Intro to Python
Part 1

Workshop Lead:
Month Date, Year

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.

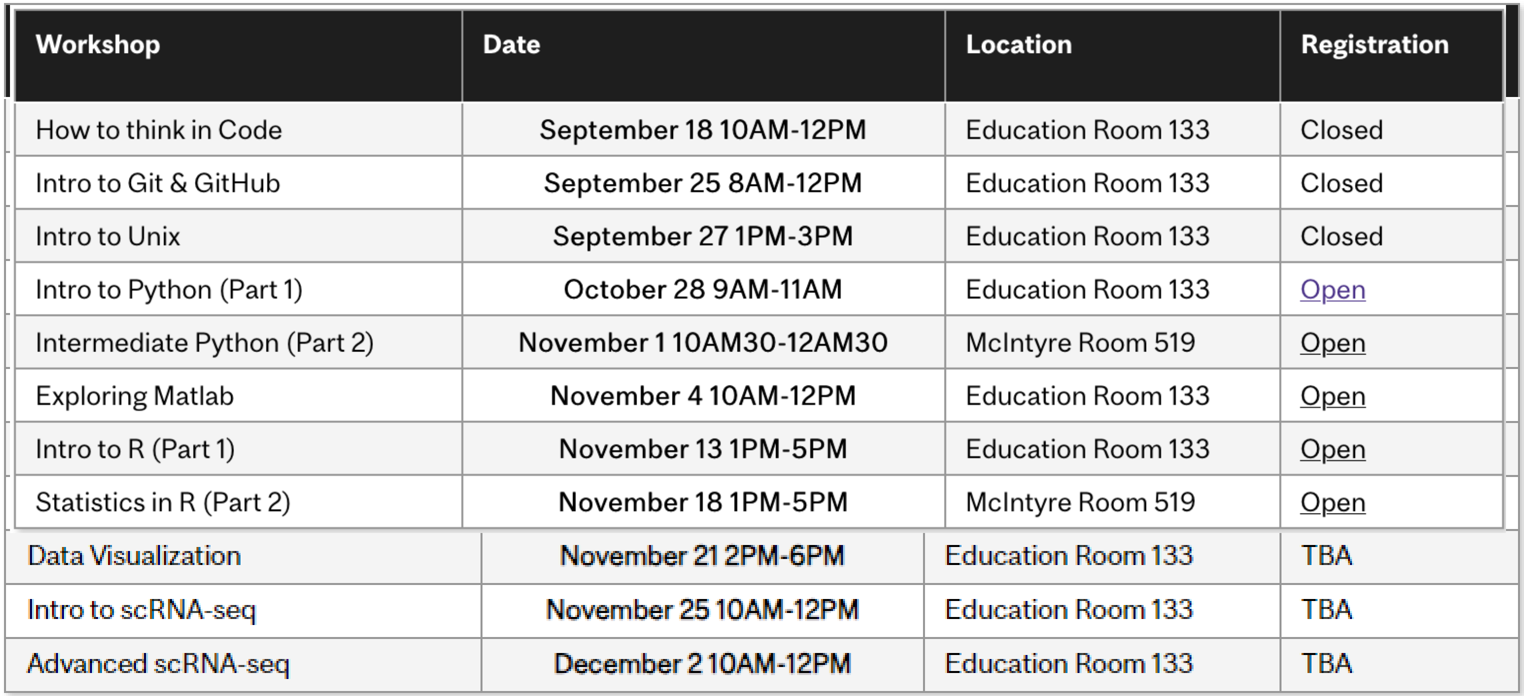
Location: 550 Sherbrooke
Street, Montreal, Quebec

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?

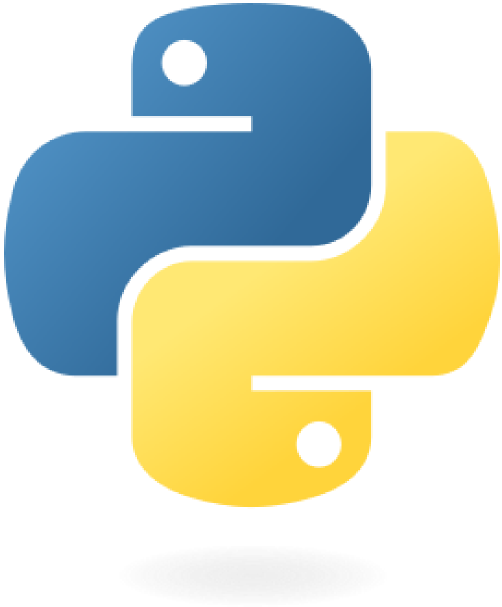
   b. How to get help and how not to get help

   c. Glimpse of other cool programming topics

Module 1

Welcome to Python

Welcome to the Python Programming Language!

