

# CSSE 120 DAY 1

Intro to Software Development

# Outline



- Roll call
- Introductions
- Introduction to course
- Hands-on Introduction to Python

# Roll Call & Introductions

- Name (nickname)
- Hometown
- Where you live on (or off) campus
- Something about you that most people in the room don't know

**Note:** this means you should be Answering Question #1 on the quiz → Q1

# Instructor Intro



# Administrivia

- Background
- Syllabus
- Schedule
- Homework 1 due at start of next class
  - ▣ Reading and Angel quiz on it
  - ▣ Programming part
    - Note: please put your **name** in a comment at the top of your Python file
    - Otherwise, you will earn 0 points
    - Style requirements will be added as course progresses.

# Exams

- Mark your Calendar!
- Exam 1: Tuesday, September 28, 7:00-9:00 PM  
No regular class that day
- Exam2: Thursday, October 21, 7:00-9:00 PM  
No regular class that day
- Final Exam: To be scheduled during Finals week

# How to succeed in CSSE120

- Read the textbook before each class
  - ▣ Try out the code
  - ▣ Take the ANGEL quiz over the reading
    - If you don't do well, read again and retake quiz
  - ▣ Ask questions on what you don't understand
- Start early on the programming assignments
  - ▣ Don't be satisfied with getting your code to work
  - ▣ Be sure you understand it. If you don't, ask!
- Work and learn with other students
  - ▣ But don't let them do your work for you
- Take advantage of instructor office hours and student assistant lab hours

# Basic Definitions

## □ Computer

- Device for manipulating data
- Under control of a changeable program

## □ Program

- Detailed set of instructions
- Step by step
- Meant to be executed by a computer



# The two ends of programming

---

1. See the Big Picture
2. Get the Details Right

Many important programming techniques are methods of getting from #1 to #2.

# Some Computer Science Questions

- What can be computed?
- How to compute it efficiently?
- What is the best way to turn a mass of raw data into usable information?

# What is an Algorithm?

---

- Step-by-step procedure for accomplishing something
- Presented at the right level of detail (and in the right language) for the one who will execute it

# Algorithm Analogy -- Recipe

---

- Bake a cake
  - Instructions for an experienced cook
  - Instructions for a 7-year-old
  - Instructions in French

# Algorithm for a very simple task

---

- For a student to execute.
- For a robot to execute.

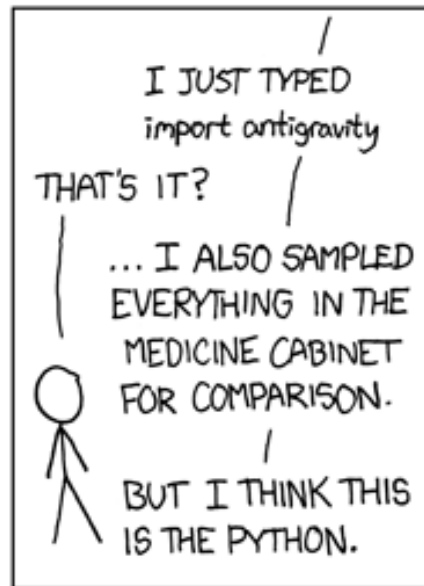
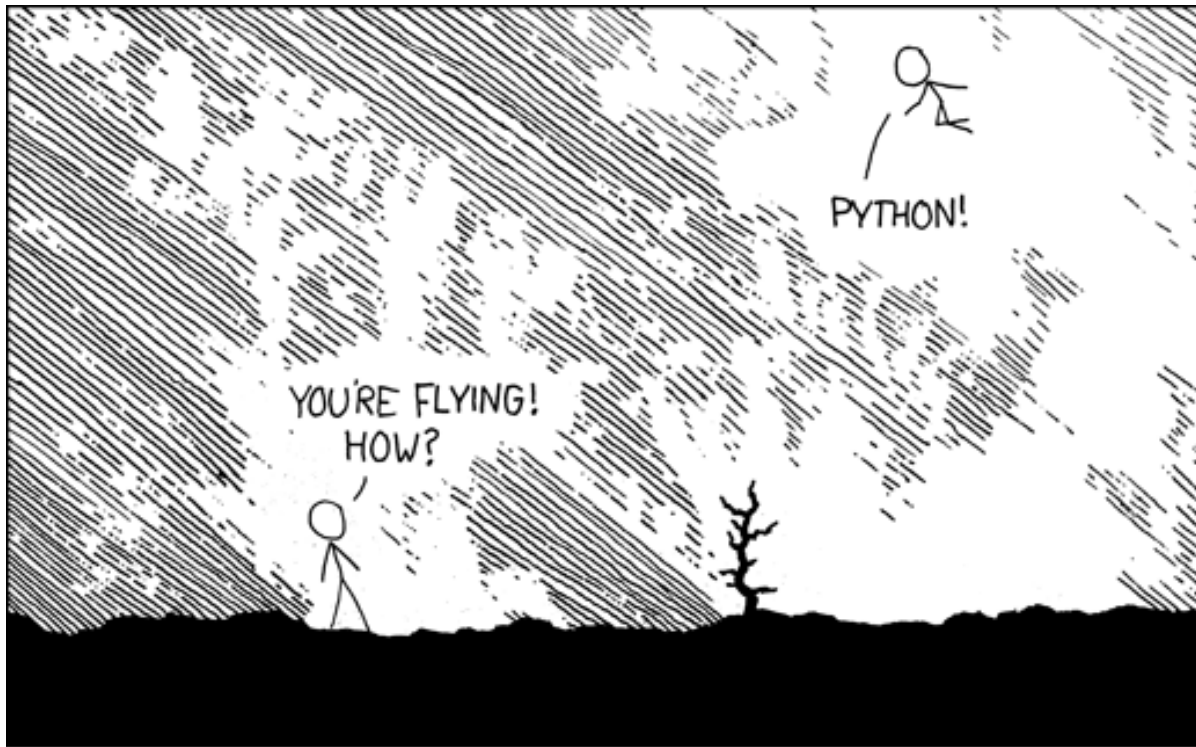
# Four important CS skills

---

- Design algorithms
- Analyze algorithms
- Evaluate algorithms
- Adapt algorithms

# Human Languages vs. Programming Languages

- Ambiguous vs. very precise
- Syntax (form) must exactly match ...
  - ▣ CaSe MAtterS
- Semantics (meaning)
- Translation
  - ▣ High-level language (Maple, Java, Python, C) to
  - ▣ Low-level language (machine language)
  - ▣ Compiler, interpreter





# PYTHON: A PROGRAMMING LANGUAGE!