

要求：在python内练习此模块并截图。

```
import random
random.random()           #生成0-1之间均匀分布的随机浮点数
random.normalvariate(0, 1) #生成1个符合均值为0，方差为1正态分布的随机数
[random.normalvariate(0, 1) for x in range(10)] #生成长度为10的正态分布序列
random.uniform(a, b)      #生成[a, b]区间内的随机浮点数
random.randint(a, b)      #生成[a, b]区间内的随机整数
random.choice(s)          #从序列s中随机获取一个值
random.shuffle(s)         #将序列s中元素打乱
random.sample(s, k)       #从序列s中获取长度为k的片段
```

In [47]:

```
import random
a = 1.5
b = 3.0
c = 10
d = 20
s = [12, 27, 29, 32, 35, 54, 60, 70, 71, 91]
k = 4
x = 0
print(type(random.random()))
print('生成0-1之间均匀分布的随机浮点数:', random.random())
print('生成1个符合均值为0，方差为1正态分布的随机数:', random.normalvariate(0, 1))
print('生成长度为10的正态分布序列', [random.normalvariate(0, 1) for x in range(10)])
print('生成[a, b]区间内的随机浮点数:', random.uniform(a, b))
print('生成[a, b]区间内的随机整数:', random.randint(c, d) )
print('从序列s中随机获取一个值:', random.choice(s))
print('将序列s中元素打乱:', random.shuffle(s))
print('从序列s中获取长度为k的片段:', random.sample(s, k))
```

<class 'builtin_function_or_method'>

生成0-1之间均匀分布的随机浮点数: 0.2730423212252896

生成1个符合均值为0，方差为1正态分布的随机数: 0.6408167477563698

生成长度为10的正态分布序列 [-0.7116373653441779, 0.3678671318832897, 0.11999051486279831, -0.33912290344278306, -0.39527100817378585, 0.7254693538141064, 0.4935706515994945, 0.48479330918014985, 0.3083243742452991, 1.4757085182052603]

生成[a, b]区间内的随机浮点数: 2.424417967592914

生成[a, b]区间内的随机整数: 11

从序列s中随机获取一个值: 35

将序列s中元素打乱: None

从序列s中获取长度为k的片段: [60, 32, 29, 71]

In []: