

Advanced Modules

May 23, 2022

0.1 Advanced Modules (Pythonda qabaqcıl modullar)

Modullar bəhsində biz şəxsi modullarımızı hazırlayırdıq. Bundan başqa python ilə bərabər gələn qabaqcıl modullar da mövcuddur.

0.1.1 Os modulu

```
[1]: import os
```

```
[8]: siyahı = []  
for i in dir(os):  
    if '_' not in i:  
        siyahı.append(i)  
print(siyahı)
```

```
['DirEntry', 'MutableMapping', 'PathLike', 'abc', 'abort', 'access', 'altsep',  
'chdir', 'chmod', 'close', 'closerange', 'curdir', 'defpath', 'devnull', 'dup',  
'dup2', 'environ', 'errno', 'error', 'execl', 'execle', 'execlp', 'execlpe',  
'execv', 'execve', 'execvp', 'execvpe', 'extsep', 'fdopen', 'fsdecode',  
'fsencode', 'fspath', 'fstat', 'fsync', 'ftruncate', 'getcwd', 'getcwdb',  
'getenv', 'getlogin', 'getpid', 'getppid', 'isatty', 'kill', 'linesep', 'link',  
'listdir', 'lseek', 'lstat', 'makedirs', 'mkdir', 'name', 'open', 'pardir',  
'path', 'pathsep', 'pipe', 'popen', 'putenv', 'read', 'readlink', 'remove',  
'removedirs', 'rename', 'renames', 'replace', 'rmdir', 'scandir', 'sep',  
'spawnl', 'spawnle', 'spawnv', 'spawnve', 'st', 'startfile', 'stat', 'strerror',  
'symlink', 'sys', 'system', 'times', 'truncate', 'umask', 'unlink', 'urandom',  
'utime', 'waitpid', 'walk', 'write']
```

Yuxarıda bir neçə funksiyamı bu bəhsdə öyrənəcəksiniz.

```
[10]: import os  
os.name #əməliyyat sistemi
```

```
[10]: 'nt'
```

Windows -> 'nt' 'dos' 'ce' ; Linux -> 'posix'

```
[17]: import os  
os.sep #Windows əməliyyat sistemi
```

```
[17]: '\\'
```

```
[18]: import os
      os.getcwd() #hansı qovluqda olduğumuzu bildirir
```

```
[18]: 'E:\\Tain\\Python3-AZ\\Advanced-Modules'
```

```
[20]: import os
      os.chdir('E:\\Tain\\Python3-AZ') #Qovluqlar arası keçid
      os.getcwd()
```

```
[20]: 'E:\\Tain\\Python3-AZ'
```

```
[21]: import os
      os.listdir() #olduğumuz qovluqda fayl və qovluqları göstərir
```

```
[21]: ['.git',
      '.idea',
      'Advanced-Modules',
      'Built-in-Function',
      'CONTRIBUTING.md',
      'DATA',
      'Data_types',
      'debug.log',
      'Decorators',
      'Error_Exceptions',
      'Files',
      'Function',
      'image',
      'Iterators-Generators',
      'Modules',
      'OOP',
      'Operators',
      'Package',
      'README.md',
      '__pycache__']
```

```
[23]: # gər təyinatı ünvan yazsanız,həmin qovluq daxilindəki fayl və qovluqları
      ↪göstərəcək(əgər qovluq və ya fayl varsa)
      import os
      os.listdir('C:')
```

```
[23]: ['$360Section',
      '$Recycle.Bin',
      '360SANDBOX',
      'Config.Msi',
      'cudnn',
```

```
'Documents and Settings',  
'Games',  
'hiberfil.sys',  
'Intel',  
'OneDriveTemp',  
'pagefile.sys',  
'PerfLogs',  
'Program Files',  
'Program Files (x86)',  
'ProgramData',  
'Recovery',  
'swapfile.sys',  
'System Volume Information',  
'Users',  
'Windows']
```

```
[7]: import os  
siyahı = []  
image_direc = '../image/'  
read_im = [image_direc+i for i in os.listdir(image_direc)]  
for i in read_im:  
    siyahı.append(i)  
    print(i)
```

```
../image/anac1.png  
../image/anac2.png  
../image/anac3.png  
../image/anaconda.png  
../image/datatypes.png  
../image/diskrt.png  
../image/f.png  
../image/ieee.jpg  
../image/inhert.png  
../image/loop.jpg  
../image/map.png  
../image/moduls.png  
../image/moduls1.png  
../image/operator.png  
../image/operators.png  
../image/pack.png  
../image/pack1.png  
../image/package.png  
../image/pyimg.jpg  
../image/pyscript.png  
../image/pytest.png  
../image/reduce.png
```

```
../image/rerunpy.png
../image/runpy.png
../image/savepy.png
../image/screen.png
../image/stack (1).jpg
../image/stack.webp
../image/start.png
../image/strindex.png
../image/t.png
```

```
[26]: import os
      os.listdir(os.getcwd()) # əməliyyat sistemində ana qovluğu göstərir.
```

```
[26]: ['.git',
      '.idea',
      'Advanced-Modules',
      'Built-in-Function',
      'CONTRIBUTING.md',
      'DATA',
      'Data_types',
      'debug.log',
      'Decorators',
      'Error_Exceptions',
      'Files',
      'Function',
      'image',
      'Iterators-Generators',
      'Modules',
      'OOP',
      'Operators',
      'Package',
      'README.md',
      '__pycache__']
```

```
[27]: import os
      os.listdir(os.pardir) #bir qovluq geri,qovluqdakı qovluq və faylları göstərir
```

```
[27]: ['AI',
      'Car_Count',
      'Computing',
      'Computing vision',
      'data',
      'dataset',
      'face-dataset',
      'Face-recognition',
      'fire',
      'FireDetection',
```

```
'model',
'New folder (2)',
'Python3-AZ',
'PythonProqramlaşdırmaDili']
```

```
[29]: import os
os.startfile('pythonaz.pdf') #Yalnız Windows əməliyyat sistemində çalışır
```

```
-----
FileNotFoundError                                Traceback (most recent call last)
<ipython-input-29-7f2b0b829323> in <module>()
      1 import os
----> 2 os.startfile('pythonaz.pdf') #Yalnız Windows əməliyyat sistemində çalışır

FileNotFoundError: [WinError 2] The system cannot find the file specified:
↳ 'pythonaz.pdf'
```

```
[31]: import os
os.mkdir('example') #Olduğunuz qovluqda yeni qovluq yaradır.Təyinatı ünvan
↳ verilərsə həmin ünvanda qovluğu yaradacaq
#os.mkdir('C:\example')
```

```
[34]: os.listdir()
```

```
[34]: ['.git',
'.idea',
'Advanced-Modules',
'Built-in-Function',
'CONTRIBUTING.md',
'DATA',
'Data_types',
'debug.log',
'Decorators',
'Error_Exceptions',
'example',
'Files',
'Function',
'image',
'Iterators-Generators',
'Modules',
'OOP',
'Operators',
'Package',
'README.md',
'__pycache__']
```

```
[35]: import os
os.makedirs('example/new') #Qovluq altında yeni qovluq yaradır.
```

```
[36]: import os
os.rename('example','rename') #Qovluq adını dəyişə bilərsiniz
```

```
[37]: os.listdir()
```

```
[37]: ['.git',
      '.idea',
      'Advanced-Modules',
      'Built-in-Function',
      'CONTRIBUTING.md',
      'DATA',
      'Data_types',
      'debug.log',
      'Decorators',
      'Error_Exceptions',
      'Files',
      'Function',
      'image',
      'Iterators-Generators',
      'Modules',
      'OOP',
      'Operators',
      'Package',
      'README.md',
      'rename',
      '__pycache__']
```

```
[ ]:
```

```
[ ]:
```

```
[41]: import os
os.replace('rename','example') #rename ilə eyni işi görür.fərq ondadırki əgər
↳həmin adda qovluq varsa qovluq tamamilə silinib yenidən yeni adla yaradılır
```

```
[47]: import os
os.rmdir('example/new')# Sadəcə boş qovluğu silir.alt qovluqları silmir və xəta
↳alacaqsınız
```

```
[48]: import os
os.removedirs('example')#rmdir fərqli olaraq başlanğıc qovluqdan alt paketlərə
↳qədər tamamilə silir
```

```
[49]: import os
os.stat('Package')# qovluqlar haqqında məlumat verir
```

```
[49]: os.stat_result(st_mode=16895, st_ino=1688849860282229, st_dev=1385644650,
st_nlink=1, st_uid=0, st_gid=0, st_size=4096, st_atime=1585513628,
st_mtime=1585513628, st_ctime=1585497963)
```

```
[50]: #st_atime -qovluğa son giriş tarixi
#st_ctime -qovluğun yaranma tarixi
#st_mtime -qovluğun dəyişdirilmə tarixi
#st_size -qovluğun həcmi
```

```
[60]: import os
os.path.abspath('cudnn64_7.dll') # faylın ünvanını göstərir
```

```
[60]: 'E:\\Tain\\Python3-AZ\\cudnn64_7.dll'
```

```
[62]: import os
os.path.dirname('cudnn64_7.dll') # faylın qovluq ünvanını göstərir
```

```
[62]: ''
```

```
[ ]:
```

