PYTHON ile VERİ ANALİZİ

Hafta 04

PYTHON ile VERİ ANALİZİ

- TUPLE Veri Yapısı

- PYTHON Programlama Dili Veri Yapılarından birisi olan tuple, integer, float, string, list ve dictionary gibi değişik veri türlerini depolayan bir dizidir.

PYTHON ile VERİ ANALİZİ

- Tuple Veri Yapısı Formatı:
- Veri elemanları parantez içinde gösterilir.
- Veri içeren bir tuple, parantez içinde veriler ile ifade edilir.
- Örnek:
- tuple001 = (100)

PYTHON ile VERİ ANALİZİ

- tuple01=() # boş bir tuple
- a = (100)
- print (a)

- Ekran Çıktısı:
- 100

PYTHON ile VERİ ANALİZİ

- Boş bir tuple:
- Tupleo1=()

PYTHON ile VERİ ANALİZİ

- # İndeks ve dilimleme işlemleri
- tuple002 = (100, 200, 300, 350)

- print (tuple002)
- print (tuple002[-1])
- print (tuple002[0:1])
- print (tuple002[0:3])
- print (tuple002[2])

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:
- (100, 200, 300, 350)
- 350
- (100,)
- (100, 200, 300)
- 300

PYTHON ile VERİ ANALİZİ

- tuple002 = (100, 200, 300, 350)

- print (tuple002)
- print (tuple002[-1])#son eleman
- print (tuple002[0:2]) #indeks o'dan itibaren iki eleman
- print (tuple002[2]) #indeks numarası 2 olan eleman

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:

- (100, 200, 300, 350)
- 350
- (100, 200)
- 300

PYTHON ile VERİ ANALİZİ

- #tuple'a yeni eleman ilave edilmesi
- tuple003 = (100, 200, 300, 400)
- print (tuple003)
- yeniTuple = tuple003
- yeniTuple += (500,)
- tuple003 += (5000,)
- print (yeniTuple)
- print (tuple003)

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:
- (100, 200, 300, 400)
- (100, 200, 300, 400, 500)
- (100, 200, 300, 400, 5000)

PYTHON ile VERİ ANALİZİ

- İndeks Oluşturulması:
- Belli bir tuple elemanına erişebilmek için parantez içindeki indeks numarası belirtilir.
- Pozitif ve negatif indeksleme yapılabilir.

PYTHON ile VERİ ANALİZİ

- sicaklik_degeri = (10, 13, 14, 9, 8, 16, 11)
- print (sicaklik_degeri[0])
- print (sicaklik_degeri[2])

- print (sicaklik_degeri[-1])
- print (sicaklik_degeri[-6])

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:
- 10
- 14
- 11
- 13

PYTHON ile VERİ ANALİZİ

- #Tuple Sıralama(Ascending Order Sorting - Küçükten büyüğe sıralama)

- sicaklik_degeri = (10, 13, 14, 9, 8, 16, 11)
- print (sicaklik_degeri, "\n")
- sicaklik_degeri = sorted(sicaklik_degeri)
- print (sicaklik_degeri)

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:
- (10, 13, 14, 9, 8, 16, 11)

- [8, 9, 10, 11, 13, 14, 16]

PYTHON ile VERİ ANALİZİ

- #Tuple Büyükten Küçüğe Sıralama (Descending Order -- Sorting)

- sicaklik_degeri = (10, 13, 14, 9, 8, 16, 11)
- print ("orijinal Tuple : ", sicaklik_degeri, "\n")
- sicaklik_degeri = sorted(sicaklik_degeri, reverse = True)
- print (sicaklik_degeri)

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:
- orjinal Tuple : (10, 13, 14, 9, 8, 16, 11)

- [16, 14, 13, 11, 10, 9, 8]

PYTHON ile VERİ ANALİZİ

- liste= [10, 30, 50]
- tuple=(60, 65, 70)
- print (liste)
- print (tuple)
- #veri elemanı değişikliği
- liste[1]=100
- #tuple[1]=300 #TypeError: 'tuple' object does not support item assignment
- print (liste)
- print (tuple)

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:
- [10, 30, 50]
- (60, 65, 70)
- [10, 100, 50]
- (60, 65, 70)

