CS 1110, Lecture 2 Announcements

Sections

- **Start this week! Yay!**
- Please go only to the Section you are enrolled in
- Need to Change your Section *or your Lecture?*
  See our Section Swapping Station on Piazza:
  https://piazza.com/class/jckqwmqflaz6i?cid=10

Enrollment

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

http://www.cs.cornell.edu/courses/cs1110/2018sp/resources/alternatives.php

2

Things to do this week

Read textbook

- Chapter 2.1-2.3, 2-5
- Chapter 3.1-3.3

Lab 1:

- Go to your registered section
- Complete lab handout
- You have **one week** to show your work:
  - to TA by end of lab, or:
  - in consulting hours up to 1 day before your lab, or:
  - in TA (not professor) office hours
    (but student questions take precedence over this)
  - to TA **within first 10 minutes** of next week's lab

3

Helping you succeed in this class

Consultants. ACCEL Lab Green Room
- Daily office hours (see website) with consultants
- Very useful when working on assignments

ENGRG 1010: AEW Workshops. Additional discussion course open to **ALL** students
- Runs parallel to this class – optional
- See website

Piazza. Online forum to ask/answer questions
- Go here first before e-mailing questions

Office Hours. Talk to the professors!
- Olin 128 between lectures

4

From last time: Types

Type: set of values & operations on them

Type **float:**
- Values: real numbers
- Ops: +, -, *, /, **

Type **int:**
- Values: integers
- Ops: +, -, *, //, %, **

Type **bool:**
- Values: integers
- Ops: not, and, or

Type **str:**
- Values: string literals
  - Double quotes: **"abc"**
  - Single quotes: **'abc'**
- Ops: + (concatenation)

7

Operator Precedence

What is the difference between:

2*(1+3) — *add, then multiply*

2*1 + 3 — *multiply, then add*

Operations performed in a set order
- Parentheses make the order explicit

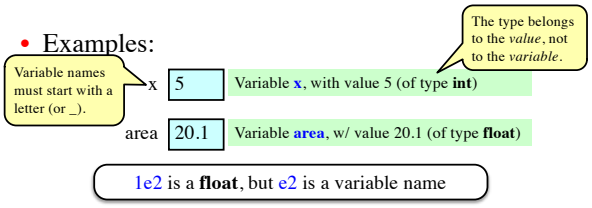
What if there are no parentheses?

→ **Operator Precedence:** fixed order to processes operators when no parentheses

13

In More Detail: Variables (Section 2.1)

- A **variable**
  - is a **named** memory location (**box**)
  - contains a **value** (in the box)
- Examples:



20

Expressions vs. Statements

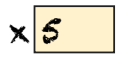
Expression	Statement
• **Represents** something ▪ Python *evaluates it* ▪ End result is a value	• **Does** something ▪ Python *executes it* ▪ Need not result in a value
• Examples: ▪ 2.3 (Value) ▪ (3+5)/4 (Complex Expression) ▪ x == 5	• Examples: ▪ x = 5

Look so similar
but they are not!

22

Execute the Statement: x = x + 2

Draw variable x on piece of paper:



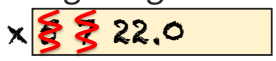
1. Evaluate the expression **x + 2**
   - For **x**, use the value in variable **x**
   - Write the expression somewhere on your paper
2. Store the value of the expression in **x**
   - Cross off the old value in the box
   - Write the new value in the box for **x**

Did you do the same thing as your neighbor ? If not, *discuss*.

28

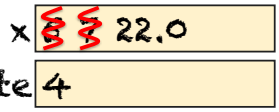
Exercise 1: Understanding Assignment

Begin with:



Declare a new variable:

```
>>> rate = 4
```



Execute this assignment:

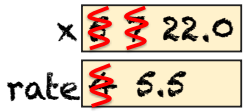
```
>>> rate = x / rate
```

Did you do the same thing as your neighbor ? If not, *discuss*.

35

Exercise 2: Understanding Assignment

Begin with:



Execute this assignment:

```
>>> rat = x + rate
```

Did you do the same thing as your neighbor ? If not, *discuss*.

38

Dynamic Typing

- Python is a **dynamically typed language**
  - Variables can hold values of any type
  - Variables can hold different types at different times
  - Use `type(x)` to find out the type of the value in x
- The following is acceptable in Python:

```
>>> x = 1          ← x contains an int value
>>> x = x / 2.0    ← x now contains a float value
```

- Alternative is a **statically typed language** (e.g. Java)
  - Each variable restricted to values of just one type

41

More Detail: Testing Types

- Command: type(<value>)
- Can test a variable:

```
>>> x = 5
>>> type(x)
<type 'int'>
```

- Can test a type with a Boolean expression:

```
>>> type(2) == int
True
```

42