McGill initiative in Computational Medicine

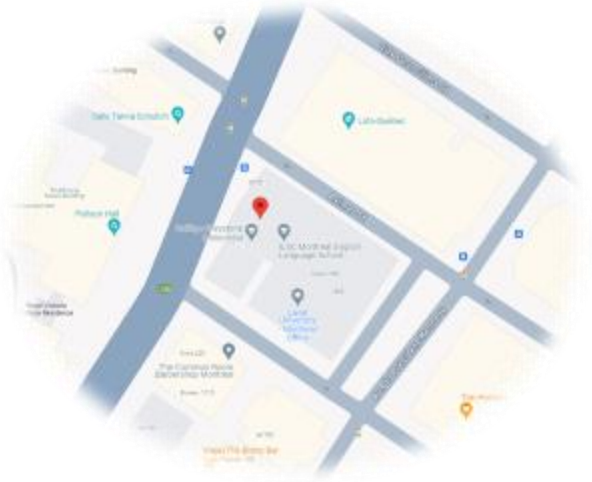
Intro to Python
Part 1

Benjamin Rudski
October 28, 2024

McGill | Quantitative Life Sciences | Sciences quantitatives du vivant

MiCM McGill initiative in Computational Medicine

QLS-MiCM mission statement: deliver quality workshops designed to help biomedical researchers develop the skills they need to succeed.



Location: 550 Sherbrooke Street, Montreal, Quebec

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

Contact: workshop-micm@mcgill.ca

McGill Quantitative Life Sciences | Sciences quantitatives du vivant

Workshop Series

Workshop	Date	Location	Registration
How to think in Code	September 18 10AM-12PM	Education Room 133	Closed
Intro to Git & GitHub	September 25 8AM-12PM	Education Room 133	Closed
Intro to Unix	September 27 1PM-3PM	Education Room 133	Closed
Intro to Python (Part 1)	October 28 9AM-11AM	Education Room 133	Open
Intermediate Python (Part 2)	November 1 10AM30-12AM30	McIntyre Room 519	Open
Exploring Matlab	November 4 10AM-12PM	Education Room 133	Open
Intro to R (Part 1)	November 13 1PM-5PM	Education Room 133	Open
Statistics in R (Part 2)	November 18 1PM-5PM	McIntyre Room 519	Open
Data Visualization	November 21 2PM-6PM	Education Room 133	TBA
Intro to scRNA-seq	November 25 10AM-12PM	Education Room 133	TBA
Advanced scRNA-seq	December 2 10AM-12PM	Education Room 133	TBA

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

Outline

1. **Module 1 – Introduction to Python (10 minutes)**
    a. Welcome to Python
2. **Module 2 – Python Basics (45 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 (45 minutes)**
    a. Introducing Objects
    b. Introducing the String!
    c. Introduction to Tuples, Lists and Dictionaries
    d. Exercise
4. **Module 4 – 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. Glimpse of other cool programming topics