PYTHON ile VERİ ANALİZİ

- liste= [10, 30, 50]
- tuple=(60,65,70)
- print (liste)
- print (tuple)
- #veri elemanı değişikliği
- liste[1]=100
- #tuple[1] elemanın değerinin 300 olarak değiştirilmesi
- n=1
- tuple= tuple[: n] + (300 ,) + tuple[n + 1 :]
- print (liste)
- print (tuple)

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:
- [10, 30, 50]
- (60, 65, 70)
- [10, 100, 50]
- (60, 300, 70)

PYTHON ile VERİ ANALİZİ

- tuple = (40, 50, 60)
- print (tuple)

- Ekran Çıktısı:
- (40, 50, 60)

PYTHON ile VERİ ANALİZİ

- tuple = 40, 50, 60
- print (tuple)

- Ekran Çıktısı:
- (40, 50, 60)

PYTHON ile VERİ ANALİZİ

- # İçiçe(Nested) Tuple oluşturulması
- nested_tuple = (4, 5, 6), (7, 8, 9, 10)
- print (nested_tuple)

- Ekran Çıktısı:
- ((4, 5, 6), (7, 8, 9, 10))

PYTHON ile VERİ ANALİZİ

- tuple=('Sakarya', 'Üniversitesi') * 4
- print (tuple)

- Ekran Çıktısı:
- ('Sakarya', 'Üniversitesi', 'Sakarya', 'Üniversitesi', 'Sakarya', 'Üniversitesi', 'Sakarya', 'Üniversitesi')

PYTHON ile VERİ ANALİZİ

- tuple = (1, 2, 2, 2, 3, 4, 2, 3, 6, 8, 2)
- print (tuple.count(2)) # 2 sayısının kaç adet olduğu

- Ekran Çıktısı:
- 5

PYTHON ile VERİ ANALİZİ

- Dictionary Veri Yapısı

- Veriler Key – Anahtar ile indekslenmiştir.

PYTHON ile VERİ ANALİZİ

- Dict veri yapısının anahtar - değer kısım çifti, birlikte item olarak adlandırılır.
- Anahtar ve değer, iki nokta üst üste (:) ile ayrılır ve her bir item ise virgül (,) ile ayrılmıştır.
- Öğeler süslü parantez(curly braces) {} içindedir.
- Boş bir dict, yalnızca süslü parantez {} ile oluşturulabilir.

PYTHON ile VERİ ANALİZİ

- port = {22: "SSH", 23: "Telnet" , 53: "DNS", 80: "HTTP" }
- print (port)

- Ekran Çıktısı:
- {80: 'HTTP', 53: 'DNS', 22: 'SSH', 23: 'Telnet'}

PYTHON ile VERİ ANALİZİ

- Dict Veri Yapısının temel özellikleri şunlardır:

- Değerler list, string, int ve benzeri olabilir
- Değerler tekrarlanabilir
- Değerler değiştirilebilir
- Dict, sıralamasız bir derlemedir; bu, bir dict veri yapısı içindeki item, oluşturulduğu sıradan farklı bir sıraya yerleştirilebilir.

PYTHON ile VERİ ANALİZİ

- ERP = {"AX": "MS AXAPTA", "NAV" :"MS Navision"}
- print (ERP)
- print (ERP['AX'])
- print (ERP['NAV'])
- Ekran Çıktısı:
- {'AX': 'MS AXAPTA', 'NAV': 'MS Navision'}
- MS AXAPTA
- MS Navision

PYTHON ile VERİ ANALİZİ

- Dictionary yazılım formatı:

- Dictionary ismi = {key: value}

PYTHON ile VERİ ANALİZİ

- # copy() fonksiyonu
- ERP = {"AX": "MS AXAPTA", "NAV" :"MS Navision", "SAP": "Systems Applications and Products"}
- print
- print (ERP)
- print
- print (ERP['AX'])
- print (ERP['SAP'])
- ERPDict=ERP.copy() #Dict kopyalama
- print (ERPDict)

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:
- {'AX': 'MS AXAPTA', 'SAP': 'Systems Applications and Products', 'NAV': 'MS Navision'}

- MS AXAPTA
- Systems Applications and Products
- {'AX': 'MS AXAPTA', 'SAP': 'Systems Applications and Products', 'NAV': 'MS Navision'}

