Discrete Structures: CMPSC 102

Oliver BONHAM-CARTER

Fall 2022

Week 2

Let's Discuss

Key Questions

How do I connect *mathematical terminology* (i.e,. *mapping*, *function*, *number*, *sequence*, and *set*), to the implementation of **Python programs** that declare and call functions and declare and manipulate variables?

Learning Objectives

To **remember** and **understand** some discrete mathematics and python programming concepts, setting the stage for the exploration of Discrete Structures.

The Best of Both Worlds

Discrete Structures = Math + Code

- **Discrete mathematics**
  - P Made up from: *symbols*, *character strings*, *truth values*, *objects*, and *collections of these entities* as stored in *sets* or *tuples* (for example)
- S Specifying and designing a **computer program**
  - Describe input, output, and internal objects
  - Use the vocabulary of discrete mathematics
  - Implement and test the program in a language
- **Our goal**:
  - To implement a program P that meets a particular specification S

Why combine mathematics and computer programming?

Applying Discrete Structures

For example: Program that Analyzes Web Pages

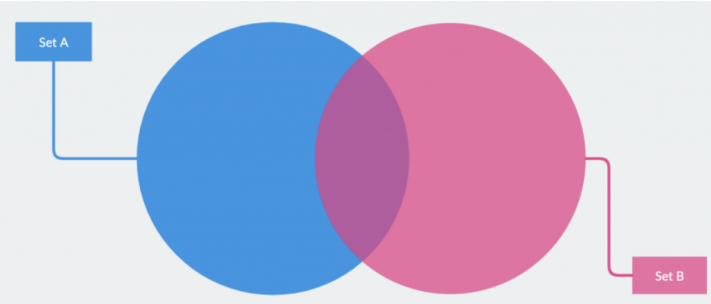
- **Informal specification**: Read two web pages and then find and output all URLs that appear in **both** pages
- Different approaches to implementing this program
  - **Informal** and **intuitive** specification
  - **Natural language specification** using **discrete structures**
  - Which one is shorter? ... clearer? ... unambiguous?
- The language of mathematics helps us to describe and implement a program that is correct, efficient, clearly documented, and maintainable!
- Where is the challenge in creating this program??

Applying Discrete Structures

Did someone say Discrete Structures Was Involved?

Mathematical *tools* in the solution

- `set`: We contain the URL information that we scrape in a `set`
- `set insertions`: We add URLs as we find them to an existing set of URLs
- `set intersection`: We will need to find the common URLs from both pages.



- Wait!! How do we use these concepts in our code?!

Program

Our Goal

Jump to different levels of abstraction (i.e., high-level versus low-level or mathematical versus technical) when we create programs

What **is** a computer program?

- Informal or intuitive specification
- Precise discrete mathematical specification
- Realization of a specification in Python program
- Bits packaged into bytes and words stored on a disk
- A process in execution on a CPU and stored in memory

Finding Solutions

How do we think about our programming?

- To find solutions, we frequently jump from a discrete mathematical specification to a Python program and back again to the specification to prepare a software solution to the problem.
- Pick the suitable level of abstraction for the problem you solve (*and the solution soon presents itself!*)

Discrete Structures with Python

Python

- Discrete structures support **precise programming**
- Benefits of using Python to explore discrete structures
- Modern language with exceptional **package support**
- Clean **syntax** and **semantics** that is easy to learn
- **Out-of-the-box support** for many **discrete structures**
- The semantics of the **language** match those of discrete structures' (the programming language language resembles the mathematics that you might employ in your work!)

About Python

- `www.python.org`
- Download python3 if you are using your own hardware

About Python...

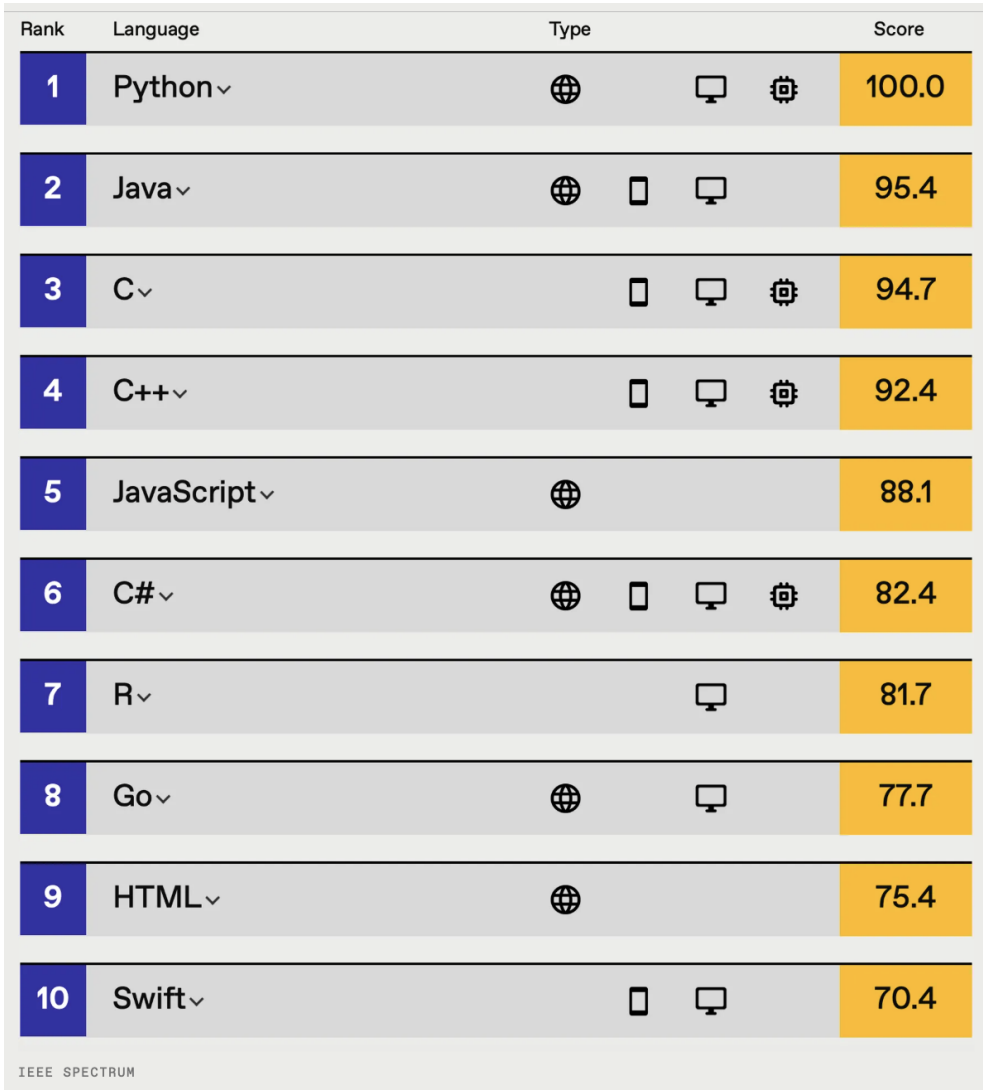
- Is an interpreted, object-oriented, high-level programming language with dynamic semantics.
- Excellent for Rapid Application Development thanks to Its high-level built in data structures, combined with dynamic typing and dynamic binding
- A scripting language for tool-making or automation
- Used for *quick and dirty* solutions, quick automation, or to connect existing components together from other languages.

About Python...

- Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance.
- Python supports modules and packages, which encourages program modularity and code reuse.
- The Python interpreter and the extensive standard library are open source and freely available in all major platforms

The Top Programming Languages

ACM 2021: Python dominates as the *de facto* platform for new technologies

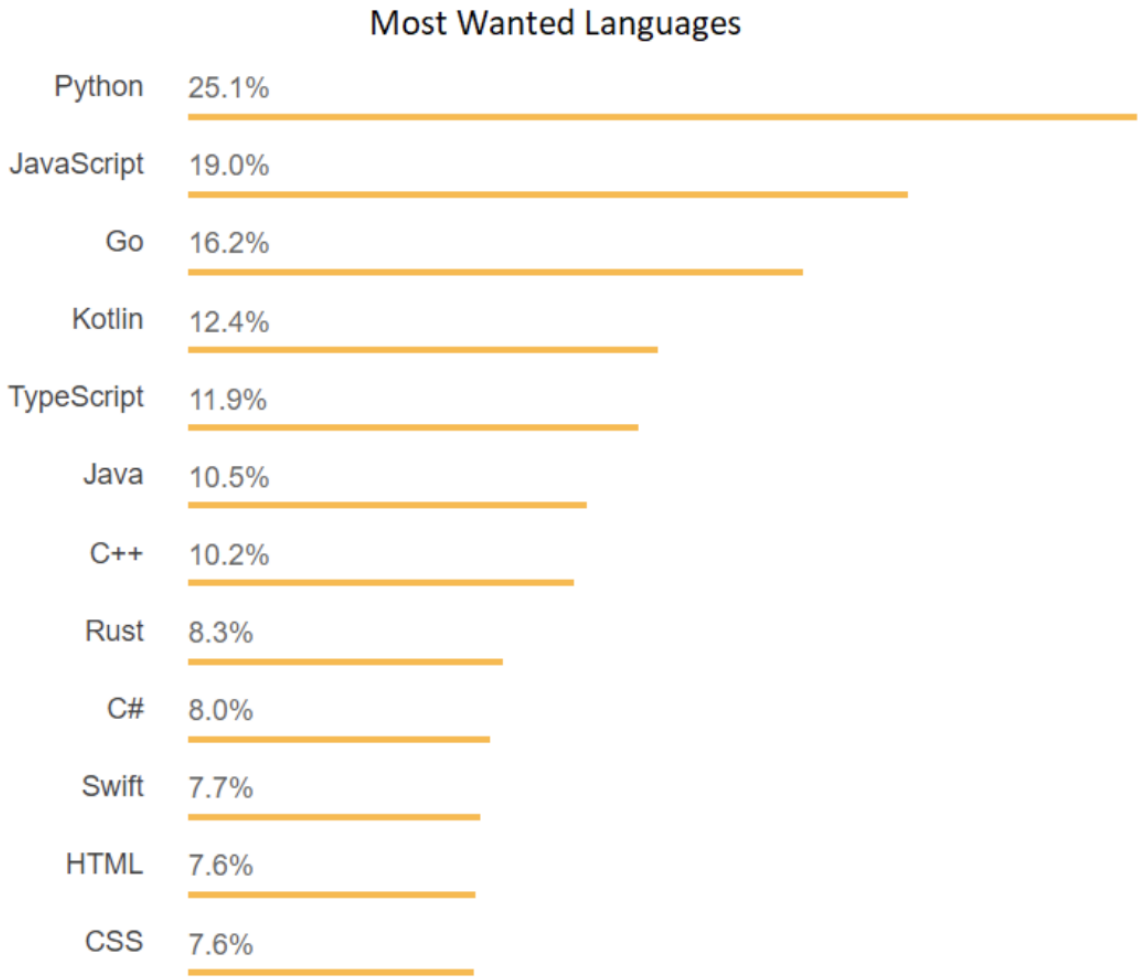


Rank	Language	Type	Score
1	Python		100.0
2	Java		95.4
3	C		94.7
4	C++		92.4
5	JavaScript		88.1
6	C#		82.4
7	R		81.7
8	Go		77.7
9	HTML		75.4
10	Swift		70.4

IEEE SPECTRUM

https://spectrum.ieee.org/top-programming-languages-2021

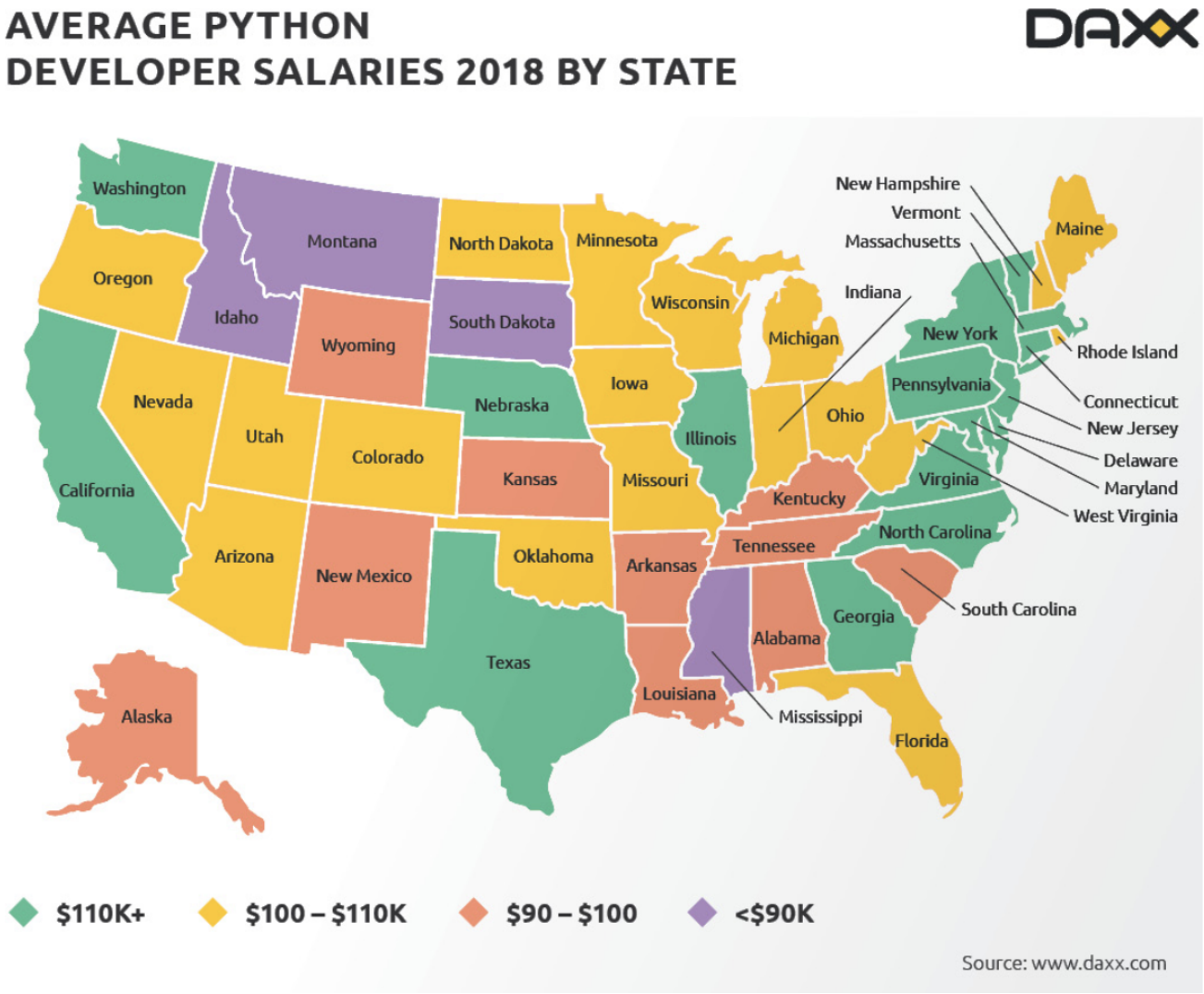
The 2018 "Most Wanted" Programming Languages



Most wanted programming languages 2018

https://www.daxx.com/article/python-developer-salary-usa

US High-Paying Python Development Jobs



https://www.daxx.com/article/python-developer-salary-usa

Average Salaries in Programming

Top companies for Python Developers in United States

Selby Jennings 3.7 ★ 7 reviews 7 salaries reported	**$245,862** per year	›
NCS 3.6 ★ 308 reviews 6 salaries reported	**$193,976** per year	›
Stefanini IT Solution 3.8 ★ 327 reviews 19 salaries reported	**$155,852** per year	›
Bank of America 3.8 ★ 31459 reviews 9 salaries reported	**$155,366** per year	›
Vaco 3.7 ★ 327 reviews 11 salaries reported	**$149,801** per year	›

Show more companies

https://www.indeed.com/career/python-developer/salaries

Who Uses Python

Users of Python Programming

- Industrial Light and Magic (George Lucas to create the FX for Star Wars).
- Google
  - Googles very first web-crawling spider was first written in Java 1.0 and was so difficult that they rewrote it into Python.
- Facebook
  - Responsible for multiple services in infrastructure management
- Netflix
  - Used to power data analyses tasks from the server side
- Dropbox
  - Built its API in Python
- And others; Instagram, Spotify, Quora, Reddit

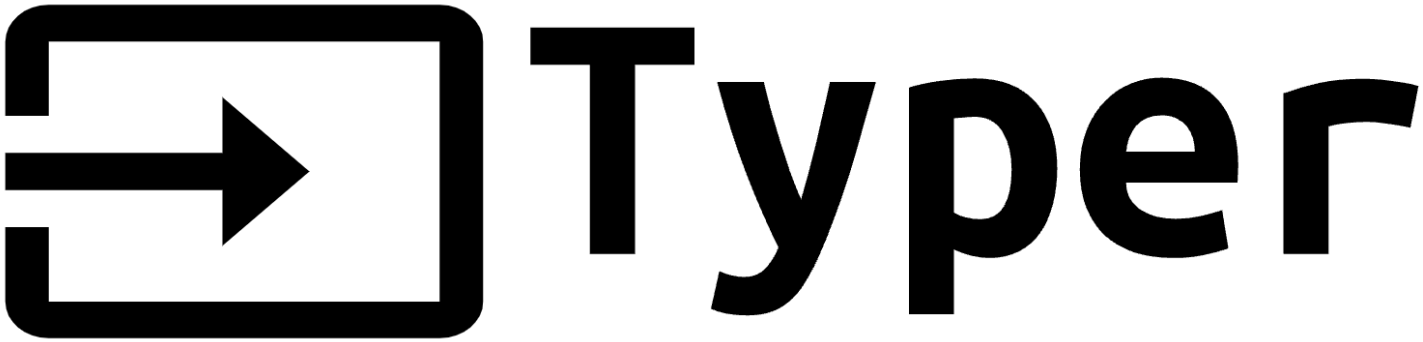
Where Can I Learn More About the Language?

Free online resources

- Think Python First Edition, by Allen B. Downey
  - `http://greenteapress.com/wp/think-python/`
- A Collection of Tutorials
  - `https://wiki.python.org/moin/BeginnersGuide/Programmers`
- Interactive Python Tutorial
  - `https://www.learnpython.org/`
- Host, run, and code Python in the cloud!
  - `https://www.pythonanywhere.com/`

Python Resource

Typer



`https://typer.tiangolo.com/`

Typer

- Command line interface support for program inputs and parameters
- Annotations: assigns types to functions that accept arguments (parameters)
- Productivity: types aid in the creation of the interface
- Checking: Confirm that inputs match expected types.

Python Resource

Poetry

`https://python-poetry.org/`

Poetry

- Management support for Python and its resources
- Environments: manage dependencies in isolation
- Package: create a stand-alone executable application
- Publish: expedite and simplify the release of program to PyPI

Python Resource

JupyterLite

`https://jupyter.cs.allegheny.edu/hub/login`

JupyterLite

- Online Python version 3 on Departmental Server
- Your own machine in the cloud with a command prompt for `git`
- To log-in, you must have a GitHub account.
- Dept of CS videos to help: `https://www.youtube.com/playlist?list=PLsYZRXov75ZHSwWiCk0-jd1RcTuu_-zmD`