Lecture 1: Introduction, Types & Expressions

(Chapter 1)

CS 1110
Introduction to Computing Using Python

Cornell Bowers CIS
Computer Science

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

CS 1110 Spring 2022: Announcements

http://www.cs.cornell.edu/courses/cs1110/2022sp

Sections

- Please go only to the Section you are enrolled in
- Use Student Center to change (swap) section if necessary

Enrollment

- There is a lot of turnover in the first week. Don't give up!
- Perhaps another class meets your needs?

http://www.cs.cornell.edu/courses/cs1110/2022sp/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/cs1110/2022sp/aew.html

Why learn to program?

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

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)

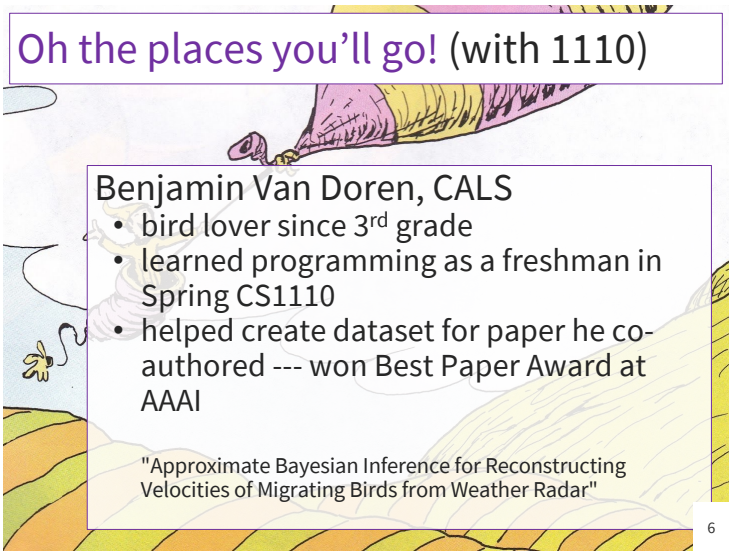
[T]he seductive intellectual core of… programming: here is a magic black box. [T]ell it to do whatever you want, within a certain set of rules, and it will do it;

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…

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



Benjamin Van Doren, CALS

- bird lover since 3rd grade
- learned programming as a freshman in Spring CS1110
- helped create dataset for paper he co-authored --- won Best Paper Award at AAAI

"Approximate Bayesian Inference for Reconstructing Velocities of Migrating Birds from Weather Radar"

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.

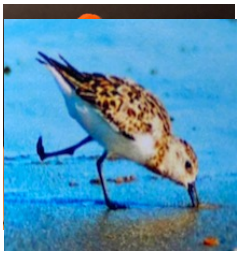
Engineering teaching awards: 1999, 2002, 2012

Carpenter Memorial Advising Award: 2009

A.B. Cornell '93, Ph.D. Harvard '97

Lowest grade ever…?

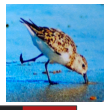
Course logo for Spring 2022



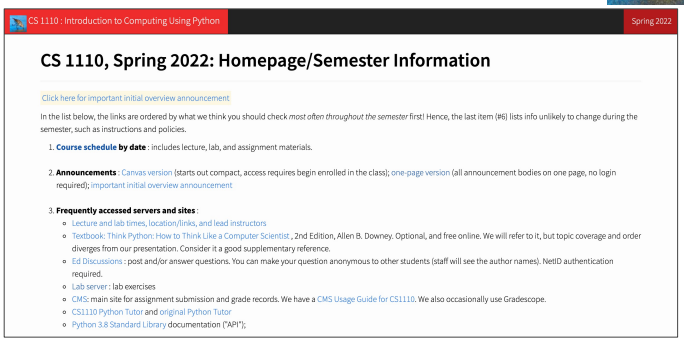
"We met up with this little bird on the beaches at Chicoteague. Either by deformity or injury, the bird's one leg was backward. This didn't deter [her]. She liked her territory enough that we were able to find her in the same place every day while we were there, poking away for snacks and keeping other birds away. She was just cute and tenacious and one of my favorite animals on the island." -- Cindy Robinson, longtime Cornellian

Hang in there, and remember we're here to help you on your journey!

Course Website



http://www.cs.cornell.edu/courses/cs1110/2022sp/



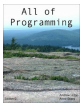
If the website doesn't look like this,
you're looking at the wrong semester.

About Professor Bracy

- BA, German Studies; BS, Symbolic Systems
- MS, Computer Science
- PhD, Computer Science

- Research Scientist, Intel Labs

- Co-Author of "All of Programming"
  - Google Play Book, Coursera Course!



- Senior Lecturer, Cornell University
  - CS 1110, 2110, 3410, 4410/4411
  - ACSU Faculty of the Year, 2016
  - Engineering Teaching Award, 2017
  - Tau Beta Pi Professor of the Year, 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	CS 1112: MATLAB
• No programming experience necessary	• No programming experience necessary
• No calculus	• 1 semester of calculus
• Non-numerical problems	• Engineering-type problems
• More about software design	• 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
- Fills up fast!

So many options:

https://www.cs.cornell.edu/courses/cs1110/2022sp/alternatives.html

CS 1110 is for students with **zero programming experience**.

Why Python?

