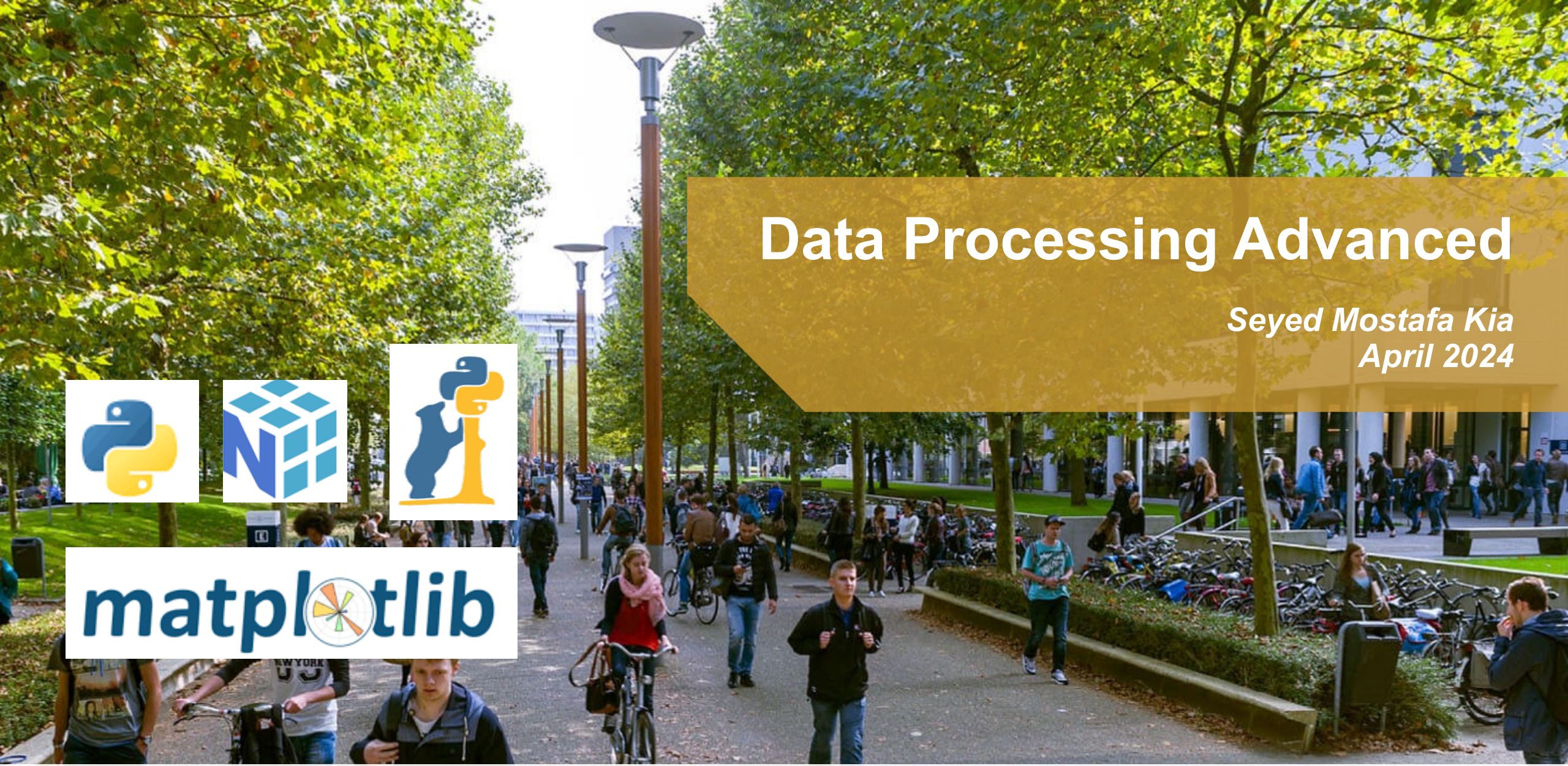


Data Processing Advanced

Seyed Mostafa Kia
April 2024

Data Processing Advanced

- Prerequisites: Familiarity with Python, and Data Processing course
- We will learn about Python tools for scientific computing:

- Learning goals:
  - Write functions to analyze data using the NumPy and Pandas
  - Implement vectorized computations to efficiently process large datasets
  - Explain the advantages/disadvantages of manipulating data in Python, NumPy, and Pandas
  - Generate visualizations of data and results

Course Structure and Routines

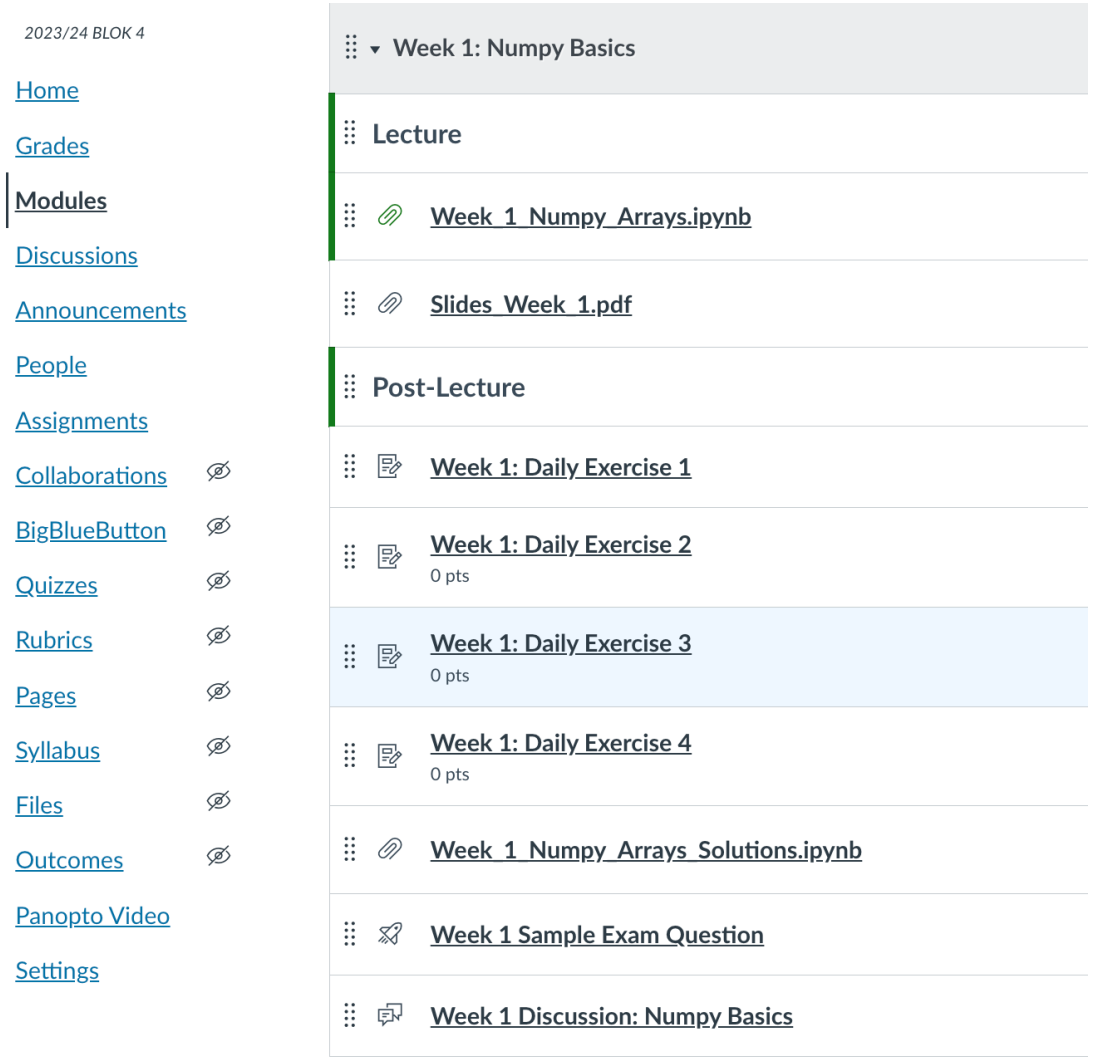
- Seven (1 hour 45 minutes) lectures:
  - Tuesday, April 9, 12:45 - 14:30 (SZ 31): Numpy Basics
  - Tuesday, April 16, 12:45 - 14:30 (SZ 31): Numpy Multi-Dimensional Arrays
  - Tuesday, April 23, 12:45 - 14:30 (SZ 31): Numpy Vectorized and Linear Algebra Operations
  - Tuesday, April 30, 12:45 - 14:30 (SZ 31): Pandas: Basic Topics
  - Tuesday, May 7, 12:45 - 14:30 (SZ 31): Pandas: Advanced Topics
  - Tuesday, May 14, 12:45 - 14:30 (SZ 31): Data Visualization using Matplotlib
  - Tuesday, May 21, 12:45 - 14:30 (SZ 31): Data Processing Tools in Action
- TBD: Exam training lab session
- Please regularly check the updates at https://rooster.uvt.nl/schedule (MyTimeTable)

Course Structure and Routines

- This is a research skill course: to excel in programming, you must train rigorously in the lab.
- Practical Sessions
  - Every Wednesday, Thursday, and Friday in Cube 242
  - More information is available in Canvas.
  - Please regularly check the updates at https://rooster.uvt.nl/schedule (MyTimeTable)

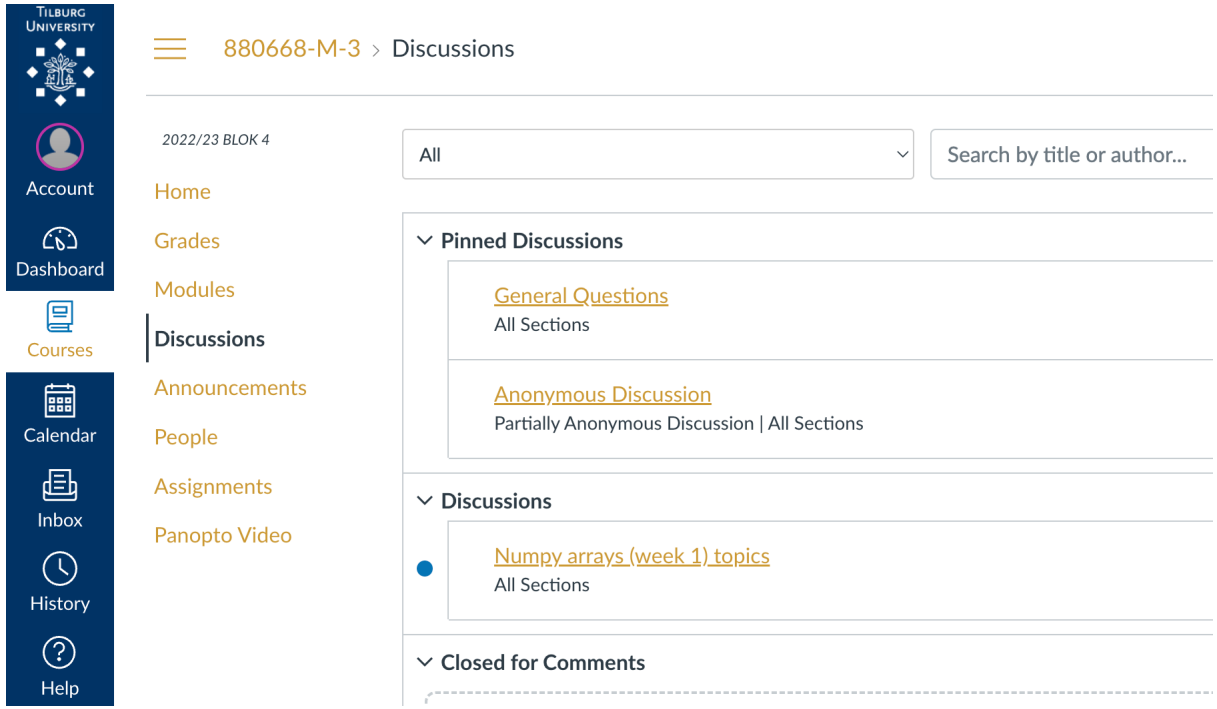
Course Structure and Routines

- Canvas: course materials and communications
  - Modules:
    - Lecture Materials:
      - Jupyter notebooks
      - Course slides
      - Lecture recordings
    - Post-Lecture Materials:
      - Daily exercises
      - Sample exam questions
      - Solutions
  - Discussions:
    - General questions
    - Anonymous discussion

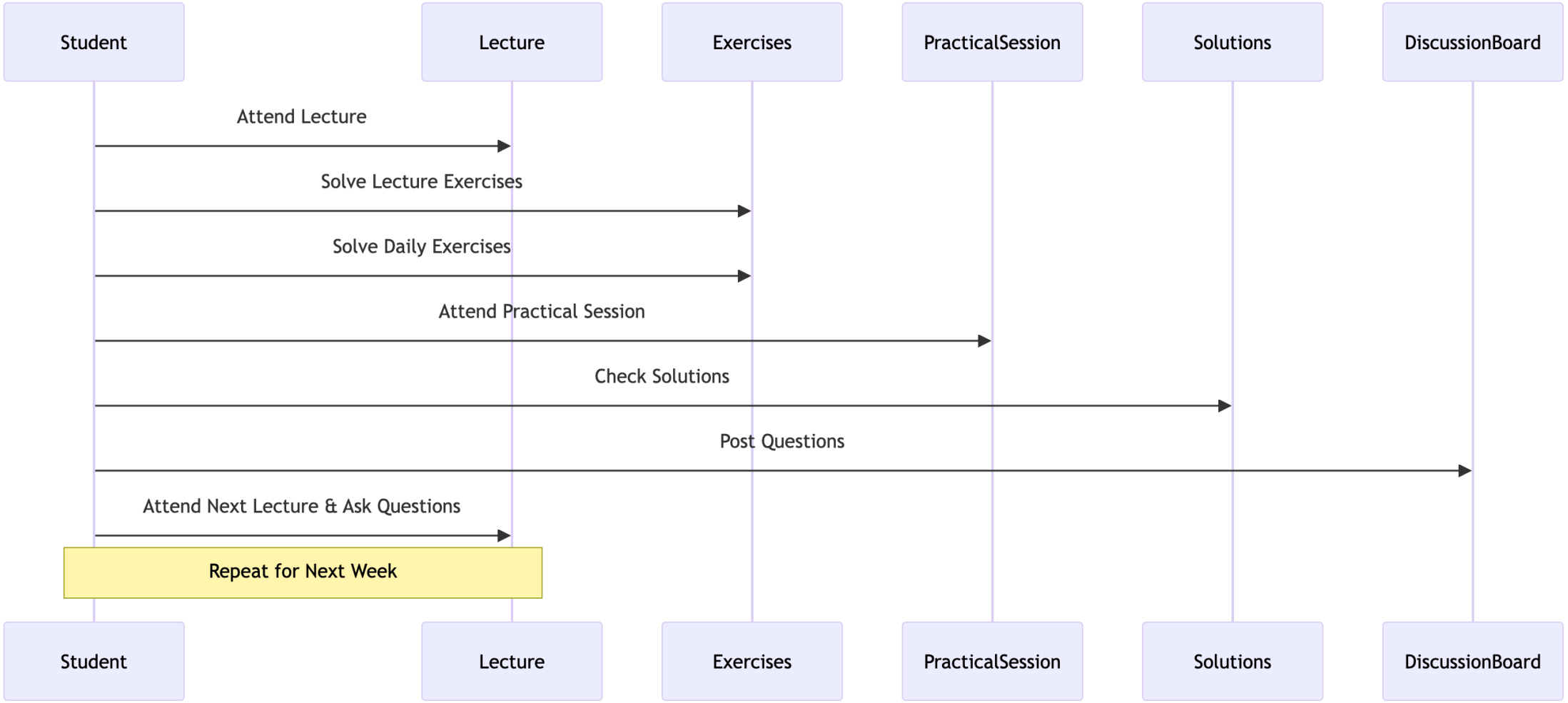


Course Structure and Routines

- Canvas: course materials and communications
  - Modules:
    - Lecture Materials:
      - Jupyter notebooks
      - Course slides
      - Lecture recordings
    - Post-Lecture Materials:
      - Daily exercises
      - Sample exam questions
      - Solutions
  - Discussions:
    - General questions
    - Anonymous discussion



Students' Weekly Activities



Practical Info on Coding

- We will use Python 3 and JupyterLab during the lectures, practical sessions and exam:
  - You can use uvt JupyterLab server: https://jupyter.uvt.nl/
  - You can install Anaconda on your local computer.

- You can also use other IDEs:

Evaluation and Exam

- Evaluation: 100% final exam.
- The final exam (and resit) is an on-campus programming exam.
- The exam will be organized via TestVision.
- The questions are mostly around programming tasks, i.e., writing a Python function for a specific task.
- Students can use all course materials with limited access to the internet.
- We will organize a training exam session before the final exam.
- Exam dates:
  - Final exam: June 13, 2024.
  - Resit: July 11, 2024.

Grading

- Exams will be graded automatically using a Python program.
- Each question will be worth in total 4 points:
  - 0 points for submissions with errors in any of test cases.
  - 0 points for submissions that return the incorrect object type in any of test cases.
  - 0 points for submissions that do not follow the instructions.
  - 1 point for returning the correct object type without any errors.
  - 2 points for passing some tests but not all (without error in all testcases).
  - 4 points for passing all tests.
- You need a score of 55% or higher to pass the exam.

Questions?

NumPy Basics: Arrays and Datatypes

Tilburg University

Understanding Society

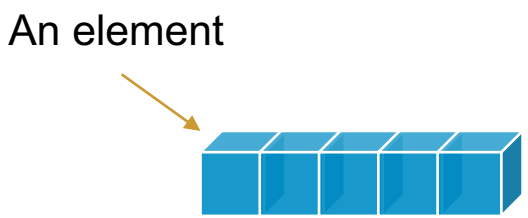
What is NumPy?

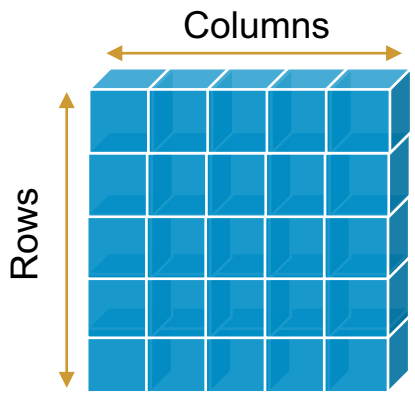
NumPy is the fundamental package for scientific computing in Python.

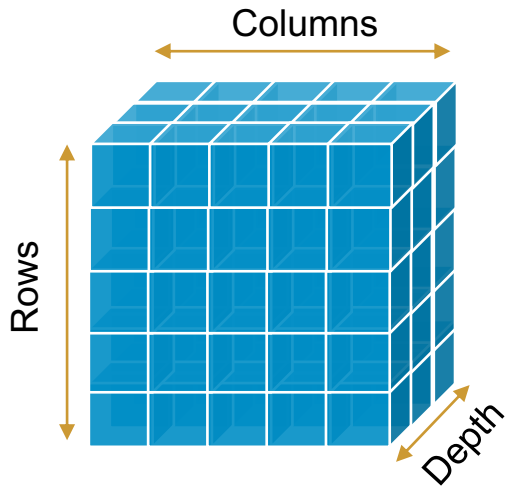
- NumPy provides the possibility for fast operations on multi-dimensional arrays:
  - Statistical, mathematical, and logical operations
  - Shape manipulation
  - Sorting
  - Basic linear algebra
  - Random data simulation

What is an array?

- Multi-dimensional arrays (ndarrays) are basic data structures in NumPy.
- A NumPy array is a structured collection of elements with the same datatype.
- Arrays are defined by the number of dimensions and the number of elements along each dimension.
  - 1-dimensional array (vector): e.g., stock price
  - 2-dimensional array (matrix): e.g., a table of data
  - N-dimensional array (tensor): e.g., an RGB image



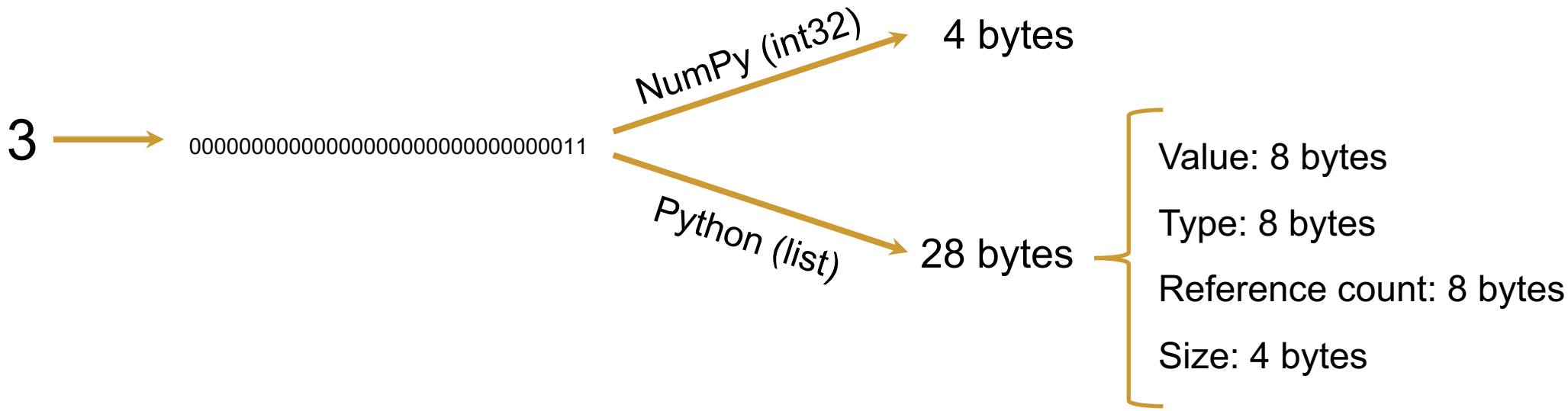




Why NumPy arrays?

Because they are very efficient (memory space and processing time) in numerical operations.

- Unlike lists in Python, NumPy uses fixed types for arrays.

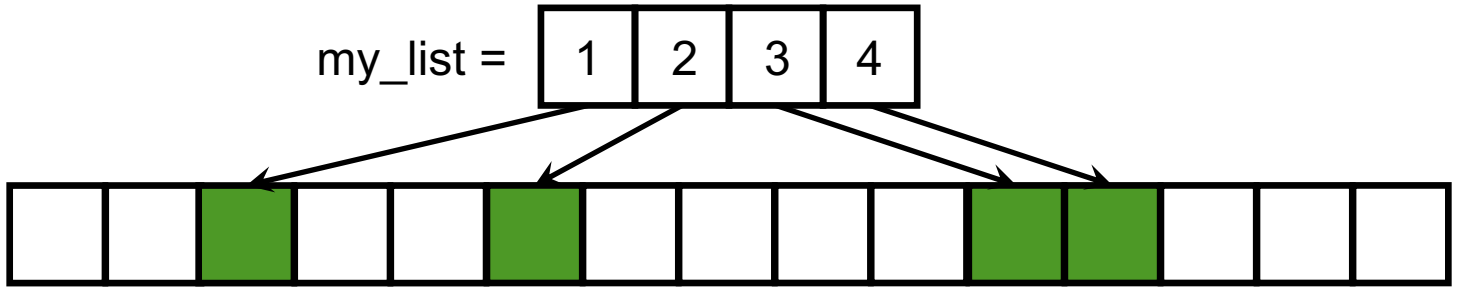


Why NumPy arrays?

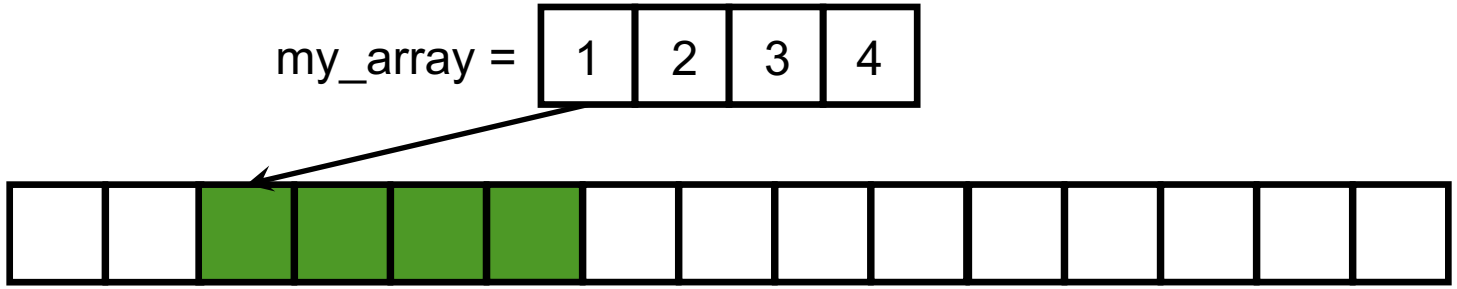
Numpy is very efficient in numerical operations on large arrays.

- Unlike lists in Python, NumPy uses contiguous memory schemes for storing arrays in memory.

Python Lists:



NumPy Array:



Creating NumPy Arrays

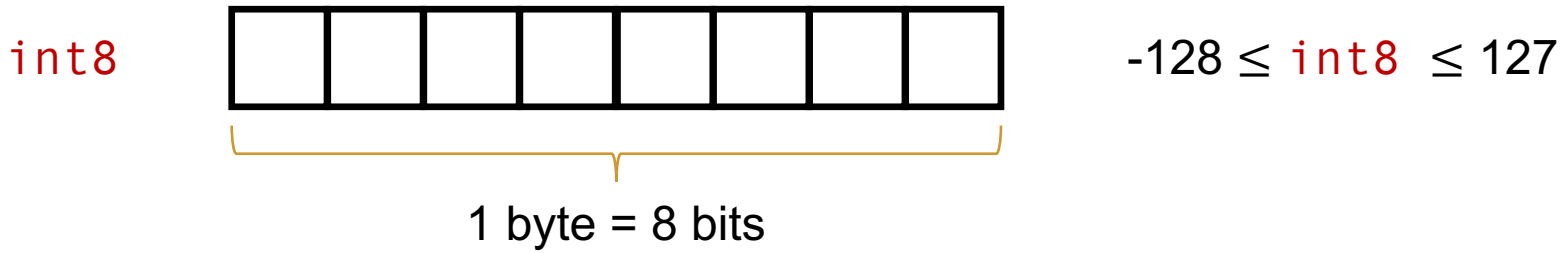
1) From a Python lists or tuples:
   - `numpy.array`
2) From a text file:
   - `numpy.loadtxt`
3) Using intrinsic NumPy array creation functions:
   - `numpy.arange`
   - `numpy.linspace`
   - `numpy.zeros`
   - `numpy.ones`
4) Using random number generation functions
   - `numpy.random.random`
   - `numpy.random.randint`
   - `numpy.random.uniform, numpy.random.normal`

Questions?

Data Types in NumPy

The 6 basic data types in Numpy arrays are:

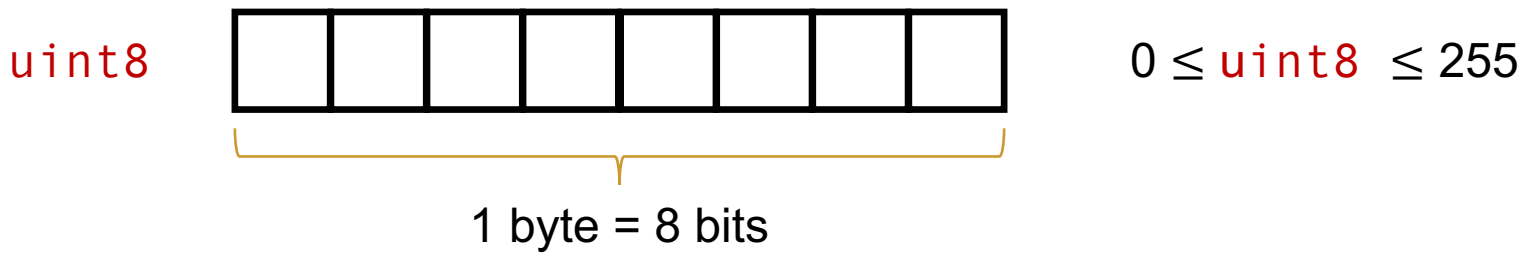
1) integer (`int8`, `int16`, `int32`, or `int64`): is used to store integers between -2^{n-1} and $2^{n-1}-1$ where n is the number of bits.



Data Types in NumPy

The 6 basic data types of Numpy arrays are:

1) integer (`int8`, `int16`, `int32`, or `int64`): is used to store integers between -2^{n-1} and $2^{n-1}-1$ where n is the number of bits.
2) unsigned integer (`uint8`, `uint16`, `uint32`, or `uint64`) : is used to store non-negative integers between 0 and 2^n-1.



Data Types in NumPy

The 6 basic data types of Numpy arrays are:

1) integer (`int8, int16, int32`, or `int64`): is used to store integers between -2^{n-1} and $2^{n-1}-1$ where n is the number of bits.
2) unsigned integer (`uint8`, `uint16`, `uint32`, or `uint64`) : is used to store non-negative integers between 0 and 2^n-1.
3) float (`float16, float32,` or `float64`): is used to store real numbers.

Data types in NumPy

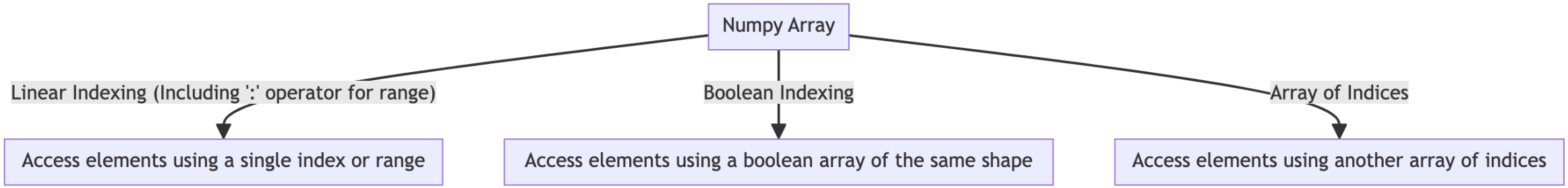
The 6 basic data types of Numpy arrays are:

1) integer (`int8`, `int16`, `int32`, or `int64`): is used to store integers between -2^{n-1} and $2^{n-1}-1$ where n is the number of bits.
2) unsigned integer (`uint8`, `uint16`, `uint32`, or `uint64`) : is used to store non-negative integers between 0 and 2^n-1.
3) float (`float16`, `float32`, or `float64`): is used to store real numbers.
4) boolean (`bool`): `True` and `False`
5) complex (`complex64` or `complex128`)
6) String: for example `<U3` or `<U64`, where the number indicates the maximum length of the strings

Questions?

Indexing Arrays in NumPy

- Linear indexing
- Boolean Indexing
- Indexing with an array of indices



Questions?

Thanks!