ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά μαθήματα ΠΠ

Εισαγωγή στους Η/Υ και τις Εφαρμογές

Ενότητα 5: Επεξεργασία δεδομένων με τη γλώσσα προγραμματισμού python

Υπο-ενότητα 6.2: Βιβλιοθήκες χειρισμού δεδομένων

Μανώλης Τζαγκαράκης, Βικτωρία Δασκάλου

Σχολή Οργάνωσης και Διοίκησης Επιχειρήσεων

Τμήμα Οικονομικών Επιστημών

Σκοποί ενότητας

- Να πραγματοποιηθεί μία εισαγωγή στις δυνατότητες της γλώσσας python
- Να παρουσιαστούν οι βασικές δυνατότητες των γλωσσών προγραμματισμού για την επεξεργασία δεδομένων

Περιεχόμενα ενότητας

1. Χαρακτηριστικά της γλώσσας python
2. Προγραμματισμός με python
   - Υπολογισμοί και μεταβλητές
   - Τύποι δεδομένων
   - **Βιβλιοθήκες χειρισμού δεδομένων**

Τρίτες βιβλιοθήκες χειρισμού δεδομένων

Σύστημα βιβλιοθηκών Python

- Στη γλώσσα Python όλα τα αρχεία που περιέχουν πηγαίο κώδικα Python (και έχουν επέκταση .py) μπορούν να επαναχρησιμοποιηθούν – ως βιβλιοθήκες ή πακέτα – μέσα σε άλλα αρχεία με πηγαίο κώδικα Python.
  - Επαναχρησιμοποίηση των μεταβλητών, συναρτήσεων κλπ που ορίζονται στο αρχείο .py που επαναχρησιμοποιείται.
- Τέτοια αρχεία πηγαίου κώδικα Python που ενσωματώνονται σε άλλα αρχεία πηγαίου κώδικα Python με στόχο την επαναχρησιμοποίηση των μεταβλητών ή συναρτήσεων που ορίζουν, καλούνται βιβλιοθήκες ή πακέτα Python (modules)
  - Ενσωμάτωση γίνεται με την ειδική εντολή **import** της Python

Σύστημα βιβλιοθηκών Python

- Παράδειγμα

Αρχείο *myModule.py*

```
a = 42
b = 43
```

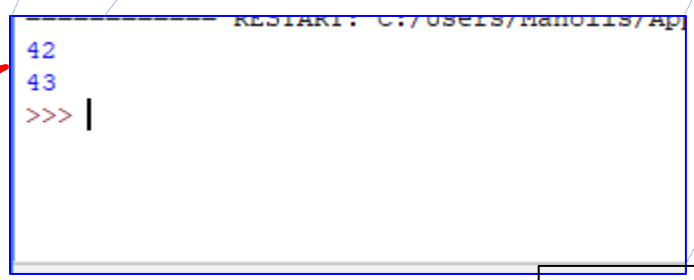
Αρχείο .py που ορίζει και αρχικοποιεί 2 μεταβλητές.

Αρχείο *importExample.py*

```
import myModule

print(myModule.a)
print(myModule.b)
```

Ενσωμάτωση αρχείου myModule.py στο αρχείο πηγαίου κώδικα Python importExample.py. Αυτό κάνει το αρχείο myModule.py βιβλιοθήκη/module. Οι μεταβλητές a και b θα είναι διαθέσιμες και στο αρχείο importExample.



Αποτέλεσμα εκτέλεσης αρχείου importExample.py. Οι τιμές είναι των μεταβλητών a και b που ορίζονται στο module myModule..py

ΣΗΜ: Αρχεία myModule.py και importExample.py πρέπει να βρίσκονται στον ίδια φάκελο

Σύστημα βιβλιοθηκών Python

- Το σύστημα βιβλιοθηκών είναι ένας τρόπος που παρέχει η Python ώστε να ελέγχεται η πολυπλοκότητα των προγραμμάτων.
- Μεταβλητές και συναρτήσεις να χωρίζονται σε ξεχωριστά αρχεία ανάλογα με τη σημασιολογία τους ή τη δουλειά που κάνουν και να επαναχρησιμοποιούνται όπου χρειάζεται
  - Π.χ. συναρτήσεις στατιστικής να ορίζονται σε μία βιβλιοθήκη (=αρχείο .py), συναρτήσεις μαθηματικών σε άλλη, συναρτήσεις ανάγνωσης αρχείων σε άλλη κλπ
    - Ενσωμάτωση στα προγράμματα μόνο όσων βιβλιοθηκών χρειάζονται.
  - «Διαίρει και βασίλευε» προσέγγιση

Σύστημα βιβλιοθηκών Python

- Το σύστημα βιβλιοθηκών Python επιτρέπει την ενσωμάτωση βιβλιοθηκών που αναπτύσσουν και διανέμουν άλλοι (τρίτοι) για τη επαναχρησιμοποίηση κώδικα και έτσι την εύκολη επέκταση των δυνατοτήτων της Python
  - Π.χ. βιβλιοθήκη **numpy** για την παροχή τύπων δεδομένων όπως vector, array που ΔΕΝ παρέχει εγγενών η python
  - Π.χ. βιβλιοθήκη **pandas** για την παροχή τύπων δεδομένων όπως data frame (πλαίσιο δεδομένων) και την ανάγνωση csv αρχείων (που ούτε αυτό παρέχει εγγενώς η Python)

