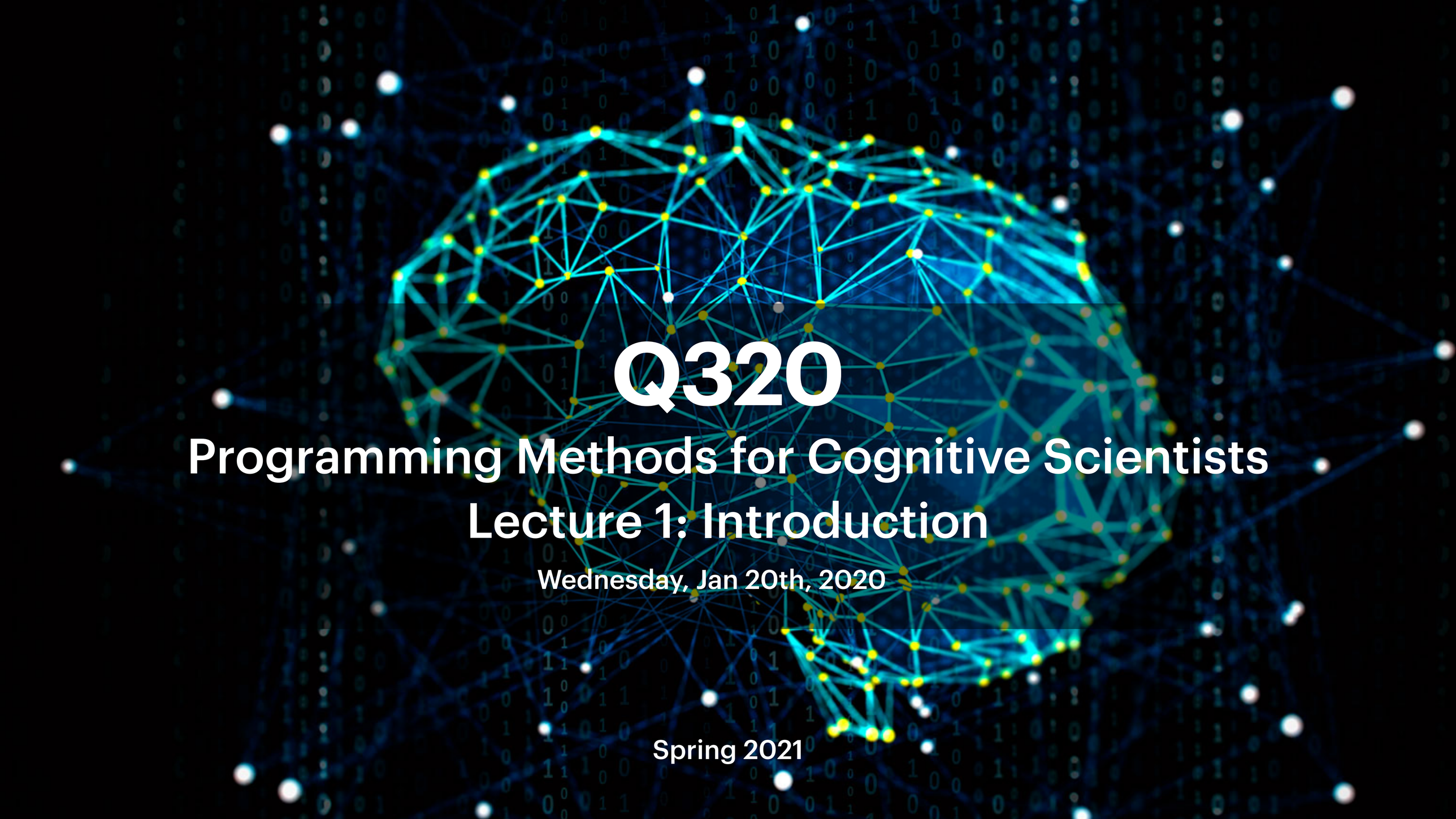


Q320

Programming Methods for Cognitive Scientists

Lecture 1: Introduction

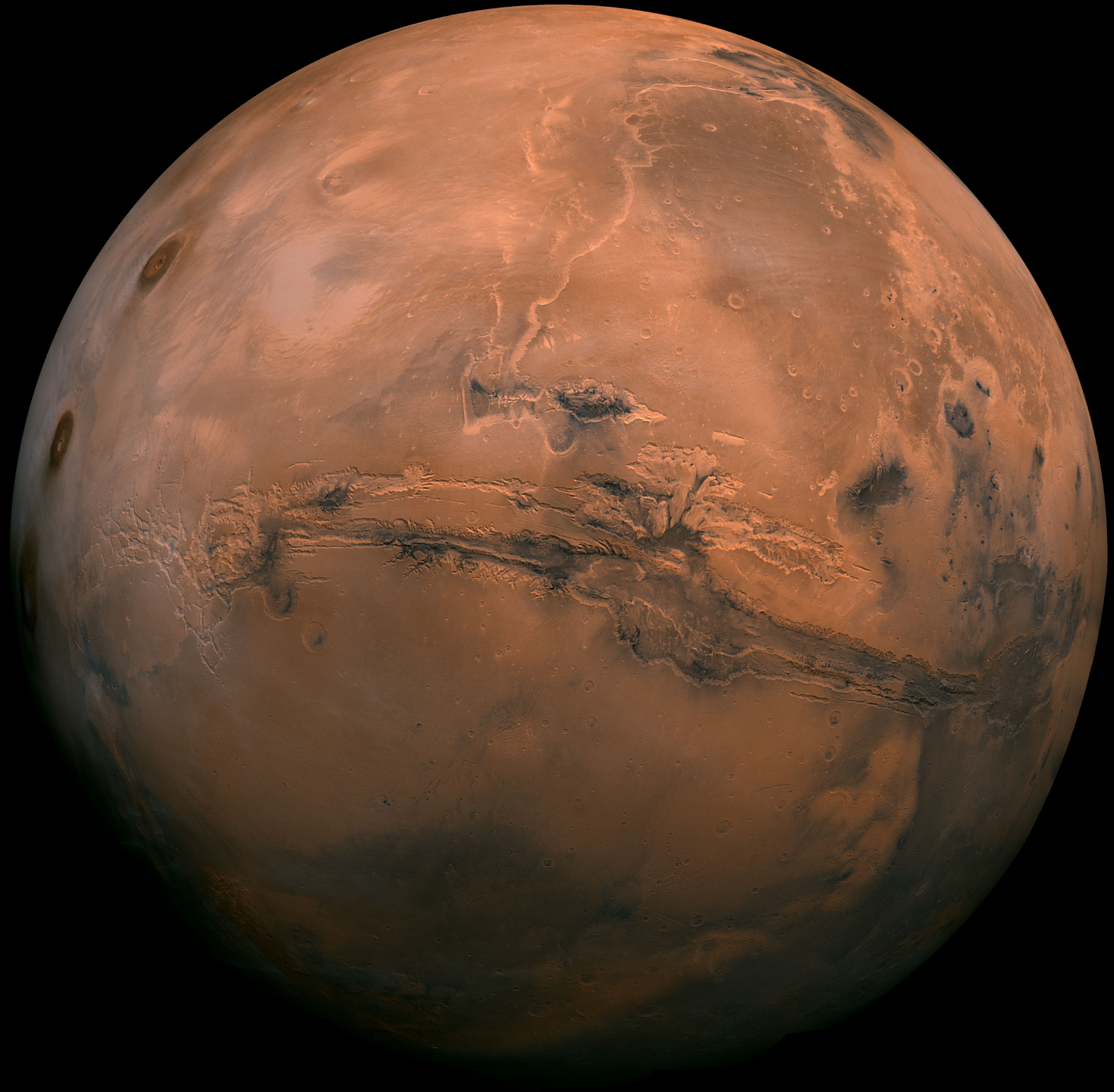
Wednesday, Jan 20th, 2020

Spring 2021

You land on Mars and find a strange “object.”

