**HAFTA 1**

**H11**

print("Ali gel")

#print("Ali "gel" dedi")#hata verir

print('Ali "gel" dedi')

#print('Ali'nin arkadaşı "gel" dedi') # hata verir

print('''Ali'nin arkadaşı "gel" dedi''') # en disa metin icinde olmayan birsey yaz

print("Ali'nin arkadaşı \"gel\" dedi").

**# \ ile alt satırdan devam edilebilir.**

print("Ali'nin \

arkadaşı \"gel\" dedi")

**Cikti 🡪** Ali'nin arkadaşı "gel" dedi

# end ile print sonundaki işaret

adi = "Ertan"

soyAdi = "ÖNER"

print(adi,end="--") # end in default değeri \n

print(soyAdi)

print() # end default \n olduğundan aşağıya geçer

print(end="---")

print(adi,end="")

print(soyAdi)

meyveler = ["elma","armut","kiraz"]

for meyve in meyveler:

    print(meyve,end="-x-")

print (len(meyveler )) #uzunluk.

**# \* ve sep ile elementler arasındaki işaret – ayirma parametresinin nasil olacagini anlatir**

print(\*"Python")

print(\*["muz","dut","nar"], sep="\n")

**Cikti 🡪**

muz

dut

nar

print(\*["muz","dut","nar"], sep="---")

**Cikti 🡪** muz---dut---nar

liste = ["muz","dut","nar"]

print(\*liste) 🡪 muz dut nar

print(\*liste, sep ="\n") 🡪 muz

dut

nar

print ("Ankara"\*2) 🡪 AnkaraAnkara

**f, format ve %s ile string oluşturma / String içinde değişken kullanma**

f, format ve %s ile stringler oluşturulabilir.

Bu stringler print, input gibi parametre olarak string alan fonksiyonlara verilebilir.

# ör1: f ile string oluşturma

ad = "Elif Yıldız"

bolum = "Yazılım Mühendisliği"

ortalama = 90

yazi = f"Sayın {ad}, {bolum} bölümünden, {ortalama} puan ortalamasıyla mezun oldunuz."

print(yazi)

# ör2: f ile string oluşturma

xx = 50; print(f"xx değeri: {xx}")

# ör3: format ile string oluşturma

ad= input("Adınızı giriniz: ")

print(f"Merhaba {ad}!")

print("Selam {}! {}".format(ad,num))

%s ile string içinde değişken

ad= input('Adınız nedir?\n')

print('Merhaba, %s. nasılsın?' % ad)

# f, format ve %

ad = "Erdinç DÖNMEZ"

not1 = 80

not2 = 90

print("Ad:",ad,", 1.Not:",not1)

print(f"Ad: {ad},  1.Not: {not1}")

print("Ad: {},  1.Not: {}".format(ad,not1))

print("Ad: %s,  1.Not: %s" %(ad, not1))

print(f"1.not {not1}, 1.not {not2}")

print("ortalama: {}".format((not1+not2)/2))

print("100'den kalan %s" %(100-(not1+not2)/2))

# %d, %s, %f

s 🡪 string

d 🡪 decimal (tam sayi olarak yazacak)

f 🡪 float olarak yazar

ad= input('Adınız nedir?\n')

print('Merhaba, %s. nasılsın?' % ad)

Cikti 🡪

Adınız nedir?

nur

Merhaba, nur. nasılsın?

**# çıktıda \n ile newline ya da yenisatır yapabiliriz.**

**# \t ile de düzenli uzun boşluk yapılabilir.**

ad = "Ahmet GÜL"

tc = "52654125785"

iban = "TR6542654265428754"

print("Müşteri bilgileri===============")

print("Adı:",ad)

print("TCKN:",tc)

print("İban no:",iban)

print()

print("Müşteri bilgileri\n===============") **🡪 uzun bosluk birakmak icin**

print("Adı:\t",ad)

print("TCKN:\t",tc)

print("İbanno:\t",iban)

print()

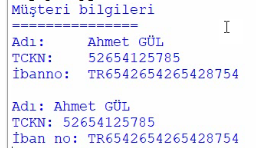
print("Müşteri bilgileri\n===============")

print("Adı:",ad)

print("TCKN: ",tc)

print("İban no:",iban)

**Cikti 🡪**



|  |  |
| --- | --- |
|  |  |

Ascii table <https://www.ascii-code.com/>

Solo learn 🡪 Pratik yapmak icin

Foldercode 🡪 24 saatligine dosyalari koymak icin

|  |  |
| --- | --- |
| print("\033[1;32;40m")  print("╔═════════════════════╗")  print("║     VEKTOREL APP    ║")  print("║                     ║")  print("║  1-Toplama          ║")  print("║  2-Çıkarma          ║")  print("║  3-                 ║")  print("║  4-                 ║")  print("║                     ║")  print("║    Seçimiz nedir?   ║")  print("╚═════════════════════╝")  print("\033[0m") | A black square with green text  AI-generated content may be incorrect. |
| print("\033[1;35;40m")  #print("╔"+"═"\*20+"╗")  print("╔═════════════════════╗")  print("║\033[1;33;40m    VEKTOREL APP   \033[1;35;40m  ║")  print("║                     ║")  print("║  1-Toplama          ║")  print("║  2-Çıkarma          ║")  print("║  3-                 ║")  print("║  4-                 ║")  print("║                     ║")  print("║    Seçimiz nedir?   ║")  print("╚═════════════════════╝") | A screenshot of a black screen  AI-generated content may be incorrect. |
| print("\033[1;33;40m")  print("╔═════════════════════╗")  print("║    HESAP MAKINESI   ║")  print("║                     ║")  print("║  1-Toplama          ║")  print("║  2-Cikarma          ║")  print("║  3-Carpma           ║")  print("║  4-Bolme            ║")  print("║                     ║")  print("║    Seçimiz nedir?   ║")  print("╚═════════════════════╝")  print("\033[0m") | A black and yellow rectangular sign with white text  AI-generated content may be incorrect. |
| print("\033[1;33;40m")  #print("╔"+"═"\*20+"╗")  print("╔═════════════════════╗")  print("║\033[1;35;40m     HESAP MAK     \033[1;33;40m  ║")  print("║                     ║")  print("║  1-Toplama          ║")  print("║  2-Çıkarma          ║")  print("║  3-                 ║")  print("║  4-                 ║")  print("║                     ║")  print("║    Seçimiz nedir?   ║")  print("╚═════════════════════╝") |  |
| #print("╔"+"═"\*20+"╗")  print("╔═════════════════════╗")  print("║ VEKTOREL APP ║")  print("║ ║")  print("║ 1-Hesaplamalar ║")  print("║ 2-Oyunlar ║")  print("║ 3-Çizimler ║")  print("║ 4- ║")  print("║ 5- ║")  print("║ 6- ║")  print("║ 7- ║")  print("║ 8- ║")  print("║ 9- ║")  print("║ 10- ║")  print("║ ║")  print("║ Seçimiz nedir? ║")  print("╚═════════════════════╝")  secim = input() **# klavyeden bilgi alacagiz**  print("Seçiminiz:",secim,"idi.")  if secim == "1":  print("Hesaplamaları seçtiniz.")  import projeoyunlarmenu    if secim == "2": print("Oyunlar seçtiniz.") | ╔═════════════════════╗  ║ VEKTOREL APP ║  ║ ║  ║ 1-Hesaplamalar ║  ║ 2-Oyunlar ║  ║ 3-Çizimler ║  ║ 4- ║  ║ 5- ║  ║ 6- ║  ║ 7- ║  ║ 8- ║  ║ 9- ║  ║ 10- ║  ║ ║  ║ Seçiminiz nedir? ║  ╚═════════════════════╝ |
| **projeoyunlarmenu.py olarak kaydet**  #print("╔"+"═"\*20+"╗")  print("╔═════════════════════╗")  print("║ OYUNLAR MENU ║")  print("║ ║")  print("║ 1-Tetris ║")  print("║ 2-Adam asmaca ║")  print("║ 3-Yılan ║")  print("║ 4- ║")  print("║ 5- ║")  print("║ 6-Ana menüye dön ║")  print("║ ║")  print("║ Seçimiz nedir? ║")  print("╚═════════════════════╝")  ss = input()  if ss == "1" : print("Tetris başlıyor..")  if ss == "6":  import proje\_ana\_ekrani |  |
| print("╔═════════════════════╗")  print("║ OYUNLAR MENU ║")  print("║ ║")  print("║ 1-Tetris ║")  print("║ 2-Adam asmaca ║")  print("║ 3-Yılan ║")  print("║ 4- ║")  print("║ 5- ║")  print("║ 6-Ana menüye dön ║")  print("║ ║")  print("║ Seçimiz nedir? ║")  print("╚═════════════════════╝")  ss = input()  if ss == "1" : print("Tetris başlıyor..")  if ss == "6":  import 10\_proje\_ana\_ekrani |  |

