Python da oluşturduğunuz her bir dosya aslında bir modüldür.

Python da iki çeşit modül var;

A)Hazır modüller

B)Kendi oluşturduğumuz modüller.

Neden Modüller?

Büyük ve üzerinde çok fazla geliştirici çalışan projeler genel olarak modüllere Türkçe kaşılığı ile parçalara ayrılır. Bu sayede parçalara ayrılan projede bize bu tür avantajları sağlar.

Modüller kod tekrarını önler.

Modüller projemizi daha okunabilir olmasını sağlıyor.

Modüller yapıda projeyi değiştirmek, güncellemek, yeni modüller eklemek basittir.

Modüller yapıda projede bakım kolaydır.

Modüller aynı projede birden fazla kişinin çalışmasına imkan sağlıyor.

Bunun için kodların yazılı olduğu bir dosya oluşturup uzantısını **py** olarak kaydedin. Örneğin aşağıdaki kodu **modulum.py** isimli bir dosyaya kaydedin:

 PYTHON Kodu

 PYTHON Kodu

**def** **karsila**(isim):

print("Merhaba, " + isim)

Modülü Kullanmak

Yukarıda oluşturduğumuz modülü kullanalım. Bunun için kodlarımızın başına **import <dosya adı>** yazıyoruz.

 PYTHON Kodu

**import** modulum

modulum.karsila("Murat")

Böylelikle başka bir modüldeki (dosyadaki) fonksiyonu kendi uygulamamızda kullandık.

Başka bir öenek

Modül kullanmanın bir avantajı da, programların tümü kolayca tasarlanabilmesidir, çünkü tüm ekip tüm kodun yalnızca bir parçası veya modülü üzerinde çalışır.

Bunu bir örnekle anlamaya çalışalım:

Bir hesap makinesi için bir program yapmak istediğinizi varsayalım. Toplama, çıkarma, çarpma, bölme vb. İşlemler yapılacaktır.

Kodu ayrı parçalara ayıracağız, tüm bu işlemler için bir modül veya her bir işlem için ayrı modüller oluşturabiliriz. Sonra bu modülleri ana program mantığımızda çağırabiliriz.

Fikir kodu en aza indirmektir. Modüller oluşturursak, bu kodu sadece bir program için kullanabileceğimiz anlamına gelmez, bu modülleri diğer programlar için de çağırabiliriz.

Şimdi modül kavramını anladığımıza göre, Python'da nasıl modül oluşturabileceğimizi anlamaya çalışalım.

Nasıl Python Modülü Oluşturabilirim?

Python'da bir modül oluşturmak, .py uzantısını kullanarak basit bir python kodu yazmaya benzer. Yukarıdaki örnekte, çeşitli işlemler için bir modül yapmaya çalışalım.

def topla(x,y):

return x + y

def cikar(x, y):

return x - y

def carp(x, y):

return x \* y

def bole(x, y):

return x / y

Yukarıdaki kodu hesapmakina.py ismi ile bir dosyaya kaydedin. Python'da bu şekilde bir modül yaratıyoruz. Bu modülde farklı fonksiyonlar yarattık

Diyelim ki main.py adlı bir python dosyamız var.

import hesapmakinesi as hesap

a = 10

b = 20

toplam = hesap.topla(a,b)

print(toplam)

Yukarıdaki kodda, as anahtar sözcüğünü kullanarak bir takma ad oluşturduk. Yukarıdaki kodun çıktısı, hesapmakinesi.py modülündeki topla fonksiyonunda belirtilen mantığı kullanılarak a ve b sayılarının toplanma işlemini gerçekleştirecektir

Modüldeki Değişkenleri Kullanmak

Modüller sadece fonksiyon içermez, ayrıca değişkenler de içerir. Bunları çağırmak için modül adı ve değişken adı yazmak yeterlidir.

 PYTHON Kodu

#modulum.py

kisi = {

"isim": "Murat",

"yas": 36,

"ulke": "Türkiye"

}

#merhaba.py

**import** modulum

a = modulum.kisi["yas"]

print(a)

Modülleri Adlandırmak

Modülün adı, dosyanın adıdır. Yukarıdaki örnekte dosya "modulum.py" olarak kaydedildiği için modül adı "modulum" olmuştur.  
  
Modül adları için kısaltma kullanabiliriz. Bunun için **as <kısaltma>** şeklinde modül çağırılabilir.

 PYTHON Kodu

**import** modulum **as** m

a = m.kisi["yas"]

print(a)

Python İç Modüller

Python'da hazır modüller bulunmaktadır. Bunlar import edilerek kullanılabilir. Örneğin aşağıda **platform** adındaki modülden faydalanıldı.

 PYTHON Kodu

**import** platform

x = platform.system()

print(x)

dir() Fonksiyonunu Kullanmak

Python'daki bir modülde bulunan tüm fonksiyonları ve değişkenleri görüntülemek için **dir(<modül adı>)** kullanılabilir.

**PYTHON Kodu**

**import** platform

x = dir(platform)

print(x)

Modülün Sadece Belli Bir Kısmını Kullanmak

Modüllerin tamamı yerine sadece belli bir fonksiyonu ya da değişkenini kullanmak isteyebiliriz. Bu durumda **from <modül adı> import <modüldeki fonksiyon ya da değişken>** yapısı kullanılabilir:

 PYTHON Kodu

**from** modulum **import** kisi

**print** (kisi["yas"])

**UYARI:** Bu tarz kullanımlarda fonksiyonu ya da değeri kullanırken modül adı yazılmaz. Yani **modulum.kisi["yas"]** şeklinde bir kullanım olamaz.

[Anasayfaya dön](iibf-excel/4mayısisletme1ogr%20-%20Kopya.xlsx)