Python – Εισαγωγή στη βιβλιοθήκη Pandas

Αρχές Γλωσσών Προγραμματισμού και Μεταφραστών
Γιάννης Γαροφαλάκης - Σπύρος Σιούτας

Γ. Γαροφαλάκης, Σ. Σιούτας, Π. Χατζηδούκας

Βιβλιοθήκη Pandas

- Αποτελεί μια βιβλιοθήκη χρήσιμη για την ανάλυση και επεξεργασία δεδομένων.
- Προσφέρει έναν εύκολο τρόπο για να δημιουργήσεις, να τροποποιήσεις και να διαχειριστείς δεδομένα.
- Παρέχει ισχυρές και εύχρηστες δομές δεδομένων, καθώς και τη δυνατότητα εκτέλεσης λειτουργιών πάνω σε αυτές με ταχύτητα και ευκολία.

Πλεονεκτήματα

- Αντιμετωπίζει εύκολα ελλιπή δεδομένα
- Χρησιμοποιεί τη δομή **Series** για μονοδιάστατα δεδομένα και τη **DataFrame** για πολυδιάστατα δεδομένα
- Παρέχει έναν αποδοτικό τρόπο για να τεμαχίζεις (slice) τα δεδομένα
- Προσφέρει έναν ευέλικτο τρόπο για συγχώνευση, συνένωση ή αναδιαμόρφωση των δεδομένων

Δομές Δεδομένων στην Pandas

• Μια **δομή δεδομένων** είναι ένας τρόπος οργάνωσης των δεδομένων, έτσι ώστε να μπορούμε να έχουμε γρήγορη πρόσβαση σε αυτά και να εκτελούμε διάφορες λειτουργίες, όπως ανάκτηση, διαγραφή, τροποποίηση κ.ά.

•Δύο είδη δομών:

- •Σειρές
- •Dataframes

Σειρές

Αποτελούν μια μονοδιάστατη δομή τύπου πίνακα με ομοιογενή δεδομένα, που συνοδεύεται από δείκτες (indexes), επιτρέποντας την εύκολη διαχείριση και επεξεργασία των δεδομένων.

Παράδειγμα

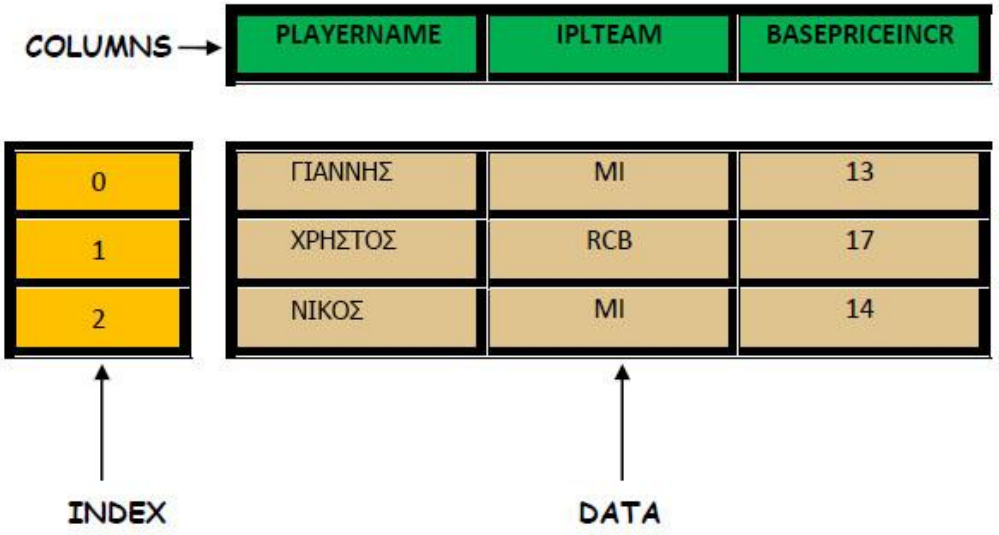
Index	Data
0	10
1	15
2	18
3	22

Dataframes

Αποτελεί μια δισδιάστατη δομή δεδομένων της βιβλιοθήκης Pandas, που οργανώνει τα δεδομένα σε γραμμές και στήλες, παρόμοια με ένα φύλλο Excel ή πίνακα SQL, και επιτρέπει την εύκολη ανάλυση και επεξεργασία τους.

