



PYTHON PROGRAMLAMA DİLİ

random, math, datetime

Dr. Öğr. Üyesi M. Ozan AKI

RANDOM

Bazı durumlarda, sıradaki sayının öngörülemediği rasgele bir sayıya ya da sayı dizisine ihtiyacımız olabilir.

```
import random
```

Özellikle oyunlarda, botlarda ya da yazılım testlerinde rasgele sayılar oldukça yararlıdırlar.

Rasgele sayıları kullanabilmek için;

random

modülü kullanılır.

RANDOM

seed(x): Rasgele sayı üretici algoritmaya bir x başlangıç değeri verir. *(aynı başlangıç sayısında aynı rasgele sayılar üretilir)*

getrandbits(b): b bitlik bir rasgele sayı döndürür.

randrange(a): 0'dan a'ya kadar rasgele bir sayı döndürür.

randrange(a,b): a ile b arasında rasgele bir sayı döndürür.

randrange(a,b,i): a ile b arasında i artımlı rasgele bir sayı döndürür.

randint(a,b): a ile b arasında rasgele bir sayı döndürür.

```
>>> import random
>>> random.seed(123)
>>>
>>> for i in range(10):
...     x=random.getrandbits(4)
...     print(f'{x} ', end='')
...
0 4 1 12 6 4 1 13 14 13
>>>
>>> for i in range(10):
...     x=random.randrange(100,200)
...     print(f'{x} ', end='')
...
104 148 168 171 142 143 106 120 117 143
>>>
>>> for i in range(10):
...     x=random.randrange(100,200,10)
...     print(f'{x} ', end='')
...
180 150 130 120 100 160 110 190 160 110
>>>
>>> for i in range(10):
...     x=random.randint(10,20)
...     print(f'{x} ', end='')
...
10 15 17 11 10 11 20 12 12 10
>>> |
```

RANDOM

choice(x): Verilen **x** listesinden rasgele bir eleman seçip döndürür.

choices(x,k): Verilen **x** listesinden **k** sayıda rasgele eleman seçerek döndürür. *(aynı elemanlar tekrar seçilebilir.)*

choices(x,weights,k): Verilen **x** listesinden **k** sayıda rasgele elemanı ağırlıklarına (**weights**) göre seçerek döndürür. *(aynı elemanlar tekrar seçilebilir.)*

```
>>> import random
>>>
>>> liste=['ali','banu','can','didem','efe']
>>>
>>> random.choice(liste)
'efe'
>>> random.choice(liste)
'didem'
>>> random.choices(liste,k=3)
['banu', 'efe', 'didem']
>>> random.choices(liste,k=3)
['banu', 'efe', 'banu']
>>> random.choices(liste,k=3)
['didem', 'can', 'efe']
>>>
```

RANDOM

shuffle(x): x ile verilen listedeki elemanların sırasını rasgele karıştırır.

sample(x,k): x ile verilen listeden k adet örnek seçer. *(aynı elemanlar tekrar seçilmez.)*

random(): 0 ile 1 arasında rasgele bir ondalıklı (float) sayı döndürür.

uniform(a,b): a ile b arasında rasgele bir ondalıklı (float) sayı döndürür.

```
>>> import random
>>>
>>> liste=['ali','banu','can','didem','efe']
>>>
>>> random.sample(liste,3)
['banu', 'didem', 'ali']
>>> random.sample(liste,3)
['ali', 'can', 'banu']
>>> random.sample(liste,3)
['can', 'didem', 'efe']
>>>
>>> print(liste)
['ali', 'banu', 'can', 'didem', 'efe']
>>> random.shuffle(liste)
>>> print(liste)
['banu', 'can', 'efe', 'ali', 'didem']
>>>
>>> for i in range(5):
...     x=random.random()
...     print(f'{x:.2f} ',end='')
...
0.51 0.71 0.08 0.90 0.80
>>>
>>> for i in range(5):
...     x=random.uniform(10,20)
...     print(f'{x:.2f} ',end='')
...
13.77 15.47 16.35 10.29 13.22
>>> |
```

MATH

Matematiksel fonksiyonlar **math** modülü altında toplanmıştır.

Bu modülde; kuvvet, karekök, trigonometrik fonksiyonlar, logaritma, sayı yuvarlama, açık dönüşümleri gibi bir çok fonksiyon bulunur.

