Αλγοριθμική Python

Ευριπίδης Βραχνός

Ζάννειο Πειραματικό Γυμνάσιο / Πανεπιστήμιο Δυτικής Αττικής
evrachnos@{gmail.com, uniwa.gr}

Περίληψη

Στην εργασία αυτή παρουσιάζουμε κάποια προκαταρκτικά ποιοτικά αποτελέσματα από τον σχεδιασμό και την υλοποίηση ενός εισαγωγικού μαθήματος προγραμματισμού σε πρωτοετείς φοιτητές βιοϊατρικής με τη γλώσσα Python. 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 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. Στο μάθημα αυτό ακολουθήθηκε μια αλγοριθμική προσέγγιση και δεν δόθηκε ιδιαίτερη βαρύτητα στα ιδιαίτερα προγραμματιστικά χαρακτηριστικά της Python. Τα συμπεράσματα παρουσιάζουν ιδιαίτερο ενδιαφέρον, γιατί δείχνουν ότι κάποιοι φοιτητές δυσκολεύονται να εκμεταλλευτούν τα αλγοριθμικά χαρακτηριστικά της Python, τα οποία πολλές φορές είναι πιο κοντά σε ψευδογλώσσα, επειδή είναι δέσμιοι των προγραμματιστικών στερεοτύπων της ΓΛΩΣΣΑΣ που έχουν διδαχθεί στο Λύκειο.

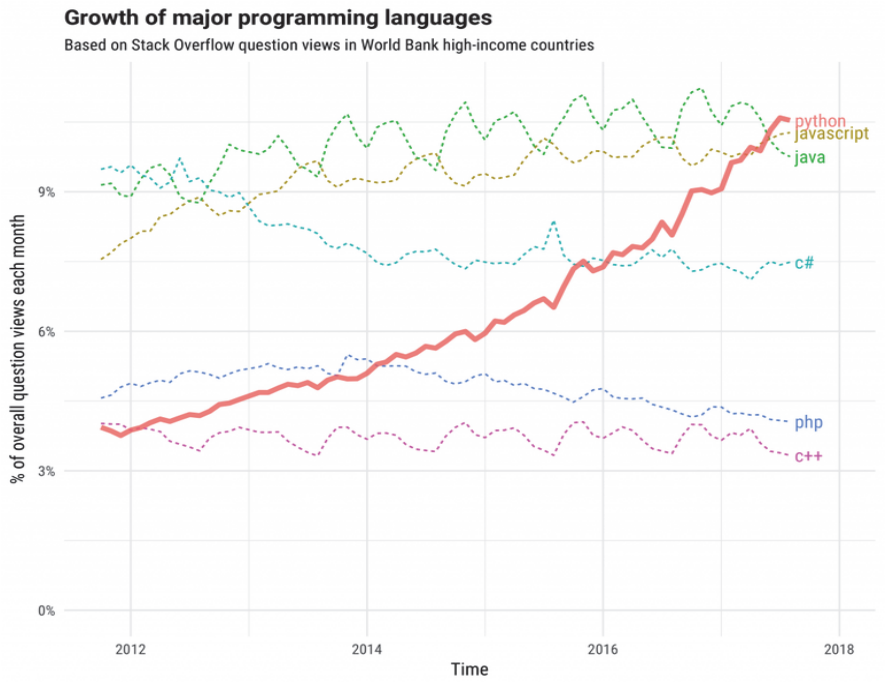
Λέξεις κλειδιά: Python, προγραμματισμός, τριτοβάθμια εκπαίδευση.

