

Modules(Modullar)

May 23, 2022

1 Modules (Modullar)

Modullar Python proqramlaşdırma dilində sinifləri,funksiyaları və verilənləri ayrı-ayrı fayllarda yerləşdirməyə imkan verir.Biz,python interaktiv moddan çıxdıqda,yazdığımız funksiyalar-ifadələr öz funksionallığını itirir.Modullardan istifadə etməklə iri həcmli proqramları hissələrə bölməklə istifadə etmək mümkündür.

Modulları hazırlamaq üçün python faylları hazırlamağımız şərtdir.Modullar təkrar kod yazmaq,kodları kopalamaq və s kimi zəhmətlərdən bizim yardımçımız olur.Asta-asta giriş etdikcə modulların necə faydalı olduğunun şahidi olacaqsınız.Və siz modullara dəfələrlə rast gələcəksiniz.Məsələn,siz triqonometriya,loqarifma diskriminant kimi məsələləri öyrənmək üçün riyaziyyat kitabına ehtiyacınız olacaq.Burda kitab bir modul daxilindəki ayrı ayrı dərslər isə funksiya və sinifləridir.

İstifadəçi tərəfindən hazırlanan modullardan savayı Python dilinin standart kitabxanasına aid olan modullar da vardır.os,sys,math və s.(trafı növbəti dərslərdə hər birini öyrənəcəyik)

1.1 *Modullar python faylı olub,sonluğu py və ya pyw ilə bitir. (modul.py)

Yazdığımız modul faylın adını qeyd edərkən mütləq diqqətli olun,standart kitabxanada olan modul adlarından istifadə etməyin

1.1.1 *Modulu çağırmaq üçün import ifadəsindən istifadə edirik

import modul_adı

modul_adı.atribut Modul_adı çağırılan python faylı və ya modul adı,atribut isə modul daxilində yazılmış funksiya,siniflər və ya digər komponentlərdir

İlk öncə Python dili standart kitabxanalarına aid olan modullara nəzər yetirək

```
[1]: import sys #sys modulunu proqrama daxil edirik(import edirik)
```

```
[2]: sys.path
```

```
[2]: ['',  
      'C:\\ProgramData\\Anaconda3\\python36.zip',  
      'C:\\ProgramData\\Anaconda3\\DLLs',  
      'C:\\ProgramData\\Anaconda3\\lib',  
      'C:\\ProgramData\\Anaconda3',
```

```
'C:\\ProgramData\\Anaconda3\\lib\\site-packages',
'C:\\ProgramData\\Anaconda3\\lib\\site-packages\\win32',
'C:\\ProgramData\\Anaconda3\\lib\\site-packages\\win32\\lib',
'C:\\ProgramData\\Anaconda3\\lib\\site-packages\\Pythonwin',
'C:\\ProgramData\\Anaconda3\\lib\\site-packages\\IPython\\extensions',
'C:\\Users\\garay\\.ipython']
```

1.2 Proqrama modullar daxil olunanda,interpretator onlari aŝağıdaki ardıcıl- lıqla axtarır.

1.2.1 1.Cari kataloq

1.2.2 2.PYTHONPATH mühitində olan kataloqlar

1.2.3 3.Platformadan asılı təyin edilən kataloqlar

Kataloqların yeri əməliyyat sistemindən asılı olaraq fərqli ola bilər.Mən hal-hazırda Ubuntu 19.10 Linux distribyutoru istifadə etdiyimdən yuxarıdakı qaydada,pythonpath kataloqu sıralandı

Modulda tərtib olunan atributları əldə etmək üçün isə dir() - ifadəsindən istifadə olunur.

```
[3]: import sys
# dir(sys) sətir boyu məlumatların kifayət qədər çox olduğundan kodu
↪ çalışdırmadım. siz dir(sys) yazaraq çalışdırın
```

```
[4]: import os # os modulu
```

```
[5]: # dir(os)
```

```
[6]: cd E:\Tain\Python3-AZ
```

E:\Tain\Python3-AZ

```
[7]: # ls
```

Və os modulunun olduğu qovluğu açırıq

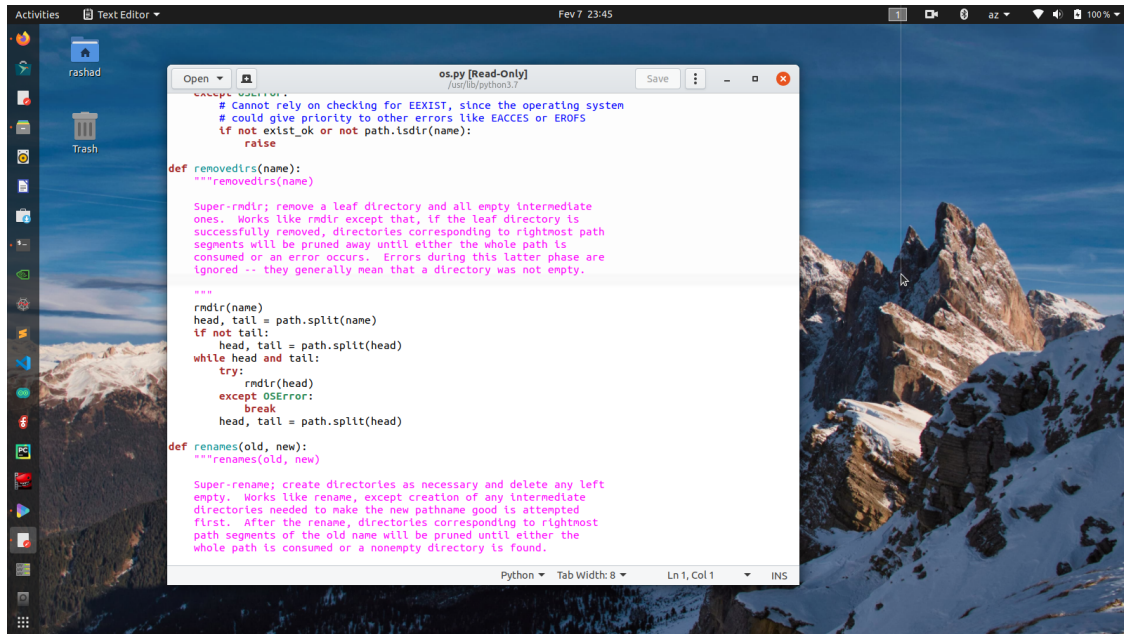
```
[8]: #Susmaya görə əvvəlki qovluğa qayıdırıq
```

```
[9]: cd E:\Tain\Python3-AZ
```

E:\Tain\Python3-AZ

```
[10]: from IPython.display import Image
Image("image/screen.png")
```

```
[10]:
```



Daxilində funksiyaların olduğunu görürük.Və biz import os yazaraq modulu import edirik daha sonra daxilindəki işimizə yararlı funksiya və ya sinifləri çağırırıq.

```
[11]: import os
      os.name
```

```
[11]: 'nt'
```

Yuxarıdakı nümunədə os modulunu çağırırıq daha sonra os modulu daxilində funksiya və sinifləri istifadə edirik

Şəxsi modulumuzu hazırlasaq,siz bu bəhsi tamamilə daha yaxşı mənimsəyəcəksiniz.

```
[12]: ls
```

```
Volume in drive E is TOSHIBA EXT
Volume Serial Number is 5297-426A
```

```
Directory of E:\Tain\Python3-AZ
```

```

03/18/2020  11:43 AM    <DIR>          .
03/18/2020  11:43 AM    <DIR>          ..
03/17/2020  01:40 AM    <DIR>          .idea
02/07/2020  10:25 AM    <DIR>          .ipynb_checkpoints
03/18/2020  11:40 AM    <DIR>          __pycache__
12/13/2019  01:00 PM                2 CONTRIBUTING.md
03/16/2020  11:53 AM                286 debug.log
03/18/2020  11:41 AM                479 discriminant.py

```

```

03/16/2020 06:28 AM          487 function.py
03/16/2020 03:47 AM      89,896 Functions.ipynb
01/19/2020 01:29 PM    3,972,047 Giris-DataTypes.ipynb
03/18/2020 11:27 AM    <DIR>      image
03/18/2020 11:43 AM    1,475,596 Modules(Modullar).ipynb
01/29/2020 12:29 PM    109,139 Operators.ipynb
12/20/2019 10:24 AM      8,524 README.md
03/16/2020 11:23 AM      125 test.py
      10 File(s)      5,656,581 bytes
      6 Dir(s)  395,332,726,784 bytes free

```

Olduğumuz qovluqda python faylı açırıq

```
[17]: echo > function.py
```

```
[18]: ! dir
```

```

Volume in drive E is TOSHIBA EXT
Volume Serial Number is 5297-426A