Module 1
Welcome to Python

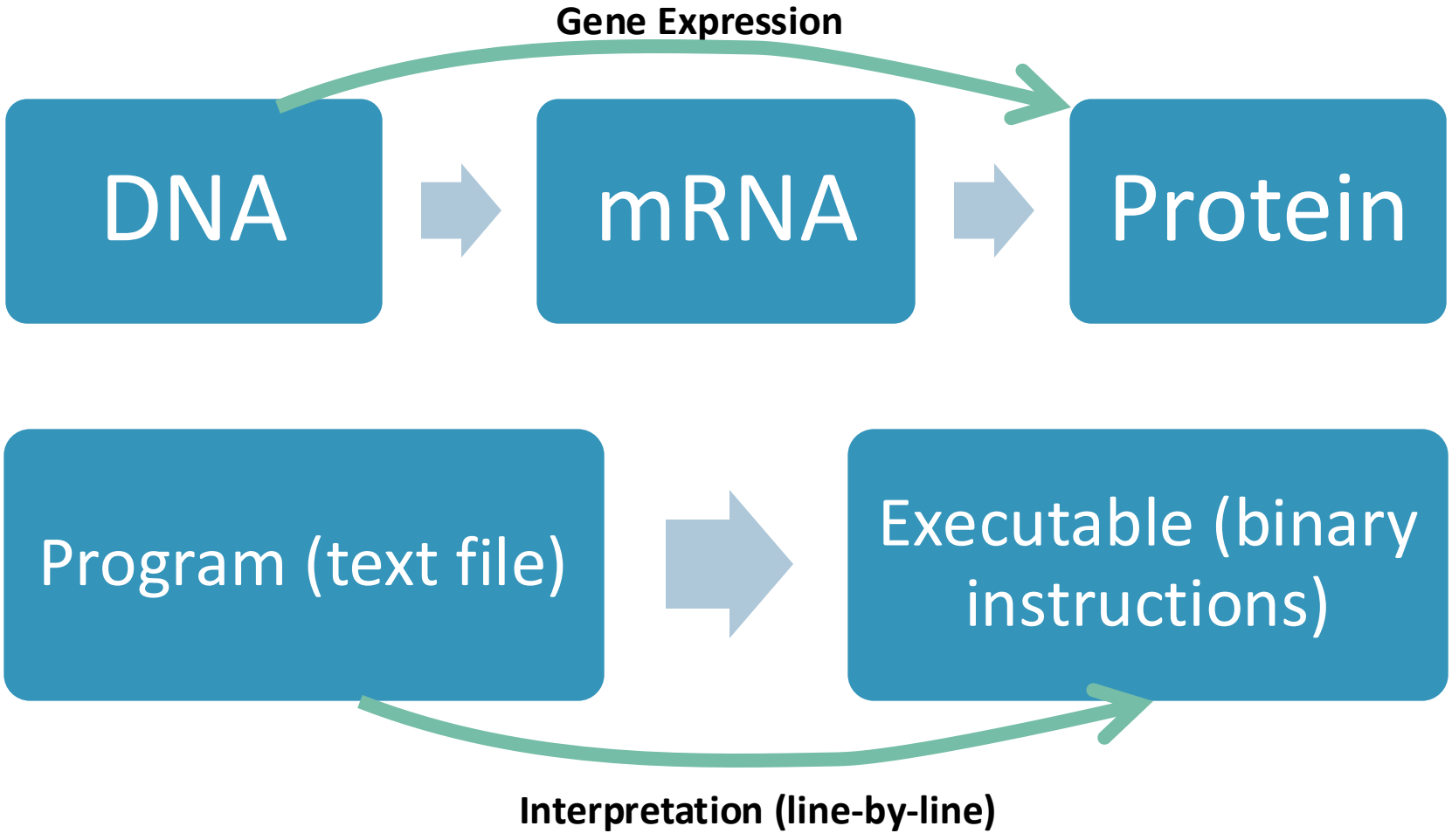
Welcome to the Python Programming Language!

