ePublishing
ΕΘΝΙΚΟ ΚΕΝΤΡΟ ΤΕΚΜΗΡΙΩΣΗΣ NATIONAL DOCUMENTATION CENTRE

Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση

Vol 1 (2016)

8ο Πανελλήνιο Συνέδριο Διδακτική της Πληροφορικής

Η Python και η «τέχνη του υπολογίζειν»: πρόταση για ένα λεξικό μοντέλων διδασκαλίας της γλώσσας

Σταύρος Δημητριάδης

To cite this article:

Δημητριάδης Σ. (2022). Η Python και η «τέχνη του υπολογίζειν»: πρόταση για ένα λεξικό μοντέλων διδασκαλίας της γλώσσας. *Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση*, *1*, 107–114. Retrieved from https://eproceedings.epublishing.ekt.gr/index.php/cetpe/article/view/3897

https://epublishing.ekt.gr | **e-Publisher:** EKT | **Downloaded at:** 26/02/2026 00:46:03

Η Python και η «τέχνη του υπολογίζειν»: πρόταση για ένα λεξικό μοντέλων διδασκαλίας της γλώσσας

Σταύρος Δημητριάδης
sdemetri@csd.auth.gr
Τμήμα Πληροφορικής, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Περίληψη

Η εισαγωγή της γλώσσας Προγραμματισμού Python στην ελληνική Επαγγελματική Εκπαίδευση γεννά την ανάγκη για έναν κατάλληλο διδακτικό μετασχηματισμό της γλώσσας, κάτι που δημιουργεί μια πρόσθετη ευθύνη για τον εκπαιδευτικό. Το παρόν άρθρο παρουσιάζει αρχικά βασικά στοιχεία δομής και λειτουργίας της γλώσσας υποστηρίζοντας την άποψη πως οι ιδιαιτερότητες της Python (ως γλώσσας δυναμικού τύπου) δημιουργούν ένα ευνοϊκό πλαίσιο για εμβάθυνση στην «τέχνη του υπολογίζειν», δηλαδή τη συγκριτική ανάλυση των διαφορετικών μορφών έκφρασης της υπολογιστικής σκέψης, έτσι όπως υλοποιούνται από ποικίλα εργαλεία (γλώσσες) Προγραμματισμού. Στη συνέχεια παρουσιάζονται έξι μοντέλα διδασκαλίας της γλώσσας («ψευδογλώσσας», «απλουστευτικό», «εργαλειακό», «επεξηγηματικό», «αντικειμενοστρεφές», και «μεταπρογραμματιστικό») σχολιάζοντας τις δυνατότητες, περιορισμούς και ιδιαίτερες προϋποθέσεις εφαρμογής του καθενός. Τα μοντέλα προτείνονται ως εργαλεία διδακτικού μετασχηματισμού για την αποδοτική αξιοποίηση της Python στην εκπαίδευση.

Λέξεις κλειδιά: Python, υπολογιστική σκέψη, Διδακτική Προγραμματισμού

Εισαγωγή

Στην Python η αντιμετάθεση τιμών δύο μεταβλητών a και b μπορεί να υλοποιηθεί απλά με μία εντολή της μορφής: a, b = b, a. Στρατιές εκπαιδευτικών Πληροφορικής που δίδαξαν στους μαθητές τους την αντιμετάθεση με χρήση της ενδιάμεσης μεταβλητής temp στέκονται προβληματισμένοι καθώς σκέφτονται: «πρέπει να διδάξω τώρα αυτό;»

Το παρόν άρθρο επιχειρεί μία εισαγωγική ανάλυση των τρόπων («μοντέλων») διδασκαλίας της Python με αφορμή την εισαγωγή της γλώσσας στην ελληνική Επαγγελματική Εκπαίδευση (και ίσως κάποια στιγμή και τη γενική) και την ανάγκη επιμόρφωσης των εκπαιδευτικών Πληροφορικής που καλούνται να τη διδάξουν. Την ανάλυση καθιστά σημαντική το γεγονός πως η Python ως γλώσσα δυναμικής διαχείρισης τύπων (ή απλά «δυναμικού τύπου») εφαρμόζει ιδιαίτερες μορφές υπολογιστικής αναπαράστασης και αυτό έρχεται σε σύγκρουση με τα νοητικά μοντέλα κατανόησης του Προγραμματισμού που συνήθως βασίζονται σε αντίστοιχες αναπαραστάσεις των γλωσσών στατικού τύπου (π.χ. C/C++, Java).

