

# Αποτίμηση της πιλοτικής διδασκαλίας του προγραμματισμού με τη γλώσσα Python σε μαθητές Γυμνασίου

Ευριπίδης Βραχνός<sup>1</sup>, Μαρία Κατσένη<sup>2</sup>

<sup>1</sup>Ζάννειο Πειραματικό Γυμνάσιο Πειραιά, <sup>2</sup>Γυμνάσιο Θρακομακεδόνων  
{evrachnos, mkatseni}@gmail.com

## Περίληψη

Η Python είναι μια σύγχρονη γλώσσα προγραμματισμού που έχει κυριαρχήσει τα τελευταία χρόνια στην τριτοβάθμια εκπαίδευση και έχει εισαχθεί πρόσφατα στην ελληνική δευτεροβάθμια επαγγελματική εκπαίδευση. Πρόκειται για μια script γλώσσα η οποία ανατρέπει πολλά από τα προγραμματιστικά στερεότυπα που υπήρχαν στην εκπαίδευση όπως ο δομημένος προγραμματισμός και οι στατικοί πίνακες και φέρνει μια εντελώς διαφορετική φιλοσοφία όχι μόνο στον τρόπο που προγραμματίζουμε αλλά κυρίως στην διδακτική προσέγγιση που ακολουθούμε στην διδασκαλία του προγραμματισμού. Στην εργασία αυτή παρουσιάζουμε τα αποτελέσματα από την πιλοτική διδασκαλία του προγραμματισμού με τη γλώσσα Python σε μαθητές Γυμνασίου. Η προσέγγιση που ακολουθήθηκε στηρίχθηκε σε δραστηριότητες διερευνητικού χαρακτήρα που έγιναν στο εργαστήριο πληροφορικής. Τα αποτελέσματα που παρουσιάζουμε είναι κυρίως ποιοτικά και χωρίζονται σε δυο κατηγορίες. Στις διδακτικές προσεγγίσεις που σχεδιάσαμε και προτείνουμε όπως και στα προβλήματα που συναντήσαμε στην υλοποίηση αυτών.

**Λέξεις κλειδιά:** Python, προγραμματισμός, Γυμνάσιο

## 1. Εισαγωγή

Η Python είναι μια σύγχρονη γλώσσα προγραμματισμού που έχει κυριαρχήσει τα τελευταία χρόνια στην τριτοβάθμια εκπαίδευση διεθνώς αλλά και στη χώρα μας. Η απλότητα της γλώσσας και η μεγάλη ποικιλία βιβλιοθηκών που παρέχει είναι ο λόγος που χρησιμοποιείται όχι μόνο από τμήματα πληροφορικής αλλά και από τμήματα Μαθηματικών και Οικονομικών. Τα τελευταία δυο χρόνια η Python έχει εισαχθεί και στην επαγγελματική εκπαίδευση με τα μαθήματα Αρχές Προγραμματισμού (Αράπογλου, Βραχνός, Κανίδης κ.α. 2015) και Προγραμματισμός Υπολογιστών (Αράπογλου, Βραχνός, Κανίδης κ.α. 2016) της Β' και Γ' ΕΠΑΛ του τομέα πληροφορικής.

Η Python είναι μια script γλώσσα όπως η PHP και η PERL χωρίς όμως τις δυσκολίες που παρουσιάζουν αυτές οι γλώσσες στην σύνταξή τους. Η Python παρόλο

που δεν σχεδιάστηκε για εκπαιδευτικούς σκοπούς, ελαχιστοποιεί τον συντακτικό θόρυβο, λόγω της πολύ απλής και λιτής της σύνταξης η οποία σε κάποιες περιπτώσεις θυμίζει ψευδοκώδικα (Agarwal, et. al., 2008; Goldwasser, & Letscher, 2008; Shein, 2015). Για αυτόν τον λόγο τα τελευταία χρόνια όλο και περισσότερες πανεπιστημιακές σχολές την χρησιμοποιούν ως εισαγωγική γλώσσα προγραμματισμού (Guo 2014).