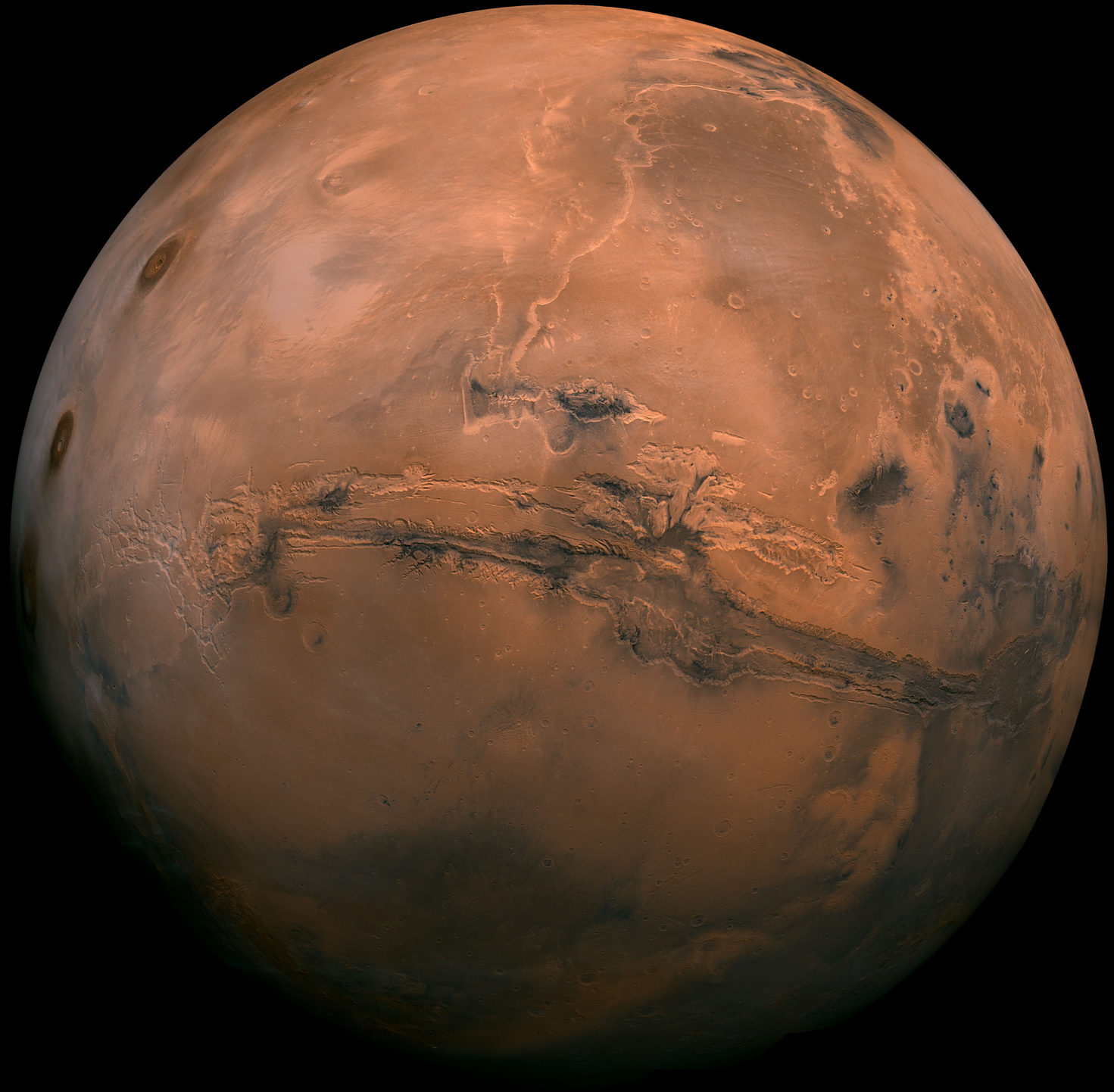
You have to decide whether it is alive or not, and whether it is cognitive or not.

Briefly state the criteria you will use for each (2-3min).



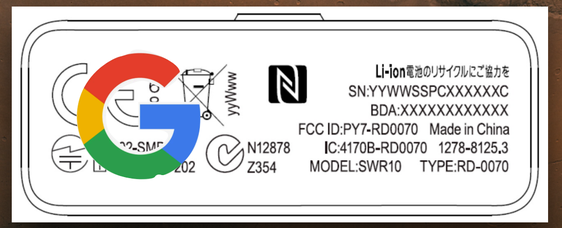
We will go around and hear everyone’s answers.

Also tell us:
Name (pronouns opt),
Major/minor,
Experience programming,
Expectations for course,
Topic most interested in,
One way that programming is used in your field that you might be interested in getting involved with,
Hobby (opt).



After you have decided whether it is alive/cognitive, you then turn it over and see a tiny label:
“Product of GoogleMars.”

Will this change your mind? Why or why not?

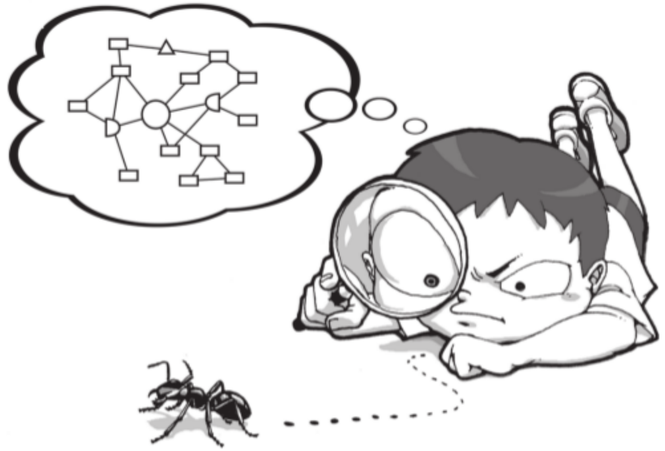


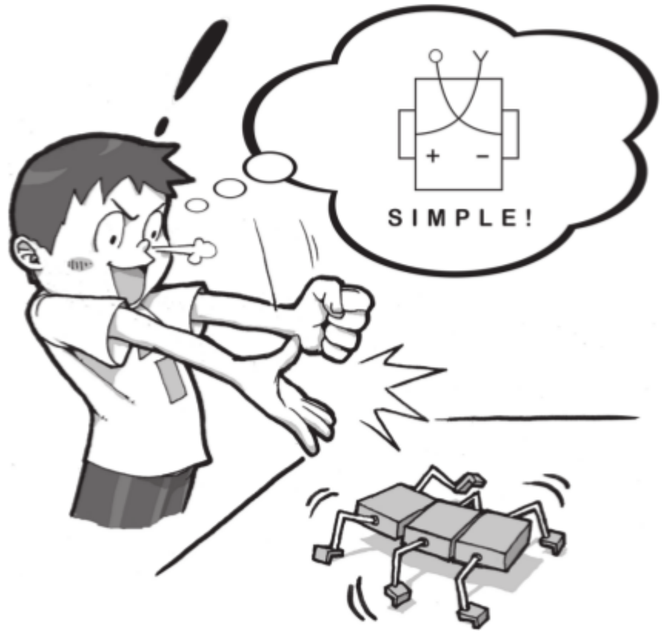
Outline

- Get to know each of you some.
- A little bit about me.
- Motivation (why programming for cognitive science?)
- Course overview (how we will improve our programming skills for cognitive science?)
- Demo