Θέση του συγγραφέα είναι πως τα ιδιαίτερα αυτά χαρακτηριστικά της Python (και γενικότερα των γλωσσών δυναμικού τύπου) αγγίζουν βαθύτερα επίπεδα της υπολογιστικής σκέψης και γεννούν έναν εξαιρετικά γόνιμο προβληματισμό σχετικά με το πώς αυτή εκφράζεται και προάγεται. Έτσι, το άρθρο τονίζει την άποψη ότι η ανθρώπινη ανάγκη για υπολογιστική σκέψη δεν ταυτίζεται με μια συγκεκριμένη αναπαράσταση, εργαλείο ή τεχνική Προγραμματισμού (κατ' αναλογία με την ανάγκη για εικαστική έκφραση η οποία δεν ταυτίζεται με συγκεκριμένα εργαλεία ή τεχνικές του εικαστικού). Η «τέχνη του υπολογίζειν» (όπως και κάθε άλλη τέχνη) περιλαμβάνει σαφώς και ένα επίπεδο ανθρώπινης δραστηριότητας που αφορά τη δημιουργία, επιλογή, συγκριτική αξιολόγηση και διδακτικό μετασχηματισμό των ποικίλων εργαλείων έκφρασης της υπολογιστικής σκέψης. Συνεπώς, ο

Τ. Α. Μικρόπουλος, Α. Τσιάρα, Π. Χαλκή (επιμ.), *Πρακτικά 8ου Πανελλήνιου Συνεδρίου «Διδακτική της Πληροφορικής»*, Ιωάννινα: ΕΤΠΕ. 23-25 Σεπτεμβρίου 2016. ISSN 2529-0908, ISBN 978-960-88359-9-3.

εκπαιδευτικός Πληροφορικής (ως «καλλιτέχνης του υπολογίζειν») θα πρέπει να μετασχηματίζει διδακτικά τα εργαλεία και τις τεχνικές υπολογισμού, ώστε να ανταποκρίνεται με επιτυχία σε ιδιαίτερους στόχους και επίπεδα μάθησης. Τελικός –και πρακτικός- στόχος του άρθρου είναι να αποκρυσταλλώσει αυτό τον προβληματισμό σε μια σειρά μοντέλων διδασκαλίας της Python, ώστε να προσφέρει στον εκπαιδευτικό ένα χρήσιμο εργαλείο διαφοροποίησης της διδακτικής προσέγγισης ανάλογα με τις εκάστοτε ανάγκες και χαρακτηριστικά των μαθητών.

Υπολογιστική σκέψη

Ο όρος «υπολογιστική σκέψη» (ΥΣ) (computational thinking) πρωτοαναφέρθηκε από τον Papert (1996) αλλά επανήλθε στην επικαιρότητα μετά τα άρθρα της Wing (2006), η οποία υποστήριξε πως η υπολογιστική σκέψη είναι μια γενική ανθρώπινη δεξιότητα, ισότιμη με τις παραδοσιακές δεξιότητες αλφαβητισμού (γραφή, ανάγνωση, αριθμητική) και επομένως θα πρέπει να καλλιεργείται στον μαθητικό πληθυσμό προς όφελος των ατόμων αλλά και των κοινωνικών ομάδων γενικότερα. Για το τι συνθέτει την υπολογιστική σκέψη έχουν γραφεί πολλά άρθρα, χωρίς οι συγγραφείς να συμφωνούν πάντοτε για το ακριβές περιεχόμενο του όρου. Ένας απλός και πρακτικός ορισμός της ΥΣ προτείνεται από τη διεθνή κοινότητα εκπαιδευτικών Πληροφορικής (Operational Definition of Computational Thinking, 2011), σύμφωνα με τον οποίο η υπολογιστική σκέψη είναι μια δεξιότητα επίλυσης προβλήματος που περιλαμβάνει μεταξύ άλλων:

- Διαμόρφωση ενός προβλήματος με τρόπο που να επιτρέπει τη χρήση υπολογιστή για την επίλυσή του.
- Λογική οργάνωση και ανάλυση δεδομένων.
- Αναπαράσταση δεδομένων με αφηρημένες δομές όπως μοντέλα και προσομοιώσεις.
- Αυτοματοποίηση λύσεων μέσω αλγοριθμικής σκέψης.

Προφανώς η αναπαράσταση δεδομένων και ο αλγόριθμος είναι σημαντικά στοιχεία της υπολογιστικής σκέψης, όμως, κανένα από αυτά δεν ορίζονται απολύτως μονοσήμαντα. Ποικίλες αναπαραστάσεις και αλγόριθμοι είναι διαθέσιμοι, καθένα με τα πλεονεκτήματα και μειονεκτήματά του (τις «ιδιαιτερότητές» του) και είναι θέμα της κοινότητας το πώς αποφασίζει να υιοθετήσει συγκεκριμένες μορφές τους για να καλύψει αντίστοιχες ανάγκες καθώς αυτές αναφύονται ή μετασχηματίζονται. Στην περίπτωση των γλωσσών Προγραμματισμού η διαφορετική διαχείριση της έννοιας του «τύπου» και της «μεταβλητής» (και ό,τι αυτό συνεπάγεται στη συνέχεια) που κάνουν οι γλώσσες στατικού και δυναμικού τύπου καταδεικνύει, ότι οι όποιες «σταθερές» της υπολογιστικής σκέψης θα πρέπει να αναζητηθούν σε ακόμη υψηλότερο επίπεδο αφαίρεσης. Αυτό έχει ενδιαφέρουσες συνέπειες για την εκπαίδευση.