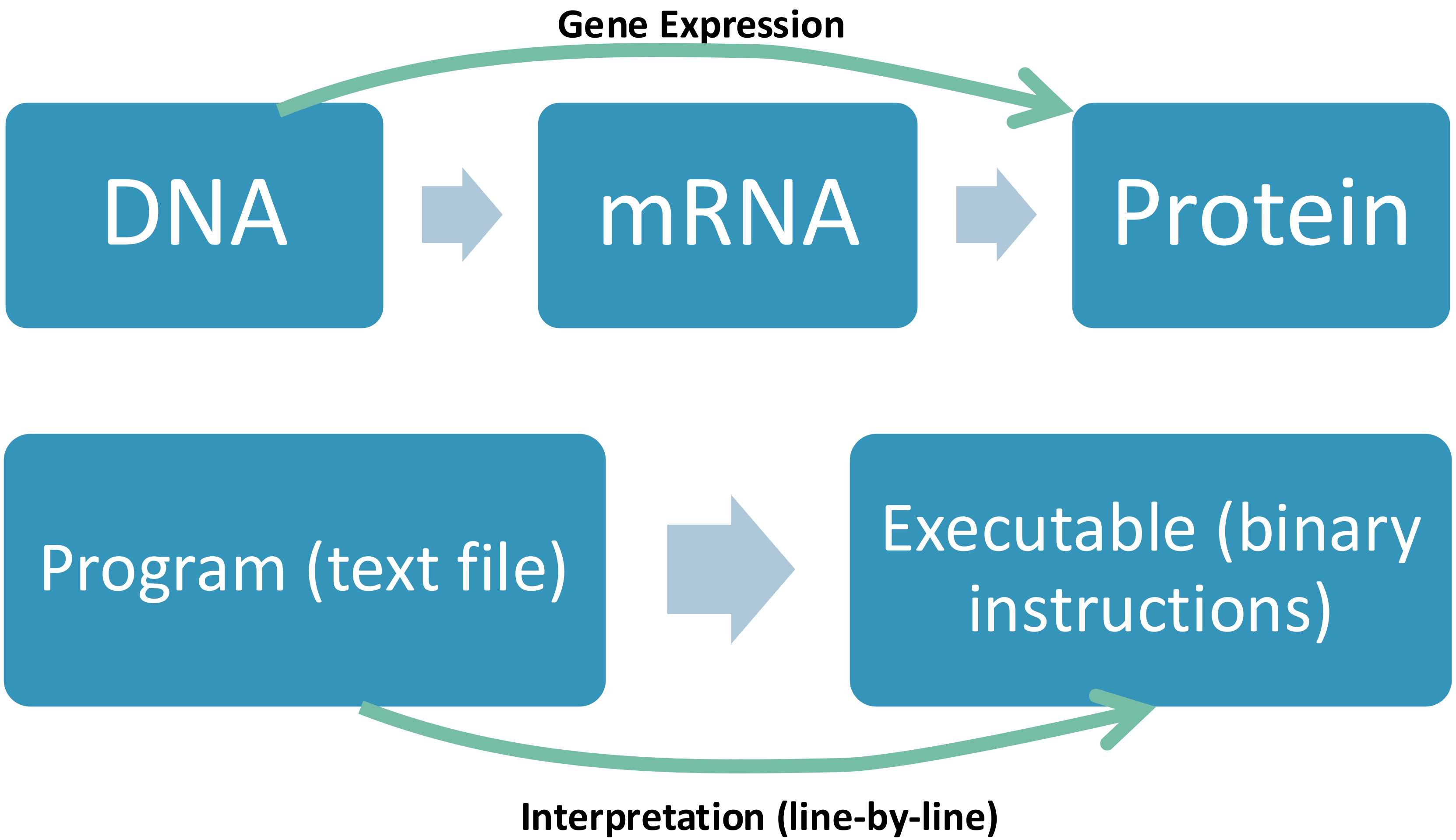


- For more history: https://en.wikipedia.org/wiki/History_of_Python
- Introduced in 1991 by Guido van Rossum
- Features:
  - Free and Open Source
  - Interpreted
  - Object-Oriented
- https://python.org

