

PYTHON İLE VERİ ANALİZİ

Hafta 03

PYTHON İLE VERİ ANALİZİ

- Python Kütüphanelerinin Yüklenmesi
- «Package Management System», kütüphaneleri yüklemek için kullanılır.
- Anaconda Prompt'tan **pip** komutu ile kütüphaneler aktif hale getirilir.
- PIP açık hali ile:
- **Pip Installs Python** veya
- **Pip Installs Packages**

PYTHON İLE VERİ ANALİZİ

- Colab Notebook veya Jupyter Notebook kullanımında, "!" operatörü, paketleri veya kütüphaneleri yüklemek için kullanılır.
- ! Operatorü, Jupyter Notebook'a, «!» Operatörü ile başlayan satırın bir Python Kodu olmadığını belirtir.
- Sadece «command line script» olduğu belirtilir.
- "!pip install keras" «command line script» ile keras kütüphanesi yüklenir.

PYTHON İLE VERİ ANALİZİ

- Keras kütüphanesi, bir makine öğrenme platformu olan TensorFlow üzerinde çalışan, Python Programlama dilinde kodlanmış bir derin öğrenme API'sidir.

PYTHON İLE VERİ ANALİZİ

- Python Veri Yapıları:
- Liste(List)
- Sözlük(Dictionary)
- Demet(Tuple)
- Küme(Set)

PYTHON İLE VERİ ANALİZİ

- Listeler(LISTS)

PYTHON İLE VERİ ANALİZİ

- Listeler(LIST), verileri saklamak için kullanılan bir container türüdür.
- Bir liste, birden fazla veri içeren bir nesnedir.
- Listeler değiştirilebilir, yani içerikleri bir program çalıştırılırken değiştirilebilir.
- Listeler **dinamik veri yapılarıdır**, yani veriler listeye eklenebilir veya listeden çıkarılabilir.
- Bir programda listeler ile çalışmak için indeks oluşturma(indexing), dilimleme(slicing) metotları kullanılabilir.

PYTHON İLE VERİ ANALİZİ

- Bir liste, köşeli parantez [] ile gösterilir.
- Listede saklanan her veriye bir **eleman**(element) denir.

PYTHON İLE VERİ ANALİZİ

- Listenin tanımlanması:
- `say1 = [2, 4, 6, 5, 10, 11]`
- `# say1 listesinin elemanları`

PYTHON İLE VERİ ANALİZİ

- #Liste Örnekleri
- `sayı = [5, 40, 15, 20, 35, 50]`
- `print (sayı)`
- Ekran Çıktısı:
- `[5, 40, 15, 20, 35, 50]`

PYTHON İLE VERİ ANALİZİ

- #Liste Örnekleri
- `sayı = [5, 40, 15, 20, 35, 50]`
- `print(sayı)`

- `sayılar = [0] * 10`
- `print (sayılar)`

PYTHON İLE VERİ ANALİZİ

- Ekran Çıktısı:
- [5, 40, 15, 20, 35, 50]
- [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

PYTHON İLE VERİ ANALİZİ

- `#range(stop)`
- `for i in range(3):`
- `print(i)`

PYTHON İLE VERİ ANALİZİ

- Ekran Çıktısı:
- 0
- 1
- 2

PYTHON İLE VERİ ANALİZİ

- `#range(start, stop)`
- `for i in range(1, 8):`
- `print(i)`

PYTHON İLE VERİ ANALİZİ

- Ekran Çıktısı:

- 1
- 2
- 3
- 4
- 5
- 6
- 7

PYTHON İLE VERİ ANALİZİ

- `#range(start, stop, step)`
- `for i in range(3, 16, 3):`
- `print(i)`

PYTHON İLE VERİ ANALİZİ

- Ekran Çıktısı:
- 3
- 6
- 9
- 12
- 15

PYTHON İLE VERİ ANALİZİ

- #liste elemanlarının iterasyonu
- sayı = [19, 100, 21, 76, 35, 67]
- for n in sayı:
- print (n)

PYTHON İLE VERİ ANALİZİ

- Ekran Çıktısı:
- 19
- 100
- 21
- 76
- 35
- 67

PYTHON İLE VERİ ANALİZİ

- #liste elemanlarının iterasyonu
- `sayı = [19, 100, 21, 76, 35, 67]`
- `for sayaç in sayı:`
- `print (sayaç)`