Τα βασικά χαρακτηριστικά της Python

Ιστορικά

Η Python εμφανίστηκε επίσημα τη δεκαετία του 1990: η έκδοση 1.0 τον Ιανουάριο του 1994 και ακολούθησαν η έκδοση 2.0 τον Οκτώβριο του 2000, ενώ η πιο πρόσφατη 3.0 τον Δεκέμβριο του 2008 (History of Python, n.d.). Η γλώσσα προτάθηκε εξ αρχής ως γλώσσα για την εύκολη εκμάθηση Προγραμματισμού από όλους αλλά και ως γλώσσα σεναρίων (scripting). Στην πορεία, η εκτίμηση ότι ο χρόνος του ανθρώπου προγραμματιστή είναι πολύτιμος έστρεψε το ενδιαφέρον σε γλώσσες οι οποίες (όπως η Python) μειώνουν τον χρόνο ανάπτυξης και εκσφαλμάτωσης κώδικα με την απλή και εύκολα κατανοητή σύνταξή τους.

Έτσι, σήμερα η Python βρίσκεται σταθερά μεταξύ των 5 δημοφιλέστερων γλωσσών Προγραμματισμού (TIOBE index, n.d.) και θεωρείται ένα εξαιρετικό εργαλείο που συνδέει ιδανικά το χώρο της εκπαίδευσης και τον επαγγελματικό χώρο καθώς συνοδεύεται και από πληθώρα βιβλιοθηκών που επιτρέπουν την αποδοτική ανάπτυξη εφαρμογών σε ποικίλες περιοχές (πχ. Radenski, 2006).

Τεχνικά χαρακτηριστικά της γλώσσας

Αντικειμενοστρέφεια και δυναμική διαχείριση τύπων

Η Python είναι εκ θεμελίων της αντικειμενοστρεφής γλώσσα. Κάθε τιμή (literal) που εμφανίζεται στον κώδικα είναι ένα προγραμματιστικό αντικείμενο που έχει κληρονομήσει ιδιότητες και μεθόδους της κλάσης του. Π.χ. στις εντολές x=1 και y="spam" οι τιμές 1 και "spam" είναι αντικείμενα που κληρονομούν ιδιότητες και μεθόδους από τις κλάσεις int & str αντίστοιχα. Αυτό προσφέρει τη βάση για τη δυναμική διαχείριση τύπων καθώς τα χαρακτηριστικά του τύπου αφορούν τις τιμές και όχι τους αναγνωριστές. Οι διάφοροι αναγνωριστές απλά «συνδέονται» με κάποιο αντικείμενο-τιμή όταν εκτελείται μια εντολή ανάθεσης. Μάλιστα πολλαπλοί αναγνωριστές μπορούν να αναφέρονται στο ίδιο αντικείμενο-τιμή (shared object reference).

Ήδη, αυτό το στοιχείο της γλώσσας δημιουργεί ερωτηματικά στην εκπαίδευση καθώς η έννοια της μεταβλητής δεν υπάρχει στην Python με τον τρόπο που συνήθως εισάγεται αλλού, δηλαδή σαν ένα όνομα που αφορά μια θέση μνήμης (περιέχον), η οποία θέση περιέχει κάποιο δεδομένο (περιεχόμενο) με τιμή που μεταβάλλεται αλλά διαθέτει συγκεκριμένο και αμετάβλητο τύπο (στατική διαχείριση μέσω δήλωσης τύπου). Στις γλώσσες δυναμικού τύπου η έννοια της «μεταβλητής» παραπέμπει περισσότερο σε ένα όνομα-ετικέτα που συνδέεται (μέσω δείκτη) με κάποιο αντικείμενο στη μνήμη, ενώ αυτή η σύνδεση μπορεί να αλλάζει δυναμικά (κάτι όπως τα αυτοκόλλητα-"sticker" που τα κολλάμε όποτε θέλουμε σε όποιο φυσικό αντικείμενο θέλουμε). Πώς θα χειριστεί ο εκπαιδευτικός το ζήτημα αυτό; Θα εμβαθύνει και θα εξηγήσει αυτή τη μορφή υπολογιστικής αναπαράστασης στους μαθητές ή θα το αποφύγει για να μην προκληθεί σύγχυση, ιδιαίτερα αν οι μαθητές έχουν προαναπαραστάσεις που συνάδουν με τη μεταφορά περιέχον-περιεχόμενο;

Βασικές δομές: Έλεγχος και επανάληψη

Οι βασικές δομές δομημένου Προγραμματισμού στην Python περιλαμβάνουν την εντολή ελέγχου if..elif..else και τις δομές επανάληψης while..else και for..range() (ή σωστότερα for..<iterable>). Ενδιαφέροντα χαρακτηριστικά εδώ είναι: α) η χρήση ενός προαιρετικού κλάδου else, β) η λειτουργία της συνάρτησης range() η οποία μπορεί να δημιουργήσει μαθησιακές δυσκολίες (Georgatos, 2002), και γ) η λειτουργία του βρόχου for με οποιαδήποτε απαριθμήσιμη δομή (iterable) στη θέση της range. Τεχνικά μιλώντας, πρόκειται για τύπο δεδομένου που μέσω της μεθόδου __next__ επιστρέφει τιμές στον δείκτη («μεταβλητή») επανάληψης του βρόχου. Η λειτουργία αυτή επιτρέπει στον προγραμματιστή να συνδέει ελεύθερα τον δείκτη και με άλλες τιμές μέσα στο βρόχο. Νέα ερωτήματα επομένως για τον εκπαιδευτικό: πώς θα εισάγει τους μαθητές σ' αυτές τις ιδιαιτερότητες της γλώσσας; Θα «αποσιωπήσει» ίσως κάποιες δυνατότητες που έρχονται σε αντίθεση με τις τεχνικές Προγραμματισμού σε γλώσσες στατικού τύπου;

