# Cython

Tamanna Haider
Professor Roualdes
Data Science - MATH 385

# What is Cython?

- Both a compiler and a language
    - Language is an extension of Python that allows putting C types on variables and calling of C functions
    - Optimising static compiler that works for both Cython and Python
        - Tries to minimize or maximize some attributes of a program
- First appeared in 2007

# Benefits of Cython

- “Python at the speed of C”
- Allows for use of C libraries
- Can use your own structs
- Runtime checks for C problems
- Memory management in both Python and C

# Drawbacks of Cython

- Can’t turn all Python code into C
- Can’t translate completely into C
- Fastest when it is only C
- Relatively new

# Cython Documentation

- Cython Documentation: https://cython.readthedocs.io/en/latest/#
- Cython: https://cython.org/#about
- Documentation Help: https://documentation.help/Cython/index.html
- GitHub
- Wikipedia

# How to Use Cython

- Installation Python, C compiler, and Cython
- Use command “pip install Cython” on JupyterLab Terminal
- Use command “%load_ext Cython” to load into Notebook
- Use prefix “%%cython” in the cell
- Use “--annotate” for analysis

# Python vs. Cython

Defining a variable

- x = 10
- myStr = “Hello World”
- c = arr.array('i', [10, 20, 30])

Struct

- No struct for Python

Function

- def functionName():

- cdef int x = 10
- cdef str myStr = “Hello World”
- cdef int[3] c = [10, 20, 30]

- cdef struct Student:

  int id_no

  str name

- cpdef functionName():