**H12**

# ; işareti ile komutlar aynı satırda yazılabilir.

print("Merhaba"); print("Nasısın") ;input()

print("Merhaba\

Nasılsın") # editörde alt satırda devam için \ kullan

print("Merhaba\nNasılsın") # çıktıda alt satıra geçmek için \n

# fonksiyon içinde başka fonksiyon kullanılabilir.

# bazı fonksiyonlar bize bir şey verir.

# fonksiyon içinde kullandığımız fonksiyonların verdiği çıktıların ne olduğu önemli.

# print içine; sayi, metin, değişken, mantıksal değer.

# sadece stringler tırnak içinde yazılır.

print(True)

print(5>8)

ahmet = True

mehmet = 8>5

print(ahmet == mehmet)

| **Shape** | **Unicode** | **Name** | **Windows Alt Code\*** |
| --- | --- | --- | --- |
| ┌ | U+250C | Box Drawings Light Down and Right | Alt + 218 |
| ┐ | U+2510 | Box Drawings Light Down and Left | Alt + 191 |
| └ | U+2514 | Box Drawings Light Up and Right | Alt + 192 |
| ┘ | U+2518 | Box Drawings Light Up and Left | Alt + 217 |
| ─ | U+2500 | Box Drawings Light Horizontal | Alt + 196 |
| │ | U+2502 | Box Drawings Light Vertical | Alt + 179 |

| **Shape** | **Unicode** | **Name** | **Windows Alt Code** |
| --- | --- | --- | --- |
| ╔ | U+2554 | Box Drawings Double Down and Right | Alt + 201 |
| ╗ | U+2557 | Box Drawings Double Down and Left | Alt + 187 |
| ╚ | U+255A | Box Drawings Double Up and Right | Alt + 200 |
| ╝ | U+255D | Box Drawings Double Up and Left | Alt + 188 |
| ═ | U+2550 | Box Drawings Double Horizontal | Alt + 205 |
| ║ | U+2551 | Box Drawings Double Vertical | Alt + 186 |

aa = "Ankara "

cc = "Çankaya "

bb = 5

d = 6

print(aa)

print(aa\*3)

print(bb\*3)

print("╔","═"\*10,"╗")

print("╔"+"═"\*10+"╗")

print(aa+cc)

print(bb+d)

a = "Ankara "

print(len("Vektorel"))

print(len(a))

print(a\*len(a)) **🡪 7 kez Ankara yazar**

print(input("Şehir adı gir:"))

print("Muş","Van","Bor")

print("Muş","Van","Bor",sep=" ")# ayırma default boşluktur

print("Muş","Van","Bor",sep="-")

print("Muş","Van","Bor",sep="-",end="\n") **# end default değeri \n**

print("Muş","Van","Bor",sep="-",end="\*") **# en son satir sonunu \* ile bitir**

print("Muş","Van","Bor",sep="-",end="\*")

print("Muş","Van","Bor",sep="-",end="\*")

a = 5

b = 6

c = "3"

d = "4"

print(a\*2)

print(c\*2)

print(int(c)\*2)

print(str(a)\*2)

print((str(a)\*2)\*2)

a = "Ankara"

b = "Vektörel"

c = a+b

c = f"{a} ile {b}" # f ile string formatlama

print(c)

print(f"{a} ile {b}")

c = "{} ile2 {}".format(a,b)

print(c)

c = "%s ile3 %s %d %f" %(a,b,5,3)

print(c)

a = "Ankara" # stringler bir tür dizidir.

b = "Vektörel"

c = a+b

c = f"{a} ile {b}" # f ile string formatlama

print(c)

print(f"{a} ile {b}")

c = "{} ile2 {}".format(a,b)

print(c)

c = "%s ile3 %s %d %f" %(a,b,5,3)

print(c)

print(a[0]) # index/indis hafızadaki yeri ifade eder.

print(b[2:4])

print(b[2:])

print(b[:4])

print(b[2:9:2]) # başlangıç:bitiş:atlama miktari

print(b[-3]) # sağdan/sondan index

print(b[-3:])

print(b[:])

print(b[::-1]) # terse çevirir

meyveler = ["Muz","Dut","Nar"] #liste türü dizi

print(meyveler)

print(meyveler[2])

print(a) 🡪 Ankara

print(\*a) 🡪 A n k a r a

print(\*a,sep="-") 🡪 A-n-k-a-r-a

print(\*a,sep="\n") 🡪 Her harfi alt alta yazar

print(\*a[::-1],sep="\n") 🡪 Kelimeyi tersten olacak sekilde her harfi alt alta yazar

##01234

**Aciklama Metinleri**

Tüm program dillerinde olduğu gibi “PYTHON”da da açıklama metinleri yazılabilir.

Program kodları dışındaki, daha sonra hatırlanması veya bilgi verilmesi amacıyla açıklama yazabilir.

Bazı durumlarda komut satırlarını devre dışı bırakmak için kullanılırlar.

Açıklama metni yazmak için;

Tek satırlık bir açıklama için # ifadesi kullanılır.