Δομές δεδομένων

Η βασική και πλέον ευέλικτη δομή δεδομένων στην Python είναι η λίστα (list) με χρήση της οποίας μπορούν να υλοποιηθούν και πίνακες (καθώς δεν υπάρχει άμεσα διαθέσιμη δομή της κλασικής μορφής "array"). Άλλες δομές όπως το λεξικό (dictionary) (που αποτελεί έναν

πίνακα κερματισμού), η πλειάδα ή ομάδα (tuple) (ουσιαστικά μια αμετάλλακτη λίστα) και το σύνολο (set) (δομή με ιδιότητες παρόμοιες με τα μαθηματικά σύνολα) επεκτείνουν ισχυρά τις δυνατότητες αναπαράστασης της γλώσσας.
Ενδιαφέροντα σημεία προβληματισμού για την εκπαίδευση: α) η δημιουργία της λίστας και β) η «μεταλλαξιμότητα» (mutability) μιας δομής. Το πρώτο είναι τεχνικά αποδοτικότερο να γίνεται με «περιγραφή λίστας» (list comprehension) κάτι που όμως είναι πιο δυσνόητο συντακτικά. Εναλλακτικά, μπορεί να γίνει με χρήση της μεθόδου append(), όπου όμως πάλι χρειάζονται πρωθύστερες εξηγήσεις για την έννοια της μεθόδου και την τεχνική της σημειογραφίας τελείας (dot notation). Όσον αφορά το δεύτερο, η μεταλλαξιμότητα μιας δομής μπορεί να δημιουργήσει παρανοήσεις στον αρχάριο, ειδικά στο θέμα κλήσης συναρτήσεων όπου κάθε αλλαγή τιμής σε μεταλλάξιμη δομή-όρισμα μέσα στη συνάρτηση επηρεάζει την ίδια δομή στη γενική (global) εμβέλεια του κώδικα.

Python και Εκπαίδευση

Γενικά, η Python θεωρείται ως εξαιρετική επιλογή ως πρώτη γλώσσα Προγραμματισμού για λόγους όπως (Guo, 2008):

- Όμορφη, καθαρή σύνταξη, και συνοπτική σε σωστό βαθμό ώστε ούτε να προκαλεί οπτική υπερφόρτωση ούτε σύγχυση ή ελλιπή κατανόηση.
- Διερμηνευόμενη, άρα η εκτέλεση εκκινεί άμεσα και παρέχεται άμεση ανάδραση στον μαθητή (καθώς κάποια σφάλματα εμφανίζονται αργότερα στο χρόνο εκτέλεσης), σε αντίθεση με μεταγλωττιζόμενες (compiled) γλώσσες στατικού τύπου όπου η εκτέλεση δεν εκκινεί αν δεν διορθωθούν όλα τα συντακτικά σφάλματα (κάτι που καθυστερεί και αποθαρρύνει τον μαθητή).
- Αλληλεπιδραστική γραμμή εντολών (command prompt), ώστε να παρέχεται άμεση ανατροφοδότηση κατά την εκτέλεση απλών εντολών κώδικα.
- Δεν υπάρχει απαίτηση για δήλωση μεταβλητών κάτι που απλοποιεί τη συγγραφή και εκτέλεση του κώδικα για τον αρχάριο.

Ο Georgatos (2002) καταγράφει γνώμες κάποιων Ελλήνων εκπαιδευτικών σε πιλοτική μελέτη σε ελληνικά σχολεία (στο πλαίσιο του μαθήματος «Ανάπτυξη εφαρμογών σε προγραμματιστικό περιβάλλον» της Γ΄ τάξης Γενικού Λυκείου). Ως θετικά σημεία αναφέρονται:

- Μικρότερος κώδικας άρα μικρότερη πιθανότητα για λάθη.
- Εύκολη γραφή και γρηγορότερη ανάπτυξη.
- Οι μαθητές περνάνε εύκολα από τον ψευδοκώδικα στον πραγματικά εκτελέσιμο κώδικα.
- Δεν απαιτείται τμήμα δηλώσεων, κάτι που επιτρέπει να εστιάζει κανείς σ΄ αυτό που είναι πιο σημαντικό.
- Οι μαθητές συχνά γράφουν κώδικα που τρέχει σωστά με την πρώτη φορά.
- Προσελκύει το ενδιαφέρον των μαθητών και είναι ενθουσιασμένοι με τη γλώσσα.
- Κατάλληλο για χρήση στην εκπαίδευση, ένα σημαντικό εργαλείο.