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



Gene Expression

DNA

mRNA

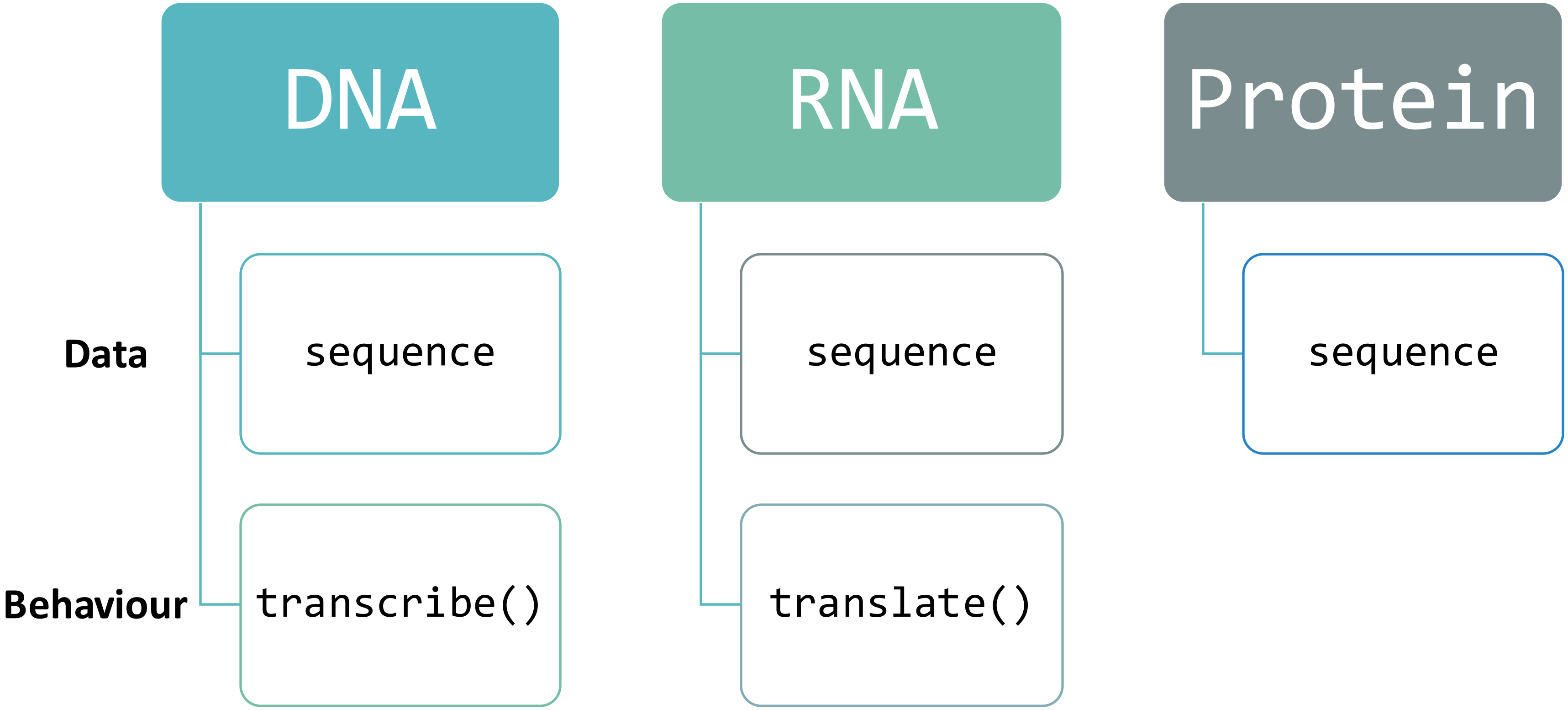
Protein

Program (text file)

Executable (binary instructions)

Interpretation (line-by-line)