0.1.2 sys modulu

```
[11]: import sys
#sys.exit() proqramdan çıxış üçün istifadə edə bilərsiniz

def sistem():
    var = int(input(':'))
    if var < 0:
        print('Proqram sonlandırıldı...')
        sys.exit()
    else:
        return var*2
sistem()
```

```
:-1
```

Proqram sonlandırıldı...

An exception has occurred, use %tb to see the full traceback.

SystemExit

```
C:\ProgramData\Anaconda3\lib\site-  
packages\IPython\core\interactiveshell.py:2918: UserWarning: To exit: use  
'exit', 'quit', or Ctrl-D.  
    warn("To exit: use 'exit', 'quit', or Ctrl-D.", stacklevel=1)
```

```
[ ]:
```

```
[12]: import sys  
      #sys.argv  
      print("This is the name of the program:", sys.argv[0])  
      print("Number of elements including the name of the program:", len(sys.argv))  
      print("Number of elements excluding the name of the program:", (len(sys.  
      ↪argv)-1))  
      print("Argument List:", str(sys.argv))
```

```
This is the name of the program: C:\ProgramData\Anaconda3\lib\site-  
packages\ipykernel_launcher.py  
Number of elements including the name of the program: 3  
Number of elements excluding the name of the program: 2  
Argument List: ['C:\\ProgramData\\Anaconda3\\lib\\site-  
packages\\ipykernel_launcher.py', '-f', 'C:\\Users\\garay\\AppData\\Roaming\\jup  
yter\\runtime\\kernel-6ae3fcb5-6ef2-4e46-b99a-620edfb656df.json']
```

```
[14]: import sys  
      sys.executable
```

```
[14]: 'C:\\ProgramData\\Anaconda3\\python.exe'
```

```
[18]: import sys  
      sys.platform #Çalışdığı əməliyyat sistemi haqqında məlumat
```

```
[18]: 'win32'
```

```
[19]: import sys  
      sys.prefix #Çalışan proqramın hansı qovluqda yerləşdiyini göstərir
```

```
[19]: 'C:\\ProgramData\\Anaconda3'
```

```
[20]: import sys  
      sys.version #çalışan proqramın versiyası haqqında məlumat verir
```

```
[20]: '3.6.10 |Anaconda, Inc.| (default, Jan 7 2020, 15:18:16) [MSC v.1916 64 bit  
(AMD64)]'
```

```
[ ]:
```


0.1.3 Python Regex (re modulu)

```
[21]: import re
```

```
[22]: print(dir(re))
```

```
['A', 'ASCII', 'DEBUG', 'DOTALL', 'I', 'IGNORECASE', 'L', 'LOCALE', 'M',  
'MULTILINE', 'RegexFlag', 'S', 'Scanner', 'T', 'TEMPLATE', 'U', 'UNICODE',  
'VERBOSE', 'X', '_MAXCACHE', '__all__', '__builtins__', '__cached__', '__doc__',  
'__file__', '__loader__', '__name__', '__package__', '__spec__', '__version__',  
'_alphanum_bytes', '_alphanum_str', '_cache', '_compile', '_compile_repl',  
'_expand', '_locale', '_pattern_type', '_pickle', '_subx', 'compile', 'copyreg',  
'enum', 'error', 'escape', 'findall', 'finditer', 'fullmatch', 'functools',  
'match', 'purge', 'search', 'split', 'sre_compile', 'sre_parse', 'sub', 'subn',  
'template']
```

```
[35]: import re  
#metod sadəcə ilk verilənə baxır  
data = 'python is very best language'  
result = re.match('python',data)  
print(result.group())
```

