Programlama

Python 101

Ahmet Anıl Dindar
(Gebze Teknik Üniversitesi)

eposta: adindar@gtu.edu.tr

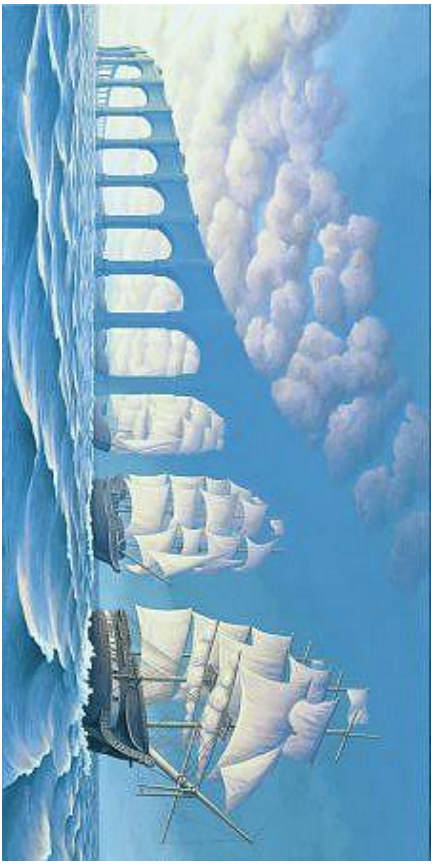
web: a2d.co

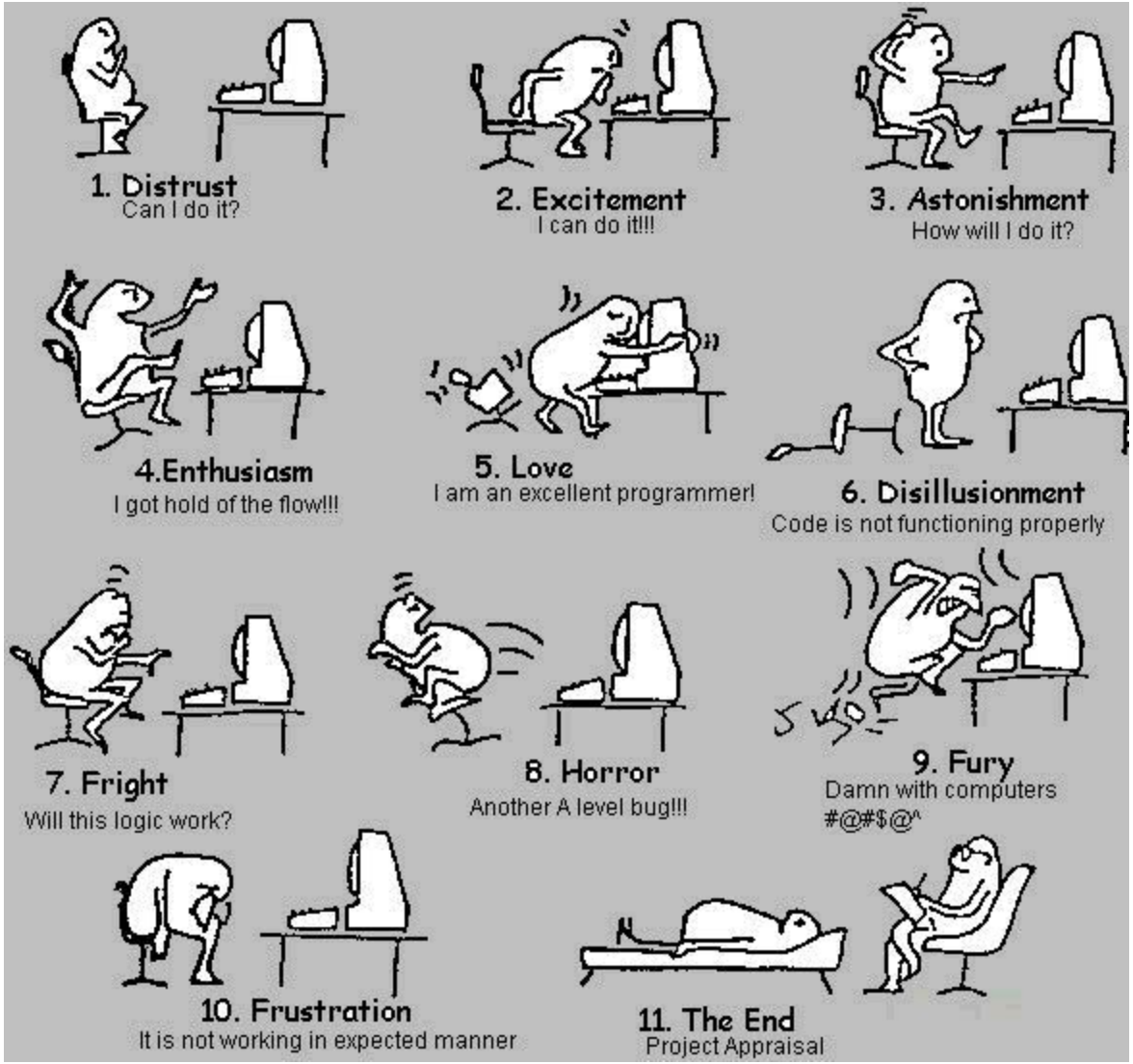
GİRİŞ

ALGORİTMA

PYTHON

SONUÇ




1. Distrust
Can I do it?
2. Excitement
I can do it!!!
3. Astonishment
How will I do it?
4.Enthusiasm
I got hold of the flow!!!
5. Love
I am an excellent programmer!
6. Disillusionment
Code is not functioning properly
7. Fright
Will this logic work?
8. Horror
Another A level bug!!!
9. Fury
Damn with computers
#@#$@^
10. Frustration
It is not working in expected manner
11. The End
Project Appraisal

Programlama;

- Gerekli mi?
- Ya olmasa?
- Nereden başlamalı?
- Ne bilmeli?
- Ne hedeflemeli?
- Nerede durmalı?

Evet

Varlığını değil, yokluğunu sorgulayalım

Ayağınızı bastığınız yerden başlayınız

Limit gökyüzü (mü)?

Ne gerekli?

Hedefiniz arttıkça gideri var...

Gerekli mi?

Gauss

$$\underbrace{1+\underbrace{2+3+4+5+\quad+99}_{2+99=101}+100}_{1+100=101}$$

Ya olmasa?

Gauss-Jordan

Write augmented matrix $\left[\begin{array}{ccc|c} 9 & -2 & -1 & 26 \\ -8 & -1 & -4 & -5 \\ -5 & -1 & -2 & -3 \end{array}\right]$

$R_1 \to R_1 + R_2$, $R_3 \to R_3 + R_1$ $\left[\begin{array}{ccc|c} 1 & -3 & -5 & 21 \\ -8 & -1 & -4 & -5 \\ 4 & -3 & -3 & 23 \end{array}\right]$

$R_2 \to R_2 + 2R_3$, $R_3 \to R_3 - 4R_1$ $\left[\begin{array}{ccc|c} 1 & -3 & -5 & 21 \\ 0 & -7 & -10 & 41 \\ 0 & 9 & 17 & -61 \end{array}\right]$

$R_2 \to R_2 + R_3$ $\left[\begin{array}{ccc|c} 1 & -3 & -5 & 21 \\ 0 & 2 & 7 & -20 \\ 0 & 9 & 17 & -61 \end{array}\right]$

$R_1 \to R_1 + R_2$, $R_3 \to R_3 - 4R_2$ $\left[\begin{array}{ccc|c} 1 & -1 & 2 & 1 \\ 0 & 2 & 7 & -20 \\ 0 & 1 & -11 & 19 \end{array}\right]$

$R_1 \to R_1 + R_3$, $R_2 \to R_2 - R_3$ $\left[\begin{array}{ccc|c} 1 & 0 & -9 & 20 \\ 0 & 1 & 18 & -39 \\ 0 & 1 & -11 & 19 \end{array}\right]$

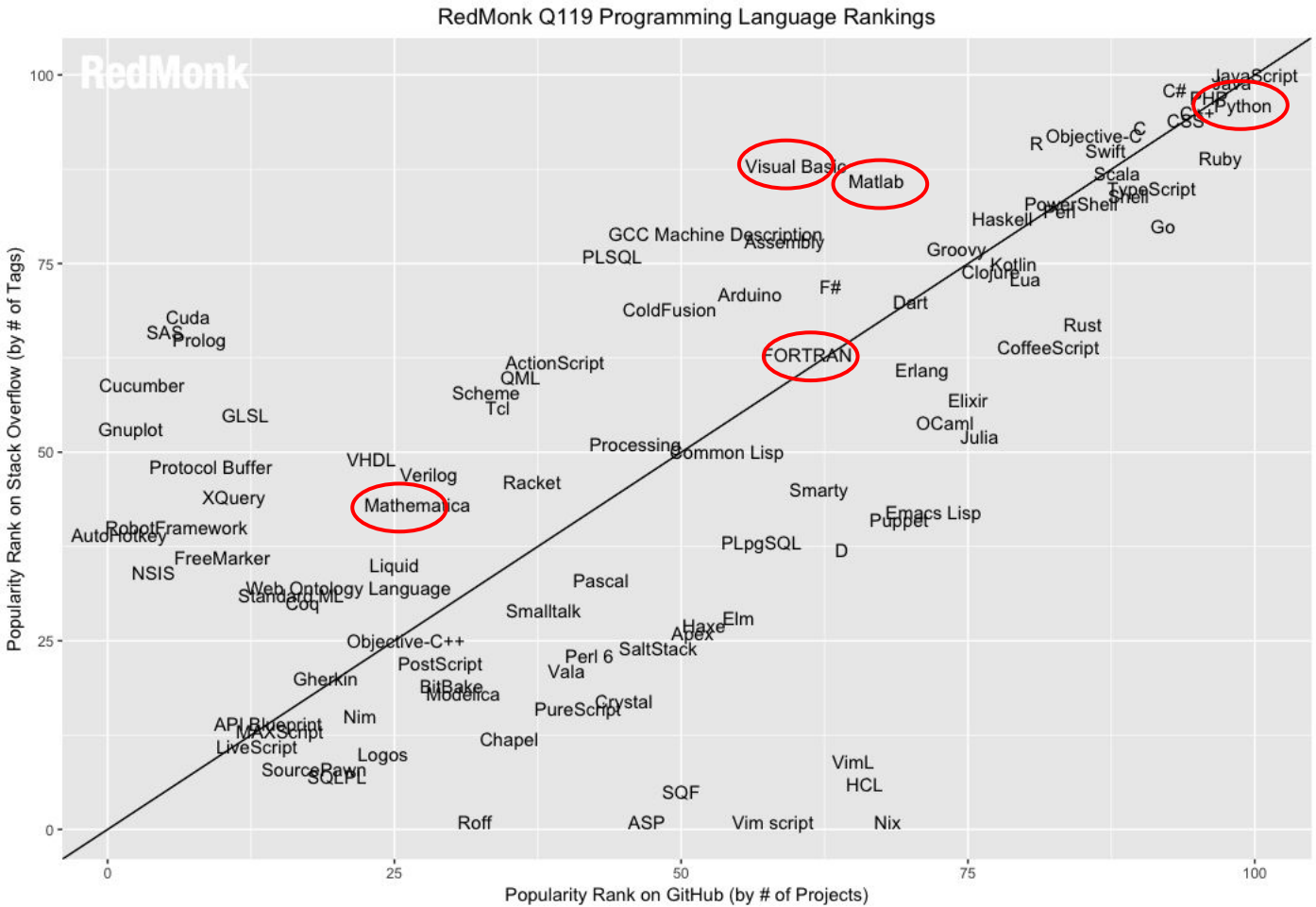
$R_3 \to R_3 - R_2$ $\left[\begin{array}{ccc|c} 1 & 0 & -9 & 20 \\ 0 & 1 & 18 & -39 \\ 0 & 0 & -29 & 58 \end{array}\right]$