Object-Oriented



DNA

RNA

Protein

Data

sequence

sequence

sequence

Behaviour

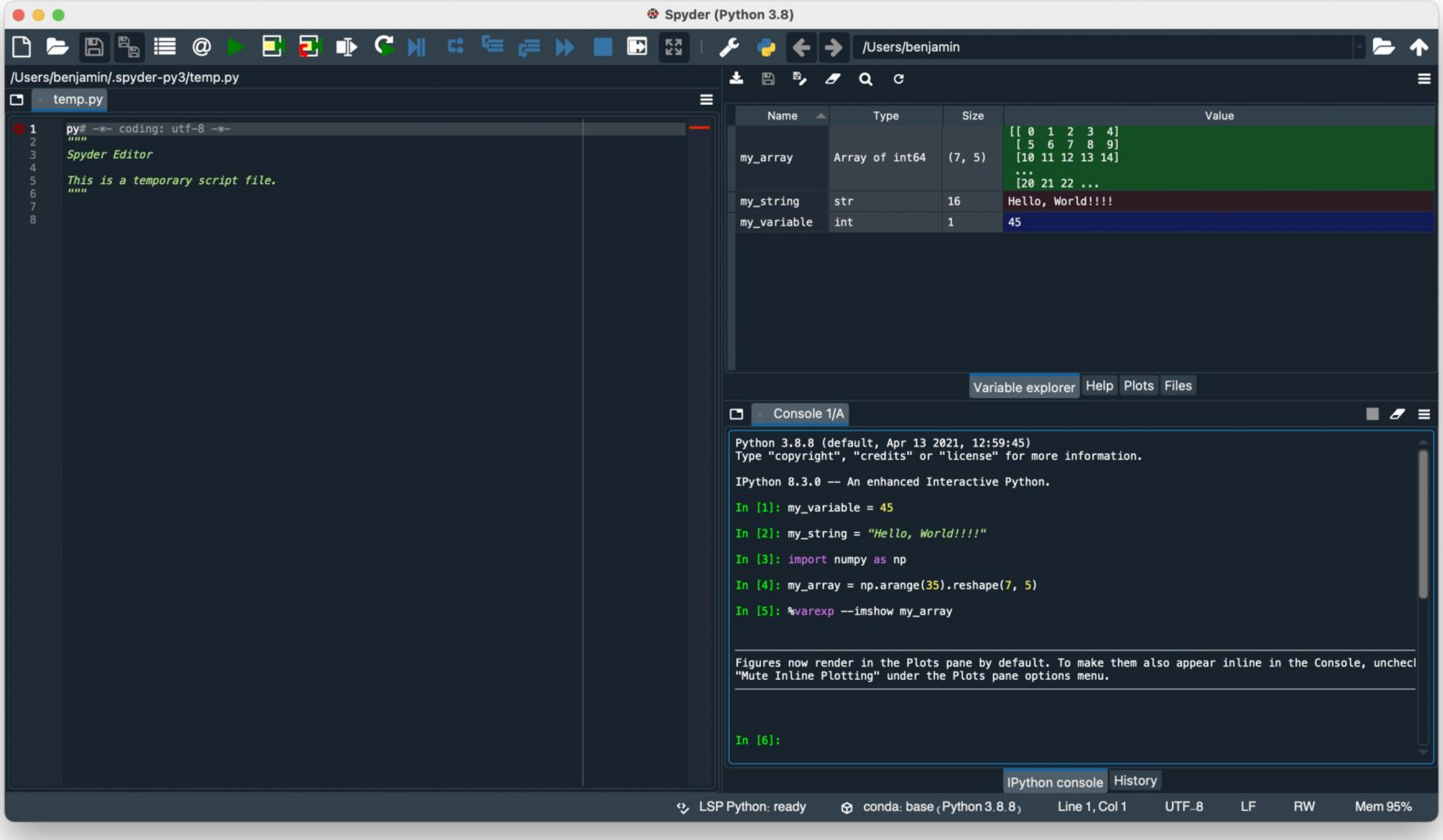
transcribe()

translate()