Çok satırlı açıklamalar için " " "  üç adet tek veya çift tırnak arasına yazılır.

sayi1 = input("1.Sayıyı gir:")

sayi2 = input("2.Sayıyı gir:")

sonuc = int(sayi1)+int(sayi2)

print("Sonuç: ",sonuc)

baslik = "Yaş hesaplama programi"

print(f"{baslik}\n{'═'\*len(baslik)}") **🡪 Altinda baslik uzunlugu kadar bu cizgi olacak**

ad = input("Adın ne?")

print(f"Adın {ad} demek. Çok güzel bir isim.")

dy = int(input(f"Hangi yılda doğdun {ad}?"))

print(f"Demek {2025-dy} yaşındasın {ad}.")

ortalama = int(input("Ortalaman kaç?"))

if ortalama>=50:

print("Sınıfı geçtin")

else :

print("Sınıfta kaldın")

ortalama = int(input("Ortalaman kaç?"))

if ortalama>=90:

print("Süpersin. Taktir aldın")

elif ortalama>=50: print("Sınıfı geçtin.")

else :

print("Sınıfta kaldın")

**Git / GitHub**

+ 🡪 New repository 🡪 -asagidakiler- 🡪 Public ve Readme seceneklerini sec

1. **Dosya** 🡪 Kendi adinla
2. **Dosya** 🡪 Ornekler icin – Vektorel Python
3. **Dosya** 🡪 Projeler icin (4 tane) – Console / Project

**Bilgileri aktarmak icin**

Ana sayfaya gel

New (yesil) 🡪 VektorelPython99 Public/Add readme / Print🡪

Ana sayfaya gel

+ 🡪 add file 🡪 upload a new file 🡪 surucudeki iki kilasoru at 🡪 commit mesaji yaz (dersin adi) 🡪 commit changes a tikla

VsCode

Dosya dizgininde goster

VsCode 🡪 source control 🡪 initialize repository (bilgisayara ilk kez repository kuruluyor) 🡪

File 🡪 new window 🡪 clone git repository

Ana sayfana gel 🡪 repositories 🡪 bir repository sec 🡪 yesil dugme ile indir ya da code daki yolu kopyala 🡪 vscode ac 🡪 solde search repository 🡪 file new window 🡪 clone git repository 🡪 hedef kilasoru goster 🡪

Github da yapilani kilasore nasil gonderiyoruz?

Dosyayi sec 🡪 vs code sag kisma gonder 🡪 commit (isim ver) 🡪 isim verirken yetkilendirme uyarisi gelecek 🡪 output a git 🡪 ilk satirdaki kodu kopyala 🡪 terminale yapistir 🡪 github email yaz 🡪 outputtan 2. kodu kopyala 🡪 terminale git username yaz 🡪 commit e git 🡪 synch changes 🡪



readme.md

**H22**

def menu():

    print("1-Oyun")

    print("2-Çizim")

    secim = input("Seçiminiz:")

    if secim == "1":oyunmenu()

    if secim == "2":

        print("Güzel çizimler")

        cizimmenu()

    if int(secim) < 1 or int(secim)>2:

        print("Yanlış seçim")

    menu()

def oyunmenu():

    print("1-Yılan")

    print("2-Tetris")

    secim = input("Seçiminiz:")

    if secim == "1":print("Yılan oyunu kodlar")

    if secim == "2":

        print("Tetris oyununa geçildi")

        # cizimmenu()

def cizimmenu():

    print("Kare")

    print("Üçgen")

print("Vektorel App")

# oyunmenu()

menu()

def anamenu():

print("\n\nANA MENU")

print("1-Hesaplamalar")

print("2-Oyunlar")

print("3-Çizimler")

s = input("Seçiminiz?")

if s=="1":

import cc\_hesaplamalar

cc\_hesaplamalar.hesapmenu()

if s=="2":

import bb\_oyunlar

if s=="3":

import d\_cizimler

d\_cizimler.cizim\_menu()

anamenu()

anamenu()

print("Oyunlar isimli dosya kodları")

print("Oyunlar isimli dosya kodları1")

def hesapmenu():

print("1-Toplama")

print("2-Çıkarma")

s = input("Seçiminiz?")

if s =="1": topla()

if s =="2": cikar()

def topla():

print("2+5=",2+5)

def cikar():

print("5-3=",5-3)

# bb\_oyunlar.py dosyası içeriği

# print("Oyunlar isimli dosya kodları")

# print("Oyunlar isimli dosya kodları1")

def cizim\_menu():

print("\n\nÇİZİM MENU")

print("1-Kare")

print("2-Desen")

print("3-Üçgen")

s = input("Seçimiz?")

if s=="1": kareciz()

if s=="3": ucgenciz()

def kareciz():

import turtle

for xx in range(4):

turtle.forward(100)

turtle.right(90)

def ucgenciz():

import turtle

for a in range(3):

turtle.forward(100)

turtle.right(120)

def menu():

print("1-Oyun")

print("2-Çizim")

secim = input("Seçiminiz:")

if secim == "1":oyunmenu()

if secim == "2":

print("Güzel çizimler")

cizimmenu()

if int(secim) < 1 or int(secim)>2:

print("Yanlış seçim")

menu()

def oyunmenu():

print("1-Yılan")

print("2-Tetris")

secim = input("Seçiminiz:")

if secim == "1":print("Yılan oyunu kodlar")

if secim == "2":

print("Tetris oyununa geçildi")

# cizimmenu()

def cizimmenu():

print("Kare")

print("Üçgen")

print("Vektorel App")

# oyunmenu()

menu()

# print("Notun kaç:")

not1 = int(input("Notun kaç:"))

# print(not1\*2)

if not1>90 :

print("Süper..")

print("Notun 5")

elif not1 > 70 : print("Notun 4")

elif not1>50 : print("Geçtin")

else : print("Kaldın")

# ------ YUKARIDAKİNİN AYNISI

# print("Notun kaç:")

not1 = int(input("Notun kaç:"))

# print(not1\*2)

if not1<=100 and not1>90 :

print("Süper..")

print("Notun 5")

if not1 <= 90 and not1>70: print("Notun 4")

if not1<=70 and not1>49 : print("Geçtin")

else : print("Kaldın")

# print("Notun kaç:")

not1 = int(input("Notun kaç:"))

# print(not1\*2)

if not1<101 and not1>0:

if not1>90 :

print("Süper..")

print("Notun 5")

elif not1 > 70 : print("Notun 4")

elif not1>50 : print("Geçtin")

else : print("Kaldın")

else : print("Notu yanlış girdiniz.")

meyve1 = "Elma"

meyveler = ["Muz","Dut","Nar",meyve1]

# print(meyveler)

cevap = input("Sende hangi meyve var?")

if cevap in meyveler:

print("Aaa bu meyve bende de var.")

else: print("Bende bu meyve yok")

not1 = int(input("Notun kaç:"))

print("Geçtin" if not1>=50 else "Kaldın")

**# Matematiksel operatorler**

print(11%3) # kalanını döndürür

print(2\*\*3) # 2 üzeri 3 = 2x2x2

print(10/3) # Böl

print(10//3) # Böl, küsüratını yazma

print(9/5) # Böl

print(9//5) # Böl, küsüratını yazma

**# Mantıksal operatorler True/False döndürür**

print(3<7)

print(11<8)

print(11<8 and 3<7)

print(11<8 or 3<7)

**# işlem önceliği**

print(2 + 3 \* 4) # 14

print((2 + 3) \* 4) # 20

**# atama işlemi/operatörleri**

a = 5

print(a)

a += 2

print(a)

a \*\*= 2 # a = a\*\*2

print(a)