Ωστόσο η μετάβαση από την Pascal ή την ελληνική ΓΛΩΣΣΑ/ψευδογλώσσα στην Python δεν είναι μια εύκολη υπόθεση για τους καθηγητές γιατί εκτός από την εκμάθηση μιας νέας γλώσσας θα πρέπει να αλλάξουν εντελώς τη φιλοσοφία της διδασκαλίας τους και τις διδακτικές στρατηγικές που χρησιμοποιούν. Η Python ανατρέπει πολλά από τα προγραμματιστικά στερεότυπα που υπήρχαν στην εκπαίδευση για δεκαετίες όπως ο δομημένος προγραμματισμός, οι στατικοί πίνακες και η χρήση υποπρογραμμάτων, εισάγοντας εντελώς διαφορετική φιλοσοφία στον τρόπο με τον οποίο διδάσκουμε προγραμματισμό. Το βασικότερο όμως από όλα είναι ότι πλέον ο προγραμματισμός διδάσκεται προγραμματίζοντας στον υπολογιστή, κάτι που θα έπρεπε να είναι αυτονόητο και όχι στον πίνακα. Ο διερμηνευτής της Python μπορεί να αξιολογηθεί στο έπακρον για τον σχεδιασμό διερευνητικών δραστηριοτήτων με πολύ μικρή καθοδήγηση από τον εκπαιδευτικό.

Στην εργασία αυτή παρουσιάζουμε τα αποτελέσματα από την πιλοτική διδασκαλία του προγραμματισμού με τη γλώσσα Python σε μαθητές Γυμνασίου. Η προσέγγιση που υιοθετήσαμε στηρίχθηκε σε δραστηριότητες διερευνητικού χαρακτήρα που έγιναν στο εργαστήριο πληροφορικής. Τα αποτελέσματα που παρουσιάζουμε είναι κυρίως ποιοτικά και χωρίζονται σε δυο κατηγορίες. Στις διδακτικές προσεγγίσεις που σχεδιάσαμε και προτείνουμε αλλά και στα προβλήματα που συναντήσαμε και τα οποία οφείλονται στις ιδιαιτερότητες που φέρνει αυτή η νέα γλώσσα στον τρόπο διδασκαλίας και μάθησης.

## ***2. Γλώσσες προγραμματισμού στην εκπαίδευση***

Είναι αποδεκτό ότι η διδασκαλία του προγραμματισμού υπολογιστών και της αλγοριθμικής σχεδίασης δυσκολεύουν πολύ τους μαθητές και τους φοιτητές. Εκτός από την εγγενή δυσκολία που παρουσιάζουν οι αντίστοιχες έννοιες και δομές ένας ακόμα παράγοντας δυσκολίας που υπεισέρχεται στη διαδικασία της μάθησης έχει να κάνει με τη γλώσσα προγραμματισμού που χρησιμοποιείται. Η επιλογή της γλώσσας προγραμματισμού με την οποία οι μαθητές και οι φοιτητές θα έχουν την πρώτη τους επαφή με τον προγραμματισμό καθορίζει σε μεγάλο βαθμό τις πρώτες αναπαραστάσεις που θα δημιουργήσουν οι μαθητές για τα προγραμματιστικά αντικείμενα και τις αλγοριθμικές δομές, όπως επίσης και την στάση των μαθητών απέναντι στον προγραμματισμό (Kelleher & Pausch, 2005; McIver, 1996).

Στην ελληνική εκπαίδευση παρατηρείται το φαινόμενο κάποιοι μαθητές να εισάγονται στον προγραμματισμό από το δημοτικό με το Scratch και να έρχονται πάλι σε επαφή με τον προγραμματισμό μετά από τρία χρόνια στην Γ' γυμνασίου όπου όμως σε πολλές περιπτώσεις προγραμματίζουν σε Logo. Στη συνέχεια στην Α' τάξη του Γενικού Λυκείου υπάρχει το μάθημα επιλογής Εφαρμογές Πληροφορικής, όπου οι μαθητές αναπτύσσουν εφαρμογές με το App Inventor, ενώ στην Β' τάξη το μάθημα γενικής παιδείας "Εισαγωγή στην Επιστήμη των Η/Υ" γίνεται σε μια γλώσσα στα ελληνικά η οποία μοιάζει με ψευδοκώδικα. Η ίδια προσέγγιση ακολουθείται και στο μάθημα κατεύθυνσης της Γ' Λυκείου όπου οι μαθητές αναπτύσσουν αλγορίθμους σε μια ελληνική γλώσσα προγραμματισμού τη ΓΛΩΣΣΑ που είχε σχεδιαστεί για την εκπαίδευση και μοιάζει αρκετά στην Pascal. Από την άλλη, στο Επαγγελματικό Λύκειο έχει εισαχθεί η γλώσσα Python από το έτος 2015-2016 στην Β Λυκείου και από το 2016-2017 στην Γ' Λυκείου, ως μάθημα του τομέα πληροφορικής. Χρησιμοποιείται όμως και από τα μαθήματα πληροφορικής γενικής παιδείας. Στην Γ'ΕΠΑΛ υπάρχει επίσης το μάθημα "Ειδικά θέματα στον Προγραμματισμό Υπολογιστών" (Βραχνός, κ.α., 2015) στο οποίο χρησιμοποιείται η γλώσσα προγραμματισμού Java από το έτος 2015-2016.