1. Εισαγωγή

Την τελευταία δεκαετία ο προγραμματισμός υπολογιστών και η αλγοριθμική έχουν γίνει απαραίτητα εφόδια για όλους τους θετικούς επιστήμονες και τους μηχανικούς. Για αυτό το λόγο σε όλες τις σχολές θετικών επιστημών, πολυτεχνείων ακόμα και οικονομικών προσφέρεται ένα τουλάχιστον μάθημα προγραμματισμού υπολογιστών. Η γλώσσα που κυριαρχεί τα τελευταία χρόνια στην τριτοβάθμια εκπαίδευση παγκοσμίως είναι η Python (Goldwasser, & Letscher, 2015; Guo, 2014; Shein, 2015; Simon et al., 2018). Εκτός από την απλότητα της σύνταξης και την απότομη καμπύλη μάθησης ένα άλλο μεγάλο πλεονέκτημά της Python είναι η πληθώρα βιβλιοθηκών που εμπλουτίζονται συνεχώς. Οι βιβλιοθήκες της Python επιτρέπουν την ανάπτυξη εφαρμογών σε όλα σχεδόν τα επιστημονικά και τεχνολογικά πεδία. Από την μοριακή βιολογία (Malmström, 2019) μέχρι την τεχνητή νοημοσύνη και τις οικονομικές επιστήμες (Aruoba, & Fernandez, 2015). Πρόσφατα μάλιστα έγινε γνωστό ότι το πρόγραμμα που δημιούργησε την πρώτη εικόνα μαύρης τρύπας ήταν γραμμένο εξ

ISBN: 978-960-578-058-6

ολοκλήρου σε Python και έκανε χρήση βιβλιοθηκών όπως matplotlib, numpy, scipy και άλλες (The EHT Collaboration et al., 2019).



Εικόνα 1. *Διαχρονική αποτύπωση της αναγνωσιμότητας γλωσσών στο Stack Overflow*

Ένα άλλο επιστημονικό πεδίο στο οποίο έχει αρχίσει να χρησιμοποιείται ευρύτατα η γλώσσα Python είναι αυτό των βιοϊατρικών επιστημών. Ένα χαρακτηριστικό παράδειγμα είναι η ανίχνευση καρκίνου με λογισμικό τεχνητής νοημοσύνης το οποίο σαρώνει την ακτινογραφία του ασθενούς (Coudray et al., 2018).

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 (Aleksic, & Ivanovic, 2016; Goldwasser, & Letscher, 2015; Guo, 2014; Shein, 2015; Simon et al., 2018).

Στην εργασία αυτή παρουσιάζουμε τα αποτελέσματα από τον σχεδιασμό και την υλοποίηση ενός νέου μαθήματος προγραμματισμού στη γλώσσα Python στους πρωτοετείς φοιτητές του τμήματος Μηχανικών Βιοϊατρικής του Πανεπιστημίου Δυτικής Αττικής. Η προσέγγιση που υιοθετήσαμε στηρίχθηκε σε δραστηριότητες διερευνητικού χαρακτήρα που έγιναν στο εργαστήριο πληροφορικής. Τα

ISBN: 978-960-578-058-6

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 του προγραμματισμού.

2. Η εξέλιξη της Python στην τριτοβάθμια εκπαίδευση

Ήδη από το 1999 ο Zelle(1999) ισχυριζόταν ότι η Python είναι η ιδανική γλώσσα για την εκπαίδευση εξαιτίας της απλότητάς της η οποία την κάνει να μοιάζει με ψευδογλώσσα. Έτσι οι φοιτητές μπορούν να επικεντρωθούν στην αλγοριθμική επίλυση προβλημάτων και όχι στις ιδιαιτερότητες της γλώσσας προγραμματισμού. Λίγα χρόνια αργότερα η Patterson-McNeil (2006) μετά από δέκα χρόνια μιας αλυσίδας τριών μαθημάτων με τη C++ αποφάσισε να αλλάξει προσέγγιση. Το πρώτο μάθημα ήταν εισαγωγικό στον προγραμματισμό, το δεύτερο στον αντικειμενοστρεφή προγραμματισμό και το τρίτο στις δομές δεδομένων. Μέχρι τότε μόνο το 10% των φοιτητών κατάφερνε να ολοκληρώσει επιτυχώς και τα τρία μαθήματα. Με τη μετάβαση σε Python το ποσοστό επιτυχίας και στα τρία μαθήματα ανέβηκε στο 40%. Σε άλλη περίπτωση (Leping et al., 2009) η μετάβαση από την Java στην Python είχε σαν αποτέλεσμα να μειωθεί το ποσοστό αποτυχίας των φοιτητών από 28% σε 14%. Σχεδόν όλες οι έρευνες που έγιναν κατά τη μετάβαση από μια γλώσσα προγραμματισμού στην Python έχουν θετικά αποτελέσματα (Ateeq et al., 2014; Koulouri, Lauria, & Macredie, 2015; Wainer, & Xavier, 2018). Μάλιστα σε κάποιες περιπτώσεις (Jayal et al., 2009) τα αποτελέσματα ήταν εντυπωσιακά. Ο Jayal πέτυχε να μειώσει το ποσοστό αποτυχίας που είχε το μάθημά του με τη Java από το υψηλό 50% στο 10% με την Python, χωρίς να αλλάξει ιδιαίτερα το πρόγραμμα σπουδών και συνεχίζοντας να ακολουθεί το αντικειμενοστρεφές υπόδειγμα.

