Generating Reproducible Random Numbers

Problem

You want to generate a sequence of random numbers, but you want to reproduce the same sequence every time your program runs.

Solution

Before running your Python code, call the **random.seed** function to initialize the random number generator to a known state:

```
> random.seed(666) # Or use any other positive integer
```

Discussion

After generating random numbers, you may often want to reproduce the same sequence of "random" numbers every time your program executes. That way, you get the same results from run to run.

I once supported a complicated Monte Carlo analysis of a huge portfolio of securities. The users complained about getting a slightly different result each time the program ran. No kidding, I replied; it's all based on random numbers, so of course there is randomness in the output.

The solution was to set the random number generator to a known state at the beginning of the program. That way, it would generate the same (quasi-)random numbers each time and thus yield consistent, reproducible results.

In Python, the random.seed function sets random number generator to a known state. The function takes one argument, an integer. Any positive integer will work, but you must use the same one in order to get the same initial state.

The function returns nothing. It works behind the scenes, initializing (or reinitializing) the random number generator. The key here is that using the same seed restarts the random number generator back at the same place:

```
> random.seed(165) # Initialize the random number generator to a known state
```