PYTHON İLE VERİ ANALİZİ

- Ekran Çıktısı:
- 19
- 100
- 21
- 76
- 35
- 67

PYTHON İLE VERİ ANALİZİ

- `liste = ["Ankara", "İstanbul", "Bursa", "İzmir", "Tekirdağ"]`
- `print(liste)` # bütün listenin ekranda gösterilmesi
- `print(liste[0])` # listedeki birinci elemanın(indeks sıfır) ekranda gösterilmesi
- `print(liste[-1])` # listedeki sonuncu elemanın(indeks -1) ekranda gösterilmesi
- `print(liste[-3])` # listenin sonundan -3. elemanın(indeks -3) ekranda gösterilmesi

PYTHON İLE VERİ ANALİZİ

- Ekran Çıktısı:
- ['Ankara', 'İstanbul', 'Bursa', 'İzmir', 'Tekirdağ']
- Ankara
- Tekirdağ
- Bursa

PYTHON İLE VERİ ANALİZİ

- `diller = ['Spanish', 'English', 'French', 'German', 'Irish', 'Chinese']`
- `# len listedeki eleman sayısını hesaplar`
- `for indeks in range(len(diller)):`
- `print('Dil Listesi : ', diller[indeks])`

PYTHON İLE VERİ ANALİZİ

- Ekran Çıktısı:
- Dil Listesi : Spanish
- Dil Listesi : English
- Dil Listesi : French
- Dil Listesi : German
- Dil Listesi : Irish
- Dil Listesi : Chinese

PYTHON İLE VERİ ANALİZİ

- `diller = ['Spanish', 'English', 'French', 'German', 'Irish', 'Chinese']`
- `for indeks in range(6):`
- `print('Dil:', diller[indeks])`

PYTHON İLE VERİ ANALİZİ

- Dil Listesi : Spanish
- Dil Listesi : English
- Dil Listesi : French
- Dil Listesi : German
- Dil Listesi : Irish
- Dil Listesi : Chinese

PYTHON İLE VERİ ANALİZİ

- `say1 = [19, 100, 21, 76, 35, 67]`
- `indeks = 0`
- `while (indeks < 6):`
 - `print (say1[indeks])`
 - `indeks += 1`

PYTHON İLE VERİ ANALİZİ

- Programın Ekran Çıktısı:

- 19
- 100
- 21
- 76
- 35
- 67

PYTHON İLE VERİ ANALİZİ

- `say1 = [19, 100, 21, 76, 35, 67]`
- `indeks = -6`
- `while (indeks < 0):`
 - `print (say1[indeks])`
 - `indeks += 1`

PYTHON İLE VERİ ANALİZİ

- Programın Ekran Çıktısı:

- 19
- 100
- 21
- 76
- 35
- 67

PYTHON İLE VERİ ANALİZİ

- `sicaklik_degerleri = [19, 10, 13, 12, 11, 9, 8]`
- `hafta_ici = sicaklik_degerleri[0:5]`
- `print (sicaklik_degerleri)`
- `print (hafta_ici)`

PYTHON İLE VERİ ANALİZİ

- Programın Ekran Çıktısı:
- [19, 10, 13, 12, 11, 9, 8]
- [19, 10, 13, 12, 11]

PYTHON İLE VERİ ANALİZİ

- #SLICING – Dilimleme İşlemi
- `sicaklik_degerleri = [19, 10, 13, 12, 11, 9, 8]`
- # bütün liste, sondan iki eleman hariç
- `hafta_ici = sıcaklik_degerleri[-7:-2]`
- `print (sicaklik_degerleri)`
- `print (hafta_ici)`

PYTHON İLE VERİ ANALİZİ

- Programın Ekran Çıktısı:
- [19, 10, 13, 12, 11, 9, 8]
- [19, 10, 13, 12, 11]

PYTHON İLE VERİ ANALİZİ

- #SLICING - Dilimleme
- `hafta_ici=sicaklik_degerleri[-1]` # listedeki son eleman
- `print (hafta_ici)`
- `hafta_ici=sicaklik_degerleri[-2:]` # listedeki son iki eleman
- `print (hafta_ici)`
- `hafta_ici=sicaklik_degerleri[:-2]` # son iki eleman hariç bütün elemanlar
- `print (hafta_ici)`

PYTHON İLE VERİ ANALİZİ

- Programın Ekran Çıktısı:
- 8
- [9, 8]
- [19, 10, 13, 12, 11]

PYTHON İLE VERİ ANALİZİ

- #SLICING – Dilimleme
-
- `hafta_ici=sicaklik_degerleri[::-1]` # Sondan başlayarak bütün elemanlar
- `print (hafta_ici)`
- `hafta_ici= sicaklik_degerleri[1::-1]` # Sondan başlayarak ilk iki eleman
- `print (hafta_ici)`
- `hafta_ici= sicaklik_degerleri[:-3:-1]` # Sondan başlayarak son iki eleman
- `print (hafta_ici)`
- `hafta_ici= sicaklik_degerleri[-3::-1]` # Sondan başlayarak son iki eleman hariç
- `print (hafta_ici)`