3. Σχεδιασμός του μαθήματος

Το τμήμα Μηχανικών Βιοϊατρικής του Πανεπιστημίου Δυτικής Αττικής προσφέρει στους πρωτοετείς φοιτητές δυο μαθήματα προγραμματισμού υπολογιστών. Μέχρι πέρυσι και τα δυο μαθήματα γίνονταν στο Matlab, το οποίο είναι χρήσιμο για εφαρμογές επεξεργασίας σήματος και ιατρικής εικόνας. Ωστόσο τα τελευταία χρόνια με την εξέλιξη της Python στο πεδίο αυτό το δεύτερο μάθημα σχεδιάστηκε εξαρχής στη γλώσσα Python. Το μάθημα που γίνεται στο πρώτο εξάμηνο είναι εστιασμένο στις μαθηματικές εφαρμογές του Matlab, όπως για παράδειγμα στον λογισμό πινάκων και όχι στην επίλυση αλγοριθμικών προβλημάτων. Έτσι το δεύτερο μάθημα επικεντρώθηκε στην εισαγωγή των φοιτητών στην αλγοριθμική επίλυση προβλημάτων με τη γλώσσα Python. Οι ενότητες που καλύφθηκαν στο μάθημα ήταν οι τύποι της Python και οι μεταβλητές, οι δομές επιλογής και επανάληψης, οι συναρτήσεις και από τις δομές δεδομένων οι λίστες, τα αλφαριθμητικά και οι πλειάδες. Επίσης μελετήθηκαν οι βασικοί αλγόριθμοι αναζήτησης και ταξινόμησης.

ISBN: 978-960-578-058-6

Στα τελευταία μαθήματα παρουσιάστηκαν οι βιβλιοθήκες numpy, matplotlib, scipy και pandas και έγινε μια σύνδεση των λειτουργιών τους με τις αντίστοιχες στο Matlab.

Το μάθημα αποτελείται από 2 ώρες θεωρία και 2 ώρες εργαστήριο που γίνονται κάθε εβδομάδα. Επίσης στους φοιτητές δόθηκαν και κάποιες προαιρετικές εργασίες για περαιτέρω εξοικείωση με τη γλώσσα.

Τα συμπεράσματα στα οποία καταλήξαμε μέσα από τη διδασκαλία του μαθήματος προέκυψαν από τέσσερις πηγές.

- Τις συζητήσεις που έγιναν κατά τη διάρκεια των διαλέξεων πάνω σε απορίες των φοιτητών ή σε θέματα που έθετε ο διδάσκων.
- Τις δυσκολίες που αντιμετώπισαν οι φοιτητές στα εργαστήρια.
- Τη μελέτη των γραπτών της εξέτασης του μαθήματος.
- Τις συνεντεύξεις που πήραμε επιλεκτικά από φοιτητές με κριτήριο τις απαντήσεις τους στα θέματα των εξετάσεων.

4. Ανάλυση των Αποτελεσμάτων

Τα αποτελέσματα που συγκεντρώσαμε από τις παραπάνω πηγές, κατηγοριοποιήθηκαν με βάση την ενότητα, π.χ. παρανοήσεις στις μεταβλητές, δυσκολίες στις δομές επανάληψης κλπ. Τα σημαντικότερα από αυτά τα αποτελέσματα παρουσιάζονται σε αυτή την παράγραφο. Εδώ πρέπει να σημειωθεί ότι παρακάτω δεν θα αναφερθούμε στη χρησιμότητα της εντολής διακλάδωσης break διότι αυτή αποτελεί το αντικείμενο άλλης εργασίας (Βραχνός, 2019), όπου αναλύεται και τεκμηριώνεται η χρησιμότητα αυτής της υποτιμημένης εντολής.