Ως σημεία προβληματισμού:

- Διαφορετική σύνταξη (σε σχέση με την ψευδογλώσσα) που όμως οι μαθητές την καταλαβαίνουν εύκολα.
- Ο τρόπος σύνταξης της for..range (δεν το αναφέρουν όμως ως πρόβλημα όλοι οι ερωτηθέντες εκπαιδευτικοί).
- Η σχέση της Python με την ψευδογλώσσα στην οποία εξετάζονται οι μαθητές πανελλαδικά.

Πρώτα συμπεράσματα

Αναφέροντας τα προηγούμενα πολύ βασικά στοιχεία για την Python, ο στόχος είναι να τονιστούν τα εξής: α) η γλώσσα έχει χαρακτηριστικά που την καθιστούν ιδανική για την εισαγωγή αρχαρίων στον Προγραμματισμό, β) η ιδιαίτερη μορφή και σύνταξη της Python σίγουρα δημιουργεί σημεία διδακτικού προβληματισμού για τον εκπαιδευτικό, και γ) η έννοια «υπολογιστική σκέψη» δεν συνδέεται με συγκεκριμένες μορφές αναπαράστασης μιας συγκεκριμένης κατηγορίας εργαλείων Προγραμματισμού, αλλά με υψηλότερου επιπέδου αφαιρέσεις που εισάγουν έναν πρόσθετο προβληματισμό στη διδακτική πρακτική. Για παράδειγμα, το κλασικό ερώτημα της Διδακτικής Προγραμματισμού «πώς να εισάγω τους μαθητές στην έννοια της μεταβλητής;» μετασχηματίζεται σε: «ποια είναι η μορφή της έννοιας της μεταβλητής με την οποία θέλω να εξοικειώσω τους μαθητές μου και πώς θα το πετύχω αυτό;» Έτσι, οι διδακτικές αποφάσεις και στόχοι μάθησης που θέτει ο εκπαιδευτικός περιλαμβάνουν και τη μορφή υπολογιστικής σκέψης και έκφρασης που αποτελεί τη βάση μιας εκπαιδευτικής δραστηριότητας, λαμβάνοντας φυσικά υπόψη τα χαρακτηριστικά και ανάγκες του μαθητικού πληθυσμού στον οποίο απευθύνεται.

Με τον τρόπο αυτό, η υπολογιστική σκέψη προβάλει ως μια «διάχυτη» (ubiquitous) νοητική δεξιότητα της οποίας η γενική στόχευση είναι μεν η υπολογιστική επίλυση του προβλήματος, τα εργαλεία όμως που χρησιμοποιεί (γλώσσες, δομές, αλγόριθμοι, κ.λπ.) βρίσκονται πάντοτε μέσα σε μια κοινωνιο-ιστορική εξελικτική διαδικασία αλληλο-διαμόρφωσης και μετασχηματισμού σε σύνδεση πάντοτε και με πρακτικές ανάγκες. Ο αναγκαίος διδακτικός μετασχηματισμός της Python δημιουργεί το κατάλληλο πλαίσιο για την κατανόηση αυτής της δυναμικής πλευράς της «τέχνης του υπολογίζειν». Προς την κατεύθυνση αυτή παρουσιάζονται στη συνέχεια μια σειρά από διδακτικά μοντέλα, τα οποία διευκρινίζουν προϋποθέσεις, συνθήκες και περιορισμούς εκπαιδευτικής χρήσης της γλώσσας.

Μοντέλα διδασκαλίας

Το μοντέλο «ψευδογλώσσας»

Ο απλούστερος τρόπος να παρουσιαστεί η Python σε αρχάριους μαθητές είναι ως μία λειτουργική (δηλ. εκτελέσιμη) μορφή ψευδογλώσσας. Έτσι, μια απλή τεχνική είναι να καθοδηγηθεί ο μαθητής να συλλάβει το πέρασμα από την απλή ψευδογλώσσα στα Ελληνικά στην «ψευδογλώσσα» Python στα Αγγλικά. Την προσέγγιση αυτή μπορεί να τη δει κανείς να εφαρμόζεται στο βιβλίο μαθητή της Γ΄ τάξης του Επαγγελματικού Λυκείου Γενικής Παιδείας (Κωτσάκης κ.ά., 2015). Καθώς μάλιστα η έκδοση 3.x είναι απολύτως συμβατή με κωδικοποίηση Unicode μπορεί κανείς να αναπτύξει μια βιβλιοθήκη με συναρτήσεις και μεθόδους με ονόματα στα Ελληνικά, ώστε ο μαθητής να γράφει π.χ. τύπωσε(x) (αντί print(x)) και ο κώδικάς του να εκτελείται.

