http://www.cs.cornell.edu/courses/cs1110/2020sp

Lecture 1: Introduction, Types & Expressions

(Chapter 1, Section 2.6)

CS 1110

Introduction to Computing Using Python

[E. Andersen, A. Bracy, D. Fan, D. Gries, L. Lee, S. Marschner, and W. White]

CS 1110 Spring 2019: Announcements

http://www.cs.cornell.edu/courses/cs1110/2020sp

Sections

- Please go only to the Section in which you are enrolled
- See our *Section Swapping Station* on Piazza: https://piazza.com/cornell/spring2020/cs1110/

Enrollment

- A lot of turnover in the first week: don't give up!
- Perhaps another class meets your needs?

http://www.cs.cornell.edu/courses/cs1110/2020sp/alternatives.html

AEW Workshops (ENGRG 1010) Open to **all** students.

Additional (optional) discussion course. Small group, collaborative learning. Non-remedial. Highly recommended.

http://www.cs.cornell.edu/courses/cs1110/2020sp/aew.html

Why learn to program?

(subtly distinct from, although a core part of, CS / IS)

*Like philosophy … **computing is worth teaching** less for the subject matter itself and more **for the habits of mind that studying it encourages.***

"Teach computing, not Word", the Economist

http://www.economist.com/blogs/babbage/2010/08/computing_schools

Why learn to program? (continued)

[T]he seductive intellectual core of… programming: here is a magic black box. [T]ell it to do whatever you want, within a certain set of rules, and it will do it; within the confines of the box you are more or less God, your powers limited only by your imagination. But the price of that power is strict discipline: you have to *really know* what you want, and you have to be able to express it clearly in a formal, structured way that leaves no room for the fuzzy thinking and ambiguity found everywhere else in life…

…The ability to make the machine dance to any tune you care to play is thrilling.

Oh the places you'll go! (with 1110)

Benjamin Van Doren, CALS

- bird lover since 3rd grade
- learned programming as a freshman in CS1110 Spring 2013
- helped create dataset for paper he co-authored: "Approximate Bayesian Inference for Reconstructing Velocities of Migrating Birds from Weather Radar"
- won Best Paper Award at AAAI 2013 workshop

About Professor Lee

Research lifetime achievement awards:

- Association for Computing Machinery (ACM), 2018
- Assoc. for the Advancement of Artificial Intelligence (AAAI), 2013
- Assoc. for Computational Linguistics, 2017

In the press: New York Times, All Things Considered, Washington Post, etc.

Engineering teaching awards: 1999, 2002, 2012

Carpenter Memorial Advising Award: 2009

A.B. Cornell ’93, Ph.D. Harvard ’97

Lowest grade ever…?

Course logo for Spring 2020

In programming, as in life,
sometimes you're the whale;
sometimes, you're the sushi.

Keep on smiling anyway;
and remember we're here to
help you on your journey!

About Professor Fan

- Interest in optimization—what is the "best" way to operate a system given constraints and uncertainties?
- Other courses:
  - Intro to computing using Matlab
  - Optimization with metaheuristics

Source: energy.gov

- Author: *Insight Through Computing: A Matlab Introduction to Computational Science and Engineering* with C. F. Van Loan
- Honors:

  National Academy of Engineering Frontiers of Engineering Education (2014)
  Carpenter Memorial Advising Award (2016)
  Engineering teaching awards (2011, 2019)

Who does what?

What you see:

What you don't see:

http://www.catonmat.net/blog/front-end-vs-back-end-comic/

Why should you take CS 1110?

Outcomes:

- **Fluency:** (Python) procedural programming
  - Use assignments, conditionals, & loops
  - Create Python modules & programs
- **Competency:** object-oriented programming
  - Recognize and use objects and classes
- **Knowledge:** searching & sorting algorithms

Intro Programming Classes Compared (1)

CS 1110: Python

- No programming experience necessary
- No calculus
- Non-numerical problems
- More about software design

CS 1112: MATLAB

- No programming experience necessary
- 1 semester of calculus
- Engineering-type problems
- Less about software design

Both serve as a pre-requisite to CS 2110

Intro Programming Classes Compared (2)

CS 1133: Python Short Course

- No programming experience necessary
- No calculus
- Very basics of programming
- Already full! ☹

CS 1380: Data Science For All

- No programming experience necessary
- No calculus
- Less programming than 1110, but also: data visualization, prediction, machine learning

Course Website

http://www.cs.cornell.edu/courses/cs1110/2020sp/

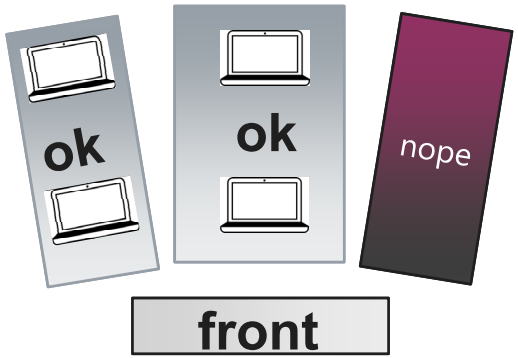
If the website doesn't look like this, with the sushi-whale logo, at the top left, you're looking at the wrong semester.