PYTHON ile VERİ ANALİZİ

- # copy() fonksiyonu
- ERP = {"AX": "MS AXAPTA", "NAV" :"MS Navision", "SAP": "Systems Applications and Products"}
- print (ERP)
- print ("\n")
- print (ERP['AX'])
- print (ERP['SAP'])
- ERPDict=ERP.copy() #Dict kopyalama
- print (ERPDict)
- print ("\n")
- ERP["Odoo"]= "Odoo ERP Software" #dict veri yapısına üye ilave etme
- print (ERP)

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:
- {'AX': 'MS AXAPTA', 'NAV': 'MS Navision', 'SAP': 'Systems Applications and Products'}

- MS AXAPTA
- Systems Applications and Products
- {'AX': 'MS AXAPTA', 'NAV': 'MS Navision', 'SAP': 'Systems Applications and Products'}

- {'AX': 'MS AXAPTA', 'NAV': 'MS Navision', 'SAP': 'Systems Applications and Products', 'Odoo': 'Odoo ERP Software'}

PYTHON ile VERİ ANALİZİ

- get() fonksiyonu, dictionary'den veri almak için kullanılır.

- ERP.get("SAP", "kayıt bulunamadı")

PYTHON ile VERİ ANALİZİ

- ERP = {"AX": "MS AXAPTA", "NAV" :"MS Navision", "SAP": "Systems Applications and Products"}
- ERPDict=ERP.copy() #Dict kopyalama
- print (ERPDict)
- print
- ERP["Odoo"]= "Odoo ERP Software" #dict veri yapısına üye ilave etme
- print (ERP)
- print (ERP.get("SAP", "not found")) #dict veri yapısında sorgu

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:
- {'AX': 'MS AXAPTA', 'SAP': 'Systems Applications and Products', 'NAV': 'MS Navision'}

- {'AX': 'MS AXAPTA', 'SAP': 'Systems Applications and Products', 'NAV': 'MS Navision', 'Odoo': 'Odoo ERP Software'}
- Systems Applications and Products

PYTHON ile VERİ ANALİZİ

- ERP = {"AX": "MS AXAPTA", "NAV" :"MS Navision", "SAP": "Systems Applications and Products"}
- print (ERP)
- print ("\n")
- ERP["Odoo"]= "Odoo ERP Software" #dict veri yapısına üye ilave etme
- print (ERP)
- print ("\n")
- print (ERP.get("SAP", "not found"))#dict veri yapısında sorgu
- print ("\n")
- print (ERP.setdefault('AX', "unknown")) #key ile sorguprint ("\n")
- print ("\n")
- print (ERP.setdefault('Dolibarr', "Dict veri yapısında mevcut değildir")) #key ile sorgu

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:
- {'AX': 'MS AXAPTA', 'NAV': 'MS Navision', 'SAP': 'Systems Applications and Products'}
- {'AX': 'MS AXAPTA', 'NAV': 'MS Navision', 'SAP': 'Systems Applications and Products', 'Odoo': 'Odoo ERP Software'}
- Systems Applications and Products
- MS AXAPTA
- Dict veri yapısında mevcut değildir

PYTHON ile VERİ ANALİZİ

- ERP = {"AX": "MS AXAPTA", "NAV" :"MS Navision", "SAP": "Systems Applications and Products"}
- print (ERP)
- print ("\n")
- ERP["Odoo"]= "Odoo ERP Software" #dict veri yapısına üye ilave etme
- print (ERP)
- print (ERP.get("SAP", "not found")) #dict veri yapısında sorgu
- print (ERP.setdefault('AX', "unknown")) #key ile sorgu
- print (ERP.setdefault('Dolibarr', "Dict veri yapisinda mevcut degildir")) #key ile sorgu
- print ("\n")
- print (ERP)

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:
- {'AX': 'MS AXAPTA', 'SAP': 'Systems Applications and Products', 'NAV': 'MS Navision'}

- {'AX': 'MS AXAPTA', 'SAP': 'Systems Applications and Products', 'NAV': 'MS Navision', 'Odoo': 'Odoo ERP Software'}
- Systems Applications and Products
- MS AXAPTA
- Dict veri yapisinda mevcut degildir

- {'AX': 'MS AXAPTA', 'SAP': 'Systems Applications and Products', 'NAV': 'MS Navision', 'Dolibarr': 'Dict veri yapisinda mevcut degildir', 'Odoo': 'Odoo ERP Software'}

PYTHON ile VERİ ANALİZİ

- has_key() fonksiyonu
- Python 2.7 versiyonunda kullanılmakta idi.
- Yeni versiyonda ‘in’ deyimi kullanılmaktadır.
- Sorguda «True veya False» boolean sonuç döndürmektedir.