Εν τούτοις το μοντέλο αυτό «καταρρέει» πολύ γρήγορα αφού πολύ νωρίς εμφανίζονται οι συντακτικές ιδιαιτερότητες της Python που την απομακρύνουν από μια απλή –και ίσως «ιδανική»- ψευδογλώσσα. Μάλιστα η έκδοση 2.x της γλώσσας δημιουργεί προβληματικές καταστάσεις ήδη στις απλές αριθμητικές πράξεις όπου δεν υπάρχει τελεστής διαίρεσης ακεραίων που να επιστρέφει δεκαδικό μέρος. Π.χ. η εντολή «print 3/2» επιστρέφει «1», κάτι που έρχεται σε αντίθεση με τη λογική προσδοκία των μαθητών για «1,5». Προτείνεται η χρήση της έκδοσης 3.x, η οποία δεν έχει τέτοιους περιορισμούς. Συνοψίζοντας, το μοντέλο «ψευδογλώσσας» μπορεί να εφαρμοστεί ως εισαγωγικό μοντέλο αρχαρίων όταν ζητούμενο είναι να αντιληφθούν ορισμένες πολύ γενικές έννοιες του υπολογίζειν με χρήση απλουστευτικής έκφρασης. Η χρήση της Python σ΄ αυτή την περίπτωση δεν στοχεύει στην

εκμάθηση της γλώσσας καθ' εαυτής αλλά εισάγεται ως λειτουργική ψευδογλώσσα που δίνει άμεση ανάδραση στις προγραμματιστικές ενέργειες του μαθητή.

Το «απλουστευτικό» μοντέλο

Το μοντέλο αυτό προτείνει ότι σε σημεία όπου η Python προκαλεί διδακτικά προβλήματα μπορεί ο εκπαιδευτικός να «απλοποιεί» τη διδασκαλία παίρνοντας κατάλληλες διδακτικές αποφάσεις, ώστε να παραμείνει κοντά στις επιθυμητές μορφές έκφρασης υπολογιστικής σκέψης.

Στην πράξη η ιδέα αυτή έχει δύο εκδοχές: «απόκρυψη» και «επέκταση». Η εκδοχή της «απόκρυψης» προτείνει ότι η διδασκαλία αποκρύπτει θέματα που θα απαιτούσαν εμβάθυνση στην ιδιαίτερη αναπαράσταση της Python και επιλέγει παρουσίαση συμβατή με τις ήδη υπάρχουσες κατανοήσεις των μαθητών ή την κατανόηση που επιδιώκει ο εκπαιδευτικός. Έτσι, η έννοια της μεταβλητής μπορεί να παρουσιαστεί χωρίς εξηγήσεις για την αναφορά σε αντικείμενα και με τρόπο που να συνάδει με τις μαθηματικές προαναπαραστάσεις των μαθητών. Παρόμοια, η αντιμετάθεση τιμών δύο μεταβλητών μπορεί να διδαχθεί με την τεχνική της ενδιάμεσης μεταβλητής, χωρίς χρήση αναπαραστάσεων που ίσως θυμίζουν «μαγικές» μεθόδους στους μαθητές.

Η εκδοχή της «επέκτασης» της γλώσσας προτείνει τη χρήση κατάλληλων συναρτήσεων (διαθέσιμων μέσω εξωτερικής βιβλιοθήκης) που προσφέρουν συντακτικά απλούστερο κώδικα στα σημεία της Python με ιδιαίτερη σύνταξη. Ένα τέτοιο παράδειγμα δίνει και ο Georgatos (2002) με τη συνάρτηση series() για τη δημιουργία λίστας, αποφεύγοντας την χρήση range, append, κ.λπ. Πάντως δεν υπάρχουν βιβλιογραφικές αναφορές για την αποδοτικότητα τέτοιων λύσεων στην εκπαίδευση.

Συνολικά, το απλουστευτικό μοντέλο -σε συνδυασμό με το μοντέλο ψευδογλώσσας- προτείνεται ως κατάλληλο για εισαγωγικά μαθήματα γενικών δεξιοτήτων Προγραμματισμού και για μαθητικούς πληθυσμούς, όπου η εμβάθυνση στις ιδιαιτερότητες της Python αποφεύγεται παρά επιδιώκεται.

Το «εργαλειακό» μοντέλο

Το εργαλειακό μοντέλο προτείνει την εκμάθηση της γλώσσας ως εργαλείου επίλυσης διαφόρων μορφών προβλημάτων με υπολογιστικό τρόπο. Ένα τέτοιο εργαλείο ενδιαφέρει κάθε επιστημονικό κλάδο όχι μόνον των θετικών (φυσικοί, μηχανικοί, κλπ.) αλλά και των ανθρωπιστικών επιστημών (γλωσσολόγοι, κοινωνιολόγοι, κ.λπ.). Το εργαλειακό μοντέλο προτείνει μια τυπική πορεία παρουσίαση της Python, η οποία σε γενικές γραμμές μπορεί να είναι: βασικοί τύποι δεδομένων και μοντέλο εκτέλεσης κώδικα, απλές εντολές εισόδου-εξόδου, βασικές δομές δομημένου Προγραμματισμού, συναρτήσεις, σύνδεση με εξωτερικές βιβλιοθήκες (π.χ. pygame, numpy, scipy, pandas), χωρίς αντικειμενοστρεφή Προγραμματισμό οπωσδήποτε. Το τελευταίο (σύνδεση με βιβλιοθήκες) είναι σημαντικό στοιχείο του εργαλειακού μοντέλου διδασκαλίας καθώς μέσω τέτοιων συνδέσεων η γλώσσα μετατρέπεται σε εξαιρετικά ισχυρό υπολογιστικό εργαλείο. Ακολουθώντας το μοντέλο αυτό ο εκπαιδευτικός παρουσιάζει την Python ως συγκεκριμένη έκφραση της υπολογιστικής σκέψης, με τις ιδιαίτερες αναπαραστάσεις, δομές και λειτουργίες της, εστιάζοντας τελικά στο να βοηθήσει τους εκπαιδευόμενους να χρησιμοποιήσουν το πακέτο Python + βιβλιοθήκες για επίλυση υπολογιστικών προβλημάτων. Προτείνεται για ακροατήρια όπως κλάδοι ειδίκευσης στην Επαγγελματική Δευτεροβάθμια, μετα-Δευτεροβάθμια εκπαίδευση, σεμινάρια ειδικών επιστημονικών κλάδων (πχ. Python για μηχανικούς, για τις κοινωνικές επιστήμες, κ.λπ.).

