

**Καινοτόμος Δράση**

<b>Τίτλος – Θέμα</b>	Τα μοντέλα της ανεστραμμένης τάξης και της μάθησης βασισμένης στο πρόβλημα στην εξ αποστάσεως διδασκαλία του προγραμματισμού.
<b>Περίοδος :</b>	Μάρτιος – Μάιος 2020
<b>Τάξεις:</b>	Α τάξη Γυμνασίου
<b>Δικτυακός τύπος:</b>	<a href="http://evripides.mysch.gr/drasesis/pdf/evrachnos_2020_1.pdf">http://evripides.mysch.gr/drasesis/pdf/evrachnos_2020_1.pdf</a>

**Περιγραφή της Δράσης**

Μετά το κλείσιμο των σχολείων στις 10 Μαρτίου, το σχολείο μας συνέχισε το εκπαιδευτικό έργο μέσα από πλατφόρμες σύγχρονης και ασύγχρονης εξ αποστάσεως εκπαίδευσης. Για τη σύγχρονη εκπαίδευση χρησιμοποιήθηκε η πλατφόρμα webex με τους λογαριασμούς που δόθηκαν στους εκπαιδευτικούς από το υπουργείο παιδείας. Στην ασύγχρονη αξιοποιήθηκαν οι πλατφόρμες eclass και edmodo. Δεδομένου ότι πλέον ένα μεγάλο βάρος της μαθησιακής διαδικασίας μεταφέρεται στους μαθητές αφού ο εκπαιδευτικός δεν είναι πλέον δίπλα τους και η σύγχρονη εκπαίδευση εξ αποστάσεως (χωρίς κάμερα) δεν μπορεί να υποκαταστήσει την εκπαιδευτική διαδικασία στην τάξη το βάρος πέφτει στην ασύγχρονη εκπαίδευση.

Οι ειδικές συνθήκες που διαμορφώθηκαν με το κλείσιμο των σχολείων συνετέλεσαν στην υιοθέτηση μαθητοκεντρικών διδακτικών μεθόδων οι οποίες ταιριάζουν στο μοντέλο της ασύγχρονης εκπαίδευσης. Δυο από αυτά τα διδακτικά μοντέλα που συνδυάσαμε στην διδασκαλία είναι το μοντέλο της ανεστραμμένης τάξης (flipped classroom) και η μάθηση βασισμένη στο πρόβλημα (Problem-Based Learning). Και οι δυο αυτές μέθοδοι ταιριάζουν στην διδασκαλία του προγραμματισμού, διότι η εκμάθηση του προγραμματισμού είναι μια διαδικασία που βασίζεται στον μαθητή και στην ικανότητά του να μαθαίνει πώς να μαθαίνει. Ο εκπαιδευτικός έχει τον ρόλο του διευκολυντή/καθοδηγητή αφού η γνώση δεν μεταφέρεται αλλά ανακαλύπτεται από τον μαθητή μέσω διερεύνησης και πειραματισμών με τον αλγόριθμο που έχει σχεδιαστεί.

**Το μοντέλο της ανεστραμμένης τάξης (flipped classroom)**

Η ανεστραμμένη μάθηση είναι μία παιδαγωγική προσέγγιση, κατά την οποία η διδασκαλία κινείται από το επίπεδο μάθησης της ομάδας στο αντίστοιχο ατομικό, με άμεσο επακόλουθο η ομάδα να μετασχηματίζεται σε δυναμικό, διαδραστικό και μαθησιακό περιβάλλον, όπου ο/η εκπαιδευτικός καθοδηγεί τους/τις μαθητές/τριες, ενώ οι ίδιοι/ες εφαρμόζουν έννοιες και τους/τις εμπλέκει δημιουργικά στο εκάστοτε γνωστικό αντικείμενο. Η ανεστραμμένη τάξη ενέχει δύο καθοριστικά στοιχεία: τη μετακίνηση της διάλεξης έξω από την τάξη και τη μεταφορά αναθέσεων πρακτικών εφαρμογών από το επίπεδο της κατ' οίκον εργασίας στο αντίστοιχο της τάξης.

Η αντίστροφη προσέγγιση στην τάξη μετατοπίζει την παιδαγωγική από διαλέξεις με επίκεντρο τον εκπαιδευτικό, παρέχοντας πληροφορίες στους μαθητές στην τάξη σε πιο σχετικές δραστηριότητες, όπως συζήτηση, επίλυση προβλημάτων, πρακτικές δραστηριότητες και άμεση καθοδήγηση εκπαιδευτικών (Akçayır & Akçayır, 2018;Roehl et al., 2013)

Η ανεστραμμένη τάξη είναι μια μορφή μεικτής μάθησης, κατά την οποία οι μαθητές/τριες μαθαίνουν το διαδικτυακό περιεχόμενο, παρακολουθώντας βίντεο-διαλέξεις, συνήθως στο σπίτι, και η κατ' οίκον εργασία συνεχίζεται στην τάξη με τους/τις εκπαιδευτικούς και τους/τις μαθητές/τριες να συζητούν και να επιλύουν απορίες. Η αλληλεπίδραση εκπαιδευτικού-μαθητή/τριας είναι πιο εξατομικευμένη, ενώ ακόμη τονίζεται ιδιαίτερα η καθοδήγηση αντί της διάλεξης. Για αυτό το μοντέλο της ανεστραμμένης τάξης ήταν το ιδανικό μοντέλο να χρησιμοποιήσουμε σε αυτές τις ειδικές συνθήκες όπου τα σχολεία ήταν κλειστά λόγω κορονοϊού και όλη η εκπαιδευτική διαδικασία έλαβε χώρα εξ αποστάσεως.

