# java.util.Random

The java.util.Random class allows you to create objects that produce pseudo-random numbers with uniform or gaussian distributions according to a linear congruential formula with a 48-bit seed.

The algorithm used is good enough for games. I wouldn't use it for cryptography.

You can choose the seed or you can let Java pick one based on the current time.

Random r = new Random(109876L);  
int i = r.nextInt();  
int j = r.nextInt();  
long l = r.nextLong();  
float f = r.nextFloat();  
double d = r.nextDouble();  
int k = r.nextGaussian();

The nextInt(), nextLong(), and nextBytes() methods all cover their respective ranges with equal likelihood. For example, to simulate a six-sided die; that is to generate a random integer between 1 and 6, you might write

Random r = new Random();  
int die = r.nextInt();  
die = Math.abs(die);  
die = die % 6;  
die += 1;  
System.out.println(die);

The nextGaussian() method returns a pseudo-random, Gaussian distributed, double value with mean 0.0 and standard deviation 1.0.

The nextBytes() method fills a byte[] array with random bytes. For example,

byte[] ba = new byte[1024];  
Random r = new Random();  
r.nextBytes(ba);  
for (int i = 0; i < ba.length; i++) {  
 System.out.println(ba[i]);  
}

[Previous](http://docs.google.com/53.html) | [Next](http://docs.google.com/55.html) | [Top](http://docs.google.com/index.html) | [Cafe au Lait](http://www.cafeaulait.org/)

Copyright 1997 Elliotte Rusty Harold

[elharo@metalab.unc.edu](mailto:elharo@metalab.unc.edu)

Last Modified September 18, 1997