McGill initiative in Computational Medicine

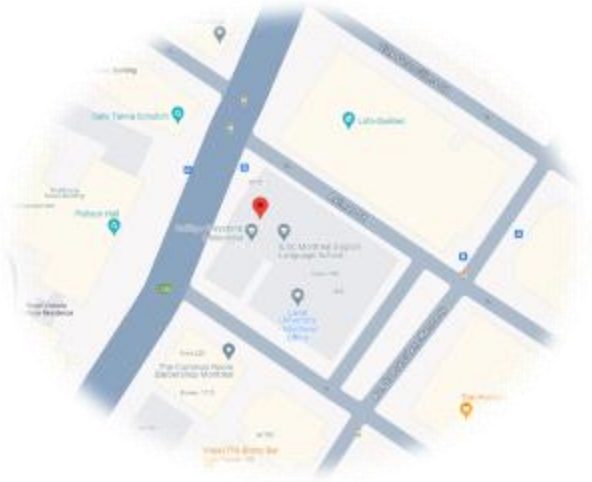
Data Processing in Python

Workshop lead: Benjamin Rudski
November 18, 2025

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

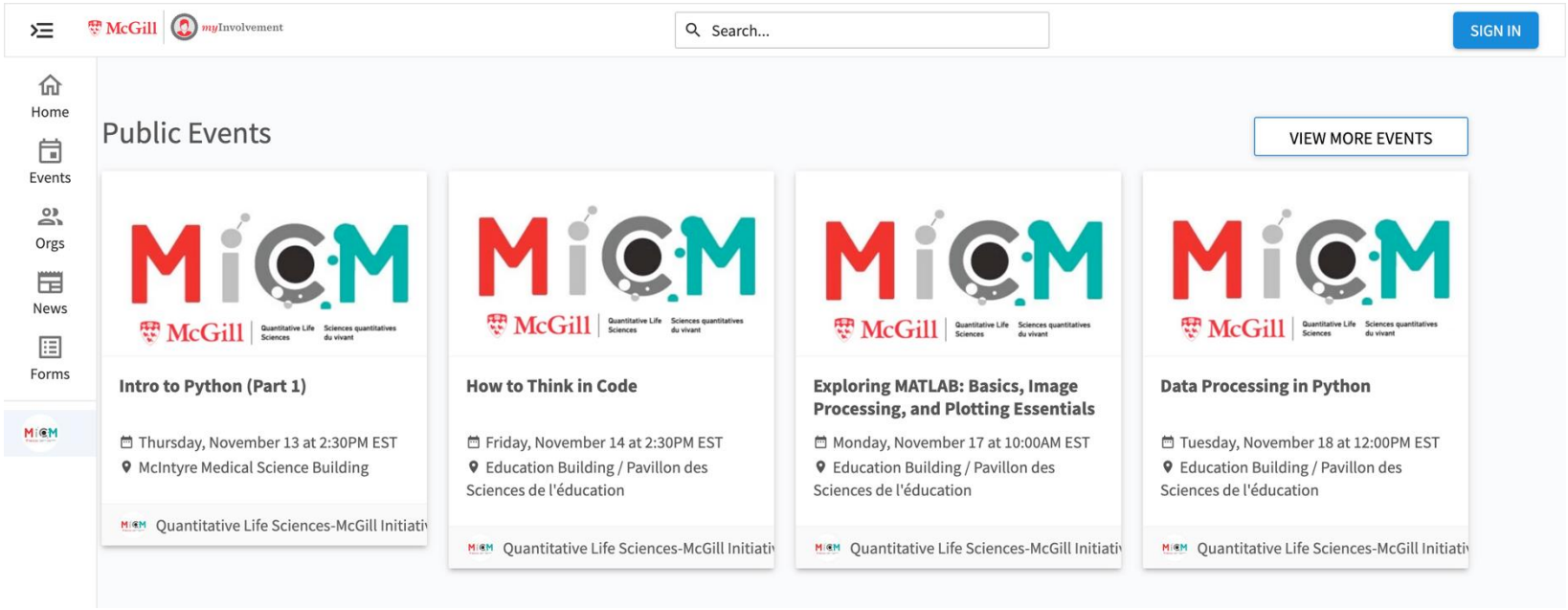
Scan the QR code to sign up for our **mailing list**

