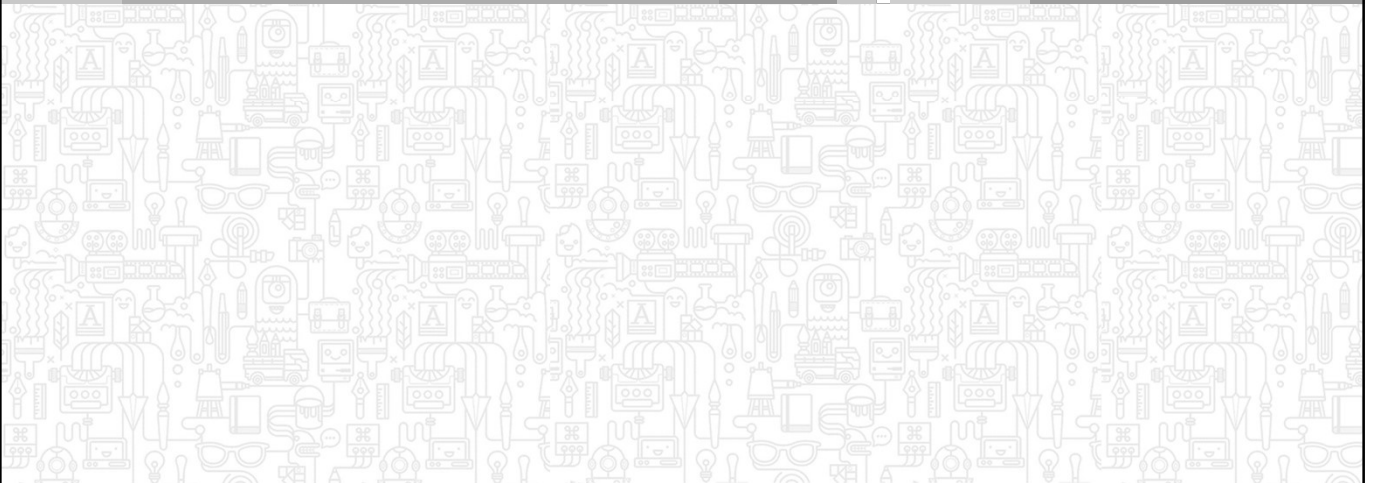


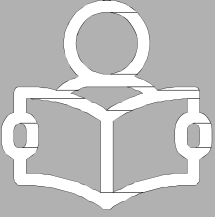
PYTHON



**Modüler programlama**

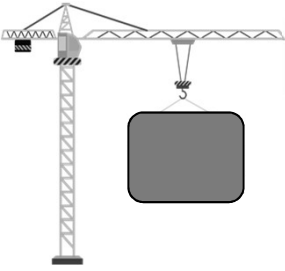


## Kapsam



- ✓ Modül Yazma ve Çağırma
- Hazır Modüllerin Kullanımı
- Random ve Math Kütüphaneleri
- Pip Paket Yükleyici Kullanımı
- Time Modülü

## Modüler Programlama



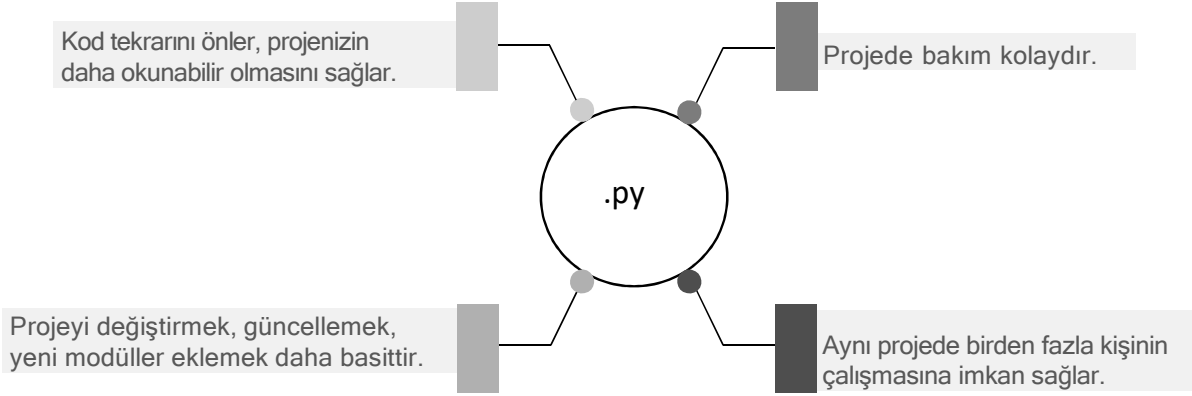
Python'da her bir dosya modüldür. Modüller içinde fonksiyonları, sınıfları ve objeleri bulundurur.