Από τα παραπάνω φαίνεται ότι υπάρχει μια προσπάθεια εκσυγχρονισμού των προγραμμάτων σπουδών της πληροφορικής κάτι το οποίο είναι επιβεβλημένο από τις ραγδαίες εξελίξεις της επιστήμης αυτής.

### 3. Η γλώσσα προγραμματισμού Python

Ένα πρόγραμμα σε Python δεν απέχει πολύ από την περιγραφή ενός αλγορίθμου σε ψευδογλώσσα, αφού η σύνταξή του είναι εξαιρετικά απλή. Δεν υπάρχει τμήμα δήλωσης μεταβλητών αφού η γλώσσα χρησιμοποιεί ένα δυναμικό σύστημα τύπων το οποίο σε συνδυασμό με τη χρήση του διερμηνευτή, διευκολύνει τον πειραματισμό των μαθητών.

**Συνάρτηση Αναζήτησης( A, key ) : Λογική**

**Μεταβλητές**

**Ακέραιες:** A[100], key, i

**Λογικές :** Βρέθηκε

**Αρχή**

i ← 1

Βρέθηκε ← Ψευδής

**Όσο Όχι Βρέθηκε και i ≤ 100 Επανάλαβε**

**Αν** A[i] = key **Τότε**

Βρέθηκε ← Αληθής

**Τέλος\_Αν**

i ← i + 1

**Τέλος\_Επανάληψης**

Αναζήτηση ← Βρέθηκε

**Τέλος\_Συνάρτησης**

**def** search ( A, key ) :

**for** item **in** A :

**if** item == key :

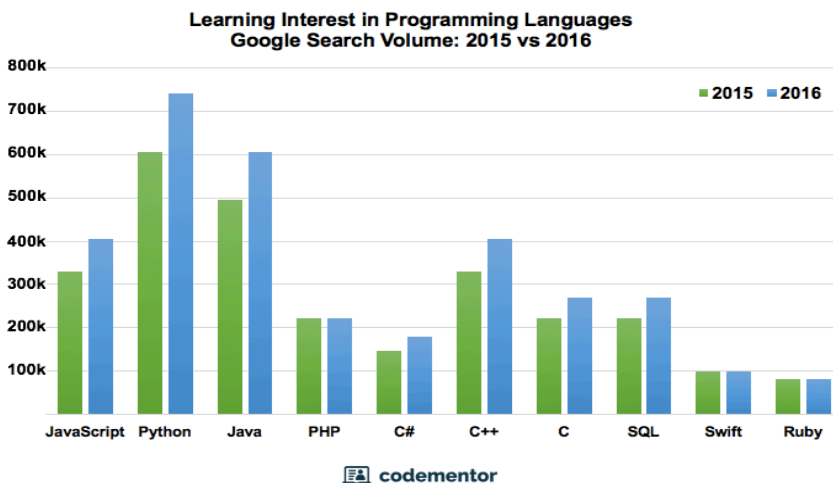
**return** True

**return** False

Παραπάνω δίνεται ο αλγόριθμος της σειριακής αναζήτησης ο οποίος ελέγχει αν ένα στοιχείο key υπάρχει στην δομή A, κωδικοποιημένος στη ΓΛΩΣΣΑ του μαθήματος της Ανάπτυξης Εφαρμογών σε Προγραμματιστικό Περιβάλλον της κατεύθυνσης Οικονομίας-Πληροφορικής του Γενικού Λυκείου και σε Python. Είναι φανερό ότι η κωδικοποίηση σε Python εκτός από σημαντικά λιγότερο κώδικα και πολύ λιγότερο συντακτικό θόρυβο είναι αρκετά πιο απλή αφού ο μαθητής δεν χρειάζεται να χρησιμοποιήσει την διπλή συνθήκη. Το είδος αυτό του αδόμητου προγραμματισμού με χρήση return ή break χρησιμοποιείται ευρέως στον σύγχρονο προγραμματισμό και δεν θεωρείται πλέον κάτι προβληματικό όπως ήταν παλιά η χρήση της εντολής goto.