Contact: workshop-micm@mcgill.ca

McGill Quantitative Life Sciences | Sciences quantitatives du vivant

McGill initiative in Computational Medicine

Workshop Series



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

https://involvement.mcgill.ca/organization/micm

McGill | Quantitative Life Sciences | Sciences quantitatives du vivant

Learning Outcomes

Summary

In this 4-hour workshop, students will learn basic data processing skills using Python. Attendees will learn how to import code from other modules and packages to take advantage of the existing Python ecosystem. After seeing how to access packages, we will explore popular data analysis packages. We will see how to use NumPy to perform operations on large data arrays and how to use Matplotlib to generate clear data visualisations. We will also scratch the surface on using pandas to store data in tables. Along the way, we will discuss how to approach a new, unfamiliar package and learn how to use it.

Learning Objectives

1. Import code from existing modules and packages.
2. Use NumPy to easily process multidimensional data.
3. Use Matplotlib to generate different types of plots to visualise data.
4. Use pandas to represent data stored in tables.
5. Approach a new package and explore its documentation and examples.

Outline

1. **Module 1 – Modules and Packages (40 minutes)**
    a. Using Modules
    b. A Brief Intro to Package Management
    c. Exercise
2. **Module 2 – Introduction to NumPy Arrays (50 minutes)**
    a. Introducing NumPy
    b. Array Operations
    c. Exercise

Outline

3. **Module 3 – Visualising Data with Matplotlib (50 minutes)**
    a. Creating Plots with Matplotlib
    b. Exploring the Matplotlib Documentation
    c. Exercise
4. **Module 4 – Intro to Tabular Data with Pandas (30 minutes)**
    a. Fundamentals of pandas
    b. Exploring the pandas Documentation
5. **Module 5 – A Brief Guide to Exploring the Unknown (10 minutes)**
    a. What to learn next? How?
    b. How to get help and how not to get help
    c. Other cool programming topics

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
Introduction to NumPy Arrays

`my_array`

(0, 0)	(0, 1)	(0, 2)	(0, 3)	(0, 4)	(0, 5)
(1, 0)	(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)
(2, 0)	(2, 1)	(2, 2)	(2, 3)	(2, 4)	(2, 5)
(3, 0)	(3, 1)	(3, 2)	(3, 3)	(3, 4)	(3, 5)
(4, 0)	(4, 1)	(4, 2)	(4, 3)	(4, 4)	(4, 5)

Shape: (5, 6)

`my_array[0]`?

(0, 0)	(0, 1)	(0, 2)	(0, 3)	(0, 4)	(0, 5)
(1, 0)	(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)
(2, 0)	(2, 1)	(2, 2)	(2, 3)	(2, 4)	(2, 5)
(3, 0)	(3, 1)	(3, 2)	(3, 3)	(3, 4)	(3, 5)
(4, 0)	(4, 1)	(4, 2)	(4, 3)	(4, 4)	(4, 5)

Shape: (5, 6)

`my_array[0]`

(0, 0)	(0, 1)	(0, 2)	(0, 3)	(0, 4)	(0, 5)
(1, 0)	(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)
(2, 0)	(2, 1)	(2, 2)	(2, 3)	(2, 4)	(2, 5)
(3, 0)	(3, 1)	(3, 2)	(3, 3)	(3, 4)	(3, 5)
(4, 0)	(4, 1)	(4, 2)	(4, 3)	(4, 4)	(4, 5)

Shape: (5, 6)

`my_array[:, 0]`?

(0, 0)	(0, 1)	(0, 2)	(0, 3)	(0, 4)	(0, 5)
(1, 0)	(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)
(2, 0)	(2, 1)	(2, 2)	(2, 3)	(2, 4)	(2, 5)
(3, 0)	(3, 1)	(3, 2)	(3, 3)	(3, 4)	(3, 5)
(4, 0)	(4, 1)	(4, 2)	(4, 3)	(4, 4)	(4, 5)

Shape: (5, 6)

```
my_array[:, 0]
```

(0, 0)	(0, 1)	(0, 2)	(0, 3)	(0, 4)	(0, 5)
(1, 0)	(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)
(2, 0)	(2, 1)	(2, 2)	(2, 3)	(2, 4)	(2, 5)
(3, 0)	(3, 1)	(3, 2)	(3, 3)	(3, 4)	(3, 5)
(4, 0)	(4, 1)	(4, 2)	(4, 3)	(4, 4)	(4, 5)

Shape: (5, 6)

`my_array[1:3, 2:4]`?

(0, 0)	(0, 1)	(0, 2)	(0, 3)	(0, 4)	(0, 5)
(1, 0)	(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)
(2, 0)	(2, 1)	(2, 2)	(2, 3)	(2, 4)	(2, 5)
(3, 0)	(3, 1)	(3, 2)	(3, 3)	(3, 4)	(3, 5)
(4, 0)	(4, 1)	(4, 2)	(4, 3)	(4, 4)	(4, 5)

Shape: (5, 6)

`my_array[1:3, 2:4]`

(0, 0)	(0, 1)	(0, 2)	(0, 3)	(0, 4)	(0, 5)
(1, 0)	(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)
(2, 0)	(2, 1)	(2, 2)	(2, 3)	(2, 4)	(2, 5)
(3, 0)	(3, 1)	(3, 2)	(3, 3)	(3, 4)	(3, 5)
(4, 0)	(4, 1)	(4, 2)	(4, 3)	(4, 4)	(4, 5)

Shape: (5, 6)

`my_array[0:5:2]`?

(0, 0)	(0, 1)	(0, 2)	(0, 3)	(0, 4)	(0, 5)
(1, 0)	(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)
(2, 0)	(2, 1)	(2, 2)	(2, 3)	(2, 4)	(2, 5)
(3, 0)	(3, 1)	(3, 2)	(3, 3)	(3, 4)	(3, 5)
(4, 0)	(4, 1)	(4, 2)	(4, 3)	(4, 4)	(4, 5)

