Κεφάλαιο 1 - Εγκατάσταση της Python

Βεβαιωθείτε ότι μπορείτε να εκτελέσετε την Python από τη γραμμή εντολών (`cmd.exe`). Όλα τα παρακάτω που ακολουθούν αφορούν εντολές στη γραμμή εντολών:

Πριν προχωρήσετε περαιτέρω, βεβαιωθείτε ότι έχετε την Python και ότι η αναμενόμενη έκδοση είναι διαθέσιμη από τη γραμμή εντολών σας. Μπορείτε να το ελέγξετε εκτελώντας:

```
py --version
```

Βεβαιωθείτε ότι μπορείτε να εκτελέσετε το `pip` από τη γραμμή εντολών.

```
py -m pip --version
```

Εάν εγκαταστήσατε την Python από το `python.org`, θα πρέπει να έχετε ήδη το `pip`. Εάν το `pip` δεν είναι ήδη εγκατεστημένο, τότε προσπαθήστε πρώτα να το εκκινήσετε από την τυπική βιβλιοθήκη:

```
py -m ensurepip --default-pip
```

Βεβαιωθείτε ότι τα `pip`, `setuptools`, `wheel` είναι ενημερωμένα:

```
py -m pip install --upgrade pip setuptools wheel
```

Χρησιμοποιήστε το `pip` για εγκατάσταση πακέτων. Για να εγκαταστήσετε την πιο πρόσφατη έκδοση ενός πακέτου, π.χ. το πακέτο `SomeProject`, γράφετε

```
py -m pip install SomeProject
```

Για αναβάθμιση πακέτων

```
1 py -m pip install --upgrade SomeProject
```

Πλέον μπορείτε να ξεκινήσετε να γράφετε κώδικα και να τον εκτελείτε με την Python εγκατεστημένη σε ολόκληρο τον υπολογιστή σας. Αυτό σημαίνει ότι η εγκατάσταση της Python είναι καθολική (παγκόσμια) στον υπολογιστή σας και αλληλεπιδρά με όλα τα υπόλοιπα μέρη του συστήματός του.

Σημείωση

Υπάρχει και ένας δεύτερος τρόπος να τρέχετε την Python στον υπολογιστή σας. Μπορείτε να έχετε εγκατεστημένη την Python μέσα σε μια συγκεκριμένη περιοχή του σκληρού σας δίσκου, απομονωμένη από το υπόλοιπο περιβάλλον του υπολογιστή σας, έτσι ώστε οποιεσδήποτε δράσεις κάνετε, όπως λήψη και εφαρμογή πακέτων, χρήση συγκεκριμένης έκδοσης της Python κτλ, να συμβαίνουν τοπικά και να μην έχουν επικοινωνία, αλληλεπίδραση και λειτουργία με άλλες εφαρμογές ή περιοχές μνήμης του υπολογιστή σας. Με άλλα λόγια, μπορείτε να έχετε τοπικά εγκατεστημένη την Python σε ένα συγκεκριμένο περιβάλλον του συστήματός σας και ό,τι κάνετε σε αυτό το περιβάλλον, μένει σε αυτό και δεν επικοινωνεί ούτε επηρεάζεται από τις υπόλοιπες λειτουργίες του υπολογιστή σας.

Τέτοια περιβάλλοντα η Python τα ονομάζει «Εικονικά Περιβάλλοντα» και όπως καταλαβαίνετε, μπορούν να υπάρχουν περισσότερα από ένα : Δηλαδή μπορείτε να έχετε την Python 3.4 με τα πακέτα Numpy, matplotlib σε ένα συγκεκριμένο μπλοκ μνήμης του σκληρού σας δίσκου (έστω στο A) και σε ένα άλλο μπλοκ (έστω στο B) την Python 3.12 εγκατεστημένη μαζί με τα πακέτα scipy, matplotlib, SymPy. Ανάλογα με τις ανάγκες του έργου σας, άλλοτε μπορείτε να εργάζεστε στο περιβάλλον A και άλλοτε στο περιβάλλον B χωρίς ποτέ αυτά τα δύο να έρθουν σε επαφή, αλληλεπίδραση ή διένεξη μεταξύ τους. Είναι δύο κόσμοι που συνυπάρχουν μεν, αλλά ο ένας δεν έχει αντίληψη του άλλου δε.

Για τη δημιουργία τέτοιων περιβαλλόντων, δείτε την ενότητα «Εικονικά Περιβάλλοντα», αλλιώς δεν χρειάζεται να την διαβάσετε.

