**Random Numbers**

Given the same inputs, most computer programs generate the same outputs every time. For some applications, though, we want the computer to be unpredictable. Games are an obvious example.

The ***random*** module provides functions that generate random numbers. The ***random()*** function returns a random float between 0.0 and 1.0 (including 0.0 but not 1.0). Each time you call random, you get the next number in a long series. To see a sample, run this loop:

import random # library of random functions

for i in range(10): # repeats 10 times

x = random.random() # generates a number between 0 and 1

print (x)

A random number from 0 to 1 is not that useful. To make the range of values bigger, multiply by a factor:

import random # library of random functions

for i in range(10):

x = 5 \* random.random() # between 0 and 5

print (x)

This generates numbers between 0 and 4.9999999…

If you want numbers between 20 and 24.999999… simply add 20:

import random

print ()

print ("This prints random numbers between 20 and 25:")

for i in range(10):

x = 5 \* random.random() +20 # between 20 and 25

print (x)

This still has a range of 5, starting from 20. So this produces numbers from 20 to 25.

Try these and see what this does:

import random

print ()

print ("This prints random numbers between ?? and ??:")

for i in range(10):

x = 10 \* random.random() + 50 # between ?? and ??

print (x)

and this one:

import random

print ()

print ("This prints random numbers between ?? and ??:")

for i in range(10):

x = 10 \* random.random() - 5 # between ?? and ??

print (x)

The general form of the equation above is:

y = range \* random.random() + lowest value

For example, if you want to generate numbers between 90 and 100, the range is 10 (since 100-90) and the lowest value is 90, so:

y = 10\*random.random() + 90

This is very similar in form to a familiar math formula for linear equations:

y = mx + b

where b is the initial value and m is the slope.

**Random Integers**

If you want integers, you can use ***randint****()* , which takes two arguments: a low and high value.

import random

for i in range(10):

x = random.randint(20,25) # int between 20 and 25

print (x)

**Random Other things**

Sometimes you want random... other things. In that case you can use IF statements to convert from an integer to whatever you need. For example if your best friend to change randomly:

import random

print ("My best friend is ", end = "")

f = random.randint(1,3) # pick a number between 1 and 3

if f == 1:

print ("Larry")

elif f == 2:

print ("Mary")

else:

print ("Gary")

Try running the code a few times to see the random effect.

Note: Notice on the second line:

print ("My best friend is ", end = "")

The part in blue tells the compiler to not put a "carriage return" at the end of the print statement. This ensures that the next print statement will continue on the same line.

**Exercises**

For each problem, write the code out before testing with the interpreter.

1. Produce a random float between 14 and 26.

2. Create a variable with a random decimal value between 1 and 6.

3. Produce a random integer between 1 and 100.

4. Create a variable that has either a value of 0 or 1.

5. Print a random value of either 5 or -5 (but nothing in between).

**Problems**

For each of these problems, you will get the computer to pick a random number and produce the output given. For each problem, you have to ask yourself: “what random numbers am I picking ?”

Create a short loop that simulates...

1. ...tossing a die Example: “You rolled a 2” \*\*

(Assume a die has 6 sides, numbered 1 to 6.)

1. ...tossing two die Example: “You rolled a 3 and a 4”
2. ...a random weather report. You have two things to report: a) the sunshine (sunny or cloudy) and b) the climate (hot, warm, cool). Here are some examples:

Today will be sunny and cool.

Today will be cloudy and hot.

Today will be cloudy and warm.

1. ...flipping a coin Output: “heads”
2. ...picking a card (2 things to pick – a suit and a number) “4 of hearts”

note: Ace is 1, Jack is 11, a Queen is 12, King is 13.

1. What are the odds of flipping a coin 5 times and landing on “tails” each time? Note: This is a math problem, not a programming problem!
2. Create a program that simulates flipping a coin 5 times. Create a variable called **tails** that counts the number of times you get a tail. Ex:

Heads, Heads, Tail, Heads, Tail

You got 2 tails.

1. Modify program 7. You will run program 7 inside a loop that repeatedly flips a coin 5 times and remembers how many tails were seen each time. To do this, you will create a variable called five\_tails that increments every time you get five tails in a row, another variable called four\_tails that does the same for getting four tails, and so on down to no\_tails.

Put your program in a loop that will run 3200 trials (five coin tosses each trial). What percentage of the time do you get 5 tails? Does it fit your calculation in question 6?

What is the most likely outcome of them all?

Fill in the following table:

| Outcome | Theoretical  Probability | Experimental  Probability |
| --- | --- | --- |
| 0 tails (5 heads) | 1/32 |  |
| 1 tail (4 heads) | 5/32 |  |
| 2 tails ( etc.) | 10/32 |  |
| 3 tails | 10/32 |  |
| 4 tails | 5/32 |  |
| 5 tails (0 heads) | 1/32 |  |
|  |  |  |

9. In a game a "bot" might move "randomly", but if the bot is just flipping a "left"-"right" coin then it would end up moving very erratically, changing its mind almost every instant. In this exercise we use a flag to simulate a bot moving for a while in one direction, and then for a while in the other. The flag does not change except in rare circumstances, allowing the bot to stay in one direction for a while.

**Version 1 (nervous bot - no flag)**

Create a while loop that never stops. Inside the while loop, generate a random integer from 1 to 100.

* if the number is less than 50, print the word "left"
* if the number is more than 50, print the word "right"
* if the number is equal to 50, quit the loop.

When you run this, the output should be erratic:

left

right

left

left

right

right

left

right

**Version 2 (smooth bot - with flag)**

Create a flag called **left** and set it to False. Create a while loop. The while loop continuously chooses a random integer from 1 to100. Now read the rules carefully:

* if the number is less than 4, set the flag to true
* if the number is more than 96, set the flag to false
* if the number is 50, stop the loop
* if the flag is true, print the word "left"
* if the flag is false, print the word "right".

You should see something like this:

left

left

left

right

right

right

right

right

right

left

left

etc.

Keywords: ***random module, random(), randint()***