Ένα άλλο σημαντικό πλεονέκτημα της παραπάνω κωδικοποίησης είναι ότι η συνάρτηση σε Python μπορεί να χρησιμοποιηθεί για αναζήτηση στοιχείων οποιουδήποτε τύπου και πλήθους ενώ στη ΓΛΩΣΣΑ θα πρέπει να γράψουμε μια διαφορετική έκδοση για κάθε τύπο δεδομένων (ακέραιο, πραγματικό, αφαριθμητικό, λογικό) και για κάθε διαφορετικό μέγεθος πίνακα. Από το παραπάνω παράδειγμα φαίνεται ότι η Python μας επιτρέπει να διατυπώνουμε αλγορίθμους στο υψηλότερο επίπεδο αφαίρεσης και για αυτό μοιάζει πολύ με ψευδογλώσσα.

Επίσης η Python έχει πολύ μεγάλη κοινότητα εκπαιδευτικών και προγραμματιστών, οι οποίοι διαθέτουν ελεύθερα υλικό (βιβλία, online μαθήματα, φύλλα εργασίας, tutorials, σημειώσεις) στο διαδίκτυο. Ενώ η γλώσσα είναι εξαιρετικά απλή και λιτή διαθέτει μια τεράστια συλλογή βιβλιοθηκών για όλες σχεδόν τις εφαρμογές η οποία επεκτείνεται συνεχώς. Για αυτό έχει πολύ απότομη καμπύλη μάθησης. Δηλαδή ένας αρχάριος προγραμματιστής θα φτάσει πολύ γρήγορα σε επίπεδο να αναπτύσσει χρήσιμες εφαρμογές.



*Εικόνα 1* Στατιστικά αναζητήσεων στο google για γλώσσες προγραμματισμού

Άλλωστε πολλές γνωστές εφαρμογές του παγκόσμιου ιστού έχουν αναπτυχθεί σε Python (YouTube, Pinterest, Google, Reddit, Quora, DropBox, Instagram, Spotify, Civilization IV).

Συμπερασματικά η Python φαίνεται σαν μια εξαιρετική επιλογή για την εκπαίδευση και έχει υιοθετηθεί από πολλά πανεπιστήμια κυρίως στην Αμερική (Agarwal, et. al., 2008, Goldwasser, & Letscher, 2008, Shein, 2015, Guo, 2014). Πρόσφατα έχει αρχίσει να μπαίνει και στον χώρο της δευτεροβάθμιας εκπαίδευσης. Με την πιλοτική εφαρμογή της σε μαθητές γυμνασίου προσπαθήσαμε να διερευνήσουμε τα πλεονεκτήματα και τα μειονεκτήματα αυτής της γλώσσας σε πιο μικρές ηλικίες.

#### **4. Διδακτικές Στρατηγικές και προβλήματα**

Η πιλοτική διδασκαλία της Python έγινε σε ένα τμήμα 13 μαθητών της Γ' Γυμνασίου του Ζαννείου Πειραματικού Γυμνασίου το σχολικό έτος 2016-2017. Όλα τα μαθήματα έγιναν στο εργαστήριο πληροφορικής όπου 7 μαθητές είχαν στην διάθεσή τους από έναν υπολογιστή ενώ οι άλλοι 6 εργάζονταν ανά ζεύγη στους υπόλοιπους 3 υπολογιστές. Κάθε διδακτική ώρα οι μαθητές καθοδηγούνταν από κατάλληλο φύλλο εργασίας το οποίο περιείχε απλές δραστηριότητες στον διερμηνευτή της Python. Στις περισσότερες περιπτώσεις δεν έγινε καμία διάλεξη από τον/την καθηγητή/τρια αλλά αφήσαμε τους μαθητές μόνους τους να ερμηνεύσουν τα αποτελέσματα του διερμηνευτή. Οι μαθητές είχαν διδαχθεί στις προηγούμενες τάξεις προγραμματισμό στο περιβάλλον Scratch. Εδώ να σημειώσουμε ότι στο πειραματικό γυμνάσιο ακολουθείται το νέο πρόγραμμα σπουδών όπου η πληροφορική διδάσκεται δυο ώρες την εβδομάδα.