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

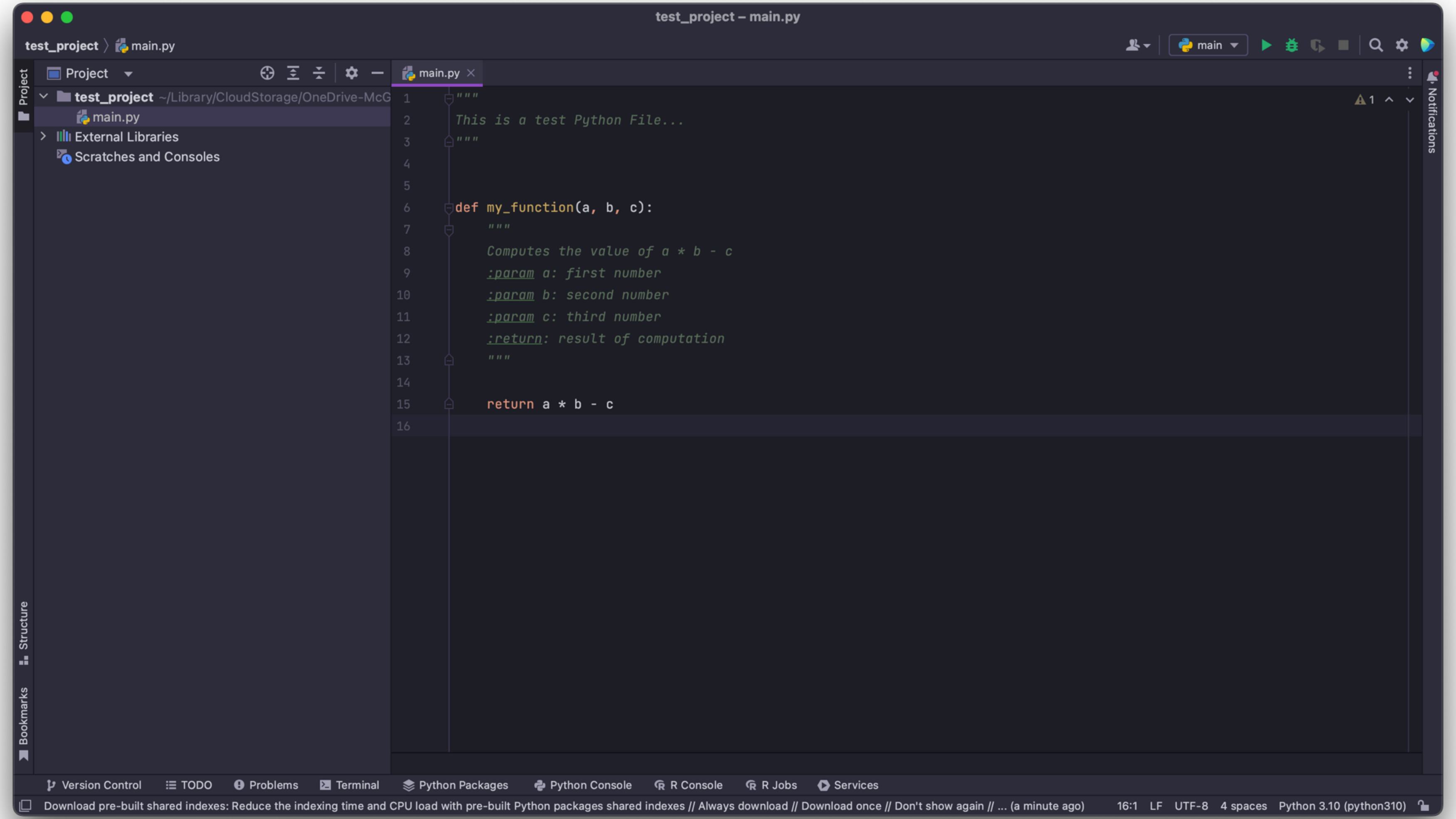


```
py# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""
```

Name	Type	Size	Value
my_array	Array of int64	(7, 5)	[[0 1 2 3 4] [5 6 7 8 9] [10 11 12 13 14] ... [20 21 22 ...
my_string	str	16	Hello, World!!!!
my_variable	int	1	45

```
Python 3.8.8 (default, Apr 13 2021, 12:59:45)
Type "copyright", "credits" or "license" for more information.

IPython 8.3.0 -- An enhanced Interactive Python.

In [1]: my_variable = 45

In [2]: my_string = "Hello, World!!!!"

In [3]: import numpy as np

In [4]: my_array = np.arange(35).reshape(7, 5)

In [5]: %varexp --imshow my_array


Figures now render in the Plots pane by default. To make them also appear inline in the Console, unchecl
"Mute Inline Plotting" under the Plots pane options menu.



In [6]:
```

