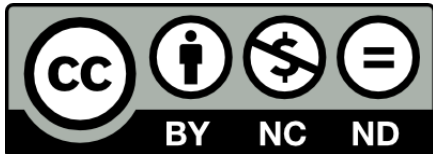


Εισαγωγή στην **Python**

2



Copyright

Το παρόν εκπαιδευτικό υλικό προσφέρεται ελεύθερα υπό τους όρους της άδειας Creative Commons:

- Αναφορά Δημιουργού - Μη Εμπορική Χρήση - Όχι Παράγωγα Έργα 3.0.

Για να δείτε ένα αντίγραφο της άδειας αυτής επισκεφτείτε τον ιστότοπο

<https://creativecommons.org/licenses/by-nc-nd/3.0/gr/>

Στ. Δημητριάδης, 2015

Περιεχόμενα

- Μοντέλο εκτέλεσης κώδικα Python

Μοντέλο εκτέλεσης κώδικα Python

Python code execution model

Διερμηνέας - Interpreter

- Τυπικά η Python είναι γλώσσα **διερμηνευόμενη** (interpreted)
- Επομένως έρχεται στη μορφή ενός πακέτου λογισμικού που ονομάζεται “**διερμηνέας**” (Interpreter).
- Όταν γράφετε ένα πρόγραμμα σε Python ο πηγαίος κώδικας αποθηκεύεται σε αρχείο με επέκταση **.py**
 - Ο διερμηνέας διαβάζει τον κώδικά σας και εκτελεί **άμεσα** τις εντολές μία προς μία
 - Τελικά δηλ. ο διερμηνέας είναι το λογισμικό που **περικλείει την απαραίτητη λογική της Python** ώστε να εκτελέσει τον κώδικα που γράψατε.
- Ανάλογα ποια έκδοση (flavor) της Python χρησιμοποιείτε ο διερμηνέας μπορεί να είναι γραμμένος σε C (**CPython**), ή να περιλαμβάνει κλάσεις Java (**Jython**), ή σε περιβάλλον .NET (**IronPython**), κλπ.

Byte Code compilation

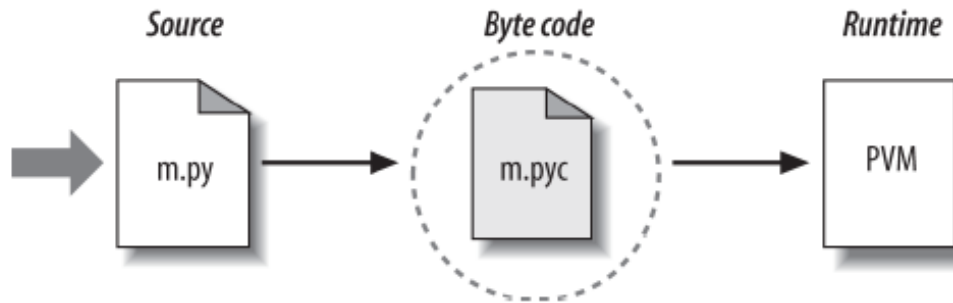
- Όμως η εργασία που κάνει ο διερμηνέας δεν είναι η απλή εκτέλεση εντολών
- Αρχικά ο κώδικας μεταφράζεται (compiled) σε μορφή “**byte code**” και στη συνέχεια εκτελείται από μία κατάλληλη “**εικονική μηχανή**” (“virtual machine”).
- **Byte code compilation**
- Ο κώδικας byte (byte code) είναι μια χαμηλότερου επιπέδου μεταφρασμένη μορφή κώδικα σε μορφή **bytes** και **ανεξάρτητη** τεχνολογικής πλατφόρμας (platform-independent)
- Αυτή η μετάφραση γίνεται ώστε τελικά να αυξηθεί η ταχύτητα εκτέλεσης της εφαρμογής σας
 - Ο κώδικας byte code εκτελείται πολύ γρηγορότερα από τον πηγαίο σας κώδικα