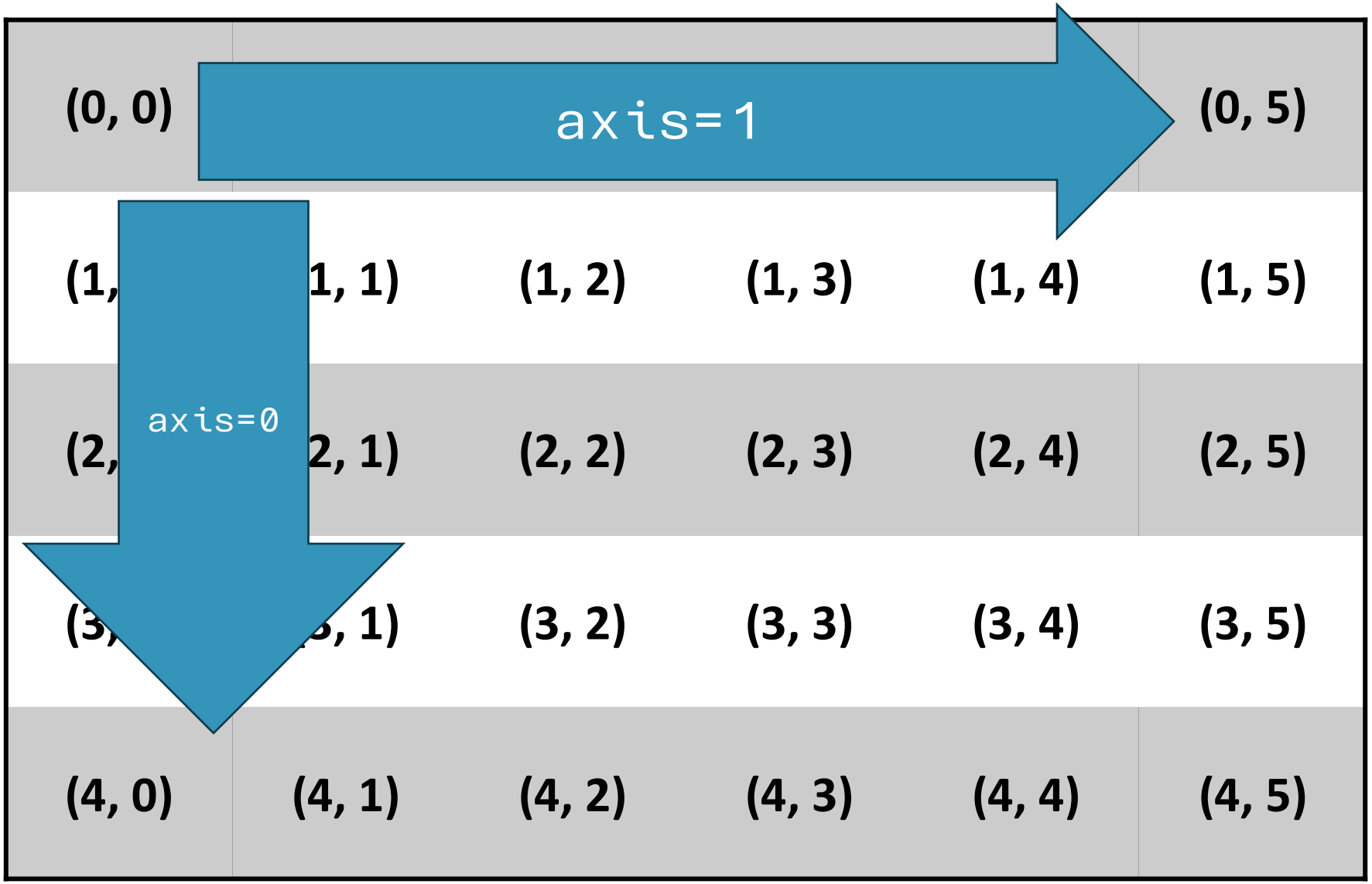
Shape: (5, 6)

`my_array[0:5:2]`

(0, 0)	(0, 1)	(0, 2)	(0, 3)	(0, 4)	(0, 5)
(1, 0)	(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)
(2, 0)	(2, 1)	(2, 2)	(2, 3)	(2, 4)	(2, 5)
(3, 0)	(3, 1)	(3, 2)	(3, 3)	(3, 4)	(3, 5)
(4, 0)	(4, 1)	(4, 2)	(4, 3)	(4, 4)	(4, 5)

Shape: (5, 6)

my_array



Shape: (5, 6)

Module 4

Intro to Tabular Data with pandas

By name: `my_df.loc`

	“series 0”	“series 1”	“series 2”
“row 0”	[“row 0”, “series 0”]	[“row 0”, “series 1”]	[“row 0”, “series 2”]
“row 1”	[“row 1”, “series 0”]	[“row 1”, “series 1”]	[“row 1”, “series 2”]
“row 2”	[“row 2”, “series 0”]	[“row 2”, “series 1”]	[“row 2”, “series 2”]

By integer number: `my_df.iloc`

	“series 0”	“series 1”	“series 2”
“row 0”	[0, 0]	[0, 1]	[0, 2]
“row 1”	[1, 0]	[1, 1]	[1, 2]
“row 2”	[2, 0]	[2, 1]	[2, 2]

To summarize

- ✓ **Modules** and **packages** allow for code written by others to be easily imported and reused.
- ✓ **NumPy arrays** allow easily storing many numbers and performing operations without having to loop.
- ✓ **Matplotlib** can be used to generate many different types of plots.
- ✓ **pandas DataFrames** represent data in tables.
- ✓ Big projects have **documentation** to explain their functionality.

Now you are ready to:

- Import code from existing modules and packages.
- Use NumPy to easily process multidimensional data.
- Use Matplotlib to generate different types of plots to visualise data.
- Approach a new package and explore its documentation and examples.

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, NumPy, Matplotlib and pandas communities!