Come and join us for the Bio-Physical programs information session:

• Chemistry - Biophysical Chemistry.
• Physics - Biological physics.
• Biology - Quantitative Biology.
• Biology – Mathematics.
• Computer Science – Biology.
• Physiology – Physics.
• Physiology - Mathematics.

September 18, 2017 4:00 PM
Leacock Bldg. room 232. Advisors will be present to answer all your questions. There will be Pizza and drinks.
Outline

1. Midterm, Quizzes.
2. Motivation / Recap
3. A Quick tour of your computer
4. Installing Python
5. Running your first Python program
In computation, the basic unit of information is the bit. The bit gives us certainty about the state of a thing as being either 0 or 1. Combining bits lets us represent more and more complex things.
An example: information flow in Biology and the Central Dogma

DNA sequence → Protein structure → Interaction networks → Cellular communication → Organisms → Ecosystems

Life is information ⇒ Computers!

---

1http://scienceonthesquares.blogspot.ca/
How to look at the world.

There is too much information in the world for any human or computer to be able to handle. So we need **models**.
Quick tour of your computer

I will take you through some of the important places and tools we will use on your computer.

1. Storage → where you store your files
2. CPU → processes commands
3. Operating System → manages communications between components and user. (more on this later)
Getting set up

**Goal:** write computer instructions (programs) to process information about the world.

What you will need:

1. A computer (obviously)
2. A place to write down your instructions. → Text Editor
3. Something to read your instructions and do (execute) them → Python interpreter
Lectures are **not** workshops. Throughout the term we will be doing live demonstrations in class, but if you have questions about your specific code or setup please go to office hours or post on discussion boards.
Writing and storing your instructions: Text Editor

- A programming text editor lets you type in **plain text** so a compiler can easily read the instructions. **DO NOT USE WORD**

- Text editors produce **text files**.

- **Sublime** is a popular one but there are many others you can use. The simpler the better.
What kinds of instructions are there?

\(^2\)Her (2013)
What kinds of instructions are there?

Computers understand a very limited number of instructions...

1. Remember a number → variables
2. Add, Subtract, etc. values → operations
3. Compare two values i.e. equal, greater than, less than → comparison
4. Repeat a command → loops
5. Write out a stored value → printing

... turns out this is pretty much all you need!

The rules for writing down these instructions is what we call a Programming Language
Python: our language of choice

These instructions can be understood by most programming languages. We choose Python.

- Python is a **general purpose** programming language.
- This **language** gets **interpreted** and turned to computer code by an interpreter (there are many.. Cython, PyPy, Jython, etc.)
- Python is **growing very rapidly** in academia and in industry. Some examples...
  - YouTube
  - Reddit
  - NASA
  - Dropbox (Guido’s current employer)
  - Google AI (TensorFlow)
  - Many more
Why is Python so popular?

- Succinct and easy to pick up syntax. (Drawback: ‘slow’)
- Lots of built in functionality (sorting, max, etc.)
- The holy trinity: free, open source, community

- Thousands of modules (more on this later) for doing many things: databases, visualization, AI, games, math, web development, etc.

Installing your Python interpreter

- We will use the cross-platform Python interpreter from Anaconda (Cython).
- Anaconda installs a local copy of the Python interpreter so you don’t mess up the rest of your computer.
- Install: https://docs.continuum.io/anaconda/install/
- More on the full functionality of Anaconda as we move forward
A crash course on your OS

- The Operating System lets you interact with your file system, software, and the CPU
- In order to run Python programs you need to be familiar with talking to your OS → Terminal/Command Line
- You’re already familiar with interacting with the OS through a visual interface.
The Terminal / Command Line

The terminal is the same thing but with text commands. Makes things a lot more efficient when programming.
Basic OS Commands

- Basic OS commands (*NIX/Windows) dollar sign $ indicates an OS command. Don’t actually type it.
  - $ pwd / dir print current directory
  - $ mkdir [NAME] create a new directory called NAME
  - $ ls / cd list files in current directory
  - $ cd [PATH] navigate to PATH
  - $ python [FILEPATH] execute python script in PATH

- More *NIX commands

- Windows commands
File System

http://itp.nyu.edu/
Hello World: your first program

In this course we will be using two types of python environments: text editor and Jupyter Notebooks.

▶ Hello world from a text editor.

1. Create a directory called COMP_364 and go inside it.
2. Open your favorite text editor (e.g. Sublime)
3. Type: `print("Hello World! This is COMP 364!")`
4. Close the file and save it as `hello_world.py`

▶ Hello world from a Jupyter Notebook

1. `$ jupyter notebook`
2. New → Notebook (Python)
3. Repeat step 3 above
4. Cell → Run Cells

Life Hack 1

Shift + Enter  →  execute current cell
Now that you’re a programmer, here is some very important advice from a fellow programmer.

1. Python file names always end in .py and contain only letters, numbers, or underscores… NO SPACES.
2. When stuck: think about it → RTFM → Google it → ask/post.

More advice to come as we move forward.

Life Hack 2

Keep the **Python style guide** handy when coding.
Data Types

We saw the print command which displays some text to us. So what exactly are the types of information that Python can understand? → data types.

▶ Numbers → 0, 1, 2, 4.5, ...
▶ Strings (always inside single, double, (or triple single/double), quotes) → "this is a string", ‘this is also a string’, """this is called a multi-line string"
▶ Boolean → True or False

Try executing some of these expressions in Python (in a file, or a notebook).

Life Hack 3

You can execute commands interactively in the terminal with $ ipython or $ python no need for saving a file or launching jupyter.
Now that we have some ‘things’ (numbers, text, booleans) we can combine and compare them. Note: >>> means execute a Python command.

- **Addition and subtraction**
  - >>> 1 + 1
  - >>> 0.5 - 1

- **Multiplication and Division**
  - >>> 5 * 100
  - >>> 5/100

- **Modulo (remainder after dividing first number by second)**
  - >>> 5 % 2
  - >>> 10 % 5

- **String concatenation**
  - >>> "Hi my name is" + " Chad"
Comparisons

Python lets us compare two values with the comparison operators `==`, `!=`, `>`, `>=`, `<`, `<=`. The result of evaluating a comparison is a boolean. Give them a try!

- >>> 5 < 3
- >>> "Monday" == "Tuesday"
- >>> 2 >= 2
- >>> "Chad" != "chad"
Boolean Operations

We can also do logical operations with booleans to get a boolean result. The basic operators for this are and, or, not. Since comparisons (previous slide) give us boolens, we can use them to make useful expressions.

- A and B → True if A and B are True
- A or B → True if A or B or both are True
- not A → True if A is False and False if A is True.

Examples:

- >>> (4 < (3 * 2)) and not ((1-4) < 0)
- >>> ((5/2) > 1) or ((5 - 2) < 10)

You can play around with these to make as many combinations as you like. You will learn how useful they are soon enough.
Homework & Next time

- Make sure you have your coding environment set up (text editor, Anaconda, jupyter)
- Play around with Python and the command line.
- Next time: variables, types (more formally), comments, control flow.