Intro to Python
(Part 1)

Workshop Lead: Benjamin Z. Rudski

Date: July 16, 2024

MiCM McGill initiative in Computational Medicine

Mission statement: deliver quality workshops designed to help biomedical researchers develop the skills they need to succeed.

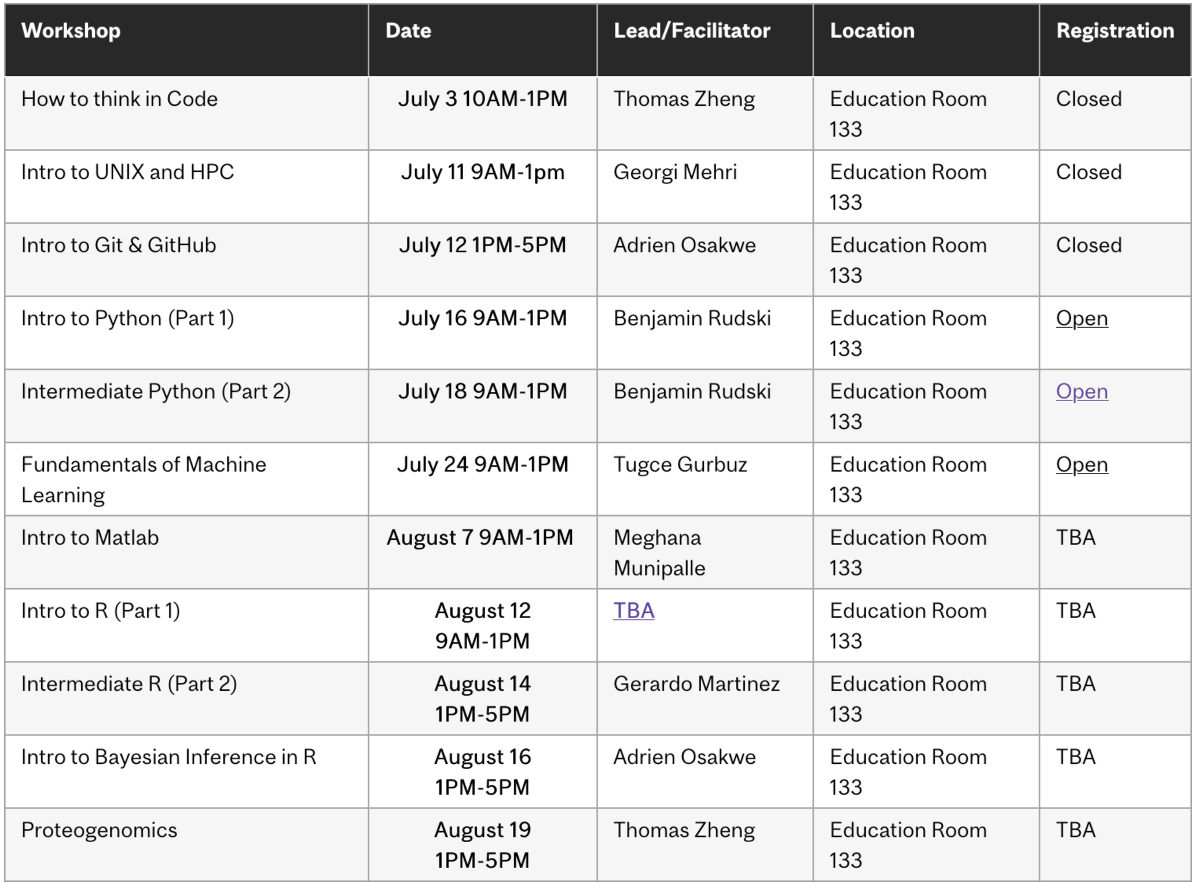
Rue University
Centre d'innovation Génome Québec et...
Dr Penfield Ave
Rue University
Rue University
Dr Penfield Ave
McTavish Reservoir
Pine Ave W
Mc Tavish St
McGill University
Famed research school founded in 1821
Mc Tavish St
d Ave