Low overhead

- Little to learn before you start "doing"
- Easier for beginners
- Designed with "rapid prototyping" in mind

Highly relevant to non-CS majors

- NumPy and SciPy heavily used by scientists

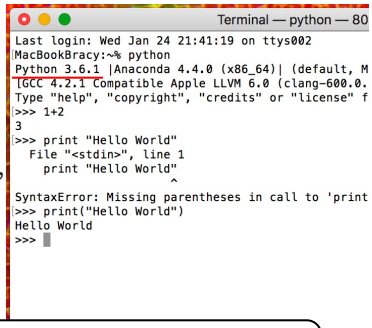
A modern language

- Popular for web applications (e.g. Facebook apps)
- Applicable to mobile app development

Getting Started with Python

- Designed to be used from the "command line"
  - OS X/Linux: **Terminal**
  - Windows: **PowerShell**
  - Purpose of the first lab
- Install, then type "python"
  - Starts the *interactive mode*
  - Type commands at >>>
- First experiments:

  evaluate *expressions*



This class uses **Python 3**
- Make sure you are, too!

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


Expressions

An expression **represents** something

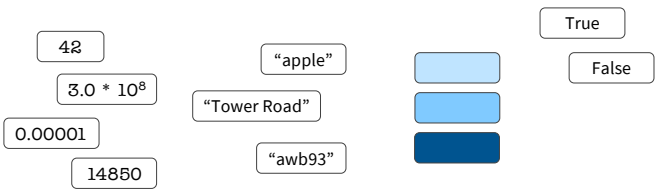
- Python ***evaluates it*** (turns it into a value)
- Similar to a calculator

Examples:

- `2.3` — Literal (evaluates to self)
- `(3 * 7 + 2) * 0.1` — An expression with four literals and some operators

Storing and Computing Data

What data might we want to work with?
(What's on your computer?)



Types

A set of values & 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)
<type 'int'>
```

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


Type: **float** (floating point)

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 −

Notice: 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 cannot store most real numbers exactly
- Similar to problem of writing 1/3 with decimals

Approximation results in **representation error**
- When combined in expressions, 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)

division (technically a float operator)

floor division

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

multiply

to power of

```
>>> 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: `i < j` `i <= j` `i >= j` `i > j`
- Equality, inequality: `i == j` `i != j`

"=" means something else!

Boolean Misconceptions

Booleans expressions *sound like* English, but subtle differences cause problems:

- In English, "A = B and C" often means "A = B and A = C"
  Example: "Ithaca is cold and snowy"
  - Means: "Ithaca is cold" and "Ithaca is snowy"
  - **Does not mean**: "Ithaca is cold" and…. "snowy"

  Python requires *fully specified* Boolean expressions
- In English, "A or B" often means "A or B ***but not both***"
  Example: "I'll take CS 1110 or CS 1112" (but not both)

  In Python, "A or B" always means "A or B ***or both***"

Type: **str** (string) for text

Values: any sequence of characters
Operation(s): **+** (catenation, or concatenation)
Again: operator **+** changes from type to type

String literal: sequence of characters in quotes
- Double quotes: `" abcex3$g<&"` or `"Hello World!"`
- Single quotes: `'Hello World!'`

Concatenation applies only to strings
- `"ab" + "cd"` evaluates to `"abcd"`
- `"ab" + 2` produces an **error**

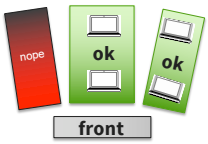
```
>>> terminal time >>>
```


Lectures

Lectures:
- Tuesday/Thursday 9:05
- Not just talking! Demos, clicker questions, *etc.*
- Slides posted to website afternoon before class

Please, no cell phones during lecture

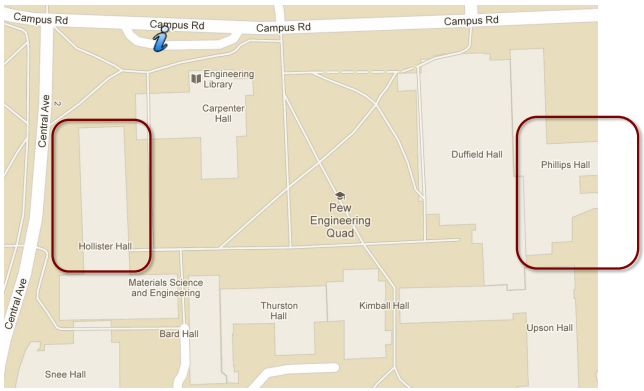
No laptop zone on one side. Please do not use your laptop there. We'll go over that again in 2 weeks in Bailey.



Lab Sections (aka Sections)

- guided exercises with TAs & consultants
- Start today: Tuesday, January 25
- **Go to the lab section you are registered for.** We can't maintain workable staff/student ratios otherwise.
- Handouts posted to the website the Monday before
- **Mandatory**. Missing > 4 units can lower your final grade.

Lab Locations



Phillips 318: lab computers provided (USB stick for files)
Hollister 401: BYOC ← please join if you have your own

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

Python. Necessary if using your own computer
- See course website for how to install
- For now? go to http://python.org & start typing!

Communication

cs1110-prof@cornell.edu
- Includes: professor & head TA
- **For sensitive correspondence**

cs1110-staff@cornell.edu
- Includes: professor, admin assistant, graduate TAs, head consultants
- **For time sensitive correspondence (i.e., emergencies)** Nobody at office hours; Lab has no printouts, *etc.*

Ed Discussion: not required, but fast

Canvas: official announcements posted here and emailed. (check your spam filters for mail from AWB93, LJL2, cs1110-prof or with [CS1110] in subject line)

All course materials on website: Canvas is 100% optional.