**Μάθηση βασισμένη στο πρόβλημα (Problem-based learning)**

Το PBL είναι μια εποικοδομητική παιδαγωγική βασισμένη στην ενεργή μάθηση και επικεντρώνεται στην ανάλυση και επίλυση αυθεντικών προβλημάτων. (El Mawas & Muntean Muntean, 2018). Η μέθοδος ‘Μάθηση βασισμένη στο Πρόβλημα’ (Problem Based Learning PBL) θεωρείται από τις πιο καινοτόμους εκπαιδευτικές μεθόδους. Ενισχύει τη μάθηση δημιουργώντας την ανάγκη για επίλυση ενός ανεπαρκώς δομημένου προβλήματος (ill-structured problem). Κατά τη διάρκεια επίλυσης του προβλήματος οι εκπαιδευόμενοι κατασκευάζουν περιεχόμενο και αναπτύσσουν δεξιότητες επίλυσης προβλημάτων (problem solving skills). Η αποτελεσματικότητά της έγκειται στο γεγονός ότι συντελεί στην ανάπτυξη της κριτικής σκέψης των εκπαιδευομένων, διευκολύνοντας τους μαθητές στην επίλυση προβλημάτων του πραγματικού κόσμου. Η “Μάθηση βασισμένη στο πρόβλημα”, είναι μια παιδαγωγική μέθοδος με μαθητοκεντρικό χαρακτήρα. Οι μαθησιακές δραστηριότητες που εννοχηστρώνονται με την παραπάνω μαθησιακή μέθοδο οργανώνονται με τρόπο που να τοποθετούν τον μαθητή στον πυρήνα της εκπαιδευτικής διαδικασίας και να τον μετατρέπουν από παθητικό δέκτη, σε ενεργό μέτοχο της μαθησιακής διαδικασίας, που διαμορφώνει με τις πράξεις του την γνωστική του πραγματικότητα. Η έμφαση δίνεται στο να αναγνωρίσουν οι ίδιοι οι μαθητές ποιες είναι οι γνώσεις που συμβάλουν στην επίλυση του προβλήματος, στη συνέχεια να κατανοήσουν ποιων γνώσεων στερούνται και να μάθουν πώς να τις αποκτήσουν. Με άλλα λόγια οι εκπαιδευόμενοι μαθαίνουν να διακρίνουν και να καταγράφουν “τι γνωρίζουν, τι δεν γνωρίζουν και εν συνεχεία να χαράσσουν ένα σχέδιο δράσης για την επίλυση του ανεπαρκώς δομημένου προβλήματος”.

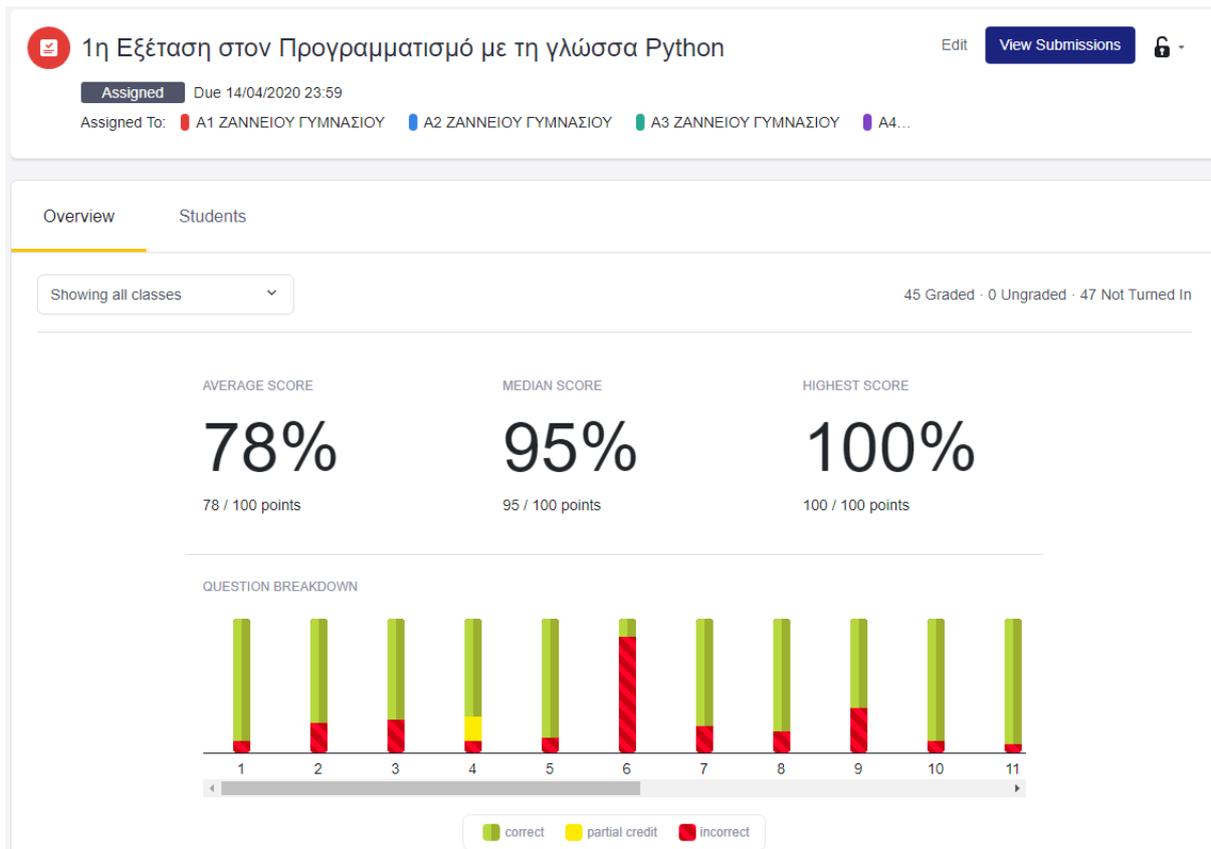
Από τα παραπάνω είναι φανερό ότι η μέθοδος αυτή ταιριάζει απόλυτα στην διδασκαλία του προγραμματισμού και της αλγοριθμικής σχεδίασης, ειδικά κάτω από τις παρούσες συνθήκες, για αυτό την χρησιμοποιήσαμε στην εισαγωγή στον προγραμματισμό με τη γλώσσα Python.