Οι προγραμματιστικές έννοιες που καλύψαμε μέσα στο σχολικό έτος ήταν οι εξής με αυτή τη σειρά : τύποι, εκφράσεις, αλφαριθμητικά, μεταβλητές, δομή επιλογής, συναρτήσεις, λίστες, δομή επανάληψης, αλγόριθμοι σειριακής και δυαδικής αναζήτησης, λεξικά.

Η διδασκαλία του προγραμματισμού με Python είναι κάτι καινούργιο στην δευτεροβάθμια εκπαίδευση. Λόγω της ιδιαιτερότητας της γλώσσας θα πρέπει να τροποποιήσουμε αρκετά ή να σχεδιάσουμε πάλι από την αρχή τις διδακτικές στρατηγικές που χρησιμοποιούσαμε τόσο καιρό. Στη συνέχεια παρουσιάζουμε τις βασικές διδακτικές πρακτικές που ακολουθήσαμε οι οποίες διαφέρουν αρκετά σε σχέση με αυτές που ξέραμε ως τώρα.

##### **4.1 Συναρτήσεις**

Η απλότητα του ορισμού μιας συνάρτησης στην Python μας διευκολύνει στην παρουσίαση του τμηματικού προγραμματισμού αρκετά νωρίς, αμέσως πριν ή μετά τη δομή επιλογής. Στους μαθητές ο ορισμός μιας συνάρτησης παρουσιάστηκε αρχικά

ως ορισμός μιας νέας εντολής ώστε να μην επαναλαμβάνεται κώδικας. Η δυνατότητα της αξιοποίησης του τμηματικού προγραμματισμού τόσο νωρίς βοήθησε αρκετά τη μαθησιακή διαδικασία όπως είδαμε στη συνέχεια. Για παράδειγμα οι μαθητές μπορούσαν να αποθηκεύουν στο ίδιο αρχείο όλες τις ασκήσεις τους με τη μορφή συναρτήσεων. Αξίζει να υπενθυμίσουμε εδώ ότι και η εκπαιδευτική γλώσσα Logo που είναι καθαρά συναρτησιακή γλώσσα εισάγει τις διαδικασίες πολύ νωρίς.

## 4.2 Λίστες

Η παρουσίαση της δομής δεδομένων Λίστα εννοιολογικά δεν προαπαιτεί την γνώση της δομής επανάληψης. Επίσης οι μαθητές πρέπει να γνωρίζουν την έννοια της λίστας γιατί η δομή επανάληψης

**for** στοιχείο **in** λίστα :

χτίζεται πάνω στην έννοια αυτή.

Οι μαθητές σε αυτό το σημείο γνώριζαν τις βασικές ιδιότητες των αλφαριθμητικών (strings) στην Python, κάποιες από τις οποίες έχουν και οι λίστες όπως η δεικτοδότηση από το 0, η πολυμορφική λειτουργία των τελεστών +,\* και το γεγονός ότι και στις δυο περιπτώσεις μιλάμε για μια ακολουθία αντικειμένων. Τα κοινά χαρακτηριστικά των δυο δομών μπορούν να χρησιμοποιηθούν ως μια σκαλωσιά μάθησης για την κατανόηση της έννοιας της λίστας.

## 4.3 Συγγραφή κώδικα

Η πληκτρολόγηση των εντολών στον διερμηνευτή της Python δεν ήταν τόσο μεγάλο πρόβλημα για τους μαθητές όσο περιμέναμε, λόγω της απλής της σύνταξης. Ωστόσο χρειάστηκε να τους επιστήσουμε την προσοχή πολλές φορές στο θέμα του ορισμού των μπλοκ εντολών με βάση τις εσοχές. Η λάθος στοίχιση ευθύνεται για λογικά λάθη τα οποία δεν είναι εύκολο να εντοπιστούν από τους μαθητές, όπως συμβαίνει στο παρακάτω παράδειγμα:

```
def sum( N ) :
    S = 0
    for i in range ( N+1 ) :
        S = S + i
    return S
>>> sum( 10 )
1
>>> sum( 256 )
1
>>> sum( 2016 )
1
```

όπου δημιουργείται η ψευδαίσθηση ότι η τελευταία εντολή είναι έξω από το σώμα της επαναληπτικής δομής.