Σύστημα βιβλιοθηκών Python

- import
  - Διάφορες μορφές σύνταξης της import
    - import <όνομα module> [as <new module name>]
      - Π.χ. import pandas as pd
    - from <όνομα module> import <μεταβλητή/συνάρτηση>

Ενσωμάτωαη της βιβλιοθήκης pandas στο πρόγραμμα με το όνομα pd.
Το όνομα της βιβλιοθήκης pandas (εδώ pd) θα χρησιμοποιηθεί ως πρόθεμα κάθε χρήσης συνάρτησης. Π..χ η συνάρτηση read_csv() που παρέχει η pandas θα αναφερθεί ως Pd.read_csv(…)

Σύστημα βιβλιοθηκών Python

- *import <όνομα βιβλιοθήκης>*
  - Ενσωματώνει **όλες** τις μεταβλητές, τύπους δεδομένων και συναρτήσεις που ορίζει η βιβλιοθήκη με όνομα <όνομα βιβλιοθήκης>
  - Αναφορά στα αντικείμενα αυτά με χρήση προθέματος/ονόματος της βιβλιοθήκης (ονοματοχώρος).
    - Π.χ. εάν ορίζει η βιβλιοθήκη μεταβλητή με όνομα age τότε η αναφορά σε αυτή θα πρέπει να γίνει ως εξής: **<όνομα βιβλιοθήκης>.age**

Σύστημα βιβλιοθηκών Python

- Παράδειγμα

Αρχείο *myModule.py*

```
a = 42
b = 43
```

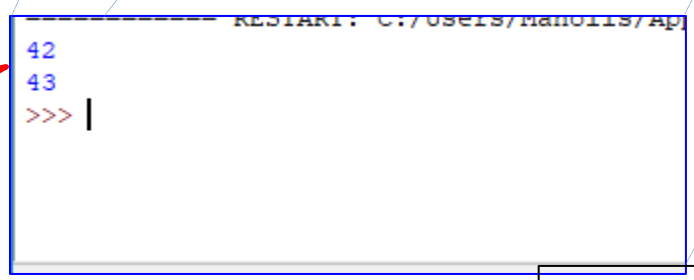
Αρχείο .py που ορίζει και αρχικοποιεί 2 μεταβλητές.

Αρχείο *importExample.py*

```
import myModule

print(myModule.a)
print(myModule.b)
```

Ενσωμάτωση αρχείου myModule.py στο αρχείο πηγαίου κώδικα Python importExample.py. Αυτό κάνει το αρχείο myModule.py βιβλιοθήκη/module. Οι μεταβλητές a και b θα είναι διαθέσιμες και στο αρχείο importExample.



Αποτέλεσμα εκτέλεσης αρχείου importExample.py. Οι τιμές είναι των μεταβλητών a και b που ορίζονται στο module myModule..py

ΣΗΜ: Αρχεία myModule.py και importExample.py πρέπει να βρίσκονται στον ίδια φάκελο

Σύστημα βιβλιοθηκών Python

- *from <όνομα βιβλιοθήκης> import <μεταβλητές,συναρτήσεις, τύποι δεδομένων>*
  - Ενσωματώνει **από την βιβλιοθήκη** *<όνομα βιβλιοθήκης>* μόνο **συγκεκριμένες** μεταβλητές, συναρτήσεις και τύπους δεδομένων που αναφέρονται ρητά. Όχι όλες!
    - Για λόγους οικονομίας μνήμης.
  - Αναφορά στα αντικείμενα αυτά με χρήση μόνο του ονόματός τους
    - Π.χ. εάν ορίζει η βιβλιοθήκη μεταβλητή με όνομα age και γίνει ενσωμάτωση με τότε η ***from <όνομα βιβλιοθήκης> import age*** η αναφορά στην age σε αυτή θα πρέπει να γίνει ως εξής: **age**

Σύστημα βιβλιοθηκών Python

- Παράδειγμα

Αρχείο *myModule.py*

```
a = 42
b = 43
```

Αρχείο .py που ορίζει και αρχικοποιεί 2 μεταβλητές.

Αρχείο *importExample.py*

```
from myModule import a

print(a)
print(myModule.b)
```

Από τη βιβλιοθήκη myModule ενσωμάτωση ΜΟΝΟ της μεταβλητής a, ΟΧΙ της b.

Αποτέλεσμα εκτέλεσης αρχείου importExample.py. Η τιμή της a θα εμφανιστεί, αλλά όχι της b (δεν έχει ενσωματωθεί)

```
42
Traceback (most recent call last):
  File "C:/Users/Manolis/AppData/Local/Programs/Python/Python38/importScriptExample.py", line 7, in <module>
    print(testlib1.b)
NameError: name 'testlib1' is not defined
>>>
```

ΣΗΜ: Αρχεία myModule.py και importExample.py πρέπει να βρίσκονται στον ίδια φάκελο

Βιβλιοθήκες χειρισμού δεδομένων

- Η Python παρέχει πλήθος βιβλιοθηκών (πακέτων/modules) που ορίζουν ειδικούς τύπους δεδομένων που επιτρέπουν τον βολικό χειρισμό δεδομένων
  - Παρέχουν τύπους δεδομένων όπως διάνυσμα και πλαίσιο δεδομένων (data frame) όπως και η R και οι οποίοι δεν παρέχονται εγγενώς από την Python.

Βιβλιοθήκες χειρισμού δεδομένων

- Βιβλιοθήκες πρέπει πρωτίστως να εγκατασταθούν στον υπολογιστή πριν την ενσωμάτωσή τους στο πρόγραμμα με χρήση της import
  - Εγκατάσταση μέσω του package manager pip:
    - Σύνταξη: pip install <όνομα βιβλιοθήκης>
  - Εγκαθίστανται ΜΙΑ μόνο φορά στον υπολογιστή.

```
C:\> pip install pandas
```

Εγκατάσταση βιβλιοθήκης pandas. Πρέπει να γίνει μία μόνο φορά.

Βιβλιοθήκες χειρισμού δεδομένων

- Από τις πιο δημοφιλείς βιβλιοθήκες χειρισμού και ανάλυσης δεδομένων στην Python: pandas
  - Παρέχει τύπους δεδομένων και συναρτήσεις για την αναπαράσταση, χειρισμό και επεξεργασία δεδομένων.

Pandas

- pandas (**pan**el **da**ta **s**ystem)
  - *Panel data*: παρατηρήσεις για πολλά υποκείμενα έρευνας σε διαφορετικές χρονικές στιγμές.
  - Παρέχει τύπους δεδομένων και συναρτήσεις για την αναπαράσταση, χειρισμό και στατιστική επεξεργασία δεδομένων
  - Παρέχει πλαίσιο δεδομένων, συναρτήσεις ανάγνωσης αρχείων csv και excel, τεμαχισμό δεδομένων κλπ

Pandas

- Ανάγνωση αρχείων csv με τη συνάρτηση read_csv() της pandas. Βασικά ορίσματα
  - Όνομα αρχείου csv
  - Εάν έχει επικεφαλίδα
    - Τιμή None εάν δεν έχει, αριθμητική τιμή εάν έχει που δηλώνει τη θέση μέσα στο αρχείο
  - Διαχωριστής τιμών σε κάθε γραμμή
    - Προκαθορισμένο όρισμα το ‘,’ (κόμμα)
- Οι γραμμές/στήλες αποθηκεύονται σε μεταβλητή που είναι τύπου δεδομένων DataFrame (πλαίσιο δεδομένων)

```
>>> import pandas as pd  #Ενσωμάτωση βιβλιοθήκης pandas ώστε να είναι
                         διαθέσιμες οι σχετικές συναρτήσεις. Η
                         αναφορά σε συνάρτηση της pandas
                         (ονοματοχώρος) θα γίνει με το πρόθεμα pd.
>>>myData = pd.read_csv('example1.csv', header=None, sep=';')
>>>
```

Ανάγνωση αρχείου csv με όνομα example1.csv το οποίο δεν έχει επικεφαλίδα (header=None) και οι τιμές σε κάθε γραμμή διαχωρίζονται με τον χαρακτήρα ;

Pandas

- Πλαίσιο δεδομένων

Πράξη	Σημασία	Παράδειγμα
.shape	Οι διαστάσεις του πλαισίου δεδομένων (αριθμός γραμμών, αριθμός στηλών)	myData.shape >>> myData.shape (1324, 60)
.dtypes	Εμφανίζει τους τύπους δεδομένων των στηλών	myData.dtypes >>> myData.dtypes 0 int64 1 object 2 object 3 int64 4 int64 5 int64

Πλαίσιο δεδομένων με: 1324 γραμμές, 60 στήλες

Object εδώ σημαίνει ότι η στήλη έχει μόνο τιμές τύπου δεδομένων συμβολοσειράς ή ανακατεμένους τύπους δεδομένων (αριθμητικές και μη-).

Pandas

- Τεμαχισμός πλαισίου δεδομένων με 3 τρόπους:

Τελεστές δεικτοδότησης/τεμαχισμού	Σημασία
[ ]	Τεμαχίζει ως προς μία διάσταση μόνο (γραμμές ή στήλες. Όχι και τα δύο) χρησιμοποιώντας μόνο ετικέτες (ονόματα). Όχι θέσεις.
.loc[ ]	Τεμαχίζει και ως προς τις δύο διαστάσεις αλλά δέχεται μόνο με ονόματα/ετικέτες ως προσδιοριστές γραμμών και στηλών (όχι αριθμό θέσης!)
.iloc[ ]	Τεμαχίζει και ως προς τις δύο διαστάσεις αλλά μόνο με θεσιακούς προσδιοριστές γραμμών και στηλών.

Pandas

- Τελεστής [ ]

```
>>> import pandas as pd  #Ενσωμάτωση βιβλιοθήκης pandas ώστε να είναι
                         διαθέσιμες οι σχετικές συναρτήσεις. Η
                         αναφορά σε συνάρτηση της pandas
                         (ονοματοχώρος) θα γίνει με το πρόθεμα pd.

>>>myData = pd.read_csv('GrCensus2011.csv', header=None, sep=';')
>>>myData[0]
0     0
1     1
2     2
3     3
4     4
     ..
1319  6
1320  6
1321  6
1322  6
1323  5
Name: 0, Length: 1324, dtype: int64
```

Ονομαστική αναφορά **στήλης** (στήλη με ετικέτα 0) εάν δεν υπάρχουν επικεφαλίδες.

