



دوره جامع پایتون:
بخش تسلط بر کدنویسی به زبان پایتون
جلسه هفتم

دکتر ذبیح اله ذبیحی

کد مورس Morse code

- **کد مورس** روشی برای انتقال پیام و اطلاعات است که در آن از یک رشته نشانه‌های بلند و کوتاه استاندارد به نام خط و نقطه استفاده می‌شود؛ و همچنین در تلگراف نیز بکار گرفته می‌شود.

حرف	کد	حرف	کد
A	• —	N	— •
B	— • • •	O	— — — —
C	— • — •	P	• — — •
D	— • •	Q	— — • —
E	•	R	• — •
F	• • — •	S	• • •
G	— — •	T	—
H	• • • •	U	• • —
I	• •	V	• • • —
J	• — — —	W	• — —
K	— • —	X	— • • —
L	• — • •	Y	— • — —
M	— —	Z	— — • •

عدد	کد
۰	— — — — —
۱	• — — — —
۲	• • — — —
۳	• • • — —
۴	• • • • —
۵	• • • • •
۶	— • • • •
۷	— — • • •
۸	— — — • •
۹	— — — — •

نقطه‌گذاری و علامات ویژه	کد
.	• — • — • —
,	— — • • — —
?	• • — — • •
-	— • • • • —
/	— • • — •
اشتباه	• • • • • •
علامت آغاز	• — • — • —
پایان پیام +	• — • — •
پایان تماس @	• • • — • —
(علامت اضطراری) (SOS) اس. او. اس	• • • — — — • • •

مثال تبدیل متن به کد مورس

```
message="HELLO. GO HOME"
MORSE_CODE_DICT = { 'A':'.-.', 'B':'-...', 'C':'-.-.', 'D':'-..', 'E':'.', 'F': '..-.', 'G':'--.', 'H': '....',
                    'I': '...', 'J': '.---', 'K': '-.-', 'L': '.-..', 'M': '--', 'N': '-.',
                    'O': '---', 'P': '.---.', 'Q': '--.-', 'R': '.-.', 'S': '...', 'T': '-.',
                    'U': '..-', 'V': '...-', 'W': '-.-', 'X': '-...', 'Y': '-.-.', 'Z': '--..',
                    '1': '.----', '2': '..---', '3': '...--', '4': '....-', '5': '.....', '6': '-....',
                    '7': '--...', '8': '---..', '9': '----.', '0': '-----', ',', ':': '--...-', '!': '.-.-.-',
                    '?': '..--..', '/': '-.-.-', '-': '-....-', '(': '-.-.-', ')': '-.-.-'}

cipher = ""
for letter in message:
    if letter != ' ':
        cipher=cipher + MORSE_CODE_DICT[letter] + ' '
    else:
        cipher=cipher+ ' '
print(cipher)
```

مثال تبدیل کد مورس به متن

```
message=".... . .-.. -.--- .--.- -.-. --- .... --- -- .""
message=message+ ' '
decipher = ""
citext = ""
for letter in message:
    if (letter != ' '):
        i = 0
        citext=citext+ letter
    else:
        i=i+1
        if i == 2 :
            decipher=decipher+' '
        else:
            decipher=decipher+ list(MORSE_CODE_DICT.keys())[list(MORSE_CODE_DICT.values()).index(citext)]
        citext = ""
print(decipher)
```

ماژول random

- ماژول random را میتوان از آن در جهت تولید اعداد تصادفی، انتخاب کارکترهای تصادفی و... استفاده نمود. این ماژول بیشتر در برنامه های مربوط به تولید رشته و عدد تصادفی مانند ساخت پسورد ها، شبیه سازی های مونت کارلو و ... مورد استفاده قرار میگیرد.

```
import random
```

تولید عدد رندوم بین صفر و یک

```
import random  
print(random.random())
```

مثال

```
import random  
for i in range(10):  
    print(random.random())
```


تولید عدد رندوم در بازه دلخواه با دستور uniform

```
import random
```

```
for i in range(10):
```

```
    print(random.uniform(10,20))
```

تولید عدد رندوم صحیح

```
import random  
print(random.randint(20,110))
```

مثال

```
import random  
for i in range(10):  
    x=random.randint(0,100)  
    print(x)
```

تابع random.randrange

- با استفاده از این تابع میتوان یک عدد تصادفی بر اساس مقدار شروع و مقدار پایان و همچنین بر اساس گام ایجاد کنیم .