$R_3 \to -\frac{1}{29} R_3$ $\left[\begin{array}{ccc|c} 1 & 0 & -9 & 20 \\ 0 & 1 & 18 & -39 \\ 0 & 0 & 1 & -2 \end{array}\right]$

$R_1 \to R_1 + 9R_3$, $R_2 \to R_2 - 18R_3$ $\left[\begin{array}{ccc|c} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & -2 \end{array}\right]$

$\therefore$ $x = 2$
$y = -3$
$z = -2$

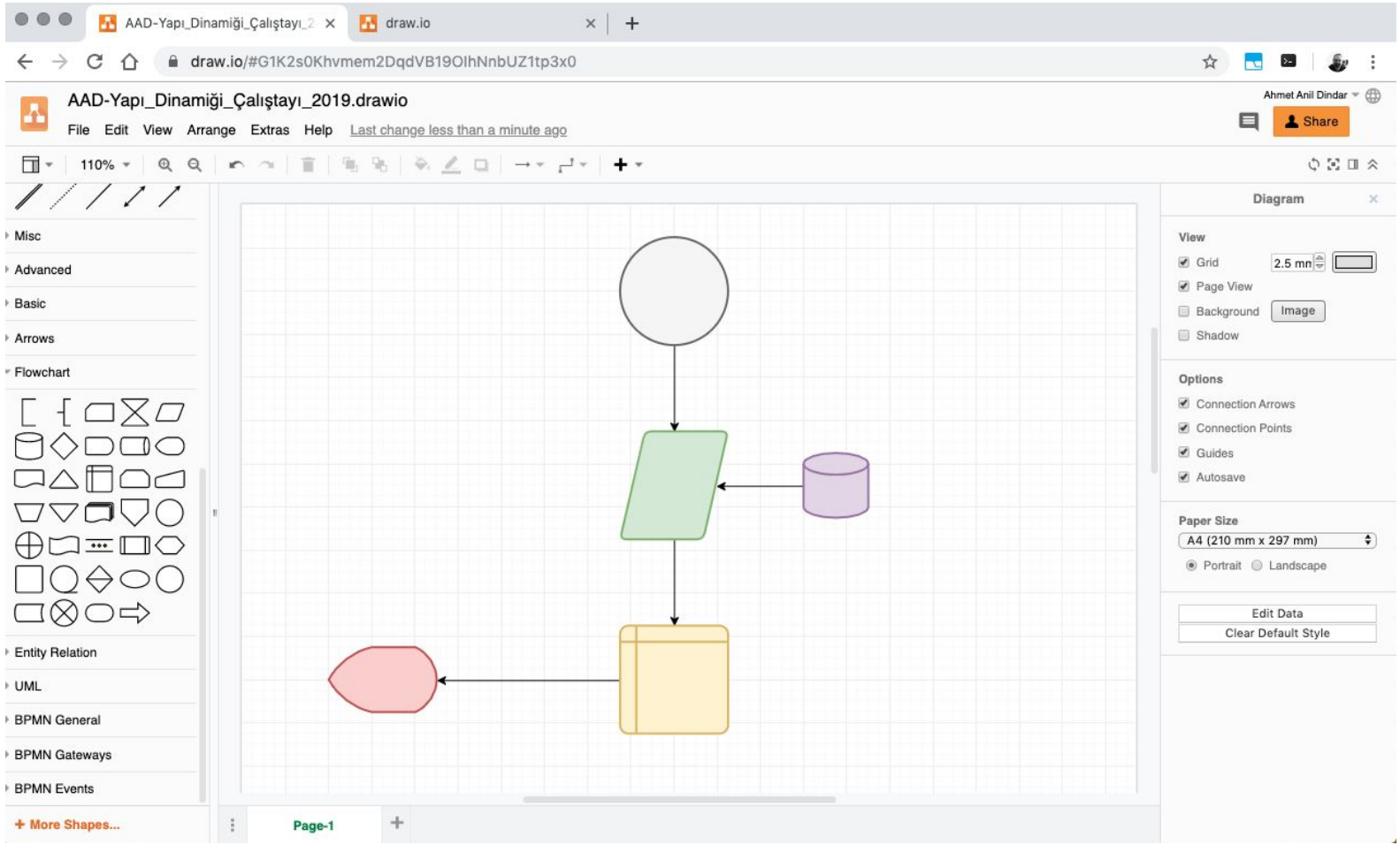
Nereden başlamalı? "Python" :)



https://redmonk.com/sogrady/2019/03/20/language-rankings-1-19/

Ne bilmeli?

DrawIO



GİRİŞ

Ne hedeflemeli?

Döngü & durum

Home » Oracle Unveils Collaborative Test Site for Construction Tech

Information technology | Construction technology | Tech

Oracle Unveils Collaborative Test Site for Construction Tech

A 4D model of the test site built from drone imagery in Reconstruct's software.

PHOTO BY JEFF RUBENSTONE/ENR

"…
Oracle has created a ***Construction and Engineering Innovation Lab****.*

"Every single day, companies reach out to us with new solutions they want to integrate. And we believe in the open ecosystem," says ***Burcin Kaplanoglu****, senior director for industry strategy and innovation at Oracle's Construction and Engineering global business unit.*

…"

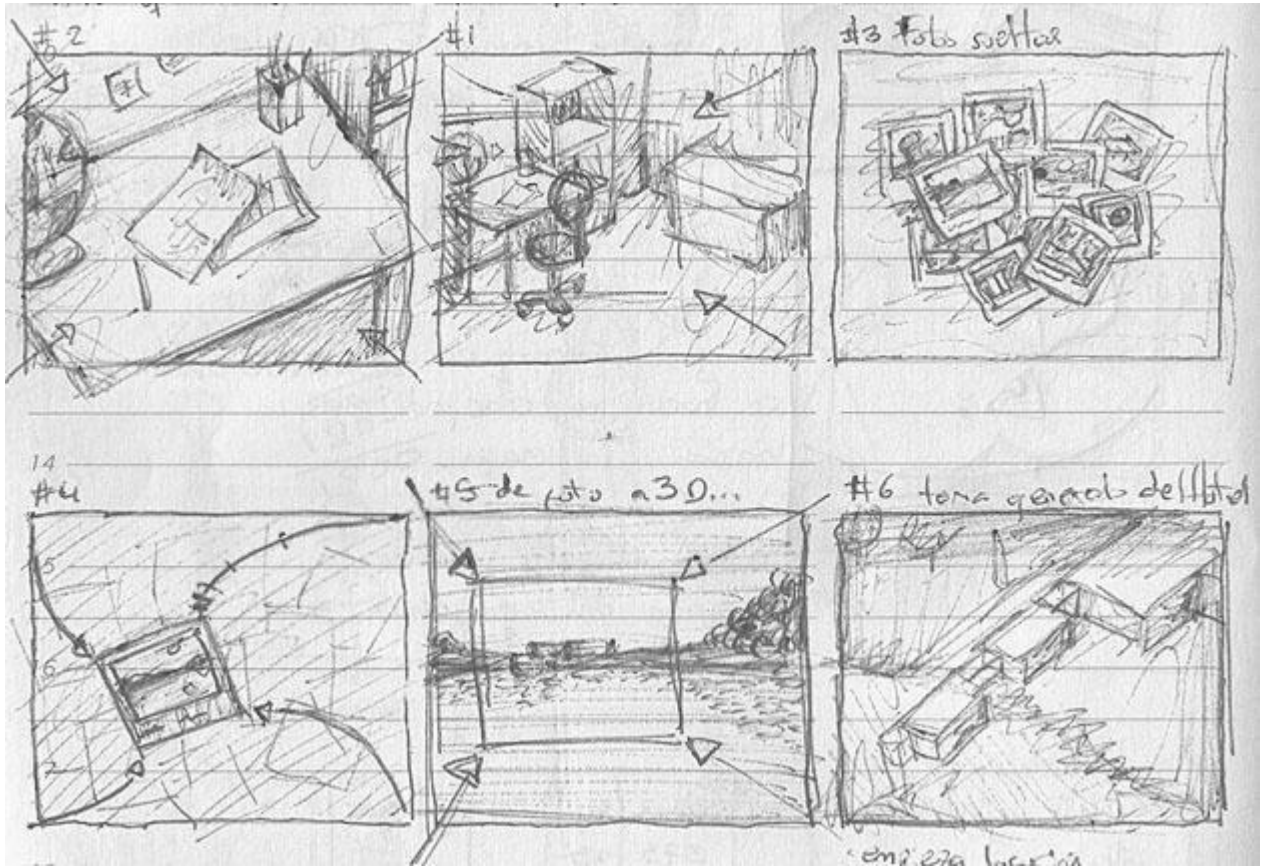
https://www.enr.com/articles/45097-oracle-unveils-collaborative-test-site-for-construction-tech

Nerede durmalı?

WWW

Kodlama bir araç olmalıdır, amaç değil.

Neyi, ne kadar biliyorsun?