Αρχεία .pyc

- Από την έκδοση 3.2 και μετά η Python αποθηκεύει τον μεταφρασμένο byte code σε:
- **Αρχείο** με επέκταση **.pyc** και μέσα σε **φάκελο** με το όνομα **__pycache__**
 - Μέσα στο directory όπου βρίσκονται τα αρχεία **.py** με τον πηγαίο σας κώδικα
 - Η αποθήκευση σε *.pyc γίνεται για τα αρχεία πηγαίου κώδικα που "εισάγονται" προς εκτέλεση (imported)
- Επιπρόσθετα στο όνομα του αρχείου .pyc προστίθεται και ένα χαρακτηριστικό στοιχείο που δηλώνει την **ειδική έκδοση** της Python που χρησιμοποιείτε, πχ. **script.cpython-33.pyc**
 - Έτσι αποφεύγεται η διαγραφή αρχείων byte code από διαφορετικές εκδόσεις Python που πιθανώς έχετε στον υπολογιστή σας

Εικονική Μηχανή Python

- Από τη στιγμή που το πρόγραμμά σας μεταφράστηκε σε byte code (ή φορτώσατε byte code από κάποιο.pyc αρχείο), ο κώδικας προωθείται για εκτέλεση από την **Εικονική Μηχανή Python** (Python Virtual Machine ή PVM)



Πηγή: Lutz, M. (2013). *Learning Python, 5th ed.*, O'Reilly: Cambridge

- Μοντέλο εκτέλεσης κώδικα της Python**
- (α) Ο πηγαίος κώδικας **μεταφράζεται** (compiled) σε **ενδιάμεσο κώδικα** byte (byte code σε αρχείο .pyc)
- (β) Ο ενδιάμεσος κώδικας byte **διερμηνεύεται** (interpreted) στη συνέχεια από την εικονική μηχανή Python Virtual Machine

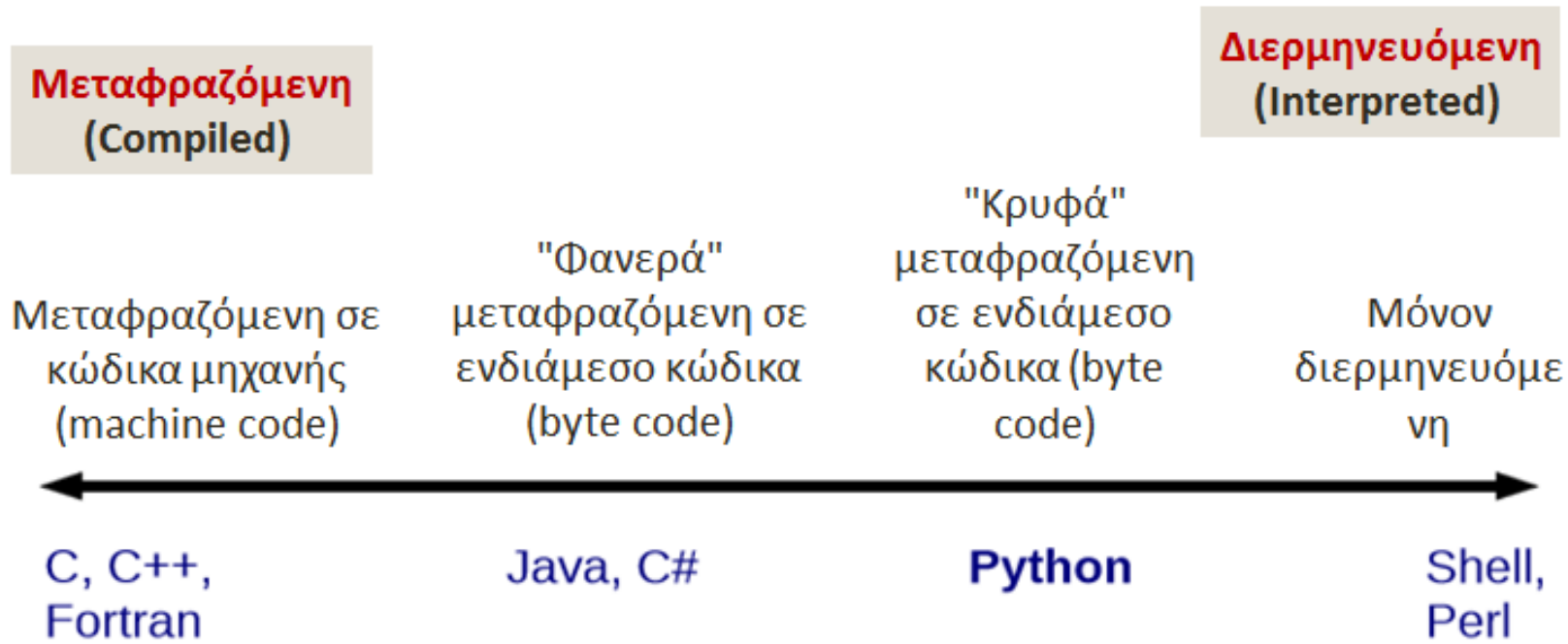
Τι σημαίνει αυτό για την ταχύτητα εκτέλεσης του κώδικα;

