NAME

random - Random variable generators.

CLASSES

\_random.**Random**(builtins.object)

Random

SystemRandom

class Random(\_random.Random)

| Random number generator base class used by bound module functions.

|

| betavariate(self, alpha, beta)

| Beta distribution.

|

| **choice(self, seq)**

| Choose **a random element** from a non-empty sequence.

|

>>> values **=** [1, 2, 3, 4, 5, 6]

>>> random**.**choice(values)

2

| expovariate(self, lambd)

| Exponential distribution.

|

| gammavariate(self, alpha, beta)

| Gamma distribution. Not the gamma function!

|

| gauss(self, mu, sigma)

| Gaussian distribution.

|

| getstate(self)

| Return internal state; can be passed to setstate() later.

|

| lognormvariate(self, mu, sigma)

| Log normal distribution.

|

| normalvariate(self, mu, sigma)

| Normal distribution.

|

| paretovariate(self, alpha)

| Pareto distribution. alpha is the shape parameter.

|

| **randint(self, a, b)**

| **Return random integer in range [a, b]**, including both end points.

|

| randrange(self, start, stop=None, step=1, \_int=<class 'int'>)

| Choose a random item from range(start, stop[, step]).

|

| This fixes the problem with randint() which includes the

| endpoint; in Python this is usually not what you want.

|

| **sample(self, population, k)**

| Chooses **k unique random elements** from a population sequence or set.

>>> random**.**sample(values, 2)

[6, 2]

|

|  **seed(self, a=None, version=2)**

| Initialize internal state from hashable object.

|

| None or no argument seeds from current time or from an operating

| system specific randomness source if available.

|

| For version 2 (the default), all of the bits are used if \*a\* is a str,

| bytes, or bytearray. For version 1, the hash() of \*a\* is used instead.

|

| If \*a\* is an int, all bits are used.

|

| setstate(self, state)

| Restore internal state from object returned by getstate().

|

| shuffle(self, x, random=None) #只是打乱顺序

| Shuffle list x in place, and return None.

|

| triangular(self, low=0.0, high=1.0, mode=None)

| Triangular distribution.

|

| Continuous distribution bounded by given lower and upper limits,

| and having a given mode value in-between.

|

| http://en.wikipedia.org/wiki/Triangular\_distribution

|

| **uniform(self, a, b)**

| **Get a random number in the range [a, b)** or [a, b] depending on rounding.

|

| vonmisesvariate(self, mu, kappa)

| Circular data distribution.

|

| mu is the mean angle, expressed in radians between 0 and 2\*pi, and

| kappa is the concentration parameter, which must be greater than or

| equal to zero. If kappa is equal to zero, this distribution reduces

| to a uniform random angle over the range 0 to 2\*pi.

|

| weibullvariate(self, alpha, beta)

| Weibull distribution.

|

| alpha is the scale parameter and beta is the shape parameter.

class SystemRandom(Random)

| Alternate random number generator using sources provided

| by the operating system (such as /dev/urandom on Unix or

| CryptGenRandom on Windows).

|

| Not available on all systems (see os.urandom() for details).

|

| Method resolution order:

| SystemRandom

| Random

| \_random.Random

| builtins.object

|

| Methods defined here:

|

| getrandbits(self, k)

| getrandbits(k) -> x. Generates an int with k random bits.

|

| getstate = \_notimplemented(self, \*args, \*\*kwds)

|

| **random(self)**

| **Get the next random number in the range [0.0, 1.0).**

|

| seed(self, \*args, \*\*kwds)

| Stub method. Not used for a system random number generator.

|

| setstate = \_notimplemented(self, \*args, \*\*kwds)

|

| ----------------------------------------------------------------------

| Methods inherited from Random:

|