- `random.randrange(start, stop , step)`

```
for i in range(4):
```

```
    x=random.randrange(10, 100 , 5)
```

```
    print(x)
```

مثال: انتخاب رندوم از یک لیست

```
import random  
x=["ali","reza","neda","zahra"]  
n=len(x)  
r=random.randint(0,n-1)  
print(x[r])
```

انتخاب تصادفی از یک لیست random.choice

```
import random  
my_list=['ali','reza','neda','zahra']  
x=random.choice(my_list)  
print(x)
```

مثال: قرعه کشی بین خریداران یک شرکت، از هر جایزه به
تعداد نامحدود وجود دارد

```
import random
```

```
list1=['ali','reza','neda','zahra']
```

```
list2=["nothing","100$","10$","House","BMW"]
```

```
for i in list1:
```

```
    x=random.choice(list2)
```

```
    print(i,x)
```

مثال: قرعه کشی بین خریداران یک شرکت، از هر جایزه به
تعداد محدود وجود دارد

```
import random
list1=["ali","reza","omid","neda","zahra"]
list2=["100$","200$","1000$","nothing","BMW"]
for i in list1:
    x=random.choice(list2)
    if x != "nothing":
        list2.remove(x)
    print(i,x)
```


مثال: انتخاب رندوم از رشته و تاپل و دیکشنری

```
x="zabiholah"
```

```
y=random.choice(x)
```

```
print(y)
```

```
import random
```

```
x=("zabiholah","omid","sasan")
```

```
y=random.choice(x)
```

```
print(y)
```

```
import random
```

```
x={32:["zabiholah","omid"],25:["sasan","neda","zahra"]}
```

```
y=random.choice(x[25])
```

```
print(y)
```

مثال: شبیه سازی انداختن تاس

```
import random
dice=[1,2,3,4,5,6]
s=0
n=int(input("n="))
for i in range(n):
    x=random.choice(dice)
    s=s+1
    print(x)
print("-----")
print("s=",s)
```

مثال: انتخاب رندوم یک کاراکتر از رشته

```
import random  
name = 'python'  
char = random.choice(name)  
print("random char is ", char)
```

بر زدن یک لیست با دستور `shuffle()`

- از تصادفی بودن می‌توان برای بر زدن تصادفی یک لیست از عناصر مانند بر زدن ورق‌های بازی استفاده کرد. از تابع `shuffle()` می‌توان برای انجام این کار، استفاده کرد. `shuffle` داده‌ها را در جای خودشان بر می‌زند، بدین معنا که لیست به عنوان یک آرگومان برای تابع `shuffle()` ارائه، بر زده و باز گردانده می‌شود. در حقیقت این تابع لیست بر زده شده را به جای کپی در یک لیست جدید، به شکل درجا ایجاد می‌کند.

مثال

```
import random  
my_list=['ali','reza','neda','zahra']  
random.shuffle(my_list)  
print(my_list)
```

شکل دیگر

```
from random import shuffle  
my_list=['ali','reza','neda','zahra']  
shuffle(my_list)  
print(my_list)
```

مثال:مثال: قرعه کشی بین خریداران یک شرکت، از هر جایزه به تعداد محدود وجود دارد

```
import random
list1=['ali','reza','neda','zahra']
list2=["nothing","100$","10$","nothing","House","BMW","nothing","nothing"]
random.shuffle(list1)
random.shuffle(list2)
j=0
for i in list1:
    print(i,list2[j])
    j=j+1
```

نکته

- تابع `shuffle()` قابل اعمال بر روی رشته و تاپل نیست.

کنترل تعداد رقم اعشار

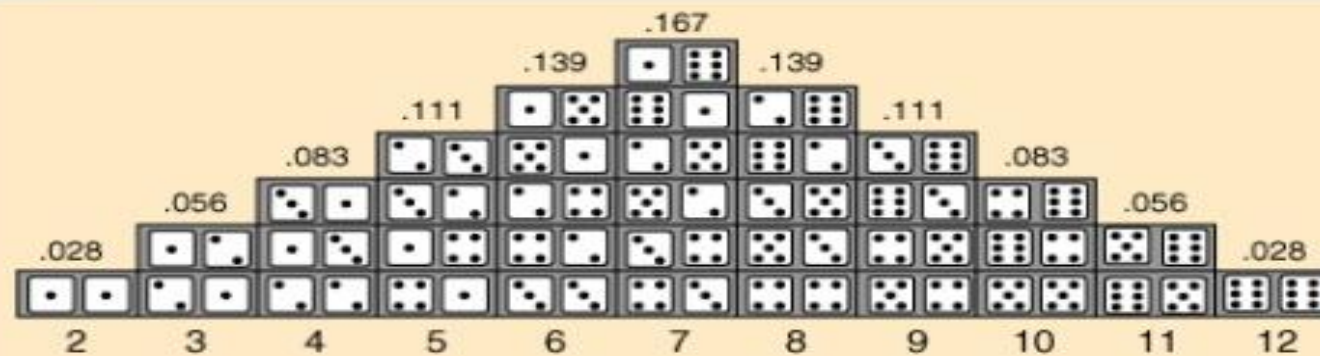
`round(3.677333,2)=3.68`

`round(3.677333,4)=3.6773`

کدی بنویسید که در آن دو تاس انداخته و احتمال اینکه مجموع دو تاس 2 تا 12 شود را تعیین کند.

EXAMPLE: ROLLING DICE

- As a simple example of a Monte Carlo simulation, consider calculating the probability of a particular sum of the throw of two dice (with each die having values one through six). In this particular case, there are 36 combinations of dice rolls:



```
import random
n=10000
n2=0
n3=0
n4=0
n5=0
n6=0
n7=0
n8=0
n9=0
n10=0
n11=0
n12=0
mylist=[1,2,3,4,5,6]
```

```
for i in range(n):  
    d1=random.choice(mylist)  
    d2=random.choice(mylist)  
  
    s=d1+d2  
  
    if s==2:  
        n2=n2+1  
  
    elif s==3:  
        n3=n3+1  
  
    elif s==3:  
        n3=n3+1  
  
    elif s==4:  
        n4=n4+1  
  
    elif s==5:  
        n5=n5+1  
  
    elif s==6:  
        n6=n6+1  
  
    elif s==7:  
        n7=n7+1  
  
    elif s==8:  
        n8=n8+1  
  
    elif s==9:  
        n9=n9+1  
  
    elif s==10:  
        n10=n10+1  
  
    elif s==11:  
        n11=n11+1  
  
    elif s==12:  
        n12=n12+1
```

$p2=n2/n$

$p3=n3/n$

$p4=n4/n$

$p5=n5/n$

$p6=n6/n$

$p7=n7/n$

$p8=n8/n$

$p9=n9/n$

$p10=n10/n$

$p11=n11/n$

$p12=n12/n$

`print("2:",p2)`

`print("3:",p3)`

`print("4:",p4)`

`print("5:",p5)`

`print("6:",p6)`

`print("7:",p7)`

`print("8:",p8)`

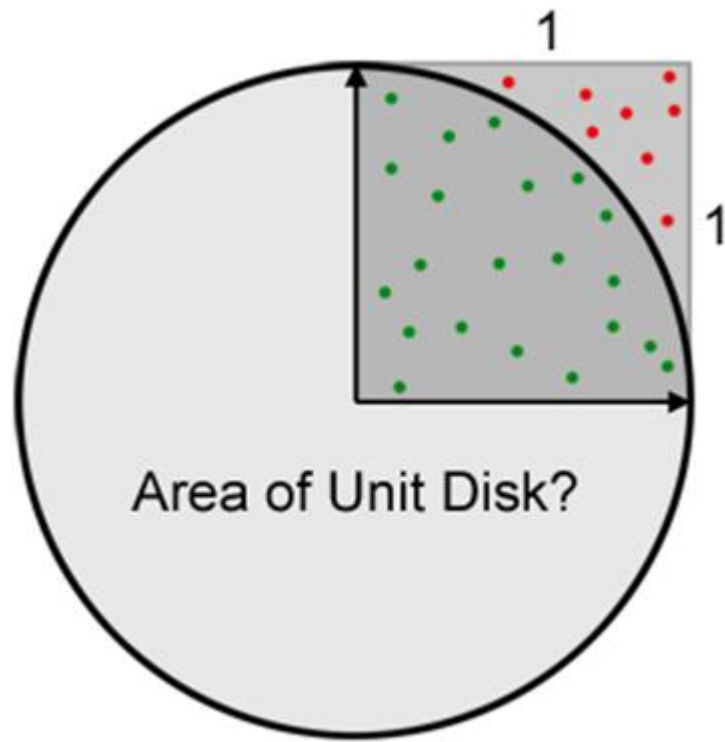
`print("9:",p9)`

`print("10:",p10)`

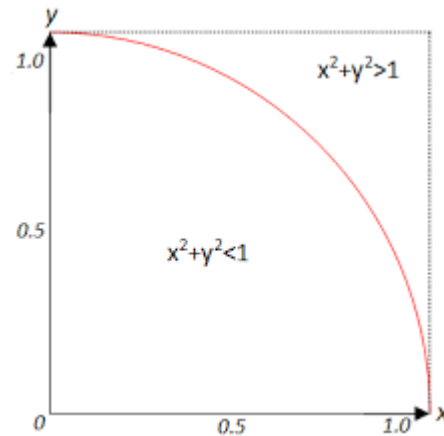
`print("11:",p11)`

`print("12:",p12)`

مثال: محاسبه عدد π با استفاده از اعداد رندوم (مونت کارلو)



$$\pi \approx 4 \frac{\text{NUMBER OF POINTS INSIDE THE CIRCLE}}{\text{NUMBER OF ALL POINTS}}$$



```
import random
import math
n=int(input("n="))
n0=0
for i in range(n):

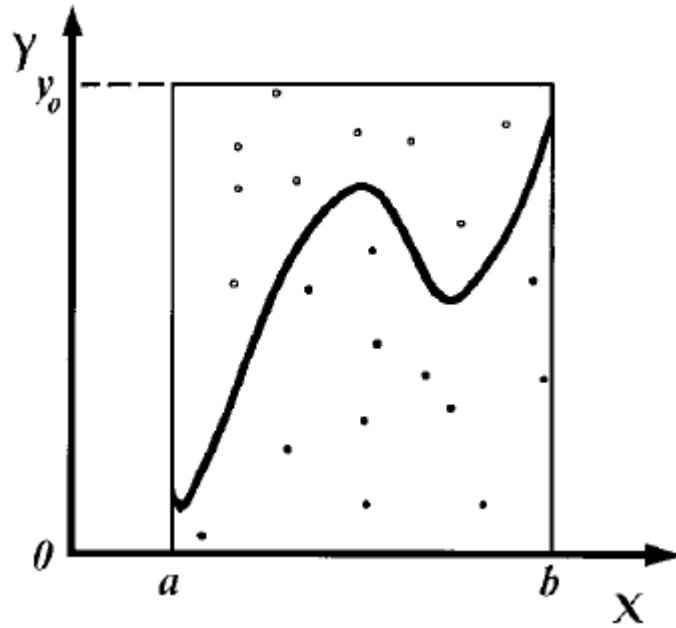
    x=random.random()
    y=random.random()
    r2=x**2+y**2
    r=math.sqrt(r2)
    if r<=1:
        n0=n0+1
pi=(4*n0)/n
print(pi)
```

```
import random
import math
n=int(input("n="))

m=10
s=0
for j in range(m):
    n0=0
    for i in range(n):

        x=random.random()
        y=random.random()
        r2=x**2+y**2
        r=math.sqrt(r2)
        if r<=1:
            n0=n0+1
    pi=(4*n0)/n
    s=s+pi
    print("pi=",pi)
pi_mean=pi/m
print("pi_mean=",pi_mean)
```

انتگرال گیری عددی



$$I = \int_a^b f(x) dx = (b-a) f_{average} = (b-a) \langle f \rangle$$

$$\langle f \rangle_N = \frac{1}{N} \sum_{i=1}^N f(x_i)$$

$$\int_a^b f(x) dx = \frac{(b-a)}{N} \sum_{i=1}^N f(x_i)$$

به منظور تغییر بازه تغییرات از $[\min, \max]$ به $[a, b]$

$$f(x) = \frac{(b-a)(x - \min)}{\max - \min} + a$$

بنابراین برای تغییر بازه اعداد رندوم از $[0, 1]$ به $[a, b]$

$$F(x) = (b-a)x + a$$

مثال

```
import random
def f(x):
    return x
a=float(input("a="))
b=float(input("b="))
n=int(input("n="))
s=0
for j in range(n):
    x=random.random()
    x1=(b-a)*x+a
    s=s+f(x1)
l=((b-a)*s)/n

print(l)
```

```
import random
def f(x):
    return x
a=float(input("a="))
b=float(input("b="))
n=int(input("n="))
s=0
for j in range(n):
    x=random.uniform(a,b)
    print(x)
    s=s+f(x)
l=((b-a)*s)/n

print(l)
```

مدیریت هزینه/زمان

	Best-case	expected	Worst-case	MC
p	80	100	120	
w	375	500	625	
u	750	800	850	
total	1250	1400	1595	

```
import random  
n=int(input("n="))
```

```
p1=float(input("p1="))  
p2=float(input("p2="))
```

```
w1=float(input("w1="))  
w2=float(input("w2="))
```

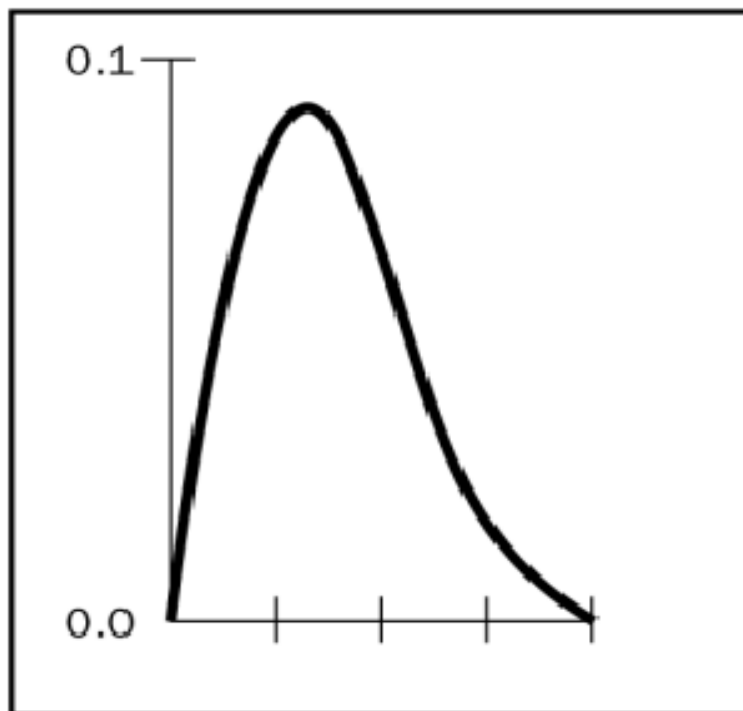
```
u1=float(input("u1="))  
u2=float(input("u2="))
```

```
sp=0  
sw=0  
su=0
```