Tools for Programming in Python

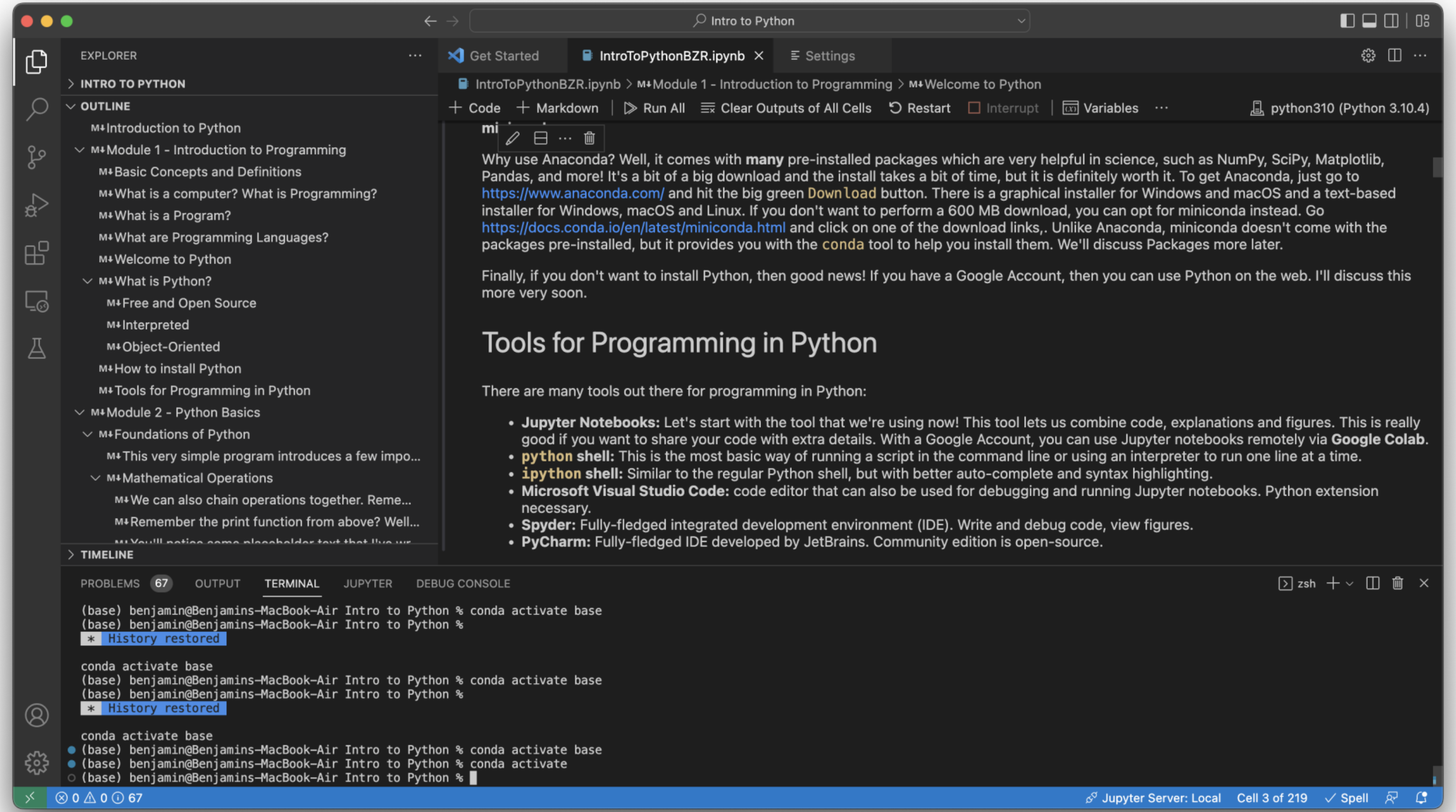


```python
"""
This is a test Python File...
"""


def my_function(a, b, c):
    """
    Computes the value of a * b - c
    :param a: first number
    :param b: second number
    :param c: third number
    :return: result of computation
    """

    return a * b - c

```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Download once // Don't show again // ... (a minute ago)

McGill | Quantitative Life Sciences | Sciences quantitatives du vivant

Tools for Programming in Python


Intro to Python
EXPLORER
INTRO TO PYTHON
OUTLINE
Introduction to Python
Module 1 - Introduction to Programming
Basic Concepts and Definitions
What is a computer? What is Programming?
What is a Program?
What are Programming Languages?
Welcome to Python
What is Python?
Free and Open Source
Interpreted
Object-Oriented
How to install Python
Tools for Programming in Python
Module 2 - Python Basics
Foundations of Python
This very simple program introduces a few impo...
Mathematical Operations
We can also chain operations together. Reme...
Remember the print function from above? Well...
TIMELINE
Get Started
IntroToPythonBZR.ipynb
Settings
IntroToPythonBZR.ipynb > Module 1 - Introduction to Programming > Welcome to Python
Code
Markdown
Run All
Clear Outputs of All Cells
Restart
Interrupt
Variables
python310 (Python 3.10.4)
PROBLEMS 67
OUTPUT
TERMINAL
JUPYTER
DEBUG CONSOLE
zsh
0 0 67
Jupyter Server: Local
Cell 3 of 219
Spell

Why use Anaconda? Well, it comes with **many** pre-installed packages which are very helpful in science, such as NumPy, SciPy, Matplotlib, Pandas, and more! It's a bit of a big download and the install takes a bit of time, but it is definitely worth it. To get Anaconda, just go to https://www.anaconda.com/ and hit the big green `Download` button. There is a graphical installer for Windows and macOS and a text-based installer for Windows, macOS and Linux. If you don't want to perform a 600 MB download, you can opt for miniconda instead. Go https://docs.conda.io/en/latest/miniconda.html and click on one of the download links,. Unlike Anaconda, miniconda doesn't come with the packages pre-installed, but it provides you with the `conda` tool to help you install them. We'll discuss Packages more later.

Finally, if you don't want to install Python, then good news! If you have a Google Account, then you can use Python on the web. I'll discuss this more very soon.

