

ماژول‌های ریاضی و تصادفی

پایتون همراه با ماژول ریاضی و ماژول تصادفی داخلی می‌آید. در این سخنرانی، ما به طور خلاصه به قابلیت‌های آن‌ها می‌پردازیم. معمولاً شما می‌توانید به سادگی تماس تابعی که به دنبال آن هستید را در مستندات آنلاین جستجو کنید.

- [ماژول ریاضی](https://docs.python.org/3/library/math.html) (<https://docs.python.org/3/library/math.html>)
- [ماژول تصادفی](https://docs.python.org/3/library/random.html) (<https://docs.python.org/3/library/random.html>)

ما به تمام توابع موجود در این ماژول‌ها نمی‌پردازیم زیرا بسیار زیاد هستند، اما برخی از توابع مفید را نشان خواهیم داد.

توابع ریاضی مفید

In [4]:

```
1 import math
```

In [5]:

```
1 help(math)
```

in the case where x is exactly halfway between two multiples of y, the nearest even value of n is used. The result is always exact.

```
sin(x, /)
    Return the sine of x (measured in radians).
```

```
sinh(x, /)
    Return the hyperbolic sine of x.
```

```
sqrt(x, /)
    Return the square root of x.
```

```
tan(x, /)
    Return the tangent of x (measured in radians).
```

```
tanh(x, /)
    Return the hyperbolic tangent of x.
```

```
trunc(x, /)
    Truncates the Real x to the nearest Integral toward 0.
```

گرد کردن اعداد

In [6]:

```
1 value = 4.35
```

In [7]:

```
1 math.floor(value)
```

Out[7]:

4

In [8]:

```
1 math.ceil(value)
```

Out[8]:

5

In [11]:

```
1 round(4.35, 5)
```

Out[11]:

4.35

ثابت های ریاضی

In [12]:

```
1 math.pi
```

Out[12]:

3.141592653589793

In [13]:

```
1 from math import pi
```

In [14]:

```
1 pi
```

Out[14]:

3.141592653589793

In [15]:

```
1 math.e
```

Out[15]:

2.718281828459045

In [16]:

```
1 math.tau
```

Out[16]:

6.283185307179586

In [17]:

```
1 math.inf
```

Out[17]:

inf

In [18]:

```
1 math.nan
```

Out[18]:

nan

مقادیر لگاریتمی

In [19]:

```
1 math.e
```

Out[19]:

2.718281828459045

In [20]:

```
1 # Log Base e  
2 math.log(math.e)
```

Out[20]:

1.0

In [21]:

```
1 # Will produce an error if value does not exist mathmatically
2 math.log(0)
```

-
ValueError Traceback (most recent call last)

Cell In[21], line 2

```
1 # Will produce an error if value does not exist mathmatically
----> 2 math.log(0)
```

ValueError: math domain error

In [22]:

```
1 math.log(10)
```

Out[22]:

2.302585092994046

In [23]:

```
1 math.e ** 2.302585092994046
```

Out[23]:

10.000000000000002

In [24]:

```
1 round(math.e ** 2.302585092994046)
```

Out[24]:

10

لگاریتم با پایه دلخواه

In [25]:

```
1 # math.Log(x,base)
2 math.log(100, 10)
```

Out[25]:

2.0

In [26]:

```
1 10**2
```

Out[26]:

100

توابع مثلثاتی

In [27]:

```
1 # Radians
2 math.sin(10)
```

Out[27]:

-0.54402111108893698

In [28]:

```
1 math.degrees(pi/2)
```

Out[28]:

90.0

In [29]:

```
1 math.radians(180)
```

Out[29]:

3.141592653589793

ماژول تصادفی

ماژول تصادفی به ما امکان ایجاد اعداد تصادفی می‌دهد. حتی می‌توانیم یک بذر (seed) تنظیم کنیم تا هر بار مجموعه‌ی تصادفی مشابهی تولید شود.