Lectures

- Tuesday & Thursday 9:05
- Not just talking! Demos, clicker questions, *etc.*
- Preview posted to website evening before class
- Slides, code examples, and video recording available on website later. Attend lecture to learn and discuss with peers—don't get behind.

Please, no cell phones during lecture

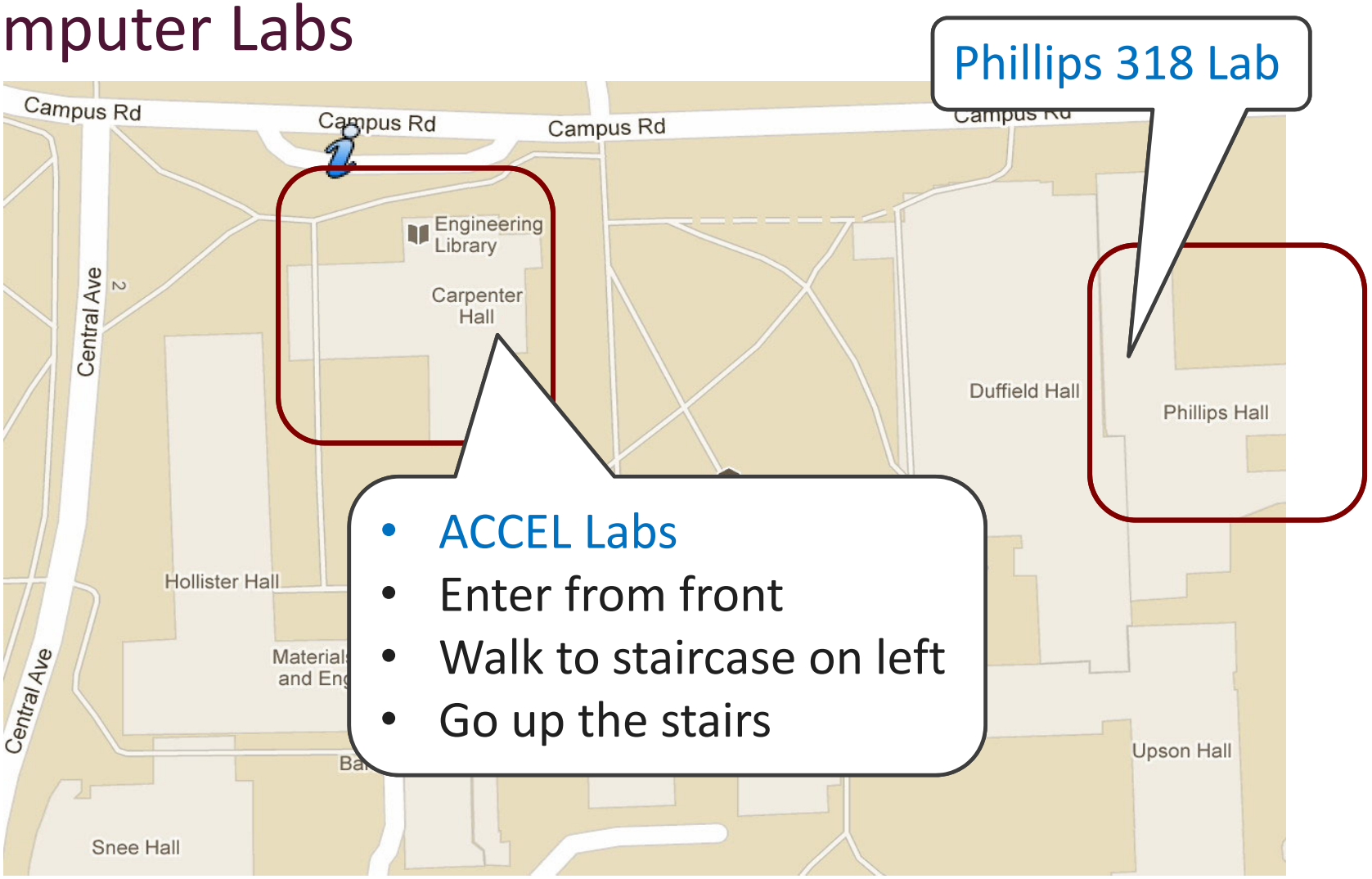
No laptop zone on your left, please do not use your laptop there



Lab Sections (aka Sections)

- Guided exercises with TAs & consultants
- Start today: Tuesday, January 21
- **Go to the lab section for which you are registered.** We can't maintain workable staff/student ratios otherwise.
  - Need a different Section? See our Section Swapping Station on Piazza: https://piazza.com/cornell/spring2020/cs1110/
  - Not enrolled in a lab section? *Don't panic.* Do the lab on your own. If a lab section opens up, check it in then.
- **Mandatory**. Missing > 2 can lower your final grade.