Location: 740 Dr. Penfield Avenue, Montreal, Quebec

Scan the QR code to sign up for our **mailing list**

Contact: workshop-micm@mcgill.ca

Summer 2024 Workshop Series



Workshop	Date	Lead/Facilitator	Location	Registration
How to think in Code	July 3 10AM-1PM	Thomas Zheng	Education Room 133	Closed
Intro to UNIX and HPC	July 11 9AM-1pm	Georgi Mehri	Education Room 133	Closed
Intro to Git & GitHub	July 12 1PM-5PM	Adrien Osakwe	Education Room 133	Closed
Intro to Python (Part 1)	July 16 9AM-1PM	Benjamin Rudski	Education Room 133	Open
Intermediate Python (Part 2)	July 18 9AM-1PM	Benjamin Rudski	Education Room 133	Open
Fundamentals of Machine Learning	July 24 9AM-1PM	Tugce Gurbuz	Education Room 133	Open
Intro to Matlab	August 7 9AM-1PM	Meghana Munipalle	Education Room 133	TBA
Intro to R (Part 1)	August 12 9AM-1PM	TBA	Education Room 133	TBA
Intermediate R (Part 2)	August 14 1PM-5PM	Gerardo Martinez	Education Room 133	TBA
Intro to Bayesian Inference in R	August 16 1PM-5PM	Adrien Osakwe	Education Room 133	TBA
Proteogenomics	August 19 1PM-5PM	Thomas Zheng	Education Room 133	TBA

https://www.mcgill.ca/micm/training/workshops-series

About me

- BSc from McGill in Hon. CS/Bio, Minor Math
- Third-year PhD candidate in Quantitative Life Sciences (QLS)
- Research on trabecular bone structure in the Reznikov Lab, McGill Bioengineering
  - 3D image processing and analysis
  - Programming is an almost-daily task in my life!

https://github.com/bzrudski

Outline

1. **Module 1 – Introduction to Programming (30 minutes)**
   a. Basic Concepts and Definitions
   b. Welcome to Python
2. **Module 2 – Python Basics (1 hour, 15 minutes)**
   a. Foundations of Python - A Brief Overview of Types and Variables
   b. Numbers and Comparisons
   c. Intro to Control Flow and Loops (if, while and for)
   d. Exercise
3. **Module 3 – Strings and Collections: An Object Primer (1 hour)**
   a. Introducing the String!
   b. Introduction to Tuples, Lists and Dictionaries
   c. Exercise
4. **Module 4 – Modules and Packages (40 minutes)**
   a. Using Modules
   b. Package Management
   c. Exercise
5. **Module 5 – Where to go from here (10 minutes)**
   a. What to learn next? How?
   b. How to get help and how not to get help?
   c. Other cool programming topics