Python birçok modül içerdiği gibi, kullanıcı kendisi de modül yazıp kullanabilir.

Programın önceden yazılmış parçaları, bütüne istenildiği zaman dahil edilerek çalıştırabilir.

Python modülleri kütüphane olarak da adlandırılır.

## Modül Kullanmanın Avantajları



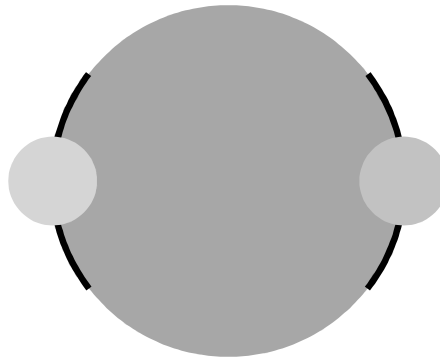
## Modül Yazma ve Çağırma

It matches people with similar interests as a basis for recommendation

Spyder ile yazdığımız .py uzantılı her dosya modüldür.

- 1- modülAdı.py uzantılı bir modül yazılır ve kaydedilir.
- 2- Çalışma dizini bu modülün olduğu dizin olarak seçilir.

```
# ilkModül.py
def cagir():
    print("Merhaba Arkadaşım")
```



3- Aynı klasörde olmak zorunda olmayan hali hazırda çalışılan modül içerisine aşağıdaki satırlardan birisi yazılır;

```
import modülAdı
from modülAdı import *
```

4- Python bu dosyaya `_pycache__` klasörünü otomatik atar. Sonraki çağırımlarda bu klasöre gitmek böylece gerekmez.

5-Modül çalışmaya hazırdır, çalıştırmak istenilen fonksiyon aşağıdaki satırdaki gibi çağırılır; modülAdı.fonksiyonAdı()

C:\Users\Bilgisayar\Desktop

```
#deneme.py
import ilkModül #modülü çağırдық
ilkModul.cagir() #modül içindeki fonksiyonu çalıştırdık
```

## dir()

Bir modülün içerisinde neler olduğunu görmek istersek;

Kod

```
import ilkModül
print(dir(ilkModül))
['__builtins__', '__cached__', '__doc__', '__file__', '__loader__', '__name__', '__package__', '__spec__', 'cagir']
```

## from...import

Program içinde sadece nesneleri çağırmak istenebilir. Bundan dolayı modülde sadece o nesneyi çağırmak gerekebilir.

from <Modül adı> import <İçericek Nesne>

Bu durumda fonksiyonu çağırırken artık modül adı kullanılmaz.