**# Matematiksel operatorler**

print(11%3) # kalanını döndürür

print(2\*\*3) # 2 üzeri 3 = 2x2x2

print(10/3) # Böl

print(10//3) # Böl, küsüratını yazma

print(9/5) # Böl

print(9//5) # Böl, küsüratını yazma

**# Mantıksal operatorler True/False döndürür**

print(3<7)

print(11<8)

print(11<8 and 3<7)

print(11<8 or 3<7)

**# işlem önceliği**

print(2 + 3 \* 4) # 14

print((2 + 3) \* 4) # 20

**# atama işlemi/operatörleri**

a = 5

print(a)

a += 2

print(a)

a \*\*= 2 # a = a\*\*2

print(a)

def selamla(): print("Merhaba"\*2)

import aa 🡪 bu dosyadaki kodun ciktilarini yazar

import bb

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.

def anamenu():

#print("╔"+"═"\*20+"╗")

print("╔═════════════════════╗")

print("║ VEKTOREL APP ║")

print("║ ║")

print("║ 1-Hesaplamalar ║")

print("║ 2-Oyunlar ║")

print("║ 3-Çizimler ║")

print("║ 10- ║")

print("║ ║")

print("║ Seçimiz nedir? ║")

print("╚═════════════════════╝")

secim = input() # input ile alınan bilgi string türündedir.

print("Seçiminiz:",secim,"idi.")

if secim<"1" or secim>"10":

print("secim 1 ile 10 arası olmalı")

anamenu()

if secim == "1":

print("Hesaplamaları seçtiniz.")

import proje\_oyunlar\_menusu

if secim == "2": print("Oyunlar seçtiniz.")

# selamla()

anamenu()

#print("╔"+"═"\*20+"╗")

print("╔═════════════════════╗")

print("║ OYUNLAR MENU ║")

print("║ ║")

print("║ 1-Tetris ║")

print("║ 2-Adam asmaca ║")

print("║ 3-Yılan ║")

print("║ 4- ║")

print("║ 5- ║")

print("║ 6-Ana menüye dön ║")

print("║ ║")

print("║ Seçimiz nedir? ║")

print("╚═════════════════════╝")

ss = input()

if ss == "1" : print("Tetris başlıyor..")

if ss == "6":

import proje\_ana\_ekrani

**H31**

def anamenu():

print("\033[1;32;40m")

#print("╔"+"═"\*20+"╗")

print("╔═════════════════════╗")

print("║\033[1;31;40m VEKTOREL APP \033[1;32;40m ║")

print("║ ║")

print("║ 1-Hesap makinesi ║")

print("║ 2-Çizimler ║")

print("║ 3-Oyunlar ║")

print("║ 4- ║")

print("║ ║")

print("║ Seçimiz nedir? ║")

print("╚═════════════════════╝")

s = int(input("Seçiniz nedir:"))

if s==1 :

import moduller.hesaplamalar

moduller.hesaplamalar.hesapmenu()

if s==2 :

import moduller.cizimler as ciz

ciz.cizimmenu()

anamenu()

# for aa in range(50,80,5): print(aa)

# for aa in range(80,50,-5): print(aa)

# for x in "Vektorel": print(x)

# meyveler = ["Dut","Muz","Nar"]

# for x in meyveler: print(x)

liste = [sayi for sayi in range(8,20,2)]

print(liste)

**import random**

random.random()

# print(random.random())

print(random.randint(10,100))

meyveler = ["Dut","Muz","Nar",33,44]

print(random.choice(meyveler))

**RASTGELE KARELER**

import random as r, turtle as t

# for a in range(4):

# t.forward(100)

# t.right(90)

t.speed(10)

for b in range(r.randint(5,20)):

renkler = ["red","green","blue"]

ku = r.randint(50,150)

x = r.randint(-300,300)

y = r.randint(-300,300)

t.color(r.choice(renkler))

t.up() **# t.penup()**

t.goto(x,y)

t.down() **# cizgi izi cikmasin diye**

for a in range(4):

t.forward(ku)

t.right(90)

input()

**RASTGELE SEKILLER**

import random as r, turtle as t

# for a in range(4):

# t.forward(100)

# t.right(90)

t.speed(10)

for b in range(r.randint(5,20)):

renkler = ["red","green","blue"]

ku = r.randint(50,150)

x = r.randint(-300,300)

y = r.randint(-300,300)

t.color(r.choice(renkler))

t.up() # t.penup()

t.goto(x,y)

t.down()

ks = r.randint(3,7)

for a in range(ks):

t.forward(ku)

t.right(360/ks)

input()

RANDOM SAYI (random.randint) – RANDOM ELEMAN (random.choice)

A computer screen shot of text

AI-generated content may be incorrect.

import random as r, turtle as t

def sekil\_ciz(ku,x,y,rn):

t.speed(10)

for b in range(r.randint(5,20)):

renkler = ["red","green","blue"]

ku = r.randint(50,150)

x = r.randint(-300,300)

y = r.randint(-300,300)

t.color(rn)

t.up() # t.penup()

t.goto(x,y)

t.down()

ks = r.randint(3,7)

for a in range(ks):

t.forward(ku)

t.right(360/ks)

for b in range(r.randint(5,20)):

renkler = ["red","green","blue"]

ku = r.randint(50,150)

x = r.randint(-300,300)

y = r.randint(-300,300)

rn = r.choice(renkler)

sekil\_ciz(ku,x,y,rn)

input()

**H3\_2**

**Matematik toplama alistirmalari**

def sayilar():

    import random

    print("\n\nMATEMATİK TOPLAMA ALIŞTIRMALARI\n")

    while True:

        seviye = int(input("Zorluk seviyesi ne olsun? (1/2/3): "))

        if 1 <= seviye <= 3:

            print(f"{seviye}. seviye seçtiniz...")

            break

        else:

            print("Lutfen gecerli bir sayi giriniz (1/2/3).")

    puan = 0

    olumlu = ["e", "evet", "evt", "olur", "devam", "ok", "x", " "]

    devam = "e" **# dongude baslangicta olumlu bir deger vermesi icin**

    while devam.lower() in olumlu:

        s1 = random.randint(1, 10 \*\* seviye)

        s2 = random.randint(1, 10 \*\* seviye)

        if int(input(f"{s1} + {s2} = ")) == s1 + s2:

            puan += 10

            print(f"Bildin, puanin {puan}.")

        else:

            puan -= 10

            print(f"Bilemedin, puanin {puan}.")

        devam = input("Devam etmek istiyor musun? ")

sayilar()

A screen shot of a computer code

AI-generated content may be incorrect.

**Seklin ici istedigimiz renkte dolu**

A screenshot of a computer code

AI-generated content may be incorrect.

**Ikiden fazla, tanimlanmamis sayiyi toplama**

A computer code with text

AI-generated content may be incorrect.