4.1 Ο διερμηνευτής της Python και το 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 φοιτητές να ανακαλύψουν μόνοι τους μέσω πειραματισμών τις βασικές ιδιότητες των εντολών της Python. Σε κάθε εργαστήριο δινόταν στους φοιτητές ένα φύλλο εργασίας το οποίο καθοδηγούσε τους φοιτητές να αξιοποιήσουν τον διερμηνευτή, διορθώνοντας μόνοι τους τα λάθη τους. Φυσικά σε πολλές περιπτώσεις κάποιοι φοιτητές ζητούσαν τη βοήθεια του διδάσκοντα. Τα σημαντικότερα προβλήματα που εντοπίσαμε κατά τη χρήση του περιβάλλοντος προγραμματισμού IDLE ήταν δυο:

Το πρώτο πρόβλημα ήταν η παρανόηση που είχαν οι φοιτητές ότι ο διερμηνευτής συμπεριφέρεται ως συντάκτης κώδικα. Όταν οι φοιτητές έκαναν κάποιο λάθος προσπαθούσαν να διορθώσουν την προηγούμενη εντολή αντί να την ξαναγράψουν

ISBN: 978-960-578-058-6

από την αρχή. Δεν είχαν καταλάβει ότι αυτό που έβλεπαν στην οθόνη ήταν το ιστορικό των εντολών που έχουν δώσει το οποίο δεν μπορεί να αλλάξει. Ωστόσο με μερικά παραδείγματα και με τη χρήση της συντόμευσης Alt+P το πρόβλημα λύθηκε. Να σημειωθεί εδώ ότι το ίδιο πρόβλημα είχαν αντιμετωπίσει και μαθητές (Βραχνός & Κατσένη, 2017) στο πρώτο μάθημα με τον διερμηνευτή της Python.

Το δεύτερο πρόβλημα που συναντήσαμε όπως ήταν αναμενόμενο είχε να κάνει με τη σωστή στοίχιση των εντολών κάτι που στην Python μπορεί να προκαλέσει συντακτικά λάθη. Πολλοί φοιτητές είτε δεν πετύχαιναν τη σωστή στοίχιση είτε ξεχνούσαν το σύμβολο ':' που υποδηλώνει την αρχή ενός μπλοκ εντολών. Αυτό το πρόβλημα πήρε λίγο περισσότερο χρόνο να διορθωθεί γιατί δεν ήταν μια απλή παρανόηση αλλά κάτι που ήθελε περισσότερη εξοικείωση. Μάλιστα στις γραπτές εξετάσεις πολλοί φοιτητές ακολούθησαν τις οδηγίες που τους δόθηκαν και χρησιμοποίησαν κάθετες γραμμές για να οριοθετήσουν ένθετα μπλοκ εντολών, όπως φαίνεται παρακάτω.

```
for row in Matrix :
    for item in row :
        if  item == 0 :
            zeros = zeros + 1
print(zeros)
```

4.2 Το παράδοξο των δομών επανάληψης

Όταν οι φοιτητές χρειάστηκε να χρησιμοποιήσουν τη δομή επανάληψης for για να σαρώσουν τα στοιχεία μιας λίστας είχαν τη δυνατότητα να επιλέξουν τον κλασικό τρόπο με χρήση της range ο οποίος παραπέμπει στην εντολή Για…από…μέχρι:

Python	Ψευδοκώδικας

`for i in range( len( List ) ) :`
`  if  List[i] == 0 :`
`  zeros = zeros + 1`
`print(zeros)`
	`Για  i από 1 μέχρι len (List)–1  :`
	`  Αν List[i] = 0 :`
	`  μηδενικά = μηδενικά + 1`
	`Γράψε μηδενικά`

ή το ιδίωμα της Python for…in που παραπέμπει στην foreach της Java

Python	Ψευδοκώδικας

`for item in List :`
`  if  item == 0 :`
`  zeros = zeros + 1`
`print(zeros)`
	`Για κάθε στοιχείο item της λίστας List :`
	`  Αν item = 0 :`
	`  μηδενικά = μηδενικά + 1`
	`Γράψε μηδενικά`

ISBN: 978-960-578-058-6

και θεωρείται πολύ πιο επεξηγηματικό και πιο κοντά στον ανθρώπινο τρόπο σκέψης.

