

Learning New Programming Languages

Aidan Hunt, University of Washington

Learning Objectives

After this lecture, students will have strategies for extending what they have learned in this course to learning new programming languages.

Check-in

- HW6 grades back
 - Final assignment and revisions due Friday 6/9
 - If you have questions about revisions, reach out
-

So, you've learned a lot about Python!

We've covered a lot of topics this quarter, from Python fundamentals to useful packages to advanced applications. While Python is widely used, there is a good chance you will use a different programming language in your career. A research lab you work for may use MATLAB rather than Python, or a company that hires you would like you to program in their own proprietary language.

So how can we extend what we have learned here to other programming languages? Here are some tips for learning a new programming language.

Focus on the fundamental concepts

Almost all programming languages share several core concepts, most of which we've covered here!

- Functions/methods
- Basic data types (strings, ints, floats, etc)
- Logical statements
- Control flow statements (for loops, while loops, if statements)
- Object oriented programming
- Data structures (arrays, dictionaries, etc)

Most of what changes between programming languages are:

- The "language of the computer": the syntax that is used to write code, what certain characters mean
- The "language of the people": names that we call these concepts (e.g., "doubles" in MATLAB vs "floats" in Python)

When learning a new programming language, spend some time learning

Learn the data structures

- Different programming languages specialize in different methods for storing data, and efficiency in working with certain types of data.
 - When learning a new programming language, consider the data types and data structures you work with in Python, and find equivalent or similar data structures in the language you are trying to learn.
 - Again, most of what changes are the names we use to refer to things, and the associated syntax.
-

Work forward from a language that you already know

- Find a guide that focuses on transitioning from the language you *know* to the language you want to know. For example, there are several MATLAB-to-Python transition guides that I found to be useful in preparing for this course.
- Choose a small script you have written in Python, and try to replicate it in the language you are trying to learn. This is very helpful since you know the output that you want, and are just trying to get there!

Learn the ways that you can debug your code

When you're picking up a new programming language, you will make more errors than normal, and this is fine. Just be prepared to respond to those errors:

- Slow down and work through a line of code at a time.
 - Write down the questions that you have (e.g., why do square brackets do X rather than Y?)
 - Find the debugger in the IDE that you are using, and use it often
 - Use an IDE with a variable viewer, and use it often!
-

Closing Thoughts

- Thank you for sticking with the class, I hope learned what you were hoping to learn, and then some.
- Please feel free to reach out if you want to talk through a coding project.
- Please fill out course evals