1.1 Χρήσιμη ορολογία

Στο σημείο αυτό θα θέλαμε να παρουσιάσουμε κάποιους όρους και λέξεις τους οποίους και τις οποίες θα χρησιμοποιούμε συχνά είτε στα Αγγλικά είτε στα Ελληνικά:

- **Python module**: Το module ή ενότητα ή μονάδα είναι ένα αρχείο που περιέχει ορισμούς και δηλώσεις Python. Μια ενότητα μπορεί να ορίσει συναρτήσεις, κλάσεις και μεταβλητές. Μια ενότητα μπορεί επίσης να περιλαμβάνει κώδικα με δυνατότητα εκτέλεσης. Η ομαδοποίηση σχετικού κώδικα σε μια ενότητα καθιστά τον κώδικα πιο κατανοητό και πιο εύκολο στη χρήση. Κάνει επίσης τον κώδικα λογικά οργανωμένο. Με άλλα λόγια, η ενότητα είναι ένα αρχείο με κώδικα Python το οποίο το έχουμε γράψει εμείς ή άλλοι άνθρωποι και το χρησιμοποιούμε στον δικό μας κώδικα, για να μην χρειαστεί να γράψουμε όλο τον κώδικα που περιέχει η ενότητα από μόνοι/ες μας από την αρχή.

- **Package**: Το package ή πακέτο ή Βιβλιοθήκη συνήθως αποτελείται από πολλές ενότητες. Φυσικά, ένα πακέτο είναι ένας φάκελος που περιέχει λειτουργικές μονάδες και ίσως άλλους φακέλους που μπορεί να περιέχουν περισσότερους φακέλους και λειτουργικές μονάδες. Εννοιολογικά, είναι ένας χώρος ονομάτων. Αυτό σημαίνει απλώς ότι οι ενότητες ενός πακέτου συνδέονται μεταξύ τους με ένα όνομα πακέτου, με το οποίο μπορούν να αναφέρονται. Με άλλα λόγια, πακέτο είναι ένα σύνολο από ενότητες το οποίο το εισάγουμε στον κώδικά μας για να κάνουμε χρήση των λειτουργιών που εκτελούν αυτές. Για παράδειγμα, θα δούμε στις ενότητες που ακολουθούν ότι εισάγοντας το πακέτο Numpy στον κώδικά μας, μπορούμε πλέον να κάνουμε πράξεις με πίνακες ή εισάγοντας το πακέτο Matplotlib μπορούμε να σχεδιάσουμε διάφορα γραφήματα, καμπύλες, 3D σχήματα κτλ.

1.2 Εικονικά Περιβάλλοντα

Τα «Εικονικά Περιβάλλοντα» της Python επιτρέπουν στα πακέτα της Python να εγκατασταθούν σε μια απομονωμένη τοποθεσία για μια συγκεκριμένη εφαρμογή, αντί να εγκατασταθούν παγκοσμίως (σε ολόκληρο τον υπολογιστή). Αυτά έχουν τους δικούς τους καταλόγους εγκατάστασης και δεν μοιράζονται βιβλιοθήκες με άλλα εικονικά περιβάλλοντα.

Επί του παρόντος, υπάρχουν δύο κοινά εργαλεία για τη δημιουργία εικονικών περιβαλλόντων στην Python: Το `venv` είναι διαθέσιμο από προεπιλογή στην Python 3.3 και σε νεότερες εκδόσεις και εγκαθιστά `pip`, `setuptools` σε δημιουργημένα εικονικά περιβάλλοντα στην Python 3.4 και μεταγενέστερα.

Το `virtualenv` πρέπει να εγκατασταθεί ξεχωριστά, αλλά υποστηρίζει Python 2.7+ και Python 3.3+ και τα `pip`, `setuptools`, `wheels` εγκαθίστανται πάντα σε δημιουργημένα εικονικά περιβάλλοντα από προεπιλογή (ανεξάρτητα από την έκδοση της Python). Η βασική χρήση έχει ως εξής:

Χρησιμοποιώντας το venv

Η ενότητα `venv` υποστηρίζει τη δημιουργία ελαφρών «εικονικών περιβαλλόντων», το καθένα με το δικό του ανεξάρτητο σύνολο πακέτων Python εγκατεστημένα στους καταλόγους του ιστότοπού του. Ένα εικονικό περιβάλλον δημιουργείται πάνω από μια υπάρχουσα εγκατάσταση Python, γνωστή ως Python “βάση” του εικονικού περιβάλλοντος, και μπορεί προαιρετικά να απομονωθεί από τα πακέτα στο βασικό περιβάλλον, επομένως μόνο εκείνα που είναι ρητά εγκατεστημένα στο εικονικό περιβάλλον είναι διαθέσιμα.

Όταν χρησιμοποιούνται μέσα από ένα εικονικό περιβάλλον, τα κοινά εργαλεία εγκατάστασης , όπως το `pip` , θα εγκαταστήσουν πακέτα Python σε ένα εικονικό περιβάλλον χωρίς να χρειάζεται να τους ζητηθεί ρητά να το κάνουν. Ένα εικονικό περιβάλλον (μεταξύ άλλων):

- Χρησιμοποιείται για να περιέχει έναν συγκεκριμένο διερμηνέα Python και βιβλιοθήκες λογισμικού και δυαδικά αρχεία που χρειάζονται για την υποστήριξη ενός έργου (βιβλιοθήκης ή εφαρμογής). Αυτά είναι από προεπιλογή απομονωμένα από λογισμικό σε άλλα εικονικά περιβάλλοντα και διερμηνείς Python και βιβλιοθήκες εγκατεστημένες στο λειτουργικό σύστημα.