```

Directory of E:\Tain\Python3-AZ

```

03/18/2020 11:43 AM    <DIR>      .
03/18/2020 11:43 AM    <DIR>      ..
03/17/2020 01:40 AM    <DIR>      .idea
02/07/2020 10:25 AM    <DIR>      .ipynb_checkpoints
12/13/2019 01:00 PM          2 CONTRIBUTING.md
03/16/2020 11:53 AM      286 debug.log
03/18/2020 11:41 AM      479 discriminant.py
03/18/2020 11:46 AM      13 function.py
03/16/2020 03:47 AM    89,896 Functions.ipynb
01/19/2020 01:29 PM    3,972,047 Giris-DataTypes.ipynb
03/18/2020 11:27 AM    <DIR>      image
03/18/2020 11:43 AM    1,475,596 Modules(Modullar).ipynb
01/29/2020 12:29 PM    109,139 Operators.ipynb
12/20/2019 10:24 AM      8,524 README.md
03/16/2020 11:23 AM      125 test.py
03/18/2020 11:40 AM    <DIR>      __pycache__
      10 File(s)      5,656,107 bytes
      6 Dir(s)  395,332,726,784 bytes free

```

Hazırladığımız function.py faylına əlavələrimizi edək

```

[21]: #function.py faylı
print("""
def toplama(x,y):
    result = x+y
    return result

```

```
"""
```

```
def toplama(x,y):  
    result = x+y  
    return result
```

Yazdığımız funksiya digər faylda olduğu üçün artıq həmin fayl bizim üçün bir modul sayılır. Modulu çağırmağın ilk metoduna nəzər salaq

```
[22]: import function
```

Modulu import ifadəsi ilə çağırdıq və aşağı sətərə xətasız keçid etdik. İlk əvvəl qeyd etdiyim kimi python-da modulu çağırarkən ilk olaraq Cari kataloq a nəzər yetirir daha sonra pythonpath kataloqu yoxlanılır. Faylımız cari kataloqda olduğu üçün pythonpath bölməsinə keçid etmədi

```
[23]: dir(function)
```

```
[23]: ['__builtins__',  
      '__cached__',  
      '__doc__',  
      '__file__',  
      '__loader__',  
      '__name__',  
      '__package__',  
      '__spec__',  
      'toplama']
```

dir funksiyası vasitəsilə modul daxilində nələrin olduğunu əyani göstərək

Yuxarıda qeyd olunanları irəliləyən bəhslərdə öyrənəcəyik. Hal-hazırda sonda addition ifadəsini görürük-bu bizim yazdığımız funksiya. gər fayl daxilində bir neçə funksiya və ya sinif qeyd etsəydik hər biri ekrana çap olunacaqdı

```
[24]: #funksiyadan istifadə edək  
import function  
function.toplama()
```

```
-----  
TypeError                                Traceback (most recent call last)  
<ipython-input-24-dcbd7f3ef360> in <module>()  
      1 #funksiyadan istifadə edək  
      2 import function  
>>> 3 function.toplama()  
  
TypeError: toplama() missing 2 required positional arguments: 'x' and 'y'
```

Və xəta aldığımız. Bu xətalara funksiyalar bəhsində öyrənmişdik. Çünki funksiya 2 arqument aldığı halda biz boş ötürdük

```
[25]: #funksiyadan istifadə edək
import function
function.toplama(2,3) #2+3 = 5
```