Pandas

- Τελεστής [ ]

```
>>> import pandas as pd  #Ενσωμάτωση βιβλιοθήκης pandas ώστε να είναι διαθέσιμες οι σχετικές συναρτήσεις. Η αναφορά σε συνάρτηση της pandas (ονοματοχώρος) θα γίνει με το πρόθεμα pd.

>>>myData = pd.read_csv('example1.csv', header=0, sep=';')
>>>myData['sex']
0    male
1    female
2    female
3    male
Name: sex, dtype: object
```

Ονομαστική αναφορά **στήλης με χρήση ετικέτας σε πλαίσιο δεδομένων με επικεφαλίδα**

```
name;age;sex;favoriteMovie
Jim;42;male;Good, the bad and the ugly
Maria;52;female;The fly
Dimitra;28;female;Sex and the city II
Kostas;59;male;Rembetiko
```

Pandas

- Τελεστής [ ]

```
>>> import pandas as pd  #Ενσωμάτωση βιβλιοθήκης pandas ώστε να είναι
                         διαθέσιμες οι σχετικές συναρτήσεις. Η
                         αναφορά σε συνάρτηση της pandas
                         (ονοματοχώρος) θα γίνει με το πρόθεμα pd.

>>>myData = pd.read_csv('GrCensus2011.csv', header=None, sep=';')
>>>myData[1:3]
```

Θεσιακή αναφορά **ΓΡΑΜΜΗΣ** (γραμμές από θέση 1 έως και 2)

```
>>> myData[1:3]
   0  1                                  2  ...     57     58     59
1  1  1                      ΒΟΡΕΙΑ ΕΛΛΑΔΑ  ...  84335  62642  36715
2  2  11  ΑΠΟΚΕΝΤΡΩΜΕΝΗ ΔΙΟΙΚΗΣΗ ΜΑΚΕΔΟΝΙΑΣ - ΘΡΑΚΗΣ  ...  65863  48511  26510

[2 rows x 60 columns]
```

Pandas

```
name;age;sex;favoriteMovie
Jim;42;male;Good, the bad and the ugly
Maria;52;female;The fly
Dimitra;28;female;Sex and the city II
Kostas;59;male;Rembetiko
```

- Τελεστής .loc[]
  - Επιλογή γραμμών και στηλών ΜΟΝΟ με χρήση ετικετών. Όχι θέσεων

```
>>> import pandas as pd  #Ενσωμάτωση βιβλιοθήκης pandas ώστε να είναι
                         διαθέσιμες οι σχετικές συναρτήσεις. Η
                         αναφορά σε συνάρτηση της pandas
                         (ονοματοχώρος) θα γίνει με το πρόθεμα pd.

>>>myData = pd.read_csv('example1.csv', header=0, sep=';')
>>>myData.loc[1:3, : ]

      name  age    sex       favoriteMovie
1    Maria   52  female             The fly
2  Dimitra   28  female  Sex and the city II
3   Kostas   59    male            Rembetiko
```

Όλες οι στήλες

Ονομαστικός (ετικέτες) προσδιορισμός γραμμών: από **ετικέτα 1 έως και ετικέτα 3** (***ΟΧΙ θέση!***)

Απόδειξη ότι το 1:3 ΔΕΝ αναφέρεται σε θέση: εμφανίζεται ΚΑΙ η γραμμή 3.

Pandas

- Τελεστής .iloc[]
  - Επιλογή γραμμών και στηλών ΜΟΝΟ με χρήση αριθμό θέσεων. Όχι ετικετών ή ονομάτων.

```
>>> import pandas as pd  #Ενσωμάτωση βιβλιοθήκης pandas ώστε να είναι διαθέσιμες οι σχετικές συναρτήσεις. Η αναφορά σε συνάρτηση της pandas (ονοματοχώρος) θα γίνει με το πρόθεμα pd.

>>>myData = pd.read_csv('example1.csv', header=0, sep=';')
>>>myData.iloc[1, 3]

The fly
```

Θεσιακός προσδιορισμός κελιού που βρίσκεται στη **γραμμή στη θέση 1 και στήλη στη θέση 3.**

```
name;age;sex;favoriteMovie
Jim;42;male;Good, the bad and the ugly
Maria;52;female;The fly
Dimitra;28;female;Sex and the city II
Kostas;59;male;Rembetiko
```

Pandas

- Τελεστής .iloc[]
  - Επιλογή γραμμών και στηλών ΜΟΝΟ με χρήση αριθμό θέσεων. Όχι ετικετών ή ονομάτων.

```
>>> import pandas as pd  #Ενσωμάτωση βιβλιοθήκης pandas ώστε να είναι διαθέσιμες οι σχετικές συναρτήσεις. Η αναφορά σε συνάρτηση της pandas (ονοματοχώρος) θα γίνει με το πρόθεμα pd.

>>>myData = pd.read_csv('example1.csv', header=0, sep=';')
>>>myData.iloc[1:3, 2:4 ]
```

	sex	favoriteMovie
1	female	The fly
2	female	Sex and the city II
3	male	Rembetiko

Θεσιακός προσδιορισμός εύρους **γραμμών και στηλών**. Τελεστής τεμαχισμού **:** λαμβάνει τα ίδια ορίσματα όπως στην περίπτωση συμβολοσειρών: *start* : *stop* : *stride*

```
name;age;sex;favoriteMovie
Jim;42;male;Good, the bad and the ugly
Maria;52;female;The fly
Dimitra;28;female;Sex and the city II
Kostas;59;male;Rembetiko
```