Αρκετοί μαθητές χρησιμοποιήσαν την πρώτη έκδοση η οποία είναι εμφανώς πιο σύνθετη από τη δεύτερη και σίγουρα λιγότερο επεξηγηματική. Αυτό το παρατηρήσαμε αρχικά στις ασκήσεις στο εργαστήριο και το σημειώσαμε αλλά μάλλον δεν δώσαμε τη δέουσα προσοχή. Όταν επαναλήφθηκε στις εξετάσεις τότε καλέσαμε τους φοιτητές που προτίμησαν αυτόν τον τρόπο και τους ρωτήσαμε γιατί ακολούθησαν αυτή την προσέγγιση. Η απάντηση ήταν ότι τους φαίνεται πιο οικείος αυτός ο τρόπος γιατί τον έχουν διδαχθεί στο σχολείο στο πλαίσιο του μαθήματος 'Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον' της Γ' Λυκείου. Αυτό που προκαλεί εντύπωση είναι ότι οι φοιτητές του τμήματος προέρχονται όλοι από την θετική κατεύθυνση, που σημαίνει ότι το μάθημα για αυτούς δεν ήταν πανελλαδικά εξεταζόμενο. Τη μόνη εξαίρεση αποτελούν οι φοιτητές που προέρχονται από την Κύπρο οι οποίοι χρησιμοποιήσαν την προσέγγιση foreach. Αυτό είναι ένα παράδειγμα που παραπέμπει την εποικοδομιστική θεωρία σύμφωνα με την οποία οι νέες γνώσεις χτίζονται πάνω σε προϋπάρχουσες γνώσεις ή εμπειρίες των φοιτητών.

4.3 Επιστροφή από συναρτήσεις

Ένα άλλο σημαντικό θέμα που χρήζει περισσότερης διερεύνησης είναι οι δυσκολίες που αντιμετώπισαν οι φοιτητές στον ορισμό και τη χρήση των δικών τους συναρτήσεων. Πέρα από τις συνήθεις δυσκολίες που αντιμετωπίζουν οι αρχάριοι προγραμματιστές στον τμηματικό προγραμματισμό παρατηρήσαμε μια ακόμα παρανόηση η οποία οφειλόταν στην υπερβολική χρήση του διερμηνευτή της Python. Δίνουμε το παρακάτω παράδειγμα για να αναδείξουμε το πρόβλημα:

```
>>> a = 10
>>> a              # Αποτίμηση του a
10
>>> print(a)       # Εκτύπωση και αποτίμηση του a
10
```

Τώρα ορίζουμε δυο συναρτήσεις

```
def return_square(x):
    return x*x

def print_square(x):
    print( x*x )
```

Και όπως φαίνεται παρακάτω αν τις εκτελέσουμε στον διερμηνευτή φαινομενικά έχουν το ίδιο αποτέλεσμα.

```
>>> return_square( 4 )
16                  # Επιστρέφει ως τιμή το 16
>>> print_square( 4 )
16                  # Επιστρέφει None, εκτυπώνει 16
```

ISBN: 978-960-578-058-6

κάτι που ξεγελάει τους φοιτητές και τους κάνει να συγχέουν τη λειτουργία της print σε μια συνάρτηση με τη λειτουργία της return. Το συγκεκριμένο λάθος παρατηρήθηκε στο 25% των φοιτητών. Προφανώς δεν μπορούμε να πούμε με βεβαιότητα ότι φταίει η γλώσσα ή ο διερμηνευτής. Το πιθανότερο είναι ότι έπρεπε να ακολουθήσουμε διαφορετική διδακτική στρατηγική και να δώσουμε περισσότερο βάρος στην λειτουργία της return.