[25]: 5

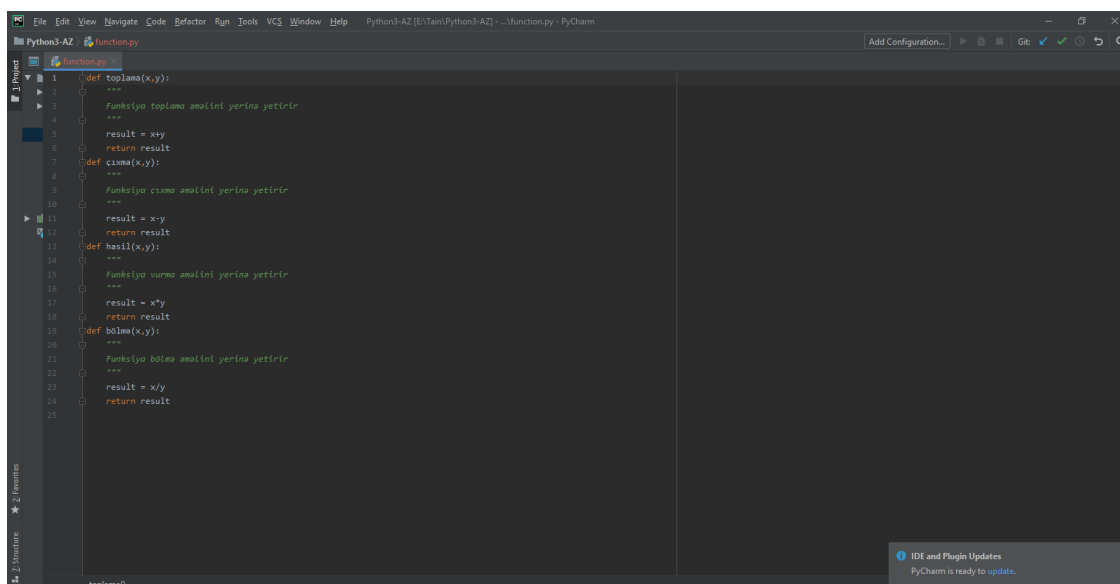
Bəli, moduldan rahatlıqla istifadə etdik.

Faylımıza digər funksiyalar da əlavə edək

Hazırladığımız python faylını pycharm mətn editorunda açırıq

```
[26]: from IPython.display import Image
Image("image/moduls.png")
```

[26]:



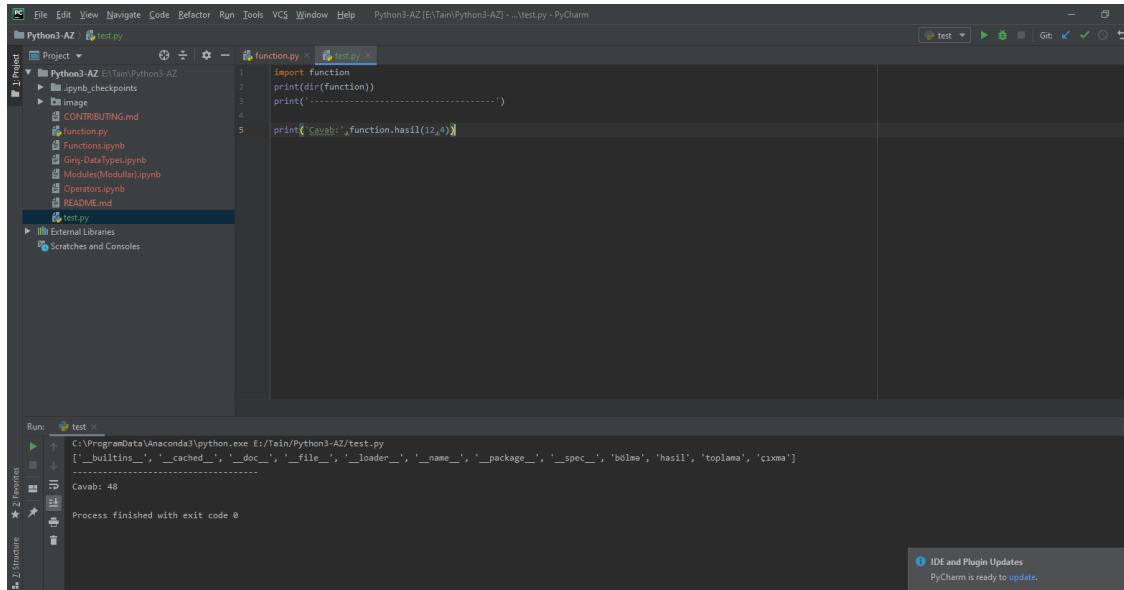
python faylına bir neçə riyazi əməlləri yerinə yetirən funksiyalar əlavə etdim. Burada function.py fayl adı yəni function modul adıdır.

Və modulu istənilən fayldan çağıraraq istifadə edə bilərik

Növbəti test.py faylını eyni qovluqda hazırlayırıq. və function modulumuzu həmin python faylından çağırmaq

```
[28]: from IPython.display import Image
Image("image/moduls1.png")
```

[28]:



Yuxarıda gördüyünüz kimi modul daxilində hasil funksiyasından istifadə etdik.İstifadədən öncə isə modul daxilində digər elementləri də ekrana çap etdik.