**PIP INSTALL**

**H41**

import time

# Bu kod, 10'dan 1'e kadar sayıları her saniye bir azaltarak ekrana yazdırır.

# time modülü, zamanla ilgili işlemler yapmamızı sağlar.

import datetime

for a in range(10,0,-1):

print(a, datetime.datetime.now()) # Şu anki tarihi ve saati de ekrana yazdırır.

time.sleep(1) # 1 saniye bekle

import datetime

zaman = datetime.datetime.now() # Şu anki tarihi ve saati alır

print("Zaman1 : ",zaman) # Şu anki tarihi ve saati de ekrana yazdırır.

print("Zaman2 : ",zaman.strftime("%d/%m/%Y %H:%M:%S"))

print("Zaman3 : ",zaman.strftime("%d-%m-%Y"))

print("Zaman3 : ",zaman.strftime("%d-%m-%y"))

print("Zaman3 : ",zaman.strftime("%y\_%m"))

print("Zaman3 : ",datetime.datetime.now().strftime("%y\_%m"))

import datetime

gunler=['Pazartesi', 'Salı', 'Çarşamba', 'Perşembe', 'Cuma', 'Cumartesi', 'Pazar']

gunlerk=['Pzt', 'Sa', 'Ça', 'Pe', 'Cu', 'Cts', 'Pz'] # Kısaltılmış gün isimleri

zaman = datetime.datetime.now() # Şu anki tarihi ve saati alır

# print(gunler[6])

# print(zaman.weekday()) # Haftanın gününü sayısal olarak verir (0=Monday, 6=Sunday)

# print("Gün : ",gunler[zaman.weekday()]) # Haftanın gününü string olarak verir

# print("Gün : ",gunlerk[zaman.weekday()]) # Haftanın gününü string olarak verir

print("Zaman2 : ",zaman.strftime("%d/%m/%Y %H:%M:%S"), zaman)

print("Zaman2 : ",zaman.strftime("%d/%m/%Y %H:%M:%S"), gunler[zaman.weekday()])

# Directive Description Example

# %a Weekday, short version Wed

# %A Weekday, full version Wednesday

# %w Weekday as a number 0-6, 0 is Sunday 3

# %d Day of month 01-31 31

# %b Month name, short version Dec

# %B Month name, full version December

# %m Month as a number 01-12 12

# %y Year, short version, without century 18

# %Y Year, full version 2018

# %H Hour 00-23 17

# %I Hour 00-12 5

# %p AM/PM PM

# %M Minute 00-59 41

# %S Second 00-59 8

# %f Microsecond 000000-999999 548513

# %z UTC offset 100

# %Z Timezone CST

# %j Day number of year 001-366 365

# %U Week number of year, Sunday as the first day of week, 00-53 52

# %W Week number of year, Monday as the first day of week, 00-53 52

# %c Local version of date and time Mon Dec 31 17:41:00 2018

# %x Local version of date 12/31/18

# %X Local version of time 17:41:00

# %% A % character %

print("YAŞ HESABI PROGRAMI")

import datetime

dogum\_yili = int(input("Doğum yılınız nedir: "))

simdiki\_yil = int(datetime.datetime.now().strftime('%Y')) # Şu anki tarihi alır

# print("Şu anki yıl: ", simdiki\_yil)

print(f"Demek {simdiki\_yil-dogum\_yili} yaşındasınız.")

from datetime import datetime

tarih\_str = "2025-07-12"

tarih\_dt = datetime.strptime(tarih\_str, "%Y-%m-%d")

print(tarih\_dt)

print(type(tarih\_dt))

import time, datetime, os

# Bu kod, 10'dan 1'e kadar sayıları her saniye bir azaltarak ekrana yazdırır.

# time modülü, zamanla ilgili işlemler yapmamızı sağlar.

for a in range(10,0,-1):

print(datetime.datetime.now().strftime('%H:%M:%S'), '=>', a) # Şu anki tarihi ve saati de ekrana yazdırır.

time.sleep(1) # 1 saniye bekle

os.system('cls') # Ekranı temizler

print("Süre doldu!")

meyveler = [] # Boş bir liste/dizi oluşturur

print(meyveler) # Boş listeyi ekrana yazdırır

meyveler.append("Elma") # Listeye "Elma" öğesini ekler

print(meyveler)

# meyveler.append("Nar",'Muz')

meyveler += ["Nar", "Muz"] # Listeye "Nar" ve "Muz" öğelerini ekler

print(meyveler)

# Örneği bitirenler;

# OBS studio yu indirip kurabil.

meyveler = [] # Boş bir liste/dizi oluşturur

print(meyveler) # Boş listeyi ekrana yazdırır

meyveler.append("Elma") # Listeye "Elma" öğesini ekler

print(meyveler)

# meyveler.append("Nar",'Muz')

meyveler += ["Nar", "Muz"] # Listeye "Nar" ve "Muz" öğelerini ekler

print(meyveler)

meyveler.pop()

print(meyveler)

meyveler = [] # Boş bir liste/dizi oluşturur

print(meyveler) # Boş listeyi ekrana yazdırır

meyveler.append("Elma") # Listeye "Elma" öğesini ekler

print(meyveler)

# meyveler.append("Nar",'Muz')

meyveler += ["Nar", "Muz"] # Listeye "Nar" ve "Muz" öğelerini ekler

print(meyveler)

meyveler.insert(2,"Portakal")

print(meyveler)

meyveler = ['Elma', 'Nar', 'Portakal', 'Muz',"Goz"]

print(meyveler) # Boş listeyi ekrana yazdırır

print("Dizi uzunluğu: ",len(meyveler))

print("1 indexli eleman:",meyveler[1]) # index değeri 1.

print("meyveler[1:]: ",meyveler[2:])

print("meyveler[:2]: ",meyveler[:2])

print("meyveler[:2]: ",meyveler[2:4])

print("meyveler[:2]: ",meyveler[:])

print("meyveler[:2]: ",meyveler[::2])

print("meyveler[:2]: ",meyveler[::-1])

print("Ankara[::-1] : ","Ankara"[::-1])

a=5 # Global variable = Genel/global değişken

print("Fonksiyon dışında a:", a)

def topla(x,y):

global a # Yerel olmayan (Global) a ile çalışacağım.

# print("Fonksiyon içinde1 a:", a)

a = 10 # Local variable = Yerel değişken

print("Fonksiyon içinde2 a:", a)

# print("Toplam:", x+y)

pass

topla(a,4)

print("Fonksiyon dışında a:", a)

import time

for a in range(10,0,-1):

print(a)

time.sleep(1) # 1 saniye bekle

print("İki sayıyı toplayan program")

# deneme1.py

sayi1 = int(input("Birinci sayıyı giriniz: "))

sayi2 = int(input("İkinci sayıyı giriniz: "))

toplam = sayi1 + sayi2

print("Toplam:", toplam)

print(type(sayi1))

print("sayi1\*2: ",sayi1\*2)

sayi1 = str(sayi1)

print(type(sayi1))

print("sayi1\*2: ",sayi1\*2)

**H42**

meyveler1 = ['Elma', 'Nar', 'Portakal', 'Muz',"Goz"] # list

meyveler2 = ('Nar', 'Portakal', 'Muz',"Goz",'Elma') # tuple

print(meyveler1) # Boş listeyi ekrana yazdırır

print(meyveler2) # Boş listeyi ekrana yazdırır

print("Dizi uzunluğu: ",len(meyveler1))

meyveler1.append("Dut")

# meyveler2.append("Dut") # tuple lara ekleme yapılamaz.

meyveler2 += ("Dut",) # += ile ekleme yapılabilir. Sondaki , önemli

print(meyveler1) # Boş listeyi ekrana yazdırır

print(meyveler2) # Boş listeyi ekrana yazdırır

print(type(('Elma', 'Nar', 'Portakal', 'Muz',"Goz")))

print(type('Elma'))

print(type(['Elma', 'Nar', 'Portakal', 'Muz',"Goz"]))

print(type(meyveler2))

aa = 44

abc = {"ad":"Ali","tel":"05076325874",44:"aaa",55:23,33:['5',8],aa:'55'} # key:value

print(abc)

print("1 indexli eleman:",meyveler1[1]) # index değeri 1.

print("1 indexli eleman:",meyveler2[1]) # index değeri 1.