Module 1

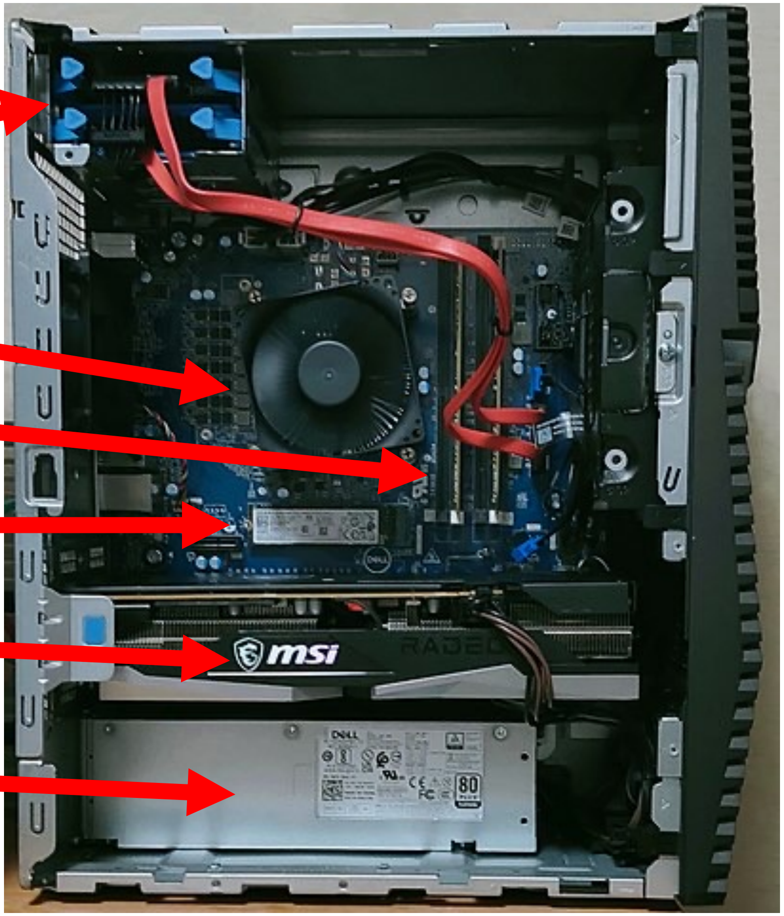
Introduction to Programming

Module Outline

a. Basic Concepts and Definitions
    a. What is a Computer?
    b. What is a Program?
    c. What are Programming Languages?
b. Welcome to Python
    a. What is Python?
    b. How to Install Python
    c. Tools for Using Python

Basic Concepts and Definitions

- What is a computer?



Hard drive

CPU

RAM

Motherboard

Graphics Card

Power Supply

Dell G5 5000 motherboard.jpg, by Project Kei, licensed under the Creative Commons Attribution-Share Alike 4.0 International license.