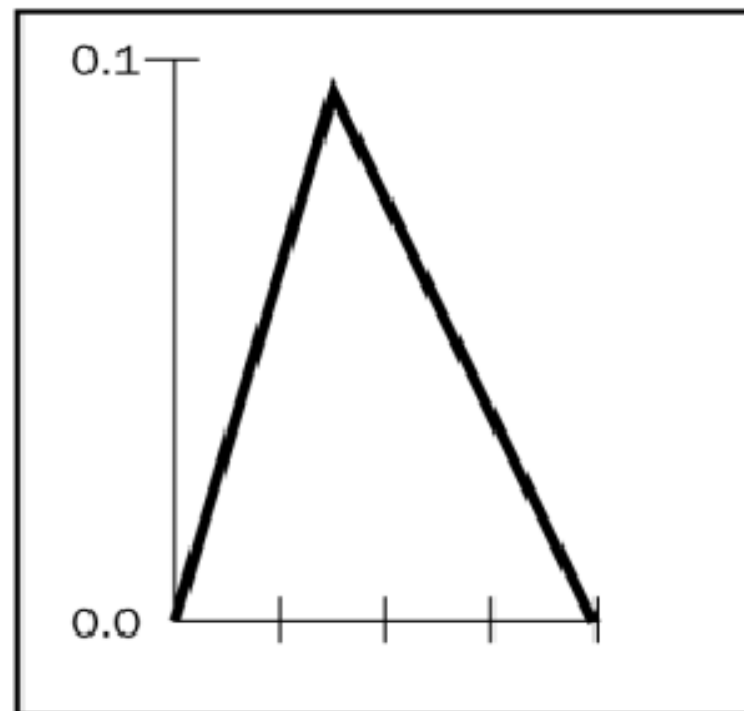
```
for j in range(n):  
    px=random.random()  
    px1=(p2-p1)*px+p1  
    sp=sp+px1  
  
    wx=random.random()  
    wx1=(w2-w1)*wx+w1  
    sw=sw+wx1  
  
    ux=random.random()  
    ux1=(u2-u1)*ux+u1  
    su=su+ux1  
  
lp=sp/n  
lw=sw/n  
lu=su/n  
ll=(lp+lw+lu)  
print(lp,lw,lu,ll)
```

توزیع های احتمالی

Beta Distribution



Triangular Distribution



تولید عدد رندوم با توزیع آماری مختلف

- `random.uniform(a, b)`
- `random.triangular(low, high, mode)`
- `random.betavariate(alpha, beta)`
- `random.expovariate(lambd)`
- `random.gammavariate(alpha, beta)`
- `random.gauss(mu, sigma)`
- `random.lognormvariate(mu, sigma)`
- `random.normalvariate(mu, sigma)`

- `random.vonmisesvariate(mu, kappa)`
- `random.paretovariate(alpha)`
- `random.weibullvariate(alpha, beta)`

تمرین

- کدی بنویسید که عبارت رمز شده با تکنیک مورس را رمز گشایی کنید.

تمرین:

انجام پروژه ای دارای سه فاز می باشد. با توجه به بهترین و بدترین مدت زمان انجام هر فاز ، کد بنویسید که تخمینی از مدت زمان انجام پروژه را به ازای تابع توزیع گوسی ارائه کنید

فاز	در بهترین حالت	در بدترین حالت		
اول	۷	۱۵		
دوم	۲۰	۴۸		
سوم	۲۳	۶۰		
کل	۵۰	۱۲۳		