# print("1 indexli eleman:",abc[1]) # index değeri 1. # Hata verir. dict de index olmaz.

print("ad keyli eleman:",abc["ad"]) # index değeri 1.

print("ad keyli eleman:",abc[44]) # index değeri 1.

print("ad keyli eleman:",abc[55]) # index değeri 1.

print("ad keyli eleman:",abc[33]) # index değeri 1.

print("ad keyli eleman:",abc[33][1]) # index değeri 1.

print("ad keyli eleman:",abc[aa]) # index değeri 1.

okul = ["Yenimahalle"]

cukurambar = ["11A","11B"]

mamak = ["10A","10B"]

print(okul)

okul += [cukurambar,mamak]

print(okul)

cukurambar[0] = ["Arkadaş","Uğur","Reyhan"]

cukurambar[1] = ["Betül","Beyza"]

print(okul)

# JSON

okul = {"Yenimahalle":{}}

cukurambar = {"cukurambarsb":{"11A":{},"11B":{}}}

mamak = {"mamaksb":{"10A":{},"10B":{}}}

print("Başlangıç okul:", okul)

okul.update(cukurambar)

okul.update(mamak)

print("cukurambar eklendi:", okul)

cukurambar["cukurambarsb"]["11A"] = ["Arkadaş","Uğur","Reyhan"]

cukurambar["cukurambarsb"]["11B"] = ["Betül","Beyza"]

print("Güncellenmiş okul:", okul)

kullanici\_tipi = {"idareci","ogrenci","veli","ogretmen"}

print(kullanici\_tipi)

# print(kullanici\_tipi[1]) # set te index veya key kavramı yok

kullanici\_tipi.update({'898'})

print(kullanici\_tipi)

# pip install opencv-python

import cv2 # opencv BGR

okunan = cv2.imread('renkler.png')

# print(okunan)

okunan[0][2]=[255,255,255]

print("\n\n",okunan)

cv2.imwrite('renkler.png',okunan)

import math

# print(dir(math))

math.pi

print(math.pi)

print(math.pi+1)

print(math.sin(180))

print(math.sqrt(16))

print(math.exp(5))

# import os # istenilen klasör yolu alma vs.

print(type(5))

a = 6

print(type(a))

a = 7.5

print(type(a))

class int1():

c = 9

b = 7

def ekle(xx):

print("ekle metodu çalıştı")

a = int1()

print(type(a))

print(a.c)

print(a.b)

# print(a.ekle())

class Ogrenci():

ad = "Tanımsız"

nu = 0

tc = "--"

dn = 0

def \_\_init\_\_(xx):

pass

def adekle(aa, adi):

aa.ad = adi

def cezaver(cc,miktar):

cc.dn += miktar

ogr1 = Ogrenci()

ogr2 = Ogrenci()

ogr3 = Ogrenci()

print(ogr1)

print(ogr1.ad)

ogr1.ad="Rüzgar"

ogr2.ad="Uğur"

print(ogr1.ad, ogr1.dn)

print(ogr2.ad, ogr2.dn)

print(ogr3.ad, ogr3.dn)

ogr2.cezaver(30)

ogr2.cezaver(20)

print(ogr2.ad, ogr2.dn)

# string işlemleri

xx = "Ali gel, okul açıldıA"

# xx = input("Bir cümle giriniz:")

print("Girdiğiniz : ",xx)

print("Büyük harf şekli : ",xx.upper())

print("İçindeki a nın yeri: ",xx.find("a"))

print("İçindeki A sayısı1 : ",xx.count("A"))

print("İçindeki A sayısı2 : ",xx.upper().count("A"))

print("a ları \_ yap : ",xx.replace("a","\_"))

print("A ları \_ yap : ",xx.upper().replace("A","\_"))

print(xx[5])

print(xx[5:])

print(xx[:5])

print(xx[5:15])

**H51**

open("deneme1.py","a")

open("deneme1.txt","a").write("Reyhan Kara, 05326589547")

open("deneme1.txt","a").write("Arkadaş Rüzgar, 05078547856")

open("deneme1.txt","a").write("Reyhan Kara, 05326589547\n")

open("deneme1.txt","a").write("Arkadaş Rüzgar, 05078547856\n")

dosya = open("deneme3.txt","w") # w modu ile dosya yoksa oluşur, varsa içine ekleme modunda açılır.

dosya.write("Betül MUTLU, 05089854785\n")

dosya.write("Hacı Ali, 05856325478\n")

dosya = open("deneme3.txt","w") # w modu ile dosya yoksa oluşur, varsa içine dekileri silerek veri ekleme modunda açar.

dosya.write("Betül MUTLU, 05089854785\n")

dosya.write("Beyza Ali, 05856325478\n")

dosya.close() # dosyalarını kaparak bir çok hatanın önlemini alın.

for a in range(10,20):

dosya = open(f"deneme{a}.txt","w") # w modu ile dosya yoksa oluşur, varsa içine dekileri silerek veri ekleme modunda açar.

dosya.write(f"içine yazılan: {a\*2}")

dosya.close() # dosyalarını kaparak bir çok hatanın önlemini alın.

# dosyaadi = "rehber.txt"

# def ekle(ad,tel):

# dosya = open(dosyaadi,"a")

# dosya.write(f"{ad},{tel}\n")

# print(" REHBER PROGRAMI ")

# a = input("Kaydedilecek ad :")

# b = input("Kaydedilecek tel:")

# ekle(a,b)

dosyaadi = "rehber.txt"

def ekle():

print(" REHBER PROGRAMI ")

a = input("Kaydedilecek ad :")

b = input("Kaydedilecek tel:")

dosya = open(dosyaadi,"a")

dosya.write(f"{a},{b}\n")

ekle()

import os

# try : os.mkdir("deneme6")

try : os.mkdir("deneme6/xyz")

except : print("Hata oluştu.")

# d = open("deneme77.txt","a+")

d = open("deneme6/abc/deneme77.txt","a+")

# "deneme6/abc/deneme77.txt" için

# dosya yolu: deneme6/abc/

# dosya adı : deneme77.txt

import os

try : os.mkdir("D:/erdincdonmez/88")

except : print("Hata oluştu.")

d = open("D:/erdincdonmez/deneme77.txt","a+")

import os, time

# try : os.mkdir("D:/erdincdonmez/88")

# except : print("Hata oluştu.")

# d = open("D:/erdincdonmez/deneme77.txt","a+")

for a in range(5):

time.sleep(1)

os.mkdir("D:/erdincdonmez/88")

time.sleep(1)

os.rmdir("D:/erdincdonmez/88")

import os

dizin = os.getcwd()

print("Aktif dizin: ",dizin)

os.chdir("D:/erdincdonmez/dersuygulamalari99/deneme6") # ""

dizin = os.getcwd()

open("sil.dat","+a")

print("Aktif dizin: ",dizin)

liste = os.listdir()

print(\*liste, sep="\n")

import os

dizin = os.getcwd()

print("Aktif dizin: ",dizin)

dosya\_sayisi = 0

dizin\_sayisi = 0

liste = os.listdir()

print()

for a in liste:

if os.path.isfile(a) :

print("Dosya\t\t", end=" ")

dosya\_sayisi += 1

if os.path.isdir(a) :

dizin\_sayisi +=1

print("Klasör\t\t",end=" ")

print(a)

print(dosya\_sayisi, "adet dosya var.")

print(dizin\_sayisi, "adet dizin var.")

import os

liste = os.listdir()

for a in liste:

ozellik = os.stat(a)

print(ozellik.st\_size,"byte\t\t", end="")

print(ozellik.st\_ctime,"\t\t", end="")

print(a)

import os

os.rename("deneme6-","deneme6-------")

# walk() # Dizin ve dosyalar arasında gezinebilirsiniz.

# https://www.geeksforgeeks.org/os-walk-python/

import os

from os.path import join, getsize

for root, dirs, files in os.walk('D:\\'):

# print(root, "consumes", end="")

# print(sum(getsize(join(root, name))

# for name in files), end="")

# print("bytes in", len(files), "non-directory files")

if 'REHBER' in dirs:

dirs.remove('REHBER') # don't visit CVS directories

file = open("deneme2.txt","r")

okunan = file.read()

print(type(okunan))

print(okunan)

f = open("deneme2.txt","r")

okunan = f.readlines()

print(type(okunan))

print(\*okunan,sep="\n")

**H52**

Şifre:170 8585

85,

90,

95,

for x in range(10,100,5):

with open(f"dosya-{x}.txt","w") as d:

d.write(f"Şifre:{x\*2} {str(x)\*2}\n\n")

for b in range(x,100,5):

# d.write(x,"\n")

d.write(f"{str(b)},\n")

d = open(f"dosya-{x}.txt","w")

d.write(f"Şifre:{x\*2} {str(x)\*2}\n\n")

d.close()

# dsy = open("fikra.txt","r",encoding="utf8") # dosyayı bulamaz, hata verir.

dsy = open("h5\_2\_dosya\_islemleri/fikra.txt","r",encoding="utf8")

print(dsy.read())

# dsy = open("fikra.txt","r",encoding="utf8") # dosyayı bulamaz, hata verir.

dsy = open("h5\_2\_dosya\_islemleri/fikra.txt","r",encoding="utf8")

# print(dsy.read(10))

# print(dsy.read(15))

# print(dsy.read(10))

for a in range(10):

print(dsy.read(10),"--",end="")

# dsy = open("fikra.txt","r",encoding="utf8") # dosyayı bulamaz, hata verir.

dsy = open("h5\_2\_dosya\_islemleri/fikra.txt","r",encoding="utf8")

print(dsy.readline())

print(dsy.readline())

print(dsy.readline())

# dsy = open("fikra.txt","r",encoding="utf8") # dosyayı bulamaz, hata verir.

dsy = open("h5\_2\_dosya\_islemleri/fikra.txt","r",encoding="utf8")

for a in range(10):

print(dsy.read(10),"--",dsy.tell(),end="")

# dsy = open("fikra.txt","r",encoding="utf8") # dosyayı bulamaz, hata verir.

dsy = open("h5\_2\_dosya\_islemleri/fikra.txt","r",encoding="utf8")

for a in range(10):

print(dsy.read(10),end="")

dsy.seek(dsy.tell()-3)

import os

# os.remove("h5\_2\_dosya\_islemleri/dosya-90.txt")

for a in range(10,85,5):

os.remove(f"h5\_2\_dosya\_islemleri/dosya-{a}.txt")

# readlines

# dsy = open("fikra.txt","r",encoding="utf8") # dosyayı bulamaz, hata verir.

dsy = open("h5\_2\_dosya\_islemleri/fikra.txt","r",encoding="utf8")

okunanlar = dsy.readlines()

print(okunanlar,"\n"\*5)

# for a in okunanlar:

# print(a)

for a in range(len(okunanlar)-1,-1,-1):

print(okunanlar[a])

# d = open("denemex.txt","x")

# d.write("yazılanbilgi1")

# d.close()

d = open("denemex.txt","r")

o = d.read()

# d.write("yazılanbilgi2")

print(o)

d.close()

d = open("denemex.txt","a")

d.write("yazılanbilgi2\n")

d = open("denemex.txt","r+")

o = d.read()

d.write("yazılanbilgi2")

print(o)

d.write("yazılanbilgi2\n")

dosya = open("rehber.txt","r",encoding="utf8")

okunan = dosya.readlines()

print(okunan)

for a in okunan:

parcali = a.split(",")

print(parcali[0],"\t\t",parcali[1])

duzelecek = input("Düzelecek ne:")

yeniveri = input("Yeni veri ne:")

dosya = open("rehber.txt","w",encoding="utf8")

yenisekli = []

for a in okunan:

parcali = a.split(",")

if parcali[0] == duzelecek: parcali[0] = yeniveri

if parcali[1] == duzelecek: parcali[1] = yeniveri

# yenisekli.append(parcali)

yazilacak = f"{parcali[0]},{parcali[1]}"

dosya.write(yazilacak)

Parayı Veren Düdüğü Çalar

Çocuklar, pazara gelen Nasreddin Hoca'nın etrafını sarmış. “Hoca, bana düdük al!” demiş biri. “Bana da, bana da!” demiş bir diğeri.

Diğerleri de sırayla:

– Ben de düdük isterim!

– Bir tane de bana!, demişler.

İçlerinden sadece biri Nasreddin Hoca’ya düdük parası vermiş. Hoca, parayı alıp pazara gitmiş.

Hoca, akşam pazardan dönünce çocuklar etrafını sarmış. Her biri düdüğünü istemiş. Cebinden bir düdük çıkaran hoca, parayı veren çocuğa vermiş.

Diğer çocuklar hep bir ağızdan bağırmış:

– Hani bizim düdüğümüz?

Nasrettin Hoca gülerek,

– Parayı veren düdüğü çalar, demiş.

**Git ve Github Farki**

Git – bilgisayardaki kod depolarini yonetmek icin kullandigimiz yapilar

Github –

Islem dogru olmadigi icin yazmaz

if 13<4:

    print("Calisti.")

**Cikti 🡪** bos

**09.08.2025**

pip.install pyinstaller 🡪 kurduktan sonra

cd

temel konular

exe dosyasina cevirme

Python Malzeme – kod paylasimi 99 bak

Roadmap

Print (2<9) 🡪 True yazar

Tek tirnak 🡪

Cift tirnak 🡪

Uc tirnak 🡪

Pypi.com 🡪 hazir paket projeler

Path tanimi nasil yapilir??