PYTHON İLE VERİ ANALİZİ

- Programın Ekran Çıktısı:
- [8, 9, 11, 12, 13, 10, 19]
- [10, 19]
- [8, 9]
- [11, 12, 13, 10, 19]

PYTHON İLE VERİ ANALİZİ

- #SLICING - Dilimleme
- `sicaklik_degerleri = [19, 10, 13, 12, 11, 9, 8]`
- `ilk__iki = sıcaklik_degerleri[-7:-5]` #bütün liste
- #sondan 5 eleman hariç
- `print (sicaklik_degerleri)`
- `print (ilk__iki)`

PYTHON İLE VERİ ANALİZİ

- Programın Ekran Çıktısı:
- [19, 10, 13, 12, 11, 9, 8]
- [19, 10]

PYTHON İLE VERİ ANALİZİ

- `def main():`
- `# liste oluşturma`
- `sicaklik = [15.0, 10.5, 13.5, 14.0, 12.5]`
- `# toplam değişkeni başlangıç değeri`
- `toplam = 0.0`
- `# liste elemanlarının toplamı.`
- `for deger in sicaklik:`
- `toplam += deger`
- `# ortalama değer hesabı`
- `ortalama = toplam / len(sicaklik)`

PYTHON İLE VERİ ANALİZİ

-
- `# ekran çıktısı`
- `print ("ortalama sıcaklık değeri", ortalama)`
- `print (sicaklik)`
- `print (toplam)`
- `# main fonksiyon`
- `main()`

PYTHON İLE VERİ ANALİZİ

- Ekran Çıktısı:
- ortalama sıcaklık değeri 13.1
- [15.0, 10.5, 13.5, 14.0, 12.5]
- 65.5

PYTHON İLE VERİ ANALİZİ

- `def main():`
- `# Liste`
- `paket_isimleri = ['AX', 'GP', 'NAV', 'SL', 'CRM']`
- `# ekran çıktısı`
- `print('Liste elemanları:')`
- `print(paket_isimleri)`
- `# sıfıncı elemanın yerine veri ilavesi`
- `paket_isimleri.insert(0, 'SAP')`
- `# ekran çıktısı`
- `print('listede veri değiştirme:')`

PYTHON İLE VERİ ANALİZİ

-
- `paket_isimleri.insert(4, 'Peoplesoft')`
- `print('Listeye veri ilave:')`
- `print(paket_isimleri)`
- `# Fonksiyon çağırma`
- `main()`

PYTHON İLE VERİ ANALİZİ

- Liste elemanları:
- ['AX', 'GP', 'NAV', 'SL', 'CRM']
- listede veri değiştirme:
- ['SAP', 'AX', 'GP', 'NAV', 'SL', 'CRM']
- Listeye veri ilave:
- ['SAP', 'AX', 'GP', 'NAV', 'Peoplesoft', 'SL', 'CRM']

PYTHON İLE VERİ ANALİZİ

- `def main():`
- `# Liste`
- `paket_isimleri = ['AX', 'GP', 'NAV', 'SL', 'CRM']`
- `# ekran çıktısı`
- `print('Listedeki veriler:')`
- `print(paket_isimleri)`
- `# sıfırncı elemanın silinmesi.`
- `paket_isimleri.remove('AX')`
- `print ('ilk eleman listeden silindi')`
- `print(paket_isimleri)`

PYTHON İLE VERİ ANALİZİ

- `# ekran çıktısı`
- `print('listedeki son elemanın silinmesi:')`
- `print(paket_isimleri)`
- `paket_isimleri.pop()`
- `print('Listeye veri silme:')`
- `print(paket_isimleri)`

- `# Fonksiyon çağırma`
- `main()`

PYTHON İLE VERİ ANALİZİ

- Listedeki veriler:
- ['AX', 'GP', 'NAV', 'SL', 'CRM']
- ilk eleman listeden silindi
- ['GP', 'NAV', 'SL', 'CRM']
- listedeki son elemanın silinmesi:
- ['GP', 'NAV', 'SL', 'CRM']
- Listeye veri silme:
- ['GP', 'NAV', 'SL']

PYTHON İLE VERİ ANALİZİ

- `otomobil = ['bmw', 'audi', 'toyota', 'subaru', 'fiat']`
- `print(" İlk Liste:")`
- `print(otomobil)`
- `print("\n Sıralı liste:")`
- `print(sorted(otomobil))`
- `print("\n Z'den A'ya sıralı liste:")`
- `print(sorted(otomobil, reverse=True))`
- `print("\n İlk Liste:")`
- `print(otomobil)`