- 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

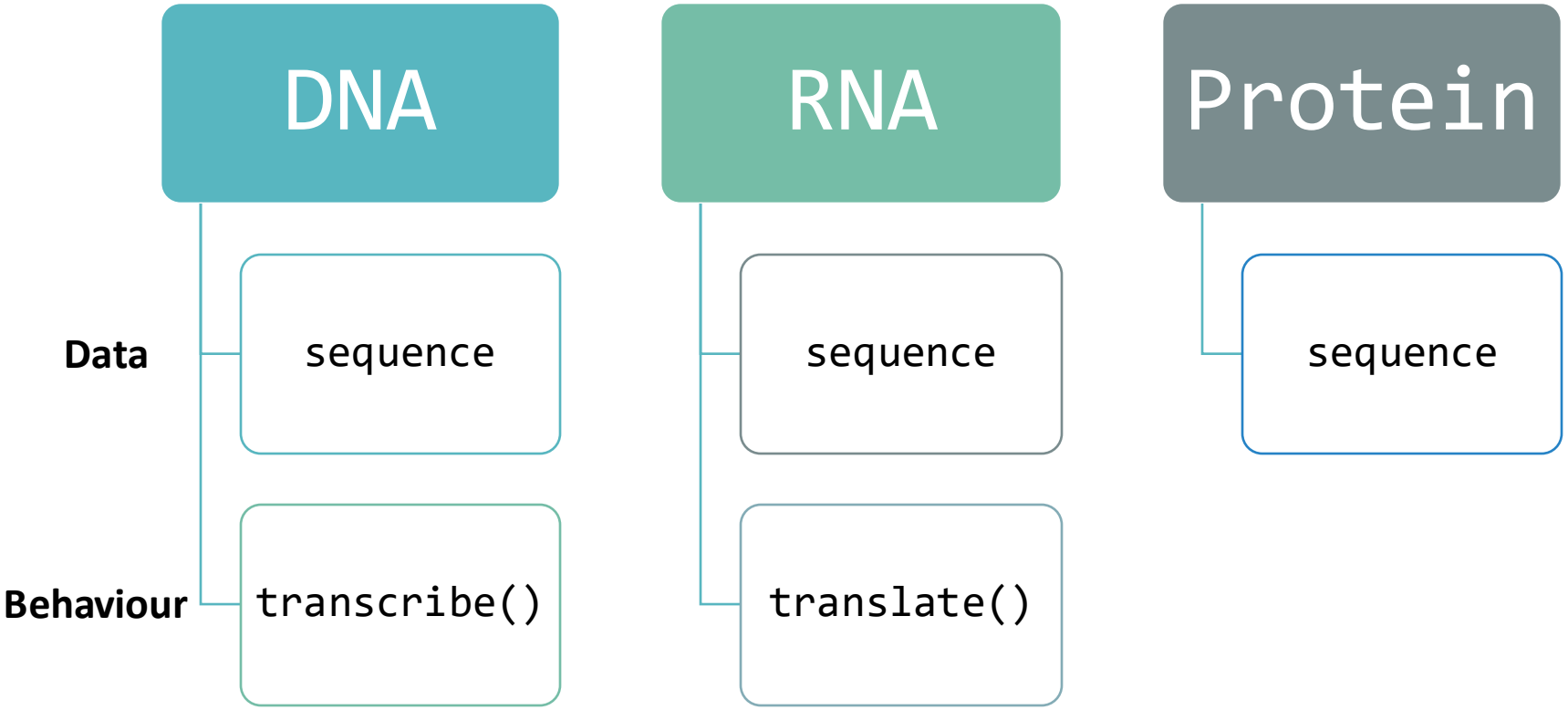
Free and Open Source

- Everyone is free to: download, use, modify and redistribute Python.
- Python is developed, in part, by **the community of users**.
- https://docs.python.org/3/license.html

