

Τμήμα Πληροφορικής και Τηλεματικής  
Χαροκόπειο Πανεπιστήμιο  
ΥΠ23 Τεχνητή Νοημοσύνη

# Tutorial: Προετοιμασία με την Python

Έκδοση 2021-1.0

Διδάσκων: Χρήστος Δίου

## 1 Εισαγωγή

Στο μάθημα θα χρησιμοποιήσουμε τα `pacman projects` που αναπτύχθηκαν στο Πανεπιστήμιο του Berkeley (<http://ai.berkeley.edu>) ως διδακτικό εργαλείο για τους αλγόριθμους Τεχνητής Νοημοσύνης που θα μάθετε αυτό το εξάμηνο. Οι λύσεις σας θα αναπτυχθούν σε γλώσσα Python.

Οι παρακάτω ενότητες δίνουν συνοπτικές οδηγίες για την εγκατάσταση των απαραίτητων εργαλείων ώστε να λύσετε τις ασκήσεις. Επίσης δίνονται κάποιες ασκήσεις στη γλώσσα Python καθώς και στον `autograder`. Οι ασκήσεις αυτές δε βαθμολογούνται ωστόσο σας προετοιμάζουν για τις επόμενες ασκήσεις του μαθήματος (που θα βαθμολογηθούν).

## 2 Python

### 2.1 Εγκατάσταση περιβάλλοντος Python

Μπορείτε να χρησιμοποιήσετε ένα δικό σας περιβάλλον (`python environment`) και κάποιο IDE όπως το Pycharm (<https://www.jetbrains.com/pycharm/>). Είναι σημαντικό να χρησιμοποιήσετε έκδοση της Python έως 3.6 (διαφορετικά δε θα λειτουργεί ο `autograder`). Παρακάτω δίνονται οδηγίες για την εγκατάσταση σε περιβάλλον UNIX/Linux μέσω Anaconda.

Αρχικά εγκαταστήστε το Anaconda (ή το Miniconda) για Python3 από τον παρακάτω ιστότοπο:  
<https://docs.anaconda.com/anaconda/install/>

Έπειτα φτιάξτε ένα `conda environment` με την έκδοση 3.6 της Python ως εξής

```
conda create --name yp23 python=3.6
```

Μπορείτε να δώσετε ότι όνομα θέλετε στο περιβάλλον σας, εδώ δώσαμε το `yp23`.

Έπειτα, μπορείτε να ενεργοποιήσετε το περιβάλλον με την εντολή

```
source activate yp23
```

Για να απενεργοποιήσετε το περιβάλλον μπορείτε να εκτελέσετε την εντολή

```
source deactivate
```

### 2.2 Εισαγωγή στην Python

Υπάρχουν πάρα πολλές πηγές online για την εκμάθηση της Python, σε περίπτωση που δε γνωρίζετε τη γλώσσα. Η επίσημη τεκμηρίωση και το tutorial είναι αρκετά καλό:

<https://docs.python.org/3/tutorial/>

Επίσης συστήνεται και το υλικό του μαθήματος “Scientific Python” από το Πανεπιστήμιο του Stanford, το οποίο εκτός από βασικά στοιχεία της γλώσσας Python, σας προσφέρει και μία εισαγωγή στις βιβλιοθήκες Numpy, Pandas και scikit-learn.

<http://web.stanford.edu/class/cme193/syllabus.html>

## 2.3 Ασκήσεις

### Άσκηση 1: Λίστες και λεξικά

Χρησιμοποιώντας την εντολή `dir` μπορείτε να βλέπετε τα περιεχόμενα ενός αντικειμένου στην Python, ενώ με την `help` μπορείτε να εκτυπώνετε τη “βοήθεια” για ένα αντικείμενο της Python. Για παράδειγμα:

```
In [4]: dir(list)
Out[4]:
['__add__',
 '__class__',
 '__contains__',
 '__delattr__',
 '__delitem__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattr__',
 '__getitem__',
 '__gt__',
 '__hash__',
 '__iadd__',
 '__imul__',
 '__init__',
 '__init_subclass__',
 '__iter__',
 '__le__',
 '__len__',
 '__lt__',
 '__mul__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__reversed__',
 '__rmul__',
 '__setattr__',
 '__setitem__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 'append',
 'clear',
 'copy',
 'count',
 'extend',
 'index',
 'insert',
 'pop',
 'remove',
 'reverse',
 'sort']
```

```
In [5]: help(list.pop)
Help on method_descriptor:
```

```
pop(self, index=-1, /)
    Remove and return item at index (default last).
```

```
Raises IndexError if list is empty or index is out of range.
```

Χρησιμοποιείτε τις `dir` και `help` για να εκτυπώσετε τη βοήθεια για διαφορετικές συναρτήσεις για λίστες και λεξικά της Python.

## Άσκηση 2: List comprehensions

Γράψτε ένα list comprehension που από μία λίστα δημιουργεί μία εκδοχή με πεζά γράμματα της κάθε συμβολοσειράς που έχει μήκος μεγαλύτερο του 5.

## Άσκηση 3: Quicksort

Υλοποιήστε μία συνάρτηση `quicksort()` που υλοποιεί τον αλγόριθμο quicksort χρησιμοποιώντας list comprehensions. Ο quicksort εκτελεί αναδρομικά τα ακόλουθα βήματα

1. Αν η λίστα είναι άδεια, επέστρεψε. Αλλιώς διάλεξε ένα στοιχείο της λίστας ως pivot.
2. Χώρισε τα στοιχεία σε δύο υπο-λίστες, αυτά που είναι μικρότερα του pivot και αυτά που είναι μεγαλύτερα.
3. Τρέξε αναδρομικά τη συνάρτηση για κάθε υπο-λίστα

Για περισσότερες πληροφορίες, μπορείτε να διαβάσετε γι τον quicksort στη wikipedia

<https://en.wikipedia.org/wiki/Quicksort>

## 3 Tutorial

Σας δίνεται ένα αρχείο `tutorial.zip` το οποίο περιέχει τρία ερωτήματα.

```
[diou@zenbook:../prep]$ unzip tutorial.zip
[diou@zenbook:../prep]$ cd tutorial
[diou@zenbook:../tutorial]$ ls
addition.py          __pycache__          test_cases           tutorialTestClasses.py
autograder.py        shopAroundTown.py    testClasses.py       tutorial.token
buyLotsOfFruit.py    shop.py               testParser.py        util.py
grading.py           shopSmart.py          textDisplay.py
projectParams.py     submission_autograder.py town.py
[diou@zenbook:../tutorial]$
```

Θα χρειαστεί να δουλέψετε με τα παρακάτω αρχεία

- `addition.py` : Άσκηση 1
- `buyLotsOfFruit.py` : Άσκηση 2
- `shopSmart.py` : Άσκηση 3
- `autograder.py` : Script του autograder

Εκτελώντας τον autograder προκύπτει αυτόματα η βαθμολόγηση των απαντήσεών σας:

```
(yp23) [diou@zenbook:../tutorial]$ python autograder.py
```

Παρατηρήστε την έξοδο της παραπάνω εντολής (προφανώς ακόμα δεν έχετε λύσει τις ασκήσεις, οπότε ο βαθμός σας είναι μηδέν).

### 3.1 Άσκηση 1: Πρόσθεση

Γράψτε μία συνάρτηση που επιστρέφει το άθροισμα δύο αριθμών τροποποιώντας το αρχείο `addition.py`. Μόλις την υλοποιήσετε, εκτελέστε ξανά τον autograder και δείτε αν η λύση σας λαμβάνεται ως σωστή.

### 3.2 Άσκηση 2: Συνάρτηση `buyLotsOfFruit`

Προσθέστε μία συνάρτηση `buyLotsOfFruit(orderList)` στο αρχείο `buyLotsOfFruit.py` που δέχεται μία λίστα από ζευγάρια `(fruit, kg)` και επιστρέφει το κόστος της λίστας. Αν υπάρχουν φρούτα που δεν υπάρχουν στη `fruitPrices` πρέπει να εκτυπώνει ένα μήνυμα σφάλματος και να επιστρέφει `None`. Μην αλλάξετε τη μεταβλητή `fruitPrices`.

Εκτελέστε τον autograder έως ότου το ερώτημα 2 να περνά όλα τα τεστ.

### 3.3 Άσκηση 3: Συνάρτηση `shopSmart`

Γράψτε τη συνάρτηση `shopSmart(orders, shops)` στο αρχείο `shopSmart.py`, η οποία παίρνει στην είσοδο μία `orderList` και μία λίστα από `FruitShop` και επιστρέφει το `FruitShop` που η παραγγελία σας κοστίζει λιγότερο. Μην αλλάξετε το όνομα του αρχείου ή τις μεταβλητές.

Εκτελέστε τον autograder έως ότου η συνάρτησή σας να περνά όλα τα τεστ.