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?

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

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

From last time: Types

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)

Operator Precedence

What is the difference between:
2*(1+3)  2*1 + 3
  
add, then multiply  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

In More Detail: Variables (Section 2.1)

• A variable
  * is a named memory location (box)
  * contains a value (in the box)

Examples:

Variable names must start with a letter (or _).

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>5</td>
</tr>
<tr>
<td>area</td>
<td>20.1</td>
</tr>
<tr>
<td>e2</td>
<td></td>
</tr>
</tbody>
</table>

The type belongs to the variable, not to the variable.

1e2 is a float, but e2 is a variable name
Expressions vs. Statements

<table>
<thead>
<tr>
<th>Expression</th>
<th>Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>• <strong>Represents</strong> something</td>
<td>• <strong>Does</strong> something</td>
</tr>
<tr>
<td>• Python <em>evaluates it</em></td>
<td>• Python <em>executes it</em></td>
</tr>
<tr>
<td>• End result is a value</td>
<td>• Need not result in a value</td>
</tr>
<tr>
<td>• Examples:</td>
<td>• Examples:</td>
</tr>
<tr>
<td>• 2.3</td>
<td>• x = 5</td>
</tr>
<tr>
<td>• (3+5)/4</td>
<td></td>
</tr>
<tr>
<td>• x == 5</td>
<td></td>
</tr>
</tbody>
</table>

Expressed vs. statements:
- **Expression**
  - Represents something
  - Python evaluates it
  - End result is a value
- **Statement**
  - Does something
  - Python executes it
  - Need not result in a value

Examples:
- 2.3
- (3+5)/4
- x == 5

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

Exercise 1: Understanding Assignment

Begin with:
- 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.

Exercise 2: Understanding Assignment

Begin with:
- `>>> x = x + 2`

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.

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