Random Numbers

Random numbers are created by a random number generator and are mainly “seeded” off the computer clock.

This seed is then put through a random number generator and a random number is found but we don’t really need to know that. We just need to use the correct commands.

Try the following program:

```
import random     # required to use random numbers

for count in range(10):
    print (random.randint(1, 100))  #randint!!!!
```

The random module has the following useful functions:

randint

Usage: random.randint(start, end)

- start and end are both integers, start **must be** less than end
- randint returns an integer that can be assigned to another variable

eg x = random.randint(1, 100)

random

Usage: random.random()

- returns a real number between 0 and 1

choice

Usage: random.choice(choices)

- choices are a range of choices to be chosen from
- the choice selected will be returned

eg flip= random.choice([‘heads’, ‘tails’])

randrange

Usage: random.randrange(start, end, step)

- start and end are both integers where start < end
- step is how much the numbers count up by

eg num = random.randrange(1, 100, 2) #generate random odd integers

Exercises

Write each of the following programs:

a) Write a program that simulates tossing a coin 300 times. Output how many times a head was tossed and how many times a tail was tossed.

b) Write a program that simulates the rolling of a die 1000 times. How many times has the die been a 5?

c) Write a program that simulates the flipping of two coins 10000 times. Output how many times the coins were the same.

d) Write a program to simulate the playing of a simple dice game (played with one die). Roll the die to get a value from 1 to 6. This we will call your point. Now keep rolling until you get the same value (your point) again and see how many rolls it takes. **Program it so you can play this game repeatedly.
** Try at your own risk.

e) Write a program that simulates the dealing of one card from a deck of cards.

f) Write a program that acts like a guessing game. Your program should randomly choose a number between 1 and 100, then have the user try to guess the number in the fewest guesses. After each incorrect guess, the program should respond with either “Too high” or “Too low”.

g) Write a program that colours each 5 by 5 block of pixels of the screen with a random colour of red, green or blue. Hint: Use a nested loop (loop within a loop).

h) Modify the program above so that the next block of pixels has a 50% chance of being the same colour as the previous colour.
For example, if the last block was red, then 50% chance of being red, 25% for both green and blue.