Bu fonksiyonlara erişmek için **math** modülü **import** edilmelidir.

```
import math
```

MATH

Matematiksel sabitler;

math.pi: Pi sayısı (3.1415...)

math.e: Euler sayısı (2.7182...)

math.tau: Tau sayısı (6.2831...)

math.inf: Artı sonsuz (float)

math.nan: NaN (Not a Number)

```
>>> import math
>>>
>>> math.pi
3.141592653589793
>>>
>>> math.e
2.718281828459045
>>>
>>> math.tau
6.283185307179586
>>>
>>> math.inf
inf
>>>
>>> math.nan
nan
>>>
```

MATH

math.fabs(x): Verilen x sayısının mutlak değerini döndürür.

math.sqrt(): Verilen x sayısının kare kökünü döndürür.

math.exp(x): e^x sayısını döndürür.

math.pow(x,y): x^y sayısını döndürür.

math.log(x): Verilen x sayısının doğal logaritmasını döndürür ($\ln x$)

math.log10(): Verilen x sayısının logaritmasını döndürür ($\log x$)

```
>>> import math
>>>
>>> math.fabs(123.456)
123.456
>>> math.fabs(-123.456)
123.456
>>>
>>> math.sqrt(49)
7.0
>>> math.exp(1)
2.718281828459045
>>> math.pow(2,10)
1024.0
>>> math.log(27)
3.295836866004329
>>> math.log10(1000)
3.0
>>>
```

MATH

math.sin(x): x'in sinüs'ünü döndürür (x: radyan)

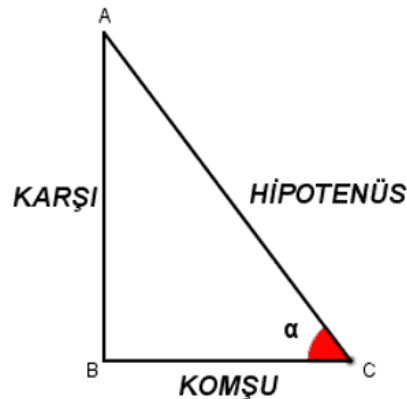
math.cos(): x'in kosinüs'ünü döndürür (x: radyan)

math.tan(): x'in tanjant'ını döndürür (x: radyan)

math.asin(): x'in arcsinüs'ünü döndürür.

math.acos(): x'in arckosinüs'ünü döndürür.

math.atan(): x'in arctanjant'ını döndürür.



$$\sin \alpha = \frac{KARŞI}{HİPOTENÜS}$$

$$\cos \alpha = \frac{KOMŞU}{HİPOTENÜS}$$

$$\tan \alpha = \frac{KARŞI}{KOMŞU}$$

$$\cot \alpha = \frac{KOMŞU}{KARŞI}$$

```
>>> import math
>>>
>>> math.sin(math.pi/2)
1.0
>>> math.sin(math.pi/4)
0.7071067811865476
>>> math.cos(0)
1.0
>>> math.cos(math.pi/4)
0.7071067811865476
>>> math.tan(math.pi/4)
0.9999999999999999
>>> math.asin(3/5)
0.6435011087932844
>>> math.acos(4/5)
0.6435011087932843
>>> math.atan(3/4)
0.6435011087932844
>>> |
```

MATH

math.trunc(x): x sayısının tam kısmını döndürür

math.floor(x): Verilen x sayısını aşağı yuvarlar.

math.ceil(x): Verilen x sayısını yukarı yuvarlar.

math.degrees(r): r radyan açısını dereceye çevirir.

math.radians(d): d derece açısını radyana çevirir.

math.factorial(x): x'in faktöriyelini alır (x!)

```
>>> import math
>>>
>>> math.trunc(123.456)
123
>>> math.floor(5.6)
5
>>> math.ceil(5.2)
6
>>> math.degrees(math.pi)
180.0
>>> math.degrees(math.pi/2)
90.0
>>> math.radians(180)
3.141592653589793
>>> math.radians(90)
1.5707963267948966
>>> math.factorial(5)
120
>>>
```

DATETIME

Tarih ve Saat ile ilgili işlemleri kolaylaştıran fonksiyonlara **datetime** modülü kullanılarak erişilir.

```
import datetime
```

datetime.date: tarihle ilgili işlemler yapan metotlar içeren nesnedir.

datetime.time: Zamanla ilgili işlemler yapan metotlar içeren nesnedir

datetime.datetime: Hem Tarih hem zaman işlemlerini birlikte yapan metotları içerir.

datetime.timedelta: İki tarih ve saat arasındaki zaman farkını döndürür.