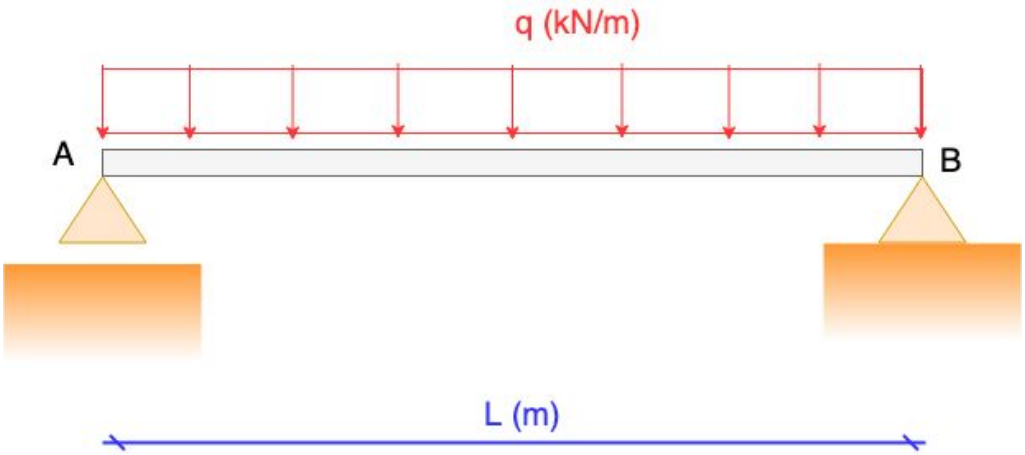
https://flic.kr/p/6kLUwn

Bana bildiğinin resmini çizer misin Abidin?

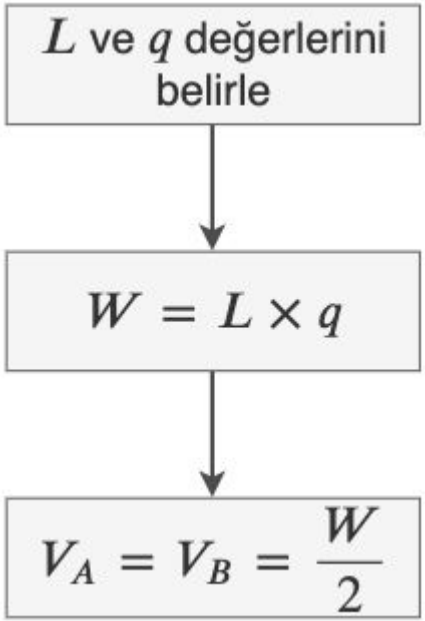
ALGORITMA

Problem :

Mesnet reaksiyonlarını hesaplayınız.

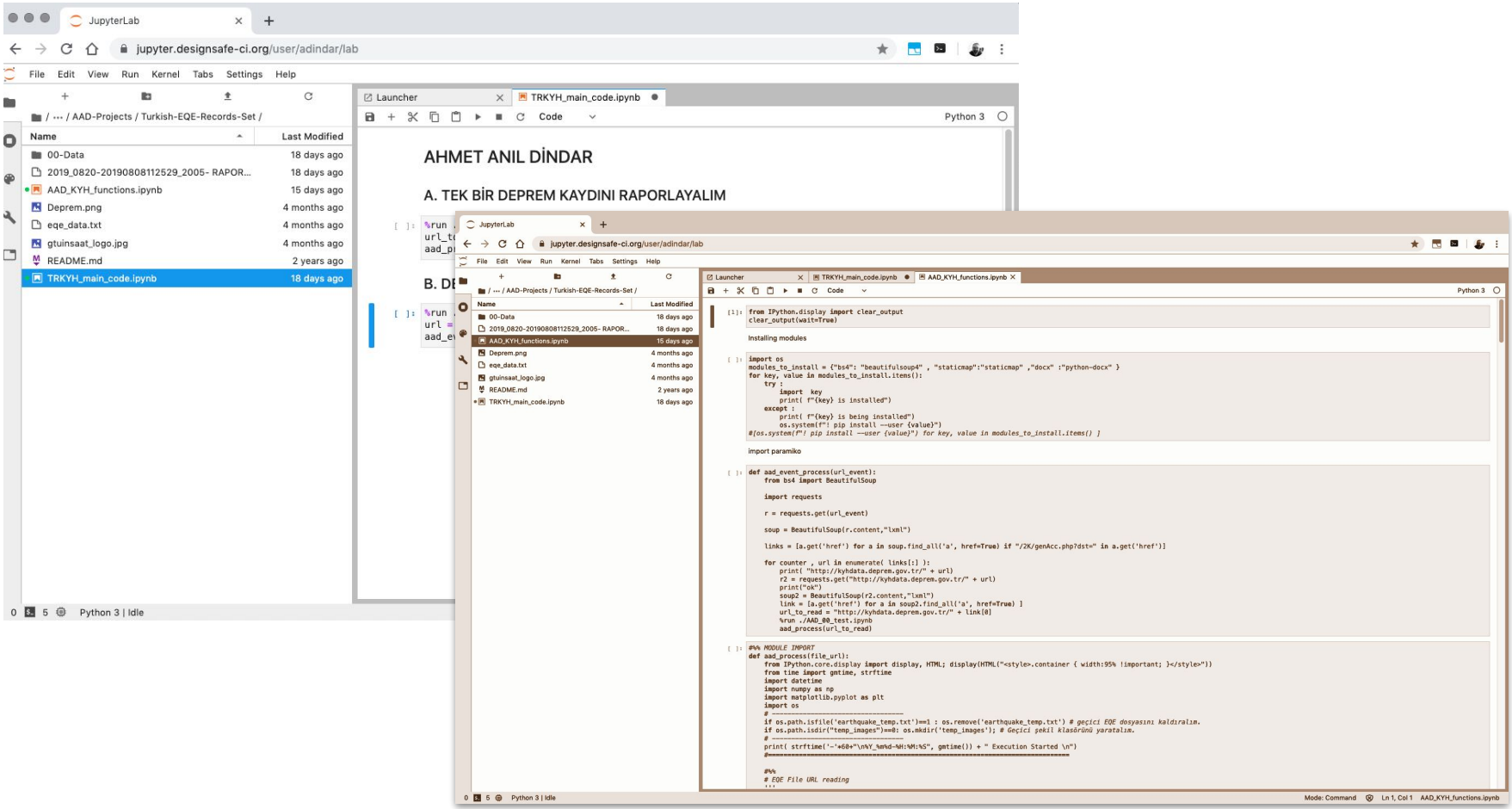


Algoritma :



Programalamdan önce planla?

ALGORITMA



01. Python Dilinin Temelleri :

PYTHON

"Hello, World"

- C

```
#include <stdio.h>

int main(int argc, char ** argv)
{
   printf("Hello, World!\n");
}
```

- Java

```
public class Hello
{
   public static void main(String argv[])
   {
      System.out.println("Hello, World!");
   }
}
```

- now in Python

```
print "Hello, World!"
```

Ücretsiz	Basit	Herkes

https://flic.kr/p/5Yk52e

01. Python Dilinin Temelleri :

PYTHON

Değişken	İşlem	Okuma/Yazma
• yazı (string) • tamsayı (integer) • ondalık (float) • liste (list) • sözlük (dictionary) • demet (tuple) • erişim	• Döngü • karar verme • matematik • fonksiyonlar • fonksiyoneller • matris • istatistik	• ASCII • BINARY • OPEN • NUMPY • PANDAS • PICKLE • MAT

01. Python Dilinin Temelleri :

PYTHON

Değişken	İşlem	Okuma/Yazma
• yazı (string)	• Döngü	• ASCII
• tamsayı (integer)	• karar verme	• BINARY
• ondalık (float)	• matematik	• OPEN
• liste (list)	• fonksiyonlar	• NUMPY
• sözlük (dictionary)	• fonksiyoneller	• PANDAS
• demet (tuple)	• matris	• PICKLE
• erişim	• istatistik	• MAT

01. Python Dilinin Temelleri :

PYTHON

Değişken

yazı

sayı

liste

sözlük

demet

01.1-Rakam

Rakam, matematiksel işlemler için belirnen büyüklükerdir ve sayıyı temsil eder.

- Tam sayı :

```
c = 1
```

- Kayan sayı:

```
d = 1.0
```

01. Python Dilinin Temelleri :

PYTHON

Değişken

yazı

sayı

liste

sözlük

demet

01.1-Yazı

Yazı değişkeni içinde alfanumerik karakterleri barındıran içeriklerdir.

Örneğin bir önceki cümledeki tüm elemanlar tekil tekil ve bütün olarak bir "yazı"dır.

```
a = "bu bir yazıdır"

b = 'bu da bir yazıdır'

c = "İzmir'deki en iyi kumrucu"
```

01. Python Dilinin Temelleri :

Değişken

yazı

sayı

liste

sözlük

demet

PYTHON

01.2-Liste (List)

Yazı veya rakamları veya başka listeleri bir dizinde tutmaya yarar.

```
e = [ 1 , 2 , 3 , 4]

f = [ a , b , c , d]

g = [ e , f]
```

01. Python Dilinin Temelleri :

Değişken

yazı

sayı

liste

sözlük

demet

PYTHON

01.3-Sözlük (Dictionary)

Tanımları ve bağlı değişkenleri veya listeleri bir yapı içinde saklamaya yarar.

```
ilk_sozluk = { "isim" : "Ahmet Anıl" , "soyisim" : "Dindar"}
```

```
ikinci_sozluk = { 1 : [10,20,40] , "isim" : ["ahmet" , 2.0 , 2 , 10]}
```

01. Python Dilinin Temelleri :

Değişken

yazı

sayı

liste

sözlük

demet

PYTHON

01.4-Demet (Tuple)

Tanımlanması ardından değişiklik yapılmayan dizilerdir.

```
meyveler = ( "elma" , "armut" )
```

01. Python Dilinin Temelleri :

Değişken

yazı

sayı

liste

sözlük

demet

PYTHON

01.5- Erişim ve Indeksleme

Tüm değişkenlere erişim ve indeksleme gereklidir. Tek bir elemandan, sıralı elemanlara kadar erişim sağlanmaktadır.

Python'da erişim köşeli parantez [] ile sağlanır.