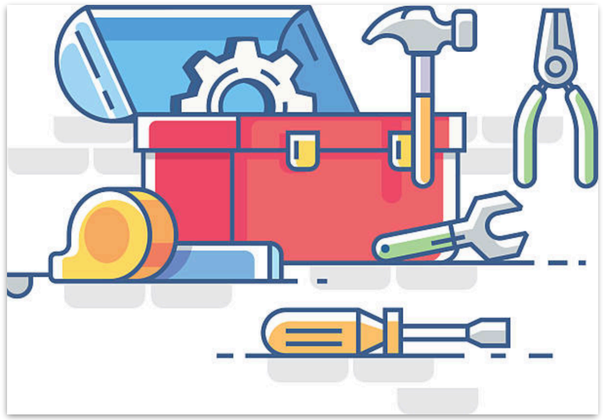
Illustrations by Shun Iwasawa in Pfeifer (2001)

General Motivation

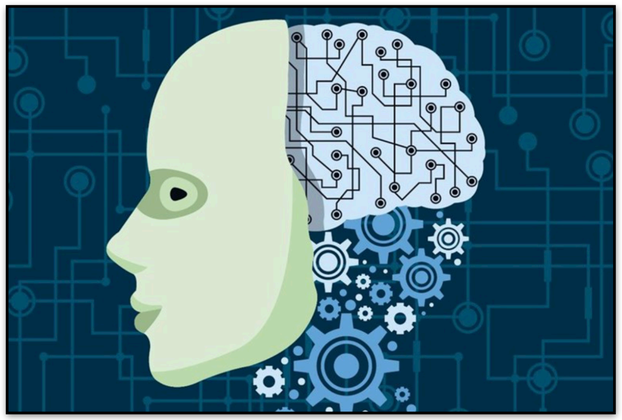




Generation of theory-driven hypothesis



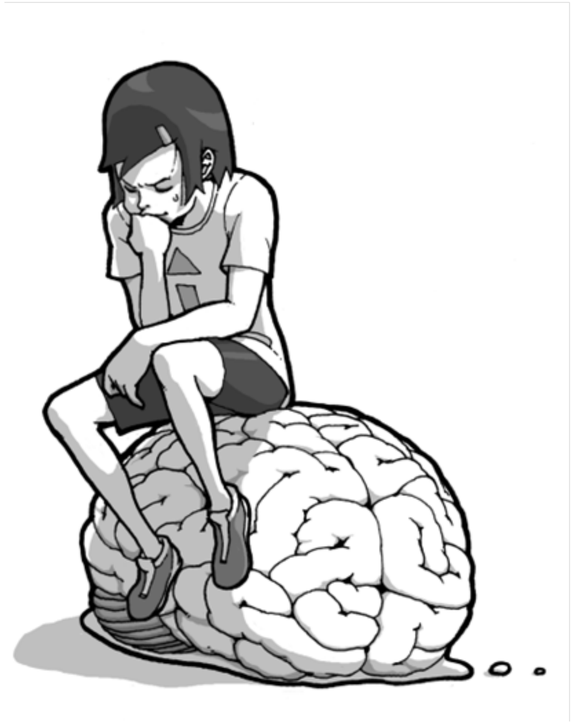
Development of tools of analysis



Improvements in Artificial Systems

Broad Challenge

Understand not just how brains work, but how behavior arises from the interaction between brain, body, and environment.

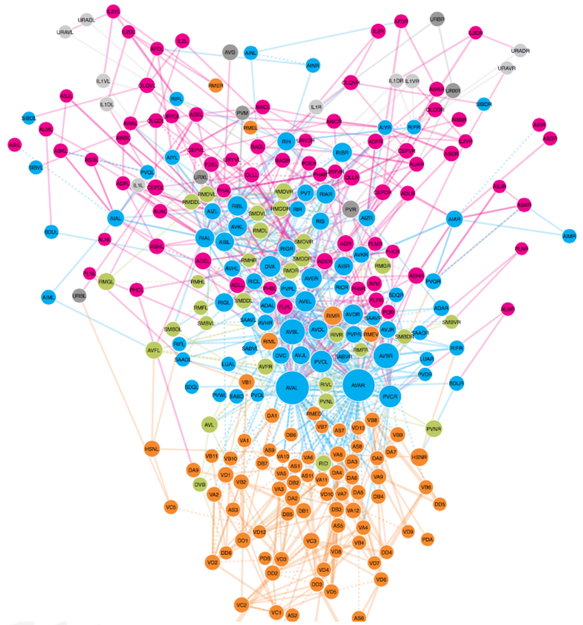


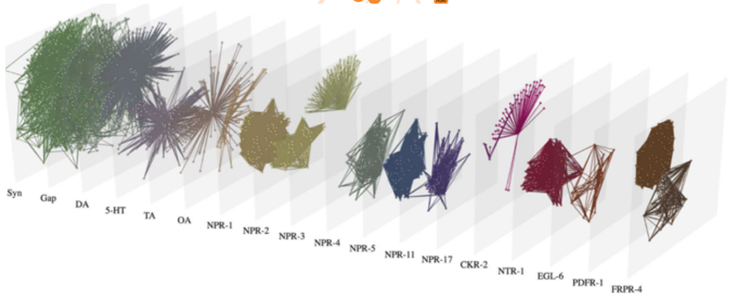
Understanding brains.

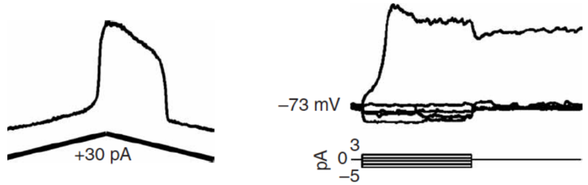
Understanding behavior.

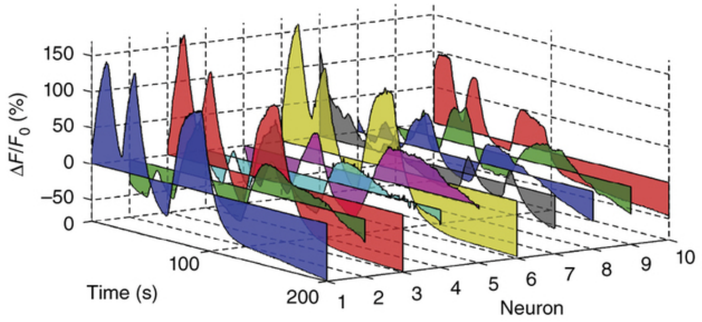
Illustrations by Shun Iwasawa in Pfeifer (2001)

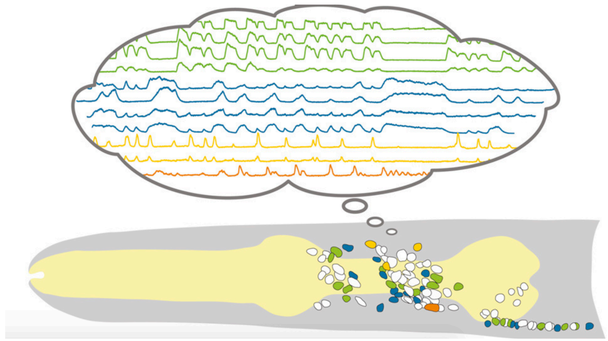
Connect structure, dynamics, and behavior

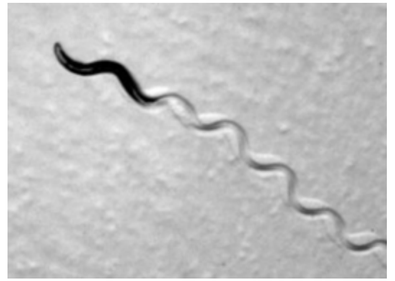


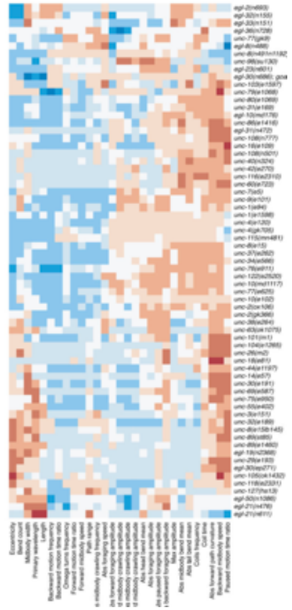


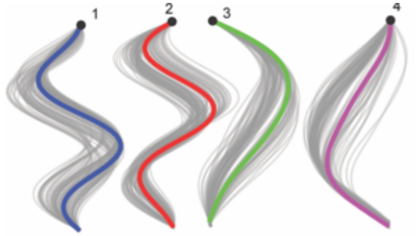


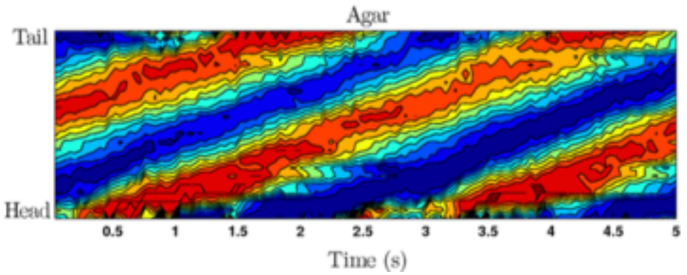












Structure: multilayer connectome, mechanical body

Dynamics: of individual neurons, muscles

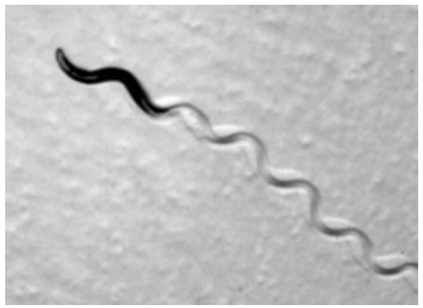
Behavior: Interactions between organism and environment

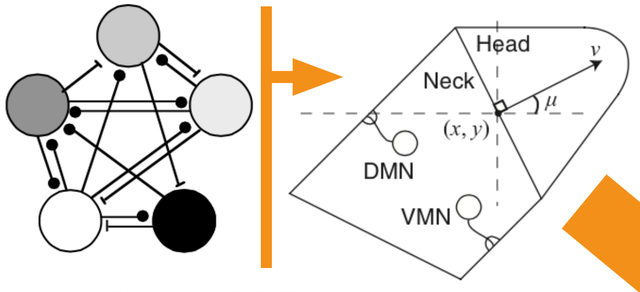
Approach

Constrained stochastic optimization and ensemble analysis

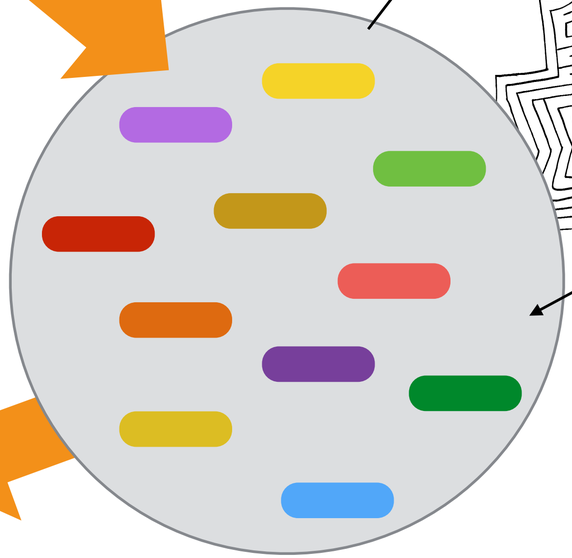
Search unknown parameters
for configurations that match behavior

Target behavior;
target organism

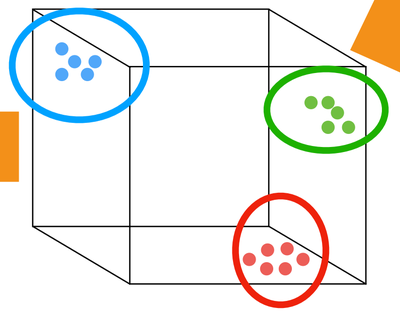




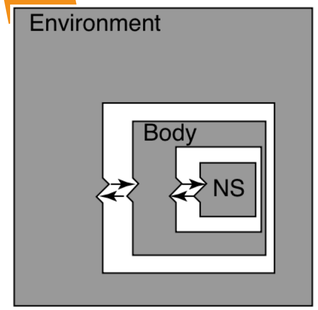
Design BBE model



Multiple runs to explore the
space of possible hypothesis



Find clusters in the ensemble of solutions

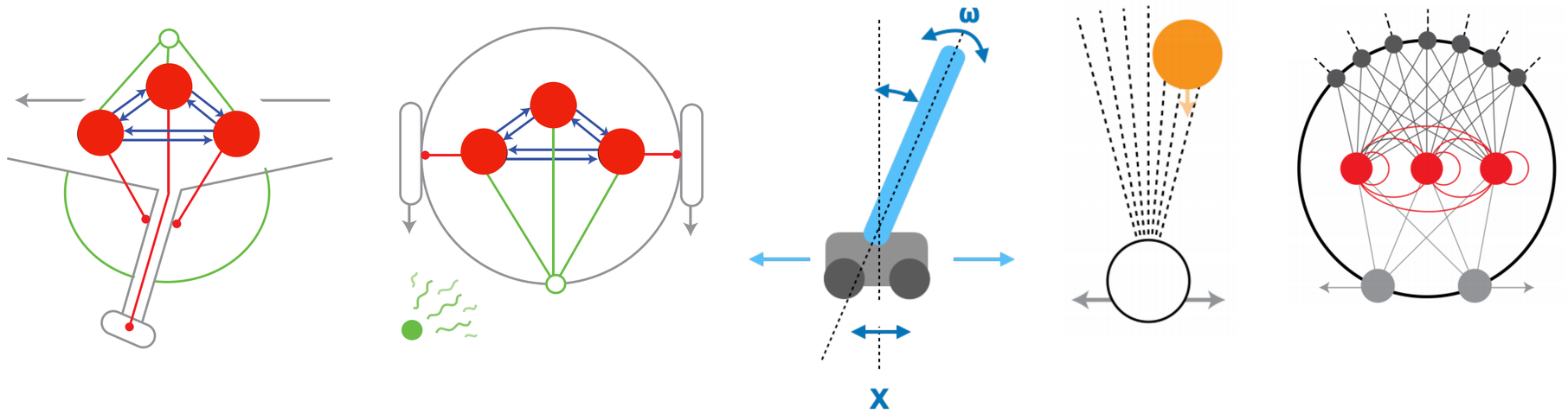


Analyze in some detail representative solutions

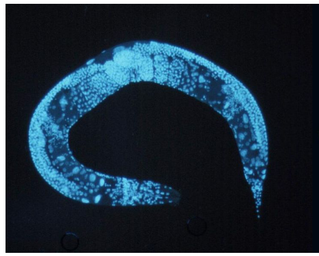
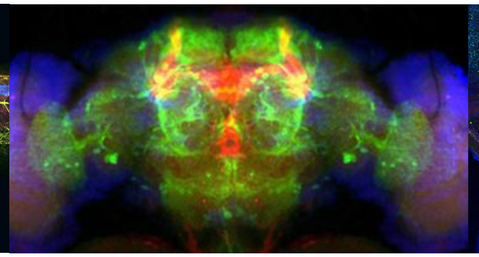
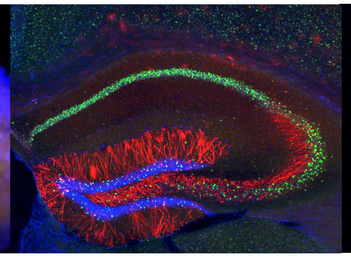
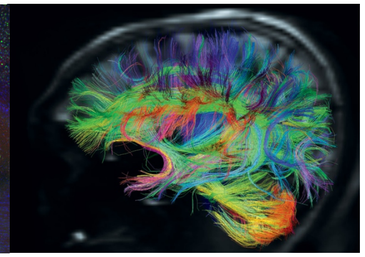
Hypotheses
And how to
experimentally test for
them

Approach

Like in many other areas of science, we start first with the relatively simplest instances of the problem of interest.



Idealized brain-body-environment systems that target general behaviors of interest.

C. elegans	Larval zebrafish	Fruit fly	Mice	Human
				
Neurons 302	100,000	250,000	71,000,000	Neurons $\sim 10^{10}$
8693 Connections	$\sim 10^{6}$	$\sim 10^{7}$	$\sim 10^{12}$	$\sim 10^{14}$ Connections

Modeling target organisms, starting with the simpler ones first.

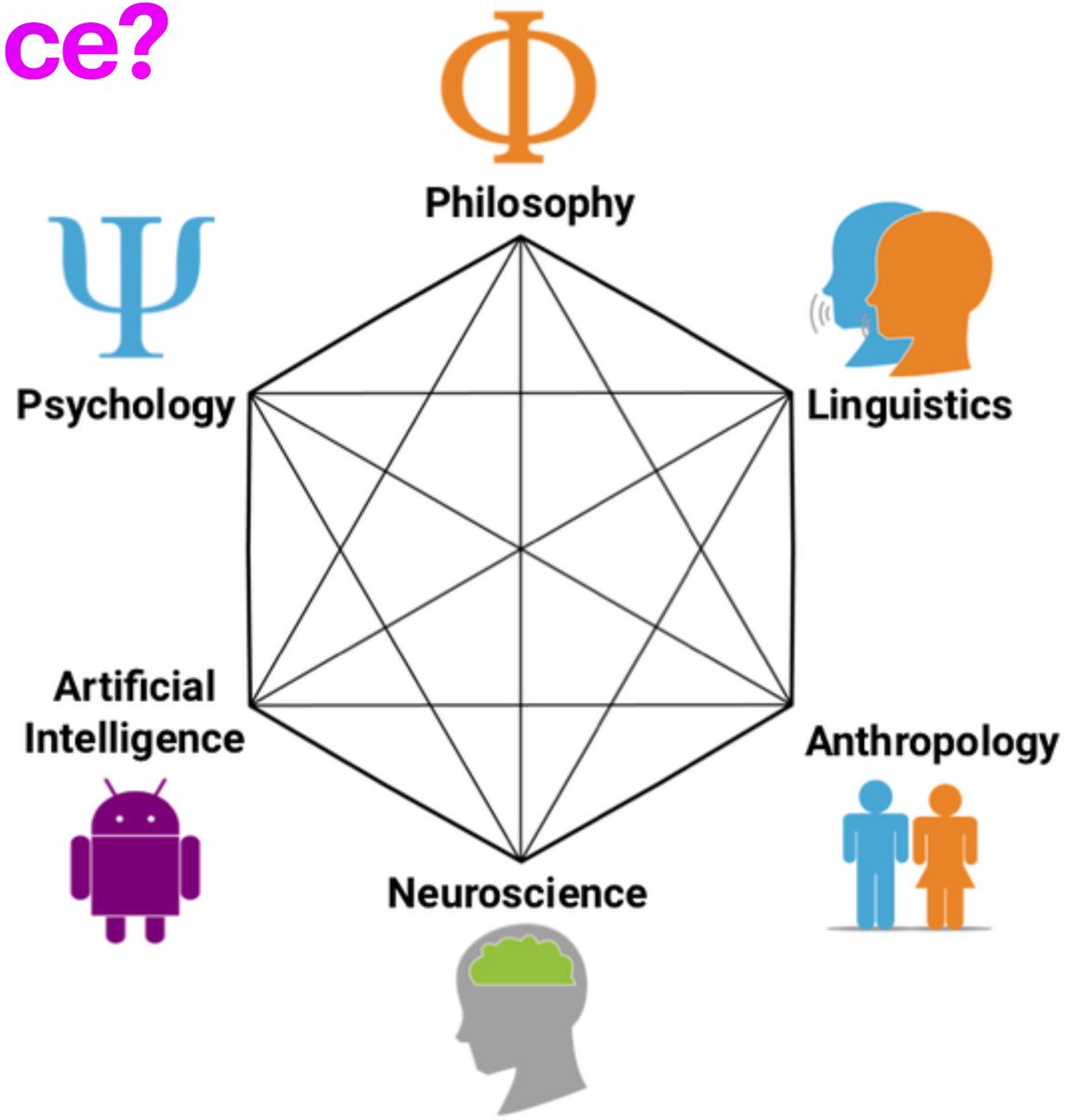
Outline

- Get to know each of you some.
- A little bit about me.
- Motivation (why programming for cognitive science?)
- Course overview (how we will improve our programming skills for cognitive science?)
- Demo

What is Cognitive Science?

“Cognitive science is the interdisciplinary study of mind and intelligence, embracing philosophy, psychology, artificial intelligence, neuroscience, linguistics, and anthropology.”

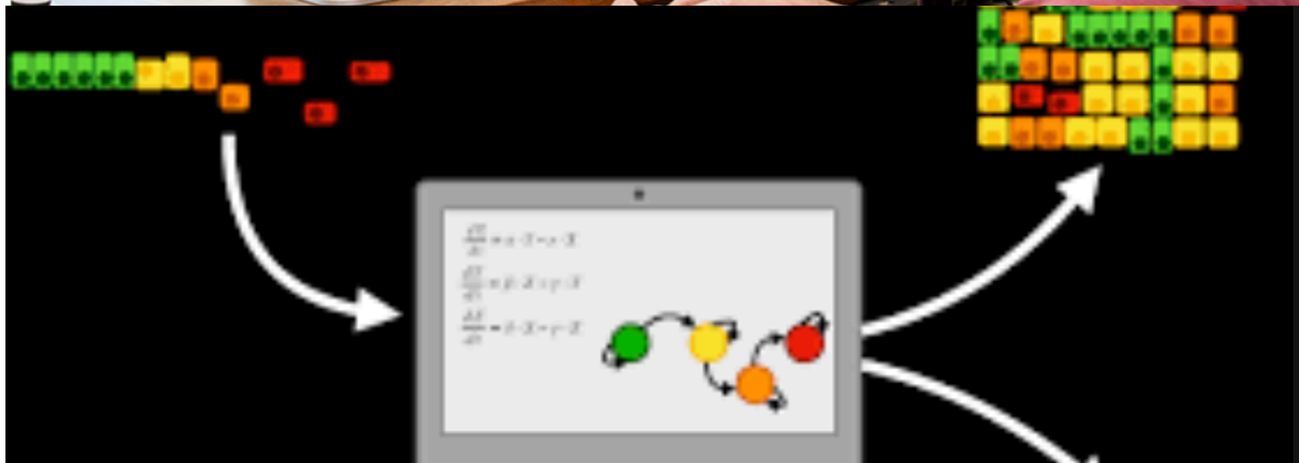
— Stanford Encyclopedia of Philosophy



Motivation

Programming is used in Cognitive and Information Sciences in three broad ways:

- To organize and analyze large amounts of experimental data.
- To generate experiments.
- To create computational models to help us understand such complex systems.



What is involved in programming?

The process of designing, writing, testing, debugging, and maintaining the source code of computer programs.

What skills are developed with programming?

Like mathematicians, you will develop the skills to use formal languages to denote ideas.

Like engineers, you will will develop the skills to design, assemble components, evaluate tradeoffs among alternatives.

Like scientists, you will develop the skills to observe the behavior of a complex system, form hypotheses, and test predictions.

Like artists, you will develop the skills have to be creative and to develop your own programming style.

What is the aim of programming?

Problem solving: How do I write a piece of code that will help me solve a specific problem?

Philosophy of mind with a screwdriver: How do I write a piece of code that will help me think about a specific phenomena: to generate predictions, to test assumptions, to validate understanding, to generate hypothesis.

Goals for this course

Provide you with a skill set of basic programming tools, and

An appreciation for how these tools can be used in cognitive science.

Learning to program is similar to learning a new language or a new physical skill.

It takes time. It takes practice.

Plan accordingly.

Philosophy for teaching progamming

No “geek gene”: competence at programming is not innate but is rather a learned skill that is improved with practice.

Live coding: Rather than using slides, we will aim to write programs live in the classroom.

Make predictions about the outcome of the demonstration before performing it.

Pair programming. Two people at a keyboard, one "driver" and one "navigator."

Stick to one language. We will work primarily in Python (with some exposure to a handful of other languages).

Philosophy for teaching progamming

Pen and paper first. Write the algorithm first on pen and paper, then as comments in the code, and only finally code.

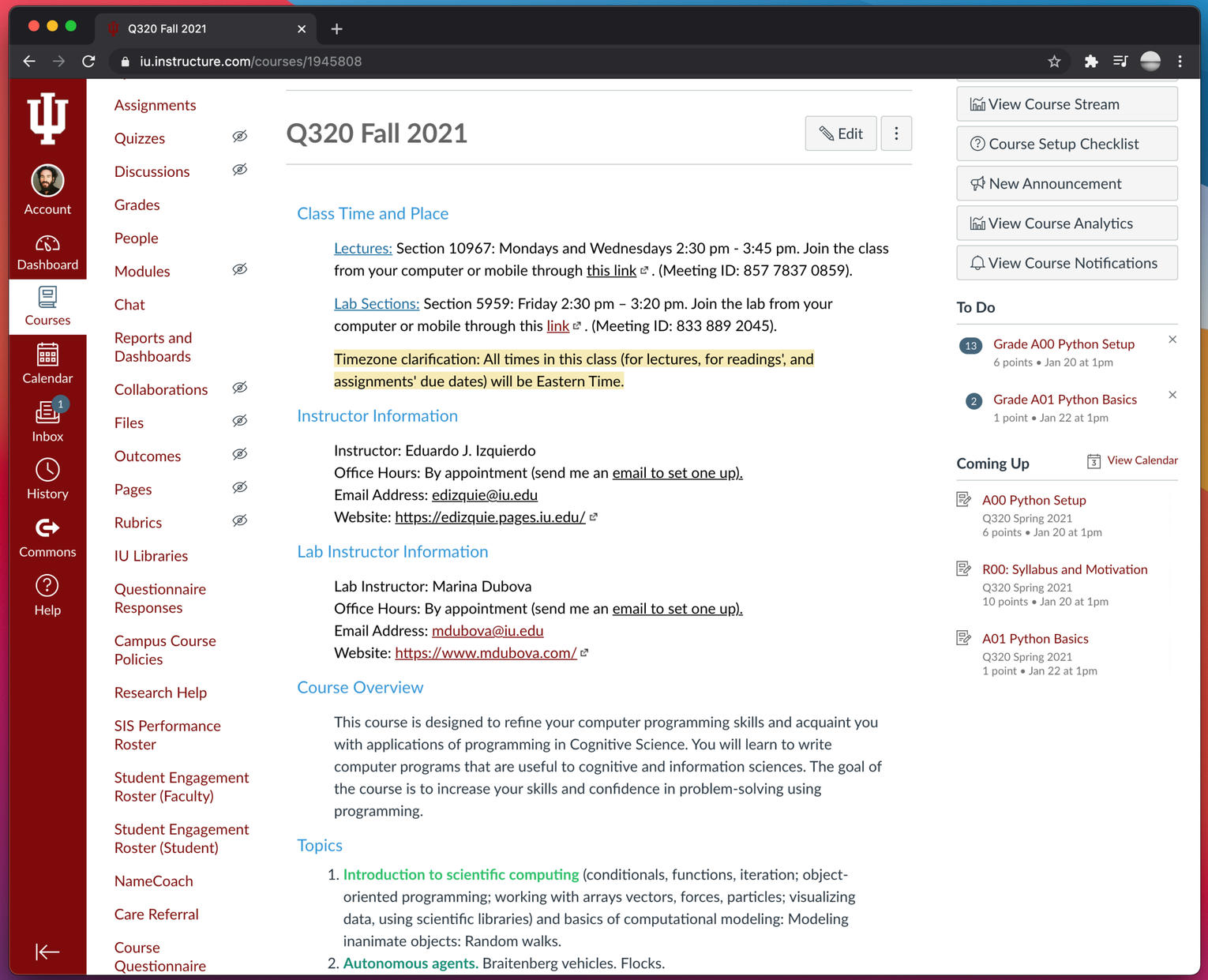
Divide and conquer. Break down the problem into subproblems, whenever possible. Solve and test solutions to the subproblems before putting them together.

Learn to debug. Understanding the problem in your code is essential. Debugging can be particularly frustrating to non-experts.

Peer instruction. Participate in class. Help each other. When I describe something, see if you can describe it differently to help others. Discuss with peers.

Outline

- Get to know each of you some.
- A little bit about me.
- Motivation (why programming for cognitive science?)
- Course overview (how we will improve our programming skills for cognitive science?)
- Demo


Q320 Fall 2021
iu.instructure.com/courses/1945808
Account
Dashboard
Courses
Calendar
Inbox
History
Commons
Help
Assignments
Quizzes
Discussions
Grades
People
Modules
Chat
Reports and Dashboards
Collaborations
Files
Outcomes
Pages
Rubrics
IU Libraries
Questionnaire Responses
Campus Course Policies
Research Help
SIS Performance Roster
Student Engagement Roster (Faculty)
Student Engagement Roster (Student)
NameCoach
Care Referral
Course Questionnaire
Q320 Fall 2021
Edit
Class Time and Place
Lectures: Section 10967: Mondays and Wednesdays 2:30 pm - 3:45 pm. Join the class from your computer or mobile through this link . (Meeting ID: 857 7837 0859).
Lab Sections: Section 5959: Friday 2:30 pm – 3:20 pm. Join the lab from your computer or mobile through this link . (Meeting ID: 833 889 2045).
Timezone clarification: All times in this class (for lectures, for readings', and assignments' due dates) will be Eastern Time.
Instructor Information
Instructor: Eduardo J. Izquierdo
Office Hours: By appointment (send me an email to set one up).
Email Address: edizquie@iu.edu
Website: https://edizquie.pages.iu.edu/
Lab Instructor Information
Lab Instructor: Marina Dubova
Office Hours: By appointment (send me an email to set one up).
Email Address: mdubova@iu.edu
Website: https://www.mdubova.com/
Course Overview
This course is designed to refine your computer programming skills and acquaint you with applications of programming in Cognitive Science. You will learn to write computer programs that are useful to cognitive and information sciences. The goal of the course is to increase your skills and confidence in problem-solving using programming.
Topics
1. Introduction to scientific computing (conditionals, functions, iteration; object-oriented programming; working with arrays vectors, forces, particles; visualizing data, using scientific libraries) and basics of computational modeling: Modeling inanimate objects: Random walks.
2. Autonomous agents. Braitenberg vehicles. Flocks.
View Course Stream
Course Setup Checklist
New Announcement
View Course Analytics
View Course Notifications
To Do
13 Grade A00 Python Setup
6 points • Jan 20 at 1pm
2 Grade A01 Python Basics
1 point • Jan 22 at 1pm
Coming Up
View Calendar
A00 Python Setup
Q320 Spring 2021
6 points • Jan 20 at 1pm
R00: Syllabus and Motivation
Q320 Spring 2021
10 points • Jan 20 at 1pm
A01 Python Basics
Q320 Spring 2021
1 point • Jan 22 at 1pm

Grading

Participation: 10%

All engagement relevant to class.

Reading: 12%

Divided uniformly between all reading assignments.

Lab assignments: 48%

Divided uniformly between all programming assignments (~12).

Final project: 30%

5% for the proposal

10% for the oral presentation

10% for the written report

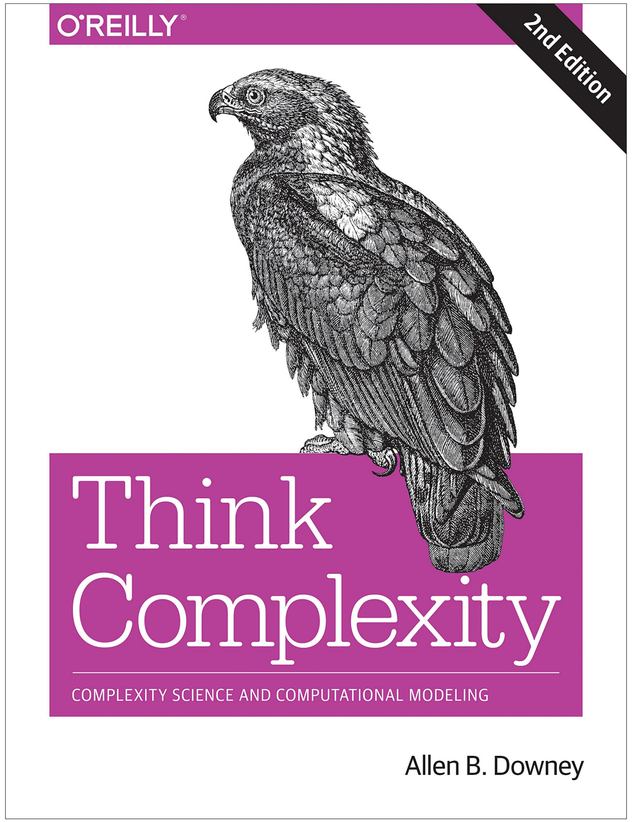
5% for the code

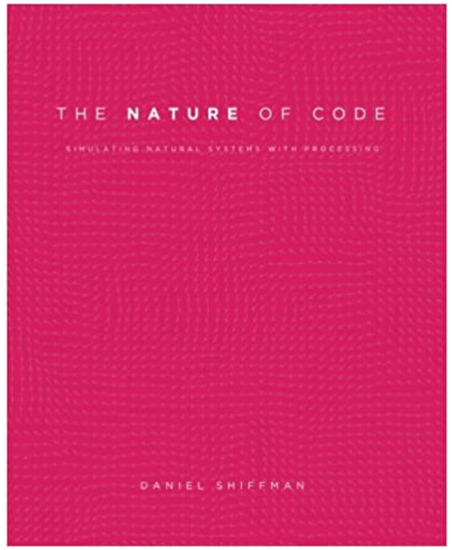
Readings

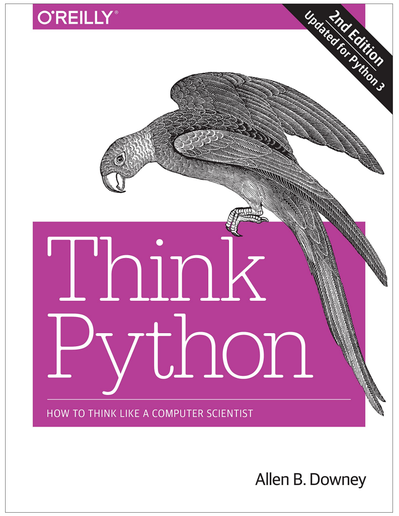
Readings will be assigned on Canvas for (nearly) every lecture.

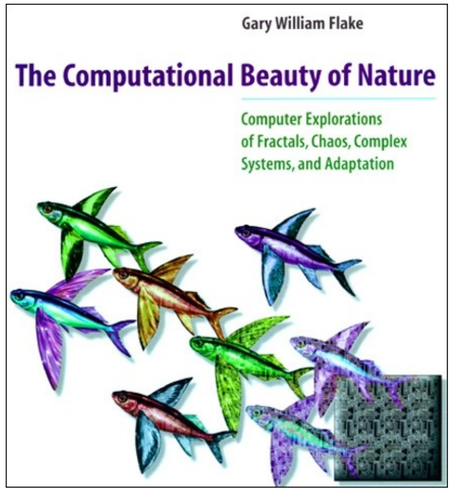
You will (typically) have to demonstrate that you've done the reading by answering some questions posted on Canvas.

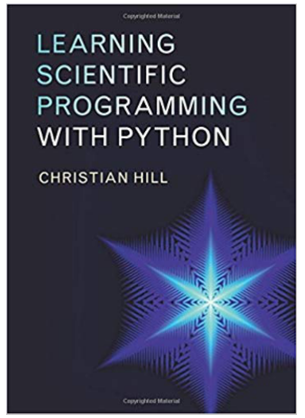
When reading, you will encounter exercises. Do them! There is only one way to learn programming: you have to program yourself.

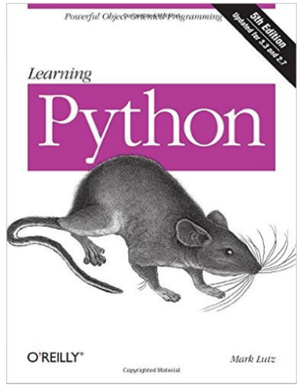


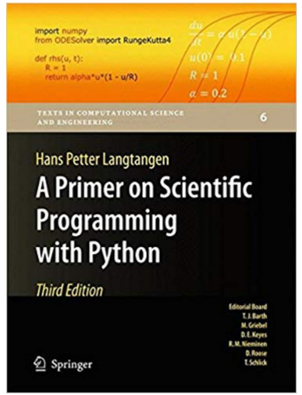


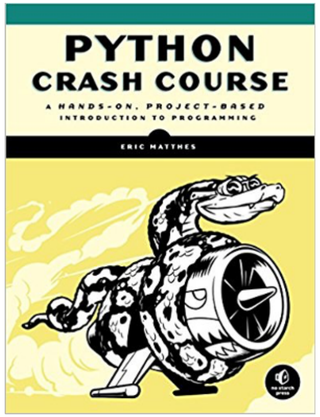






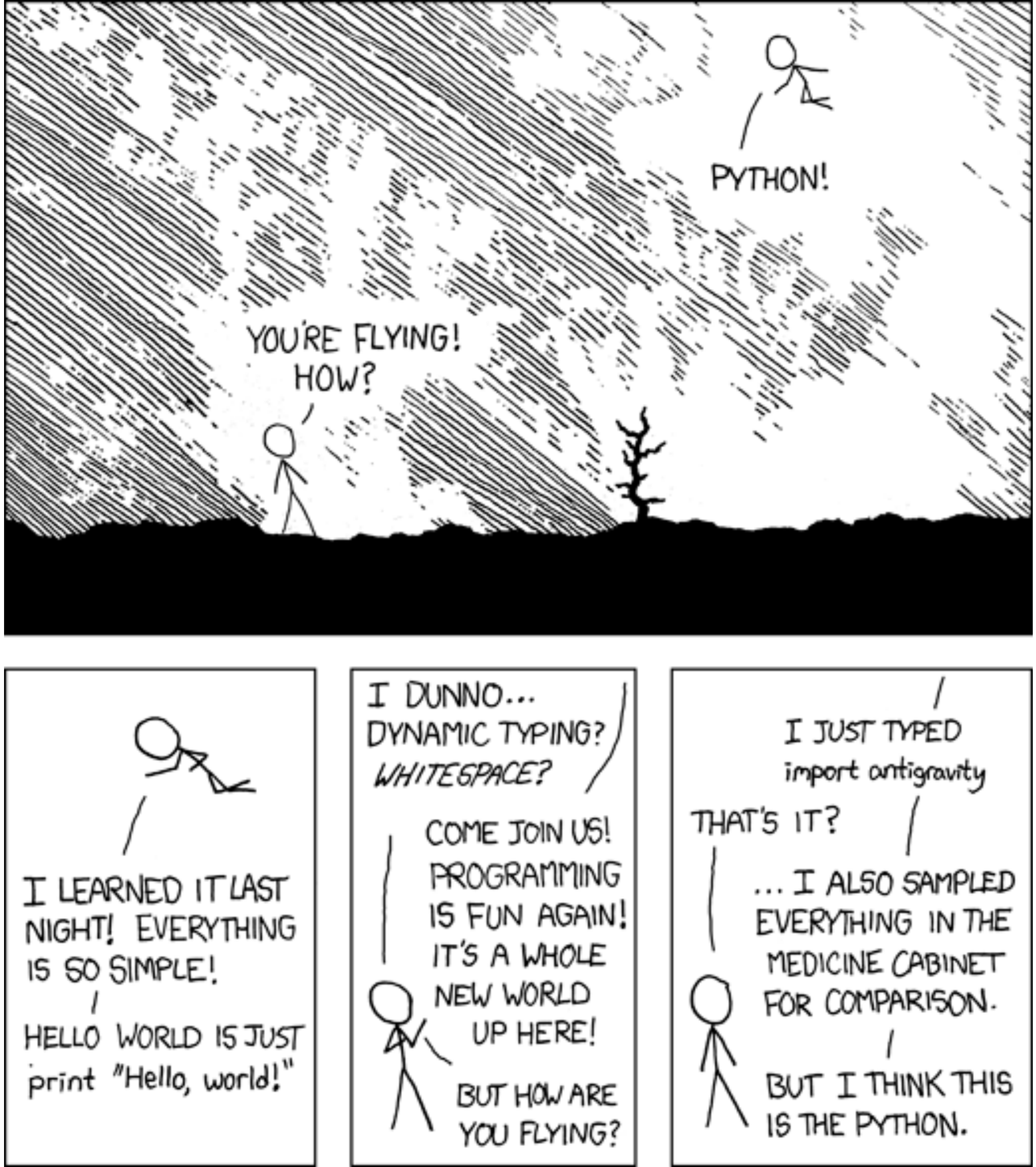






Why Python?

- Simple but powerful.
- Python was designed to be easy for humans to read and to minimize the amount of time required to write code.
- It is increasingly used as the first language of preference in most of the top CS schools.
- It is considered a general-purpose programming language.
- Major companies like Google, Dropbox, etc use Python in their core applications.
- Cross-platform.
- Python community.



Programming Assignments

- Weekly for (roughly) the first 12 weeks.
- Posted on Canvas each week and discussed in class.
- Further developments of the topics covered in class.
- Submit lab assignments online via Canvas.
- Late work is permitted but will be deducted by 20% per week from the original due date.
- You are welcome to collaborate on assignments with your peers. Each student must turn in their own assignment. Additionally, each student must state in their report who they collaborated with and what each of your contributions were.
- If you have problems with the homework, seek help - early.

Final Project

The final 3-4 weeks of the semester will be dedicated to working on a final project on a topic of interest and potential relevance to your graduate research.

- Proposal: 500 word abstract proposing how you will use programming to study an aspect of cognition (broadly defined).
- Oral presentation: 15 minute presentation to share your final project and to get feedback from your peers.
- Written report: Write a short paper (over 1000 words, four or more pages in length, 1.5 spaced, 12 point font) reporting on the motivation for your project, the methods, a justification of the model and the programming, the results, and a discussion. The report should include figures and references.
- Code. You will also be asked to turn in the programming code.

Part 1: Refresh Programming Basics

- Programming languages.
- Python and programming environments.
- Basics: Values. Data types. Operators. Errors. Debugging. Flow control. Functions. Recursion.
- File input and output.
- Visualizing data.
- Programming style.
- Object-oriented programming.

Part 2: Applications

1. **Autonomous agents.** Braitenberg vehicles. Flocks.
2. **Complex systems.** Interactions and emergence: 1D Cellular automata. 2D Cellular automata (Game of life). Sand piles and self-organized criticality.
3. **Agent-based models.** Segregation model. Game theory - modeling competition and cooperation. Prisoner's dilemma. Ant trails. Sugarscape. Opinion dynamics.
4. **Learning, Growth, and Adaptation.** Evolutionary algorithms. Learning algorithms. L-systems and fractal growth.
5. **Dynamical systems.** Euler method. Lotka-Volterra. Ecological models of multiple interacting species. Biophysical model of individual neurons (FitzHugh-Nagumo, Hodgkin-Huxley). Dynamical recurrent neural networks.
6. **Artificial neural networks.** Perceptron. Feedforward neural network. Backprop. Introduction to libraries that do a lot of this work automatically.

Part 3: Final Project

What are you interested in?

Demo

Running Python from a Terminal, from iDLE, from Spyder, and from Jupyter