Computer Labs



Computers available for you to use whenever labs are open (see website FAQ). Bring a USB stick to save your work b/c you can't save files on these machines (for assignments).

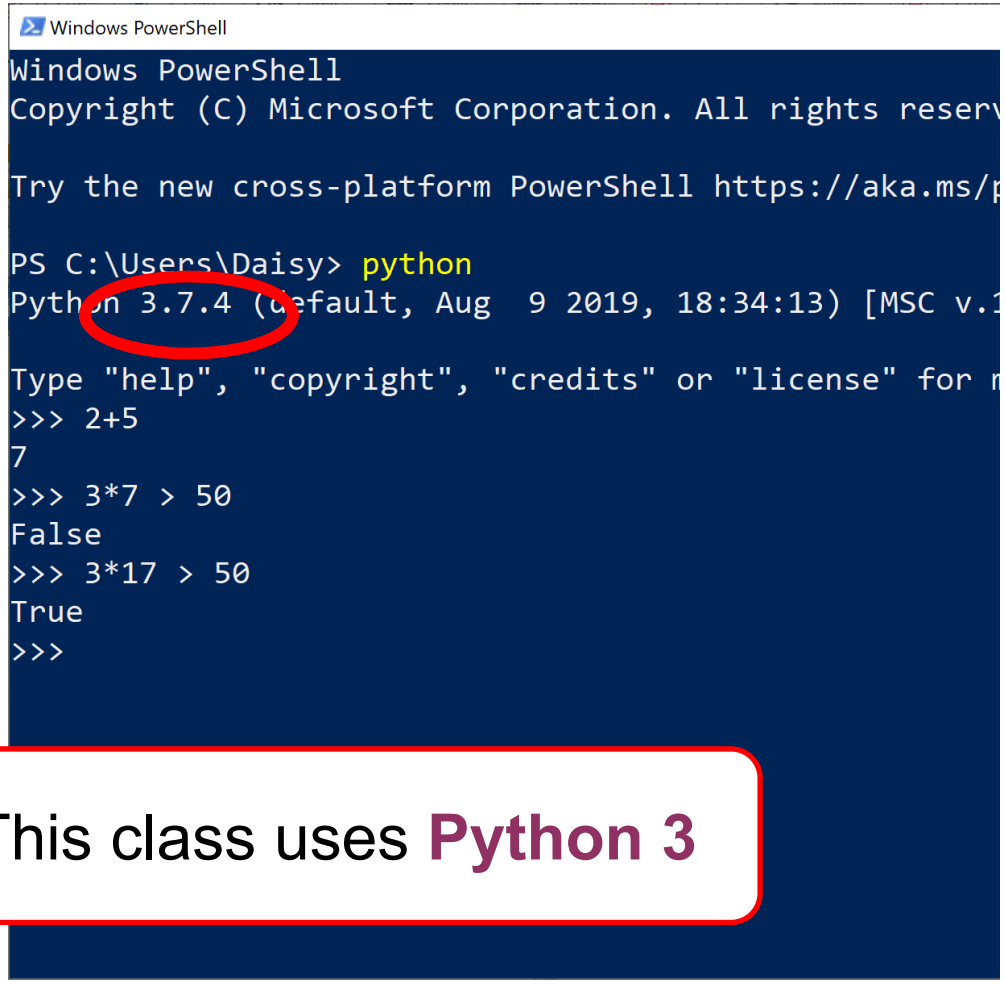
Getting started with Python

- Designed to be used from the “command line”
  - OS X/Linux: **Terminal**
  - Windows: **PowerShell**

    (old: **Command Prompt**)
  - Purpose of the first lab
- Install, then type “python”
  - Starts the *interactive mode*
  - Type commands at **>>>**
- First experiments:

  evaluate *expressions*

```
>>> terminal time >>>
```



Python not installed yet? Use a python interactive shell at www.python.org/shell

Storing and computing data

What data might we want to work with?

(What's on your computer?)