## Μεθοδολογία

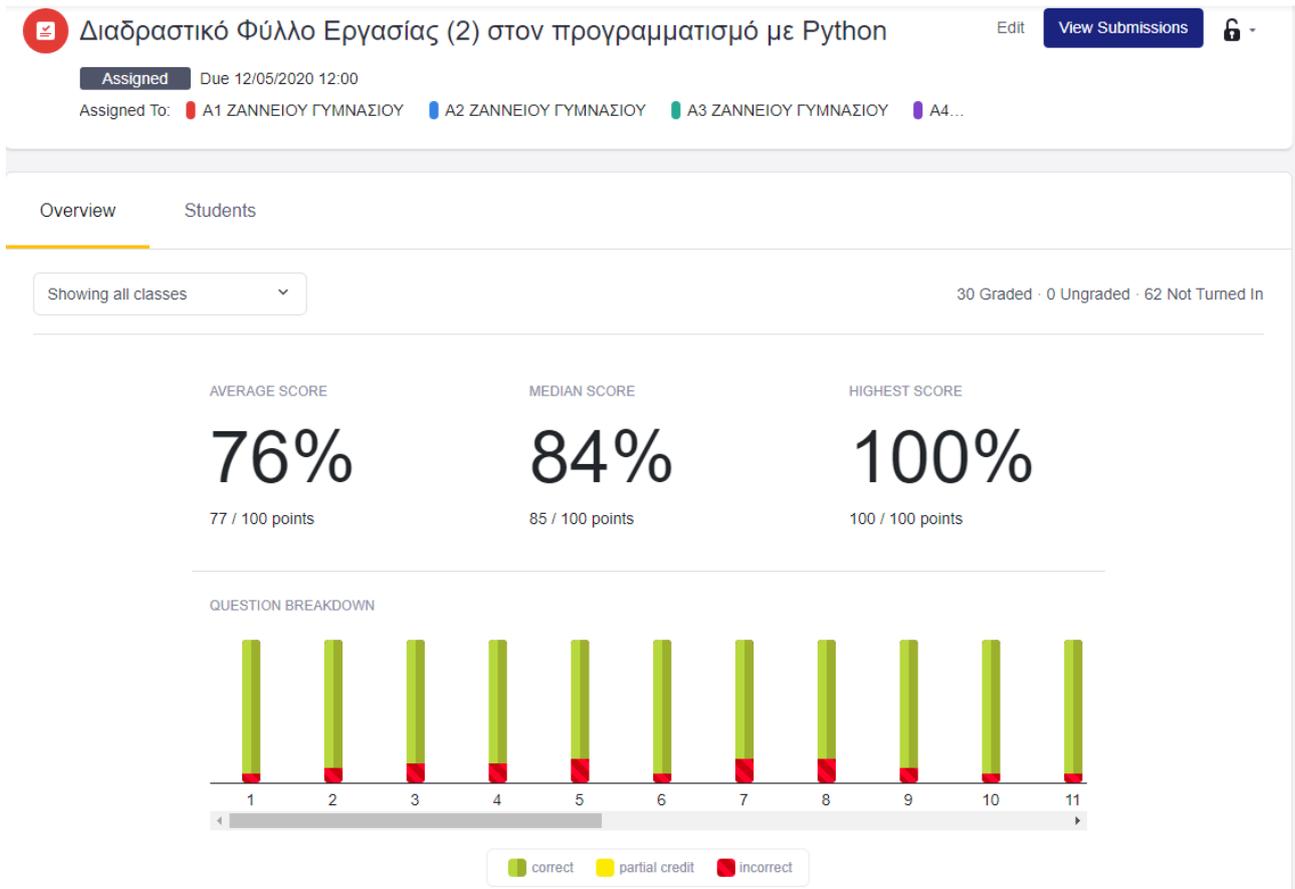
Για το τμήμα της ασύγχρονης εκπαίδευσης χρησιμοποιήσαμε τρεις διαφορετικές πλατφόρμες:

- Το Edmodo ( [www.edmodo.com](http://www.edmodo.com) )
- Τις φόρμες google για σύγχρονη εξέταση
- Την ηλεκτρονική τάξη

Παρακάτω φαίνεται μια εξέταση που δόθηκε στους μαθητές στο Edmodo πάνω σε απλές προγραμματιστικές έννοιες. Το πλεονέκτημα του Edmodo είναι ότι πέρα από τη δημιουργία ερωτήσεων πολλαπλών επιλογών έχει πολύ καλή εφαρμογή για χρήση από κινητό τηλέφωνο. Δεδομένου ότι πολλοί μαθητές δεν είχαν πρόσβαση σε Η/Υ όλες τις ώρες της ημέρας, αυτό αποδείχθηκε σωτήριο για τη συμμετοχή των μαθητών.