Indeks değeri Matlab'den farklı olarak 0'dan başlar.

```
print( a )

print ( a[0] )
```

```
bu bir yazıdır
b
```

```
print( a[ : 5 ])
```

```
bu bi
```

```
print( a[ -5 : ])
```

```
zıdır
```

```
print( a[ 3 : 10 ])
```

```
bir yaz
```

01. Python Dilinin Temelleri :

PYTHON

Değişken

yazı

sayı

liste

sözlük

demet

- Listelerde erişim benzerdir.

```
print( e[0])
```

```
1
```

```
print( e[ 2 : ])
```

```
[3, 4]
```

```
print( e[ 1 :  3 ] )
```

```
[2, 3]
```

01. Python Dilinin Temelleri :

Değişken

yazı

sayı

liste

sözlük

demet

PYTHON

- Sözlüklerde erişim anahtarlar üzerinden olur

```
ilk_sozluk.keys()
```

```
dict_keys(['isim', 'soyisim'])
```

```
ilk_sozluk["isim"]
```

```
'Ahmet Anıl'
```

```
ilk_sozluk["isim"][0: 5]
```

```
'Ahmet'
```

01. Python Dilinin Temelleri :

Değişken

yazı

sayı

liste

sözlük

demet

PYTHON

- Demetlerde de erişim listelerdeki gibidir.

```
meyveler[0]
```

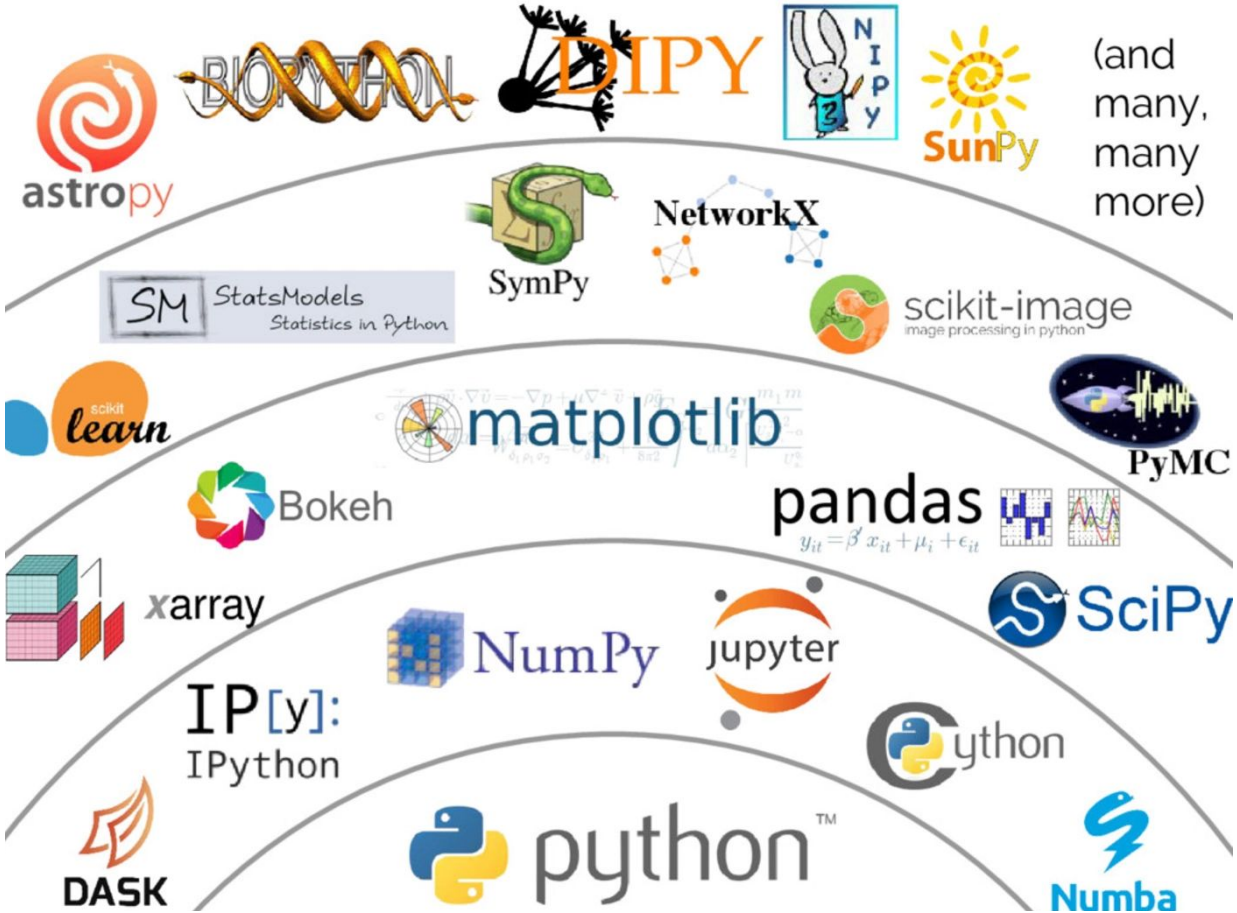
```
'elma'
```

```
meyveler[:]
```

```
('elma', 'armut')
```

01. Python Dilinin Temelleri :

Modüller



PYTHON

01. Python Dilinin Temelleri :

NumPy

PYTHON

NumPy bilimsel hesaplamalardaki temel pakettir.

- Çok boyutlu dizin işlemlerini yapar
- Geniş kapsamlı fonksiyonları vardır
- C/C++ ve Fortran kodlarını entegre edilmesini sağlar
- Lineer cebir, Fourier Dönüşümü, rasgele sayı gibi işlemleri içerir

Bu çalıştayda matris işlemlerini bu paketle yapacağız.

01. Python Dilinin Temelleri :

SciPy.org

SciPy, NumPy gibi bilimsel araştırmalarda kullanılan bir pakettir. Farklı olarak

- sayısal integrasyon
- enterpolasyon
- optimizasyon
- lineer cebir
- istatistik fonksiyonları

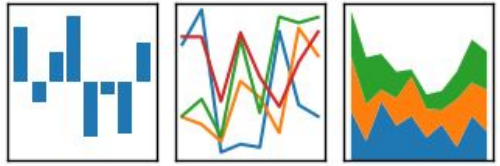
içermektedir.

PYTHON

01. Python Dilinin Temelleri :

pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$



PYTHON

Pandas, yüksek performanslı kolaylaştırılmış veri yapısı ve veri analizi araçlarını içermektedir.

Bu çalıştayda verilerin birleştirilmesi, dosya okuma/yazma işlemlerinde bu paketi kullanacağız.

01. Python Dilinin Temelleri :

matplotlib

Matplotlib, iki boyutlu görselleştirme kütüphanesi olup, çok farklı formatlarda grafiklerin üretilmesini sağlayan bir pakettir.

Python'un temel görselleştirme aracıdır.

Bu çalıştayda tüm grafikleri bu paketle oluşturacağız.

01. Python Dilinin Temelleri :

PYTHON

https://flic.kr/p/6oJEU8

01. Python Dilinin Temelleri :

PYTHON

Değişken	İşlem	Okuma/Yazma
yazı (string)	Döngü	ASCII
tamsayı (integer)	karar verme	BINARY
ondalık (float)	matematik	OPEN
liste (list)	fonksiyonlar	NUMPY
sözlük (dictionary)	fonksiyoneller	PANDAS
demet (tuple)	matris	PICKLE
erişim	istatistik	MAT

02. Python'da İşlemler :

PYTHON

İşlem
Döngü
karar verme
matematik
fonksiyonlar
fonksiyoneller
matris
istatistik

02.1- Döngüler

- Python'da döngüler "for", "while" ile başlar ancak "end" ile bitmez.
- Bunun yerine, satırların başlangıçları aynı hizada olacak şekilde girinti (indent) yapılır.

02. Python'da İşlemler :

PYTHON

İşlem

Döngü

karar verme

matematik

fonksiyonlar

fonksiyoneller

matris

istatistik

02.1.1- For

Değişkenlerin sırayla işleme alınması, tekrarlı işlemlere döngü denmektedir. Tipik döngüler, bir dizin içindeki elemanlar üzerinden olmaktadır. Döngüler başladıktan sonra istenen şartların yerine gelmesi halinde sonuna kadar kalmadan tamamlanabilir.

```
1  dongu_listesi = [1,2,3,4,5,6,7,8,9,10]
2
3  for i in dongu_listesi:
4      print(i)
5
```

02. Python'da İşlemler :

PYTHON

İşlem

Döngü

karar verme

matematik

fonksiyonlar

fonksiyoneller

matris

istatistik

02.1.1- For

```
dongu_listesi = [10,20,30,40,50,60,70,80,90,100]

for i,j in enumerate( dongu_listesi ) :
    print(i,j)

```

```
0 10
1 20
2 30
3 40
4 50
5 60
6 70
7 80
8 90
9 100
```

02. Python'da İşlemler :

İşlem

Döngü

karar verme

matematik

fonksiyonlar

fonksiyoneller

matris

istatistik

PYTHON

02.1.2- While

Döngünün belli şartları sağlamasına bağlı olarak sürdürülmesidir.

```
limit_deger = 10
sayac = 0

while sayac < limit_deger :

    print( np.random.rand())

    sayac += 1
```

02. Python'da İşlemler :

PYTHON

İşlem

Döngü

karar verme

matematik

fonksiyonlar

fonksiyoneller

matris

istatistik

02.2- Karar verme

İşlemler sırasında değişkenlerin istenen şartları sağlaması veya sağlamaması halinde sürecin değişimini sağlar.

```
birinci_veri = 20
ikinci_veri = 10

if birinci_veri > ikinci_veri:

    print( f"{birinci_veri}, {ikinci_veri}'den büyüktür")

    # yeni bir değişken yaratalım

    yeni_veri = birinci_veri + ikinci_veri

    print( yeni_veri)

```

```
20, 10'den büyüktür
30
```

02. Python'da İşlemler :

PYTHON

İşlem

Döngü

karar verme

matematik

fonksiyonlar

fonksiyoneller

matris

istatistik

02.2- Karar verme

Peki birden fazla durum söz konusu olamaz mı?

```python
birinci_veri , ikinci_veri , ucuncu_veri = 50, 60 , 40

if birinci_veri > ikinci_veri:

    print( f"{birinci_veri}, {ikinci_veri}'den büyüktür")

elif  birinci_veri < ikinci_veri and birinci_veri > ucuncu_veri:

    print( f"{birinci_veri}, {ikinci_veri}'den küçük ve {ucuncu_veri}'den büyüktür")

```

```
50, 60'den küçük ve 40'den büyüktür
```

02. Python'da İşlemler :

PYTHON

İşlem

Döngü

karar verme

matematik

fonksiyonlar

fonksiyoneller

matris

istatistik

02.2- Karar verme

Peki birden fazla durum söz konusu olamaz mı?

```python
birinci_veri = 50
ikinci_veri = 10
ucuncu_veri = 40

if birinci_veri > ikinci_veri:

    print( f"{birinci_veri}, {ikinci_veri}'den büyüktür")

elif birinci_veri > ucuncu_veri:

    print( f"{birinci_veri}, {ucuncu_veri}'den büyüktür")