Tools for Programming in Python

There are many tools out there for programming in Python:

- **Jupyter Notebooks:** Let's start with the tool that we're using now! This tool lets us combine code, explanations and figures. This is really good if you want to share your code with extra details. With a Google Account, you can use Jupyter notebooks remotely via **Google Colab**.
- **`python` shell:** This is the most basic way of running a script in the command line or using an interpreter to run one line at a time.
- **`ipython` shell:** Similar to the regular Python shell, but with better auto-complete and syntax highlighting.
- **Microsoft Visual Studio Code:** code editor that can also be used for debugging and running Jupyter notebooks. Python extension necessary.
- **Spyder:** Fully-fledged integrated development environment (IDE). Write and debug code, view figures.
- **PyCharm:** Fully-fledged IDE developed by JetBrains. Community edition is open-source.

```
(base) benjamin@Benjamins-MacBook-Air Intro to Python % conda activate base
(base) benjamin@Benjamins-MacBook-Air Intro to Python %
 *  History restored

conda activate base
(base) benjamin@Benjamins-MacBook-Air Intro to Python % conda activate base
(base) benjamin@Benjamins-MacBook-Air Intro to Python %
 *  History restored

conda activate base
(base) benjamin@Benjamins-MacBook-Air Intro to Python % conda activate base
(base) benjamin@Benjamins-MacBook-Air Intro to Python % conda activate
(base) benjamin@Benjamins-MacBook-Air Intro to Python %
```

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!

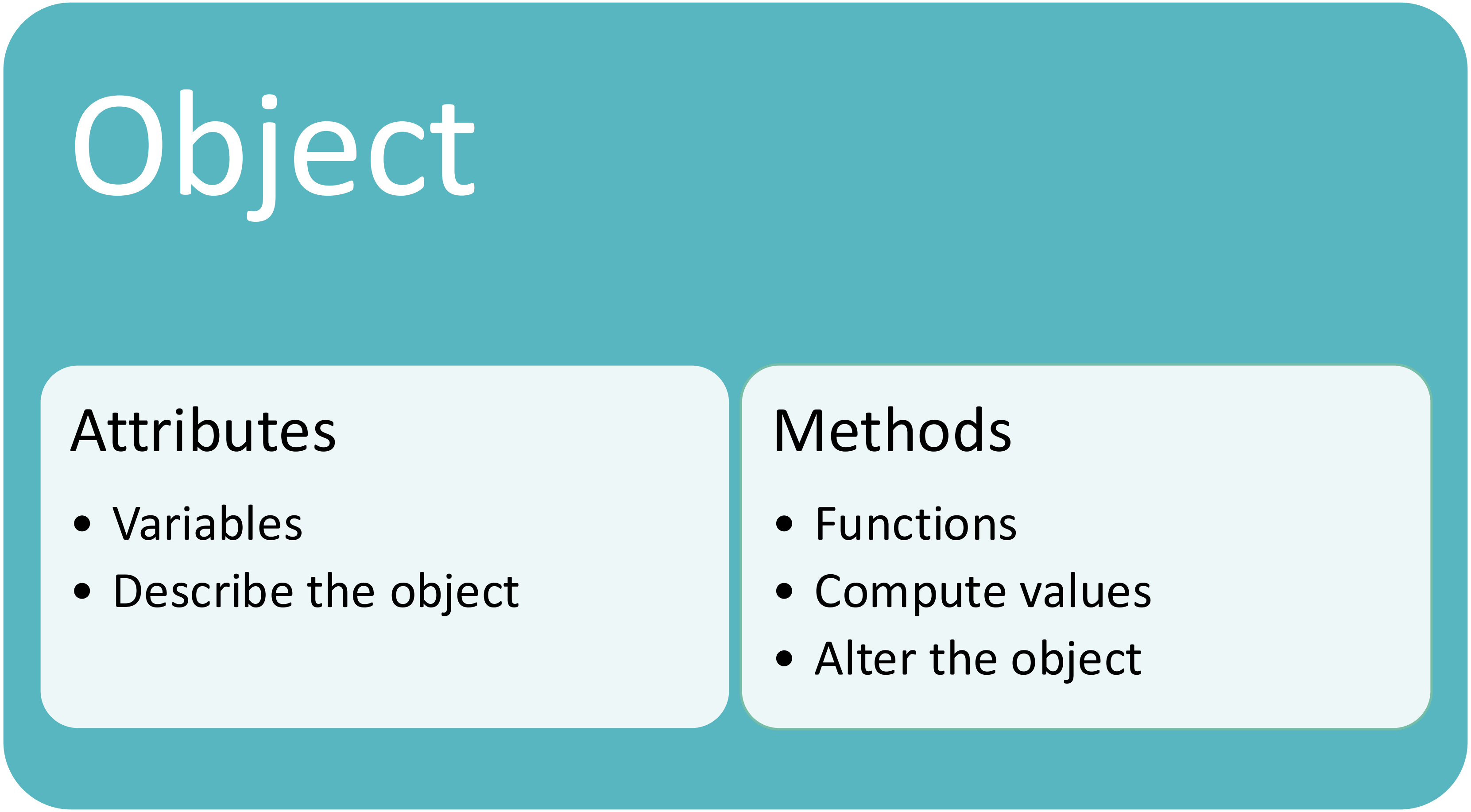
- That’s pretty much all that will be in the slides… For the rest, we’ll go to a Jupyter Notebook:

To the repository!