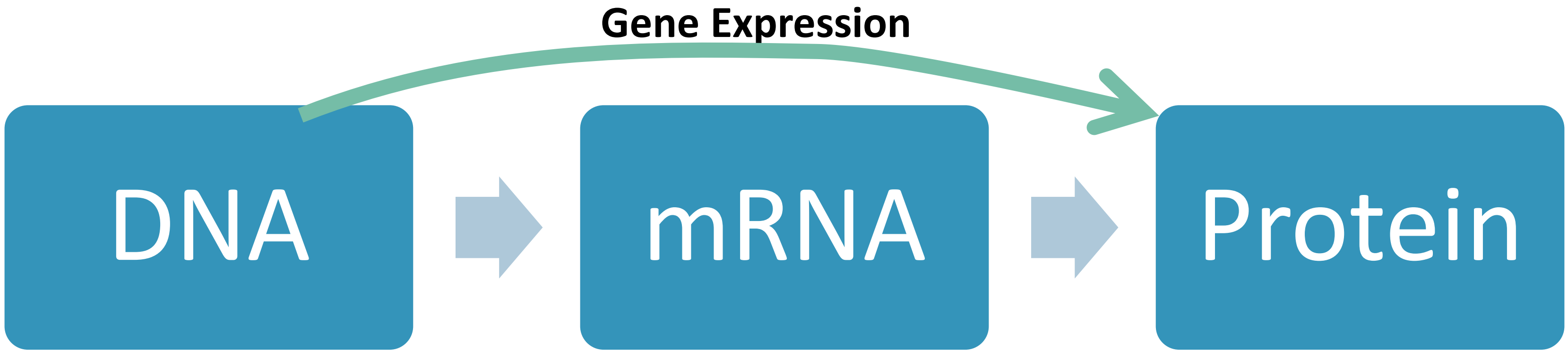
Basic Concepts and Definitions

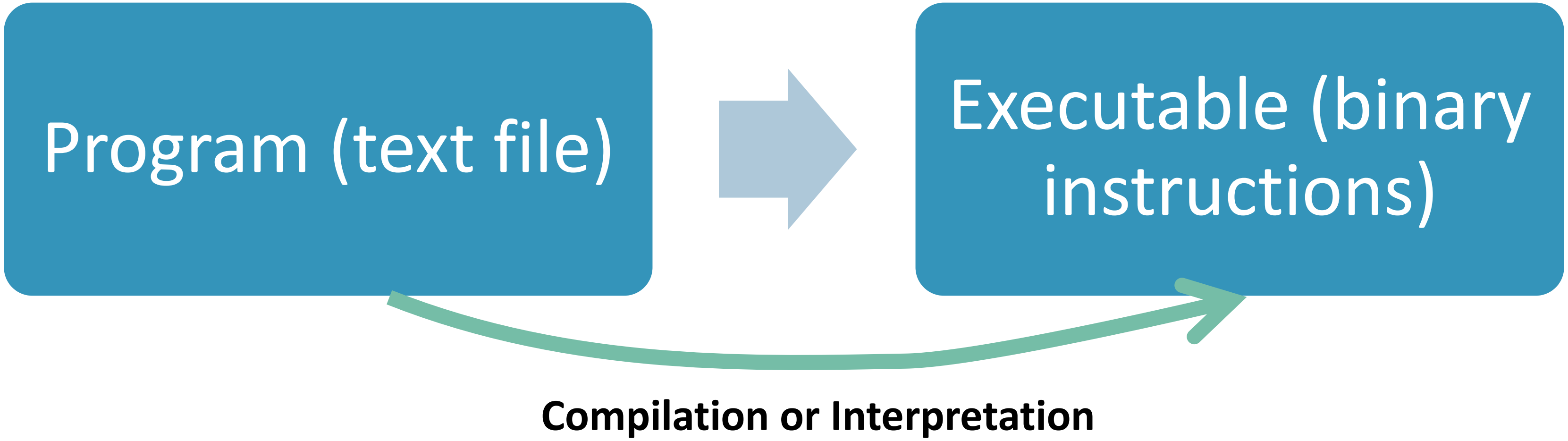
- What is a computer?
  - RAM: memory – store data
  - CPU: processor – perform operations on data
- How do we tell it what operations to do on what data?...
- **Programming!**
- Program is a **text file** that contains instructions:
  - What operations to do
  - On what data

Basic Concepts and Definitions

- What is a program? – Instructions
- How do we write a program?
  - Using a **programming language**
- **Poll:** Who does the programming language help?
  - (a) The computer
  - (b) The programmer
  - (c) Nobody… it's a useless waste of time!
- Let's see why…

Basic Concepts and Definitions




Program (text file)
Executable (binary instructions)
Compilation or Interpretation

Welcome to the Python Programming Language!

- For more history: https://en.wikipedia.org/wiki/History_of_Python
- Introduced in 1991 by Guido van Rossum
- Features:
  - Free and Open Source
  - Interpreted
  - Object-Oriented
- https://python.org

Welcome to the Python Programming Language!

- Free and Open Source
  - Anyone can download, use, **modify and distribute** the Python programming language.
- Interpreted
  - Python scripts are run line-by-line
  - Can easily launch it from the command line and have access to **interactive shell**
- Object-Oriented
  - “Objects” – collections of data and manipulations that make it easier to represent the real world

Interactive Workshop!

- That’s pretty much all that will be in the slides… For the rest, we’ll go to a Jupyter Notebook:

To the repository!

To summarize

- ✓ Computers are machines that store data and perform operations.
- ✓ Data can be represented as numbers, strings or more complicated collections that can be processed using various functions.
- ✓ Packages and modules offer additional functionality not included by default.

Now you are ready to:

- Use variables and collections to represent data in Python code.
- Use control flow and loops to write powerful code.
- Use functions from existing packages and modules.

Acknowledgements

- Thank you to MiCM for giving me this opportunity and for helping me along the way.
- Thank you to the professors from the McGill School of Computer Science for helping me along my programming journey and for inspiring me to share my programming experience with others.
- Thank you to Professor Mathieu Blanchette, whose COMP 204 course helped introduce me to Python (back in Fall 2018).