SURGE

Science Unleashed

Discover Python

Welcome to Discover Python!

Introducing our teaching team:

- Aaron Newman
- Jordan Gardiner
- Murwan Bashir
- Danny Godfrey
- Simon Leger
- Hayam Mahmoud-Ahmed

Equity, Diversity, & Inclusion

SURGE is a safe space where everyone should feel welcome and included, free to pursue opportunities, and free express divergent opinions in the spirit of productive academic exchange.

Recognition of Mi’kmaq Territory

Dalhousie University is located in Mi’kma’ki, the ancestral and unceded territory of the Mi’kmaq. We are all Treaty people.

Code of Conduct

In addition to Dalhousie's Code of Student Conduct, in Discover Coding we employ the Carpentries' Code of Conduct:

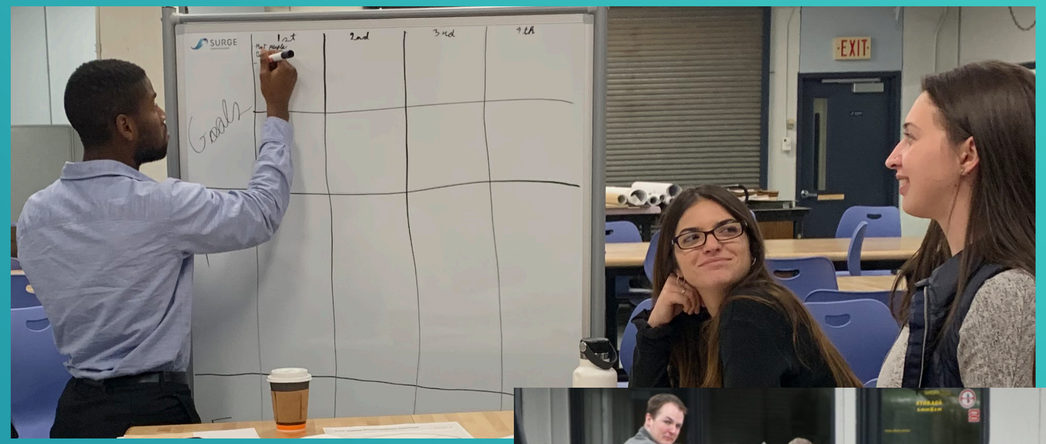
- Use welcoming and inclusive language
- Be respectful of different viewpoints and experiences
- Gracefully accept constructive criticism
- Focus on what is best for the community
- Show courtesy and respect towards other community members

https://docs.carpentries.org/topic_folders/policies/code-of-conduct.html

SURGE

What is SURGE?

- One of the NS innovation sandboxes
- Provides experiences in:
  - Applying science to real world problems
  - Creativity, innovation, design thinking
  - Entrepreneurship
  - Leadership

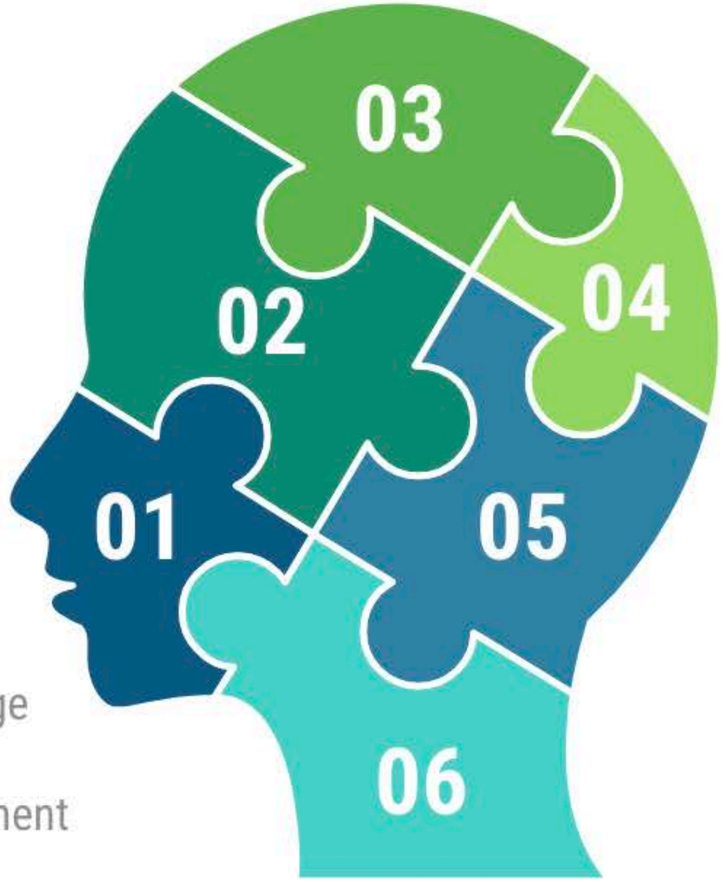


What We Teach

Problem definition
Needs finding
Customer discovery
Business models

Finding a Solution
Human-centered design
Ideation
Brainstorming

Technical understanding
Scientific & technical knowledge
Intellectual property
Technology readiness assessment



Soft Skills
Working in teams
Networking
Professionalism

Personal development
Self-driven learning
Pitching an idea

Business knowledge
Corporate ethics
Corporate structuring
Finance

SURGE
Science Unleashed
Science Unleashed

About Discover Python

Discover Python: Our Approach

Goal: teach people working in science how to use Python as a tool for working with data:

- learning the fundamentals of Python
- learning the fundamentals of programming logic
- using Python for data science, including:
  - reading data
  - manipulating/processing data (e.g., extracting specific data, splitting data according to variables, applying functions, combining data)
  - exploratory data analysis
  - basic statistical analyses of data sets

What is data science?

- “…an umbrella term to describe the entire complex and multistep processes used to extract value from data.” (Wing, 2019)
- The ability to “bring structure to large quantities of formless data and make analysis possible” (Davenport & Patil, 2012, p.73)
- Storage, manipulation, visualization, filtering, and preparation of data, as well as statistics to derive conclusions from existing data, and machine learning to make predictions from data that will generalize to other data
- Also the “back end”: engineering, hardware, databases to support data science

Learning Objectives

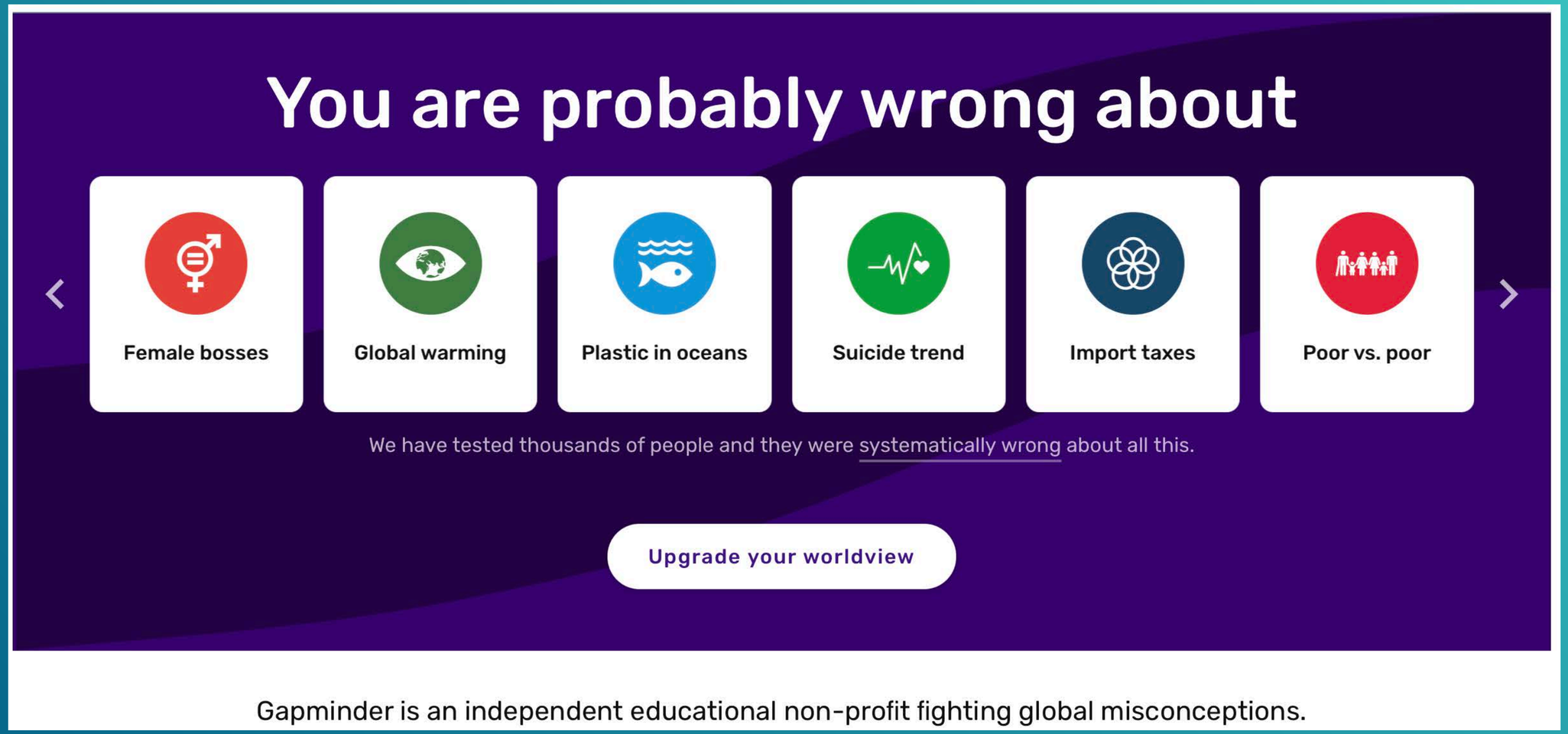
Upon completing this workshop, you will be able to:

- understand and use variables
- work with common Python data types like integers, floats, strings, characters, lists, dictionaries, as well as NumPy arrays and Pandas DataFrames.
- read data from text files
- obtain basic summary statistics from data files
- manipulate and extract data from pandas DataFrames
- visualize data using Python's Matplotlib package, and customize these plots
- use basic flow control, including for loops and conditionals
- write Python code according to standard style guidelines

Origins

- **The Carpentries** (carpentries.org) teaches foundational coding, and data science skills to researchers worldwide.
  - Provide an open-source set of workshops, under Software Carpentry, Data Carpentry, and Library Carpentry
  - A diverse, global community that includes Instructors, helpers, Trainers, Maintainers, Mentors, community champions, member organisations, supporters, workshop organisers, staff and more
- **Discover Python** is based on Software Carpentries' **Plotting and Programming in Python** workshop
  - Adapted for our platform and based on our experience
  - Uses open-source Gapminder data

Gapminder.org




Share
Add to list
Like
Recommend
Hans Rosling | TED2006
The best stats you've ever seen
19:37

Gapminder data in this workshop

- GDP (gross domestic product) for each country in the world, over a >100 year period
- Life expectancy data by country
- Open source, accessible, relatively easy to understand

Tools

- Open-source programming language
- First released in 1991 by Guido Van Rossum
- Named after Monty Python, not snakes
- Under continuous development by large community
- Relatively “high level” compared to some common languages like C or Java
  - Simpler and more elegant to write and read
- Widely used in science and data science

TIOBE Programming Community Index

- Indicator of the popularity of programming languages.
- Ratings based on the number of skilled engineers world-wide, courses and third party vendors.
- Popular search engines such as Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube and Baidu are used to calculate the ratings
- https://www.tiobe.com/tiobe-index/

Feb 2021	Feb 2020	Change	Programming Language	Ratings
1	2	↑	C	16.34%
2	1	↓	Java	11.29%
3	3		Python	10.86%
4	4		C++	6.88%
5	5		C#	4.44%
6	6		Visual Basic	4.33%
7	7		JavaScript	2.27%
8	8		PHP	1.75%
9	9		SQL	1.72%
10	12	↑	Assembly language	1.65%
11	13	↑	R	1.56%
12	26	⇈	Groovy	1.50%
13	11	↓	Go	1.28%
14	15	↑	Ruby	1.23%
15	10	⇊	Swift	1.13%
16	16		MATLAB	1.06%
17	18	↑	Delphi/Object Pascal	1.02%
18	22	⇈	Classic Visual Basic	1.01%
19	19		Perl	0.93%
20	20		Objective-C	0.89%

- A “notebook” environment for data science
- Code, output, and commentary all in one document
- Excellent for reproducible, open science

jupyter **Notebook Example** Last Checkpoint: a few seconds ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code nbdiff

This is a Jupyter notebook file

This first cell is written in **markdown**, which allows *rich text formatting* including:

- bullet points
- numbered bullets

Headers

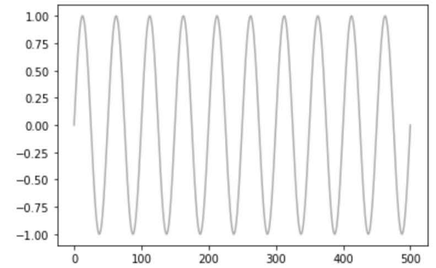
And even pictures of cats!

Below this cell is a Python code cell:

In [1]:
```
print("Hello world!")
```
```
Hello world!
```

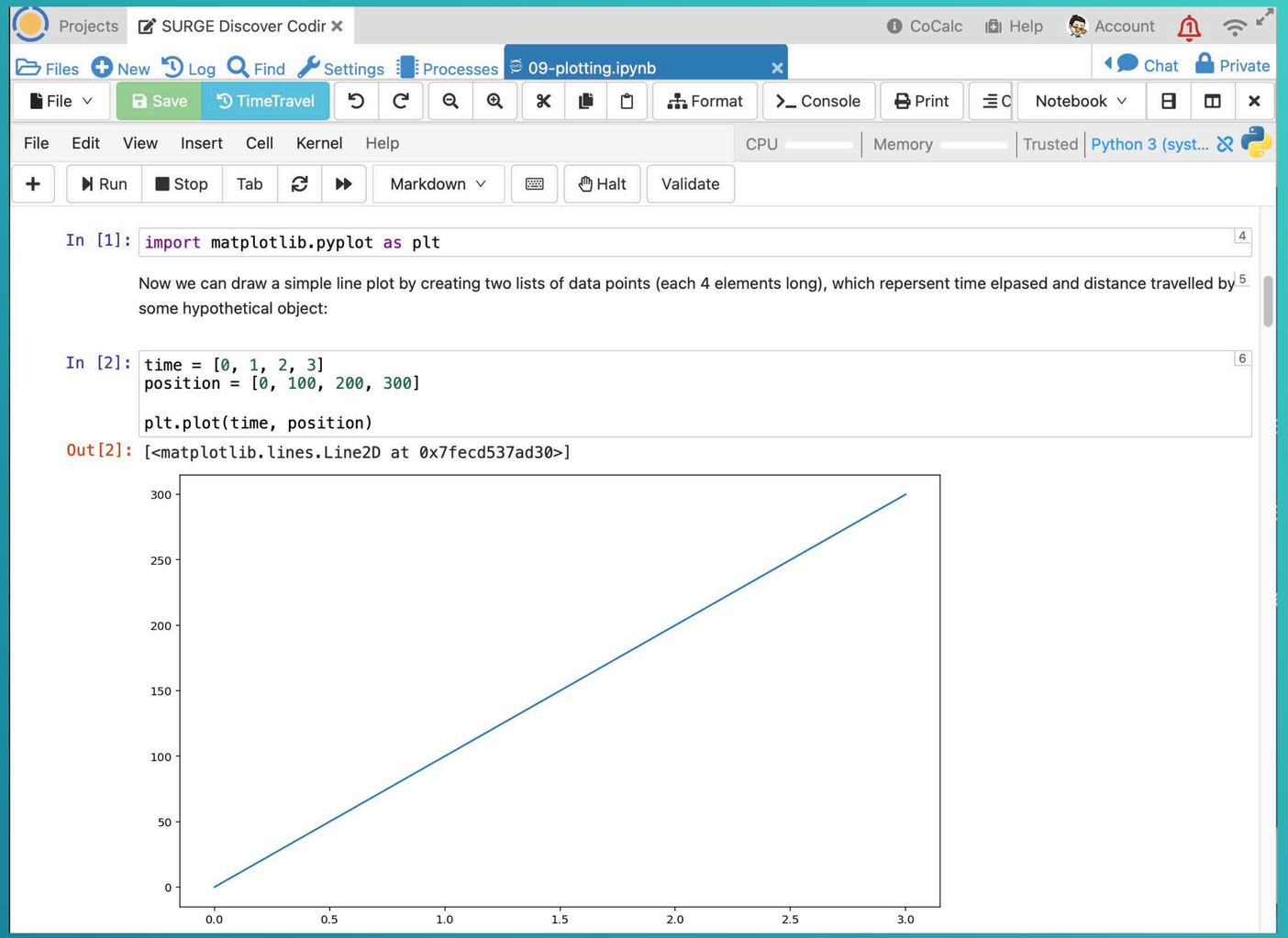
Here's some fancier code

In [8]:
```
import numpy as np
import matplotlib
import matplotlib.pyplot as plt

srate1 = 500
time_samp = 1 # sec
alpha = 10 # Hz

x1 = np.linspace(0, srate1, srate1*time_samp)

y1 = np.sin(alpha * 2*np.pi * x1 /srate1)

plt.plot(x1,y1, color='.67');
```



In [ ]:

- Cloud-based Jupyter environment
- Nothing to install
- Designed for teaching
- Teaching team and “jump in” and view your work to provide help



Workshop Mechanics

Type code yourself

- Don’t cut and paste
- Learning is deeper if you type the code
  - Learn from your mistakes
  - Parse/analyze the code better

Getting Help

- **Helpers** are here to... help!
  - Use the “no” reaction in Zoom to request help
    - A helper will send you a direct message in the Zoom chat
    - They can start a breakout room to talk to you, and/or look at your work on CoCalc
- We will do regular **check-ins**
- **Ask the instructor**
  - Use the helpers for help with your code
  - Ask the instructor content/theory/conceptual questions
    - Raise hand (Zoom reaction)
    - Ask during check-ins