Δομή Dataframes

- Ένα DataFrame έχει άξονες (δείκτες):
  - ➢ Δείκτης γραμμής (axis=0)
  - ➢ Δείκτης στήλης (axis=1)
- Είναι παρόμοιο με ένα φύλλο εργασίας (spreadsheet), όπου ο δείκτης γραμμής ονομάζεται απλά δείκτης (index) και ο δείκτης στήλης ονομάζεται όνομα στήλης (column name).
- Ένα DataFrame μπορεί να περιέχει ετερογενή δεδομένα.
- Το μέγεθος ενός DataFrame μπορεί να είναι μεταβλητό.
- Τα δεδομένα μέσα σε ένα DataFrame είναι επίσης μεταβλητά.



Δημιουργία Dataframe

Ένα dataframe μπορεί να δημιουργηθεί χρησιμοποιώντας οποιοδήποτε από τα παρακάτω:

- Σειρές
- Λίστες
- Λεξικό
- Δισδιάστατο διάνυσμα

```
: import pandas as pd
  df=pd.DataFrame()
  print(df)

  Empty DataFrame
  Columns: []
  Index: []
```

Dataframe από λίστα λεξικών

- Κάθε λεξικό στην λίστα αντιστοιχεί σε μία γραμμή του DataFrame.
- Τα κλειδιά των λεξικών γίνονται τα ονόματα των στηλών του DataFrame.

```
import pandas as pd

# Λίστα λεξικών (κάθε λεξικό είναι μία γραμμή του DataFrame)
data = [
    {'ΟΝΟΜΑΠΑΙΚΤΗ': 'ΓΙΑΝΝΗΣ', 'ΟΜΑΔΑ': 'ΠΑΟ'},
    {'ΟΝΟΜΑΠΑΙΚΤΗ': 'ΝΙΚΟΣ', 'ΟΜΑΔΑ': 'ΟΣΦΠ'},
    {'ΟΝΟΜΑΠΑΙΚΤΗ': 'ΑΝΤΩΝΗΣ', 'ΟΜΑΔΑ': 'ΠΑΟ'}
]

# Δημιουργία DataFrame
df = pd.DataFrame(data)

# Εμφάνιση
print(df)
```

Επανάληψη σε Γραμμές και Στήλες

Αν θέλουμε να αποκτήσουμε πρόσβαση σε εγγραφές ή δεδομένα από ένα DataFrame ανά γραμμή ή ανά στήλη, τότε χρησιμοποιείται επανάληψη (iteration).
Η βιβλιοθήκη Pandas παρέχει 2 συναρτήσεις για να εκτελέσουμε επαναλήψεις:

- iterrows()
- iteritems()

Iterrows()

Χρησιμοποιείται για την προσπέλαση των δεδομένων ανά γραμμή.

Κώδικας

```
import pandas as pd

# Λίστα λεξικών (κάθε λεξικό είναι μία γραμμή του DataFrame)
data = [
    {'ΟΝΟΜΑΠΑΙΚΤΗ': 'ΓΙΑΝΝΗΣ', 'ΟΜΑΔΑ': 'ΠΑΟ'},
    {'ΟΝΟΜΑΠΑΙΚΤΗ': 'ΝΙΚΟΣ', 'ΟΜΑΔΑ': 'ΟΣΦΠ'}
]

# Δημιουργία DataFrame
df = pd.DataFrame(data)
print(df)

# Χρήση iterrows
for row_index, row_value in df.iterrows():
    print('\n Δείκτης γραμμής είναι ::', row_index)
    print('Τιμή γραμμής είναι ::')
    print(row_value)
```

Έξοδος

```
  ΟΝΟΜΑΠΑΙΚΤΗ  ΟΜΑΔΑ
0    ΓΙΑΝΝΗΣ   ΠΑΟ
1      ΝΙΚΟΣ  ΟΣΦΠ

 Δείκτης γραμμής είναι :: 0
Τιμή γραμμής είναι ::
ΟΝΟΜΑΠΑΙΚΤΗ    ΓΙΑΝΝΗΣ
ΟΜΑΔΑ             ΠΑΟ
Name: 0, dtype: object

 Δείκτης γραμμής είναι :: 1
Τιμή γραμμής είναι ::
ΟΝΟΜΑΠΑΙΚΤΗ     ΝΙΚΟΣ
ΟΜΑΔΑ            ΟΣΦΠ
Name: 1, dtype: object
```

iteritems()

Χρησιμοποιείται για την προσπέλαση των δεδομένων ανά στήλη.