Η οριοθέτηση των εντολών που ανήκουν σε ένα μπλοκ με βάση τη στοίχισή τους είναι κάτι ριζικά καινούργιο για τους περισσότερους καθηγητές και αρκετοί το αντιμετωπίζουν με σκεπτικισμό. Το πρόβλημα γίνεται πιο έντονο όταν ζητήσουμε από τους μαθητές να συντάξουν χειρόγραφα έναν αλγόριθμο σε python. Σε αυτή την περίπτωση η στοίχιση κάθε μπλοκ είναι ορατή από κάθετες γραμμές όπως φαίνεται παρακάτω:

```

for row in range( len( A ) ) :
    for col in range( len( A[row] ) ) :
        if A[row][col] == 0 :
            zeros = zeros + 1
print zeros

```

Αντίστοιχη οδηγία δόθηκε φέτος στις πανελλήνιες εξετάσεις του μαθήματος *Προγραμματισμός Υπολογιστών*. Με αυτό το τέχνασμα οι μαθητές μπορούν να εξεταστούν γραπτά ή να κρατήσουν σημειώσεις χωρίς να υπάρχει πρόβλημα με τη στοίχιση των εντολών.

Για να κατανοήσουν οι μαθητές τη σημασία της στοίχισης των εντολών τους δόθηκαν μικρά προγράμματα χωρίς καμία στοίχιση και τους ζητήθηκε να τα διορθώσουν θέτοντας πριν από κάθε εντολή τη σωστή εσοχή.

#### 4.4 Python 2 ή Python 3;

Οι δυο εκδόσεις της Python δεν έχουν πολύ σημαντικές διαφορές. Το σημαντικό πλεονέκτημα της Python 3 είναι η καλύτερη διαχείριση των ελληνικών χαρακτήρων και η πιο καθαρή σύνταξη.

<pre> # Python 2 &gt;&gt;&gt; word = 'Ζάννειο' &gt;&gt;&gt; word '\xc6\xdc\xed\xed\xe5\xe9\xef' </pre>	<pre> # Python 3 &gt;&gt;&gt; word = 'Ζάννειο' &gt;&gt;&gt; word 'Ζάννειο' </pre>
--	---

Ωστόσο κάποιες αλλαγές στην Python 3 δυσκολεύουν αρκετά τη διδασκαλία κάποιων εννοιών. Η εντολή print είναι πλέον συνάρτηση, ωστόσο αν θέλουμε να εμφανίσουμε πολλά αντικείμενα στην ίδια γραμμή η σύνταξη είναι λίγο πιο σύνθετη.

<pre> # Python 2 &gt;&gt;&gt; print number, </pre>	<pre> # Python 3 &gt;&gt;&gt; print ( number, end = "" ) </pre>
--	---

Ένα πιο σημαντικό πρόβλημα είναι ότι η input πλέον δεν μορφοποιεί αυτόματα το αντικείμενο εισόδου αλλά επιστρέφει πάντα ένα αλφαριθμητικό το οποίο είναι ευθύνη του μαθητή/προγραμματιστή να το μετατρέψει στον κατάλληλο τύπο δεδομένων.

**# Python 2**

```
>>> number = input('number = ')
```

Η πρακτική αυτή στα πρώτα μαθήματα προκάλεσε σύγχυση στους μαθητές διότι στο Γυμνάσιο οι μαθητές δεν διδάσκονται στα μαθηματικά τον συμβολισμό  $f(x)$  για τις συναρτήσεις. Αυτό καθιστά λίγο δύσκολο να κατανοήσουν εννοιολογικά την σύνθεση συναρτήσεων `int( input( ) )`. Η αυτόματη μετατροπή των στοιχείων εισόδου στον κατάλληλο τύπο με την `input` της έκδοσης 2 διευκόλυνε πολύ τους μαθητές στα πρώτα μαθήματα.

Το σημαντικότερο όμως πρόβλημα από διδακτικής πλευράς είναι ότι πλέον το αποτέλεσμα πολλών σημαντικών συναρτήσεων όπως είναι η πολύ χρήσιμη `range` δεν είναι μια λίστα αλλά ένα αντικείμενο, κάτι που έγινε για λόγους απόδοσης. Αυτή η αλλαγή καθιστά δύσκολη την παρουσίαση στους μαθητές της έννοιας της λίστας και της συνάρτησης `range` που χρειάζεται για τη δομή επανάληψης `for`.