**Εικόνα 1.** Αποτελέσματα μαθητών κατά τη διάρκεια εξέτασης στο Edmodo



**Εικόνα 2.** Αποτελέσματα μαθητών κατά τη διάρκεια εξέτασης στο edmodo

Δίπλα δίνεται μια από τις ερωτήσεις που τέθηκαν στους μαθητές σε αυτό την εξέταση. Οι μαθητές απλά πρέπει να απαντήσουν τι θα εμφανίσει το πρόγραμμα. Αν θέλουν μπορούν να το δοκιμάσουν εκείνη τη στιγμή στον διερμηνευτή της Python.

Αυτό αποτέλεσε ένα κίνητρο για τους μαθητές για να χρησιμοποιήσουν το περιβάλλον προγραμματισμού αφού ήθελαν να επιβεβαιώσουν την απάντησή τους πριν την υποβάλουν.

Τι θα εμφανίσει το παρακάτω πρόγραμμα αν εκτελεστεί στην Python;

- 1 attachment -

```
a = 10
b = 28
temp = b
b = a
a = temp
print(a, b)
```

A. 10 28

B. 28 10

C. 10 10

D. 28 28

E. 38 38

**Εικόνα 3.** Άσκηση στο edmodo

Οι παρακάτω ασκήσεις ανήκουν σε μια άλλη κατηγορία και επίπεδο. Εδώ οι μαθητές δεν μπορούν να δοκιμάσουν άμεσα τις εντολές στην Python γιατί δεν δίνεται τιμή για το x, ώστε να τους υποχρεώσει να κάνουν υποθέσεις και να διερευνήσουν αυτό το τμήμα κώδικα. Θα μπορούσαν να χαρακτηριστούν υπό κάποια έννοια και ως θεωρητικές ασκήσεις, αφού απαιτούν την κατανόηση του μαθητή σε ένα υψηλότερο επίπεδο αφαίρεσης.

Ποιο είναι το αποτέλεσμα μετά την εκτέλεση των παρακάτω εντολών;

```
x = x + 10
x = x - 6
x = x - 3
```

- 1 attachment -

Τι θα εμφανίσει το παρακάτω πρόγραμμα αν εκτελεστεί στην Python για κάθε τιμή του number; Προσοχή! Δεν γνωρίζουμε τι τιμή έχει η μεταβλητή number, άρα δεν μπορείτε να της δώσετε συγκεκριμένη τιμή.

- 1 attachment -

```
x = number % number
print(x)
```

A. Η μεταβλητή x αυξάνεται κατά 10

B. Η μεταβλητή x αυξάνεται κατά 4

C. Η μεταβλητή x αυξάνεται κατά 3

D. Η μεταβλητή x αυξάνεται κατά 6

E. Η μεταβλητή x αυξάνεται κατά 1

F. Η μεταβλητή x παραμένει σταθερή

A. 0

B. 1

C. Η τιμή του number

D. Δεν ορίζεται είναι λάθος

**Εικόνα 4.** Θεωρητικές ασκήσεις στο edmodo

Εδώ ο αναγνώστης ίσως παρατήρησε ότι οι επιδόσεις των μαθητών παραπάνω ήταν πολύ υψηλές ίσως σε ύποπτο βαθμό. Αυτό έχει εξήγηση και ήταν μέρος της στρατηγικής μας. Όταν ένας μαθητής δεν πήγαινε καλά στο τεστ είχε τη δυνατότητα να το ξανακάνει. Μερικοί μαθητές το έκαναν ακόμα και τρεις φορές. Αυτό το χαρακτηριστικό βοήθησε πολύ κάποιους μαθητές να μην εγκαταλείψουν την προσπάθεια αλλά να χτίσουν σιγά-σιγά μια αυτοπεποίθηση. Επίσης δεν είχαν τον φόβο ότι μπορεί να πάνε άσχημα γιατί ήξεραν ότι μπορεί να έχουν ακόμα μια ευκαιρία και έτσι δεν απογοητεύονταν.

Εκτός από το Edmodo χρησιμοποίησα και τις φόρμες Google και πιο συγκεκριμένα την δυνατότητα που παρέχουν για τη δημιουργία εξετάσεων-κουίζ που βαθμολογούνται αυτόματα. Παρακάτω φαίνονται τα συγκεντρωτικά αποτελέσματα από μια εξέταση.



**Εικόνα 5.** Συγκεντρωτικά αποτελέσματα εξέτασης με τις φόρμες Google.

Στη συνέχεια ακολουθούν κάποιες ενδεικτικές ερωτήσεις αυτής της εξέτασης. Όπως θα δείτε δεν διαφέρουν πολύ από τις ερωτήσεις του Edmodo. Ένα μεγάλο πλεονέκτημα είναι ότι παράγουν άμεσα συγκεντρωτικά αποτελέσματα σε πραγματικό χρόνο.

Ακολουθεί ένα παράδειγμα ερώτησης σε αυτό το τεστ.