Το «επεξηγηματικό» μοντέλο

Πρόκειται για ένα μοντέλο εμβάθυνσης και επεξήγησης των ιδιαίτερων υπολογιστικών αναπαραστάσεων και λειτουργιών της Python σε σχέση με άλλες γλώσσες (ειδικά εκείνες στατικής διαχείρισης τύπων με τις οποίες είναι εξοικειωμένοι όλοι οι προγραμματιστές). Πρόκειται για μοντέλο κατάλληλο για ειδικευμένα ακροατήρια με ενδιαφέρον για τα χαρακτηριστικά της γλώσσας (πχ. πανεπιστημιακά μαθήματα σε τμήματα Πληροφορικής, σεμινάρια προγραμματιστών, κ.ά.). Το ειδικό χαρακτηριστικό του μοντέλου είναι ότι εστιάζει σε σημεία που δίνουν την ευκαιρία για εμβάθυνση όχι απλά στον τρόπο λειτουργίας της γλώσσας αλλά και σε τεχνικές υπολογιστικής αναπαράστασης σε σύγκριση με άλλες γλώσσες Προγραμματισμού. Τέτοια σημεία –ενδεικτικά- μπορεί να είναι:

- Η διαμοιρασμένη αναφορά αντικειμένου (shared reference)
- Η έννοια της μεταλλάξιμης δομής (mutability)
- Η λειτουργία της range() και γενικότερα των συναρτήσεων-γεννητόρων (generators) και της yield (σε σύγκριση με τη return)
- Η λειτουργία των συναρτήσεων-διακοσμητών (decorators)
- Κλάσεις-αντικείμενα και ιδιαίτερες λειτουργίες τους (όπως π.χ. οι μέθοδοι κλάσης και αντικειμένων-στιγμιότυπων, οι στατικές μέθοδοι, κ.λπ.)

Συνολικά το επεξηγηματικό μοντέλο προωθεί ένα βαθύτερο -περισσότερο αφηρημένο- επίπεδο υπολογιστικής σκέψης και σε συνδυασμό με το προηγούμενο (εργαλειακό μοντέλο) αποτελεί ένα ισχυρό τρόπο εκμάθησης και αξιοποίησης της γλώσσας για ποικίλα ακροατήρια σε προχωρημένο επίπεδο.

Το «αντικειμενοστρεφές» (ή «πρώτα-αντικείμενα») μοντέλο

Πρόκειται για το γνωστό "object first" (ή "OO first") μοντέλο το οποίο προτείνει την εισαγωγή των μαθητών στις ιδέες και δομές του αντικειμενοστρεφούς Προγραμματισμού (Object Oriented Programming, OOP) νωρίς στην πορεία της διδασκαλίας. Σε ένα τέτοιο μοντέλο μετά την πρώτη ενότητα που παρουσιάζει τις πολύ βασικές έννοιες (συνήθως απλοί τύποι δεδομένων, εκφράσεις, μεταβλητές και ανάθεση), η πορεία εκπαίδευσης στρέφεται στις κλάσεις, αντικείμενα και μεθόδους (Gries, 2008). Η διδασκαλία των βασικών δομών Προγραμματισμού και δεδομένων έρχεται αργότερα και γίνεται στο πλαίσιο διαχείρισης κλάσεων και αντικειμένων. Εντούτοις το «αντικειμενοστρεφές» μοντέλο θεωρείται γενικά δύσκολο τόσο για τον εκπαιδευτικό όσο και για τους μαθητές (Radinsky, 2006) και δεν ενθαρρύνεται η υιοθέτησή του τουλάχιστον για το επίπεδο της Δευτεροβάθμιας Εκπαίδευσης.