Module 2
Strings and Collections

An Object Primer

What is an Object?



Object

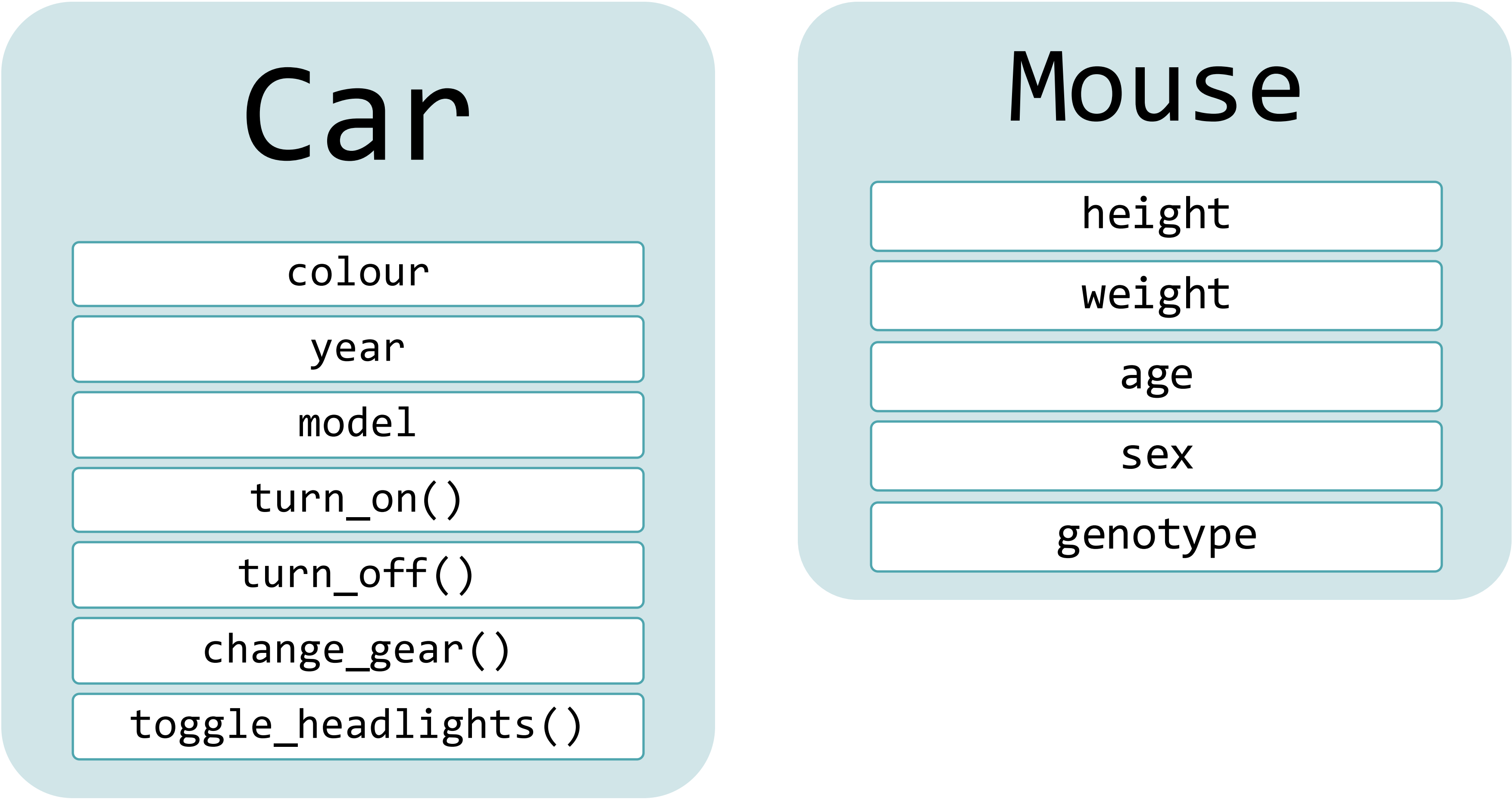
Attributes

- 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!