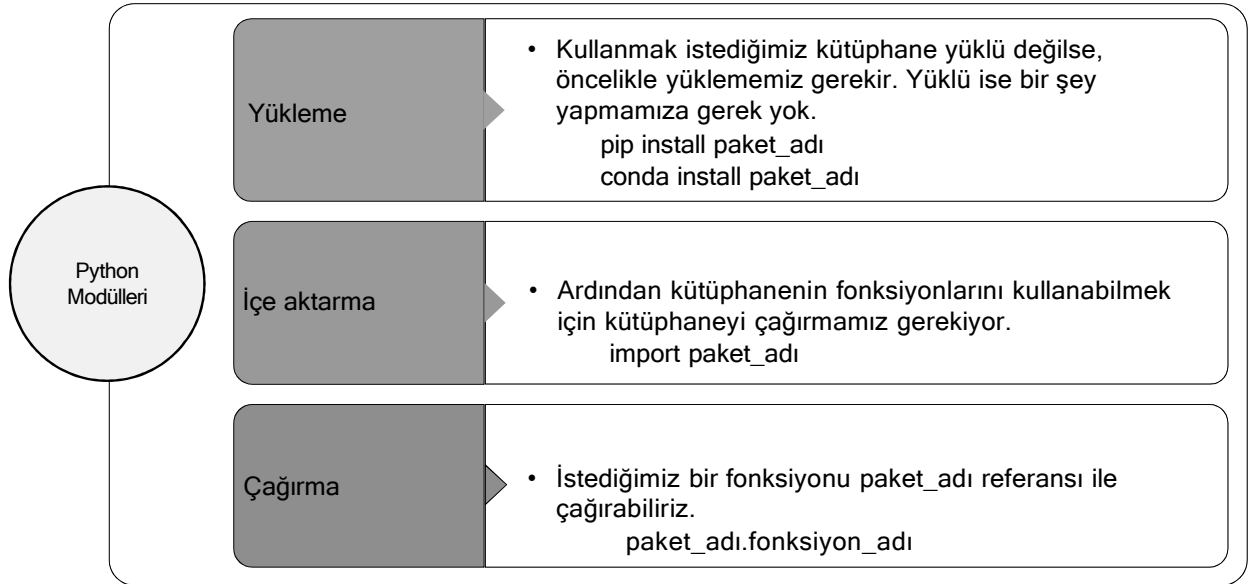
Kod

```
from ilkModül import cagir #modülün içerisindeki cagir fonksiyonunu çağırdık
cagir()#fonksiyonu çalıştırdık
```

Tüm nesneler kullanılmak istendiğinde ise; from <Modül adı> import \* şeklinde tanımlama gerekmektedir.

Yine bu şekilde modül yüklenirse, fonksiyon çağırırken modül adına gerek kalmaz

## Hazır Modüllerin Kullanımı



## Math Kütüphanesi

`math` modülü matematiksel işlemler yapmanızı kolaylaştırmak için yazılmış bir modüldür. Örneğin, `math.pi`, bize  $\pi$  sayısını ve `math.pow` ise sayının üssünü alarak değerini vermektedir.

Kod

```
import math
yaricap=4
alan=math.pi*(math.pow(yaricap,2))
print(alan)

fak=math.factorial(4)
cosinus=math.cos(120)
karekok=math.sqrt(81)
```

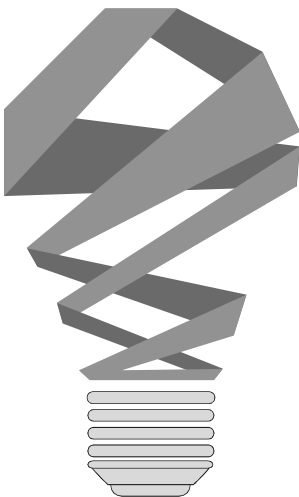
## Random Kütüphanesi

Python'da rastgele sayı üretilmesini sağlamaktadır.

Kod

```
import random
a=random.random() #[0, 1) aralığında rastgele sayı üretir
b=random.randint(1, 10) #1 ile 10 (dahil) arasında rastgele tamsayı üretir
cicekler = ['gül', 'karanfil', 'papatya', 'begonya', 'menekşe']
random.choice(cicekler) #listeden rastgele bir öge seçilir
```

## Pip Paket Yükleyici Kullanımı



01

Pip bir paket yöneticisidir (Aynı zamanda Anaconda'nın conda adında bir paket yükleyicisi de vardır). Python kütüphanesi ve modüllerini kurmanızı sağlayan yardımcı bir programdır. Python'da standart paketlerin dışındaki modüllere <https://pypi.org/> adresinden ulaşabilirsiniz. Şu an 326750 kütüphane bulunmaktadır.