```

```
50, 10'den büyüktür
```

02. Python'da İşlemler :

PYTHON

İşlem
Döngü
karar verme
matematik
fonksiyonlar
fonksiyoneller
matris
istatistik

02.3.1- Temel İşlemler

Toplama, Çıkarma, Çarpma, Bölme

```
ali , veli = 45 , 55

print ( ali + veli )

print ( ali - veli )

print ( ali * veli )

print ( ali / veli )

```

```
100
-10
2475
0.8181818181818182
```

02. Python'da İşlemler :

PYTHON

İşlem
Döngü
karar verme
matematik
fonksiyonlar
fonksiyoneller
matris
istatistik

02.3.1- Temel İşlemler

Üssel ifadesi : Elveda ^ işareti

```
1  print( ali * ali )
2
3  print( ali ** 2 )
4
```

```
2025
2025
```

02. Python'da İşlemler :

PYTHON

İşlem
Döngü
karar verme
matematik
fonksiyonlar
fonksiyoneller
matris
istatistik

02.3.1- Temel İşlemler

Yuvarlama : Numpy nerede karşımıza çıkacak diyorduk :)

```
print( np.floor( 1.4 ))

print( np.ceil( 1.4 ))
```

```
1.0
2.0
```

02. Python'da İşlemler :

PYTHON

İşlem
Döngü
karar verme
matematik
fonksiyonlar
fonksiyoneller
matris
istatistik

02.3.1- Temel İşlemler

Yuvarlama : Ondalık yuvarlama

```
1  round( 1.23456 , 2)
```

```
1.23
```

```
1  np.round( 1.23456 , 2)
```

```
1.23
```

02. Python'da İşlemler :

PYTHON

İşlem
Döngü
karar verme
matematik
fonksiyonlar
fonksiyoneller
matris
istatistik

02.3.2- İleri işlemler

Integral: Trapezoidal integral

```
from numpy import random

listem = [11,22,33,44]

from scipy import integrate

print( integrate.trapz( listem))

```

```
82.5
```

02. Python'da İşlemler :

PYTHON

İşlem
Döngü
karar verme
matematik
fonksiyonlar
fonksiyoneller
matris
istatistik

02.3.2- İleri işlemler

Integral: Trapezoidal integral

```
from numpy import random

listem = [11,22,33,44]

from scipy import integrate

print( integrate.cumtrapz( listem))

```

```
[16.5 44.  82.5]
```

02. Python'da İşlemler :

PYTHON

İşlem
Döngü
karar verme
matematik
fonksiyonlar
fonksiyoneller
matris
istatistik

02.3.2- İleri işlemler

Farklar

```
listem = [21 , 3 , 5 , 4, 99]

import numpy as np

print ( np.diff( listem ) )
```

```
[-18   2  -1  95]
```

02. Python'da İşlemler :

PYTHON

İşlem
Döngü
karar verme
matematik
fonksiyonlar
fonksiyoneller
matris
istatistik

02.3.2- İleri işlemler

Farklar

```
listem = [21 , 3 , 5 , 4, 99]

import numpy as np

print ( np.diff( listem ) )
```

```
[-18   2  -1  95]
```

02. Python'da İşlemler :

PYTHON

İşlem
Döngü
karar verme
matematik
fonksiyonlar
fonksiyoneller
matris
istatistik

02.4 Fonksiyonlar

İşlemler dizisini bir tanım altına almaktır.

```
def eleman_secimi( liste_verilen , eleman_secilen):

    secelim = liste_verilen[ eleman_secilen]

    return(secelim)
```

```
bunebe = eleman_secimi( [1,"a",3,4,"he"] , 3 )

print( bunebe)
```

4

02. Python'da İşlemler :

PYTHON

İşlem
Döngü
karar verme
matematik
<u>fonksiyonlar</u>
fonksiyoneller
matris
istatistik

02.4 Fonksiyonlar

! DİKKAT !

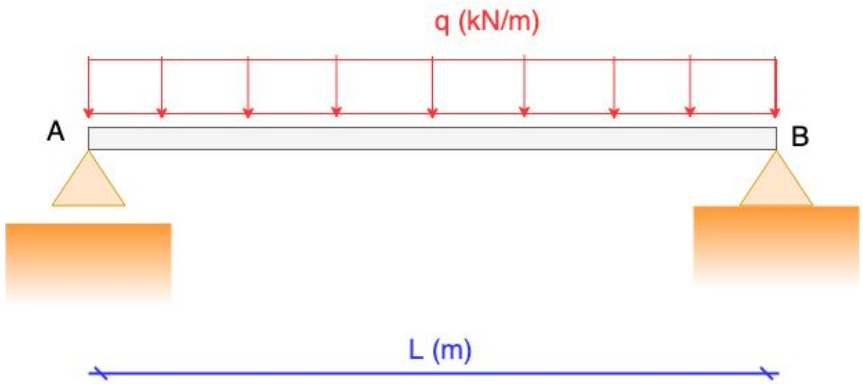
Bilimsel çalışmalarda, üretilen tüm **algoritmaların** **fonksiyon** haline getirilmesi **stratejisi önerilmektedir**.

02. Python'da İşlemler :

PYTHON

İşlem
Döngü
karar verme
matematik
<u>fonksiyonlar</u>
fonksiyoneller
matris
istatistik

02.4 Fonksiyonlar



L ve q değerlerini belirle

↓

$$W = L \times q$$

↓

$$V_A = V_B = \frac{W}{2}$$

```python
def basit_kiris( q , L):

    W =  ( q * L)

    v = 0.5 * W

    return( v)

basit_kiris( 10 , 5 )
```

```
25.0
```

02. Python'da İşlemler :

PYTHON

İşlem
Döngü
karar verme
matematik
fonksiyonlar
<u>fonksiyoneller</u>
matris
istatistik

02.5 Fonksiyoneller

İçiçe geçmiş fonksiyonlardır.

```
def bolme_islemi( a , b ):
    return( yuvarlama_programi(  a / b))

def yuvarlama_programi( a ) :
    return( round(a , 4) )

print( bolme_islemi(9 , 4.333) )

```

```
2.0771
```

02. Python'da İşlemler :

PYTHON

İşlem
Döngü
karar verme
matematik
fonksiyonlar
fonksiyoneller
matris
istatistik

02.6 Matrisler

- Python, MATLAB gibi değildir. Değişkenleri bir Matris gibi düşünmez.
- Bu sebeple MATLAB'de alışılan yaklaşım ve notasyondan farklı bir durum vardır.
- Her ne kadar Numpy içinde "matrix" diye bir sınıf bulunuyor olsa da gelecekte iptal edilecektir.
- Bundan dolayı, numpy içindeki "array" sınıfını kullanmayı önermekteyiz.

02. Python'da İşlemler :

PYTHON

İşlem
Döngü
karar verme
matematik
fonksiyonlar
fonksiyoneller
matris
istatistik

02.6.1 Çok boyutlu dizin = Matris işlemleri

Bir boyutlu dizin yaratma

```
import numpy as np

matris_1d = np.array([[1,2]])

print( matris_1d )
```

```
[[1 2]]
```

02. Python'da İşlemler :

PYTHON

İşlem

Döngü

karar verme

matematik

fonksiyonlar

fonksiyoneller

matris

istatistik

02.6.1 Çok boyutlu dizin = Matris işlemleri

iki boyutlu dizin yaratma

```
import numpy as np

matris_2d = np.array([[1,2] , [3,4]])

print( matris_2d )
```

```
[[1 2]
 [3 4]]
```

02. Python'da İşlemler :

PYTHON

İşlem
Döngü
karar verme
matematik
fonksiyonlar
fonksiyoneller
matris
istatistik

02.6.1 Çok boyutlu dizin = Matris işlemleri

Belirli matris yaratma

```
matris_1d_range = np.arange( 10 )

print( matris_1d_range)
```

```
[0 1 2 3 4 5 6 7 8 9]
```

```
matris_sifirlar = np.zeros(5)

print( matris_sifirlar)
```

```
[0. 0. 0. 0. 0.]
```

```
matris_birler = np.ones(5)

print( matris_birler)
```

```
[1. 1. 1. 1. 1.]
```

02. Python'da İşlemler :

PYTHON

İşlem
Döngü
karar verme
matematik
fonksiyonlar
fonksiyoneller
matris
istatistik

02.6.1 Çok boyutlu dizin = Matris işlemleri

Belirli matris yaratma

```
matris_birim = np.eye(3)

print(matris_birim)
```

```
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
```

02. Python'da İşlemler :

PYTHON

İşlem
Döngü
karar verme
matematik
fonksiyonlar
fonksiyoneller
matris
istatistik

02.6.1 Çok boyutlu dizin = Matris işlemleri

Belirli matris yaratma

```
matris_diagonal = np.diag([1,2,3,4])

print( matris_diagonal)
```

```
[[1 0 0 0]
 [0 2 0 0]
 [0 0 3 0]
 [0 0 0 4]]
```

02. Python'da İşlemler :

PYTHON

İşlem

Döngü

karar verme

matematik

fonksiyonlar

fonksiyoneller

matris

istatistik

02.6.1 Çok boyutlu dizin = Matris işlemleri

Belirli matris yaratma

```
matris_1d_range = np.arange( 10 )

print( matris_1d_range)
```

```
[0 1 2 3 4 5 6 7 8 9]
```

```
matris_1d_range_2boyut = matris_1d_range.reshape(5,2)

print( matris_1d_range_2boyut)
```

```
[[0 1]
 [2 3]
 [4 5]
 [6 7]
 [8 9]]
```

02. Python'da İşlemler :

PYTHON

İşlem
Döngü
karar verme
matematik
fonksiyonlar
fonksiyoneller
matris
istatistik

02.6.1 Çok boyutlu dizin = Matris işlemleri

Matris boyutlarının gösterimi

```
matris_1d_range_2boyut.shape
```

```
(5, 2)
```

Eleman sayısını gösterme

```
matris_1d_range_2boyut.size
```

```
10
```

02. Python'da İşlemler :

PYTHON

İşlem
Döngü
karar verme
matematik
fonksiyonlar
fonksiyoneller
matris
istatistik

02.6.2 Matris Erişim

Bir boyutlu matris erişim

```
matrisim = np.arange(1,100,10)
```

```
matrisim[:]
```

```
array([ 1, 11, 21, 31, 41, 51, 61, 71, 81, 91])
```

```
matrisim[1]
```

```
11
```

02. Python'da İşlemler :

PYTHON

İşlem

- Döngü
- karar verme
- matematik
- fonksiyonlar
- fonksiyoneller
- matris
- istatistik

02.6.2 Matris Erişim

İki boyutlu matris erişim

```
matrisim_2b = np.array([[1,3,5] , [2,4,6] , [3,5,7]])

print( matrisim_2b)
```

```
[[1 3 5]
 [2 4 6]
 [3 5 7]]
```

```
matrisim_2b[:,0]
```

```
array([1, 2, 3])
```

```
matrisim_2b[:2  , 1 ]
```

```
array([3, 4])
```

02. Python'da İşlemler :

PYTHON

İşlem
Döngü
karar verme
matematik
fonksiyonlar
fonksiyoneller
matris
istatistik

02.6.2 Matris Erişim

Matris güncelleme

```
matrisim_2b[1,1] = 100
print(matrisim_2b)
```

```
[[  1   3   5]
 [  2 100   6]
 [  3   5   7]]
```

02. Python'da İşlemler :

PYTHON

İşlem
Döngü
karar verme
matematik
fonksiyonlar
fonksiyoneller
matris
istatistik

02.6.3 Matris Lineer Cebir İşlemleri

Transpoze alma

```
matrisim_2b.transpose()
```

```
array([[  1,   2,   3],
       [  3, 100,   5],
       [  5,   6,   7]])
```

02. Python'da İşlemler :

PYTHON

İşlem
Döngü
karar verme
matematik
fonksiyonlar
fonksiyoneller
matris
istatistik

02.6.3 Matris Lineer Cebir İşlemleri

Tersini alma - Karmaşık işlem olduğundan "linalg" sınıfı kullanılmaktadır.

```
1  np.linalg.inv( matrisim_2b )
```

```
array([[-0.87239583, -0.00520833,  0.62760417],
       [-0.00520833,  0.01041667, -0.00520833],
       [ 0.37760417, -0.00520833, -0.12239583]])
```

02. Python'da İşlemler :

PYTHON

İşlem
Döngü
karar verme
matematik
fonksiyonlar
fonksiyoneller
matris
istatistik

02.6.3 Matris Lineer Cebir İşlemleri

Toplama : Eleman elemana

```
birinci_matris = np.array([[1,9] , [0,10]])
ikinci_matris = np.array([[5,4] , [2,7]])

sonuc = np.add( birinci_matris , ikinci_matris )

print( sonuc)

```

```
[[ 6 13]
 [ 2 17]]
```

02. Python'da İşlemler :

PYTHON

İşlem
Döngü
karar verme
matematik
fonksiyonlar
fonksiyoneller
matris
istatistik

02.6.3 Matris Lineer Cebir İşlemleri

Çıkarma : Eleman elemana

```
birinci_matris = np.array([[1,9] , [0,10]])
ikinci_matris = np.array([[5,4] , [2,7]])

sonuc = np.subtract(birinci_matris , ikinci_matris)

print( sonuc)

```

```
[[-4  5]
 [-2  3]]
```

02. Python'da İşlemler :

PYTHON

İşlem
Döngü
karar verme
matematik
fonksiyonlar
fonksiyoneller
matris
istatistik

02.6.3 Matris Lineer Cebir İşlemleri

Çarpma : Eleman elemana

```
birinci_matris = np.array([[1,9] , [0,10]])
ikinci_matris = np.array([[5,4] , [2,7]])

sonuc = np.multiply(birinci_matris , ikinci_matris )

print( sonuc)

```

```
[[ 5 36]
 [ 0 70]]
```

02. Python'da İşlemler :

PYTHON

İşlem
Döngü
karar verme
matematik
fonksiyonlar
fonksiyoneller
matris
istatistik

02.6.3 Matris Lineer Cebir İşlemleri

Bölme : Eleman elemana

```
birinci_matris = np.array([[1,9] , [0,10]])
ikinci_matris = np.array([[5,4] , [2,7]])

sonuc = np.divide(birinci_matris , ikinci_matris )

print( sonuc)

```

```
[[0.2        2.25      ]
 [0.         1.42857143]]
```

02. Python'da İşlemler :

PYTHON

İşlem
Döngü
karar verme
matematik
fonksiyonlar
fonksiyoneller
matris
istatistik

02.6.3 Matris Lineer Cebir İşlemleri

Çarpma : Cebirsel çarpım

```
birinci_matris = np.array([[1,9] , [0,10]])
ikinci_matris = np.array([[5,4] , [2,7]])

sonuc = np.dot(birinci_matris , ikinci_matris)

print( sonuc)

```

```
[[23 67]
 [20 70]]
```

02. Python'da İşlemler :

PYTHON

İşlem
Döngü
karar verme
matematik
fonksiyonlar
fonksiyoneller
matris
istatistik

02.6.3 Matris Lineer Cebir İşlemleri

Determinant

```
matrisim =  np.array([[1,2,3,4],
                      [10,11,12,2],
                      [3,4,5,6],
                      [2,1,4,5]])

np.linalg.det( matrisim )
```

```
88.00000000000003
```

02. Python'da İşlemler :

PYTHON

İşlem
Döngü
karar verme
matematik
fonksiyonlar
fonksiyoneller
matris
istatistik

02.7.1- Temel istatistikler

En büyük, küçük

```
matrisim =  np.array([[1,2,3,4],
                      [10,11,12,2],
                      [3,4,5,6],
                      [2,1,4,5]])

print( matrisim.max() )
print( matrisim.min() )
```

```
12
1
```

02. Python'da İşlemler :

PYTHON

İşlem

Döngü

karar verme

matematik

fonksiyonlar

fonksiyoneller

matris

istatistik

02.7.1- Temel istatistikler

Ortalama değer ve Standart Sapma

```
matrisim =  np.array([[1,2,3,4],
                      [10,11,12,2],
                      [3,4,5,6],
                      [2,1,4,5]])

print( matrisim[0,:].mean() )

print( matrisim[0,:].std() )
```

```
2.5
1.118033988749895
```

02. Python'da İşlemler :

PYTHON

https://flic.kr/p/KBmDAL

03. Python'da Veri Okuma/Yazma :

PYTHON

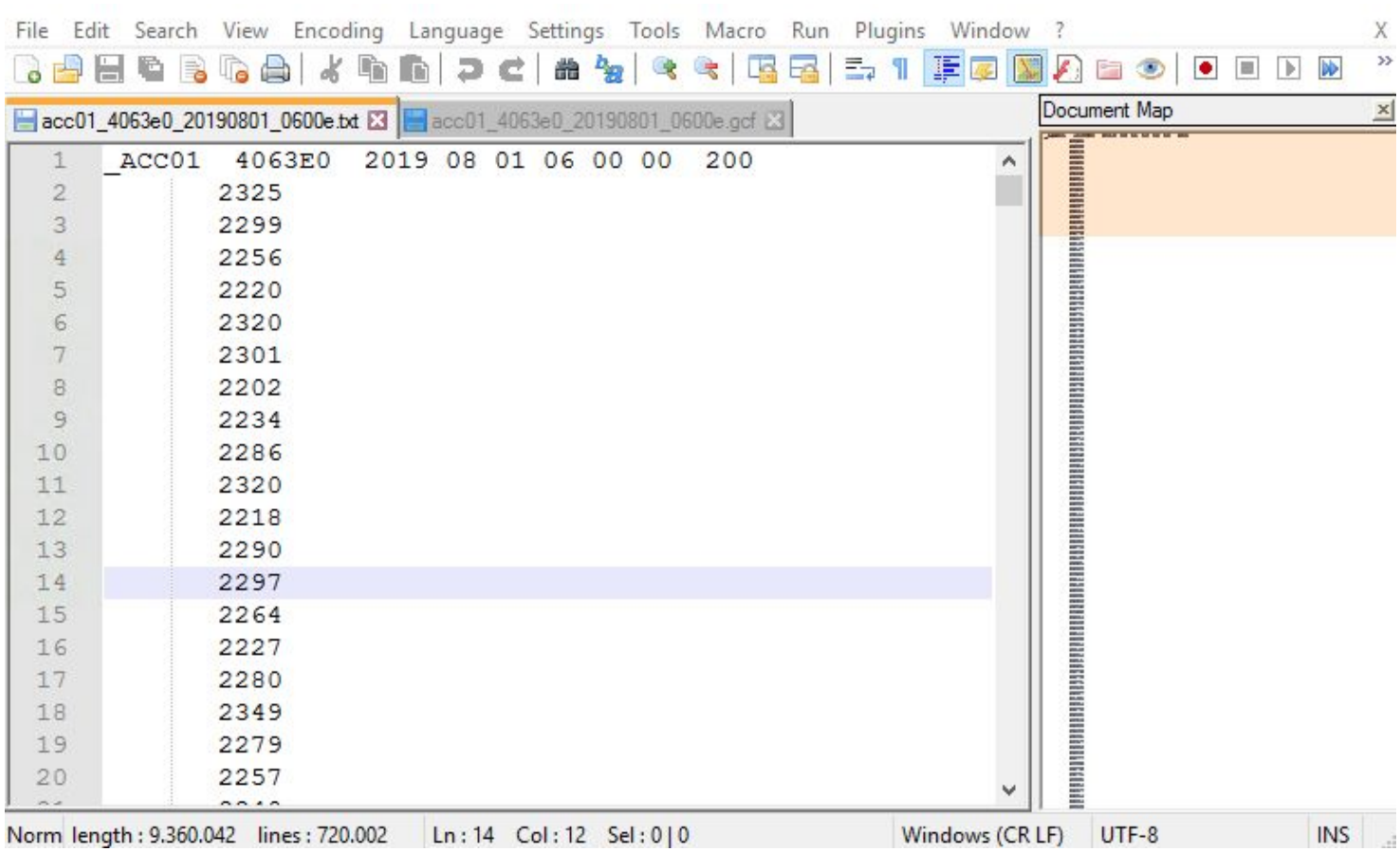
Değişken	İşlem	Okuma/Yazma
• yazı (string) • tamsayı (integer) • ondalık (float) • liste (list) • sözlük (dictionary) • demet (tuple) • erişim	• Döngü • karar verme • matematik • fonksiyonlar • fonksiyoneller • matris • istatistik	• ASCII • BINARY • OPEN • NUMPY • PANDAS • PICKLE • MAT