**# Python 2**

```
>>> range( 6 )
```

```
[0, 1, 2, 3, 4, 5]
```

```
>>> type( range( 6 ) )
```

```
<type 'list'>
```

**# Python 3**

```
>>> number = int( input('number = ' ) )
```

**#Python 3**

```
>>> range( 6 )
```

```
range(0, 6)
```

```
>>> type( range( 6 ) )
```

```
<class 'range'>
```

```
>>> list( range( 6 ) )
```

```
[0, 1, 2, 3, 4, 5]
```

Για να δούμε μια λίστα στην Python πρέπει να χρησιμοποιήσουμε τον τελεστή/συνάρτηση `list` οπότε πάλι βρίσκουμε μπροστά μας την σύνθεση συναρτήσεων `list( range(..) )` με την οποία δυσανασχετούν πολλοί μαθητές.

Λόγω των παραπάνω προβλημάτων ενώ είχαμε ξεκινήσει με την έκδοση 3 λόγω της υποστήριξης ελληνικών, γρήγορα γυρίσαμε στην έκδοση 2.

#### 4.5 Ο διερμηνευτής του IDLE

Η χρήση του διερμηνευτή αποτελεί έναν από τους βασικούς πυλώνες πάνω στους οποίους στηρίζεται η διδακτική προσέγγιση που ακολουθήσαμε. Ο διερμηνευτής δίνει τη δυνατότητα στους μαθητές να ανακαλύψουν μόνοι τους μέσω πειραματισμών τις βασικές ιδιότητες των εντολών της Python.

Ωστόσο μια παρανόηση των μαθητών είναι ότι ο διερμηνευτής συμπεριφέρεται ως συντάκτης κώδικα. Έχουν την αίσθηση ότι μπορούν να διορθώσουν προηγούμενες λανθασμένες εντολές. Αντιμετωπίσαμε αυτή την παρανόηση με μια κατάλληλα σχεδιασμένη δραστηριότητα από τα πρώτα μαθήματα όπου οι μαθητές κλήθηκαν να εισάγουν λανθασμένες εντολές τις οποίες θα πρέπει στη συνέχεια να διορθώσουν.



#### 4.6 Οι δραστηριότητες

Πέρα από τα απλά φύλλα εργασίας που υλοποίησαν οι μαθητές μέσα στη χρονιά και τα οποία είχαν διερευνητικό χαρακτήρα δώσαμε στους μαθητές και 2-3 μεγάλες δραστηριότητες των 4-6 διδακτικών ωρών όπου οι μαθητές καλούνταν να υλοποιήσουν εφαρμογές-παίγνια όπως το *Πέτρα-Ψαλίδι-Χαρτί* και το παιχνίδι *Βρες τον Αριθμό* μέσα από το οποίο έγινε η εισαγωγή της σειριακής και της δυαδικής αναζήτησης. Η παρουσίαση της δυαδικής αναζήτησης έγινε με τη βοήθεια του παιχνιδιού όπως προτείνεται και στο σχολικό βιβλίο του μαθήματος “Προγραμματισμός Υπολογιστών” της Γ’ ΕΠΑΛ.

Κάτι που βοήθησε πολύ στην ανάπτυξη αυτών των παιχνιδιών ήταν η εύκολη χρήση της συνάρτησης `randint` της βιβλιοθήκης `random` της Python για την παραγωγή τυχαίων αριθμών. Τα φύλλα εργασίας που χρησιμοποιήσαμε είναι διαθέσιμα στην ιστοσελίδα <http://evripides.mysch.gr>.

#### 5. Συμπεράσματα

Στην εργασία αυτή παρουσιάσαμε τα κάποια θετικά και αρνητικά στοιχεία της πιλοτικής διδασκαλίας του προγραμματισμού με τη γλώσσα Python σε ένα τμήμα μαθητών του Ζαννείου Πειραματικού Γυμνασίου το σχολικό έτος 2016-2017. Ακολούθησαμε μια διερευνητική προσέγγιση με σημαντικό βαθμό καθοδήγησης από το φύλλο εργασίας. Η καθοδήγηση από τον εκπαιδευτικό κατά τη διάρκεια των δραστηριοτήτων ήταν πολύ μικρή και η ενθαρρύνουμε τη συνεργασία μεταξύ των ομάδων.