DATETIME

dt=datetime.now(): Şimdiki zamanı döndürür.

dt.year: Yıl bilgisini döndürür

dt.month: Ay bilgisini döndürür

dt.day: Gün bilgisini döndürür

dt.hour: Saat bilgisini döndürür

dt.minute: Dakika bilgisini döndürür

dt.second: Saniye bilgisini döndürür

```
>>> from datetime import datetime
>>>
>>> simdi=datetime.now()
>>> print(simdi)
2024-08-29 19:30:35.692927
>>> print(simdi.year)
2024
>>> print(simdi.month)
8
>>> print(simdi.day)
29
>>> print(simdi.hour)
19
>>> print(simdi.minute)
30
>>> print(simdi.second)
35
>>>
```

DATETIME

Tarih ve saati belirterek kendimiz bir datetime nesnesi oluşturmak istersek;

dt=datetime(2024,12,30)

Ya da

dt=datetime(2024,12,30,23,59,59)

Şeklinde Yıl, Ay, Gün ya da Yıl, Ay, Gün, Saat, Dakika, Saniye bilgilerini girerek oluşturabiliriz.

```
>>> from datetime import datetime
>>> dt=datetime(2024,12,30)
>>> print(dt)
2024-12-30 00:00:00
>>> dt=datetime(2024,12,30,23,59,59)
>>> print(dt)
2024-12-30 23:59:59
>>> dt.year
2024
>>> dt.month
12
>>> dt.day
30
>>> dt.hour
23
>>> dt.minute
59
>>> dt.second
59
>>>
```

DATETIME – strftime

Eğer Tarih ya da saati, varsayılan formatta değil de kendi istediğimiz gibi yazdırmak istersek;

datetime.strftime(dt,"...")

Metodunu kullanabiliriz. Bu metot, tarih ve saati verilen biçimlendirme (format) **string** ifadesine göre oluşturarak bir **string** olarak döndürür.

Format ifadesinde kullanılabilecek tüm kodlar bu sununun sonunda topluca verilmiştir.

```
>>> from datetime import datetime
>>> 
>>> simdi=datetime.now()
>>> datetime.strftime(simdi,"%H:%M")
'20:10'
>>> datetime.strftime(simdi,"%d.%m.%Y")
'29.08.2024'
>>> datetime.strftime(simdi,"%d-%m-%Y %H:%M:%S")
'29-08-2024 20:10:33'
>>> datetime.strftime(simdi,"%m %B %Y %H:%M")
'08 August 2024 20:10'
>>> datetime.strftime(simdi,"%Y %B %m %A")
'2024 August 08 Thursday'
>>> datetime.strftime(simdi,"Tarih: %m %B %Y %A Saat %H:%M")
'Tarih: 08 August 2024 Thursday Saat 20:10'
>>> datetime.strftime(simdi,"%d%m%Y%H%M%S")
'29082024201033'
>>> |
```

DATETIME – timestamp

Datetime.timestamp(dt): Verilen tarih ve saate kadar, 01.01.1970 tarihinden itibaren, geçen toplam saniyeyi döndürür. Dönen değer bir tamsayıdır.

Bu yöntem, tarih ve saat işlemlerinde (iki tarih ve saat arasındaki zaman farkını bulma gibi) kolaylık sağlar.