Κώδικας

```
import pandas as pd

# Λίστα λεξικών (κάθε λεξικό είναι μία γραμμή του DataFrame)
data = [
    {'ΟΝΟΜΑΠΑΙΚΤΗ': 'ΓΙΑΝΝΗΣ', 'ΟΜΑΔΑ': 'ΠΑΟ'},
    {'ΟΝΟΜΑΠΑΙΚΤΗ': 'ΝΙΚΟΣ', 'ΟΜΑΔΑ': 'ΟΣΦΠ'}
]

# Δημιουργία DataFrame
df = pd.DataFrame(data)
print(df)

# Χρήση iteritems για επανάληψη σε στήλες
for column_name, column_data in df.iteritems():
    print('\nΌνομα στήλης ::', column_name)
    print('Τιμές στήλης ::')
    print(column_data)
```

Έξοδος

```
  ΟΝΟΜΑΠΑΙΚΤΗ  ΟΜΑΔΑ
0    ΓΙΑΝΝΗΣ   ΠΑΟ
1      ΝΙΚΟΣ  ΟΣΦΠ

Όνομα στήλης :: ΟΝΟΜΑΠΑΙΚΤΗ
Τιμές στήλης ::
0    ΓΙΑΝΝΗΣ
1      ΝΙΚΟΣ
Name: ΟΝΟΜΑΠΑΙΚΤΗ, dtype: object

Όνομα στήλης :: ΟΜΑΔΑ
Τιμές στήλης ::
0     ΠΑΟ
1    ΟΣΦΠ
Name: ΟΜΑΔΑ, dtype: object
```

Επιλογή δεδομένων σε DataFrame

Για να προσπελάσουμε τα δεδομένα μιας στήλης, μπορούμε να αναφέρουμε το όνομα της στήλης ως δείκτη (subscript).

Παράδειγμα

- df[Κωδικός] Αυτό μπορεί επίσης να γίνει και με τη μορφή df.Κωδικός

Για να προσπελάσουμε πολλές στήλες, μπορούμε να γράψουμε:

df[[col1, col2, ---]]

Παράδειγμα Κώδικα 1/2

Κώδικας

```python
import pandas as pd

# Λίστα δεδομένων υπαλλήλων
dedomena = {
    'ΚΩΔΙΚΟΣ': [101, 102, 103, 104, 105, 106],
    'ONOMA': ['Γιάννης', 'Μαρία', 'Κώστας', 'Ελένη', 'Νίκος', 'Άννα'],
    'ΗΜΕΡΟΜΗΝΙΑ_ΕΝΑΡΞΗΣ': ['12-01-2012', '15-01-2012', '05-09-2007',
                           '17-01-2012', '05-09-2007', '16-01-2012']
}

# Δημιουργία του DataFrame
df = pd.DataFrame(dedomena)

# Εμφάνιση του DataFrame
print(df)

# Πρόσβαση στα δεδομένα της στήλης 'ΚΩΔΙΚΟΣ'
print("\nΠρόσβαση στη στήλη 'ΚΩΔΙΚΟΣ' με δύο τρόπους:")
print(df['ΚΩΔΙΚΟΣ'])      # Μέθοδος με subscript
print(df.ΚΩΔΙΚΟΣ)         # Μέθοδος με dot notation (αν το όνομα δεν έχει κενά ή σύμβολα)
```

