Introduction to Python

CMPT 211: Winter 2026

Bernal Manzanilla

Concordia University of Edmonton

January 13, 2026

What is Python?

- **High-level Language:** Python abstracts away complex details of the computer (memory management, pointers).
- **Interpreted Language:**
  - Unlike C or C++, Python code is not compiled directly to machine code before execution.
  - It is translated into *bytecode* which is then executed by the Python Virtual Machine (PVM).
  - This allows for rapid development and interactivity: Read–Eval–Print Loop (REPL).

Performance & Numerical Computing

The "Glue" Language

Python is famous for being excellent "glue" code. It connects different software components effortlessly.

Under the Hood:

- While pure Python can be slower than C++, libraries like **NumPy** and **SciPy** bypass this limitation.
- These libraries run highly optimized numerical routines written in **C** and **FORTRAN**.
- **Result:** You get the ease of Python with the speed of compiled languages for heavy math operations.

Is Python JIT Compiled?

Standard Python (CPython)

The standard version of Python you install is **not** JIT compiled. It is an interpreter.

When is JIT used?

- **PyPy:** An alternative implementation of Python that uses a JIT compiler to run faster.
- **Numba:** A library that allows you to JIT compile specific functions in your code to machine code (e.g., using decorators like `@jit`).

Two Main Paradigms

Python supports multiple ways to structure code. We will focus on two:

1. **Structured Programming (Imperative)**
   - The code is a sequence of instructions.
   - Focuses on control flow: loops, conditions, and functions.
   - *Note: This is the primary paradigm we will use to build logic in this course.*

2. **Object-Oriented Programming (OOP)**
   - Data and functions are bundled together into "Objects".
   - Useful for large systems, but adds complexity.