψήφιος ")
print ("Αριθμός ")
```

#### Λύση με Απλή Επιλογή

```
number = input ("Δώσε έναν ακέραιο αριθμό: ")
if number < 0 :
    print ("Αρνητικός ")
if number >= 0 and number < 10 :
    print ("Θετικός Μονοψήφιος ")
if number >= 10 and number < 100:
    print ("Θετικό Διψήφιος ")
if number >= 100:
    print ("Θετικό Πολυψήφιος ")
print ("Αριθμός ")
```

**Ερώτηση 1:** Τα δύο παραπάνω προγράμματα είναι ισοδύναμα; Δηλαδή αν κάποιος εισάγει διάφορους αριθμούς τα παραπάνω προγράμματα θα επιφέρουν τα ίδια αποτελέσματα;

---

**Ερώτηση 2:** Αν ένας χρήστης ως είσοδο δώσει τους αριθμούς -2, 1000 και 9, πόσες συνθήκες θα ελέγξει ο υπολογιστής για το πρόγραμμα με την πολλαπλή δομή επιλογής και για το πρόγραμμα με τις 4 απλές δομές επιλογής για κάθε εκτέλεση (1, 2, 3 ή 4 ;); Η απάντηση αφορά και τα δύο προγράμματα για κάθε είσοδο τιμής.

---

**Ερώτηση 3:** Ένα πρόγραμμα για να χαρακτηριστεί ως "καλό" πρέπει να αποτρέπει τον υπολογιστή από το να κάνει περιττούς ελέγχους και να σπαταλά χρόνο. Άρα ποια από τα δύο προγράμματα/λύσεις πιστεύετε ότι είναι καλύτερο; Ή ποιο από τα δύο προγράμματα θα κάνει λιγότερους ελέγχους και ενδεχομένως να τελειώσει πιο γρήγορα;

---

**Ερώτηση 4:** Σύμφωνα με τις παραπάνω λύσεις, ένα πρόβλημα που απαιτεί πολλούς ελέγχους και που λύνεται με πολλαπλή επιλογή μπορεί να λυθεί και με πολλές απλές δομές επιλογής. Θέλοντας όμως το πρόγραμμά μας να είναι όσο "καλύτερο" γίνεται, πότε ενδείκνυται να χρησιμοποιούμε την πολλαπλή επιλογή και πότε την απλή;

#### Προβλήματα επιλογής :

**Πρόβλημα:** Να γράψεις ένα πρόγραμμα που να ζητά από τον χρήστη 4 αριθμούς. Να υπολογίζει τον μέσο όρο τους και να ελέγχει αν αυτός είναι μεγαλύτερος από 15. **Αν** ισχύει αυτό να εμφανίζεται το μήνυμα 'SUCCESS' **διαφορετικά** το μήνυμα 'FAIL'.

**Πρόβλημα:** Να γράψεις ένα πρόγραμμα που διαβάζει τις ώρες, που δούλεψε κάποιος αυτήν τη βδομάδα και υπολογίζει και εκτυπώνει την αμοιβή του. Για την αμοιβή ισχύουν τα εξής :

- Μέχρι τις 20 ώρες ορίζεται ως 4 ευρώ την ώρα
- Μετά τις 20 ώρες ο εργαζόμενος αμείβεται 6 ευρώ για κάθε επιπλέον ώρα

**Πρόβλημα:** Γράψε πρόγραμμα που να κάνει αφαίρεση 2 αριθμών ώστε να προκύπτει πάντα θετικό αποτέλεσμα. Για να επιτευχθεί αυτό, γίνεται έλεγχος για το ποιος από τους 2 αριθμούς είναι μεγαλύτερος. Αν οι 2 αριθμοί είναι ίδιοι, το αποτέλεσμα είναι μηδέν.

**Πρόβλημα :** Γράψε πρόγραμμα που θα ζητάει από τον χρήστη να εισάγει έναν φυσικό αριθμό. Αν ο αριθμός αυτός δεν είναι φυσικός θα εμφανίζει κατάλληλο μήνυμα λάθους. Αν όμως ανήκει στους φυσικούς, όπως είναι το αναμενόμενο, θα βρίσκει αν ο αριθμός είναι άρτιος ή περιττός, εκτυπώνοντας κατάλληλο μήνυμα.

>Οι φυσικοί αριθμοί είναι όλοι οι θετικοί ακέραιοι, δηλ. 0,1,2,3,.....

**Πρόβλημα:** Να γραφεί πρόγραμμα που να διαβάζει το βαθμό ενός μαθητή και να υπολογίζει την αντίστοιχη αξιολόγηση του με βάση το βαθμό του και σύμφωνα με τον πίνακα:

Βαθμός	Αποτέλεσμα
<9.5	απορρίπτεται
[9.5-13.5)	Μέτρια
[13.5-15.5)	Καλά
[15.5-17.5)	Πολύ καλά
17.5+	Άριστα

**Πρόβλημα:** Γράψε πρόγραμμα το οποίο να ζητάει από το χρήστη να εισάγει έναν αριθμό και να του εμφανίζει αν ο αριθμός αυτός είναι αρνητικός ή θετικός. Αν είναι θετικός να του εμφανίζει αν είναι μονοψήφιος, διψήφιος ή πολυψήφιος.

Κάποιος έδωσε την εξής λύση:

1. `if number < 0 :`
2. `print ("Αρνητικός ")`
3. `elif number<10`
4. `print("Μονοψήφιος ")`
5. `elif number < 100:`
6. `print ("Διψήφιος ")`
7. `else:`
8. `print ("Πολυψήφιος ")`
9. `print ("Αριθμός ")`

Γράψε μία λύση που να λύνει το παραπάνω πρόβλημα με εμφωλευμένη δομή επιλογής.

Tip: Σκέψου τι ιδιότητα έχουν όλοι οι υπόλοιποι αριθμοί που δεν επαληθεύουν την πρώτη συνθήκη και θα μπουν σε κάποιο από τα επόμενα μπλοκ εντολών.

>Ποιος τρόπος θεωρείς πως είναι ο καλύτερος;

### Δραστηριότητα Διακλάδωσης

**Πρόβλημα:** Να γράψεις ένα πρόγραμμα που διαβάζει έναν ακέραιο. Θα ελέγχει αν ο αριθμός αυτός είναι διψήφιος ή όχι. Αν δεν είναι διψήφιος θα εμφανίζει κατάλληλο μήνυμα λάθους. Ενώ αν είναι διψήφιος θα ελέγχει αν είναι θετικός ή αρνητικός και εκτυπώνει το αντίστοιχο μήνυμα. Αν είναι θετικός θα κάνει τον κατάλληλο υπολογισμό και εμφανίζει ένα από τα παρακάτω μηνύματα :

- "Both digits are even"
- "Both digits are odd"
- "First digit is even and second digit is odd"
- "First digit is odd and second digit is even"

### Δραστηριότητα Διακλάδωσης

**Πρόβλημα :** Γράψε πρόγραμμα που να υπολογίζει την τιμή εισιτηρίου που πρέπει να πληρώσει κάποιος ενήλικος για μία θεατρική παιδική παράσταση. Αν ο ενδιαφερόμενος **δεν έχει παιδιά** η τιμή του εισιτηρίου είναι 5 ευρώ. Αν όμως **έχει παιδιά** τότε η τιμή αυτή διαμορφώνεται ανάλογα με το αριθμό των παιδιών που έχει φέρει μαζί του για την παράσταση. Αν έχει **1 ή 2** παιδιά τότε στην τιμή προστίθενται 2 ευρώ επιπλέον για το κάθε παιδί. Αν έχει από **3 και πάνω** τότε πάλι προστίθενται 2 ευρώ στην αρχική τιμή για το κάθε παιδί αλλά δικαιούται επιπλέον έκπτωση 20% της τελικής τιμής που θα προκύψει. Αρχικά το πρόγραμμα πρέπει να ρωτάει τον χρήστη αν έχει παιδιά ή όχι και αν έχει να τον ρωτάει στην συνέχεια πόσα έχει. Στο τέλος θα του εμφανίζει την συνολική τιμή που πρέπει να πληρώσει.

### Δραστηριότητα Επιλογής

Επίλεξε τη σωστή Απάντηση στις παρακάτω ερωτήσεις κατανόησης χωρίς να εκτελέσεις τις εντολές. Μπορείς να τις εκτελέσεις στην συνέχεια για να επαληθεύσεις το αποτέλεσμα:

**Ερώτηση 1:** Έστω ότι κάποιος εκτελεί το παρακάτω πρόγραμμα και ως είσοδο για την ηλικία του βάλει το '17' και για το αν είναι φοιτητής ή όχι το 'ναι' .

1. `age=int(input(" Πόσο χρονών είσαι : "))`
2. `price_of_ticket=10`
3. `if age<18:`
4. `price_of_ticket= price_of_ticket // 2`
5. `student=input("Είσαι φοιτητής; :")`
6. `if student=="ΝΑΙ" or student=="ναι":`
7. `print("Κερδίζεις άλλο ένα εισιτήριο δωρεάν!!")`
8. `print("Η τιμή του εισιτηρίου σου είναι: ", price_of_ticket, "ευρώ")`

Ποια μηνύματα θα εκτυπωθούν στην οθόνη;

1	Κερδίζεις άλλο ένα εισιτήριο δωρεάν!! Η τιμή του εισιτηρίου σου είναι: 10 ευρώ
2	Η τιμή του εισιτηρίου σου είναι: 5 ευρώ
3	Κερδίζεις άλλο ένα εισιτήριο δωρεάν!! Η τιμή του εισιτηρίου σου είναι: 5 ευρώ

**Ερώτηση 2:** Το παρακάτω πρόγραμμα υπολογίζει τον μέσο όρο 3 βαθμών που ζητά από τον χρήστη και εκτυπώνει κατάλληλο μήνυμα επιτυχίας ή αποτυχίας ανάλογα με το αν ο τελικός βαθμός είναι προσβάσιμος ή όχι.

```

1. a = int(input("Δώσε βαθμό 1: "))
2. b = int(input("Δώσε βαθμό 2: "))
3. c = int(input("Δώσε βαθμό 3: "))
4. average = (a+b+c)/4
5. if average >= 10:
6.     result = 'SUCCESS'
7. else:
8.     result = 'FAIL'
9. print(result)

```

- Σωστό
- Λάθος

**Ερώτηση 3:** Έστω ότι κάποιος εκτελεί το παρακάτω κομμάτι κώδικα

```

1. meal_cost=6
2. discount=False
3. answer=input("Είσαι μαθητής;")
4. if answer=="Ναι":
5.     discount=True
6. date=input("Τι μέρα είναι;")
7. if date=="Δευτέρα" and discount:
8.     meal_cost=meal_cost-2
9. if (date=="Σάββατο" or date=="Κυριακή") and (not discount):
10.    meal_cost=meal_cost+2
11. print("Το γεύμα σας κοστίζει ",meal_cost,"ευρώ")

```

Επίλεξε ποια από τις παρακάτω προτάσεις είναι σωστή.

1. Αν σε μία εκτέλεση ο χρήστης ως είσοδο πληκτρολογήσει τις λέξεις: 'Ναι' και 'Δευτέρα', το μήνυμα που θα του εμφανιστεί στην οθόνη είναι το: 'Το γεύμα σας κοστίζει 6 ευρώ'.
2. Αν σε μία εκτέλεση ο χρήστης πληκτρολογήσει ως είσοδο τις λέξεις: 'Όχι' και 'Κυριακή', το μήνυμα που θα του εμφανιστεί στην οθόνη είναι το: 'Το γεύμα σας κοστίζει 6 ευρώ'.
3. Αν σε μία εκτέλεση ο χρήστης ως είσοδο πληκτρολογήσει τις λέξεις: 'Όχι' και 'Πέμπτη', το μήνυμα που θα του εμφανιστεί στην οθόνη είναι το: 'Το γεύμα σας κοστίζει 6 ευρώ'.

## Υποστηρικτική Δραστηριότητα

### Η απλή και η σύνθετη δομή επιλογής

Με τον όρο δομή επιλογής εννοούμε **ένα σύνολο εντολών που θα εκτελεστούν** αν ισχύει μια υπόθεση, ή με άλλα λόγια **αν μια συνθήκη είναι αληθής**. Σκέψου το πολύ απλό παράδειγμα από την καθημερινότητά σου: "Αν έξω έχει ηλιοφάνεια τότε θα πάρω γυαλιά ηλίου!". Αυτό το σενάριο-αλγόριθμο μπορούμε να το μεταφέρουμε και στις γλώσσες προγραμματισμού.

#### Γενική Μορφή Απλής δομής επιλογής:

```
if <συνθήκη>:  
    μπλοκ_εντολών  
εντολές
```

#### Λειτουργία:

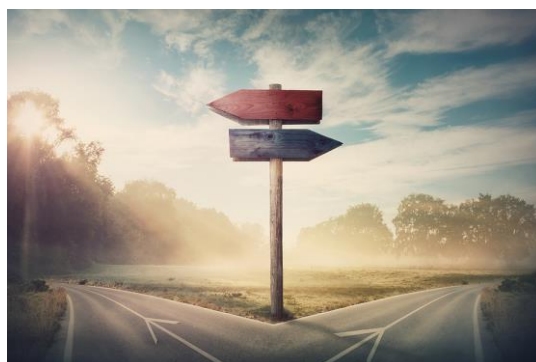
- Αν η <συνθήκη> είναι **True** εκτελείται το **μπλοκ\_εντολών** και στη συνέχεια οι **εντολές**.
- Αν είναι **False**, το πρόγραμμα αγνοεί το **μπλοκ\_εντολών** και εκτελούνται κατευθείαν οι **εντολές**.

#### Γενική Μορφή Σύνθετης δομής επιλογής:

```
if <συνθήκη>:  
    μπλοκ_εντολών1  
else:  
    μπλοκ_εντολών2  
εντολές
```

#### Λειτουργία:

- Αν η <συνθήκη> είναι **True** εκτελείται το **μπλοκ\_εντολών1** και στην συνέχεια οι **εντολές**
- Αν η <συνθήκη> είναι **False** εκτελείται το **μπλοκ\_εντολών2** και στη συνέχεια εκτελούνται οι **εντολές**



### Η πολλαπλή δομή επιλογής

Η πολλαπλή δομή επιλογής χρησιμοποιείται όταν σε ένα σενάριο/πρόγραμμα υπάρχουν από **2 και πάνω περιπτώσεις/συνθήκες** που πρέπει να εξεταστούν/ελεγχθούν ώστε να εκτελεστούν στην συνέχεια τα αντίστοιχα βήματα. Η πολλαπλή επιλογή στην Python συντάσσεται με τον εξής τρόπο:

## Γενική Μορφή Πολλαπλής Επιλογής

```
if <συνθήκη1>:
    μπλοκ_εντολών1
elif <συνθήκη2>:
    μπλοκ_εντολών2
.....
elif <συνθήκη N>:
    μπλοκ_εντολώνN
else:
    μπλοκ_εντολών_else
εντολές
```

**Παράδειγμα:** Πρόγραμμα το οποίο παίρνει ως είσοδο έναν αριθμό και εμφανίζει αν ο αριθμός αυτός είναι θετικός, αρνητικός ή μηδέν.

