



ΠΕΡΙΕΧΟΜΕΝΑ:

1. Σειριοποίηση
2. Αποσειριοποίηση

## ΜΑΘΗΜΑ 5.1: Το module pickle

### 1. Σειριοποίηση

- Το **module pickle** περιέχει μεθόδους για:
  - την **σειριοποίηση (serialization)** μεταβλητών και αντικειμένων
  - την **απόσειριοποίηση (deserialization)** δυαδικών σειριοποιημένων δεδομένων

Σε αντίθεση με άλλους τρόπους (π.χ. JSON, XML)

- η σειριοποίηση γίνεται στο δυαδικό και τα δεδομένα δεν είναι αναγνώσιμα χωρίς αποσειριοποίηση
- Τα δεδομένα που παράγονται είναι διαχειρίσιμα μόνο από προγράμματα Python.
- Με εύκολο τρόπο μπορεί να σειριοποιήσει αντικείμενα δικών μας κλάσεων.

#### Σειριοποίηση:

- Συναρτήσεις:

Μέθοδος	Επεξήγηση
<code>dumps(obj)</code>	Επιστρέφει τη μετατροπή σε συμβ/ρα του αντικειμένου <code>obj</code> (στο δυαδικό)
<code>dump(obj, file)</code>	Αποθηκεύει στο αρχείο <code>file</code> τα δυαδικά δεδομένα του αντικειμένου <code>obj</code>

- Σώζει/μετατρέπει όλους τους συνήθεις τύπους της python, αντικείμενα χρήστη (λίγες εξαιρέσεις προχωρημένων χαρακτηριστικών όπως λαμδα, νήματα, συνδέσεις ΒΔ κ.α.)

#### Παράδειγμα 1: `ser.py`

```
from pickle import dump, dumps

class Data:
    def __init__(self, alist, adict, anint, aset, adata):
        self.alist = alist
        self.adict = adict
        self.anint = anint
        self.aset = aset
        self.adata = adata

l = [1, 2, 3]
print(dumps(l))

with open("data.pkl", "w"):
    d1 = Data([1, 2], {1: 1, 2: 2}, 1, {1, 2}, None)
    dumps(d1)
    d2 = Data([3], {3: 3}, 3, {3}, d1)
    dumps(d2)
    d3 = Data([], {}, 0, set(), d2)
    dumps(d3)
```

#### Σημείωση:

- Οι συναρτήσεις δέχονται προαιρετικό όρισμα το `protocol` (0..5) που καθορίζει την έκδοση του λογισμικού κωδικοποίησης (def: 5)

Αποσειριοποίηση:

- Συναρτήσεις:

Μέθοδος	Επεξήγηση
loads(obj)	Επιστρέφει το αντικείμενο το οποίο έχει κωδικοποιηθεί με τη dumps
load(obj,f)	Διαβάζει από το αρχείο και επιστρέφει το αντικείμενο το οποίο έχει κωδικοποιηθεί με τη dump

Παράδειγμα 2: pickle loads.py

```
from pickle import dumps, loads

d = [(1,3),{1,2},dumps]

serialized = dumps(d)
print(serialized)

deserialized = loads(serialized)
print(deserialized)
```

Παράδειγμα 3: pickle\_load.py

```
from pickle import load

class Data:
    def __init__(self, alist, adict, anint, aset, adata):
        self.alist = alist
        self.adict = adict
        self.anint = anint
        self.aset = aset
        self.adata = adata

    def __str__(self):
        return str(self.alist) + ", " + str(self.adict) + ", " + \
            str(self.anint) + ", " + str(self.aset) + \
            ", Data(" + str(self.adata) + ")"

with open("data.pkl", "rb") as f:
    try:
        ob = load(f)
        while ob is not None:
            print(ob)
            ob = load(f)
    except EOFError:
        pass
```