03. Python'da Veri Okuma/Yazma :

PYTHON

Okuma Yazma

- ASCII
- BINARY
- OPEN
- NUMPY
- PANDAS
- PICKLE
- MAT



03. Python'da Veri Okuma/Yazma :

PYTHON

Okuma Yazma

ASCII

<u>BINARY</u>

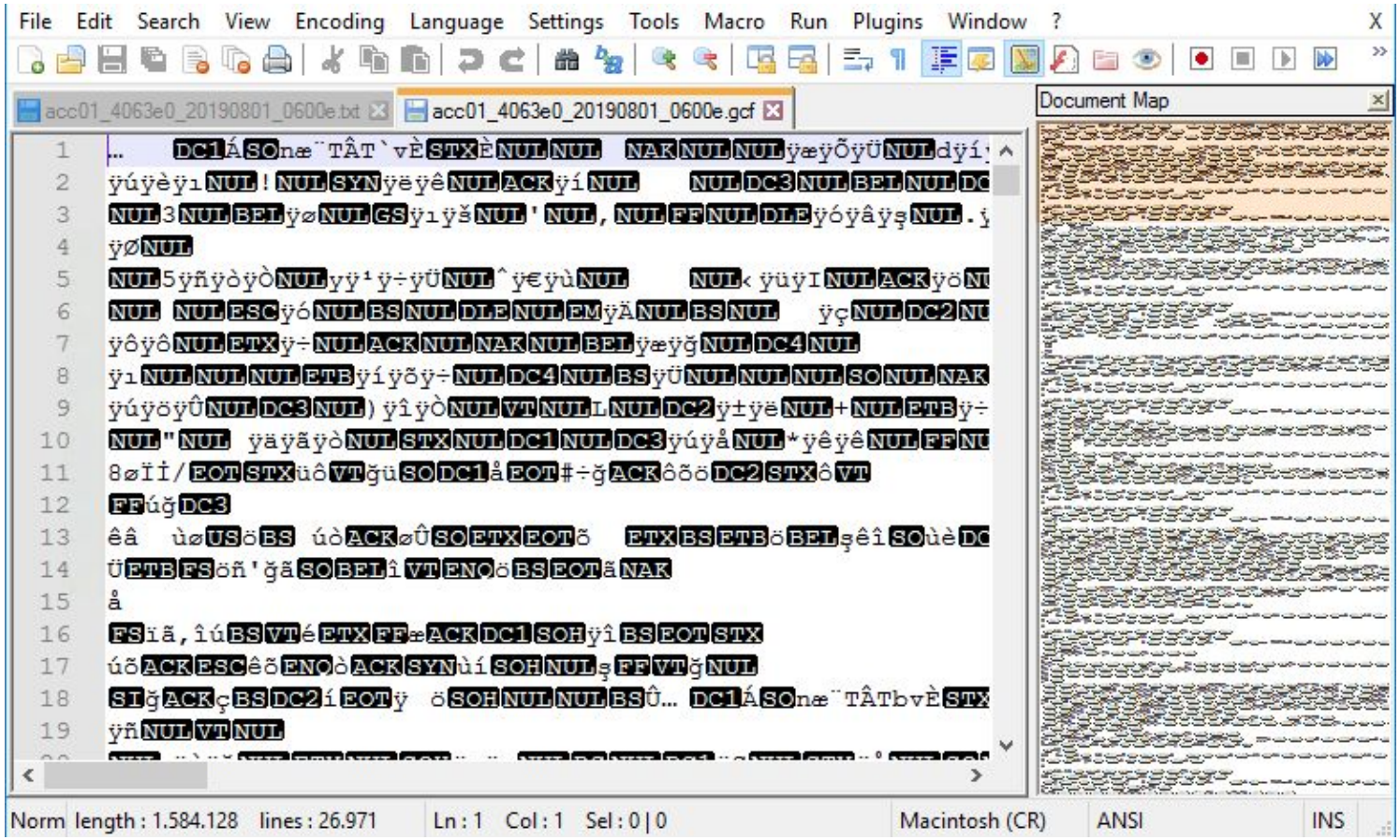
OPEN

NUMPY

PANDAS

PICKLE

MAT



03. Python'da Veri Okuma/Yazma :

PYTHON

Okuma Yazma

ASCII

BINARY

OPEN

NUMPY

PANDAS

PICKLE

MAT

Name	Type	Size
acc01_4063e0_20190801_0600e.gcf	GCF File	1.547 KB
acc01_4063e0_20190801_0600e.txt	TXT File	9.141 KB
acc01_4063n0_20190801_0600n.gcf	GCF File	1.787 KB
acc01_4063n0_20190801_0600n.txt	TXT File	9.141 KB
acc01_4063z0_20190801_0600z.gcf	GCF File	1.762 KB
acc01_4063z0_20190801_0600z.txt	TXT File	9.141 KB
acc02_4055e0_20190801_0600e.gcf	GCF File	1.497 KB
acc02_4055e0_20190801_0600e.txt	TXT File	9.141 KB

03. Python'da Veri Okuma/Yazma :

PYTHON

Okuma Yazma

ASCII

BINARY

OPEN

NUMPY

PANDAS

PICKLE

MAT

ASCII okuma

```
dosya = open( "acc01_4063e0_20190801_0600e.txt",  "r")
```

```
icerik = dosya.readlines()
```

```
type(icerik)
```

```
list
```

```
icerik[:5]
```

```
['_ACC01  4063E0  2019 08 01 06 00 00  200\n',
 '       2325\n',
 '       2299\n',
 '       2256\n',
 '       2220\n']
```

```
dosya.close()
```

03. Python'da Veri Okuma/Yazma :

PYTHON

Okuma Yazma

ASCII

BINARY

OPEN

NUMPY

PANDAS

PICKLE

MAT

BINARY okuma

```
dosya = open( "acc01_4063e0_20190801_0600e.gcf",  "rb")
```

```
icerik = dosya.readlines()
```

```
type(icerik)
```

```
list
```

```
icerik[:5]
```

```
[b'\x85\t\x11\xc1\x0en\xe6\xa8T\xc2T`v\xc8\x02\xc8\x00\x00\
\xff\xe6\xff\xd5\xff\xdc\x00d\xff\xed\xff\x9d\x00 \x004\x00'
\x00\x07\xff\xdf\xff\xdb\x005\x00E\xff\xba\xff\xea\xff\xef\
4\x00\x13\x00L\xff\xcb\xff\xcd\x00)\x00.\xff\xde\xff\xdb\x0
\xfa\xff\xfa\x00\r\xff\xfa\xff\xe8\xff\xfd\x00!\x00\x16\xff
00\x06\xff\xed\x00\t\x00\x13\x00\x07\x00\x11\x00\x06\xff\xd
\x00\x1d\x00\x0c\xff\xc0\x00\x1f\x00$\xff\xdd\xff\xb7\x00"\
ff\xde\x00&\x00\x1e\xff\xe0\xff\xf9\x00#\xff\xfb\xff\xd1\xf
0\xff\xfa\xff\xbd\x00&\x00\x17\xff\xc8\xff\xea\x00K\x00 \xf
\x004\xff\xf6\xff\xc5\x00\x1a\x00F\xff\xc6\x00\x05\xff\xff\
\'\x00\x15\x00*\xff\xa4\xff\xd5\x00\n',
 b'\x003\x00\x07\xff\xf8\x00\x1d\xff\xfd\xff\x9a\x00\'\x00,
```

03. Python'da Veri Okuma/Yazma :

PYTHON

Okuma Yazma

ASCII

BINARY

OPEN

NUMPY

PANDAS

PICKLE

MAT

Anlık okuma

```
with open( "acc01_4063e0_20190801_0600e.txt",  "r") as dosya :

    for satir in dosya.readlines():

        print(satir)
```

```
_ACC01  4063E0  2019 08 01 06 00 00  200

       2325

       2299

       2256
```

03. Python'da Veri Okuma/Yazma :

Okuma
Yazma

ASCII

BINARY

OPEN

NUMPY

PANDAS

PICKLE

MAT

PYTHON

03.3.2- Yazma

Verileri dosyaya yazma amacıyla kullanılır.

```
import numpy as np
listem = [ np.random.randn() for item in range(10)]

print( listem )
```

```
[-0.2965819842880848, -0.749426059303648, 0.9950813354208898, -1.47153361
06756207, 0.5340299308182604, -0.33909302640759315, -0.2371604389560698,
0.0842704040092735, -0.8376988516454047, 0.453560320832434]
```

```
with open("yeni bir dosya yaziyorum.txt" , "w") as dosya:
    for satir in listem:
        dosya.write( str( satir ) + "\n")
```

yeni bir dosya yaziyorum.txt Open with TextEdit

```
-0.2965819842880848
-0.749426059303648
0.9950813354208898
-1.4715336106756207
0.5340299308182604
-0.33909302640759315
-0.2371604389560698
0.0842704040092735
-0.8376988516454047
0.453560320832434
```

03. Python'da Veri Okuma/Yazma :

Okuma
Yazma

ASCII

BINARY

OPEN

NUMPY

PANDAS

PICKLE

MAT

PYTHON

03.3.2- Yazma

Verileri dosyaya yazma amacıyla kullanılır.

```python
with open("yeni bir dosya yaziyorum 2.txt" , "w") as dosya:
    for satir in listem:
        dosya.write( f"{satir} , {satir*2} \n")
```

yeni bir dosya yaziyorum 2.txt Open with TextEdit

```
-0.2965819842880848 , -0.5931639685761696
-0.749426059303648 , -1.498852118607296
0.9950813354208898 , 1.9901626708417797
-1.4715336106756207 , -2.9430672213512414
0.5340299308182604 , 1.068059861636521
-0.33909302640759315 , -0.6781860528151863
-0.2371604389560698 , -0.4743208779121396
0.0842704040092735 , 0.168540808018547
-0.8376988516454047 , -1.6753977032908094
0.453560320832434 , 0.907120641664868
```

03. Python'da Veri Okuma/Yazma :

PYTHON

Okuma Yazma

ASCII

BINARY

OPEN

NUMPY

PANDAS

PICKLE

MAT

03.4- NUMPY :

Okuma örneği

```
import numpy as np

verilerim = np.loadtxt( "yeni bir dosya yaziyorum.txt" )

type(verilerim)
```

```
numpy.ndarray
```

```
verilerim
```

```
array([-0.29658198, -0.74942606,  0.99508134, -1.47153361,  0.53402993,
       -0.33909303, -0.23716044,  0.0842704 , -0.83769885,  0.45356032])
```

03. Python'da Veri Okuma/Yazma :

PYTHON

Okuma Yazma

ASCII

BINARY

OPEN

NUMPY

PANDAS

PICKLE

MAT

03.4- NUMPY :

Okuma örneği

```
import numpy as np

verilerim_2 = np.loadtxt( "yeni bir dosya yaziyorum 2.txt" , delimiter = "," )

type(verilerim_2)
```

```
numpy.ndarray
```

```
verilerim_2
```

```
array([[-0.29658198, -0.59316397],
       [-0.74942606, -1.49885212],
       [ 0.99508134,  1.99016267],
       [-1.47153361, -2.94306722],
       [ 0.53402993,  1.06805986],
       [-0.33909303, -0.67818605],
       [-0.23716044, -0.47432088],
       [ 0.0842704 ,  0.16854081],
       [-0.83769885, -1.6753977 ],
       [ 0.45356032,  0.90712064]])
```