PYTHON İLE VERİ ANALİZİ

- Ekran Çıktısı:
- İlk Liste:
['bmw', 'audi', 'toyota', 'subaru', 'fiat']
- Sıralı liste:
['audi', 'bmw', 'fiat', 'subaru', 'toyota']
- Zden Aya sıralı liste:
['toyota', 'subaru', 'fiat', 'bmw', 'audi']
- İlk Liste:
['bmw', 'audi', 'toyota', 'subaru', 'fiat']

PYTHON İLE VERİ ANALİZİ

- `karma__liste = ['bmw', 'audi', 'toyota', 'subaru', 'fiat', 'AX', 'Dolibarr', 'Odoo', 'ERP5', 'ADempiere']`
- `print("İlk Liste:")`
- `print(karma__liste)`
- `print("\n Sıralı liste:")`
- `print(sorted(karma__liste))`
- `print("\n Z den A ya sıralı liste:")`
- `print(sorted(karma__liste, reverse=True))`
- `print("\n İlk Liste :")`
- `print(karma__liste)`

PYTHON İLE VERİ ANALİZİ

- **Ekran Çıktısı:**
- İlk Liste:
- ['bmw', 'audi', 'toyota', 'subaru', 'fiat', 'AX', 'Dolibarr', 'Odoo', 'ERP5', 'ADempiere']
- Sıralı liste:
- ['ADempiere', 'AX', 'Dolibarr', 'ERP5', 'Odoo', 'audi', 'bmw', 'fiat', 'subaru', 'toyota']
- Z den A ya sıralı liste:
- ['toyota', 'subaru', 'fiat', 'bmw', 'audi', 'Odoo', 'ERP5', 'Dolibarr', 'AX', 'ADempiere']
- İlk Liste :
- ['bmw', 'audi', 'toyota', 'subaru', 'fiat', 'AX', 'Dolibarr', 'Odoo', 'ERP5', 'ADempiere']

PYTHON İLE VERİ ANALİZİ

- `sayi = [19, 100, 21, 76, 35, 67]`
- `index = 0`
- `while (index < 6):`
- `print (sayi[index])`
- `index += 1`
- `print("\n")`
- `index = -6`
- `while (index < 0):`
- `print (sayi[index])`
- `index += 1`

PYTHON İLE VERİ ANALİZİ

- Ekran Çıktısı

- 19
- 100
- 21
- 76
- 35
- 67

- 19
- 100
- 21
- 76
- 35
- 67

PYTHON İLE VERİ ANALİZİ

- `sicaklik_degerleri = [19, 10, 13, 12, 11, 9, 8]`
- `hafta_ici = sicaklik_degerleri[0:5]`
- `print (sicaklik_degerleri)`
- `print (hafta_ici)`

PYTHON İLE VERİ ANALİZİ

- Ekran Çıktısı
- [19, 10, 13, 12, 11, 9, 8]
- [19, 10, 13, 12, 11]

PYTHON İLE VERİ ANALİZİ

- pandas, panel veriden (ekonometrik bir terim) ve Python veri analizinden ismini almıştır.
- pandas açık kaynak Python kütüphanesidir.
- Dersin kapsamında temel pandas fonksiyonları, veri yapıları ve işlemleri hakkında konular işlenecektir.

PYTHON İLE VERİ ANALİZİ

- Konular:
- pandas Series
- pandas DataFrame
- Veri temizleme-munging-cleaning-wrangling-cleansing
- pandas Verileri Sorgulanması
- pandas DataFrame ile veri toplama
- DataFrame ekleme ve birleştirme
- Kayıp(missing) veriler ile ilgili işlemler

PYTHON İLE VERİ ANALİZİ

- #import deyimi
- `import pandas as pd` #pandas kütüphanesi kullanımı

PYTHON İLE VERİ ANALİZİ

Kaynaklar:

https://www.w3schools.com/python/python_lists.asp

<http://www.i-programmer.info/programming/python/3942-arrays-in-python.html>

Beginning Programming with Python, John Paul Mueller, 2014.

https://github.com/Musicachic/CITP_110/tree/master/CITP%20110/Source%20Code/Gaddis_Python_2E_Source_Code/Chapter%2008

Starting Out with Python, Tony Gaddis, 2009.

<https://docs.python.org/2.3/whatsnew/section-slices.html>

<https://www.digitalocean.com/community/tutorials/understanding-lists-in-python-3>

https://www.w3schools.com/python/python_pip.asp

<https://stackoverflow.com/questions/509211/understanding-slice-notation>

<https://www.dataquest.io/blog/tutorial-advanced-for-loops-python-pandas/>