Άσκηση

- Έστω αρχείο δεδομένων csv με όνομα data.csv που περιέχει τις ακόλουθες γραμμές:

```
A;    B;    C
1;    2;    3;
4;    5;    6;
7;    8;    9;
10;   11;   12;
```

Ο σωστός τρόπος ανάγνωσης του περιεχομένου του αρχείου σε πλαίσιο δεδομένων στην Python είναι:

i. myData = pandas.read_csv('data.csv', header=None, sep=';')

ii. myData = pandas.read_csv('data.csv', header=0, sep=',')

iii. myData = pandas.read_csv('data.csv', header=0, sep=';')

Άσκηση

- Έστω αρχείο δεδομένων csv με όνομα data.csv που περιέχει τις ακόλουθες γραμμές:

```
A;   B;   C
1;   2;   3;
4;   5;   6;
7;   8;   9;
10;  11;  12;
```

Ο σωστός τρόπος ανάγνωσης του περιεχομένου του αρχείου σε πλαίσιο δεδομένων στην Python είναι:

i. myData = pandas.read_csv('data.csv', header=None, sep=';')

ii. myData = pandas.read_csv('data.csv', header=0, sep=',')

iii. myData = pandas.read_csv('data.csv', header=0, sep=';')

Άσκηση

- Έστω αρχείο δεδομένων csv με όνομα data.csv που περιέχει τις ακόλουθες γραμμές:

A; B; C
1; 2; 3;
4; 5; 6;
7; 8; 9;
10; 11; 12;

το οποίο διαβάζεται σε μία μεταβλητή με όνομα myData που είναι πλαίσιο δεδομένων. Η έκφραση:

myData.loc[3:, 1:2]

θα εμφανίσει στην οθόνη :

i. 10

ii. 11

iii. 11 12

iv. Κανένα από τα παραπάνω.

Άσκηση

- Έστω αρχείο δεδομένων csv με όνομα data.csv που περιέχει τις ακόλουθες γραμμές:

```
A;   B;   C
1;   2;   3;
4;   5;   6;
7;   8;   9;
10;  11;  12;
```

το οποίο διαβάζεται σε μία μεταβλητή με όνομα myData που είναι πλαίσιο δεδομένων. Η έκφραση:

```
myData.loc[ 3:, 1:2]
```

θα εμφανίσει στην οθόνη :

i. 10

ii. 11

iii. 11 12

iv. Κανένα από τα παραπάνω.

Pandas

- Μία έκφραση τεμαχισμού που εκτελείται, μπορεί να επιστρέψει δύο είδη τύπους δεδομένων, ανάλογα με το που εφαρμόζεται:
  - Πλαίσιο δεδομένων (Data frame), εάν επιστρέφονται παραπάνω από μία στήλες.
  - Σειρά (Series) εάν επιστρέφεται μία μόνο στήλη.

```
>>> import pandas as pd
>>>myData = pd.read_csv('example1.csv', header=0, sep=';')
>>>type( myData [ ['age', 'name'] ] )
<class 'pandas.core.frame.DataFrame'>

>>>type( myData [ 'age' ] )
<class 'pandas.core.series.Series'>
```

Επιστρεφόμενος τύπος δεδομένων από εκφράσεις τεμαχισμού σε πλαίσιο δεδομένων.

Pandas

- Σειρά/Series
  - Τύπος δεδομένων που ορίζεται από τη βιβλιοθήκη Pandas *είναι μία μονοδιάστατη αναπαράσταση δεδομένων οποιουδήποτε τύπου δεδομένων (int, str, list, dict etc) με ετικέτες*.
    - Σειρές μπορούν να μετατραπούν σε πίνακες ή λίστες με την .values

```
>>> import pandas as pd
>>>myData = pd.read_csv('example1.csv', header=0, sep=';')
>>>type( myData [ 'age' ] )
<class 'pandas.core.series.Series'>

>>>type( myData [ 'age' ] .values )
<class 'numpy.ndarray'>
```

Μετατροπή και εμφάνιση της Σειράς 'age' ρπίνακα, λίστα τιμών με χρήση της .values.

Pandas

- Τεμαχισμός πλαισίου δεδομένων με κριτήρια
  - Π.χ. «*Όλες οι γραμμές όπου η στήλη X έχει τιμή y*»
- Βασίζεται στο ότι όλοι οι τελεστές δεικτοδότησης/τεμαχισμού ([], loc, iloc) δέχονται λίστες με λογικές τιμές που δηλώνουν ποιες γραμμές και στήλες να επιλεγούν

```
>>> import pandas as pd
>>>myData = pd.read_csv('example1.csv', header=0, sep=';')
>>>myData [ [True, False, False, False] ]

name  age   sex          favoriteMovie
0  Jim   42  male  Good, the bad and the ugly
```

Λίστα με τιμές True, False, μία για κάθε γραμμή που δηλώνει εάν η αντίστοιχη γραμμή να εμφανιστεί (True) ή όχι (False)

Pandas

- Τεμαχισμός πλαισίου δεδομένων με κριτήρια
  - Π.χ. «*Όλες οι γραμμές όπου η στήλη X έχει τιμή y*»
- **ΠΡΟΣΟΧΗ! Η λίστα με τις λογικές τιμές πρέπει να έχει το ίδιο πλήθος στοιχείων με το πλαίσιο δεδομένων όπου εφαρμόζεται!**

```
>>> import pandas as pd
>>>myData = pd.read_csv('example1.csv', header=0, sep=';')
>>>myData [ [True, False, False] ]

ValueError: Item wrong length 3 instead of 4.
```

Πλαίσιο δεδομένων myData έχει 4 γραμμές, ενώ η σειρά που προσδιορίζει τις επιθυμητές γραμμές έχει 3 τιμές.