Alt shift asagi 🡪 cogaltmak icin

Haftaya kadar sinavi bitir

**HAFTA 7**

**Yerel / Global Degerler**

|  |  |
| --- | --- |
| xx = 11  def fonksiyon1():      xx = 22      print("Yereldeki:",xx)  fonksiyon1()  print("Globaldeki:",xx)  **Cikti 🡪**  Yereldeki: 22  Globaldeki: 11 | xx = 11  def fonksiyon1():      global xx      xx = 22      print("Yereldeki2:",xx)  fonksiyon1()  print("Globaldeki:",xx)  **Cikti 🡪**  Yereldeki2: 22  Globaldeki: 22 |

def rapor\_olustur(\*\*gelenveri):

    # print("gelenveri:\n",gelenveri)

    print("📋 Rapor Özeti:")

    # for key, value in gelenveri.items():

    for a, b in gelenveri.items():

        print(f"- {a}: {b}")

# Kullanım

rapor\_olustur(

    Ad="Erdinç Dönmez",

    Yaş=27,

    Şehir="Ankara",

    Meslek="Yazılım Geliştirici",

    Hastaslık="Kemikte çatlak tespiti",

    İstirahat="5 gün istirahati uygundur")

**Cikti 🡪**

📋 Rapor Özeti:

- Ad: Erdinç Dönmez

- Yaş: 27

- Şehir: Ankara

- Meslek: Yazılım Geliştirici

- Hastaslık: Kemikte çatlak tespiti

- İstirahat: 5 gün istirahati uygundur

**LAMBDA**

Lambda satır içi basit fonksiyon demek

yerine lambda fonksiyonu sayesinde kullanımı kolaylaştırıyoruz.

lambda fonksiyonunun adı olmadığından buna anonim fonksiyon da denir.

|  |  |
| --- | --- |
| def aa(x):      return x\*2  print(aa("Merhaba"))  sayilar = [11,22,33]  def ikikati(x):      return x\*2  yeniliste = []  for a in sayilar:      yeniliste.append(ikikati(a))  print(yeniliste)  **Cikti** 🡪 [22, 44, 66] | print((lambda x:x\*2)("Merhaba"))  **Map(): Bir listedeki elemana fonksiyonu uygular**  sayilar = [11,22,33]  print(list(map(lambda a:a\*2,sayilar)))  **Cikti** 🡪 [22, 44, 66]  print(\*list(map(lambda a:a\*2,sayilar)))  **Cikti** 🡪 22 44 66  print(\*list(map(lambda a:a\*2, [11,22,33])))  **Cikti** 🡪 22 44 66  kare = lambda x: x \*\* 2  print(kare(5))  **Çıktı** → 25 |

**KWARGS**

def fonksiyon(\*\*gelenler):

  print("Kullanıcı tipi " + gelenler["sad"])

fonksiyon(ad = "Ahmet", sad = "Öğretmen", brans ="mat",sicilno=22)

**Cikti** 🡪 Kullanıcı tipi Öğretmen

**MAP**

A screenshot of a computer

AI-generated content may be incorrect.

**FILTER**

sayilar = [1, 2, 3, 4]

cift = list(map(lambda x: x % 2 == 0, sayilar))

print(cift)

**Cikti → [False, True, False, True]**

sayilar = [11,22,3,6,8]

def tekMi(xx):

    if xx%2!=0: return True

yeniDizi=list(filter(tekMi,sayilar))

print("sayilar: ",sayilar)

print("yeniDizi: ",yeniDizi)

**Cikti 🡪**

sayilar: [11, 22, 3, 6, 8]

yeniDizi: [11, 3]

**YIELD**

A screen shot of a computer code

AI-generated content may be incorrect. **Cikti 🡪** 

A screenshot of a computer

AI-generated content may be incorrect. **Cikti 🡪** A red line with black text

AI-generated content may be incorrect.

**A screenshot of a computer

AI-generated content may be incorrect. Cikti 🡪**

A screenshot of a computer

AI-generated content may be incorrect. **Cikti 🡪A screenshot of a computer

AI-generated content may be incorrect.**

**HATA MESAJI**

A screenshot of a computer

AI-generated content may be incorrect. Cikti 🡪 A close up of a text

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

**Cikti 🡪**

A screenshot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect. **Cikti** 🡪 Deger yanlis girildi

A screen shot of a computer code

AI-generated content may be incorrect. **Cikti 🡪**  A close up of a screen

AI-generated content may be incorrect.

A screen shot of a computer code

AI-generated content may be incorrect. **Cikti 🡪** A close up of a text

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect. **Cikti 🡪** A screenshot of a computer screen

AI-generated content may be incorrect.

A computer code with many colorful text

AI-generated content may be incorrect. **Cikti 🡪** 

A computer code with text

AI-generated content may be incorrect.

Normalde value error yok, biz urettik

**Cikti 🡪**

A computer screen shot of a computer code

AI-generated content may be incorrect.

class abcHatasi(Exception):       # Özel bir hata sınıfı tanımlıyoruz, Exception sınıfından alıyor

    """Kendi hata mesajınızı oluşturun"""

    def \_\_init\_\_(self, mesaj="Hata oluştuuuuuu!"):   # Sınıfın yapıcı (constructor) metodu

        self.mesaj = mesaj            # Hata mesajını sınıfın bir özelliği olarak saklar

        super().\_\_init\_\_(f"Dikkat! {mesaj}")   # Üst sınıf (Exception) yapıcısına mesaj gönderilir

try:                                  # Hata yakalama bloğunu başlatıyoruz

    sayilar=[2,30,5]                  # Üç elemanlı bir liste tanımlıyoruz

    yazilacak = int(input("Kaçıncı sayı:"))   # Kullanıcıdan bir index al (sayıya çeviriyoruz)

    if yazilacak>len(sayilar)-1:       # girilen index listedeki en büyük index'ten büyükse

        raise abcHatasi("Geçersiz değer!")   # Özel oluşturduğumuz hatayı fırlatıyoruz

    else:                             # Eğer index geçerliyse

        print(sayilar[yazilacak])     # Listedeki ilgili elemanı ekrana yazdırıyoruz

except IndexError:                     # Eğer Python kendi IndexError hatasını fırlatırsa

    print("Bu kadar sayı yok....")     # Bu mesajı gösteriyoruz

# except ValueError:

except ValueError as aa:              # kullanıcı sayı yerine geçersiz veri girerse (örn. harf)

    print("Bu kadar sayı yok.",aa)     # Hata mesajı ile birlikte yazdırıyoruz

except abcHatasi as aa:               # Eğer bizim özel hatamız tetiklenirse

    print("Bu kadar sayı yokkk.",aa)  # Özel hata mesajını yazdırıyoruz

**Cikti 🡪**

Kaçıncı sayı:7

Bu kadar sayı yokkk. Dikkat! Geçersiz değer!

Kaçıncı sayı:e

Bu kadar sayı yok. invalid literal for int() with base 10: 'e'

Kaçıncı sayı:!

Bu kadar sayı yok. invalid literal for int() with base 10: '!'

**RECURSION**

A computer code with many colored text

AI-generated content may be incorrect. Cikti 🡪 A close up of a white background

AI-generated content may be incorrect.

A computer screen shot of a computer code

AI-generated content may be incorrect. Cikti 🡪