توضیح در مورد اینکه چگونه یک رایانه تلاش می‌کند برای تولید اعداد تصادفی، خارج از دامنه‌ی این دوره است زیرا شامل ریاضیات سطح بالاتری می‌شود. اما اگر به این موضوع علاقه‌مند هستید، به لینک‌های زیر مراجعه کنید:

- https://en.wikipedia.org/wiki/Pseudorandom_number_generator
- https://en.wikipedia.org/wiki/Pseudorandom_number_generator
- https://en.wikipedia.org/wiki/Random_seed (https://en.wikipedia.org/wiki/Random_seed)

درک بذر (seed)

In [30]:

```
1 import random
```

In [34]:

```
1 random.randint(0,100)
```

Out[34]:

39

In [44]:

```
1 random.randint(0,100)
```

Out[44]:

96

In [49]:

```
1 # The value 101 is completely arbitrary, you can pass in any number you want
2 random.seed(101)
3 # You can run this cell as many times as you want, it will always return the same nu
4 random.randint(0,100)
```

Out[49]:

74

In [51]:

```
1 random.randint(0,100)
```

Out[51]:

69

In [53]:

```
1 # The value 101 is completely arbitrary, you can pass in any number you want
2 random.seed(101)
3 print(random.randint(0,100))
4 print(random.randint(0,100))
5 print(random.randint(0,100))
6 print(random.randint(0,100))
7 print(random.randint(0,100))
```

74

24

69

45

59

اعداد تصادفی صحیح

In [54]:

```
1 random.randint(0,100)
```

Out[54]:

6

انتخاب تصادفی

انتخاب تصادفی از میان اعضای یک لیست

In [55]:

```
1 mylist = list(range(0,20))
```

In [56]:

```
1 mylist
```

Out[56]:

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]

In [59]:

```
1 random.choice(mylist)
```

Out[59]:

19

In [60]:

```
1 mylist
```

Out[60]:

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]

نمونه‌برداری با جایگزینی

در این حالت، تعداد نمونه‌ها را انتخاب می‌کنیم و انتخاب عناصر به صورت چندین بار تکراری مجاز است. تصور کنید که یک کیسه‌ی توپ‌های شماره‌دار لاتاری دارید. شما دست خود را داخل کیسه می‌گذارید و یک توپ لاتاری به صورت تصادفی انتخاب می‌کنید، سپس پس از نوشتن شماره، آن را دوباره درون کیسه قرار می‌دهید و سپس ادامه می‌دهید و توپ دیگری را انتخاب می‌کنید.

In [62]:

```
1 random.choices(population=mylist,k=10)
```

Out[62]:

```
[9, 6, 12, 8, 4, 8, 6, 2, 16, 13]
```

نمونه برداری بدون جایگزینی

یکباری که یک عنصر به صورت تصادفی انتخاب شود، دیگر نمی‌توان آن را انتخاب کرد. تصور کنید که یک کیسه‌ی توپ‌های شماره‌دار لاتاری دارید. شما دست خود را داخل کیسه می‌گذارید و یک توپ لاتاری به صورت تصادفی انتخاب می‌کنید، سپس پس از نوشتن شماره، آن را از کیسه خارج می‌کنید و سپس ادامه می‌دهید و توپ دیگری را انتخاب می‌کنید.

In [63]:

```
1 random.sample(population=mylist,k=10)
```

Out[63]:

```
[8, 12, 18, 7, 13, 1, 6, 11, 10, 15]
```

برهم زدن ترتیب اعضای لیست

توجه: این دستور لیست اصلی را تغییر خواهد داد!

In [64]:

```
1 # Don't assign this to anything!  
2 random.shuffle(mylist)
```

In [65]:

```
1 mylist
```

Out[65]:

```
[8, 10, 17, 7, 1, 16, 9, 2, 4, 13, 18, 15, 5, 19, 12, 11, 3, 6, 14, 0]
```

توزیع های تصادی

توزیع یکنواخت (https://en.wikipedia.org/wiki/Uniform_distribution)

In [84]:

```
1 # Continuous, random picks a value between a and b, each value has equal change of b
2 random.uniform(a=0,b=100)
```

Out[84]:

81.72235646759873

[توزیع نرمال یا گوسین \(https://en.wikipedia.org/wiki/Normal_distribution\)](https://en.wikipedia.org/wiki/Normal_distribution)

In [95]:

```
1 random.gauss(mu=0,sigma=1)
```

Out[95]:

2.6001538661204027

یادداشت نهایی: اگر متوجه شدید که از این کتابخانه‌ها بسیار استفاده می‌کنید، به کتابخانه NumPy برای پایتون نگاهی بیندازید که تمامی این قابلیت‌ها را با کارایی بسیار بالا پوشش می‌دهد. ما در دوره‌های علم داده و یادگیری ماشین به این کتابخانه و بسیاری دیگر می‌پردازیم.