python

```
[37]: # search() metodu
```

```
[45]: import re  
data = 'python is very best language'  
result = re.search('best',data)  
print(result.group())
```

best

```
[68]: #findall() metodu  
#Metod mətn daxilində təkrar olunan elementləri göstərir.  
import re  
text="""  
There are 10 types of people.  
Those who know binary and those who don't.  
I said: 'There are 10 types of people.'  
I also said: 'Those who know binary and those who don't.'  
Isn't that joke so funny?! False  
This is the left side of...a string with a right side.  
"""  
print(re.findall('There',text))
```

['There', 'There']

```
[69]: print("""
      \d işarəsi 0-dan 9-a qədər ədədləri ifadə edir [0-9].
      \D [^0-9].
      \s [ \t\n\r\f\v].
      \S [^ \t\n\r\f\v].
      \w [a-zA-Z0-9_].
      \W [^a-zA-Z0-9_].

      """)
```

```
\d işarəsi 0-dan 9-a qədər ədədləri ifadə edir [0-9].
\D [^0-9].
\s [
    ].
\S [^
    ].
\w [a-zA-Z0-9_].
\W [^a-zA-Z0-9_].
```

```
[73]: import re
      list=['john','mohn','leon','julion','jon']
      for i in list:
          if re.match('j.',i):
              print(i)
```

```
john
julion
jon
```

```
[74]: import re
      list=['eat','meat','missing','meal','mui']
      for i in list:
          if re.search('ea*',i):
              print(i)
```

```
eat
meat
meal
```

```
[80]: import re
list=['ədalət','fəlakət','kəramət','məlahət','ülviyyə','Nizami','Eldar','Arzu']
for i in list:
    if re.match('.*u',i):
        print(i)
```

Arzu

```
[82]: import re,os
i=os.listdir(os.getcwd())
for v in i:
    if re.match('.*mp',v):
        print(i)
```

```
[88]: import re,os
def search():
    qovluq=os.listdir('E:\programlar\OS-system')
    for i in qovluq:
        if re.match('.*iso',i):
            print(i)
search()
```

linuxmint-19.1-cinnamon-64bit.iso
 MX-18.3_x64.iso
 ubuntu-19.04-desktop-amd64_2.iso
 Win10_1909_English_x64.iso

```
[91]: import re
list=['ədalət','fəlakət','kəramət','ət','ələt','məlahət','ülviyyə']
for i in list:
    if re.search('.*əl',i):
        print (i)
```

fəlakət
 məlahət

```
[93]: import re
list=['s','sa','saa','saat','saaat','slaaat']
for i in list:
    if re.match('sa{2}t',i):
        print(i)
```

saat

```
[94]: import re
list=['s','sa','saa','saat','saaat','slaaat']
for i in list:
    if re.match('sa{1,3}t',i):
```

```
print(i)
```

saat
saaat

```
[95]: import re
data=['564343','fgdg454','45gfhfh','fghf34hg','fgdgd','rtyuqw']
for i in data:
    if re.search('[0-9]',i):
        print(i)
```

564343
45gfhfh

```
[96]: import re
data=['564343','fgdg454','45gfhfh','fghf34hg','fgdgd','rtyuqw','AzsAy12s']
for i in data:
    if re.search('[a-z]',i):
        print(i)
```

fgdg454
fghf34hg
fgdgd
rtyuqw

```
[97]: import re
data=['564343','Az1','azAb12']
for i in data:
    if re.search('[A-Z]',i):
        print(i)
```

Az1

```
[98]: import re
data=['12$','10$','30€']
for i in data:
    if re.match('[0-9]+\$',i):
        print(i)
```