PYTHON ile VERİ ANALİZİ

- # program
- ERP = {"AX": "MS AXAPTA", "NAV" :"MS Navision", "SAP": "Systems Applications and Products"}
- print (ERP)
- print ("\n")
- ERP["Odoo"]= "Odoo ERP Software" #dict veri yapısına üye ilave edilir
- print (ERP)
- print ("\n")
- print ("SAP" in ERP) #dict veri yapısında sorgu
- print ("\n")
- print (ERP)

PYTHON ile VERİ ANALİZİ

- {'AX': 'MS AXAPTA', 'SAP': 'Systems Applications and Products', 'NAV': 'MS Navision'}
- {'AX': 'MS AXAPTA', 'SAP': 'Systems Applications and Products', 'NAV': 'MS Navision', 'Odoo': 'Odoo ERP Software'}
- True
- {'AX': 'MS AXAPTA', 'SAP': 'Systems Applications and Products', 'NAV': 'MS Navision', 'Odoo': 'Odoo ERP Software'}

PYTHON ile VERİ ANALİZİ

- #program
- ERP = {"AX": "MS AXAPTA", "NAV" :"MS Navision", "SAP": "Systems Applications and Products"}
- print (ERP)
- print
- ERP["Odoo"]= "Odoo ERP Software" #dict veri yapısına üye ilave edilir
- print (ERP)
- print ("Adempiere" in ERP) #dict veri yapısında sorgu

PYTHON ile VERİ ANALİZİ

- {'AX': 'MS AXAPTA', 'SAP': 'Systems Applications and Products', 'NAV': 'MS Navision'}

- {'AX': 'MS AXAPTA', 'SAP': 'Systems Applications and Products', 'NAV': 'MS Navision', 'Odoo': 'Odoo ERP Software'}
- False

PYTHON ile VERİ ANALİZİ

- keys() metodu
- Dictionary veri yapısındaki key'leri listeler.

PYTHON ile VERİ ANALİZİ

- ERP = {"AX": "MS AXAPTA", "NAV" :"MS Navision", "SAP": "Systems Applications and Products"}
- print (ERP)
- print ("\n")
- ERP.keys()
- print (ERP.keys())

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:
- {'AX': 'MS AXAPTA', 'NAV': 'MS Navision', 'SAP': 'Systems Applications and Products'}

- dict_keys(['AX', 'NAV', 'SAP'])

PYTHON ile VERİ ANALİZİ

- values() metodu

- Dictionary veri yapısındaki değerler listeler.

PYTHON ile VERİ ANALİZİ

- ERP = {"AX": "MS AXAPTA", "NAV" :"MS Navision", "SAP": "Systems Applications and Products"}
- print (ERP)
- print ("\n")
- ERP.values()
- print (ERP.values())

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:
- {'AX': 'MS AXAPTA', 'NAV': 'MS Navision', 'SAP': 'Systems Applications and Products'}

- dict_values(['MS AXAPTA', 'MS Navision', 'Systems Applications and Products'])

PYTHON ile VERİ ANALİZİ

- ERP = {"AX": "MS AXAPTA", "NAV" :"MS Navision", "SAP": "Systems Applications and Products"}
- print (ERP)
- print ("\n")
- ERP.items() # items() metodu veya fonksiyonu
- print (ERP.items())

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:
- {'AX': 'MS AXAPTA', 'NAV': 'MS Navision', 'SAP': 'Systems Applications and Products'}

- dict_items([('AX', 'MS AXAPTA'), ('NAV', 'MS Navision'), ('SAP', 'Systems Applications and Products')])

PYTHON ile VERİ ANALİZİ

- clear() metodu

- ERP.clear()

PYTHON ile VERİ ANALİZİ

- clear() metodu

- clear () metodu, dictionary-sözlükteki tüm ögeleri(items) silmektedir.
- Formatı: dict.clear ()
- Parametreler: clear () metodu herhangi bir parametre almaz.
- Değer Döndürme: clear () yöntemi herhangi bir değer döndürmez.