Interpreted



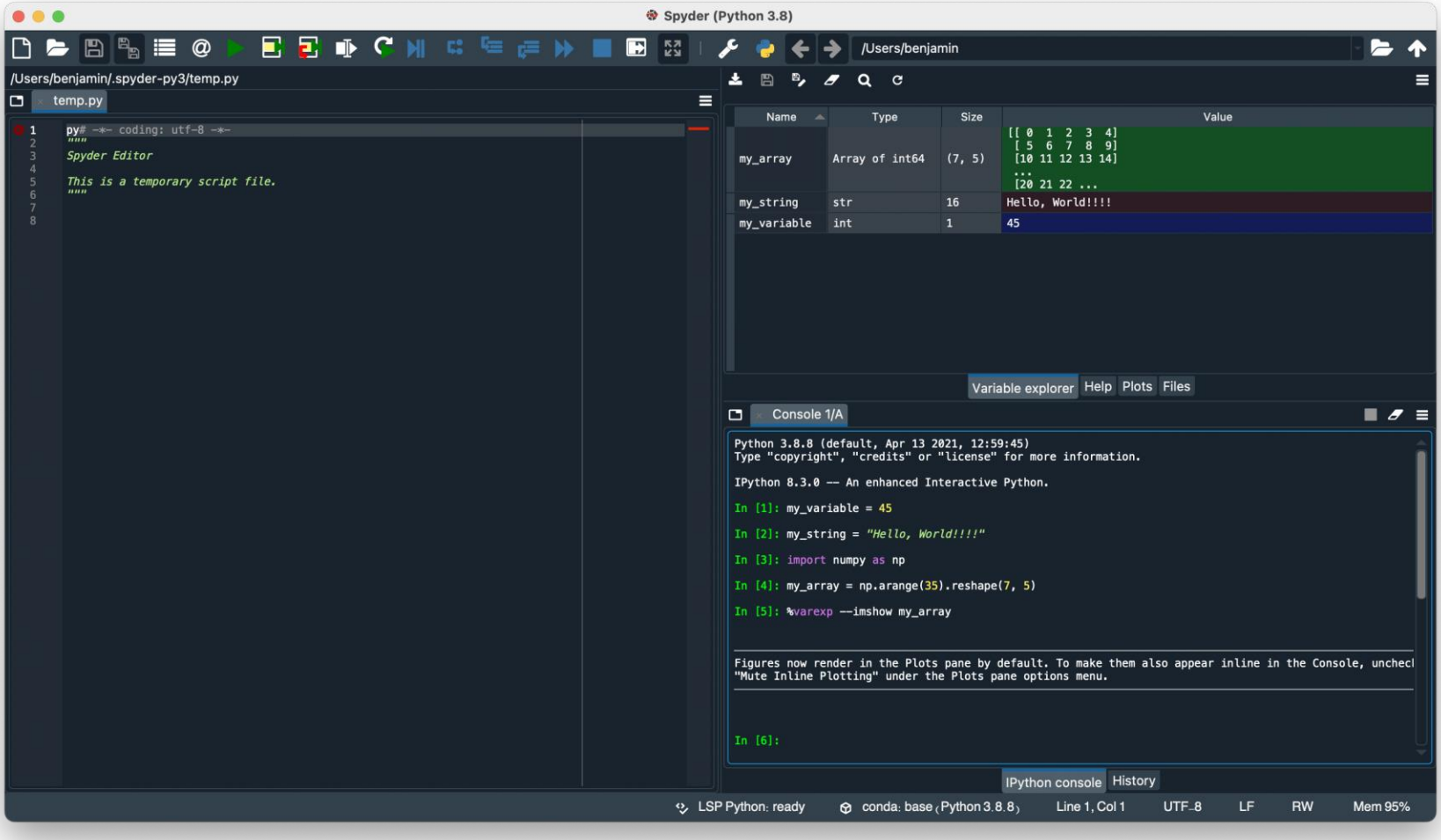
Object-Oriented



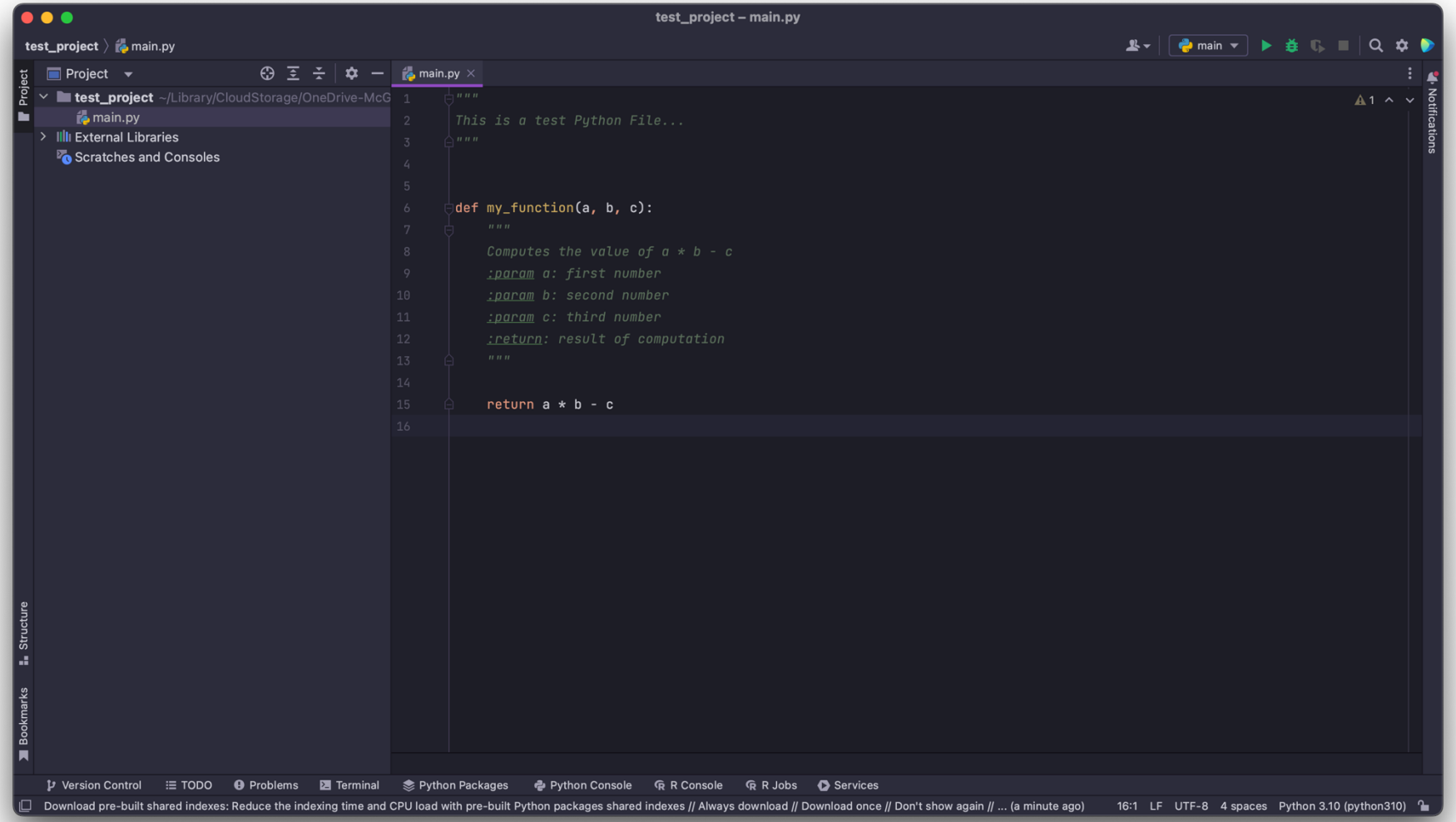
Installing Python

	Official Installer	Miniconda	Anaconda
Includes Python			
Includes `pip`			
Includes `conda`			
Allows easily installing multiple versions			
Includes many packages			