Orta məktəbdə keçdiyimiz diskriminant başqa sözlə kvadrat tənlik həllini python dilində ifadə etmək Kvadrat tənlik nədir ilk öncə izah edim daha sonra modul hazırlayaraq digər fayllardan diskriminant funksiyasından istifadə etmək

```
[29]: from IPython.display import Image
Image("image/diskrt.png")
```

[29]:

Kvadrat tənlik — $ax^2 + bx + c = 0$, ($a \neq 0$) şəklində olan **tənliyə** deyilir.

-Burada a,b,c sabit ədədlər, x isə məchuldur. a- birinci əmsal,b- ikinci əmsal,c- sərbəst hədd adlanır.

- Birinci həddin əmsalı (yəni a) 1-ə bərabər olan kvadrat tənlik Çevrilmiş kvadrat tənlik adlanır

Kvadrat tənlik yuxarıdakı şəkildə ifadə olunur.

həqiqi əmsallı kvadrat tənliyinin $D=b^2-4ac$ diskriminantının qiymətindən asılı olaraq 1 ya 2 kökü ola bilər, ya da kökü olmaz

gər $D>0$ olduqda tənliyin 2 müxtəlif kökü var

$D=0$ olduqda tənliyin 2 bərabər kökü var

$D<0$ olduqda isə tənliyin həqiqi kökü yoxdur.

[Mənbə](#)

[46]: !dir

Volume in drive E is TOSHIBA EXT
Volume Serial Number is 5297-426A

Directory of E:\Tain\Python3-AZ

```
03/18/2020  11:57 AM    <DIR>          .
03/18/2020  11:57 AM    <DIR>          ..
03/17/2020  01:40 AM    <DIR>          .idea
02/07/2020  10:25 AM    <DIR>          .ipynb_checkpoints
12/13/2019  01:00 PM                2 CONTRIBUTING.md
03/16/2020  11:53 AM                286 debug.log
03/18/2020  11:55 AM                449 discriminant.py
03/18/2020  11:50 AM                488 function.py
03/16/2020  03:47 AM            89,896 Functions.ipynb
01/19/2020  01:29 PM        3,972,047 Giris-DataTypes.ipynb
03/18/2020  11:27 AM    <DIR>          image
03/18/2020  11:57 AM        1,476,665 Modules(Modullar).ipynb
01/29/2020  12:29 PM        109,139 Operators.ipynb
12/20/2019  10:24 AM                8,524 README.md
03/18/2020  11:53 AM                56 test.py
03/18/2020  11:54 AM    <DIR>          __pycache__
          10 File(s)          5,657,552 bytes
          6 Dir(s)  395,332,706,304 bytes free
```

Eyniadlı qovluqda növbəti discriminant.py adlı python faylı hazırladım və daxilinə aşağıdakı funksiyamı əlavə etdim

```
[55]: print("""

def diskriminant(a,b,c):

    a = float(a)
    b = float(b)
    c = float(c)
    D = b**2 - 4*a*c;
    if D>0:
        x1 = (-b + (D)**2)/(2*a);
        x2 = (-b - (D)**2)/(2*a);
        return ('X1: {} X2: {}'.format(x1,x2))

    elif D==0:
        x1 = -b/(2*a);
        return ('X1=X2 : {}'.format(x1))
    else:
        print('Tənliyin həqiqi kökləri yoxdur..')
```



```
"""
```

```
def diskriminant(a,b,c):  
  
    a = float(a)  
    b = float(b)  
    c = float(c)  
    D = b**2 - 4*a*c;  
    if D>0:  
        x1 = (-b + (D)**2)/(2*a);  
        x2 = (-b - (D)**2)/(2*a);  
        return ('X1: {} X2: {}'.format(x1,x2))  
  
    elif D==0:  
        x1 = -b/(2*a);  
        return ('X1=X2 : {}'.format(x1))  
    else:  
        print('Tənliyin həqiqi kökləri yoxdur..')
```

Və modulu eləcə də modulun funksiyasını çağırmaq üçün

```
[56]: import discriminant
```

```
[57]: dir(discriminant)
```

```
[57]: ['__builtins__',  
      '__cached__',  
      '__doc__',  
      '__file__',  
      '__loader__',  
      '__name__',  
      '__package__',  
      '__spec__',  
      'diskriminant']
```

```
[59]: discriminant.diskriminant(1,2,-1.5)
```

```
a:1  
b:2  
c:-1.5  
X1: 49.0 X2: -51.0
```

```
[59]: (49.0, -51.0)
```