Bu nedenle 01.01.1970 tarihi, programcıların milat tarihidir 😊

```
>>> from datetime import datetime
>>> 
>>> simdi=datetime.now()
>>> print(simdi)
2024-08-29 20:24:11.625984
>>> datetime.timestamp(simdi)
1724952251.625984
>>> dt=datetime(1971,1,1)
>>> datetime.timestamp(dt)
31525200.0
>>> 365*24*60*60
31536000
```

DATETIME - timedelta

Tarihler üzerinde işlem yaparken, örneğin, iki tarih ya da saat arasındaki farkı alırsak ya da bir tarihin üzerine belirli bir zaman eklemek istersek, dönen değer ya da eklemek istediğimiz değer bir zaman farkı (**timedelta**) olacaktır.

timedelta

Bir özel bir zamanı değil, süre, geçen zaman miktarını belirtir. Örneğin, **timedelta** saatinin 22 olması saatin 22 olduğu anlamına gelmez, 22 saatlik bir süreyi ya da zaman farkını ifade eder.

```
>>> from datetime import datetime
>>> from datetime import timedelta
>>>
>>> dt=datetime(2020,2,1)
>>> td=timedelta(days=60)
>>> print(dt+td) # 60 gun sonrasi
2020-04-01 00:00:00
>>> print(dt-td) # 60 gun oncesi
2019-12-03 00:00:00
>>>
>>> dt=datetime(2020,1,1,23,30,0)
>>> td=timedelta(hours=3)
>>> print(dt+td) # 3 saat sonrasi
2020-01-02 02:30:00
>>> print(dt-td) # 3 saat oncesi
2020-01-01 20:30:00
>>>
>>> dogum_tarihi=datetime(1990,5,25)
>>> simdiki_tarih=datetime.now()
>>> gecen_sure=simdiki_tarih-dogum_tarihi
>>> gecen_sure.days
12515
>>> gecen_sure.days/365
34.28767123287671
>>> |
```

DATE TIME – strptime

Eğer tarih ve saat bilgisini string bir kaynaktan elde ediyorsak (örneğin bir metin dosyasından okuyorsak);

dt=datetime.strptime(zaman,format)

Metodu ile tarih ve saat bilgisini çözümleyerek (parse) bir datetime nesnesi döndürebiliriz.

Zaman olarak bir string içinde tarih ve saat bilgisi olmalıdır.

Format olarak, bu sunu sonunda verilen tarih ve saat format kodları kullanılır.

Okunacak veride sabit bilgiler varsa, aynen girilir.

```
>>> from datetime import datetime
>>> 
>>> tarih_saat='2014.10.23'
>>> dt=datetime.strptime(tarih_saat,"%Y.%m.%d")
>>> print(dt)
2014-10-23 00:00:00
>>> 
>>> tarih_saat='10-23-2024 23:59'
>>> dt=datetime.strptime(tarih_saat,"%m-%d-%Y %H:%M")
>>> print(dt)
2024-10-23 23:59:00
>>> 
>>> tarih_saat='Tarih:23/10/2024 Saat:23:59:59'
>>> dt=datetime.strptime(tarih_saat,"Tarih:%d/%m/%Y Saat:%H:%M:%S")
>>> print(dt)
2024-10-23 23:59:59
>>>
```

DATE TIME – FORMAT KARAKTERLERİ

Format	Açıklama	Örnek
%a	Kısaltılmış gün ismi	Çar
%A	Gün ismi	Çarşamba
%w	Sayı olarak haftanın kaçınıcı günü (Pazar 0, Cumartesi 6)	3
%d	Sayı olarak ayın kaçınıcı günü	30
%b	Kısaltılmış ay ismi	Haz
%B	Ay ismi	Haziran
%m	Sayı olarak ay (tek haneli ise başına 0 eklenerek)	06
%y	Yıl – son iki rakam olarak	20
%Y	Yıl	2020
%H	Saat (24-Saatlik format) (0 eklenmiş sayı)	07
%I	Saat (12-Saatlik format) (0 eklenmiş sayı)	07
%p	AM / ÖÖ – PM / ÖS bilgisi	AM

Format	Açıklama	Örnek
%M	Dakika – 0 eklenmiş sayı	06
%S	Saniye – 0 eklenmiş sayı	05
%f	Mikrosaniye – 0 eklenmiş sayı	000000
%j	Yılın kaçınıcı günü	273
%U	Yılın kaçınıcı haftası (Pazar gününden başlayarak)	39
%W	Yılın kaçınıcı haftası (Pazartesi gününden başlayarak)	39
%c	Yerel biçim ayarlarına göre tarih ve saat	Çar Haz 10 07:06:05 2020
%x	Yerel biçim ayarlarına göre tarih	06/10/20
%X	Yerel biçim ayarlarına göre tarih ve saat	07:06:05
%%	Karakter girdisi	%