Pandas

- Τεμαχισμός πλαισίου δεδομένων με κριτήρια
  - Π.χ. «*Όλες οι γραμμές όπου η στήλη X έχει τιμή y*»
- Τελεστής **.loc** και λίστα λογικών τιμών

```
>>> import pandas as pd
>>>myData = pd.read_csv('example1.csv', header=0, sep=';')
>>>myData.loc[ [True, False, False, False], : ]

name  age   sex          favoriteMovie
0  Jim   42  male  Good, the bad and the ugly
```

Λίστα με τιμές True, False, μία για κάθε γραμμή που δηλώνει εάν η αντίστοιχη γραμμή να εμφανιστεί (True) ή όχι (False)

Pandas

- Τεμαχισμός πλαισίου δεδομένων με κριτήρια
  - Π.χ. «*Όλες οι γραμμές όπου η στήλη X έχει τιμή y*»
- Τελεστής **.loc** και λίστα λογικών τιμών για γραμμές και στήλες

```
>>> import pandas as pd
>>>myData = pd.read_csv('example1.csv', header=0, sep=';')
>>>myData.loc[ [True, False, False, False], [True, False, False, True] ]

name    favoriteMovie
0  Jim    Good, the bad and the ugly
```

Λίστα με τιμές True, False, για κάθε γραμμή και μία για κάθε στήλη που δηλώνει εάν η αντίστοιχη γραμμή να εμφανιστεί (True) ή όχι (False) και για τις γραμμές αυτές, ποιες στήλες.

Pandas

- Τεμαχισμός πλαισίου δεδομένων με κριτήρια
  - Π.χ. «*Όλες οι γραμμές όπου η στήλη X έχει τιμή y*»
- Τελεστής **.iloc** και λίστα λογικών τιμών για γραμμές και στήλες

```
>>> import pandas as pd
>>>myData = pd.read_csv('example1.csv', header=0, sep=';')
>>>myData.iloc[ [True, False, False, False], [True, False, False, True] ]

name    favoriteMovie
0  Jim    Good, the bad and the ugly
```

Λίστα με τιμές True, False, για κάθε γραμμή και μία για κάθε στήλη που δηλώνει εάν η αντίστοιχη γραμμή να εμφανιστεί (True) ή όχι (False) και για τις γραμμές αυτές, ποιες στήλες.

Pandas

- Εφαρμογή τελεστών (<,>,==,>=,=<, !=) πάνω σε ολόκληρη στήλη πλαισίου δεδομένων επιστρέφει Σειρά (Series) με τιμές True/False
  - Μία για κάθε γραμμή του πλαισίου δεδομένων που δηλώνει εάν για τη γραμμή αυτή ισχύει η λογική συνθήκη

```
>>> import pandas as pd
>>>myData = pd.read_csv('example1.csv', header=0, sep=';')
>>>myData['age']  > 30
0    True
1    True
2   False
3    True
Name: age, dtype: bool
```

Αποτέλεσμα Σειράς με τιμές True/False μία για κάθε γραμμή που δηλώνει εάν η συνθήκη (>30) ισχύει ή όχι.

Pandas

- Εφόσον οι τελεστές σύγκρισης πάνω σε στήλες επιστρέφουν σειρές με τιμές True/False, αυτές οι εκφράσεις μπορούν να μπουν ως προσδιοριστές γραμμών και στηλών και να γίνει επιλογή γραμμών βάσει κριτηρίων.
  - Ισχύει για όλους τους τελεστές: [], loc, iloc

```
>>> import pandas as pd
>>>myData = pd.read_csv('example1.csv', header=0, sep=';')
>>>myData[ myData['age']  > 30 ]

name  age    sex          favoriteMovie
0    Jim   42    male  Good, the bad and the ugly
1   Maria   52  female              The fly
3  Kostas   59    male             Rembetiko
```

Τελεστή σύγκρισης, που επιστρέφει σειρά τιμών True/False, που χρησιμοποιείται ως προσδιοριστής γραμμών: θα εμφανιστούν μόνο εκείνες οι γραμμές (όλες οι στήλες) όπου ηλικία > 30

Pandas

- Εφόσον οι τελεστές σύγκρισης πάνω σε στήλες επιστρέφουν τιμές True/False, αυτές οι εκφράσεις μπορούν να μπουν ως προσδιοριστές γραμμών και στηλών και να γίνει επιλογή γραμμών βάσει κριτηρίων.
  - Ισχύει για όλους τους τελεστές: [], .loc, .iloc

```
>>> import pandas as pd
>>>myData = pd.read_csv('example1.csv', header=0, sep=';')
>>>myData.loc[ myData.loc[:,'age'] > 30, 'sex']

0     male
1   female
3     male
Name: sex, dtype: object
```