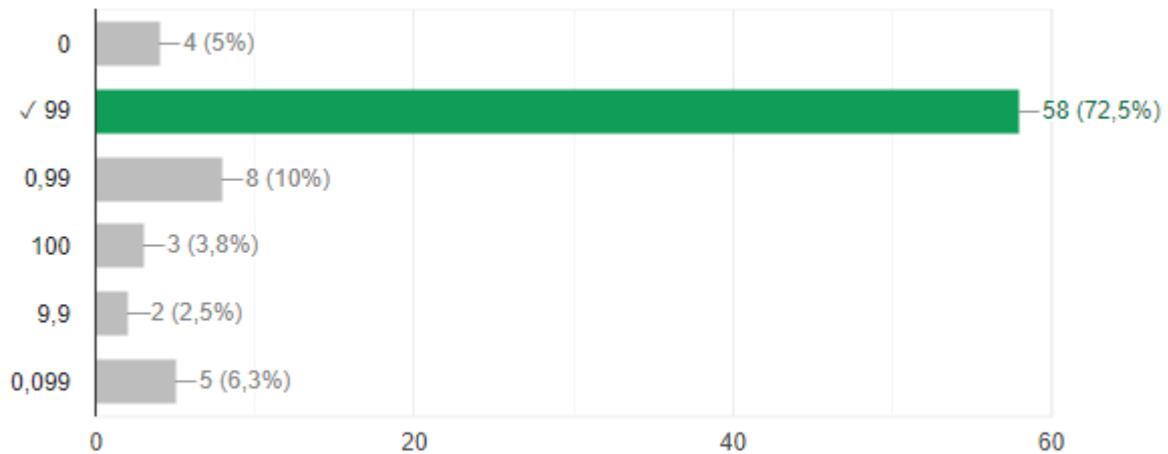
Ποιο είναι το αποτέλεσμα της παρακάτω εντολής; \*

```
>>> print( 99 % 100 )
```

- 0
- 99
- 0,99
- 100
- 9,9
- 0,099

**Εικόνα 6.** Ερώτηση σε εξέταση με το google forms

Και στη συνέχεια τα αποτελέσματα για αυτή την ερώτηση.



**Εικόνα 7.** Συγκεντρωτικά αποτελέσματα απαντήσεων μαθητών

☑ Επιλέξτε τις σωστές απαντήσεις:

---

Ποιο είναι το αποτέλεσμα της παρακάτω εντολής ; 10 βαθμοί

```
>>> print ( 120 % 100 )
```

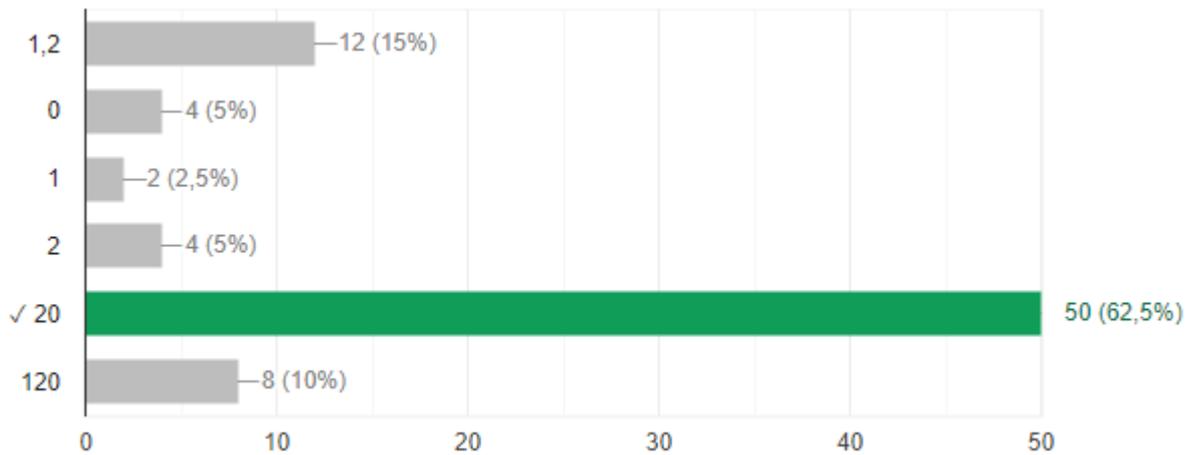
1,2  
 0  
 1  
 2  
 20 ✓  
 120

Σχόλια για λανθασμένες απαντήσεις ✎ 🗑

Υπόλοιπο ακεραίας διαίρεσης

[↪ Περισσότερα μπορείς να ...](#)

**Εικόνα 8.** Ερώτηση σε εξέταση με το google forms



**Εικόνα 9.** Συγκεντρωτικά αποτελέσματα απαντήσεων μαθητών

Ποιο είναι το αποτέλεσμα της παρακάτω εντολής :

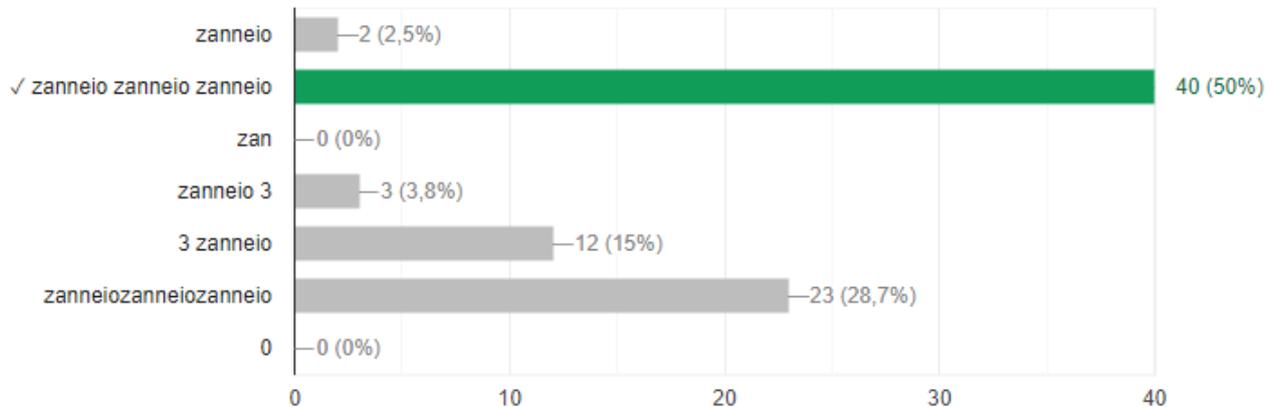
10 βαθμοί

```
>>> print ( 3 * "zanneio " )
```