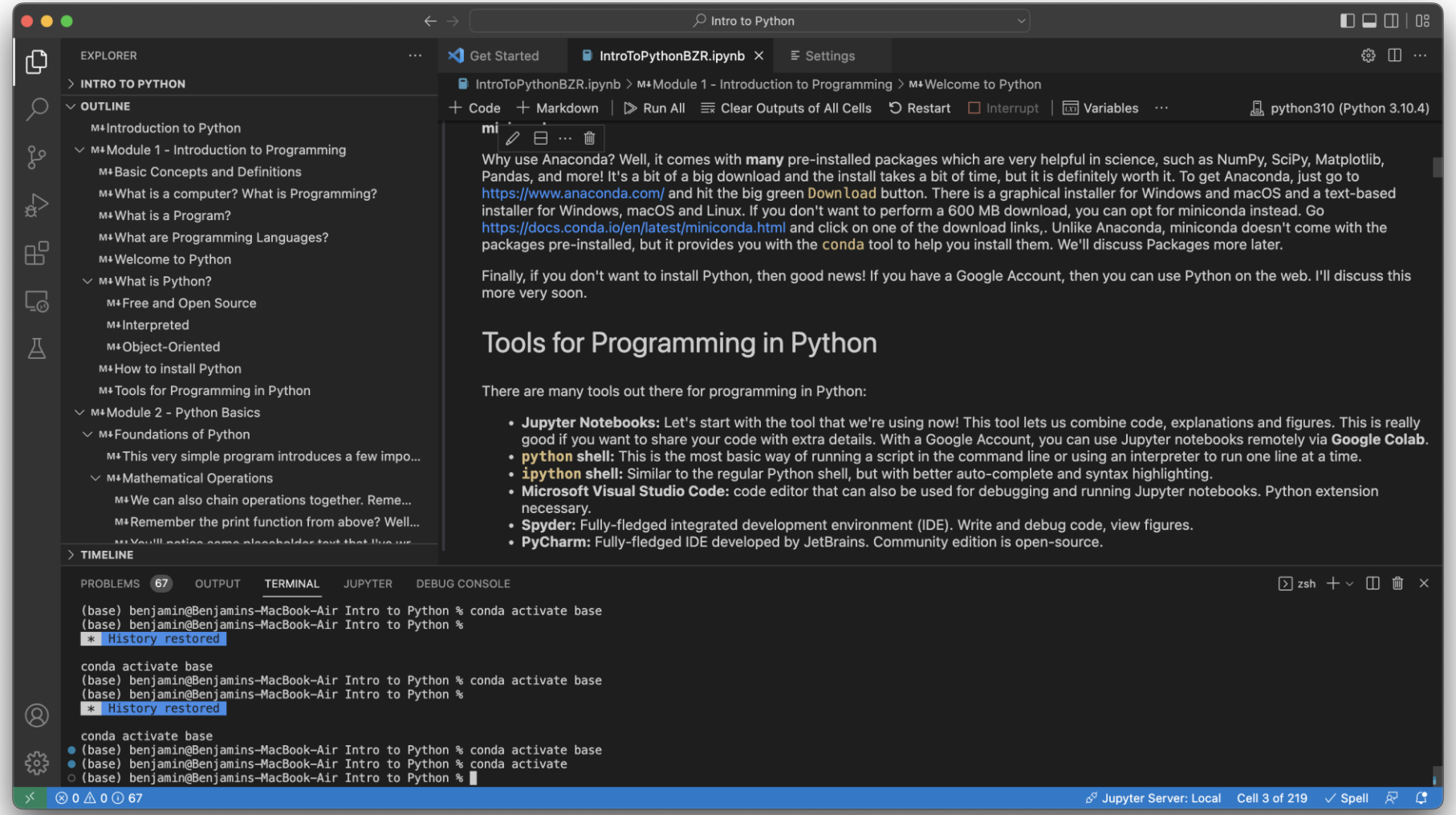
Tools for Programming in Python



Tools for Programming in Python



Tools for Programming in Python



McGill | Quantitative Life Sciences | Sciences quantitatives du vivant

Module Summary

- **Python** is a programming language that is **open source, interpreted and object-oriented**.
- There are **various ways** to **install** Python.
- We can use a variety of **tools** to program in Python.

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!

Module 2
Strings and Collections
An Object Primer

What is an Object?

Object

Attributes

- Variables
- Describe the object

Methods

- Functions
- Compute values
- Alter the object

Objects

Car

- colour
- year
- model
- turn_on()
- turn_off()
- change_gear()
- toggle_headlights()

Mouse

- height
- weight
- age
- sex
- genotype

To summarize

- ✓ Python is a **free and open-source**, **interpreted object-oriented** language.
- ✓ Data can be stored in **variables** of several types, including **strings**, **integers**, **floating point numbers** and **Booleans.**
- ✓ **Collection types**, such as **tuples**, **lists** and **dictionaries** can be used to store **multiple** data points.
- ✓ **Control flow** and **loops**, help decide which lines to run and allow lines to be repeated.

Now you are ready to:

- Store data in variables and collections.
- Perform basic operations on these data.
- Use control flow and loops to write powerful code.

Acknowledgements

- Thank you to QLS-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).
- Thank you to the Python community!