Στην έκφραση κριτηρίων θα μπορούσε να μπει και ο τελεστής [] δλδ
myData['age'] > 30

Pandas

- Προσοχή στη χρήση της .iloc εάν χρησιμοποιείται για τεμαχισμό με κριτήρια!
  - Το πρώτο όρισμα ΔΕΝ πρέπει να είναι Σειρά με τιμές True/False. ΠΡΕΠΕΙ να είναι πίνακας ή λίστα με τιμές True/False

```
>>> import pandas as pd
>>>myData = pd.read_csv('example1.csv', header=0, sep=';')
>>>myData.iloc[ (myData.iloc[:,1] > 30).values, 2]
0     male
1   female
3     male
Name: sex, dtype: object
>>>myData.iloc[ myData.iloc[:,1'] > 30, 'sex']
NotImplementedError: iLocation based boolean indexing on an integer type is
not available
```

Η .iloc απαιτεί ως πρώτο όρισμα πίνακα ή λίστα με τιμές True/False, όχι σειρά.

Άσκηση

- Έστω αρχείο δεδομένων csv με όνομα data.csv που περιέχει τις ακόλουθες γραμμές:

A; B; C

1; 2; 3;

4; 5; 6;

7; 8; 9;

10; 11; 12;

το οποίο διαβάζεται σε μία μεταβλητή με όνομα myData που είναι πλαίσιο δεδομένων. Η έκφραση:

myData.iloc[(myData.iloc[:, 0] > 6).values, 'C']

θα εμφανίσει στην οθόνη :

i) 9

12

ii) 9

10

iii) Τίποτα από τα παραπάνω.

Άσκηση

- Έστω αρχείο δεδομένων csv με όνομα data.csv που περιέχει τις ακόλουθες γραμμές:

A; B; C
1; 2; 3;
4; 5; 6;
7; 8; 9;
10; 11; 12;

το οποίο διαβάζεται σε μία μεταβλητή με όνομα myData που είναι πλαίσιο δεδομένων. Η έκφραση:

myData.iloc[(myData.iloc[:, 0] > 6).values, 'C']

θα εμφανίσει στην οθόνη :

i) 9
12

ii) 9
10

iii) Τίποτα από τα παραπάνω.

Άσκηση

- Έστω ότι εκτελείται το παρακάτω τμήμα κώδικα Python:

```
import pandas as pd
grC = pd.read_csv('GrCensus2011.csv', header=None, sep=';')
```

Η εκτέλεση της εντολής

```
print( grC[ grC[0] == 5 ] )
```

Θα εμφανίσει στην οθόνη

i. Όλες τις στήλες μόνο όλων των δημοτικών ενοτήτων
ii. Μόνο τη στήλη 5 μόνο όλων των δήμων.
iii. Όλες τις στήλες μόνο όλων των δήμων
iv. Μόνο τη στήλη 5 μόνο όλων των δημοτικών ενοτήτων.

Άσκηση

- Έστω ότι εκτελείται το παρακάτω τμήμα κώδικα Python:

```
import pandas as pd
grC = pd.read_csv('GrCensus2011.csv', header=None, sep=';')
```

Η εκτέλεση της εντολής

```
print( grC[ grC[0] == 5 ] )
```

Θα εμφανίσει στην οθόνη

i. Όλες τις στήλες μόνο όλων των δημοτικών ενοτήτων

ii. Μόνο τη στήλη 5 μόνο όλων των δήμων.

iii. Όλες τις στήλες μόνο όλων των δήμων

iv. Μόνο τη στήλη 5 μόνο όλων των δημοτικών ενοτήτων.

Pandas

- Κριτήρια επιλογής με τελεστές σύγκρισης μπορούν να συνδυαστούν με λογικούς τελεστές που κάνουν την πράξη ανά στοιχείο

&	Λογικό ΚΑΙ ανά στοιχείο
|	Λογικό Η ανά στοιχείο
~	Άρνηση

```
>>> import pandas as pd
>>>myData = pd.read_csv('example1.csv', header=0, sep=';')
>>>myData[ (myData['age'] > 50) & (myData['sex']=='male') ] ['favoriteMovie']
3    Rembetiko
Name: favoriteMovie, dtype: object
```

Εμφάνιση αγαπημένης ταινίας εκείνων που έχουν ηλικία > 50 και είναι άντρες.

Άσκηση

- Έστω αρχείο δεδομένων csv με όνομα data.csv που περιέχει τις ακόλουθες γραμμές:

A; B; C

1; 2; 3;

4; 5; 6;

7; 8; 9;

10; 11; 12;

το οποίο διαβάζεται σε μία μεταβλητή με όνομα myData που είναι πλαίσιο δεδομένων. Η έκφραση:

myData.loc[myData.iloc[1:3, 2] < 5, ”B”]

θα εμφανίσει στην οθόνη :

i. 2

ii. 11

iii. 2 5 8

iv. Κανένα από τα παραπάνω.