- zanneio
- zanneio zanneio zanneio ✓
- zan
- zanneio 3
- 3 zanneio
- zanneiozanneiozanneio
- 0

Σχόλια για λανθασμένες απαντήσεις ✎ 🗑  
 Αντιγράφει 3 φορές τη λέξη zanneio

**Εικόνα 10.** Συγκεντρωτικά αποτελέσματα απαντήσεων μαθητών



**Εικόνα 11.** Συγκεντρωτικά αποτελέσματα απαντήσεων μαθητών

Στην τελευταία αυτή ερώτηση οι μαθητές δεν πήγαν τόσο καλά όσο στις προηγούμενες. Ο λόγος είναι ότι οι μαθητές δεν είναι εξοικειωμένοι με τη διαχείριση των αλφαριθμητικών σε αντίθεση με τα αριθμητικά δεδομένα. Για αυτό δεν διέκριναν ότι το κενό (space) είναι και αυτό χαρακτήρας όπως τα γράμματα και τα ψηφία των αριθμών.

**Το παιχνίδι Πέτρα – Ψαλίδι – Χαρτί**

Αφού οι μαθητές εξοικειώθηκαν με την έννοια της μεταβλητής και των τύπων της Python, τους θέσαμε ένα πρόβλημα, την κατασκευή του γνωστού παιχνιδιού Πέτρα-Ψαλίδι-Χαρτί. Για τον σκοπό δόθηκε στους μαθητές ένα φύλλο εργασίας 5 σελίδων το οποίο είναι σχεδιασμένο σε ένα πλαίσιο φθίνουσας καθοδήγησης. Το φύλλο εργασίας παρατίθεται στο παράρτημα. Το ενδιαφέρον μέρος ήταν στο τέλος που ζητείται από τους μαθητές να σχεδιάσουν το παιχνίδι ώστε να κλέβει και να κερδίζει πάντα τον παίκτη.

**Συμπεράσματα**

Οι ειδικές συνθήκες των μηνών Μαρτίου – Απριλίου – Μαΐου μας οδήγησαν στην υιοθέτηση νέων εκπαιδευτικών μεθόδων και στρατηγικών έτσι ώστε να προσαρμοστούμε στα νέα δεδομένα. Το μοντέλο της ανεστραμμένης τάξης και η μάθηση βασισμένη στο πρόβλημα είναι δυο μέθοδοι που ταιριάζουν απόλυτα όχι μόνο στην ειδική κατάσταση που διαμορφώθηκε αλλά και στο γνωστικό μας αντικείμενο. Δεδομένου ότι ο προγραμματισμός απαιτεί δεξιότητες επίλυσης προβλήματος οι δυο αυτές μέθοδοι ταίριαξαν πολύ καλά και σε συνδυασμό με την χρήση περιβαλλόντων ασύγχρονης εκπαίδευσης όπως το Edmodo, η ηλεκτρονική τάξη και τα κουίζ των google forms, έδωσαν τη δυνατότητα σε πολλούς μαθητές και μαθήτριες να ασχοληθούν ενεργά με την υλοποίηση ενός παιχνιδιού στη γλώσσα προγραμματισμού Python. Πιστεύουμε ότι αυτά τα εργαλεία θα μπορούσαν να χρησιμοποιηθούν παράλληλα με την δια ζώσης διδασκαλία με εξαιρετικά αποτελέσματα. Ωστόσο δεν μπορούν σε καμία περίπτωση να την υποκαταστήσουν, δεδομένου ότι η μάθηση έχει και έναν κοινωνικό συγκρουσιακό χαρακτήρα, από τον οποίο αναδύεται η νέα γνώση.

**Βιβλιογραφία**

Akçayır, G., & Akçayır, M. (2018). The flipped classroom: A review of its advantages and challenges. *Computers & Education*, 126, 334–345. <https://doi.org/10.1016/j.compedu.2018.07.021>

Chis, A. E., Moldovan, A.-N., Murphy, L., Pathak, P., & Muntean, C. H. (2018). Investigating Flipped Classroom and Problem-based Learning in a Programming Module for Computing Conversion Course. *Educational Technology & Society*, 21 (4), 232–247.

El Mawas, N., & Muntean, C. H. (2018) 2018). Supporting lifelong learning through development of 21st century skills. In *Proceedings of the 10th annual International Conference on Education and New Learning Technologies (EDULEARN)* (pp. 73437343-73507350). Palma de Mallorca, Spain Spain: IATED.

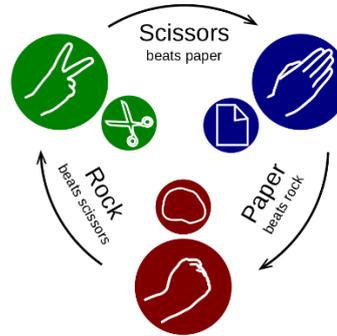
Topalli D, Cagiltay NE. (2018). Improving programming skills in engineering education through problem-based game projects with Scratch. *Computers & Education*. 2018; 120:64–74.