Έξοδος

```
   ΚΩΔΙΚΟΣ     ONOMA ΗΜΕΡΟΜΗΝΙΑ_ΕΝΑΡΞΗΣ
0      101   Γιάννης          12-01-2012
1      102     Μαρία          15-01-2012
2      103    Κώστας          05-09-2007
3      104     Ελένη          17-01-2012
4      105     Νίκος          05-09-2007
5      106      Άννα          16-01-2012
```

```
0    101
1    102
2    103
3    104
4    105
5    106
Name: ΚΩΔΙΚΟΣ, dtype: int64
```

```
0    101
1    102
2    103
3    104
4    105
5    106
Name: ΚΩΔΙΚΟΣ, dtype: int64
```

Παράδειγμα Κώδικα 1/2

```
print(df[['ΚΩΔΙΚΟΣ', 'ΟΝΟΜΑ']])
```

```
   ΚΩΔΙΚΟΣ     ΟΝΟΜΑ
0      101   Γιάννης
1      102     Μαρία
2      103    Κώστας
3      104     Ελένη
4      105     Νίκος
5      106      Άννα
```

Πρόσθεση και μετονομασία στήλης Dataframe

Κώδικας

```python
import pandas as pd

# Δημιουργία Series
s = pd.Series([10, 15, 18, 22])

# Δημιουργία DataFrame από το Series
df = pd.DataFrame(s)

# Μετονομασία της προεπιλεγμένης στήλης σε 'List1'
df.columns = ['List1']  # Rename the default column of DataFrame as List1

# Δημιουργία νέας στήλης 'List2' με όλες τις τιμές 20
df['List2'] = 20  # Create a new column List2 with all values as 20

# Δημιουργία νέας στήλης 'List3' ως το άθροισμα των List1 και List2
df['List3'] = df['List1'] + df['List2']  # Add List1 and List2 and store in List3

# Εμφάνιση του DataFrame
print(df)
```

Έξοδος

```
   List1  List2  List3
0     10     20     30
1     15     20     35
2     18     20     38
3     22     20     42
```

Διαγραφή στήλης Dataframe

Η διαγραφή μιας στήλης από ένα dataframe πραγματοποιείται με οποιοδήποτε από τους παρακάτω τρόπους:

- del
- pop()
- drop()

Εντολή del

Μπορούμε απλά να διαγράψουμε μια στήλη περνώντας το όνομα της στήλης ως δείκτη στο df

Εντολή

```
del df['List3']
print(df)
```

Έξοδος

```
   List1  List2
0     10     20
1     15     20
2     18     20
3     22     20
```

Εντολή pop()

Μπορούμε απλά να διαγράψουμε μια στήλη περνώντας το όνομά της μέσα στη μέθοδο pop.

Εντολή

```
df.pop('List2')
print(df)
```

Έξοδος

```
   List1
0     10
1     15
2     18
3     22
```

Εντολή drop()

Η μέθοδος drop() χρησιμοποιείται για τη διαγραφή στηλών ή γραμμών από ένα DataFrame, ορίζοντας axis=1 για στήλες και axis=0 για γραμμές.

Κώδικας

```
import pandas as pd

s = pd.Series([10, 20, 30, 40])
df = pd.DataFrame(s)
df.columns = ['List1']

df['List2'] = 40

# Διαγραφή στήλης List2
df1 = df.drop('List2', axis=1)  # (axis=1) σημαίνει διαγραφή στήλης

# Διαγραφή γραμμών με index 2 και 3
df2 = df.drop(index=[2, 3], axis=0)  # (axis=0) σημαίνει διαγραφή γραμμών

print(df)
print("After deletion::")
print(df1)
print("After row deletion::")
print(df2)
```

Έξοδος

```
   List1  List2
0     10     40
1     20     40
2     30     40
3     40     40

After deletion::
   List1
0     10
1     20
2     30
3     40

After row deletion::
   List1  List2
0     10     40
1     20     40
```

Μέθοδος head() και tail()

Η μέθοδος head() εμφανίζει τις πρώτες 5 γραμμές και η μέθοδος tail() επιστρέφει τις τελευταίες 5 γραμμές.

