

# Importing Modules

Python comes with an extensive library of built-in modules that make it easy to accomplish everyday tasks. With just a few lines of code, you can do anything from generating random numbers and drawing graphics to sending emails and accessing websites.

## Content Learning Objectives

*After completing this activity, students should be able to:*

- Use the `random` module to generate random float and integer sequences.
- 

## Process Skill Goals

*During the activity, students should make progress toward:*

- Navigating the Python standard library documentation. (Information Processing)



Copyright © 2019 T. Shepherd, C. Mayfield, and H. Hu. This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

## Model 1 Random Numbers

You can generate a sequence of numbers using the Python `random` module. A mathematical function is used to produce the sequence based on a *seed* value. (If no seed is given, the current system time is used.) The sequence is more accurately described as *pseudorandom*, since its output is inherently predictable.

| Python code                             | Shell output                  |
|---|-------------------------------|
| <code>import randint</code>             | no module named 'randint'     |
| <code>import random</code>              | _____                         |
| <code>randint(1, 10)</code>             | name 'randint' is not defined |
| <code>random.randint(1, 10)</code>      | 4                             |
| <code>from random import randint</code> | _____                         |
| <code>randint(1, 10)</code>             | 5                             |
| <code>seed(100)</code>                  | name 'seed' is not defined    |
| <code>random.seed(100)</code>           | _____                         |
| <code>random.random()</code>            | 0.1456692551041303            |
| <code>random.random()</code>            | 0.45492700451402135           |
| <code>random.seed(100)</code>           | _____                         |
| <code>random.random()</code>            | 0.1456692551041303            |
| <code>random.random()</code>            | 0.45492700451402135           |

### Questions (20 min)

Start time: \_\_\_\_\_

- What is the name of the module that must be imported before generating a random number?

random

- Based on Model 1, what are the names of three functions defined in the `random` module?

randint, seed, random

- Identify the syntax of the statement to import:

a) a module      `import random`

b) a function      `from random import randint`

from a module → <sup>2</sup> import this function

4. Identify the syntax of a function call assuming:  
a) the module was imported random.seed()

b) the function was imported randint()

5. How could you eliminate the need for typing the word "random" twice (in a function call) to generate a random number?

from random import random  
random()

6. Compare the shell output of your team with at least one other team. Describe the similarities and differences observed.

## ONLINE STUDENT

7. What is the effect on the random numbers generated after calling the seed method?

After calling seed() the random variable will be saved.

8. Describe one reason to set the same seed each time a program is run, and one reason to not use the seed method.

Having the same seed each time a program is run ensures there is always the same starting point, making for better comparison between each run of the program.

Having a different seed allows for a wider range of random numbers to be generated.

9. Run random.random() multiple times. Based on the results, describe:

a) the range of numbers returned by the random function floats between 0 and 1

b) the nature of the distribution of numbers generated. (Do they appear clustered around a particular value, or are they spread out uniformly over the range?)

evenly distributed between 0 and 1

10. Run random.randint(1, 10) multiple times. Based on the results, describe:

a) the range of numbers returned by the randint function ints between 0 and 10

b) the nature of the distribution of numbers generated. (Do they appear clustered around a particular value, or are they spread out uniformly over the range?)

evenly distributed between 0 and 10