- Περιέχεται σε έναν κατάλογο, με συμβατικό όνομα είτε `venv` είτε `.venv` στον κατάλογο του έργου, είτε κάτω από έναν κατάλογο κοντέινερ για πολλά εικονικά περιβάλλοντα, όπως `/.virtualenvs`. Δεν έχει ελεγχθεί σε συστήματα ελέγχου πηγής, όπως το Git.

- Θεωρείται ως αναλώσιμο – θα πρέπει να είναι απλό να το διαγρά-

ψετε και να το δημιουργήσετε ξανά από την αρχή. Δεν τοποθετείτε κανέναν κώδικα έργου στο περιβάλλον. Δεν θεωρείται ως κινητό ή αντιγράψιμο – απλώς αναδημιουργείτε το ίδιο περιβάλλον στη θέση προορισμού.

1.2.1 Δημιουργία εικονικών περιβαλλόντων

Η δημιουργία εικονικών περιβαλλόντων γίνεται με την εκτέλεση της εντολής `venv`:

```
py -m venv <DIR>
<DIR>\Scripts \activate
```

Όπου `DIR` ο κατάλογος μέσα στον οποίο θέλετε να δημιουργήσετε το εικονικό περιβάλλον σας. Η εκτέλεση αυτής της εντολής δημιουργεί τον κατάλογο προορισμού (δημιουργώντας γονικούς καταλόγους που δεν υπάρχουν ήδη) και τοποθετεί ένα αρχείο `pyvenv.cfg` σε αυτόν με ένα κλειδί `home` που δείχνει την εγκατάσταση Python από την οποία εκτελέστηκε η εντολή (ένα κοινό όνομα για τον κατάλογο στόχο είναι `.venv`). Δημιουργεί επίσης έναν υποκατάλογο `bin` (ή `Scripts` στα Windows) που περιέχει ένα αντίγραφο/σύνδεσμο του δυαδικού/δυαδικού Python (όπως είναι κατάλληλο για την πλατφόρμα ή τα ορίσματα που χρησιμοποιούνται κατά τη δημιουργία περιβάλλοντος). Δημιουργεί επίσης έναν (αρχικά κενό) υποκατάλογο `lib/pythonX.Y/site-packages` (στα Windows, αυτός είναι το `Lib\packages`). Εάν καθοριστεί ένας υπάρχων κατάλογος, θα χρησιμοποιηθεί ξανά. Στα Windows η `Scripts` περιέχει δύο αρχεία - `activate.bat` και `Activate.ps1` - τα οποία μπορείτε να εκτελέσετε για να ενεργοποιήσετε το εικονικό σας περιβάλλον. Στα Unix-ά συστήματα, το αντίστοιχο αρχείο είναι το `activate`. Μπορείτε να το ενεργοποιήσετε εκτελώντας την εντολή «source» ή «.», καθώς και το «activate».

1.3 Άλλες επισημάνσεις

Η Python είναι γλώσσα η οποία δίνει πάρα πολύ μεγάλη βαρύτητα στις εσοχές των γραμμών του κώδικα. Κάθε γραμμή πρέπει να είναι μετατοπισμένη κατά 4 κενά από την αμέσως προηγούμενή της όταν, και μόνο, όταν οι δύο γραμμές περιέχουν εντολές.

Παράδειγμα

```
# Initialize list `a` with four integers
a = [1, 2, 3, 4]

# Initialize list `b` with four different integers
b = [5, 6, 7, 8]
```

Δεν περιλαμβάνεται καμία εντολή, οπότε το `b` γράφεται στην ίδια κάθετο με το `a`. Όμως:

```
# Iterate over each element in the list `a`
for i in a:
    # Print the current element in the iteration
    print(i)


    # Output :1,2,3,4
```

Εδώ το `for` είναι εντολή, οπότε η αμέσως επόμενη σειρά (που αποτελεί τον κώδικα που θα εκτελεστεί με βάση τη λειτουργία της `for` εντολής) γράφεται κατά 4 κενά δεξιότερα από την κάθετο που βρίσκεται το γράμμα `f` της `for` εντολής.

Τελειώνοντας την εισαγωγή αυτή, θα θέλαμε να αναφέρουμε το PEP 8 (Python Enhancement Proposal) για την καλύτερη συγγραφή κώδικα στην Python. Το PEP σημαίνει Πρόταση Βελτίωσης Python. Ένα PEP είναι ένα έγγραφο σχεδιασμού που παρέχει πληροφορίες στην κοινότητα της Python ή περιγράφει μια νέα δυνατότητα για την Python ή τις διαδικασίες ή το περιβάλλον της. Το PEP θα πρέπει να παρέχει μια συνοπτική τεχνική προδιαγραφή του χαρακτηριζόμενου αντικειμένου και μια λογική για αυτό.