Άσκηση

- Έστω αρχείο δεδομένων csv με όνομα data.csv που περιέχει τις ακόλουθες γραμμές:

A; B; C
1; 2; 3;
4; 5; 6;
7; 8; 9;
10; 11; 12;

το οποίο διαβάζεται σε μία μεταβλητή με όνομα myData που είναι πλαίσιο δεδομένων. Η έκφραση:

myData.loc[myData.iloc[1:3, 2] < 5, ”B”]

θα εμφανίσει στην οθόνη :

i. 2

ii. 11

iii. 2 5 8

iv. Κανένα από τα παραπάνω.

Στατιστικές συναρτήσεις και pandas

- Η βιβλιοθήκη pandas επιτρέπει και την εφαρμογή συναρτήσεων περιγραφικής στατιστικής πάνω σε τεμαχισμένες γραμμές και στήλες

.mean()	Μέσος όρος
.var() και .std()	Διακύμανση και τυπική απόκλιση δείγματος
.min() και .max()	Ελάχιστη και μέγιστη τιμή

```
>>> import pandas as pd
>>>myData = pd.read_csv('example1.csv', header=0, sep=';')
>>>myData[1:3]['age'].mean()
40.0
```

Μέσος όρος μόνο των ηλικιών στις γραμμές από 1 έως και 2. Ισοδύναμη έκφραση με χρήση της .iloc: **myData.iloc[1:3, 1].mean()**

Άσκηση

- Έστω ότι εκτελείται το παρακάτω τμήμα κώδικα Python:

```
import pandas as pd
grC = pd.read_csv('GrCensus2011.csv', header=None, sep=';')
```

Το αρχείο GrCensus2011.csv είναι το ίδιο αρχείο που έχει χρησιμοποιηθεί στις εργασίες 1 και 2.

Η έκφραση

```
grCensusData.iloc[6:9, 3].max()
```

θα εμφανίσει στην οθόνη

i. Τον μέγιστο πληθυσμό, ανεξαρτήτου φύλου και ηλικίας, όλων των δήμων

ii. Τον μέγιστο πληθυσμό, ανεξαρτήτου φύλου και ηλικίας, όλων των δημοτικών ενοτήτων.

iii. Τον μέγιστο πληθυσμό, ανεξαρτήτου φύλου και ηλικίας, όλων των δημοτικών ενοτήτων του δήμου Κομοτηνής

iv. Τίποτα από τα παραπάνω.

Άσκηση

- Έστω ότι εκτελείται το παρακάτω τμήμα κώδικα Python:

```
import pandas as pd
grC = pd.read_csv('GrCensus2011.csv', header=None, sep=';')
```

Το αρχείο GrCensus2011.csv είναι το ίδιο αρχείο που έχει χρησιμοποιηθεί στις εργασίες 1 και 2.

Η έκφραση

```
grCensusData.iloc[6:9, 3].max()
```

θα εμφανίσει στην οθόνη

i. Τον μέγιστο πληθυσμό, ανεξαρτήτου φύλου και ηλικίας, όλων των δήμων

ii. Τον μέγιστο πληθυσμό, ανεξαρτήτου φύλου και ηλικίας, όλων των δημοτικών ενοτήτων.

iii. Τον μέγιστο πληθυσμό, ανεξαρτήτου φύλου και ηλικίας, όλων των δημοτικών ενοτήτων του δήμου Κομοτηνής

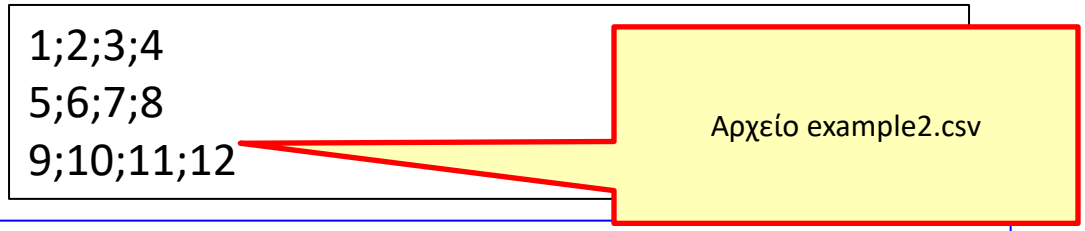
iv. Τίποτα από τα παραπάνω.

Αθροίσματα γραμμών και στηλών στην pandas

- Η συνάρτηση .sum() μπορεί να εφαρμοστεί σε γραμμές ή στήλες για τον υπολογισμό αθροισμάτων ανά γραμμή ή στήλη
  - Δέχεται όρισμα με όνομα δέχεται axis καθορίζει εάν το άθροισμα θα γίνει για κάθε γραμμή (τιμή 1) αθροίζοντας τιμές στηλών στην ίδια γραμμή ή για κάθε στήλη (τιμή 0) αθροίζονται τιμές γραμμών

Αθροίσματα γραμμών και στηλών στην pandas

- .sum(axis=0) : προσθέτει τιμές σε γραμμές υπολογίζοντας άθροισμα στήλης
- .sum(axis=1): προσθέτει τιμές σε στήλες υπολογίζοντας άθροισμα γραμμών




```
>>> import pandas as pd
>>>myData = pd.read_csv('example2.csv', header=None, sep=';')
>>>myData.sum(axis=0)
0  15
1  18
2  21
3  24
dtype: int64
>>>myData.sum(axis=1)
0  10
1  26
2  42
dtype: int64
```

Τέλος Υπο-ενότητας