**Παράρτημα**

**Εργασία :** Υλοποίηση του παιχνιδιού Πέτρα- Ψαλίδι - Χαρτί

**Εισαγωγή**

Ο σκοπός αυτής της δραστηριότητας είναι η υλοποίηση του γνωστού παιχνιδιού Πέτρα – Ψαλίδι – Χαρτί το οποίο παίζουν δυο παίκτες, εσείς και ο υπολογιστής. Σε περίπτωση που δεν έχετε ξαναπαίξει θυμίζουμε ότι οι δυο παίκτες επιλέγουν ένα αντικείμενο και η Πέτρα κερδίζει το Ψαλίδι, το Ψαλίδι κερδίζει το Χαρτί το οποίο κερδίζει την Πέτρα.



**Η τύχη βοηθάει τους/τις τολμηρούς/ές**

Θέλουμε ο υπολογιστής να επιλέξει στην τύχη ένα εκ των Πέτρα/Ψαλίδι/Χαρτί. Για αυτό τον λόγο θα χρησιμοποιήσουμε την εντολή `randint(0,2)` η οποία επιστρέφει με τυχαίο τρόπο έναν εκ των αριθμών 0, 1 ή 2. Δοκιμάστε να καλέσετε την `randint` στον διερμηνευτή πολλές φορές για να βεβαιωθείτε ότι παράγει με τυχαίο τρόπο τους τρεις αυτούς αριθμούς:

```
>>> from random import randint
>>> print ( randint(0, 2) )
>>> print ( randint(0, 2) )
>>> print ( randint(0, 2) )
```

Με χρήση της εντολής επανάληψης **while** μπορείτε να εμφανίσετε πολλούς ακέραιους αριθμούς στο διάστημα [0,2] με τυχαίο τρόπο. Μπορείτε να μετρήσετε πόσοι αριθμοί θα εμφανιστούν αν εκτελέσουμε το παρακάτω τμήμα κώδικα;

```
from random import randint # Για βίαιη διακοπή πατήστε Ctrl+C ☺
while True: # Η παράμετρος end = " " δίνει στην print την οδηγία
    number = randint(0, 2) να μην κατέβει στην επόμενη γραμμή αλλά να αφήσει
    print ( number, end = " ") 1 κενό πριν εμφανίσει τον επόμενο αριθμό.
```

Με την εντολή **break** την οποία όμως πρέπει να χρησιμοποιούμε με προσοχή η επανάληψη τερματίζει όταν ικανοποιηθεί (**True**) η συνθήκη που βρίσκεται στην **if**.

```
from random import randint
while True:
    number = randint(0, 2)
    print ( number, end = " ")
    if number == 2 :
        break;
    print ( number, end = " Είμαι μέσα στην επανάληψη ")
print ( " Βγήκα από την επανάληψη ")
```

### Η σειρά του υπολογιστή

Ο υπολογιστής θα παίζει σε κάθε γύρο καλώντας την randint. Αν η συνάρτηση επιστρέψει 0 σημαίνει ότι ο υπολογιστής παίζει Πέτρα (R), 1 για Ψαλίδι (S) και 2 για Χαρτί (P). Να συμπληρώσετε το παρακάτω τμήμα κώδικα έτσι ώστε στη μεταβλητή computer να καταχωρείται η επιλογή του υπολογιστή και στη συνέχεια να εμφανίζεται στην οθόνη το γράμμα αλλά και ο αντίστοιχος αριθμός για να ελέγξετε αν ο κώδικάς σας είναι σωστός.

```

from random import randint
print ( "*****" )
print ( "* ΠΕΤΡΑ – ΨΑΛΙΔΙ ΧΑΡΤΙ ΕΚΔΟΣΗ: 1.0 *" )
print ( "* του/της <το όνομά σας> *" )
print ( "*****" )
choice = randint(0, 2)

if choice == 0 :
    computer = 'P'
elif _____ :
    _____
else :
    _____
print ( ' computer = ', computer, ' choice = ', choice)
    
```

**Βελτιστοποίηση:** Μπορείτε να σκεφτείτε έναν τρόπο αντιστοίχησης των 0,1,2 στα 'R', 'S', 'P' χωρίς να χρησιμοποιήσετε την εντολή if; **Υπόδειξη:** Χρησιμοποιήστε μια λίστα.

### Τώρα είναι η σειρά σου

Τώρα είναι η σειρά του παίκτη. Το πρόγραμμα θα ζητάει από τον παίκτη την επιλογή του η οποία θα είναι 'R', 'S', ή 'P' με χρήση της input όπως φαίνεται παρακάτω:

```

player = input ( 'Δώσε την επιλογή σου: R για Πέτρα, P για χαρτί και S για ψαλίδι' )
    
```

Στη συνέχεια ελέγχοντας όλους τους συνδυασμούς μεταξύ computer και player το πρόγραμμα θα εμφανίζει κατάλληλο μήνυμα για το νικητή ή την ισοπαλία του γύρου, και στο τέλος θα εμφανίζει τις επιλογές και των δυο, ώστε να φαίνεται ότι δεν ..... κλέβει.

```

if computer == 'P' and player == 'S' : # το ψαλίδι κερδίζει το χαρτί
    print ( ' Κέρδισες!!! ' )
elif computer == 'P' and player == 'R' :
    .....
    .....
print ( ' You = ', player )
print ( ' Computer = ', computer )
    
```

Παρακάτω δίνεται μια οθόνη εκτέλεσης του παιχνιδιού, με ελαφρά διαφορετικά μηνύματα από αυτά που έχω παραπάνω. Μπορείτε να βάλετε και εσείς τα δικά σας μηνύματα στα αγγλικά ή στα ελληνικά.

```

*****
*      Rock-Paper-Scissors Game Ver. 0.1      *
*      programmed by Euripides                *
*****

Keep Calm, it is your turn now
Enter choice : P: Paper, S: Scissors and R: Rock
P
You lost, I won!!!
You : P
Computer : S
    
```

### Επαναλαμβανόμενοι Γύροι

Μέχρι στιγμής έχουμε σχεδιάσει ένα παιχνίδι το οποίο όμως έχει ένα μόνο γύρο. Εμείς όμως θέλουμε να παίζουμε συνέχεια μέχρι να βαρεθούμε. Για να το πετύχουμε αυτό θα πρέπει να βρούμε έναν τρόπο να επαναλαμβάνεται η παραπάνω διαδικασία μέχρι να δώσουμε εμείς ένα μήνυμα ότι βαρεθήκαμε και θέλουμε να σταματήσουμε. Ένας εύκολος τρόπος θα ήταν να θέσουμε όλη τη διαδικασία που υλοποιήσαμε προηγουμένως μέσα σε μια επανάληψη while όπως φαίνεται παρακάτω:

```

while True :
    choice = randint(0, 2)
    player = input ( 'Δώσε την επιλογή σου ..... : (R/P/S) ' )
    .....
    .....

print ( 'You =', player )
print ( 'Computer = ', computer )
    
```

Τώρα μπορείτε να παίξετε ΠΨΧ όσες φορές θέλετε.

### Πότε σταματάμε;

Όπως σύντομα θα διαπιστώσετε έχουμε δυο μικρά προβλήματα. Το πρώτο είναι ότι ο παραπάνω αλγόριθμος δεν σταματάει ποτέ, και το δεύτερο ότι πρέπει να κρατάμε το σκορ στο χαρτί. Ας ασχοληθούμε με το πρώτο. Το πρόβλημα είναι η εντολή **while True** η οποία θα πρέπει να αντικατασταθεί από μια συνθήκη που θα διαπιστώνει αν εμείς θέλουμε να συνεχίσουμε το παιχνίδι ή όχι. Καλείστε λοιπόν τώρα να ρωτήσετε τον χρήστη αν θέλει να συνεχίσει. Η ερώτηση αυτή θα πρέπει να βρίσκεται στο τέλος της επανάληψης:

```

answer = _____
while answer == 'Y' or answer == 'y' :
    .....
    answer = input ( 'Θέλεις να συνεχίσεις το παιχνίδι; (Y/N) ' )
    
```

Από την απάντηση θα κριθεί ο τερματισμός ή όχι της επανάληψης. Ποια πρέπει να είναι η αρχική τιμή της μεταβλητής answer έτσι ώστε να εκτελεστούν οι εντολές μέσα στην επανάληψη;

### Κρατάμε σκορ;

Ας λύσουμε τώρα το δεύτερο προβληματάκι που διαπιστώσαμε, ότι δηλαδή ενώ έχουμε τη δυνατότητα να παίζουμε πολλούς γύρους, στο τέλος δεν φαίνεται το σκορ. Για να κρατάμε το σκορ αρκεί να μετράμε τις νίκες κάθε παίκτη και στο τέλος του παιχνιδιού να εμφανίζουμε το τελικό σκορ. Έτσι θα χρειαστούμε 3 μεταβλητές, μια για τις δικές μας νίκες, μια για τις νίκες του υπολογιστή και μια για τις ισοπαλίες. Ανάλογα με το αποτέλεσμα θα αυξάνεται η αντίστοιχη μεταβλητή, όπως φαίνεται στο παρακάτω τμήμα κώδικα:

```

win = ____
while ..... :
    if computer == 'P' and player == 'S' : # το ψαλίδι κερδίζει το χαρτί
        print ( 'Κέρδισες σε αυτόν το γύρο!!! ' )
        win = win + 1
    elif computer == 'P' and player == 'R' : # το χαρτί κερδίζει την πέτρα
        print ( 'Έχασες σε αυτόν τον γύρο !!! ' )
        lost = lost + 1
    elif computer == 'P' and player == 'P' :
        print ( 'Ισοπαλία σε αυτόν τον γύρο !!! ' )
        draw = draw + 1
    .....

    if win ____ lost :
        print ( 'Κέρδισες!!! ' )
    elif _____ :
        print ( 'Έχασες !!! ' )
    else :
        print ( 'Ισοπαλία !!! ' )

print ( " *****" )
print ( "      Game Over      " )
print ( " *****" )
    
```

- Ποιες πιστεύετε ότι πρέπει να είναι η αρχικές τιμές των win, lost, draw?
- Τι θα συμβεί αν ο χρήστης δεν δώσει κάποιο από τα κεφαλαία γράμματα R,P,S. Πως μπορούμε να αντιμετωπίσουμε αυτό το πρόβλημα;
- Μπορείτε να τροποποιήσετε τη συνθήκη της while έτσι ώστε το παιχνίδι να σταματάει όταν κάποιος κερδίζει με διαφορά 10 νίκες;

### Απαιτητικό

Να δώσετε μια δεύτερη έκδοση του προγράμματός σας στην οποία ο υπολογιστής θα προσπαθεί να κερδίσει τον άνθρωπο με τον εξής τρόπο: Δεν θα επιλέγει στην τύχη αλλά θα καθορίζει την επιλογή έτσι ώστε να κερδίζει πάντα, αφού θα γνωρίζει την επιλογή του παίκτη. Θα διαβάζει πρώτα την επιλογή και αμέσως θα εμφανίζει την δική του αλλά δεν θα είναι τυχαία (θα κλέβει 😊)