```
# θετικός ή αρνητικός ή μηδέν
a = int( input ( "Δώσε αριθμό: " ))
if a > 0:
    print ("θετικός")
elif a < 0:
    print ("αρνητικός")
else:
    print ("μηδέν")
```

>Ως **συνθήκη1, 2,..N** μπορεί να θεωρηθεί οποιαδήποτε λογική έκφραση.

>Ως **μπλοκ\_εντολών** θεωρείται ένα οποιοδήποτε σύνολο εντολών.

### Λειτουργία:

- Αρχικά ελέγχεται η **<συνθήκη1>**. Αν η **<συνθήκη1>** είναι **True**, δηλαδή αληθής, τότε εκτελείται το **μπλοκ\_εντολών1**.
- Αν η **<συνθήκη1>** είναι **False**, δηλαδή ψευδής, ελέγχεται η **<συνθήκη2>**. Αν η **<συνθήκη2>** είναι **True**, τότε εκτελείται το **μπλοκ\_εντολών2**.
- Αν η **<συνθήκη2>** είναι ψευδής, ελέγχεται η **<συνθήκη3>**, κ.ο.κ. μέχρι κάποια συνθήκη να είναι αληθής, οπότε και εκτελείται η αντίστοιχη ομάδα εντολών.
- Αν καμία συνθήκη από τις παραπάνω δεν είναι αληθής, τότε εκτελείται η **ομάδα\_εντολών\_else**.
- Σε κάθε περίπτωση, μετά την ολοκλήρωση των πιο πάνω ελέγχων, εκτελούνται οι **εντολές** που υπάρχουν μετά τη δομή επιλογής.

### Παρατηρήσεις:

- Το **elif** σημαίνει **else if**. **Μετά από κάθε elif, ακολουθείται πάντα μία συνθήκη. Οι εντολές μετά την elif είναι στοιχισμένες δεξιότερα** και θα εκτελεστούν αν η αντίστοιχη συνθήκη είναι αληθής και όλες οι προηγούμενες συνθήκες που υπάρχουν πιο πάνω είναι ψευδείς.
- Σε μια πολλαπλή δομή επιλογής, **μετά από μία if μπορούν να προστεθούν όσες elif είναι απαραίτητες**.
- Όπως και στην σύνθετη δομή επιλογής, **οι εντολές που ακολουθούν την else είναι επίσης στοιχισμένες δεξιότερα** και θα εκτελεστούν μόνο εφόσον οι προηγούμενες συνθήκες είναι όλες ψευδείς.
- Το **σύμβολο :** **μετά τις συνθήκες και την else είναι απαραίτητο** για την σωστή σύνταξη και εκτέλεση του προγράμματος
- **Μια πολλαπλή δομή επιλογής δεν είναι απαραίτητο να τελειώνει με την else.** Αυτό που κάνει μια δομή επιλογής να είναι πολλαπλή είναι η ύπαρξη μίας ή περισσότερων **elif**.

- Σε μια πολλαπλή δομή επιλογής **κάθε φορά ικανοποιείται μόνο μια λογική συνθήκη** (ανεξάρτητα από το πόσες έχουμε). **Μόλις βρεθεί η λογική συνθήκη η οποία είναι αληθής, εκτελείται η ομάδα εντολών που ανήκει στη συνθήκη αυτή και στην συνέχεια η δομή επιλογής τερματίζεται.**



### Η εμφωλευμένη δομή επιλογής

Η **εμφωλευμένη δομή επιλογής** είναι ουσιαστικά **μία δομή επιλογής η οποία βρίσκεται μέσα σε μία άλλη**. Η εμφωλευμένη δομή επιλογής μπορεί να είναι απλή, σύνθετη ή πολλαπλή και μπορεί να περιέχεται μέσα σε μία απλή, σύνθετη ή πολλαπλή εξωτερική δομή επιλογής. Ό,τι ξέρεις ήδη για τις διάφορες δομές επιλογής ισχύουν και εδώ με τη διαφορά μόνο ότι η μία βρίσκεται μέσα στη άλλη.

### Παράδειγμα

```
age= int ( input ( "Πόσο χρονών είσαι: " ) )
if age>=18:
    answer= input ( "Είσαι φοιτητής; ΝΑΙ ή ΟΧΙ : " )
    if answer=="ΝΑΙ":
        ticket=12
    else:
        ticket=15
elif age<18:
    ticket=8
print(" Το εισιτήριο κοστίζει ", ticket, " ευρώ")
```

Ο παραπάνω κώδικας αναφέρεται στο εξής πρόβλημα:

Ζητάμε από το χρήστη να εισάγει την ηλικία του ώστε να του εμφανίσουμε την τιμή του εισιτηρίου του. Η τιμή εξαρτάται από τα εξής:

1. Αν είναι ανήλικος η τιμή είναι 8 ευρώ, δηλαδή μειωμένο.
2. Αν είναι ενήλικος, εξαρτάται από το αν είναι φοιτητής ή όχι.
  - a) Αν είναι φοιτητής η τιμή είναι 12 ευρώ, καθώς έχει έκπτωση
  - b) Ενώ αν δεν είναι φοιτητής, η τιμή είναι 15 ευρώ, δηλαδή ολόκληρο.

Από τη περιγραφή του προβλήματος, καταλαβαίνει κανείς, ότι ο ενδιαφερόμενος θα ερωτηθεί αν είναι φοιτητής ή όχι μόνο στην περίπτωση που είναι ενήλικος και στην συνέχεια ο υπολογιστής θα υπολογίσει την τιμή του εισιτηρίου. Το παραπάνω πρόβλημα λοιπόν, όπως και πολλά άλλα, απαιτεί εμφωλευμένη δομή επιλογής για να υλοποιηθεί αποτελεσματικά.

**Ωστόσο πολλά προβλήματα που στη λύση τους περιέχουν εμφωλευμένη δομή επιλογής, μπορούν αντίστοιχα να λυθούν και χωρίς αυτή εισάγοντας σύνθετες λογικές εκφράσεις στην συνθήκες ελέγχου με τη χρήση των λογικών τελεστών or, and και not.**

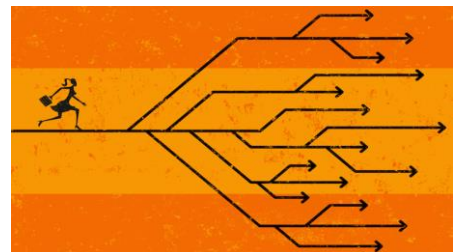
### Λύση με εμφωλευμένη δομή επιλογής

```
if 0<x:
    if x<10:
        print( "x is a positive single-digit
number" )
```

### Λύση χωρίς εμφωλευμένη δομή επιλογής

```
if 0<x and x<10:
    print( "x is a positive single-digit
number" )
```

- Μην ξεχνάς τις εσοχές μετά τα if,elif,else. Ειδικά αν έχεις μία εμφωλευμένη δομή επιλογής πρέπει να είσαι ακόμη πιο προσεκτικός. Από τις εσοχές ο διερμηνευτής καταλαβαίνει σε ποιο if,elif και else αντιστοιχεί κάθε μπλοκ εντολών.



### Λογικές Εκφράσεις

Λογική ονομάζεται μία έκφραση που λαμβάνει αποκλειστικά δύο τιμές, αληθής και ψευδής, δηλαδή **True** και **False**. Οι δύο αυτές τιμές ανήκουν στον τύπο δεδομένων που ονομάζεται **Boolean** ή πιο σύντομα **bool**. Για τη σύνταξη μιας λογικής έκφρασης ή συνθήκης χρησιμοποιούνται σταθερές, μεταβλητές, συναρτήσεις, αριθμητικές παραστάσεις, συγκριτικοί και λογικοί τελεστές, καθώς και παρενθέσεις.

```
>>> type(True)
<class 'bool'>
>>> type(False)
<class 'bool'>
```

**answer="Yes"**

**13>=12 and answer=="Yes" ⇒ True**

Υπάρχουν δύο είδη λογικών εκφράσεων :

- οι απλές λογικές εκφράσεις π.χ. **2!=4**
- οι σύνθετες λογικές εκφράσεις π.χ. **3<5 or age==17**

Στις απλές λογικές εκφράσεις γίνεται σύγκριση της τιμής μίας έκφρασης, που βρίσκεται αριστερά από τον συγκριτικό τελεστή με την τιμή μιας άλλης έκφρασης που βρίσκεται δεξιά.

### Σχεσιακοί Τελεστές

Ο συγκριτικός τελεστής "==" συγκρίνει δύο εκφράσεις και αν είναι ίσες τότε το αποτέλεσμα είναι True, αλλιώς το αποτέλεσμα είναι False.

```
>>> 5 == 5
True
>>> 5 == 6
False
```



Συνηθισμένο λάθος αποτελεί η χρήση μονού συμβόλου του ίσων (=) αντί για δύο (==). Προσοχή, καθώς το πρώτο αποτελεί τελεστή ανάθεσης, σε αντίθεση με το δεύτερο που είναι συγκριτικός τελεστής.

Οι σχεσιακοί τελεστές είναι οι εξής :

### Τελεστές ελέγχου ισότητας

$x==y$  # Αν το x είναι ίσο με το y, τότε η συνθήκη αληθεύει

$x!=y$  # Αν το x δεν είναι ίσο με το y, τότε η συνθήκη αληθεύει

### Τελεστές σύγκρισης

$x > y$  # Αν το x είναι μεγαλύτερο από το y, τότε η συνθήκη αληθεύει

$x < y$  # Αν το x είναι μικρότερο από το y, τότε η συνθήκη αληθεύει

$x >= y$  # Αν το x είναι μεγαλύτερο ή ίσο με το y, τότε η συνθήκη αληθεύει

$x <= y$  # Αν το x είναι μικρότερο ή ίσο με το y, τότε η συνθήκη αληθεύει

### Λογικοί Τελεστές

Σε πολλά προβλήματα οι επιλογές δεν αρκεί να γίνονται με απλές λογικές εκφράσεις, αλλά χρειάζεται να συνδυαστούν μία ή περισσότερες λογικές παραστάσεις, δημιουργώντας έτσι μια σύνθετη λογική έκφραση. Αυτό επιτυγχάνεται με τη χρήση των τριών λογικών τελεστών : **and**, **or** , **not**

- Η παράσταση “  $x > 0$  and  $x < 10$  ” είναι αληθής αν και μόνο αν το x είναι μεγαλύτερο του μηδενός και μικρότερο του 10. Σε αντίθετη περίπτωση είναι ψευδής.
- Η παράσταση “  $n\%2 == 0$  or  $n\%3 == 0$  ” είναι αληθής αν οποιαδήποτε από τις δύο εκφράσεις είναι αληθείς. Στο συγκεκριμένο παράδειγμα είναι αληθής αν το n διαιρείται ακριβώς με το 2 ή με το 3.
- Τέλος ο τελεστής not αντιστρέφει μια λογική έκφραση. Δηλαδή η έκφραση “ **not ( x > y)** ” είναι αληθής αν η εσωτερική έκφραση είναι ψευδής. Δηλαδή είναι αληθής αν το x είναι μικρότερο ή ίσο με το y.

Παρατήρηση : ένας οποιοσδήποτε μη μηδενικός αριθμός θεωρείται αληθής σε κάποια λογική έκφραση

```
>>> 17 and True
True
```

**Προτεραιότητα Πράξεων:** Όταν σε μία πρόταση υπάρχουν πολλοί λογικοί τελεστές, η προτεραιότητα που εκτελείται είναι η εξής: **not**, **and**, **or**.

### Μερικά παραδείγματα:

$3 + 2 < 7 + 3 \Rightarrow 5 < 10 \Rightarrow \text{True}$

$3 > 2$  and  $7 <= 3 \Rightarrow \text{True}$  and  $\text{False} \Rightarrow \text{False}$

$10 < 20$  and  $30 < 20$  or  $30 < 40 \Rightarrow \text{True}$  and  $\text{False}$  or  $\text{True} \Rightarrow \text{False}$  or  $\text{True} \Rightarrow \text{True}$

not  $10 < 20$  or  $30 < 20 \Rightarrow$  not  $\text{True}$  or  $\text{False} \Rightarrow \text{False}$  or  $\text{False} \Rightarrow \text{False}$

## Μάθημα 4: Οι συμβολοσειρές στην Python

Στα προηγούμενα μαθήματα έμαθες βασικά πράγματα σχετικά με τις συμβολοσειρές στην Python. Έμαθες πως να ορίζεις



συμβολοσειρές, πως να αναθέτεις συμβολοσειρές σε μεταβλητές καθώς και πώς να ενώνεις παραπάνω από δύο συμβολοσειρές. Επίσης γνωρίζεις ότι κάθε είσοδος από το πληκτρολόγιο αποθηκεύεται ως μία συμβολοσειρά(string) και στην περίπτωση που η είσοδος αυτή αντιστοιχεί σε αριθμό, πρέπει να χρησιμοποιήσεις κατάλληλη συνάρτηση μετατροπής της συμβολοσειράς αυτής σε ακέραια ή δεκαδική τιμή.

Έτσι δεν είναι;

Ή μήπως όχι;

## Σύντομες Ερωτήσεις

Απάντησε προσεκτικά στις παρακάτω ερωτήσεις χωρίς να εκτελέσεις τις εντολές:

**Ερώτηση 1:** Ποιο είναι το μήνυμα εκτύπωσης του παρακάτω προγράμματος;

- `number_1="275"`
- `number_2="574"`
- `number_3=number_2+number_1`
- `print(number_3)`

1. 849
2. 275274
3. 274275

**Ερώτηση 2:** Τι θα εκτυπωθεί αν ως είσοδο στην ερώτηση: "Πόσο χρονών είσαι;", ο χρήστης πληκτρολογήσει το 17;

- `name="Clara"`
- `age=input("Πόσο χρονών είσαι;")`
- `new_age=age+3`
- `print(new_age)`

1. Δε θα εκτελεστεί, διότι περιέχει λάθος
2. Θα εκτυπώσει την νέα ηλικία, που είναι το: 20
3. Θα εκτυπώσει το: 173

**Ερώτηση 3:** Ποιο είναι το αποτέλεσμα του παραπάνω προγράμματος;

- `txt_1="Γεια σου!"`
- `txt_2='Τι κάνεις;'`
- `print( txt_1, txt_2, "Εγώ είμαι η Χαρά" )`

1. Γεια σου! Τι κάνεις; Εγώ είμαι η Χαρά
2. Δε θα εκτελεστεί αφού δεν είναι καλά ορισμένες οι συμβολοσειρές.
3. Γεια σου! Εγώ είμαι η Χαρά

## Σύντομη Επανάληψη

- Μια συμβολοσειρά είναι μία ακολουθία χαρακτήρων (γραμμάτων, αριθμών, συμβόλων) οι οποίοι εγκλείονται σε εισαγωγικά μονά ή διπλά και έχουν σταθερό μέγεθος και περιεχόμενο.
- Μπορείς να ενώσεις δύο συμβολοσειρές με τη χρήση του τελεστή +.

- Όταν χρησιμοποιείς την συνάρτηση `input()` για είσοδο τιμής από το πληκτρολόγιο, αυτή η τιμή, ακόμη και αν είναι αριθμός, επιστρέφεται και αποθηκεύεται πάντα ως συμβολοσειρά. Για να την μετατρέψεις σε αριθμητική τιμή, ακέραια ή πραγματική, πρέπει να χρησιμοποιήσεις κατάλληλα τις συναρτήσεις `int()` και `float()` αντίστοιχα.

Πρόγραμμα με Εντολές	Εκτέλεση και αποτέλεσμα στην Οθόνη
<pre>txt_1= "Γεια! Τι κάνεις; " txt_2= 'Today is a wonderful day' print (txt_2) txt_3 = txt_1+txt_2 print (txt_3) number = int(input ( "give a number")) sum=number+3 print (number, " + 3 = " , sum)</pre>	<pre>Γεια! Τι κάνεις; Today is a wonderful day give a number: 4 4 + 3 = 7</pre>

Μια συμβολοσειρά ...είναι μια ακολουθία από χαρακτήρες. Οι χαρακτήρες αυτοί αποθηκεύονται σε διαδοχικές θέσεις μνήμης. Επομένως μπορούμε να έχουμε πρόσβαση στα διάφορα στοιχεία μιας συμβολοσειράς αλλά και να τα "μετατρέπουμε".

### Μαθησιακοί Στόχοι:

Εκτέλεσε τις παρακάτω δραστηριότητες και μάθε πως:

- Να αποσπάς ένα συγκεκριμένο μέρος μιας συμβολοσειράς.
- Να δημιουργείς και να αποθηκεύεις μια νέα συμβολοσειρά
- Να ελέγχεις αν μια συμβολοσειρά είναι μέρος μιας μεγαλύτερης
- Να βρίσκεις το μέγεθος μιας συμβολοσειράς
- Να συγκρίνεις 2 συμβολοσειρές μεταξύ τους.



### Δραστηριότητα 1

Μελέτησε τις παρακάτω εντολές το παρακάτω πρόγραμμα

- ```
1. name= input ( "Πως σε λένε; " )
2. how_many_letters= len(name)
3. first_letter= name[0]
4. print ( "Γεια σου ", name, "το όνομά σου ξεκινάει με ", first_letter, "και έχει ", how_many_letters, "γράμματα")
```

**Ερώτηση 1:** Αν `len()` συνάρτηση που επιστρέφει το μήκος μιας συμβολοσειράς ποιο πιστεύεις είναι το αποτέλεσμα της εκτέλεσης του παραπάνω προγράμματος;

Εκτέλεσε τις εντολές και επαλήθευσε αν μάντεψες σωστά καταγράφοντας το αποτέλεσμα της εκτέλεσης

**Ερώτηση 2:** Τι τιμή πήρε η μεταβλητή `first_letter` και ποιος ο ρόλος του δείκτη 0 στην εντολή 3;

**Ερώτηση 3:** Ποιες εντολές θα πρέπει να προστεθούν ώστε να εμφανιστεί και το τελευταίο γράμμα του ονόματος που περιέχει η μεταβλητή `name`; Εκτέλεσε τον καινούριο σου κώδικα για να ελέγξεις αν σκέφτηκες σωστά!

## Δραστηριότητα 2

Μελέτησε τις παρακάτω εντολές:

```
1. date= "Τετάρτη24Φεβρουαρίου"  
2. day=date[:7]  
3. the_date=date[7:9]  
4. month=date[9:]  
5. print ( "Σήμερα είναι ", day, " και έχουμε ", the_date, "του μήνα ", month)
```

**Ερώτηση 1:** Ποιο πιστεύεις ότι είναι το αποτέλεσμα εκτέλεσης των παραπάνω εντολών;

Εκτέλεσε τις εντολές και επαλήθευσε αν μάντεψες σωστά καταγράφοντας το αποτέλεσμα της εκτέλεσης.

**Ερώτηση 2:** Τι τιμές παίρνουν οι μεταβλητές `day`, `the_date` και `month` και πως προκύπτουν αυτές οι τιμές;

Έστω ότι στο παραπάνω πρόγραμμα προστίθενται οι εντολές 6 και 7:

```
6. date=date+"2021"  
7. print(date)
```

**Ερώτηση 3:** Ποια τιμή πιστεύεις ότι παίρνει η μεταβλητή `date`;

Εκτέλεσε το πρόγραμμα για να ελέγξεις αν σκέφτηκες σωστά καταγράφοντας το αποτέλεσμα της εκτέλεσης.

**Ερώτηση 4:** Με ποιον τρόπο η μεταβλητή `date` επέκτεινε τη συμβολοσειρά που ήδη είχε από πριν;

## Πρόβλημα 1:

Γράψε πρόγραμμα που να ζητάει από τον χρήστη μία συμβολοσειρά μήκους 10. Αν η συμβολοσειρά είναι μήκους 10 τότε στη συνέχεια θα εμφανίζει την ανεστραμμένη συμβολοσειρά. Δηλαδή αν ο χρήστης εισάγει το Αναξαγόρας, τότε θα εμφανίσει το: ζαρόγαξαναΑ. Αν η συμβολοσειρά ωστόσο δεν είναι μήκους 10 τότε θα του εμφανίζει το μήνυμα "Δεν εισήγαγες συμβολοσειρά μήκους 10."

**Ερώτηση :** Κάποιος έγραψε τις παρακάτω εντολές:

```
1. my_name="marra"
2. print( "Γεια σου, με λένε " ,my_name)
3. print( "Μάλλον με συγχωρείς !" )
4. my_name[3]= "i"
5. print( "Με λένε " ,my_name)
```

Ποιο νομίζεις ότι θα είναι το αποτέλεσμα της εκτέλεσης του παραπάνω προγράμματος;

1. Μάλλον δεν θα εκτελεστεί σωστά εμφανίζοντας κάποιο μήνυμα λάθους...
2. Θα εμφανίσει το μήνυμα " Με λένε Maria"..

### Δραστηριότητα 3

Μελέτησε τις παρακάτω εντολές

```
1. my_name= "mara"
2. print( "Γεια σου, με λένε " , my_name)
3. print( "Μάλλον με συγχωρείς !" )
4. my_name=my_name[:3] + "i" + my_name[3]
5. print( "Με λένε " ,my_name)
```

**Ερώτηση 1:** Ποιο πιστεύεις είναι αυτή τη φορά το αποτέλεσμα της εκτέλεσης των παραπάνω εντολών;

Εκτέλεσε τις εντολές επαληθεύοντας αν μάντεψες σωστά πριν καταγράφοντας το αποτέλεσμα της εκτέλεσης.

**Ερώτηση 2:** Ποια τιμή παίρνει η μεταβλητή my\_name στην εντολή 1 και ποια αποκτά τελικά στην εντολή 4; Πως τροποποιήθηκε αυτή η τιμή;

Στις παραπάνω εντολές προστίθενται οι παρακάτω:

```
friend_name= "Lora"
friend_name=friend_name.replace( "o" , "a" )
print( "και από εδώ η φίλη μου η " , friend_name)
```

**Ερώτηση 3:** Τι τιμή πιστεύεις αποκτά η μεταβλητή `friend_name` στην εντολή 6 και ποια στην 7;

---

Εκτέλεσε τις εντολές για να επαληθεύσεις αν μάντεψες σωστά καταγράφοντας το αποτέλεσμα της εκτέλεσης!

---

**Ερώτηση 4:** Πως μετατράπηκε η συμβολοσειρά αυτή ή αλλιώς ποια είναι η λειτουργία της μεθόδου `.replace()`;

```
1. your_name=input( "Εσένα πως σε λένε;" )
2. your_name=your_name.upper()
3. print( "Γεια σου " , your_name)
```

**Ερώτηση 5:** Ποια τιμή πιστεύεις έχει η μεταβλητή `your_name` μετά την εκτέλεση της εντολής 9 και ποια αποκτά μετά την εκτέλεση της εντολής 10;

---

Εκτέλεσε τον κώδικα για να επαληθεύσεις αν σκέφτηκες σωστά καταγράφοντας το αποτέλεσμα της εκτέλεσης.

---

**Ερώτηση 6:** Με ποιον τρόπο αυτή η τιμή μεταβλήθηκε ή αλλιώς ποια είναι η λειτουργία της μεθόδου `upper()`;

---

**Ερώτηση 7:** Κάποιος έγραψε την εξής εντολή:

- `today="είΝαι ΜΙΑ ωΡΑΙΑ μέρα ΣήΜΕΡα"`

Αντίστοιχα με τη μέθοδο `upper()`, υπάρχει η μέθοδος `lower()` για την μετατροπή μιας συμβολοσειράς στην αντίστοιχη με μικρά γράμματα.

Γράψε ποιες εντολές χρειάζονται ώστε 1) να μετατραπεί το περιεχόμενο της μεταβλητής `today` σε συμβολοσειρά που περιέχει μόνο μικρά γράμματα και 2) να εμφανίζεται τελικά στην οθόνη.

---

## Πρόβλημα 2:

Γράψε πρόγραμμα το οποίο να ζητάει από τον χρήστη να εισάγει 5 συμβολοσειρές (λέξεις) που να περιλαμβάνουν τουλάχιστον 5 χαρακτήρες. Για κάθε είσοδο θα γίνεται κατάλληλος έλεγχος. Αν κάποια από τις συμβολοσειρές που δώσει ο χρήστης έχει μήκος μικρότερο από 5, τότε το πρόγραμμα τερματίζει εμφανίζοντας κατάλληλο μήνυμα " Εισήγαγες συμβολοσειρά με λιγότερους από 5 χαρακτήρες!". Αν η είσοδος είναι κατάλληλη τότε συνεχίζει ζητώντας την επόμενη είσοδο από τον χρήστη. Σε περίπτωση που δοθούν 5 κατάλληλες συμβολοσειρές τότε θα εμφανίζει μήνυμα με την τελική συμβολοσειρά που θα δημιουργηθεί. Η συμβολοσειρά αυτή θα έχει τον **1ο χαρακτήρα από την πρώτη λέξη**, τον **2ο χαρακτήρα από την 2η λέξη**,....., τον **5ο χαρακτήρα από την 5η λέξη** και θα περιέχει αυστηρά μικρά γράμματα.

Δηλαδή αν δοθούν οι συμβολοσειρές:

**καρέκλα, ουρανοξύστης, τραπέζι, κράνος, κουκούλα,**  
η τελική συμβολοσειρά που θα εκτυπωθεί θα είναι: **κυανο!**

#### Δραστηριότητα 4

Κάποιος εκτέλεσε το παρακάτω πρόγραμμα 2 φορές. Την πρώτη φορά έδωσε ως είσοδο το :**" μπλε κίτρινο κόκκινο"** και τη δεύτερη το: **" μαύρο γαλάζιο κίτρινο "**

```
1. answer= input ( " Πες μου 3 αγαπημένα σου χρώματα:" )
2. if ("ΜΠΛΕ" in answer) or ("μπλε" in answer):
3.     print ("Τέλεια! Και εμένα μ' αρέσει πολύ το μπλε")
4. else:
5.     print( "Α! Ωραίο το γαλάζιο αλλά εμένα μ' αρέσει πολύ το
    μπλε...." )
```

**Ερώτηση 1:** Αν in υπαρξιακός τελεστής, ποια πιστεύεις ότι είναι τα αποτελέσματα των δύο εκτελέσεων;

---

Εκτέλεσε τις εντολές με τις παραπάνω δύο εισόδους και επαλήθευσε αν μάντεψες σωστά πριν καταγράφοντας τα αποτελέσματα των εκτελέσεων.

---

**Ερώτηση 2:** Πότε εκτελέστηκε η εντολή 2 και πότε η εντολή 5; Γιατί; (Πώς συνέβαλε στο αποτέλεσμα αυτό ο τελεστής in;)

#### Δραστηριότητα 5

Μελέτησε τις παρακάτω εντολές

```
1. fruit_1= "apple"
2. fruit_2= "banana"
3. if fruit_1>fruit_2:
4.     print ( "Πρέπει να φας μήλο σήμερα!" )
5. else:
6.     print ( "Πρέπει να φας μπανάνα σήμερα!" )
```

**Ερώτηση 1:** Ποιο μήνυμα πιστεύεις θα εκτυπωθεί στην οθόνη;

---

Εκτέλεσε τις εντολές και επαλήθευε αν μάντεψες σωστά καταγράφοντας το αποτέλεσμα της εκτέλεσης.

---

Έστω τώρα ότι η εντολή 2 αντικαθιστάται με την εξής εντολή: **fruit\_2= "Banana"**.

```
fruit_1= "apple"
fruit_2= "Banana"
if fruit_1>fruit_2:
    print ( "Πρέπει να φας μήλο σήμερα!" )
else:
    print ( "Πρέπει να φας μπανάνα σήμερα!" )
```

**Ερώτηση 2:** Ποιο μήνυμα πιστεύεις θα εκτυπωθεί αυτή τη φορά στην οθόνη;

---

Ξανά εκτέλεσε τις εντολές καταγράφοντας το αποτέλεσμα της εκτέλεσης.

---

**Ερώτηση 3:** Κατάλαβες με ποιον τρόπο συγκρίνονται οι συμβολοσειρές;

---

### Δραστηριότητα Διακλάδωσης

Εκτέλεσε το παρακάτω πρόγραμμα δίνοντας ως είσοδο την χρονιά που γεννήθηκες π.χ. 2004.

```
1. year=int( input ( " Να σε ρωτήσω κάτι; "Ποια χρονιά γεννήθηκες;" ) )
2. age=2021-year
3. txt= "Άρα είσαι {} χρονών"
4. txt=txt.format(age)
5. print (txt)
```

**Ερώτηση 1:** Η μεταβλητή txt ποια τιμή/συμβολοσειρά αποκτά στην εντολή 4;

---

**Ερώτηση 2:** Ποια είναι η λειτουργία της μεθόδου format()

---

### Δραστηριότητα Διακλάδωσης

#### Πρόβλημα μέτριας δυσκολίας στον ένα κλάδο

Γράψε πρόγραμμα το οποίο να ζητάει από τον χρήστη να εισάγει ένα όνομα. Αν το όνομα λήγει σε 'ος' ή 'ας' τότε θα εμφανίζει το όνομα στη γενική. Δηλαδή αν για παράδειγμα ο χρήστης εισάγει το 'Γιώργος' ή το 'Κώστας', θα πρέπει να εμφανίζει τελικά το 'Γιώργου' ή το 'Κώστα', αντίστοιχα. Αν λήγει σε κάποια άλλη κατάληξη τότε το πρόγραμμα θα εμφανίζει κατάλληλο μήνυμα "Λυπάμαι δεν ξέρω την γενική κλήση αυτού του ονόματος".



| Κατάληξη ονομαστικής | Κατάληξη Γενικής |
|----------------------|------------------|
| Γιώργ-ος             | Γιώργ-ου         |
| Κώστ-ας              | Κώστ-α           |

 Tips:

- Για να λύσεις το πρόβλημα αυτό σκέψου ότι πρέπει να διακρίνεις 2 περιπτώσεις.
- Απομόνωσε την κατάληξη από το όνομα που εισάγει ο χρήστης.
- Στη συνέχεια έλεγξε, ανάλογα με την κατάληξη σε ποια από τις 2 κατηγορίες ανήκει.
- Διαμόρφωσε κατάλληλα το όνομα στην Γενική σύμφωνα με τους παραπάνω κανόνες.

### Το ίδιο πρόβλημα αυξημένης δυσκολίας στο δεύτερο κλάδο

Γράψε πρόγραμμα το οποίο θα ζητάει από τον χρήστη να εισάγει ένα όνομα στην Ονομαστική κλήση και στη συνέχεια θα εμφανίζει το όνομα στη Γενική κλήση. Δηλαδή αν ο χρήστης εισάγει το: Ελένη, θα πρέπει να εμφανιστεί τελικά το: Ελένης.

Οι πιο κοινοί κανόνες είναι οι εξής (ενδεχομένως να μην περιλαμβάνονται κάποια πολύ σπάνια αρχαία ονόματα):

| Κατάληξη ονομαστικής | Κατάληξη Γενικής |
|----------------------|------------------|
| Γιώργ-ος             | Γιώργ-ου         |
| Κώστ-ας              | Κώστ-α           |
| Χάρ-ης               | Χάρ-η            |
| Βίκτ-ωρ              | Βίκτ-ωρος        |
| Πειρίθ-ους           | Πειρίθ-ου        |
| Συμε-ών              | Συμε-όνος        |
| Ελέν-η               | Ελέν-ης          |
| Γιάνν-α              | Γιάνν-ας         |
| Κλει-ώ               | Κλει-ώς          |
| Αμαρυλλ-ίς           | Αμαριλλ-ίδος     |

 Tips:

- Για να λύσεις το πρόβλημα αυτό σκέψου ότι αυτή τη φορά έχεις να διαχειριστείς περισσότερες κατηγορίες καταλήξεων.
- Στον παραπάνω πίνακα μπορεί να παρουσιάζονται 10 κατηγορίες αλλά μπορείς να διακρίνεις κάποιες κατηγορίες όπου η μετατροπή από την Ονομαστική κλήση στην Γενική είναι ίδια. Για παράδειγμα αν το όνομα τελειώνει σε -α ή σε -η στην Γενική κλήση απλά προστίθεται ένα -ς στο τέλος.

## Υποστηρικτική Δραστηριότητα

Τι είναι μία συμβολοσειρά;

Πως αποθηκεύεται μια συμβολοσειρά στην μνήμη του υπολογιστή;

Υπάρχουν μεταβλητές που δεσμεύουν συγκεκριμένο σύνολο από θέσεις στην μνήμη για να αποθηκεύσουν τις τιμές τους;

Δες το [βίντεο](#) για να βρεις τις απαντήσεις στα παραπάνω ερωτήματα.



Για παραπάνω πληροφορίες σχετικά με τη διαχείριση συμβολοσειρών στη Python μπορείς να επισκεφτείς τον εξής ιστότοπο: [https://www.w3schools.com/python/python\\_strings.asp](https://www.w3schools.com/python/python_strings.asp)

## Θεωρία

### Μήκος συμβολοσειράς

- Συμβολοσειρά είναι οποιαδήποτε ακολουθία χαρακτήρων η οποία εγκλείεται σε μονά ή διπλά εισαγωγικά.
- Μπορούμε να μετρήσουμε το μήκος μιας συμβολοσειράς με τη συνάρτηση `len()`.
- Η `len()` είναι μία συνάρτηση που δέχεται ως είσοδο μια συμβολοσειρά και επιστρέφει έναν ακέραιο αριθμό που αντιστοιχεί στο μήκος της, δηλαδή από πόσους χαρακτήρες αποτελείται.

| ➤ Παράδειγμα/Εντολές                                                        | ➤ Εκτέλεση και Αποτέλεσμα στην οθόνη |
|-----------------------------------------------------------------------------|--------------------------------------|
| <pre>➤ fruit="banana" ➤ print(fruit, " = ", len(fruit),"χαρακτήρες" )</pre> | <pre>➤ banana = 6 χαρακτήρες</pre>   |

## Παρατήρηση!

Στην Python δεν υπάρχει κάποιος τύπος για τους χαρακτήρες. Όταν αναθέτουμε σε μία μεταβλητή έναν μόνο χαρακτήρα, τότε πρόκειται για μία συμβολοσειρά μήκους 1!

## Πρόσβαση στις Συμβολοσειρές

Μπορείς να έχεις πρόσβαση σε κάποιον χαρακτήρα μιας συμβολοσειράς μέσω ενός δείκτη που αναφέρεται στη θέση του χαρακτήρα μέσα στη συμβολοσειρά.

| Πρόγραμμα με Εντολές                                                                                                        | Εκτέλεση και Αποτέλεσμα στην οθόνη                                     |
|-----------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------|
| <pre>fruit="banana" print( "το δεύτερο γράμμα είναι: ", fruit[1]) print( "η λέξη ξεκινάει από το γράμμα: ", fruit[0])</pre> | <p>το δεύτερο γράμμα είναι: a<br/>η λέξη ξεκινάει από το γράμμα: b</p> |

Επίσης μπορείς να έχεις πρόσβαση σε ένα τμήμα της συμβολοσειράς με χρήση του τελεστή διαμέρισης (slice operator) “:”. Όταν γράφεις **word[a:b]**, επιστρέφεται το μέρος της συμβολοσειράς που ξεκινάει από τον χαρακτήρα στη θέση **a** μέχρι τη θέση **b**, χωρίς να περιλαμβάνεται ο χαρακτήρας στη θέση **b**.

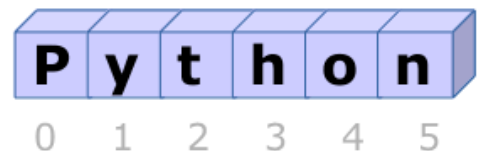
| Εντολές                                                                        | Αποτέλεσμα               |
|--------------------------------------------------------------------------------|--------------------------|
| <pre>fruit="bananas" print(fruit[:2]) print(fruit[2:4]) print(fruit[4:])</pre> | <p>ba<br/>na<br/>nas</p> |

## Αρίθμηση Θέσεων

Η αρίθμηση των θέσεων σε μια συμβολοσειρά (ή σε μια λίστα) **ξεκινάει από το 0**.

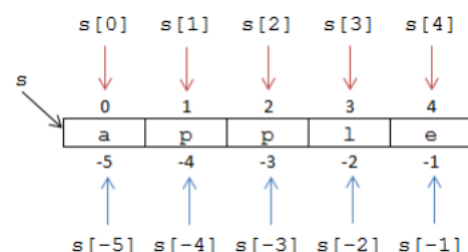
Για να βρεθεί η θέση ενός χαρακτήρα, το μέτρημα ξεκινάει από το 0 και όχι από το 1!

Σκέψου ότι είναι σαν να λες στον υπολογιστή πόσες θέσεις πρέπει να προχωρήσει από την αρχή της συμβολοσειράς, για να φτάσει στον χαρακτήρα που θέλουμε!



## Παρατηρήσεις!

- Δεν επιτρέπεται να έχεις πρόσβαση εκτός ορίων της συμβολοσειράς.
- Αν στην εντολή **word[a:b]** αφήσουμε κενό τον χαρακτήρα **a**, τότε εννοείται ότι **a=0**, ενώ αν αφήσουμε κενό τον χαρακτήρα **b** τότε εννοείται **όλο το υπόλοιπο τμήμα** μέχρι το τέλος της συμβολοσειράς!
- Η αρίθμηση θέσεων μιας συμβολοσειράς μπορεί να γίνει και αντίστροφα ξεκινώντας



από την τελευταία θέση.

## Τροποποίηση Συμβολοσειρών

Οι συμβολοσειρές στην Python έχουν σταθερό μέγεθος και μη μεταβαλλόμενο περιεχόμενο. Δηλαδή, **δεν μπορείς να προσθέσεις ή να αφαιρέσεις χαρακτήρες, ούτε να τροποποιήσεις το περιεχόμενό τους.**

Υπάρχουν ωστόσο μερικές μέθοδοι που μπορούν να μετατρέψουν μία συμβολοσειρά σε μία αντίστοιχη.

**upper()** -> μετατροπή συμβολοσειράς σε αντίστοιχη με κεφαλαία γράμματα

**lower()** -> μετατροπή συμβολοσειράς σε αντίστοιχη με μικρά γράμματα

**replace()**-> μετατροπή ενός χαρακτήρα με έναν άλλον

| Παράδειγμα                                                                                                        | Αποτέλεσμα                   |
|-------------------------------------------------------------------------------------------------------------------|------------------------------|
| <pre>word=word.upper() print (word) word=word.lower() print (word) word=word.replace("i", "ή") print (word)</pre> | <pre>ΚΛΙΜΑ κλίμα κλήμα</pre> |

Οι παραπάνω μέθοδοι δεν μετατρέπουν την ίδια τη συμβολοσειρά, αλλά **δημιουργούν μία καινούρια** την οποία την αναθέτουν εκ νέου σε μια μεταβλητή.

## Υπαρξιακός Τελεστής

Ο **in** είναι υπαρξιακός τελεστής και ελέγχει αν ένα αντικείμενο ανήκει σε ένα σύνολο αντικειμένων.

```
>>" dog " in " i have 3 dogs "
True
```

Δεδομένου ότι οι συμβολοσειρές μπορούν να θεωρηθούν ως σύνολα χαρακτήρων, ο τελεστής **in** χρησιμοποιείται για να ελέγξει αν μια συμβολοσειρά περιέχεται σε μία δεύτερη μεγαλύτερη συμβολοσειρά.

- ❖ Παραπάνω το **" dog " in " i have 3 dogs "** αποτελεί μία λογική έκφραση και το αποτέλεσμα είναι **True** ή **False** ανάλογα με το αν το **" dog "** περιέχεται στο **" i have 3 dogs "**.

## Σύγκριση Συμβολοσειρών

Στις γλώσσες προγραμματισμού μπορούμε να συγκρίνουμε και συμβολοσειρές!

Κάθε σύμβολο ή γράμμα ουσιαστικά αντιστοιχεί σε μία **ακολουθία από 0 και 1**, δηλαδή τιμές που καταλαβαίνει ο

```
>>'banana'>'apple'
True
>>'Banana'>'apple'
False
```

υπολογιστής! Αυτή η ακολουθία με τη σειρά της, αντιστοιχεί σε έναν αριθμό.

Έτσι, κάθε χαρακτήρας αντιστοιχεί σε έναν αριθμό και επομένως όταν συγκρίνονται δύο συμβολοσειρές, συγκρίνονται ένας-ένας οι χαρακτήρες μεταξύ τους και επομένως συγκρίνονται οι αριθμοί που αντιστοιχούν στους χαρακτήρες αυτούς.

- Γενικά να θυμάσε ότι, ό,τι αφορά τους λατινικούς χαρακτήρες, το **a** είναι μικρότερο από το **b** το οποίο είναι μικρότερο από το **c** κ.ο.κ. Επίσης τα κεφαλαία γράμματα είναι μικρότερα από τα πεζά.

Δηλαδή ισχύει το εξής: **A<B<C<.....<Y<Z<a<b<.....<y<z**

- Θυμήσου ότι όταν συγκρίνεις 2 συμβολοσειρές, πρώτα συγκρίνονται οι πρώτοι χαρακτήρες. Αν είναι ίδιοι, μετά οι δεύτεροι κ.ο.κ.
- Όλοι οι γνωστοί συγκριτικοί τελεστές (**, >=, ==, !=**) ισχύουν και στις συμβολοσειρές.

**banana>apple => True (Γιατί b>a)**

**Banana>apple =>False (Γιατί B<a)**

Μπορείς να δεις το βίντεο που αναφέρεται σχετικά με την αναπαράσταση των συμβολοσειρών σε ακολουθίες 0 κ' 1.

Μπορείς να βάλεις αγγλικούς υπότιτλους και να παρακολουθήσεις το τμήμα: 6.42-10.13

<https://www.youtube.com/watch?v=1GSjbWt0c9M>

### Μετατροπή τιμής σε συμβολοσειρά

Στην Python δεν μπορείς να ενώσεις συμβολοσειρές και αριθμούς κάπως έτσι:

Μπορείς όμως να το κάνεις αυτό μέσω της μεθόδου `format()`, η οποία παίρνει μια είσοδο (π.χ. αριθμητική τιμή), τη μετατρέπει και την προσθέτει στη συμβολοσειρά στη θέση που υπάρχουν οι αγκύλες `{}`.

Μπορεί να πάρει παραπάνω από μία εισόδους!

- Με τη μέθοδο αυτή μπορείς να μετατρέπεις τις αριθμητικές σου τιμές σε συμβολοσειρές!

```
age = 36
txt = "My name is John, I am " + age
print(txt)
```

| Εντολές                                                                         | Εκτέλεση και Αποτέλεσμα στην οθόνη      |
|---------------------------------------------------------------------------------|-----------------------------------------|
| <pre>age = 36 txt = "My name is John, and I am {" print (txt.format(age))</pre> | <pre>My name is John, and I am 36</pre> |

|                                                                                                                                                                      |                                                      |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------|
| <pre>quantity = 3 itemno = 567 myorder = "I want {} pieces of item {}." print (myorder.format(quantity, itemno))</pre>                                               | I want 3 pieces of item 567.                         |
| <pre>quantity = 3 itemno = 567 price = 49.95 myorder = "I want to pay {2} dollars for {0} pieces of item {1}." print (myorder.format(quantity, itemno, price))</pre> | I want to pay 49.95 dollars for 3 pieces of item 567 |

## Μάθημα 5: Οι Λίστες στην Python

### Μαθησιακοί Στόχοι

Μέχρι τώρα έχεις μάθει να ορίζεις και να διαχειρίζεσαι αριθμητικές, αλφαριθμητικές και λογικές μεταβλητές. Οι παραπάνω μεταβλητές περιέχουν μία συγκεκριμένη τιμή. Ωστόσο υπάρχουν και **μεταβλητές που περιέχουν παραπάνω από μια τιμές**. Η **λίστα** είναι ένας τέτοιος τύπος μεταβλητής!

Είσαι έτοιμος λοιπόν να μάθεις πως:

- να δημιουργείς τις δικές σου λίστες
- να διαχειρίζεσαι τις τιμές μιας λίστας
- να τροποποιείς τις τιμές που περιέχει μια λίστα
- να χρησιμοποιείς συναρτήσεις και μεθόδους της Python ώστε να υπολογίζεις το μέγεθος μιας λίστας, να προσθέτεις και να αφαιρείς τιμές σε αυτή



### Σύντομη Επανάληψη

Απάντησε στις παρακάτω ερωτήσεις πολλαπλής επιλογής σχετικά με τις τιμές και τις μεταβλητές.

#### Ερώτηση 1: Διάλεξε τον τύπο που ανήκουν οι παρακάτω τιμές

- 6
- "Καλημέρα", "590"
- True

- 5948.8

**Ερώτηση 2:** Τι τιμή πιστεύεις ότι παίρνει η μεταβλητή age στο παρακάτω μπλοκ εντολών:

- age=7
- print("Η εβδομάδα έχει", age, "μέρες")
- age=age+10
- print("Ο αδερφός μου είναι", age, "χρονών")

1. Η μεταβλητή age στο τέλος αποκτά την ακέραια τιμή 17.
2. Η μεταβλητή age παραμένει ως το τέλος με την τιμή 7.
3. Η μεταβλητή age διαθέτει την τιμή 7 αλλά και την 17 αφού της ανατέθηκαν και οι δύο μέσα στο πρόγραμμα.

**Ερώτηση 3:** Έχουμε τις εξής εντολές:

- year=2021
- year="2021"

Η μεταβλητή year τελικά τι τιμή είναι;

1. Η year στο τέλος της εκτέλεσης των εντολών είναι μία αλφαριθμητική μεταβλητή
2. Η year στο τέλος της εκτέλεσης των εντολών είναι μία αριθμητική μεταβλητή.

## Δραστηριότητα 1

**Η λίστα για τον μανάβη.**

Μελέτησε το παρακάτω πρόγραμμα :

```
1. #Ας φτιάξω τη λίστα μου με φρούτα και λαχανικά που θα δώσω στον μανάβη
2. fruits_list = [ "μπανάνες", "μήλα", "καρπούζι", "αβοκάντο", "κεράσια", "λεμόνια" ]
3. print ( "Πελάτης: Τα πράγματα που θα ήθελα είναι:", fruits_list )
4. extra_fruit=input( "Πελάτης: Θέλω να προσθέσω μερικά: " )
5. if extra_fruit not in fruits_list:
6.     fruits_list.append( extra_fruit )
7. #Μάλλον δε θα πάρω το καρπούζι
8. fruits_list.remove( "καρπούζι" )
9. print ( "Μανάβης: Κάτι άλλο εκτός από:", fruits_list , ";" )
10.#Μήπως αντί για κεράσια να πάρω σμέουρα;
11.fruits_list[3]="σμέουρα"
12.number_of_fruits=len( fruits_list )
13.print ( "Μανάβης: Η παραγγελία σας περιλαμβάνει ", number_of_fruits, " προϊόντα
    που είναι τα εξής: " )
14.for item in fruits_list:
15.     print ( item )
```

**Ερώτηση 1:** Στην εντολή 2 ορίζεται η λίστα fruits\_list, τι είδους και ποιες τιμές περιέχει;

**Ερώτηση 2:** Κάποιος εκτελεί το πρόγραμμα 2 φορές και στην εντολή 4 ο χρήστης δίνει ως είσοδο από το πληκτρολόγιο το "πορτοκάλια" την πρώτη φορά και το "μήλα" τη δεύτερη φορά, σε ποια από τις 2 εκτελέσεις πιστεύεις ότι εκτελείται η εντολή 6;

---

**Ερώτηση 3:** Μετά την εκτέλεση της εντολής 8 πιστεύεις ότι το καρπούζι τελικά θα υπάρχει στη λίστα;

---

**Ερώτηση 4:** Ποια τιμή πιστεύεις ότι παίρνει η μεταβλητή `number_of_fruits` στην εντολή 12;

---

**Ερώτηση 5:** Ποια τιμή πιστεύεις ότι παίρνει η μεταβλητή `item` στις εντολές 14-15;

---

Εκτέλεσε το πρόγραμμα 2 φορές. Στην εντολή 4 ζητάει είσοδο από το πληκτρολόγιο. Την πρώτη φορά βάλε την τιμή "πορτοκάλια" και τη δεύτερη "μήλα". Για την εκτέλεση του προγράμματος χρησιμοποίησε το εξής εργαλείο οπτικοποίησης: <http://pythontutor.com/live.html#mode=edit>.

**Ερώτηση 6:** Τελικά σε ποια από τις δύο εκτελέσεις που έκανες, εκτελέστηκε η εντολή 6 και γιατί; Ποια είναι η λειτουργία του τελεστή `not in` και ποια της συνάρτησης `append()`;

---

**Ερώτηση 7:** Στην εντολή 8 οι τιμές της `fruits_list` μεταβλήθηκαν και αν ναι πως; Ποια είναι η λειτουργία της συνάρτησης `remove()`;

---

**Ερώτηση 8:** Τι τιμές έχει η `fruits_list` πριν την εκτέλεση της εντολής 11 και ποιες μετά; Ποια είναι η λειτουργία της εντολής αυτής;

---

**Ερώτηση 9:** Ποια τιμή παίρνει τελικά η μεταβλητή `number_of_fruits` στην εντολή 12; Ποια είναι η λειτουργία της συνάρτησης `len()`;

---

**Ερώτηση 10:** Πόσες φορές εκτελούνται οι εντολές 14 κ'15 και ποιες τιμές παίρνει η μεταβλητή `item`;

---

**Ερώτηση 11:** Πως μπορείς να εκτυπώσεις τις τιμές μιας λίστας; Γράψε τους δύο τρόπους που υπάρχουν στον κώδικα.

---

## Πρόβλημα 1

Γέμισμα λίστας με αριθμούς:



Προσπάθησε να λύσεις το παρακάτω πρόβλημα που περιλαμβάνει μία λίστα με αριθμούς.

Φτιάξε πρόγραμμα το οποίο:

- Στην αρχική λίστα `numbers_list` [20, 14.5, 8.1, -3.9, 67.7] να προστίθενται άλλοι 3 τυχαίοι αριθμοί οι οποίοι βρίσκονται στο διάστημα [80,200]. Επίσης στην ίδια λίστα θα προστίθενται άλλες δύο τιμές που θα δίνονται ως είσοδο από το πληκτρολόγιο αφού ελεγχθεί ότι δεν υπάρχουν ήδη στη λίστα.
- Αφού δημιουργηθεί η τελική λίστα, να εκτυπώνονται οι τιμές που περιέχει.
- Στη συνέχεια να ορίζεται μία νέα λίστα `new_list`. Στη λίστα αυτή θα αποθηκεύονται οι τιμές της αρχικής λίστας αυξημένες όμως κατά τον πρώτο της αριθμό. Δηλαδή αν `numbers_list` [20, 14.5, 8.1, -3.9, 67.7, 199, 90, 4, 7] τότε όλες οι τιμές θα αυξηθούν κατά 20 και θα προκύψει η `new_list`[40, 34.5,28.1, 16.1, 87.7, 219, 110, 24, 27].
- Τέλος πρέπει να εκτυπώνονται οι τιμές της καινούριας λίστας.
- Extra: Να τροποποιεί την `numbers_list` αυξάνοντας τις τιμές της κατά την πρώτη της τιμή. Δηλαδή οι νέες τιμές να μην αποθηκεύονται σε νέα λίστα αλλά να τροποποιείται η αρχική `numbers_list`.

## Δραστηριότητα Διακλάδωσης

### Το Πρόβλημα της Ντουλάπας

Έστω ότι η λίστα `wordrobe` που αναφέρεται στο παρακάτω πρόγραμμα αναπαριστά τα διάφορα συρτάρια που υπάρχουν σε μία ντουλάπα με το ανάλογο περιεχόμενο.

```
1. wordrobe=['μπλούζες', 'παντελόνια', 'πουλόβερ', 'φόρμες', 'πουκάμισα']
2. position_of_trousers=wordrobe.index('παντελόνια')
3. print('Τα παντελόνια βρίσκονται στο ', position_of_trousers, 'ράφι της βιβλιοθήκης')
4. two_last=wordrobe[3:]
5. print('Τα δυο τελευταία ράφια περιέχουν:', two_last)
```

**Ερώτηση 1:** Στην εντολή 2 η μεταβλητή `position_of_trousers` ποια τιμή πιστεύεις παίρνει;

---

**Ερώτηση 2:** Η μεταβλητή `two_last` στην εντολή 4 τι τιμή/τιμές πιστεύεις παίρνει και πως τις αποκτά;

---

Εκτέλεσε το παραπάνω πρόγραμμα στο εργαλείο οπτικοποίησης: <http://pythontutor.com/live.html#mode=edit> :

**Ερώτηση 3:** Στην εντολή 2 τελικά η μεταβλητή `position_of_trousers` ποια τιμή παίρνει; Ποια είναι η λειτουργία της συνάρτησης `index()`;

---

**Ερώτηση 4:** Η μεταβλητή `two_last` στην εντολή 4 τι τιμή/τιμές παίρνει και πως τις αποκτά;

---

Έστω ότι στις παραπάνω εντολές προστίθεται η εντολή 6:

- `print(wordrobe[4:2:-1])`

**Ερώτηση 5:** Ποιο πιστεύεις θα είναι το αποτέλεσμα την εκτέλεσης της εντολής αυτής; Εκτέλεσε τις εντολές 1-6. Ποιο είναι τελικά το αποτέλεσμα της εκτέλεσης και τι σημαίνουν οι αριθμοί 4,2 κ' -1 που βρίσκονται μέσα στις αγκύλες ( wordrobe[4:2:-1]);

---

Έστω ότι αποφασίζεις να κρεμάσεις τα πουκάμισα που υπάρχουν στο τελευταίο συρτάρι στις κρεμάστρες και θες να καταχωρήσεις στην νέα σου μεταβλητή new\_wordrobe όλα τα ρούχα που είχες πριν, με την ίδια σειρά, εκτός όμως από τα πουκάμισα. Κάποιος σου προτείνει τις εξής δύο πιθανές εντολές ως λύση στο πρόβλημα:

- new\_wordrobe=wordrobe[:3]
- new\_wordrobe=wordrobe[:4]

**Ερώτηση 6:** Ποια εντολή από τις παραπάνω είναι η λύση για το παραπάνω πρόβλημα και γιατί; Εκτέλεσε το καινούριο πρόγραμμα εμφανίζοντας στο τέλος την νέα λίστα new\_wordrobe. Ποια είναι τελικά η σωστή εντολή;

---

## Δραστηριότητα της ίδιας Διακλάδωσης

### Πρόβλημα: Βρες τη διαδρομή

Ο Ηλεκτρικός της Αθήνας εγκαινιάστηκε το 1869 και συνέδεε τον Πειραιά με το Θησείο. Μετά από 88 χρόνια η διαδρομή επεκτάθηκε μέχρι την Κηφισιά. Σήμερα αριθμεί 24 σταθμούς.

Υλοποίησε ένα πρόγραμμα που θα βοηθά το χρήστη να μετακινηθεί με τον Ηλεκτρικό. Ο χρήστης θα δίνει το όνομα του σταθμού από τον οποίο θ' αναχωρήσει και το όνομα του σταθμού στον οποίο θέλει να φτάσει και το πρόγραμμα θα του εμφανίζει την κατεύθυνση που θα ακολουθήσει (προς Πειραιά ή προς Κηφισιά) και τα ονόματα όλων των ενδιάμεσων σταθμών της διαδρομής.

### Ένα παράδειγμα αλληλεπίδρασης με το πρόγραμμα:

Δώστε σταθμό αφετηρίας: ΠΕΤΡΑΛΩΝΑ

Δώστε σταθμό προορισμού: ΦΑΛΗΡΟ

Σταθμοί από ΠΕΤΡΑΛΩΝΑ προς ΦΑΛΗΡΟ με κατεύθυνση ΠΕΙΡΑΙΑΣ

ΤΑΥΡΟΣ

ΚΑΛΛΙΘΕΑ

ΜΟΣΧΑΤΟ

### TIPS:

- Όρισε αρχικά τη λίστα με τους σταθμούς: `stations= ["ΠΕΙΡΑΙΑΣ", "ΦΑΛΗΡΟ", "ΜΟΣΧΑΤΟ", "ΚΑΛΛΙΘΕΑ", "ΤΑΥΡΟΣ", "ΠΕΤΡΑΛΩΝΑ", "ΘΗΣΕΙΟ", "ΜΟΝΑΣΤΗΡΑΚΙ", "ΟΜΟΝΟΙΑ", "ΒΙΚΤΩΡΙΑ", "ΑΤΤΙΚΗ", "ΑΓΙΟΣ ΝΙΚΟΛΑΟΣ", "ΚΑΤΩ ΠΑΤΗΣΙΑ", "ΑΓΙΟΣ ΕΛΕΥΘΕΡΙΟΣ", "ΑΝΩ ΠΑΤΗΣΙΑ", "ΠΕΡΙΣΣΟΣ", "ΠΕΥΚΑΚΙΑ", "ΝΕΑ ΙΩΝΙΑ", "ΗΡΑΚΛΕΙΟ", "ΕΙΡΗΝΗ", "ΝΕΡΑΤΖΙΩΤΙΣΣΑ", "ΜΑΡΟΥΣΙ", "Κ.Α.Τ.", "ΚΗΦΙΣΙΑ"]`
- Στην συνέχεια ζήτη τον σταθμό αναχώρησης και προορισμού. Καλό είναι να υπάρχει έλεγχος των τιμών που δόθηκαν.

- Αν οι τιμές είναι επιτρεπτές και υπάρχουν στην λίστα,
  - βρες την κατεύθυνση της διαδρομής με τη βοήθεια της συνάρτησης `index()` που επιστρέφει τη θέση μιας τιμής στη λίστα.
  - και φτιάξε μια νέα λίστα `final_route[]` παίρνοντας το κατάλληλο τμήμα της λίστας `stations` με τους σταθμούς. Δώσε προσοχή στην αρχική/τελική θέση του τμήματος της λίστας και στο βήμα που θα βάλεις !
- Αν οι τιμές δεν είναι επιτρεπτές και δεν υπάρχουν στη λίστα τότε,
  - εμφάνισε κατάλληλο μήνυμα "Δυστυχώς οι σταθμοί που δώσατε δεν ήταν σωστοί!"

## Δραστηριότητα 2

Πρόβλημα: Κατάταξη μαθητών

Ο διευθυντής ενός σχολείου διαθέτει στα χέρια του έναν κατάλογο, `B_list`, με αιτήσεις, για τον διαγωνισμό ρομποτικής. Ο κατάλογος αυτός περιέχει τα ονόματα των μαθητών της Β' Λυκείου, τους αντίστοιχους τελικούς βαθμούς τους (0-20) στο μάθημα της πληροφορικής και το τμήμα που ανήκουν (B1, B2, B3).

Θέλει λοιπόν, αρχικά να κατατάξει τους μαθητές σε νέους καταλόγους ανάλογα με το τμήμα τους, `B1_list`, `B2_list`, `B3_list`, να βρει τον μαθητή με την μικρότερη και τη μεγαλύτερη βαθμολογία, καθώς και να βγάλει το μέσο όρο βαθμολογίας μαθητών της Β' τάξης που έχουν κάνει την αίτηση.

Γράψε κατάλληλο πρόγραμμα το οποίο:

- να εκτυπώνει την αρχική λίστα που διαθέτει ο διευθυντής
- να υπολογίζει όλα τα παραπάνω που ζητούνται
- τέλος να εκτυπώνει τα αποτελέσματα δηλαδή, τους νέους καταλόγους με τους κατανομημένους μαθητές στα τμήματα τους, τα στοιχεία του μαθητή με την μεγαλύτερη και τη μικρότερη βαθμολογία αλλά και το μέσο όρο βαθμολογίας της τάξης.

| <code>B_list</code>   | <code>B1_list</code> | <code>B2_list</code> | <code>B3_list</code> |
|-----------------------|----------------------|----------------------|----------------------|
| ['Μανώλης','B1',17.2] | [Μανώλης, B1,17.2]   | [Χρήστος,B2, 19.5]   | [Τάκης,B3,18.3]      |
| ['Τάκης','B3',18.3]   | [Ανδρέας,B1, 16.3]   | [Ερρικα,B2,15]       | [Αθηνά,B3,17]        |
| ['Αθηνά','B3',17]     | [Κατερίνα,B1, 16.6]  | [Λευτέρης,B2,14.9]   | [Μάρθα,B3,20]        |
| ['Ανδρέας','B1',16.3] | [Ρίτα,B1,17.4]       | [Νίκος,B2,15.5]      |                      |
| ['Χρήστος','B2',19.5] |                      |                      |                      |
| ['Ερρικα','B2',15]    |                      |                      |                      |

|                        |  |  |  |
|------------------------|--|--|--|
| ['Λευτέρης','B2',14.9] |  |  |  |
| ['Μάρθα','B3',20]      |  |  |  |
| ['Κατερίνα','B1',16.6] |  |  |  |
| ['Ρίτα','B1',17.4]     |  |  |  |
| ['Νίκος','B2',15.5]    |  |  |  |

Μελέτησε το παρακάτω πρόβλημα και προσπάθησε να λύσεις το παραπάνω!

### Πρόβλημα : Κατάταξη αιτήσεων νέων μελών

Ο Αντρέας είναι προπονητής τένις. Στην αρχή της χρονιάς δέχεται τις αιτήσεις για τις νέες εγγραφές. Πρέπει όμως να τους κατατάξει στο τμήμα ενηλίκων και ανηλίκων αντίστοιχα. Προσπαθεί λοιπόν να τους χωρίσει σε δύο λίστες. Επίσης θέλει να βρει το άτομο με τη μεγαλύτερη ηλικία γιατί σκέφτεται να δημιουργήσει στο μέλλον και ένα τρίτο τμήμα με άτομα άνω των 40.

```

1. #δημιουργία λίστας με τα στοιχεία των ατόμων
2. new_members_list=[['Γιώργος',26],['Μαρία',10],['Τάσος',19],['Γιάννα',7],['Σπύρος',15],['Ανατολή',14],['Κλειώ',21],['Τάκης',30],['Κατερίνα',23],['Λεωνίδα',17]]
3. #διάσχιση λίστας για να βρω το άτομο με τη μεγαλύτερη ηλικία
4. max_age=new_members_list[0][1]
5. oldest_member=new_members_list[0][0]
6. for person in new_members_list:
7.     if person[1]>max_age:
8.         #αποθήκευση ονόματος και ηλικίας του μεγαλύτερου μέλους
9.         max_age=person[1]
10.        oldest_member=person[0]
11. print('Ο', oldest_member, 'είναι το μεγαλύτερο μέλος και είναι',max_age,'χρονών')
12. adult_list=[]
13. children_list=[]
14. #διάσχιση αρχικής λίστας και 'γέμισμα' των επιμέρους λιστών
15. for person in new_members_list:
16.     if person[1]>=18:
17.         adult_list.append(person)
18.     else:
19.         children_list.append(person)
20. print('Το τμήμα με τα παιδιά αποτελείται από τους: ', children_list)
21. print('Το τμήμα με τους ενηλίκους αποτελείται από τους: ', adult_list)

```

Για τη λύση του παραπάνω προβλήματος δημιουργήθηκαν λίστες λιστών και έχοντας πρόσβαση στις κατάλληλες τιμές της λίστας έγιναν οι απαιτούμενες λειτουργίες.

Προσπάθησε να λύσεις το πρόβλημα με την κατάταξη των μαθητών με την ίδια λογική που λύθηκε το πρόβλημα της κατάταξης νέων μελών με βάση την ηλικία τους και παράθεσέ την παρακάτω!

### Ερώτηση

Στο πρόγραμμα με την κατάταξη των νέων μελών σε τμήματα για το τένις, για να βρεθεί το μέλος με τη μεγαλύτερη ηλικία έγινε αναζήτηση στην αρχική λίστα με όλα τα μέλη. Στην συνέχεια βέβαια φτιάχτηκαν δύο νέες λίστες, μία με τα ανήλικα μέλη και μία με τα ενήλικα μέλη. Θεωρείς ότι η αναζήτηση αυτή έγινε με τον βέλτιστο τρόπο;

1. Ναι, ήταν ο βέλτιστος τρόπος για να βρεθεί το μέλος με τη μεγαλύτερη ηλικία.
2. Όχι, υπάρχει άλλος βέλτιστος τρόπος για να γίνει η αναζήτηση αυτή.

### Ερώτηση

Αν στο πρόγραμμα με την κατάταξη των μαθητών ανά τμήμα ήθελες να βρεις τον μαθητή με την υψηλότερη βαθμολογία του τμήματος B1 σε ποια λίστα από τις 4 που δημιουργήθηκαν θα έψαχνες;

(Σκέψου την περίπτωση να είχες 1000 τέτοιες τιμές με ονόματα μαθητών και οι τελικές λίστες με τα τμήματα να περιείχαν περίπου από 200-500 τιμές ονομάτων. Που θα έψαχνες; Στη λίστα με τα 1000 ονόματα ή στη λίστα με τα 200-500 ονόματα; Γιατί;)

1. **B\_list**
2. **B1\_list**
3. **B2\_list**
4. **B3\_list**

### Δραστηριότητα Διακλάδωσης

#### Πρόβλημα: Από την Ονομαστική στη Γενική

Γράψε πρόγραμμα το οποίο να ζητάει από τον χρήστη να εισάγει ένα όνομα και να του εμφανίζει το όνομα αυτό στη γενική. Δηλαδή αν ο χρήστης εισάγει το: Γιώργος, θα πρέπει να εμφανιστεί τελικά το: Γιώργου. Οι πιο κοινοί κανόνες είναι οι εξής (ενδεχομένως να μην περιλαμβάνονται κάποια πολύ σπάνια αρχαία κυρίως ονόματα):

| Κατάληξη ονομαστικής | Κατάληξη Γενικής |
|----------------------|------------------|
| Γιώργ-ος             | Γιώργ-ου         |
| Κώστ-ας              | Κώστ-α           |
| Χάρ-ης               | Χάρ-η            |
| Βίκτ-ωρ              | Βίκτ-ωρος        |

|            |              |
|------------|--------------|
| Πειρίθ-ους | Πειρίθ-ου    |
| Συμε-ών    | Συμε-όνος    |
| Ελέν-η     | Ελέν-ης      |
| Γιάνν-α    | Γιάνν-ας     |
| Κλει-ώ     | Κλει-ώς      |
| Αμαρυλλ-ίς | Αμαριλλ-ίδος |

Προσπάθησε να λύσεις το παραπάνω πρόβλημα με τη χρήση λίστας λιστών.

### Tips:

- Φτιάξε λίστα λιστών  
`my_list=[['ος','ου',2],['ας','α',2],['ης','η',2],['ωρ','ωρος',2],['ους','ου',3],['ών','όνος',2],['η','ης',1],['α','ας',1],['ώ','ώς',1],['ίς','ίδος',2]]`, όπου σε κάθε υπολίστα υπάρχει η κατάληξη στην ονομαστική, η κατάληξη στην γενική και από πόσα γράμματα αποτελείται η κατάληξη στην ονομαστική για κάθε μία από τις παραπάνω κατηγορίες ονομάτων.
- Χρησιμοποίησε την κατάλληλη επαναληπτική δομή, `for...in...`, για να διατρέξεις όλα τα στοιχεία της λίστας, δηλαδή κάθε υπολίστα.
- Θυμήσου πως μπορείς να έχεις πρόσβαση στα στοιχεία μιας λίστας και μιας υπολίστας
- Θυμήσου πως μπορείς να πάρεις μέρος μιας συμβολοσειράς ώστε να αποσπάσεις την κατάληξη του ονόματος
- Με κατάλληλο έλεγχο της κατάληξης βρες σε ποια κατηγορία ανήκει το όνομα.
- Διαμόρφωσε τον νέο όνομα στην γενική παίρνοντας κατάλληλο μέρος του δοθέντος ονόματος και ενώνοντας το με την αντίστοιχη κατάληξη στη γενική

## Υποστηρικτική Δραστηριότητα

### Δομές Δεδομένων και Λίστες

Οι λίστες, δομές απλές κι ευέλικτες, έχουν βαρύνουσα σημασία στην ιστορία της Πληροφορικής. Η δεύτερη γλώσσα προγραμματισμού που αναπτύχθηκε, η LISP του John McCarthy, βασίζεται στις λίστες. Η Logo, με την σειρά της, μια ιστορική εκπαιδευτική γλώσσα προγραμματισμού, βασίζεται κυρίως στη LISP.



Οι λίστες είναι ένα παράδειγμα οργάνωσης των δεδομένων μας, έτσι ώστε να αποτελούν μια ενιαία συλλογή με συγκεκριμένη δομή. Υπάρχουν αναρίθμητοι τρόποι να οργανώσουμε τα δεδομένα μας, γι' αυτό και υπάρχουν και πάρα πολλές δομές δεδομένων. Ακόμα και οι πιο θεμελιώδεις έχουν εξωτικά ονόματα όπως στοίβες, ουρές, δέντρα, σωροί ή γράφοι. Ο τρόπος οργάνωσης καθορίζει πόσο εύκολο είναι να επεξεργαστούμε τα δεδομένα μας με συγκεκριμένο τρόπο. Γι' αυτό και επιλέγουμε τη δομή που θα χρησιμοποιήσουμε κάθε φορά ανάλογα με τον συγκεκριμένο τρόπο που θέλουμε να επεξεργαστούμε τα δεδομένα μας, στα πλαίσια ενός συγκεκριμένου προβλήματος.

Μέχρι να αντιμετωπίσει κανείς στοιχειωδώς περίπλοκα προβλήματα είναι δύσκολο να διακρίνει γιατί χρειάζεται να οργανωθούν τα δεδομένα – οι μεμονωμένες μεταβλητές φαίνονται υπεραρκετές, αλλά δεν είναι.

## Αρίθμηση από το Μηδέν

Υπάρχουν δύο ειδών άνθρωποι στον κόσμο: (1) αυτοί που ξεκινούν την αρίθμηση από το ένα και (1) αυτοί που ξεκινούν την αρίθμηση από το μηδέν. Αν πιάσατε το αστεράκι, έχετε καταλάβει αυτό το λεπτό σημείο του κεφαλαίου. Η χρήση της αρίθμησης από το μηδέν ξεκίνησε τη δεκαετία του '60 από ανάγκη, για καθαρά τεχνικούς λόγους και είναι αλήθεια ότι σε εκείνους που την συναντούν για πρώτη φορά φαίνεται από άβολη έως παρανοϊκή. Ωστόσο, έχει και σημαντικά πλεονεκτήματα. Ο ιδιόρρυθμος Edsger Dijkstra, ένας από τους ανθρώπους που συνέβαλαν καθοριστικά στην καθιέρωση της Πληροφορικής ως επιστήμη, έγραψε το 1982 τρεις ολόκληρες (χειρόγραφες) σελίδες με τίτλο "Γιατί η αρίθμηση πρέπει να ξεκινά από το μηδέν". Τώρα, η αρίθμηση από το μηδέν είναι πια από τα πολύ χαρακτηριστικά ιδιώματα της Πληροφορικής. Στο παιδικό βιβλίο "Lauren Ipsum" του Carlos Bueno, η αρίθμηση των κεφαλαίων ξεκινά από το μηδέν και η ηρωίδα, καθώς αρχίζει την περιπέτειά της, συναντά μια πινακίδα όπου το πρώτο μίλι της διαδρομής είναι φυσικά... το μηδενικό.

Μπορείς να παρακολουθήσεις και ένα [βίντεο](#) σχετικό με τις δομές δεδομένων.

## Θεωρία

### Γενικά....

Λίστα είναι ένας τύπος μεταβλητής όπου μπορούν να τις ανατεθούν παραπάνω από μία τιμές.

Οι τιμές αυτές εσωκλείονται σε αγκύλες και χωρίζονται με κόμμα.

Οι τιμές αυτές δεν είναι αναγκαστικό να είναι του ίδιου τύπου. Μπορεί μια λίστα να περιέχει μόνο αριθμητικές τιμές, μπορεί όμως να περιέχει μαζί αριθμητικές τιμές, λογικές τιμές, συμβολοσειρές κ.ο.κ. Επίσης μια λίστα δεν έχει σταθερό περιεχόμενο. Οι τιμές που διαθέτει καθώς και το μέγεθός της μπορούν να τροποποιηθούν.

Παραδείγματα:

**nums** = [3,5,8,13,21,34,55]

**users** = ['Αργυρώ', 'Γιάννης', 'Κυριάκος']

**my\_list** = ["Πέμπτη", 29, "Απριλίου"]

**empty** = [ ]

### >Πρόσβαση

σε στοιχείο της λίστας αποκτάς γράφοντας το όνομα της λίστας και τη θέση του στοιχείου μέσα σε [ ]. Η αρίθμηση των θέσεων μπορεί να ξεκινάει είτε από την αρχή όπου πρώτο

στοιχείο βρίσκεται στη θέση 0, είτε ανάποδα, δηλαδή από το τελευταίο στοιχείο ξεκινώντας το μέτρημα από το -1

Σε περίπτωση που αναφερθούμε σε μια θέση που δεν υπάρχει στη λίστα, δηλαδή σε έναν αριθμό μεγαλύτερο ή ίσο με το πλήθος των στοιχείων της τότε προκύπτει σφάλμα.

Παραδείγματα:

`nums[3] = 1`, δίνει στο τέταρτο στοιχείο την τιμή 1

`nums[-1] = 89`, δίνει στο τελευταίο στοιχείο την τιμή 89

`print(nums[0])`, εμφανίζει το πρώτο στοιχείο της λίστας

### >Έλεγχος ύπαρξης τιμής

σε μία λίστα γίνεται με τον υπαρξιακό τελεστή `in`, ελέγχει δηλαδή αν μία τιμή βρίσκεται στη λίστα και επιστρέφει αντίστοιχα την τιμή `True` αν βρίσκεται ή `False` σε αντίθετη περίπτωση.

Παράδειγμα:

```
if 21 in nums:
    print('Το βρήκα')
```

### >Συνάρτηση `len()`

επιστρέφει το **πλήθος** των στοιχείων μιας ακολουθίας, όπως η λίστα.

Παράδειγμα:

```
if len(nums) == 0:
    print("Η λίστα είναι κενή")
```

### >Μέθοδος `append()`

**προσθέτει** ένα νέο στοιχείο στο τέλος της λίστας.

Παραδείγματα:

```
nums.append(89)
users.append('Μυρσίνη')
```

### >Μέθοδος `insert()`

Δέχεται ως παράμετρο τη θέση και το στοιχείο εισαγωγής και **εισάγει** ένα νέο στοιχείο στη θέση αυτή της λίστας.

Παράδειγμα:

```
users.insert(1, 'Μελίνα'),
```

εισάγει την τιμή 'Μελίνα' στη 2η θέση της λίστας

### >Μέθοδος `remove()`

Δέχεται ως παράμετρο ένα στοιχείο της λίστας και το **αφαιρεί**. Αν το στοιχείο δεν υπάρχει στη λίστα τότε προκύπτει σφάλμα.

Παράδειγμα:

```
nums.remove(13)
```

αφαιρεί τη τιμή 13 από τη λίστα `nums`

### >Μέθοδος `pop()`

**αφαιρεί** το τελευταίο στοιχείο της λίστας.

Παράδειγμα:

```
lastnum = nums.pop()
```



αφαιρεί την τελευταία τιμή από τη λίστα `nums` και καταχωρείται στην μεταβλητή `lastnum`

### >Μέθοδος `index()`

δέχεται ως παράμετρο ένα στοιχείο της λίστας, αναζητά και **επιστρέφει τη θέση** του στοιχείου αυτού μέσα στη λίστα. Αν το στοιχείο δεν υπάρχει στη λίστα τότε προκύπτει σφάλμα.

Παράδειγμα:

```
pos = nums.index('Στέλλα')
```

### >Μέθοδος `reverse()`

εφαρμόζεται σε μια λίστα κι αντιστρέφει τα στοιχεία της.

Παράδειγμα:

```
nums.reverse() αντιστρέφει τη λίστα nums.
```

### >Συνάρτηση `sorted()`

δέχεται σαν παράμετρο μια λίστα και επιστρέφει μια νέα ταξινομημένη λίστα, χωρίς να πειράζει την αρχική.

### >Η μέθοδος `sort()`

εφαρμόζεται σε μια λίστα και ταξινομεί τα στοιχεία της.

### >Διάσχιση λίστας

Η εντολή `for` είναι μια εντολή επανάληψης που διατρέχει τα στοιχεία μιας ακολουθίας τιμών, όπως μια λίστα, με τη σειρά που εμφανίζονται. Σε κάθε επανάληψη η τιμή του επόμενου στοιχείου της ακολουθίας ανατίθεται σε μια μεταβλητή απαρίθμησης που χρησιμοποιούμε στην `for`. Η μεταβλητή αυτή ορίζεται αυθαίρετα από τον προγραμματιστή.

Παράδειγμα:

```
for ar in nums:  
    print(ar)
```

### >Τμήμα λίστας

Γράφοντας `name_list[a,b,v]`, παίρνουμε ένα τμήμα από τη λίστα `name_list`, το οποίο ξεκινάει από το στοιχείο που βρίσκεται στη θέση `a` μέχρι το στοιχείο που βρίσκεται στη θέση `b-1`, και με βήμα `v`, δηλαδή ανά πόσα στοιχεία θα περιλαμβάνονται στο τμήμα αυτό και με ποια φορά.

Αν παραλειφθεί η αρχική θέση, το τμήμα αυτό ξεκινάει από το πρώτο στοιχείο της λίστας, αν παραλειφθεί η τελική θέση φτάνει μέχρι το τέλος της λίστας, αν παραλειφθεί το βήμα τότε παίρνει την τιμή `+1`.

Παραδείγματα:

```
nums[1:4], παίρνει τμήμα της λίστας nums από τη θέση 1 έως και τη θέση 3 (το στοιχείο στη θέση 4 δεν περιλαμβάνεται).
```

```
nums[:2], παίρνει τμήμα της λίστας nums ξεκινώντας από την αρχική θέση μέχρι και τη θέση 1 (το στοιχείο στη θέση 2 δεν περιλαμβάνεται).
```

```
nums[::-1], αντιστρέφει τα στοιχεία της λίστας nums, ή αλλιώς παίρνει τμήμα της λίστας nums ξεκινώντας από την τελευταία θέση ως την αρχική με βήμα -1
```

```
nums[4:2:-1], παίρνει τμήμα της λίστας nums ξεκινώντας από τη θέση 4 έως τη θέση 3 (το στοιχείο στη 2 δεν περιλαμβάνεται) με βήμα -1
```

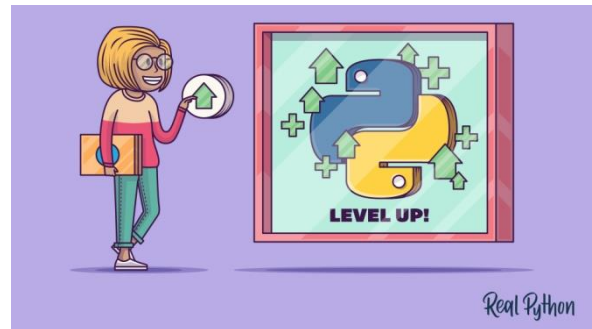
## Μάθημα 6: Η Δομή Επανάληψης Ασκήσεις

Στα προηγούμενα μαθήματα έμαθες για τις λίστες στην Python αλλά και την επαναληπτική δομή `for` μέσω της οποίας μπορούσες να διασχίσεις μία λίστα. Στο μάθημα αυτό θα μάθεις παραπάνω λεπτομέρειες για την επαναληπτική δομή `for` καθώς και για την επαναληπτική δομή `while`.

### Μαθησιακοί Στόχοι:

Εκτελώντας τις επόμενες δραστηριότητες θα μπορείς:

- Να αναλύουν τη λειτουργία της επαναληπτικής δομής `for` και `while`.
- Να προσδιορίζεις πόσες φορές εκτελείται μία δομή επανάληψης.
- Να εξηγείς πως μεταβάλλονται οι τιμές μεταβλητών σε κάθε διαδοχική επανάληψη.
- Να εντοπίζεις τη συνθήκη τερματισμού μιας επαναληπτικής δομής.



### Ερωτήσεις Κατανόησης:

Έστω ότι οι παρακάτω εντολές αποτελούν κομμάτι ενός μεγαλύτερου προγράμματος και `L` μία λίστα με αριθμητικές τιμές που έχει οριστεί σε κάποιο σημείο παραπάνω του προγράμματος αυτού:

1. `sum = 0.0`
2. `for number in L :`
3. `sum = sum + number`
4. `average = sum / len( L )`

**Ερώτηση 1:** Ποια είναι η λειτουργία των παραπάνω εντολών; Αν έστω `L[1,2,5,8]`, η μεταβλητή `average` τι τιμή θα πάρει;

**Ερώτηση 2:** Η εντολή 3 πόσες φορές εκτελείται;

1. Αν έστω `x` το πλήθος των στοιχείων που περιέχει η λίστα `L`, τότε θα εκτελεστεί `x` φορές.
2. Μία φορά

**Ερώτηση 3:** Αν `L` μια λίστα που δεν έχει καθοριστεί από τον προγραμματιστή, αλλά διαμορφώνεται κατά την εκτέλεση του προγράμματος και οι τιμές και το μέγεθος καθορίζεται από το χρήστη που εκτελεί το πρόγραμμα αυτό, θα μπορούσες να διαπεράσεις την λίστα και να αθροίσεις τα στοιχεία της με έναν μη επαναληπτικό τρόπο;

1. Όχι (Αν όχι, για ποιο λόγο το πιστεύεις αυτό...)
2. Ναι (Αν ναι, σκέψου μια ενδεικτική λύση...)

## Δραστηριότητα 1

### Η συνάρτηση range()

Η συνάρτηση range( A, M, B ) επιστρέφει μια λίστα αριθμών ξεκινώντας με τον αριθμό A μέχρι το M με βήμα B. Το M δεν συμπεριλαμβάνεται στη λίστα. Αν το

Παρακάτω υπάρχουν τα εξής 2 προγράμματα:

#### Πρόγραμμα 1

```
1. for item in [1,2,3,4,5,6,7,8] :  
2.     print (item)  
3.     print('Hi!')
```

#### Πρόγραμμα 2

```
1. for item in ..... :  
2.     print (item)  
3.     print('Hi!')
```

Αφού εκτελέσεις το πρόγραμμα 1 απάντησε στις παρακάτω ερωτήσεις:

Το πρόγραμμα 1 εκτυπώνει τους αριθμούς 1,2,3,4,5,6,7,8 οι οποίοι αποτελούν τιμές μιας λίστας όπως γνωρίζεις από προηγούμενο μάθημα. Συμπλήρωσε κατάλληλα το πρόγραμμα 2 ώστε να εκτυπώνονται οι ίδιοι αριθμοί χρησιμοποιώντας την συνάρτηση range().

**Ερώτηση 1:** Πόσες φορές πιστεύεις ότι εκτελούνται οι εντολές 2 και 3 στο πρόγραμμα 1 και πόσες στο πρόγραμμα 2;

---

Εκτέλεσε τα δύο προγράμματα στο περιβάλλον οπτικοποίησης <https://pythontutor.com/visualize.html#mode=edit>

**Ερώτηση 2:** Τελικά πόσες φορές εκτελούνται; Γιατί συμβαίνει αυτό;

---

Έστω οι εξής εντολές:

```
1. for i in range(12,23):  
2.     print("Hello!")
```

**Ερώτηση 3:** Ποιο πιστεύεις είναι το αποτέλεσμα της εκτέλεσης των εντολών και πόσες φορές θα εκτελεστεί η εντολή 2;

---

Εκτέλεσε τις εντολές στην εφαρμογή οπτικοποίησης.

**Ερώτηση 4:** Τελικά ποιο ήταν το αποτέλεσμα και πόσες φορές εκτελέστηκε η εντολή 2, γιατί;

## Δραστηριότητα Διακλάδωσης

Μελέτησε τα παρακάτω δύο προγράμματα:

## Πρόγραμμα 1

```
1. numbers_list = [ 6, 28, 496, 8128 ]
2. for item in numbers_list :
3.     print (item )
```

## Πρόγραμμα 2

```
1. numbers_list = [ 6, 28, 496, 8128 ]
2. for index in range(0,4) :
3.     print (numbers_list[index] )
```

**Ερώτηση 1:** Ποια πιστεύεις είναι η λειτουργία των παραπάνω 2 προγραμμάτων και ποιο το αποτέλεσμα της εκτέλεσης;

---

Εκτέλεσε τα δύο προγράμματα στην εφαρμογή οπτικοποίησης.

**Ερώτηση 2:** Ποιο ήταν τελικά το αποτέλεσμα της εκτέλεσης των παραπάνω δύο προγραμμάτων; Θεωρείς ότι τα προγράμματα αυτά είναι ισοδύναμα;

---

**Ερώτηση 3:** Πόσες φορές εκτελέστηκε η εντολή 3 στα προγράμματα 1 και 2; Από τι εξαρτήθηκε ο αριθμός αυτός της επαναληπτικής εκτέλεσης στο πρόγραμμα 1 και από τι στο πρόγραμμα 2;

---

**Ερώτηση 4:** Τι τιμές πήρε η μεταβλητή item στο πρόγραμμα 1 και τι τιμές πήρε η μεταβλητή index στο πρόγραμμα 2;

---

## Ερώτηση

Ένας τρόπος εύρεσης του μεγαλύτερου αριθμού ανάμεσα σε τρεις είναι ο ακόλουθος:

```
1. a = int(input ('a = '))
2. maximum = a
3. b = int(input ('b = '))
4. if b > maximum :
5.     maximum = b
6. c = int(input ('c = '))
7. if c > maximum :
8.     maximum = c
9. print("Ο μεγαλύτερος αριθμός είναι ο :", maximum)
```

Έστω ότι θέλεις να βρεις τον μεγαλύτερο αριθμό ανάμεσα σε 100 αριθμούς. Ποιον αλγόριθμο θα ακολουθούσες για τη λύση του προβλήματος;

1. Απλά θα συνεχίσω το παραπάνω πρόγραμμα με την ίδια λογική και θα προσθέσω άλλες 97 εντολές εισόδου τιμής και άλλους 97 ελέγχους για τους υπόλοιπους 97 νέους αριθμούς που θα δώσει ο χρήστης. Ο κώδικας θα είναι μεγάλος αλλά αποτελεσματικός!
2. Θα χρησιμοποιήσω την επαναληπτική δομή for. Θα παίρνω κάθε φορά τον νέο αριθμό, θα τον συγκρίνω με τον maximum και η διαδικασία αυτή θα επαναλαμβάνεται μέχρι να φτάσω στον 100οστό αριθμό. Θα γίνουν οι συνολικά 99 συγκρίσεις που απαιτούνται χωρίς να χρειαστεί να γράψω 300 γραμμές κώδικα.
3. Δεν ξέρω... Δεν απαντώ!

## Πρόβλημα

Πρόβλημα εύρεσης μεγίστου

Στον προηγούμενο κώδικα οι μεταβλητές `b` και `c` αντικαθίστανται με την μεταβλητή `new_number`. Όπως μπορείς να δεις στον παρακάτω κώδικα κάποιες εντολές επαναλαμβάνονται.

```
1. a = int(input("a = "))
2. maximum = a
3. new_number = int(input("new number = "))
4. if new_number > maximum :
5.     maximum = new_number
6. new_number = int(input("new number = "))
7. if new_number > maximum :
8.     maximum = new_number
9. print("Ο μεγαλύτερος αριθμός είναι ο :", maximum)
```

Προσπάθησε να λύσεις το πρόβλημα της εύρεσης του μεγαλύτερου αριθμού ανάμεσα σε 100 αριθμούς χρησιμοποιώντας κατάλληλα την επαναληπτική δομή `for...in range()`.

>Επειδή οι αριθμοί πρέπει να εισάγονται από το πληκτρολόγιο δοκίμασε πρώτα για 10 αριθμούς και μετά τροποποίησέ το για 100.

## Δραστηριότητα Διακλάδωσης

### Πρόβλημα ενός κλάδου: Τα πολλαπλάσια του 9

Να φτιαχτεί πρόγραμμα που να ελέγχει όλους τους τριψήφιους ακέραιους αριθμούς [100,999] και να εμφανίζει τα πολλαπλάσια του 9. Οι αριθμοί που είναι πολλαπλάσια του 9, διαιρούνται ακριβώς με αυτό, δηλαδή το υπόλοιπο της διαίρεσης του αριθμού με το 9, είναι 0. Στο τέλος να τυπώνει και το πλήθος των αριθμών αυτών.

### Πρόβλημα δεύτερου κλάδου: Πλήθος φωνηέντων

Να γραφτεί πρόγραμμα το οποίο να μετρά και να τυπώνει το πλήθος των φωνηέντων σε μια ακολουθία χαρακτήρων, που θα δοθεί ως είσοδος από το πληκτρολόγιο.

Προσοχή! Ανάλογα με το τι χαρακτήρες, λατινικούς, ελληνικούς ή και τα δύο, επιτρέπουμε στον χρήστη να εισάγει πρέπει να λαμβάνονται υπόψη και οι κατάλληλοι έλεγχοι!

Tips!

- Αν `my_word` μεταβλητή που περιέχει μια συμβολοσειρά
  - `len(my_word)`-> επιστρέφει το μήκος της συμβολοσειράς που έχει η `my_word`
  - `my_word[i]`-> έχω πρόσβαση στον χαρακτήρα που βρίσκεται στη θέση `i` της συμβολοσειράς
- Για να κάνουμε κατάλληλους ελέγχους χρησιμοποιούμε
  - την απλή δομή επιλογής
    - `if <συνθήκη>`:

- εντολές
- ή την πολλαπλή δομή επιλογής
  - if <συνθήκη>:
  - εντολές
  - elif <συνθήκη>:
  - εντολές
- Για να δημιουργήσουμε πιο σύνθετες συνθήκες ελέγχου χρησιμοποιούμε λογικούς τελεστές or, and, not .

### Δραστηριότητα Διακλάδωσης

Πήγαινε στον παρακάτω σύνδεσμο:

<https://trinket.io/python3/8d89f8bd9e?outputOnly=true&runOption=run>

Και αφού εκτελέσεις το πρόγραμμα, απάντησε στις παρακάτω ερωτήσεις.

**Ερώτηση 1:** Πόσες φορές το πρόγραμμα σε ρώτησε αν θες να μαντέψεις αν ο επόμενος αριθμός είναι πρώτος ή όχι;

---

**Ερώτηση 2:** Αυτός ο αριθμός πιστεύεις ότι ήταν προκαθορισμένος από τον προγραμματιστή ή εξαρτιόταν από κάτι άλλο;

---

**Ερώτηση 3:** Πότε τερμάτισε το πρόγραμμα ή με άλλα λόγια από τι εξαρτήθηκε αυτός ο τερματισμός;

---

**Ερώτηση 4:** Για να βρεις αν ένας αριθμός είναι πρώτος ή όχι, τι πράξεις θα έκανες ή τι αλγόριθμο θα ακολουθούσες; Στον αλγόριθμο που σκέφτηκες υπάρχουν πράξεις που επαναλαμβάνονται;

---

Παρακάτω είναι ο κώδικας του προβλήματος που έτρεξες προηγουμένως.

```
1. import random
2. question=input("Γεια σου! Θα ήθελες να κάνεις μια μαντεψιά; ΝΑΙ ή ΟΧΙ")
3. while question=='ΝΑΙ':
4.     #τυχαίος αριθμός
5.     number=random.randint(3,300)
6.     #
7.     while number%2==0:
8.         number=random.randint(3,300)
9.     print('Πιστεύεις ότι ο', number, end=' ')
10.    answer=input(' είναι πρώτος; ΝΑΙ ή ΟΧΙ')
11.    #εύρεση διαιρετών
12.    divisors_list=[]
13.    for i in range(2, number):
14.        if number%i==0:
```

```
15.     divisors_list.append(i)
16.     #εμφάνισε κατάλληλα μηνύματα ανάλογα με τον αν η μαντεψιά ήταν σωστή ή όχι
17.     if len(divisors_list)==0 and answer=='NAI':
18.         print('Σωστά! Ο ', number, ' είναι πρώτος αριθμός με μοναδικούς διαιρέτες το 1
           και τον εαυτό του')
19.     elif len(divisors_list)!=0 and answer=='OXI':
20.         print('Σωστά! Ο ', number, ' είναι δεν είναι πρώτος αριθμός. Οι διαιρέτες του
           εκτός από το 1 και τον εαυτό του είναι:', divisors_list)
21.     elif len(divisors_list)==0 and answer=='OXI':
22.         print('Λάθος! Ο ', number, ' είναι πρώτος αριθμός με μοναδικούς διαιρέτες το 1
           και τον εαυτό του')
23.     elif len(divisors_list)!=0 and answer=='NAI':
24.         print('Λάθος! Ο ', number, ' είναι δεν είναι πρώτος αριθμός. Οι διαιρέτες του
           εκτός από το 1 και τον εαυτό του είναι:', divisors_list)
25.     question=input("Γεια σου! Θα ήθελες να κάνεις μια μαντεψιά; ΝΑΙ ή ΟΧΙ")
26. print('OK! Γεια!!!')
```

Μελέτησε τον και απάντησε στις παρακάτω ερωτήσεις.

Μπορείς να το εκτελέσεις [εδώ](#) για να κατανοήσεις καλύτερα την ροή του προγράμματος.

Ερώτηση 5: Η μεταβλητή question τι τιμές παίρνει στο πρόγραμμα και ποιος είναι ο ρόλος της;

---

Ερώτηση 6: Στην εντολή 3 με το while ορίζεται και ξεκινάει μια επαναληπτική δομή. Ποιες εντολές περιέχονται στην δομή αυτή; Πότε τερματίζει αυτή η επαναληπτική δομή;

---

Ερώτηση 7: Στις εντολές 7 και 8 υπάρχει άλλη μια μικρότερη και εμφωλευμένη δομή επιλογής. Ποια είναι η λειτουργία αυτής της επαναληπτικής δομής; Πότε τερματίζει;

---

Ερώτηση 8: Ποια είναι η λειτουργία των εντολών 13-15; Η μεταβλητή i τι τιμές παίρνει και ποια είναι η λειτουργία της;

---

Ερώτηση 9: Η επαναληπτική δομή που ορίζεται από τη for πότε τερματίζει; Πότε εκτελείται η εντολή 15 ή με άλλα λόγια, πότε προστίθεται μια τιμή στην λίστα divisors\_list;

---

## Πρόβλημα

### Ο αλγόριθμος του Ευκλείδη

Ο Ευκλείδης, στο έβδομο βιβλίο των Στοιχείων του, περιγράφει μια μέθοδο για την εύρεση του μέγιστου κοινού διαιρέτη δύο ακεραίων. Μπορεί κανείς να βρει πολλές διαφορετικές εκδοχές του αλγορίθμου, αλλά βασικά ο αλγόριθμος μπορεί να συνοψιστεί ως εξής:

Αφαίρεσε τον μικρότερο αριθμό από τον μεγαλύτερο

και επανάλαβε τη διαδικασία μέχρι οι δύο αριθμοί να γίνουν ίσοι.

Η τιμή των δύο αριθμών όταν τελειώσει η διαδικασία είναι ο μέγιστος κοινός διαιρέτης τους.

Γράψτε ένα πρόγραμμα το οποίο ζητάει από το χρήστη δύο ακέραιους αριθμούς και υπολογίζει τον μέγιστο κοινό διαιρέτη τους, χρησιμοποιώντας τη μέθοδο του Ευκλείδη.

Τσέκαρε την ορθότητα της λύσης σου χρησιμοποιώντας τα εξής ζευγάρια αριθμών:

- 6,35 ->ΜΚΔ=1
- 105,252 ->ΜΚΔ=21
- 90,25->ΜΚΔ=5

## Πρόβλημα Διακλάδωσης

### Πέτρα-Ψαλίδι-Χαρτί

Να κατασκευάσεις ένα πρόγραμμα το οποίο θα παίζει επαναληπτικά το γνωστό παιχνίδι «Πέτρα-Ψαλίδι-Χαρτί» με το χρήστη. Σε κάθε γύρο, το πρόγραμμα θα ζητά από το χρήστη την επιλογή του (1 για την Πέτρα, 2 για το Ψαλίδι ή 3 για το Χαρτί) και στη συνέχεια θα εμφανίζει τη δική του επιλογή (του υπολογιστή) και θα ανακοινώνει το νικητή του γύρου. Αν ο χρήστης πληκτρολογήσει οτιδήποτε άλλο εκτός από τις τρεις έγκυρες επιλογές τότε θεωρείται ότι επιθυμεί να τερματίσει το παιχνίδι. Στο τέλος θα εμφανίζει στον χρήστη σε πόσα από τα συνολικά παιχνίδια κέρδισε.

## Δραστηριότητα Διακλάδωσης

Πολλές φορές στην ζωή μας εμείς οι άνθρωποι καλούμαστε να αναζητήσουμε μία τιμή. Το ίδιο συμβαίνει και στον προγραμματισμό. Η αναζήτηση μιας συγκεκριμένης τιμής είναι ένα πολύ συχνό πρόβλημα που τίθεται προς επίλυση.

Το πρόβλημα είναι το εξής: Έστω πρόγραμμα το οποίο δημιουργεί τυχαία τιμή από το 1-32 και πρέπει ο χρήστης να μαντέψει αυτήν την αριθμητική τιμή το πολύ σε 4 προσπάθειες. Ανάλογα με την απάντηση του χρήστη το πρόγραμμα τον βοηθά αν ο τυχαίος αριθμός είναι μικρότερος ή μεγαλύτερος από τον αριθμό που έδωσε.

Έχουμε τα εξής δύο προγράμματα:

### Πρόγραμμα 1

|   |                                                  |
|---|--------------------------------------------------|
| 1 | <code>import random</code>                       |
| 2 | <code>secret = random.randint(1,32)</code>       |
| 3 | <code>print("Μάντεψε τον αριθμό: 1 - 32")</code> |
| 4 | <code>found = False</code>                       |
| 5 | <code>tries = 4</code>                           |



```

6 while not found and tries > 0:
7     print("Απομένουν",tries,"προσπάθειες.")
8     tries = tries - 1
9     number = int(input())
10    if number > secret:
11        print("Λάθος. Είναι μικρότερος.")
12    elif number < secret:
13        print("Λάθος. Είναι μεγαλύτερος.")
14    else:
15        print("Σωστά!")
16        found = True
17 if found==False:
18    print("Ήταν ο", secret)

```

## Πρόγραμμα 2

```

1 import random
2 secret = random.randint(1,32)
3 print("Μάντεψε τον αριθμό: 1 - 32")
4 found = False
5 for tries in range(4,0,-1):
6     print("Απομένουν", tries, "προσπάθειες.")
7     number = int(input())
8     if number > secret:
9         print("Λάθος. Είναι μικρότερος.")
10    elif number < secret:
11        print("Λάθος. Είναι μεγαλύτερος.")
12    else:
13        print("Σωστά!")
14        found=True
15        break
16 if found==False:
17    print("Ήταν ο", secret)

```

Μελέτησε τα παραπάνω προγράμματα

**Ερώτηση 1:** Πόσες επαναλήψεις το πολύ πιστεύεις θα γίνουν στο πρόγραμμα 1 και πόσες στο πρόγραμμα 2; Γιατί;

---

**Ερώτηση 2:** Η μεταβλητή `found` τι τύπου είναι και ποιος πιστεύεις είναι ο ρόλος της μέσα στα δύο προγράμματα;

---

**Ερώτηση 3:** Στο πρόγραμμα 1 η επαναληπτική δομή `while` πότε πιστεύεις θα τερματίσει; Στο πρόγραμμα 2 πότε πιστεύεις θα σταματήσει η επαναληπτική δομή `for` που υπάρχει;

---

Εκτέλεσε τα δύο προγράμματα [εδώ](#) και απάντησε στις παρακάτω ερωτήσεις.

**Ερώτηση 4:** Όταν δεν κατάφερες να μαντέψεις τον αριθμό, τελικά πόσες φορές συνολικά προσπάθησες να δώσεις αριθμό στο πρόγραμμα 1 και πόσες στο πρόγραμμα 2;

---

**Ερώτηση 5:** Η μεταβλητή `found` πότε παίρνει την τιμή `True` και ποιος είναι ο ρόλος της μέσα στα δύο προγράμματα;

---

**Ερώτηση 6:** Στο πρόγραμμα 1 η επαναληπτική δομή `while` που υπάρχει πότε τερματίζει; Τι τιμές πρέπει να έχουν οι μεταβλητές `found` και `tries` για να συνεχίσει τις επαναλήψεις η δομή `while`;

---

**Ερώτηση 7:** Στο πρόγραμμα 2 πότε θα σταματήσει η επαναληπτική δομή `for` που υπάρχει; Ποιος είναι ο ρόλος της εντολής `break` που υπάρχει;

---

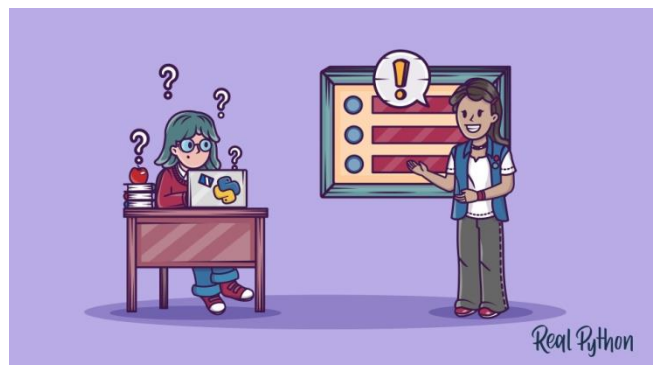
**Ερώτηση 8:** Τελικά ποιος τρόπος αναζήτησης τιμής σου φάνηκε πιο εύκολος;

---

## Πρόβλημα στην ίδια Διακλάδωση

### Πρόγραμμα Εβδομάδας

Γράψε πρόγραμμα το οποίο να ζητά από τον χρήστη να εισάγει ημέρα βδομάδας και δραστηριότητα και να δημιουργεί λίστα με τις τιμές αυτές. Οι τιμές για τη μέρα και τη δραστηριότητα θα αποθηκεύονται σε μορφή υπολίστας στη λίστα `date_list`. Ο χρήστης πρέπει να δώσει εισόδο για κάθε μέρα της βδομάδας χωρίς να απαιτείται να εισάγει τις μέρες με την σειρά. Το πρόγραμμα λοιπόν πρέπει να ελέγχει αν η μέρα που δόθηκε υπάρχει στη λίστα ή όχι. Αν δεν υπάρχει εισάγονται οι νέες τιμές στην λίστα ενώ αν υπάρχει εμφανίζει μήνυμα λάθους και συνεχίζει να ζητάει από τον χρήστη μέρα και δραστηριότητα. Το πρόγραμμα



σταματάει να ρωτάει τον χρήστη όταν η λίστα περιέχει δραστηριότητες για κάθε μέρα της βδομάδας. Στο τέλος να εμφανίζει τις τιμές της λίστας `date_list`.

## Υποστηρικτική Δραστηριότητα

### Θεωρία

Πολλές φορές ένα πρόγραμμα πρέπει να εκτελέσει κάποια εντολή πολλές φορές. Τύπωσε 20 φορές τη λέξη 'Καλημέρα', ζήτη ως είσοδο 30 αριθμούς, έλεγξε τις ηλικίες 100 ατόμων κ.ο.κ. Έτσι για να μη χρειάζεται οι προγραμματιστές να γράψουν όλες αυτές τις εντολές αξιοποιούν τις επαναληπτικές δομές ώστε οι εντολές να εκτελεστούν να μεν επαναληπτικά αλλά έχοντας γραφτεί μία φορά από τον προγραμματιστή.

**Βέβαια χρειάζεται προσοχή ώστε να χρησιμοποιηθεί η κατάλληλη επαναληπτική δομή στο εκάστοτε πρόβλημα.**

### Η δομή επανάληψης `for`:

Η επαναληπτική δομή `for` χρησιμοποιείται όταν έχουμε ως δεδομένο τον αριθμό των επαναλήψεων. Για παράδειγμα

- όταν διατρέχουμε μία λίστα με 20 στοιχεία ξέρουμε ότι επαναλήψεις που θα γίνουν θα είναι 20 αφού τόσα είναι τα στοιχεία της λίστας
- όταν θέλουμε να βρούμε τον μεγαλύτερο ανάμεσα σε 100 αριθμούς ξέρουμε ότι θα πρέπει να κάνουμε τους αντίστοιχους ελέγχους 100 φορές για τους αντίστοιχους αριθμούς
- όταν θέλουμε να βρούμε το μέσο όρο 1000 αριθμών ξέρουμε ότι θα κάνουμε 1000 προσθέσεις

### Γενική Μορφή 1

1. `<εντολές 1>`
2. `for i in my_list :`
3. `<εντολές 2>`
4. `<εντολές 3>`

### Γενική Μορφή 2

1. `<εντολές 1>`
2. `for i in range(A,B,M):`
3. `<εντολές 2>`
4. `<εντολές 3>`

Για κάθε τιμή που βρίσκεται στην `my_list` ή στην λίστα που επιστρέφει η συνάρτηση `range()`, εκτελούνται οι `<εντολές 2>`. Η μεταβλητή `i` παίρνει κάθε φορά και από μια τιμή της λίστας. Όταν πια δεν υπάρχει άλλη τιμή μέσα στη λίστα που διασχίζεται, τότε η επαναληπτική δομή τερματίζεται και εκτελούνται οι `<εντολές 3>`.

Παράδειγμα:

### Πρόγραμμα/Εντολές

```
print("Hello")
for i in range(0,3):
    print("Good morning")
print("Bye")
```

### Εκτέλεση και αποτέλεσμα στην Οθόνη

```
Hello
Good morning
Good morning
Good morning
Bye
```

Η συνάρτηση `range( A, M, B )` επιστρέφει μια λίστα αριθμών που ξεκινάνε από τον αριθμό `A` μέχρι το `M`, χωρίς να περιλαμβάνεται το `M`, με βήμα `B`. Όταν το `A` και το `B` παραλείπονται τότε εννοείται ότι το `A=0` και το `B=1`

### Η δομή επανάληψης `while`:

Ωστόσο, υπάρχουν και περιπτώσεις που **δεν γνωρίζουμε εξ αρχής το πλήθος των επαναλήψεων** που θα εκτελεστούν. Σε αυτές τις περιπτώσεις χρησιμοποιούμε τη δομή επανάληψης `while`. Η εντολή `while` εκτελεί επαναλαμβανόμενα ένα σύνολο αριθμών όσο μία συνθήκης είναι αληθής, λέμε δηλαδή ότι έχουμε επανάληψη υπό συνθήκη.

Για παράδειγμα:

- διάβασε από το πληκτρολόγιο αριθμούς μέχρι να δοθεί το 0 και στη συνέχεια εμφάνισέ τους.

### Γενική Μορφή

1. <εντολές 1>
2. `while <συνθήκη ελέγχου>` :
3. <εντολές 2>
4. <εντολές 3>

Η <συνθήκη ελέγχου> κάθε φορά ελέγχεται και **όσο** είναι αληθής, **εκτελούνται** οι <εντολές 2>. Όταν <συνθήκη ελέγχου> γίνει ψευδής, η επαναληπτική δομή τερματίζεται και εκτελούνται οι <εντολές 3>. Η <συνθήκη ελέγχου> είναι λογικό να περιέχει μια μεταβλητή, η τιμή της οποίας θα μεταβάλλεται με κάποιο τρόπο στις <εντολές 2>.

### Η εντολή `break`:

Η εντολή `break` έχει την ιδιότητα να **τερματίζει μία επαναληπτική δομή**. Οι εντολές που ανήκουν στην επαναληπτική δομή εκτελούνται επαναλαμβανόμενα όσο ισχύει η συνθήκη της επανάληψης. Ωστόσο πολλές φορές, ανάλογα με το πρόβλημα που έχουμε να αντιμετωπίζουμε, επιθυμούμε η δομή αυτή να τερματίσει νωρίτερα για κάποιο λόγο.

Για παράδειγμα, το παρακάτω πρόγραμμα έχει ως λειτουργία να εμφανίζει στην οθόνη κάθε χαρακτήρα της συμβολοσειράς PYTHON, μέχρι όμως τον χαρακτήρα H.

### Κώδικας/Εντολές

```
s = 'PYTHON'  
# Using for loop  
for letter in s:  
    print(letter)  
    # break the loop as soon it sees 'H'  
    if letter == 'H' :  
        break  
print("Out of for loop")
```

### Εκτέλεση και Αποτέλεσμα στην Οθόνη

```
P  
Y  
T  
Out of for loop
```

## ΑΝΑΦΟΡΕΣ

- [1] Becker D., &. Dwyer M (1994). Using hypermedia to provide learner control. *Journal of Educational Multimedia and Hypermedia*, 3(2), pp. 155-172.
- [2] Παπαδάκης Σ. Πασχάλης Γ., Ρώσσιου Ε., Δόβρος Ν. (2010). Εκπαίδευση και Πρακτική με το Ελεύθερο Ανοικτό Διαδικτυακό Σύστημα Διαχείρισης Μαθησιακών Δραστηριοτήτων (LAMS) Στο Κολτσάκης Β., Σαλονικίδης Ι., Δοδοντσής Μ. (επιμ.), *Πρακτικά 2ο Πανελλήνιο Εκπαιδευτικό Συνέδριο Ημαθίας, Βέροια-Νάουσα, 23-25 Απριλίου 2010*. <https://www.ekped.gr/praktika10/gen/141.pdf>. [Προσπελάστηκε 29/10/2021]
- [3] Kopper R., Tattersall C. (2005). *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training*. Springer, 2005
- [4] Morgan, H. (2014). Maximizing student success with differentiated learning. *The Clearing House: A Journal of Educational Strategies, Issues and Ideas*, 87(1), 34–38.
- [5] Kelly, K. (1997). Role of lesson plans in instructional planning. Paper presented at the Annual Reading/Literacy Conference (8th, Bakersfield, California, August 22-23, 1997)
- [6] Whitton, D., Sinclair, C., Barker, K., Nanlohy, P., & Nosworthy, M. (2004). *Learning for teaching: Teaching for learning*. Southbank, Vic: Thomson Social Science Press
- [7] McCutcheon, G. (1980). How do elementary school teachers plan? The nature of planning and influences on it. *The Elementary School Journal*, 81(1), 4-23.
- [8] Marsh, C. (2004). *Becoming a teacher: Understandings, skills and issues*. Frenchs Forest, NSW: Pearson Education Australia
- [9] L.Cameron . Picture this: My lesson. How LAMS is being used with pre-service teachers to develop effective classroom activities. <http://mq.academia.edu/LeanneCameron/Papers/95569/Picture-this--My-lesson--How-LAMS-is-being-used-with-pre-service-teachers-to-develop-effective-classroom-activities> [Προσπελάστηκε 29/10/2021]
- [10] Dalziel, J. (2003). *Implementing Learning Design: The Learning Activity Management System (LAMS)*, Sydney: E-learning Centre of Excellence (MELCOE), Macquarie University. <https://research-management.mq.edu.au/ws/portalfiles/portal/19501627/mq-8129-Author+final+version.pdf> [Προσπελάστηκε 29/10/2021]
- [11] Σ. Παπαδάκης, Β. Ζήσκος (2015). Design and Implementation of Differentiated Instruction Digital Scenarios in LAMS. *Management of Innovative Business & Education System MIBES TRANSACTIONS International journal Vol 9, Issue2, 2015 ISSN 2459-3028* [http://mibes.uth.gr/vol9\\_issue2\\_2015/Papadakis,%20Ziskos.pdf](http://mibes.uth.gr/vol9_issue2_2015/Papadakis,%20Ziskos.pdf) [Προσπελάστηκε 29/10/2021]
- [12] Britain S. (2004). *A Review of Learning Design: Concept, Specifications and Tools*. A report for the JISC E-learning Pedagogy Programme. <https://staff.blog.ui.ac.id/harrybs/files/2008/10/learningdesigntoolsfinalreport.pdf> [Προσπελάστηκε 29/10/2021]
- [13] Hotaman, D. (2009). The teaching profession: knowledge of subject matter, teaching skills and personality traits. *Procedia Social and Behavioral Sciences* 2 pp. 1416–1420. [https://www.researchgate.net/publication/248607615\\_The\\_teaching\\_profession\\_knowledge\\_of\\_subject\\_matter\\_teaching\\_skills\\_and\\_personality\\_traits](https://www.researchgate.net/publication/248607615_The_teaching_profession_knowledge_of_subject_matter_teaching_skills_and_personality_traits) [Προσπελάστηκε 29/10/2021]
- [14] Ιστότοπος LAMS [https://www.lamsfoundation.org/about\\_home.htm](https://www.lamsfoundation.org/about_home.htm) [Προσπελάστηκε 29/10/2021]
- [15] Ιστότοπος LessonLams <https://www.lessonlams.com/lams/cloud/index.do> [Προσπελάστηκε 29/10/2021]
- [16] Ιστότοπος LamsCommunity <https://www.lamscommunity.org/> [Προσπελάστηκε 29/10/2021]
- [17] Ιστότοπος κοινότητας μάθησης και πρακτικής Ελλήνων Εκπαιδευτικών του LAMS <http://blogs.sch.gr/groups/lams/> [Προσπελάστηκε 29/10/2021]

- [18] Κονταξή Ιωάννα, Ρώσσιου Ελένη (2018) Ευχρηστία του LAMS και Διαφοροποιημένη Διδασκαλία: η πλευρά του μαθητή. Δημοσίευση στο Επιστημονικό Εκπαιδευτικό Περιοδικό “εκπ@ιδευτικός κύκλος” [https://journal.educircle.gr/images/teuxos/2018/teuxos3/teyxos\\_6\\_3\\_4.pdf](https://journal.educircle.gr/images/teuxos/2018/teuxos3/teyxos_6_3_4.pdf) [Προσπελάστηκε 29/10/2021]
- [19] Φακιολάκης Γ., Παπαδάκης Σ. (2011) Εργαλείο Δραστηριοτήτων LAMS [https://blogs.sch.gr/?get\\_group\\_doc=48/1528097085-5.LAMsv6.pdf](https://blogs.sch.gr/?get_group_doc=48/1528097085-5.LAMsv6.pdf) [Προσπελάστηκε 29/10/2021]
- [20] Οδηγός LAMS: Εργαλεία δραστηριοτήτων <https://lams.sch.gr/οδηγοί-βοηθοί/εργαλεία-δραστηριοτήτων/> [Προσπελάστηκε 29/10/2021]
- [21] Downey A., Elkner J. and Meyers C. 2013. Python to learn programming <https://iopscience.iop.org/article/10.1088/1742-6596/423/1/012027/pdf>[Προσπελάστηκε 29/10/2021]
- [22] Μ. Γρηγοριάδου, Ε. Γουλη, Α. Γόγουλου, Ε., Γλέζου, Κ., Μπούμπουκα, Μ., Παπανικολάου, Κ., Τσαγκάνου, Γ., Κανίδης, Ε., Βεργίνης, Η., Δουκάκης, (Αθήνα 2009) Διδακτικές Προσεγγίσεις και Εργαλεία για τη διδασκαλία της Πληροφορικής, Εκδόσεις Νέων Τεχνολογιών σ. 11, 19-26, 83-92, 285-314