PYTHON ile VERİ ANALİZİ

- ERP = {"AX": "MS AXAPTA", "NAV" :"MS Navision", "SAP": "Systems Applications and Products"}
- print (ERP)
- print ("\n")
- ERP.clear() # clear() metodu
- print (ERP.clear())
- print (ERP)

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:

- {'AX': 'MS AXAPTA', 'NAV': 'MS Navision', 'SAP': 'Systems Applications and Products'}

- None
- {}

PYTHON ile VERİ ANALİZİ

```
spanish = dict()
spanish['hello'] = 'hola'
spanish['yes'] = 'si'
spanish['one'] = 'uno'
spanish['two'] = 'dos'
spanish['three'] = 'tres'
spanish['red'] = 'rojo'
spanish['brown'] = 'marron'
spanish['green'] = 'verde'
spanish['blue'] = 'azul'

print(spanish)
print ("\n")
print(spanish['two'])
print(spanish['red'])
```

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:
- {'hello': 'hola', 'yes': 'si', 'one': 'uno', 'two': 'dos', 'three': 'tres', 'red': 'rojo', 'brown': 'marron', 'green': 'verde', 'blue': 'azul'}

- dos
- rojo

PYTHON ile VERİ ANALİZİ

- dict = {'Marka': 'Toyota', 'Yıl': 2016}
- print ("Marka : %s" % dict.get('Yıl'))
- print ("Value : %s" % dict.get('Servis', "Mevcut Değil"))

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:
- Marka : 2016
- Value : Mevcut Değil

PYTHON ile VERİ ANALİZİ

- import pandas as pd

- data = {
- "isim ": ["Kaya", "Meltem", "Yağmur"],
- "yaş": [23, 25, 28]
- }
- print(data)

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:

- {'isim ': ['Kaya', 'Meltem', 'Yağmur'], 'yaş': [23, 25, 28]}

PYTHON ile VERİ ANALİZİ

- print(data)

- {'isim ': ['Kaya', 'Meltem', 'Yağmur'], 'yaş': [23, 25, 28]}

PYTHON ile VERİ ANALİZİ

- # veri türünün bulunması
- type(data)
- Ekran Çıktısı:
- dict

PYTHON ile VERİ ANALİZİ

- # Python dictionary veri yapısının pandas dataFrame'e dönüştürülmesi

- df = pd.DataFrame(data)

PYTHON ile VERİ ANALİZİ

- print(df)

- Ekran Çıktısı:

- isim yaş
- 0 Kaya 23
- 1 Meltem 25
- 2 Yağmur 28

PYTHON ile VERİ ANALİZİ

- # veri türünün bulunması

- type(df)

- Ekran Çıktısı:
- pandas.core.frame.DataFrame

PYTHON ile VERİ ANALİZİ

- # Veri türlerinin minimum ve maksimum değerleri

- import numpy as np

- print(np.iinfo('int8'))
- print(np.iinfo('int16'))
- print(np.iinfo('int32'))
- print(np.iinfo('int64'))

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:

- Machine parameters for int8
- ---
- min = -128
- max = 127
- ---

- Machine parameters for int16
- ---
- min = -32768
- max = 32767
- ---

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:

- Machine parameters for int32
- ----------
- min = -2147483648
- max = 2147483647
- ----------

- Machine parameters for int64
- ----------
- min = -9223372036854775808
- max = 9223372036854775807
- ----------

PYTHON ile VERİ ANALİZİ

- Bir başka Python veri yapısı ise Küme(SET) veri yapısıdır.
- Kümeler, Matematiksel küme teorisini dikkate alan, küme işlemlerini destekleyen, benzersiz ve değişmez nesnelerin sırasız bir veri türüdür.

PYTHON ile VERİ ANALİZİ

- Kümeler, aynı ögenin birden fazla oluşumuna izin vermediğinden, tekrarlanan değerleri önlemek için kullanılabilirler.
- Bir küme, bir nesneler koleksiyonudur (üyeler veya ögeler olarak adlandırılır).
- Kümedeki veriler öge veya üye olarak adlandırılırlar.

PYTHON ile VERİ ANALİZİ

- A kümesi 1 ile 10 arasında çift sayıları kapsamaktadır.
- A Kümesi:
- {2,4,6,8,10}

PYTHON ile VERİ ANALİZİ

- B kümesi 1 ile 10 arasında tek sayılar oluşturulmuştur.
- B Kümesi
- {1,3,5,7,9}

PYTHON ile VERİ ANALİZİ

- # Küme Örnekleri
- s1 = set([1,2,3,4,5,6])
- print(s1)
- s2 = set([1,2,2,3,4,4,1,1,5,6,6])
- print(s2)
- s3 = set([3,2,3,4,5,6,6,6,1,1,2])
- print(s3)

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:
- {1, 2, 3, 4, 5, 6}
- {1, 2, 3, 4, 5, 6}
- {1, 2, 3, 4, 5, 6}

PYTHON ile VERİ ANALİZİ

- set_4 = {'AX','MS AXAPTA', 'SAP', 'NAV', 'oracle'}
- print(set_4)

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:

- {'SAP', 'MS AXAPTA', 'AX', 'NAV', 'oracle'}

PYTHON ile VERİ ANALİZİ

- #kümeye veri ilave edilmesi
- set_4.add('Google Colab')
- print(set_4)

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:

- {'SAP', 'MS AXAPTA', 'AX', 'Google Colab', 'NAV', 'oracle'}

PYTHON ile VERİ ANALİZİ

- #pandas kullanımı
- import pandas as pd

- import deyimi, bir modüldeki veya kütüphanedeki bir fonksiyonu kullanmak için kullanılmaktadır.
- import deyimi, import keyword'ü ve modül ismi birlikte kullanılır.

PYTHON ile VERİ ANALİZİ

- pandas, panel veriden (ekonometrik bir terim) ve Python veri analizinden ismini almıştır.
- Bir açık kaynaklı Python kütüphanesidir.
- Yüksek performanslı, kullanımı kolay veri yapıları ve veri analizi için bir tool'dur
- PYTHON Programlama Dili ile VERİ ANALİZİ için tasarlanmıştır.

PYTHON ile VERİ ANALİZİ

- Konular:
- pandas DataFrame
- pandas Series
- pandas Verileri Sorgulama
- pandas DataFrame ile İstatistik
- pandas DataFrame ile veri toplama
- DataFrame ekleme ve birleştirme
- Eksik Veriler(Missing Data) ile İşlem

PYTHON ile VERİ ANALİZİ

```
import random


# 1 – 15 arası 10 adet integer sayı üretilmesi
for i in range(10):
    print(random.randint(1, 15))
```

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:
- 13
- 15
- 3
- 1
- 4
- 4
- 9
- 15
- 5
- 1

PYTHON ile VERİ ANALİZİ

- import random

- # 1 – 15 arası 10 adet ondalık sayı üretilmesi
- for i in range(10):
- print(random.uniform(1, 15))

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:
- 1.4575047837669597
- 8.24232336018019
- 2.8888574319267297
- 11.3781682174114 62
- 12.531484315990259
- 7.5152397980027 06
- 14.288748518226 85
- 12.808882151291954
- 1.5196109928400494
- 12.6375398882434 65

PYTHON ile VERİ ANALİZİ

```
import random


# 10 adet tesadüfi float sayı üretilmesi
for sayaç in range(10):
    print(random.uniform(10.5, 75.5))
```

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:
- 20.342650722290337
- 64.00286067304337
- 41.64991274168381
- 17.971661822347O1
- 62.784561511811276
- 11.2019313483934 26
- 33.6843880706655 55
- 58.44387841680983
- 15.9368601533464 69
- 47.016825786844045

PYTHON ile VERİ ANALİZİ

Kaynaklar:

Python for Data Analysis, William Wesley McKinney, O'Reilly Media, Inc., 2017.

Python Crash Course: A Hands-On, Project-Based Introduction to Programming, Eric Matthes, 2015.

Beginning Programming with Python, John Paul Mueller, 2014.

Learn Python in 7 Days, Mohit; Bhaskar N. Das, Packt Publishing, 2017.

https://realpython.com/python-lists-tuples/

https://www.digitalocean.com/community/tutorials/how-to-import-modules-in-python-3

https://thispointer.com/python-tuple-append-insert-modify-delete-elements-in-tuple/

http://anh.cs.luc.edu/python/hands-on/3.1/handsonHtml/dictionaries.html

https://thispointer.com/python-how-to-find-keys-by-value-in-dictionary/

https://www.learnpython.org/en/String_Formatting