12\$
10\$

```
[99]: import re
data=['12$','10$','30€']
for i in data:
    if re.match('[0-9]+\€',i):
        print(i)
```

30€

```
[100]: import re
from urllib.request import urlopen
url="http://www.asan.gov.az/"
i=urlopen(url)
v="href=+.css"
for k in i:
    n=re.search(v,str(k))
    if n:
        print(n.group())
```

```
href="https://asan.gov.az/css/bootstrap.min.css
href="https://asan.gov.az/css/style.css
href="https://asan.gov.az/css/responsive.css
```

```
[104]: import re
#Simvol,daxilində boşluqları olan verilənləri aşkar edir
data=['5 Frank','Octan 7','5elli']
for i in data:
    x=re.search("[A-Za-z]+\s[0-9]",i)
    if x:
        print(x.group())
```

Octan 7

```
[105]: import re
data=['java_3.1','python.4','oracle_2.6','nagios_1.2','litr12']
tertib=re.compile('[a-z_]+[0-9]\.[0-9]')
for i in data:
    v=tertib.search(i)
    if v:
        print(v.group())
```

```
java_3.1
oracle_2.6
nagios_1.2
```

```
[ ]:
```

0.1.4 datetime modulu

```
[106]: import datetime
```

```
[107]: print(dir(datetime))
```

```
['MAXYEAR', 'MINYEAR', '__builtins__', '__cached__', '__doc__', '__file__',
 '__loader__', '__name__', '__package__', '__spec__', 'date', 'datetime',
```

```
'datetime_CAPI', 'sys', 'time', 'timedelta', 'timezone', 'tzinfo']
```

```
[108]: from datetime import datetime
```

```
[109]: print(dir(datetime))
```

```
['__add__', '__class__', '__delattr__', '__dir__', '__doc__', '__eq__',  
 '__format__', '__ge__', '__getattribute__', '__gt__', '__hash__', '__init__',  
 '__init_subclass__', '__le__', '__lt__', '__ne__', '__new__', '__radd__',  
 '__reduce__', '__reduce_ex__', '__repr__', '__rsub__', '__setattr__',  
 '__sizeof__', '__str__', '__sub__', '__subclasshook__', 'astimezone', 'combine',  
 'ctime', 'date', 'day', 'dst', 'fold', 'fromordinal', 'fromtimestamp', 'hour',  
 'isocalendar', 'isoformat', 'isoweekday', 'max', 'microsecond', 'min', 'minute',  
 'month', 'now', 'replace', 'resolution', 'second', 'strftime', 'strptime',  
 'time', 'timestamp', 'timetuple', 'timetz', 'today', 'toordinal', 'tzinfo',  
 'tzname', 'utcfromtimestamp', 'utcnow', 'utcoffset', 'utctimetuple', 'weekday',  
 'year']
```

```
[111]: from datetime import datetime
```

```
[112]: time = datetime.now()
```

```
[113]: print(time)
```

```
2020-03-30 06:37:36.323085
```

```
[114]: print(time.year)
```

```
2020
```

```
[115]: print(time.month)
```

```
3
```

```
[116]: print(time.day)
```

```
30
```

```
[117]: print(time.hour)
```

```
6
```

```
[118]: #today()  
from datetime import datetime  
zaman = datetime.today()
```

```
[119]: print(zaman)
```

```
2020-03-30 06:39:01.456433
```

```
[120]: print(zaman.year)
```

2020

```
[121]: print(zaman.month)
```

3

```
[122]: #ctime
from datetime import datetime
zaman = datetime.ctime
```

```
[123]: print(zaman)
```

<method 'ctime' of 'datetime.datetime' objects>

```
[124]: from datetime import datetime
zaman = datetime.ctime(datetime.now())
```

```
[125]: print(zaman)
```

Mon Mar 30 06:41:13 2020

```
[126]: #strftime()
```

%a -həftəni ifadə edirik.(ixtisarla) %A -həftəni bütünlükdə ifadə edirik. %b -ayı ifadə edir.(İxtisarla) %B -ayı bütünlükdə ifadə edir. %c -tarix və saati ifadə edir. %d -günü sözlə ifadə edir. %j -tarixin hansısa bir ildə uyğun gəldiyi gununu ifadə edirik. %m -ayı rəqəmlə ifadə edir. %U -bir tarixin ilin neçənci həftəsinə uyğun gəldiyini təyin edir. %y -İlin son iki rəqəmi %Y -ili ifadə edir. %x -tarixi bütünlükdə ifadə edir. %X -saati bütünlükdə ifadə edir.

```
[130]: zaman = datetime.now()
zaman = datetime.strftime(zaman,'%A')
print(zaman)
```

Monday

```
[131]: zaman = datetime.now()
zaman = datetime.strftime(zaman,'%c')
print(zaman)
```

Mon Mar 30 06:43:26 2020

```
[132]: zaman = datetime.now()
zaman = datetime.strftime(zaman,'%B')
print(zaman)
```

March

```
[138]: import locale
       #Olduğunuz coğrafi bölgəyə uyğun zaman məlumatını bölgə dilinə uyğun
```

```
[139]: locale.setlocale(locale.LC_ALL, 'az')
```

```
[139]: 'az'
```

```
[140]: zaman = datetime.now()
       zaman = datetime.strftime(zaman, '%B')
       print(zaman)
```

Mart

```
[144]: zaman = datetime.now()
       zaman = datetime.strftime(zaman, '%A')
       print(zaman)
```

bazar ertəsi

```
[145]: zaman = datetime.now()
       zaman = datetime.strftime(zaman, '%d %B %Y')
       print(zaman)
```

30 Mart 2020

0.1.5 time modulu

```
[146]: import time
       seconds = time.time()
       print("Seconds since epoch =", seconds)
```

Seconds since epoch = 1585576082.9815543

```
[147]: import time
       time.gmtime()
```

```
[147]: time.struct_time(tm_year=2020, tm_mon=3, tm_mday=30, tm_hour=13, tm_min=48,
       tm_sec=17, tm_wday=0, tm_yday=90, tm_isdst=0)
```

```
[149]: import time

       seconds = 0
       local_time = time.ctime(seconds)
       print(local_time)
```

Wed Dec 31 16:00:00 1969


```
[151]: import time
print("Salam,Gözləyin...")
time.sleep(4)
print("Dörd saniyə gözlətdirdim...")
```

Salam,Gözləyin...
Dörd saniyə gözlətdirdim...

```
[ ]:
```

0.1.6 random modulu

```
[158]: import random

random.seed(0)
print(random.random())
```

0.8444218515250481

```
[168]: import random
#syntax
#random.randrange(start, stop, step)
print(random.randrange(2,10))
```

9

```
[169]: import random
print(random.randrange(2,10))
```

7

```
[170]: import random
print(random.randrange(0,100,10))
```

90

```
[171]: import random
#syntax
#random.randint(start, stop)
print(random.randint(2,10))
```

5

```
[172]: import random
#syntax
#random.randint(start, stop)
print(random.randint(2,10))
```

10

```
[175]: import random
#syntax
#random.sample(sequence, k)
data = ['Arzu', 'Nadir', 'Mətanət', 'Xəyalə', 'Ruslan', 'Fazil', 'Vahid']
result = random.sample(data,1)
print(result)
```

['Mətanət']

```
[178]: import random

data = [i for i in range(100)]
result = random.sample(data,1)
print(result)
```

[12]

```
[181]: data = ['Arzu', 'Nadir', 'Mətanət', 'Xəyalə', 'Ruslan', 'Fazil', 'Vahid']
result=random.shuffle(data)
print(data)
```

['Fazil', 'Arzu', 'Xəyalə', 'Nadir', 'Vahid', 'Mətanət', 'Ruslan']

```
[182]: import random
def f():
    return 0.1

data = ['Arzu', 'Nadir', 'Mətanət', 'Xəyalə', 'Ruslan', 'Fazil', 'Vahid']
random.shuffle(data, f)

print(data)
```

['Nadir', 'Mətanət', 'Xəyalə', 'Ruslan', 'Fazil', 'Vahid', 'Arzu']

```
[184]: data = [i for i in range(51)]
print(data)
random.choice(data)
```

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50]

[184]: 4

[]: