Warmup (L21)

- In modular arithmetic, all math is done *modulo* some number (the modulus). For instance, in mod-5 arithmetic, 3+3==1, because 6%5==1.
- For example, time is mod-12 or mod-24: 23:00 plus 2 hours is 1:00
- Create a class `Modular` that contains an int value and modulus, and can do modular arithmetic with other `Modular`s of the same modulus (**`assert`** that the modulus is equal)

Escaping Jupyter's gravity

CS114

Administrata note

- This module is considered outside of the regular course material; that is, it won't be on the exam
- I'm covering it because while Jupyter is usually used for prototyping, Python code is more often run outside of Jupyter
- This module explains how programmers usually use Python
- There are no in-lecture quizzes in this module, but you're encouraged to follow along on your computer

Installing Python

CS114 (MVSC)

What is Python?

- You've all seen what Python (programming language) is, but not what Python (piece of software) is
- To be clearer, I'll call it: the Python interpreter
- The Python interpreter is the software that takes programs in Python and turns them into action

Getting Python

- Linux: You almost certainly already have it
- macOS: Your system has it, but the system version is olde. Install a new one.
- Windows: You don't already have it. Install it.

- Android/iOS: The University of Waterloo provides several labs with grown-up computers

Getting Python

- https://python.org/ , click Downloads
- Windows: Install the Python Install Manager. It will then run and install Python. Default options should be fine.
- macOS: The macOS package installs Python, not an installer. Just run it. Default options should be fine.

What did I just install?

- The Python interpreter is just a program to run Python programs
- Technically, you can run it and interact directly with Python, but we need another program to usefully write Python code
- We need an *IDE*: Integrated Development Environment
- Popular one nowadays: Visual Studio Code

VS Code

CS114 (MVSC)

What is Visual Studio Code

- VS Code styles itself as an “AI code editor”
- It has nothing to do with AI, but we’re currently in a nightmare timeline where pretending otherwise is good branding
- It’s a code editor: a text editor that’s suited particularly to editing code (like Jupyter is) that can also run code

Get VS Code

- https://code.visualstudio.com/
- There's a big "download" button, can't miss it
- Once you install it, it'll go through a setup wizard with many steps, but you can skip all of them and just close the "setup walkthrough" tab

VS Code + Python

CS114 (MVSC)

VSCode extensions

- VSCode installs with the ability to do little other than edit text files
- You need to install an extension for any language you want to use
- Click the “extensions” button on the left, and search for “Python”, then click its “install” button

Setting up the extension

The Python extension can set itself up. You can ignore its “checklist”.

At this point you might want to close all the garbage VSCode has opened on both sides of the screen. The middle section is for editing Python.

Making a Python file

- VSCode is really meant for whole projects with many files, but that's beyond the scope of this course
- Ctrl+N creates a text file
  - You can then change it to Python. See the instructions it shows on screen. (Ctrl+K then M)
- File→New File lets you make any kind of file, including Python (which we want!)

Editing Python

- You'll notice that once you're in Python mode, it highlights Python keywords and such
- Let's start writing a Python program

Editing Python

- You'll notice that once you're in Python mode, it highlights Python keywords and such
- Let's start writing a Python program

```
import numpy
a = np.array([1.1, -1.1, 2.2])
a[a < 0] = 0
print(a)
```

Running Python

- Now let's run the program
- In the menu, Run→Start Debugging (or F5)
  - Running with or without debugging will make little difference for Python. That's more helpful for some other programming languages.
- Oops! No `numpy`!

Managing Python packages

CS114 (MVSC)

VSCode's Python environment

- `numpy` doesn't come with Python
- We need to install it
- VSCode has a ~~needlessly confusing~~ interface to help us do this
- There should be a Python button on the left (Python's logo is two snakes)
- If there's not, we need to get that first

Python Environments

- In their infinite wisdom, Visual Studio Code broke the Python package in two mere days before today's lecture
- You may have gotten the complete version (in which case you have the Python logo) or not

Python Environments

- If you don’t have the Python button:
  - File→Preferences→Settings (or Ctrl+comma)
  - Search for “use environments extension”
  - Enable it
- Now back to our regularly scheduled package management…

Information overload

- The Python screen we just opened has more information than we need
- The first group, “Python Projects” is not needed unless you intend to make multi-file Python projects
- The second group, “Environment Managers”, is what we want

The global environment

- Under environment managers, “global” is your installation of Python
  - Other environments are for specific Python projects; we won’t need them
- You’ll see under “global”, e.g., “Python 3.14.0”
- To the right of the Python version is a box “Manage packages”. Click it.

The package manager

- It's popped up a list of all the packages available for Python
- Type "numpy", click the checkbox next to it, then click "apply" to install!
- Let's install matplotlib too
- `numpy` will work now, but you should probably close and re-open VSCode so it finds it

Aside: Running

- When you run, VSCode may ask you *how* to run the file
- That just means it hasn't recognized that this is a Python file
- Just select "Python" and it'll run

Let’s run it again

- If you run the program (now that numpy is installed), you’ll get the expected output below
- The tab is called “terminal”
- This is actually a text-based environment for interacting with your computer
- VSCode will run the commands for you; you don’t need to learn the terminal!

Code and files

CS114 (MVSC)

The editor

- Once you have a language set up, VSCode *loves* to scream at you
- It'll tell you types, it'll tell you functions available, it'll tell you if you make certain errors
- Don't get overwhelmed: you can always just type!

Saving

- When you save, you (of course) choose where to save
- As long as you chose Python, VSCode will suggest naming your file with .py
- This is the same file type as the “executable script” that Jupyter outputs!

Saving

- Your Python program will run where you saved it
- When a program is run, it has a *working directory*, which is… the directory it's working in
  - Directories are also known as "folders"
- That has implications for reading/writing files

Accessing files

- When you read a file, it will look next to your .py file
- Same when you write a file
- So, make sure your data is where your Python code expects
- Perfectly common to have all your data and your Python programs in the same directory

Accessing files

- Let’s read and print out CSV file in Python, now running locally

Jupyter and VSCode

CS114 (MVSC)

Python is imperative!

- A normal Python file runs in order from top to bottom
- Jupyter *mostly* works like that… but you can run the cells in any order
- A subtle further implication: if one cell crashes, you can still run the next one
- A normal Python file is basically a single Jupyter cell

Jupyter to Python

- We've been converting Jupyter files to Python scripts all along; they're what you submit to MarkUs!
- File→Save and Export Notebook As→Executable Script
- You can open, edit, and run that file in VSCode

Python to Jupyter

- You can upload a .py file to Jupyter…
- but you can’t do much with it.
- Jupyter’s really meant for notebooks
- If you wanted your Python in a notebook, just copy-paste it in

General fun with code

CS114 (MVSC)

It’s Python!

- Now that we have VSCode set up, we can do all the Python stuff we did before
- Let’s remake our StatsList class, read in a CSV file, and make a StatsList file for each column

Course Perceptions Survey

https://perceptions.uwaterloo.ca/

The remaining time today is free for you to fill in your (CS114) course perceptions surveys