4.4 Πλεονεκτήματα των λιστών

Ένα μεγάλο πλεονέκτημα που έχουν οι λίστες στην Python είναι ότι βρίσκονται πιο κοντά στην ανθρώπινη σκέψη παρά στη μηχανή. Δηλαδή ο άνθρωπος σκέφτεται ένα σύνολο αντικειμένων ως μια δυναμική λίστα και όχι ως ένα στατικό πίνακα. Αυτό φάνηκε όταν δώσαμε στις εξετάσεις στους φοιτητές την παρακάτω άσκηση:

Να αναπτύξετε μια συνάρτηση σε Python η οποία θα δέχεται δυο λίστες και θα επιστρέφει σε μια νέα λίστα τους κοινούς θετικούς αριθμούς των δυο λιστών. Κάθε θετικός αριθμός να εμφανίζεται στη νέα λίστα ακριβώς μια φορά.

```
def positive_union( L, W) :
    Result = [ ]
    for item in L :
        if item > 0 and item in W and item not in Result :
            Result = Result + [ item ]
    return Result
```

Ο παραπάνω αλγόριθμος δείχνει πόσο κοντά βρίσκεται η Python σε αυτό που λέμε Αλγοριθμική. Ο τρόπος σκέψης που περιγράφεται είναι ακριβώς ίδιος με τον τρόπο σκέψης που θα χρησιμοποιούσε κάποιος που δεν γνωρίζει προγραμματισμό για την επίλυση αυτού τους προβλήματος. Και μόνο η σκέψη της υλοποίησης αυτού του αλγορίθμου σε μια γλώσσα όπως η Pascal , η C ή η σχολική ΓΛΩΣΣΑ φαντάζει αρκετά πολύπλοκη για έναν αρχάριο προγραμματιστή.

5. Συμπεράσματα

Στην εργασία αυτή παρουσιάσαμε κάποια προκαταρκτικά ποιοτικά αποτελέσματα από τον σχεδιασμό και την υλοποίηση ενός εισαγωγικού μαθήματος προγραμματισμού σε πρωτοετείς φοιτητές βιοϊατρικής με τη γλώσσα Python. Στο μάθημα αυτό ακολουθήθηκε μια αλγοριθμική προσέγγιση και δεν δόθηκε ιδιαίτερη βαρύτητα στα ιδιαίτερα προγραμματιστικά χαρακτηριστικά της Python. Τα συμπεράσματα παρουσιάζουν ιδιαίτερο ενδιαφέρον, γιατί δείχνουν ότι κάποιοι φοιτητές δυσκολεύονται να εκμεταλλευτούν τα αλγοριθμικά χαρακτηριστικά της

ISBN: 978-960-578-058-6

γιατί έχουν ήδη δημιουργήσει νοητικά μοντέλα που έρχονται σε σύγκρουση με τη λογική της Python. Αυτό φάνηκε κυρίως στη χρήση των δομών επανάληψης όπου οι φοιτητές προτίμησαν να χρησιμοποιήσουν μια δομή τη μορφής Για…από…μέχρι σε συμφωνία με την ΓΛΩΣΣΑ του Λυκείου σε αντίθεση με το εξαιρετικά απλό ιδίωμα for…in της python. Επίσης οι φοιτητές δυσκολεύτηκαν σε θέματα που είχαν να κάνουν με τη λειτουργία του διερμηνευτή ο οποίος είναι εν μέρει υπεύθυνος και για μια παρανόηση που εντοπίσαμε αναφορικά με την εντολή return στον ορισμό μιας συνάρτησης.

Εν κατακλείδι μπορούμε να πούμε ότι τα χαρακτηριστικά της Python διευκόλυναν πολύ τους φοιτητές στην ανάπτυξη αλγορίθμων, καταδεικνύοντας ότι η Python είναι μια αλγοριθμική γλώσσα με τον ελάχιστο συντακτικό θόρυβο η οποία επικεντρώνει στην αλγοριθμική επίλυση των προβλημάτων. Προς την κατεύθυνση αυτή βοηθούν και οι λίστες αντί για τους πίνακες που χρησιμοποιούν άλλες γλώσσες. Οι λίστες είναι πιο κοντά στη φυσική αναπαράσταση ενός συνόλου απτών αντικειμένων και για αυτό οι αλγόριθμοι που έχουν να κάνουν με συγχώνευση ή διαχωρισμό λιστών υλοποιούνται πολύ πιο απλά στην Python όπως φάνηκε στα εργαστήρια του μαθήματος αλλά και στις εξετάσεις.