02

Pip ile paket kurmak için cmd açılmalı ve cmd ekranına aşağıdaki kod yazılmalıdır: (Anaconda Prompt üzerinden de yapılabilir, ayrıca buradan conda kullanılır)

```
pip install paket_adı # paketi kurmak için
pip uninstall paket_adı # kurulu paketi kaldırmak için
pip install paket_adı==1.0.4 # istenilen versiyonu kurar
pip install paket_adı>=1.0.4 # alt limit ile verilen versiyonu kurar
```

03

install : Yeni bir paket yükler. #pip install django  
Uninstall : Varolan bir paketi siler. #pip uninstall django  
Freeze : Yüklü tüm paketleri requirements formatında listesini çıktıya verir. # pip freeze  
List : Yüklü tüm paketleri normal listesini çıktıya verir. #pip list  
Show : Yüklü paketler hakkında bilgi verir. #pip show django  
Upgrade : Kurulu bir paketi güncellemek için kullanılır. #pip install django -- upgrade

## Time Modülü



Time modülü zaman değerlerini düzenlemekle ilgili birçok görevi yerine getirebilmektedir. Zamanı göstermek için iki standart bulunmaktadır. Birincisi zamanı Epoch'tan itibaren saniye olarak vermektir.

Epoch Unix zaman başlangıcı olarak alınır ve takvime göre 1 Ocak 1970'e denk gelmektedir.

Eğer Epoch türünden şimdiki zaman saniye biçiminde alınmak istenirse şu yöntemi kullanabilirsiniz.

```
import time
print (time.time())
```



Ekran çıktısı bize 1 Ocak 1970'ten itibaren kaç saniye geçtiğini vermektedir.

Elde edilen değeri `gmtime()` fonksiyonu kullanarak okunabilir tarih formatına çevrilebilir.

```
import time
print (time.gmtime(time.time()))
```

## Time Modülü

Kod

```
import time
print(time.localtime()) #zaman bilgilerini sıralı bir tüp şeklinde vermektedir
print(time.ctime()) #içinde bulunulan zaman bilgilerini vermektedir
print (time.strftime("%d/%m/%Y")) #kendinize ait zaman cümlesi oluşturabilirsiniz
time.sleep(10) #programın belirlenen süre (sn) boyunca durdurur
```

#örnek

```
print ("Başlangıç : %s" % time.ctime()) #bugünün tarihinde saniyede var
time.sleep(5)
print ("Bitiş : %s" % time.ctime())
```



1

**0-20'ye kadar sayıları döngü yapısı ile ekrana yazdırırken her bir sayı arasına yarım saniye duraklamalar koyarak programı yapınız.**





1

**0-20'ye kadar sayıları döngü yapısı ile ekrana yazdırırken her bir sayı arasına yarım saniye duraklamalar koyarak programı yapınız.**

```
import time
for i in range(1,21):
    time.sleep(0.5)
    print(i)
```



Doğru Cevap

---

2

**Os modülünde name fonksiyonu işletim sistemi hakkında bilgi vermektedir. İşletim sisteminiz hakkında bilgi almak için gerekli olan kodları yazınız.**



2

**Os modülünde name fonksiyonu işletim sistemi hakkında bilgi vermektedir. İşletim sisteminiz hakkında bilgi almak için gerekli olan kodları yazınız.**

```
import os  
print (os.name)
```



Doğru Cevap

3

**İki sayının toplamını yaptıran programı fonksiyon ve modül yazma kullanarak yapınız.**



3

İki sayının toplamını yaptıran programı fonksiyon ve modül yazma kullanarak yapınız.

```
#modul.py dosyası;  
def toplama(sayi1,sayi2):  
    toplam=int(sayi1)+int(sayi2)  
    return toplam
```

```
#deneme.py dosyası;  
import modul  
a = input("birinci sayıyı giriniz:")  
b = input("ikinci sayıyı giriniz:")  
sonuc = int(modul.toplama(a, b))  
print (sonuc)
```

Doğru Cevap



# Teşekkürler