Τα αποτελέσματα ήταν πολύ ενθαρρυντικά, και επιβεβαίωσαν όχι μόνο το υψηλό ενδιαφέρον των μαθητών για τον προγραμματισμό με τη γλώσσα Python, αλλά και τα πλεονεκτήματα της γλώσσας για την ανάπτυξη εφαρμογών και την κατανόηση βασικών προγραμματιστικών εννοιών και τεχνικών. Η συνδρομή του διερμηνευτή της γλώσσας σε αυτή την προσπάθεια ήταν καταλυτική. Η αλληλεπίδραση και ο πειραματισμός των μαθητών με διαφορές εντολές στον διερμηνευτή ενίσχυσε τη διερευνητική αλλά και ανακαλυπτική πλευρά της μαθησιακής διαδικασίας. Σκοπεύουμε να συνεχίσουμε την εφαρμογή της διδακτικής αυτής προσέγγισης και τα επόμενα έτη σε περισσότερους μαθητές ώστε τα συμπεράσματά μας να είναι περισσότερο αξιοποιήσιμα.

Εν κατακλείδι η γλώσσα Python μπορεί να χρησιμοποιηθεί για τη διδασκαλία του προγραμματισμού σε μικρές ηλικίες με πολύ καλά αποτελέσματα εφόσον η διδασκαλία δεν γίνεται στον πίνακα αλλά στο εργαστήριο υπολογιστών με μια διερευνητική-πειραματική προσέγγιση.

## *Αναφορές*

Agarwal, K., Agarwal, A., & Celebi, E. (2008). Python puts a squeeze on java for CS0 and beyond. *J. Comput. Sci. Coll.* 23, 6, 49-57.

Goldwasser, M., & Letscher, D. (2008). Teaching an object-oriented CS1 -: with Python. *SIGCSE Bull.* 40, 3 (June 2008), 42-46.

Guo, P., (2014). *Python is now the Most Popular Introductory Teaching Language at Top U.S. Universities*. Survey published at the Communications of the ACM blog (CACM blog).

Jayal, A., Lauria, S., Tucker, A., & Swift, S. (2011). Python for teaching introductory programming: A quantitative evaluation, *ITALICS* , vol. 10, no. 1, pp. 86–90, 2011.

Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Comput. Surv.* 37, 2, 83-137.

McIver, L., & Conway, D. (1996). Seven Deadly Sins of Introductory Programming Language Design. In *Proceedings of the 1996 International Conference on Software Engineering: Education and Practice (SE:EP '96)*. IEEE Computer Society.

Shein, E. (2015). Python for beginners. *Commun. ACM* 58, 3, 19-21.

Αράπογλου, Α., Βραχνός, Α., Κανίδης, Ε., Μακρυγιάννης, Π., Μπελεσιώτης, Β., και Τζήμας, Δ. (2015). *Αρχές Προγραμματισμού Υπολογιστών – Β' ΕΠΑΛ*. ΙΤΥΕ Διόφαντος.

Αράπογλου, Α., Βραχνός, Α., Κανίδης, Ε., Λέκκα, Δ., Μακρυγιάννης, Π., Μπελεσιώτης, Β., Παπαδάκης, Σπ., και Τζήμας, Δ. (2016). *Προγραμματισμός Υπολογιστών – Γ' ΕΠΑΛ*. ΙΤΥΕ Διόφαντος.

Βραχνός, Ε., Κουρέτας, Ι., Μακρυγιάννης, Π., και Παραδείση, Α. (2015). *Ειδικά θέματα στον Προγραμματισμό Υπολογιστών – Γ' ΕΠΑΛ*. ΙΤΥΕ Διόφαντος.

### **Abstract**

Python is a modern programming language that has dominated the past few years in tertiary education and has recently been introduced into Greek secondary vocational education. Python is a script language that brings a new style of programming that leaves behind structural programming, static data structures and strict type systems. This new programming paradigm requires a new lab-centric and inquiry-based teaching approach that takes advantage of the special properties of the language and the interpreter of the programming environment. In this paper we present the results from the pilot teaching of Python programming to high-school students. The results we present are mainly qualitative and are divided into two categories. The didactic approaches we have designed and proposed as well as the problems we encountered in applying them.

**Keywords:** Python, programming, secondary education