- (α) Ο κώδικας εκτελείται **αμέσως** μόλις τον γράψετε.
 - Δεν υπάρχει κάποια χρονοβόρα φάση όπως πχ. build, make, link κλπ.
- (β) Ο byte code **ΔΕΝ** είναι δυαδικός κώδικας μηχανής (binary machine code) αλλά είναι μια ενδιάμεση αναπαράσταση που δημιουργεί η Python
 - Κατά την εκτέλεση η PVM πρέπει να ερμηνεύσει τον bytecode
 - Έτσι μπορεί τελικά το πρόγραμμα σε Python να εκτελείται πιο αργά σε σύγκριση με το αντίστοιχο γραμμένο σε άμεσα μεταφραζόμενη γλώσσα όπως η C
- (γ) Σε σχέση όμως με τον κλασικό διερμηνέα κώδικα, η Python **ΔΕΝ** χρειάζεται να αναλύσει και να ερμηνεύσει πάλι τον κώδικα καθώς υπάρχει ήδη η μετάφραση στο αρχείο .pyc
- (δ) Το τελικό αποτέλεσμα είναι πως ο κώδικας Python μπορεί να εκτελείται με **ταχύτητες μεταξύ** της ταχύτητας μιας καθαρά μεταφραζόμενης (compiled) γλώσσας και μιας καθαρά διερμηνευόμενης (interpreted) γλώσσας

...συνέχεια...

- Στην Python, ο μεταφραστής (compiler) είναι πάντοτε **παρών** κατά την εκτέλεση (runtime) και αποτελεί μέρος του συστήματος που εκτελεί τον κώδικα.
- Αυτό επιτρέπει έναν πολύ **γρήγορο κύκλο ανάπτυξης** (rapid development cycle).
 - Δεν υπάρχει ανάγκη για μετάφραση και σύνδεση (precompile and link) πριν την εκτέλεση του κώδικα
- Αυτό, με τη σειρά του δίνει έναν **δυναμικό χαρακτήρα στη γλώσσα** – συχνά είναι πολύ πρακτικό να διαμορφώνεται η μορφή του κώδικα Python την ώρα της εκτέλεσης
- Κάτι που επιτρέπει την **ευέλικτη διαμόρφωση του προϊόντος λογισμικού** (customization)
 - — ακριβώς επειδή ο κώδικας Python αλλάζει την ώρα της εκτέλεσης (on the fly), οι προγραμματιστές μπορούν να αλλάζουν στοιχεία του κώδικα ανάλογα με τις ανάγκες που προκύπτουν χωρίς να χρειάζονται προηγουμένως μετάφραση και σύνδεση κλπ.

Πού βρίσκεται η Python στον άξονα compiled – interpreted γλωσσών;



Ποια εκδοχή της Python «τρέχω»;

- Η **CPython** είναι η τυπική υλοποίηση (standard implementation) της γλώσσας και είναι αυτή που χρησιμοποιούν οι περισσότεροι χρήστες
 - ειδικά οι αρχάριοι για την εκμάθησή της
- Όλες οι άλλες υλοποιήσεις της Python έχουν αναπτυχθεί με ειδικό στόχο και ρόλο, παρόλο που φυσικά μπορούν να χρησιμοποιηθούν όπως και η CPython
- Όλες οι εκδοχές της γλώσσας "διαβάζουν" την ίδια μορφή πηγαίου κώδικα αλλά μεταφράζουν και εκτελούν τον κώδικα σε διαφορετικό περιβάλλον
 - Πχ. η JPython σε περιβάλλον Java, η IronPython σε περιβάλλον .NET (C#) κλπ.