03. Python'da Veri Okuma/Yazma :

PYTHON

Okuma Yazma

ASCII

BINARY

OPEN

NUMPY

PANDAS

PICKLE

MAT

03.4- NUMPY :

```
listem =  [ np.random.randn() for item in range(10)]

np.savetxt ( "veri yazma.csv", [ listem ] , delimiter=",")

```

veri yazma.csv Open with Microsoft Excel

-1.7771758016631178 38e+00	1.0972478498680195 98e+00	-7.9683744

03. Python'da Veri Okuma/Yazma :

PYTHON

Okuma Yazma

ASCII

BINARY

OPEN

NUMPY

PANDAS

PICKLE

MAT

03.5- Pandas: Okuma

Pandas çok güçlü bir veri okuma aracıdır. Çok mucizeler yaratır. Veri tabanı yapısına benzer bir yapısı vardır.

```python
import pandas as pd

veri_df = pd.read_csv( "ilk pandas okuma.txt")

print( type( veri_df ))
```

```
<class 'pandas.core.frame.DataFrame'>
```

```python
veri_df
```

	kolon1	kolon2	kolon3
0	1	2	3
1	10	20	30
2	100	300	300

03. Python'da Veri Okuma/Yazma :

Okuma Yazma

ASCII

BINARY

OPEN

NUMPY

PANDAS

PICKLE

MAT

PYTHON

03.5- Pandas: Okuma

Pandas çok güçlü bir veri okuma aracıdır. Çok mucizeler yaratır. Veri tabanı yapısına benzer bir yapısı vardır.

```
1  import pandas as pd
2
3  veri_df = pd.read_csv( "ilk pandas okuma.txt")
4
5  print( type( veri_df ))
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
1  veri_df
```

	kolon1	kolon2	kolon3
0	1	2	3
1	10	20	30
2	100	300	300

03. Python'da Veri Okuma/Yazma :

PYTHON

Okuma Yazma

ASCII

BINARY

OPEN

NUMPY

PANDAS

PICKLE

MAT

03.5- Pandas: Okuma

Pandas mucizesi görelim mi?

```
1  veri_df.describe()
```

	kolon1	kolon2	kolon3
count	3.000000	3.000000	3.000000
mean	37.000000	107.333333	111.000000
std	54.744863	167.096778	164.234588
min	1.000000	2.000000	3.000000
25%	5.500000	11.000000	16.500000
50%	10.000000	20.000000	30.000000
75%	55.000000	160.000000	165.000000
max	100.000000	300.000000	300.000000

03. Python'da Veri Okuma/Yazma :

PYTHON

Okuma Yazma

ASCII

BINARY

OPEN

NUMPY

PANDAS

PICKLE

MAT

03.5- Pandas: Yazma

Pandas çok farklı şekillerde yazım yapabilir.

```
sozluk_hatirla = {"ilki":[1,2,3] , "ikincisi":[10,20,30]}

sozluk_df = pd.DataFrame( sozluk_hatirla)

sozluk_df
```

	ilki	ikincisi
0	1	10
1	2	20
2	3	30

```
sozluk_df.to_csv("ikinci panda yazmam.txt")
```

ikinci panda yazmam.txt Open with TextEdit

```
,ilki,ikincisi
0,1,10
1,2,20
2,3,30
```

03. Python'da Veri Okuma/Yazma :

PYTHON

Okuma Yazma

ASCII

BINARY

OPEN

NUMPY

PANDAS

PICKLE

MAT

03.6- Pickle : Yazma

Eğer üretilen veya okunan verileri yani değişkenleri daha sonra kullanmak üzere saklamak istiyorsanız "turşu" yapımı işinize yarayacaktır :)

```
1  veri_1 = [1,11,21,31,41]
2  veri_2 = ["adem", "madem" , "badem"]
3
4  import pickle
5
6  dosya_ismi = open("ilk tursu.pkl" , "wb")
7  pickle.dump((veri_1,veri_2) , dosya_ismi )
8  dosya_ismi.close
```

03. Python'da Veri Okuma/Yazma :

PYTHON

Okuma Yazma

ASCII

BINARY

OPEN

NUMPY

PANDAS

PICKLE

MAT

03.6- Pickle : Okuma

Verinin okunması demet olarak olmaktadır.

```
d = open("ilk tursu.pkl" , "rb")
vv = pickle.load(d)
d.close()
```

```
type(vv)
```

```
tuple
```

```
vv
```

```
([1, 11, 21, 31, 41], ['adem', 'madem', 'badem'])
```

03. Python'da Veri Okuma/Yazma :

Eski dost MATLAB

Okuma Yazma

ASCII

BINARY

OPEN

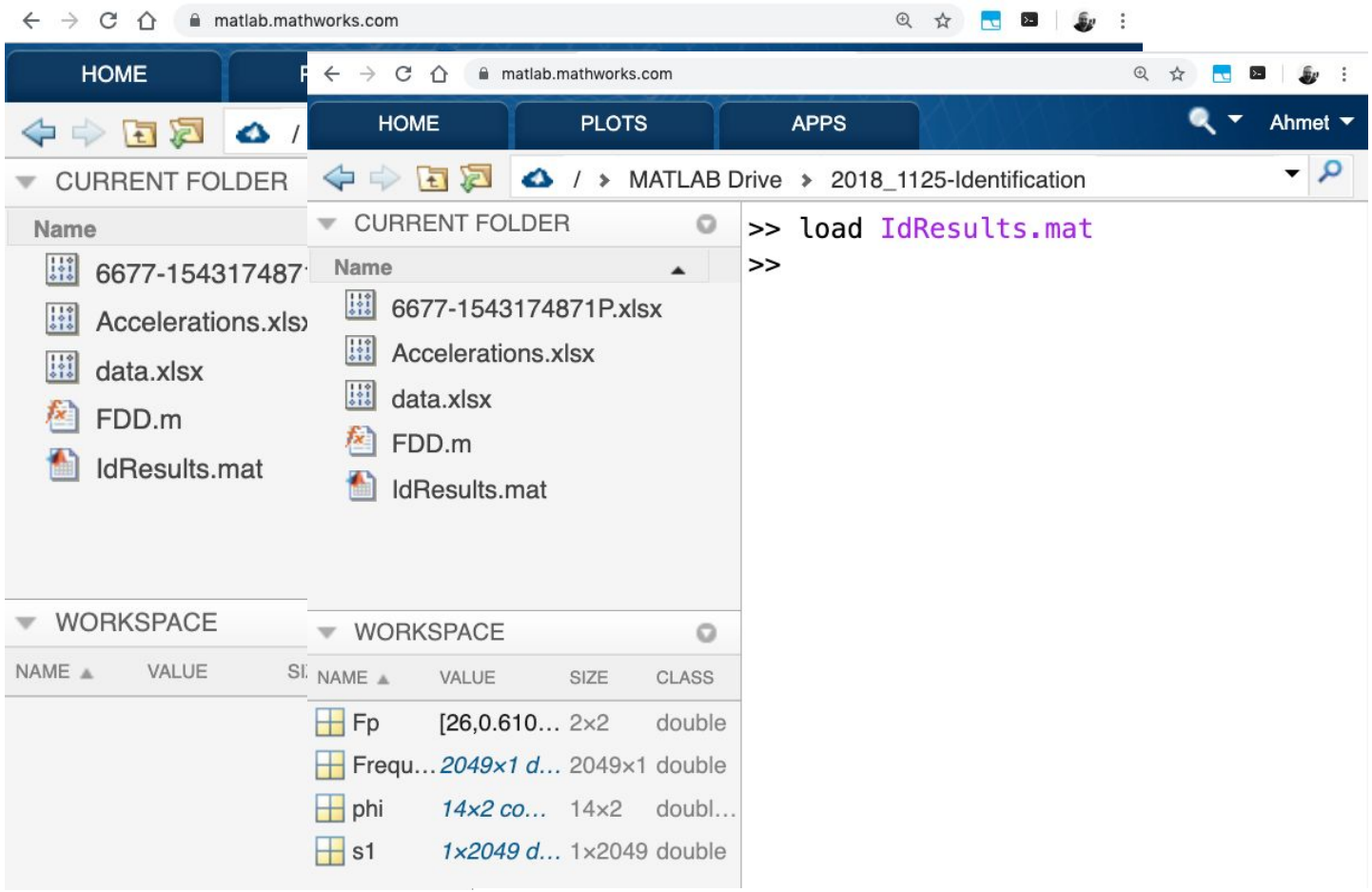
NUMPY

PANDAS

PICKLE

MAT

PYTHON



03. Python'da Veri Okuma/Yazma :

Eski dost MATLAB, yeni kanka Python :)

PYTHON

Okuma Yazma

ASCII

BINARY

OPEN

NUMPY

PANDAS

PICKLE

MAT

```python
import scipy.io
mat = scipy.io.loadmat('IdResults.mat')
```

```python
type(mat)
```

```
dict
```

```python
mat.keys()
```

```
dict_keys(['__header__', '__version__', '__globals__', 'phi', 'Frequencie
s', 'Fp', 's1'])
```

```python
mat["phi"]
```

```
array([[-5.42955901e-02+3.71605993e-17j, -3.40551166e-02+1.52704986e-17j],
       [-6.71222944e-02+7.13290493e-03j, -4.37361927e-02+3.39666292e-03j],
       [-5.25653547e-03-7.48580828e-03j,  5.12664765e-02-7.09038612e-04j],
       [ 1.93585913e-02+7.84105453e-03j,  1.60026217e-03-3.24258784e-04j],
       [-4.04083446e-01+2.18546219e-01j, -2.45882596e-01-2.31662766e-01j],
       [-4.03383389e-01+2.03065196e-01j, -1.45179030e-01-4.86197103e-01j],
       [-3.66600407e-01+2.04416369e-01j, -2.01538155e-01-2.01515078e-01j],
```

03. Python'da Veri Okuma/Yazma :

https://flic.kr/p/88EtHa

Sonuç :

1. Programlama bilimsel çalışmalarda kaçınılmaz...
2. Programlama mühendislik çalışmalarında kaçınılmaz...
3. Programlama gelecek için bir yatırımdır...
4. Kolaydan başlayıp hedefe odaklanarak ilerlemek gerekir...
5. Hayat boyu öğrenme sürecidir...
6. Bu çalıştay sizler için iyi bir başlangıçtır...
7. Scriptinize bug değmesin...

Sonuç :

https://flic.kr/p/baD7qH