Η Python είναι μια πλήρως αντικειμενοστρεφής γλώσσα, η οποία όμως δεν υποχρεώνει ούτε τον προγραμματιστή να υιοθετήσει οπωσδήποτε τεχνικές αντικειμενοστρεφούς Προγραμματισμού, ούτε και τον εκπαιδευτικό να διδάξει αμέσως κλάσεις και αντικείμενα. Έτσι, ορισμένοι ερευνητές (π.χ. Radinsky, 2006) προτείνουν ότι με την Python μπορεί να εφαρμοστεί ιδανικά μια ενδιάμεση μορφή διδασκαλίας, η οποία να συνδυάζει την κλασική «πρώτα-εντολές» (statements-first) προσέγγιση με την «πρώτα-αντικείμενα» (object-first) προσέγγιση. Σύμφωνα με την ενδιάμεση τεχνική παρουσιάζεται στους μαθητές ευθύς εξ αρχής η δομή της συνάρτησης και μέσω των συναρτήσεων οι εντολές ελέγχου και επανάληψης, έτσι ώστε οι μαθητές να γράφουν καλά δομημένο κώδικα και να προετοιμαστούν για τη συνέχεια, όπου θα παρουσιαστούν κλάσεις και αντικείμενα.

Το «μεταπρογραμματιστικό» μοντέλο

Ο όρος «μεταπρογραμματισμός» (metapogramming) αναφέρεται γενικά σε τεχνικές Προγραμματισμού που στοχεύουν στο να μεταβάλλουν τον ίδιο τον κώδικα αντιμετωπίζοντάς τον ως δεδομένα. Το μεταπρογραμματιστικό μοντέλο, προφανώς, δεν αφορά εισαγωγικά επίπεδα εκμάθησης της γλώσσας. Πρόκειται για μοντέλο που εμβαθύνει στις τεχνικές μεταπρογραμματισμού της γλώσσας και προϋποθέτει την σε βάθος κατανόηση του μοντέλου κλάσεων-αντικειμένων της Python. Μπορεί να συνδυαστεί με το επεξηγηματικό και το αντικειμενοστρεφές μοντέλο για να εισάγει τους προγραμματιστές (ο όρος «μαθητής» είναι μάλλον ακατάλληλος) σε κατανοήσεις που αγγίζουν τα θεμέλια οικοδόμησης της Python καθώς και θέματα θεωρίας γλωσσών Προγραμματισμού.

Τελικά συμπεράσματα

Η Python αποτελεί σήμερα ένα εξαιρετικό εργαλείο τόσο για την εκπαίδευση όσο και για τον επαγγελματία προγραμματιστή. Οι διαφορές της σε σχέση με τις γλώσσες στατικού τύπου δίνουν το έναυσμα για εμβάθυνση σε θέματα υπολογιστικής σκέψης και ταυτόχρονα δημιουργικού προβληματισμού για τον τρόπο αξιοποίησής της στη διδασκαλία ανάλογα με τις ανάγκες και τα χαρακτηριστικά του μαθητικού πληθυσμού. Προς την κατεύθυνση αυτή το άρθρο αυτό πρότεινε μια σειρά διδακτικών μοντέλων της γλώσσας ως πρακτικό εργαλείο για τον διδακτικό μετασχηματισμό της γλώσσας. Οι εκπαιδευτικοί Πληροφορικής μπορούν να επιλέγουν ή συνδυάζουν τα μοντέλα αυτά (ή φυσικά να τα επεκτείνουν, ώστε να παράγουν τα δικά τους) στο πλαίσιο αποφάσεων που αφορούν και διαμορφώνουν τη μορφή υπολογιστικής έκφρασης και τεχνικών Προγραμματισμού που πρέπει να διδαχθούν σε κάθε περίσταση.

Αναφορές

Georgatos, F. (2002). How applicable is Python as first computer language for teaching programming in a pre-university educational environment, from a teacher's point of view? *Master Thesis*, AMSTEL Institute, Faculty of Science, Universiteit of Amsterdam.

Gries, D. (2008). A Principled Approach to Teaching OO First. In *Proceedings of the 39th SIGCSE technical symposium on Computer science education (SIGCSE '08)*, (pp. 31-35), ACM.

Guo, P. (2008, July 2). *Why Python is a great language for teaching beginners in introductory programming classes*. Retrieved from http://pgbovine.net/python-teaching.htm.

History of Python (n.d.) Wikipedia, Retrieved September 1 2015 from https://en.wikipedia.org/wiki/History_of_Python.

Operational Definition of Computational Thinking (2011). International Society for Technology in Education (ISTE) and the Computer Science Teachers Association, (CSTA). Retrieved September 1 2015 from http://www.csta.acm.org/Curriculum/sub/CurrFiles/CompThinkingFlyer.pdf.

Papert, S. (1996). An exploration in the space of mathematics educations. *International Journal of Computers for Mathematical Learning 1*, doi:10.1007/BF00191473.

Radenski, A. (2006). "Python First": A Lab-Based Digital Introduction to Computer Science. In *Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education (ITICSE '06)*, (pp. 197-201).

TIOBE index (n.d.) *TIOBE programming community index*. Retrieved from http://www.tiobe.com/tpci.htm.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*, 33–35. Retrieved September 1 2015 from http://dx.doi.org/10.1145/1118178.1118215.

Κωτσάκης, Σ., Μακρυγιάννης, Η., Παραδείση, Α., και Ταταράκη, Α. (2015). *Εισαγωγή στις αρχές της επιστήμης των ηλεκτρονικών υπολογιστών. Σημειώσεις μαθητή*. Ινστιτούτο Τεχνολογίας Υπολογιστών & Εκδόσεων «Διόφαντος».