Κώδικας

```python
import pandas as pd

# Δημιουργία DataFrame
data = {
    'Όνομα': ['Γιάννης', 'Μαρία', 'Κώστας', 'Ελένη', 'Άννα', 'Νίκος'],
    'Ηλικία': [25, 30, 22, 28, 24, 35]
}
df = pd.DataFrame(data)

# Πρώτες 5 γραμμές
print("Πρώτες 5 γραμμές:")
print(df.head())

# Τελευταίες 5 γραμμές
print("\nΤελευταίες 5 γραμμές:")
print(df.tail())
```

Έξοδος

```
Πρώτες 5 γραμμές:
     Όνομα  Ηλικία
0  Γιάννης      25
1    Μαρία      30
2   Κώστας      22
3    Ελένη      28
4     Άννα      24

Τελευταίες 5 γραμμές:
     Όνομα  Ηλικία
1    Μαρία      30
2   Κώστας      22
3    Ελένη      28
4     Άννα      24
5    Νίκος      35
```

Συγχώνευση Dataframe

Δύο DataFrame μπορεί να περιέχουν διαφορετικά είδη πληροφοριών για την ίδια οντότητα και να συνδέονται μέσω κάποιου κοινού χαρακτηριστικού/στήλης. Για να συνδυάσουμε αυτά τα DataFrame, η βιβλιοθήκη pandas παρέχει διάφορες συναρτήσεις όπως merge.

Παράδειγμα Συγχώνευσης

Κώδικας

```python
# Δημιουργία λεξικών με κοινή στήλη id
import pandas as pd

dicA = {
    'id': ['1', '2', '3'],
    'Όνομα': ['Γιώργος', 'Μαρία', 'Νίκος'],
    'Βαθμός': [85, 90, 78]
}

dicB = {
    'id': ['2', '3', '4'],
    'Μάθημα': ['Μαθηματικά', 'Φυσική', 'Χημεία'],
    'Έτος': [2023, 2023, 2023]
}

# Δημιουργία DataFrames
dfA = pd.DataFrame(dicA)
dfB = pd.DataFrame(dicB)

# Συγχώνευση των δύο πινάκων βάσει του 'id'
df_merged = pd.merge(dfA, dfB, on='id')

# Εκτύπωση αποτελέσματος
print(df_merged)
```

Έξοδος

```
  id   Όνομα  Βαθμός       Μάθημα  Έτος
0  2  Μαρία      90  Μαθηματικά  2023
1  3   Νίκος      78      Φυσική  2023
```

Συγχώνευση DataFrames με Διαφορετικά Ονόματα Στηλών

Μπορεί να συμβεί η στήλη με βάση την οποία θέλεις να συγχωνεύσεις τα DataFrame να έχει διαφορετικά ονόματα (σε αντίθεση με την τρέχουσα περίπτωση). Για τέτοιες συγχωνεύσεις, θα πρέπει να καθορίσεις τα ορίσματα left_on ως το όνομα της στήλης του αριστερού DataFrame και right_on ως το όνομα της στήλης του δεξιού DataFrame.

Παράδειγμα Συγχώνευση DataFrames με Διαφορετικά Ονόματα Στηλών

Κώδικας

```
import pandas as pd

dic1 = {
    'id': ['1', '2', '3', '4', '5'],
    'Value1': ['A', 'C', 'E', 'G', 'I'],
    'Value2': ['B', 'D', 'F', 'H', 'J']
}

dic2 = {
    'id': ['2', '3', '6', '7', '8'],
    'Value1': ['K', 'M', 'O', 'Q', 'S'],
    'Value2': ['L', 'N', 'P', 'R', 'T']
}

dic3 = {
    'id': ['1', '2', '3', '4', '5', '7', '8', '9', '10', '11'],
    'Value3': [12, 13, 14, 15, 16, 17, 15, 12, 13, 23]
}

df1 = pd.DataFrame(dic1)
df2 = pd.DataFrame(dic2)
df3 = pd.concat([df1, df2])
df4 = pd.DataFrame(dic3)
df5 = pd.merge(df3, df4, left_on='id', right_on='id')

print(df5)
```

Εντολή

```
  id Value1 Value2  Value3
0  1      A      B      12
1  2      C      D      13
2  2      K      L      13
3  3      E      F      14
4  3      M      N      14
5  4      G      H      15
6  5      I      J      16
7  7      Q      R      17
8  8      S      T      15
```

Ανάγνωση CSV

- df = pd.read_csv(data.csv')
- print(df.head())

// Εισαγωγή δεδομένων από αρχείο CSV.