42

$3.0 * 10^8$

0.00001

14850

"apple"

"Tower Road"

"awb93"

True

False

Expressions

An expression **represents** something

- Python ***evaluates it*** (turns it into a value)
- Similar to a calculator

Examples:

- 2.3

Literal
(evaluates to self)

- (3 * 7 + 2) * 0.1

An expression with four literals and some operators

Types

A type is a set of values and the operations on these values

- Examples of operations: +, –, /, *
- Meaning of operations depends on type

Memorize this definition!

How to tell the type of a value?

Command: **type**(<value>)

Example:

```
>>> type(2)
<class 'int'>
```

Corrected after lecture: the result shown during lecture

```
<type 'int'>
```

was from using Python 3.6 instead of 3.7

```
>>> terminal time >>>
```

Type: `float` (floating point number)

Values: (approximations of) real numbers

- With a ".": a **float** literal (*e.g.,* 2.0)
- Without a decimal: an **int** literal (*e.g.,* 2)

Operations: +, –, *, /, **, unary –

to power of

Note: operator meaning can change from type to type

Exponent notation useful for large (or small) values

- –22.51e6 is $-22.51 * 10^{6}$ or –22510000
- 22.51e–6 is $22.51 * 10^{-6}$ or 0.00002251

A second kind of **`float`** literal

Floating Point Errors

Python stores floats as **binary fractions**

- Integer mantissa times a power of 2
- Example: 1.25 is $5 * 2^{-2}$

Can't write most real numbers this way exactly

- Similar to problem of writing 1/3 with decimals
- Python chooses the closest binary fraction it can

Approximation results in **representation error**

- When combined in expressions, the error can get worse
- **Example**: `0.1 + 0.2`

```
>>> terminal time >>>
```

Type: `int` (integers)

Values: …, –3, –2, –1, 0, 1, 2, 3, 4, 5, …

More Examples: 1, 45, 43028030

(no commas or periods)

Revised after lecture: historically called "int division" but misleading. We'll call it "floor division" because that's what it does!

division (technically a float operator)

"floor division": *divide then round down*

Operations: +, –, *, **, /, //, %, unary –

remainder

```
>>> terminal time >>>
```

Type: **bool** (boolean)

Values: `True`, `False`

- Boolean literals True and False (must be capitalized)

Operations: `not`, **and**, **or**

- **not** b: **True** if b is false and **False** if b is true
- b **and** c: **True** if both b and c are true; **False** otherwise
- b **or** c: **True** if b is true or c is true; **False** otherwise

Often come from comparing **int** or **float** values

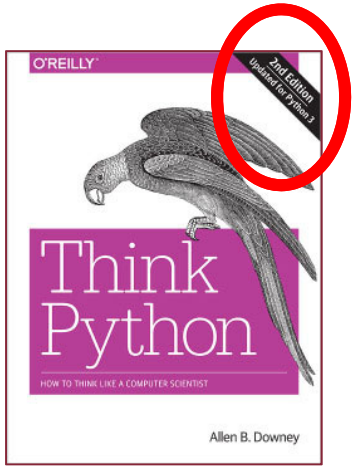
- Order comparison: k < j k <= j k >= j k > j
- Equality, inequality: k == j k != j

"=" means something else!

Class Materials

sash means 2nd ed

Textbook. *Think Python,* ***2nd ed.*** by Allen Downey



- *Supplemental;* does not replace lecture
- Available for free as PDF or eBook
- First edition is for the Python 2 (bad!)

iClicker. Optional but useful

- Will periodically ask questions during lecture
- **Not** part of the grade → no registration
- We do not support REEF Polling

Python. Necessary if using your own computer

- See course website for how to install

Things to do before next class

1. Read textbook
   - Ch 1, Sections 2.1-2.3, 2.5
2. (If using your own computer) Install Python **following instructions on the website under Materials**
3. Go to Lab!
4. (optional) Join Piazza, a Q&A forum

Lots of information on the website!

- Class announcements
- Consultant calendar
- Reading schedule
- Lecture slides
- Exam dates
- Piazza instructions

Read it thoroughly:

www.cs.cornell.edu/courses/cs1110/2020sp/

Communication

cs1110-prof@cornell.edu

- Includes: both professors & head TA
- **For sensitive correspondence.** Don't email one prof, or both separately.

cs1110-staff@cornell.edu

- Includes: both profs, admin assistant, graduate TAs, head consultants
- **For time sensitive correspondence (i.e., emergencies).** E.g., Nobody at office hours.

Piazza: not required, but fast (https://piazza.com/cornell/spring2020/cs1110/)

Email from us: please check your spam filters for mail from kdf4, LJL2, cs1110-prof, or with [CS1110] in the subject line