Αναφορές

Aleksic, V. & Ivanovic, M. (2016). Introductory Programming Subject in European Higher Education, *Informatics in Education, 15*(2), 163-182.

Aruoba, S. & Fernandez., V. J. (2015). A comparison of programming languages in macroeconomics, *Journal of Economic Dynamics and Control, 58* , pp. 265-273.

Ateeq, M., Habib, H., Umer, A., & Rehman, M. U. (2014) C++ or Python? Which One to Begin with: A Learner's Perspective. In 2014 International Conference on Teaching and Learning in Computing and Engineering, IEEE.

Goldwasser, M., & Letscher, D. (2008). Teaching an object-oriented CS1 -: with Python. *SIGCSE Bulletin. 40*(3), 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. *Innovation in Teaching and Learning in Information and Computer Sciences*, *10*(1), pp. 86–90.

ISBN: 978-960-578-058-6

Kaplan, R. (2010). Choosing a first programming language. In Proceedings of the 2010 ACM conference on Information technology education (SIGITE '10). ACM, New York, NY, USA, 163-164

Koulouri, T., Lauria, S., & Macredie, R. (2015). Teaching Introductory Programming: A Quantitative Evaluation of Different Approaches. *ACM Transactions on Computing Education*. *14*(4).

Leping, V., Lepp, M., Niitsoo, M., Tonisson, E., Vene, V., & Villems, A. (2009) Python Prevails. In Proceedings of the International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing (New York, NY, USA, 2009), CompSysTech '09, ACM, pp. 87:1-87:5.

Malmström L. (2019) Computational Proteomics with Jupyter and Python. In: Evans C., Wright P., Noirel J. (eds) Mass Spectrometry of Proteins. Methods in Molecular Biology, vol 1977. Humana Press, New York, NY

Patterson-McNeill, H. (2006). Experience: From C++ to Python in 3 Easy Steps. *J. Comput. Sci. Coll. 22*(2) 92-96.

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

Simon, Mason, R., Crick, T., Davenport, J., & Murphy, E. (2018). Language Choice in Introductory Programming Courses at Australasian and UK Universities. In Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18). ACM, New York, 852-857.

The EHT Collaboration et al. (2019). "First M87 Event Horizon Telescope Results. III. Data Processing and Calibration." ApJL, 875, Pp. 3. Publisher's Version

Coudray, N., Ocampo, P., Sakellaropoulos, T., Narula, N., Snuderl, M., Fenyö, D., Moreira, S., Razavian, N. & Tsirigos, A. (2018). Classification and mutation prediction from non–small cell lung cancer histopathology images using deep learning, *Nature Medicine 24*, 1559-1567.

Wainer, J., & Xavier, E. C. (2018). A Controlled Experiment on Python vs C for an Introductory Programming Course. *ACM Transactions on Computing Education 18*(3), 1-16.

Zelle, J. M. (1999). Python as a First Language. Department of Mathematics, Computer Science, and Physics Wartburg College.
https://mcsp.wartburg.edu/zelle/python/python-first.html

ISBN: 978-960-578-058-6

Βραχνός, Ε. & Κατσένη, Μ. (2017). Αποτίμηση της πιλοτικής διδασκαλίας του προγραμματισμού με τη γλώσσα Python σε μαθητές Γυμνασίου. 9th Conference on Informatics in Education, σελ. 80-90.

Βραχνός, Ε. (2019). Υπερβαίνοντας τα στερεότυπα του δομημένου προγραμματισμού. 11th Conference on Informatics in Education.

Abstract

In this article we present some preliminary qualitative results that arise from the design and implementation of an introductory programming course with Python to Biomedical Engineering students. Python is a script language that brings a new style of programming that leaves behind structural programming, static data structures and strict type systems. Python brings a completely different philosophy not only to the way we program but mainly to the way we teach programming. This course has adopted an algorithmic approach and did not give particular emphasis on Python's programming features. The findings are particularly interesting because they show that some students find it difficult to exploit Python's algorithmic features, which are often closer to pseudo-language, because they are bound by structured programming stereotypes taught in Lyceum.

Keywords: Python, computer science, programming, Algorithmic design.

ISBN: 978-960-578-058-6