

Week 3 - Overview

This week we begin to master *recursion*, or breaking complex problems down into simpler components.

All recursive algorithms must obey three important laws:

1. A recursive algorithm must have a base case.
2. A recursive algorithm must change its state and move toward the base case.
3. A recursive algorithm must call itself, recursively.

Once we have mastered the application of recursion, our programs will be more powerful and more efficient.

Objectives

During this week, you will:

- Explain how recursion is used to increase the efficiency of a program.
- Write a recursive function.

Readings

You will be responsible for reading the following chapters this week:

- *Data Structures and Algorithms in Python* by Michael Goodrich
 - 4. Recursion

Additional pages are available in this module reviewing:

- [Overview of Recursion \(https://maryville.instructure.com/courses/43640/pages/overview-of-recursion\)](https://maryville.instructure.com/courses/43640/pages/overview-of-recursion)
- [Three Laws of Recursion \(https://maryville.instructure.com/courses/43640/pages/three-laws-of-recursion\)](https://maryville.instructure.com/courses/43640/pages/three-laws-of-recursion)
- [Binary Search \(https://maryville.instructure.com/courses/43640/pages/binary-search\)](https://maryville.instructure.com/courses/43640/pages/binary-search)
- [Designing Recursive Algorithms \(https://maryville.instructure.com/courses/43640/pages/designing-recursive-algorithms\)](https://maryville.instructure.com/courses/43640/pages/designing-recursive-algorithms)
- [Do Together: The Tower of Hanoi \(https://maryville.instructure.com/courses/43640/pages/do-together-the-tower-of-hanoi\)](https://maryville.instructure.com/courses/43640/pages/do-together-the-tower-of-hanoi)
- [Dynamic Programming \(https://maryville.instructure.com/courses/43640/pages/dynamic-programming\)](https://maryville.instructure.com/courses/43640/pages/dynamic-programming)

You may use the links above or use the next and previous buttons below to review the entire module.

Resources

This week highlights resources available through Runestone Interactive. This open education resource provides open source textbooks for computer science and programming content. The materials are licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International Licence. You may access the source material using the link below.

- Miller, B. & Ranum, D. (2011). *Problem solving with algorithms and data structures using python* (2nd ed.). Retrieved from <http://interactivepython.org/runestone/static/pythonds/index.html> (<http://interactivepython.org/runestone/static/pythonds/index.html>)