

Lecture 4:

USER INPUT

CSC111: Introduction to CS through Programming

R. Jordan Crouser

Assistant Professor of Computer Science

Smith College



CRYPTOCURRENCY AND BLOCKCHAIN WITH COINBASE

Have you ever heard of cryptocurrency, bitcoin, and blockchain? They are hot words in the FinTech industry. Don't really know what they are but want to learn more? Come to our cryptocurrency & blockchain session hosted by Coinbase! Want to learn about more opportunities at Coinbase? Come to the event! A Coinbase recruiter will hold an info session after the crypto workshop for internship and full time opportunities!

TIME:

September 19th, 2018 4:30-6:30pm

LOCATION:

Campus Center 204

HOST BY:

Coinbase

Pizza will be served!



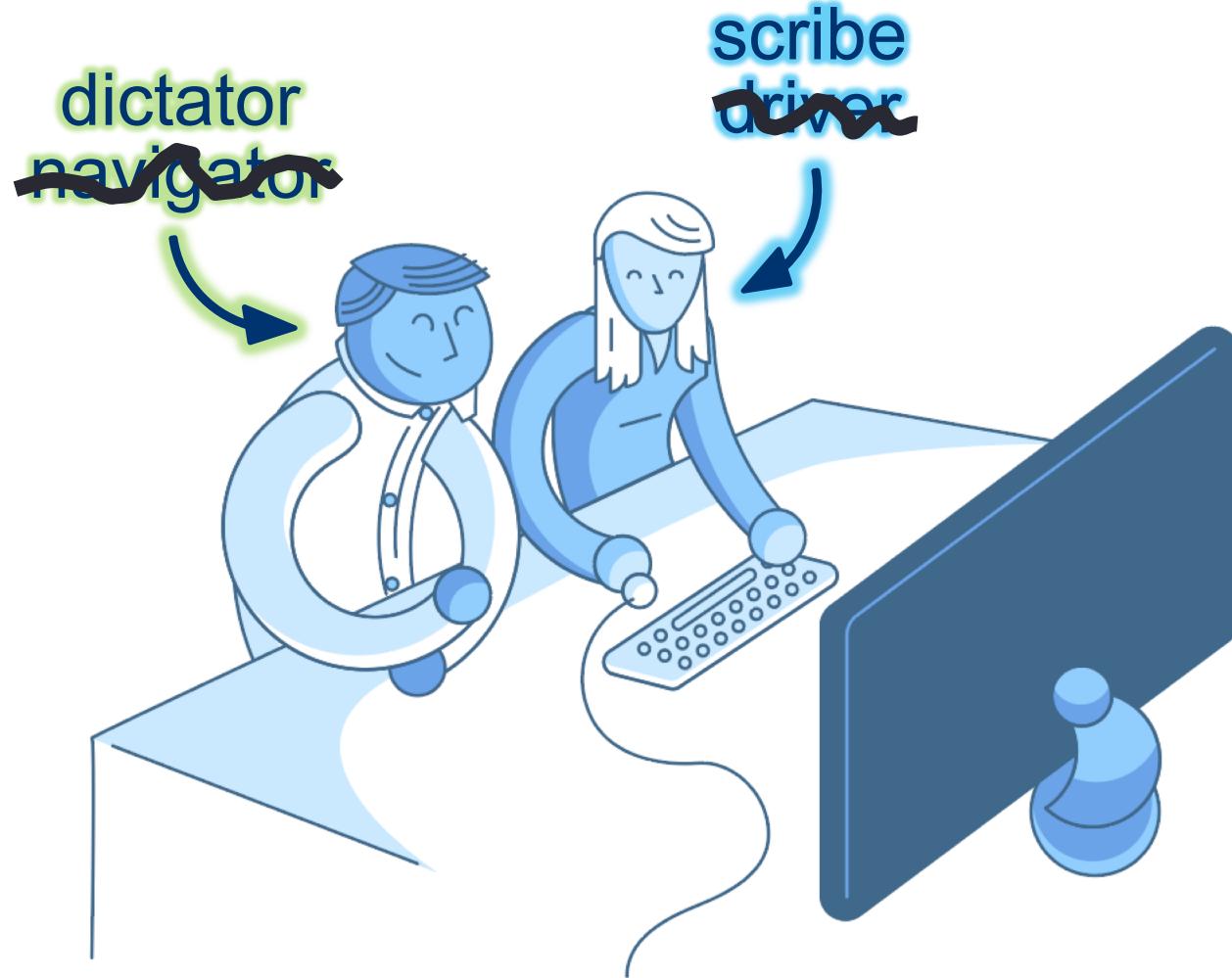


Microsoft®
New England Research
& Development Center

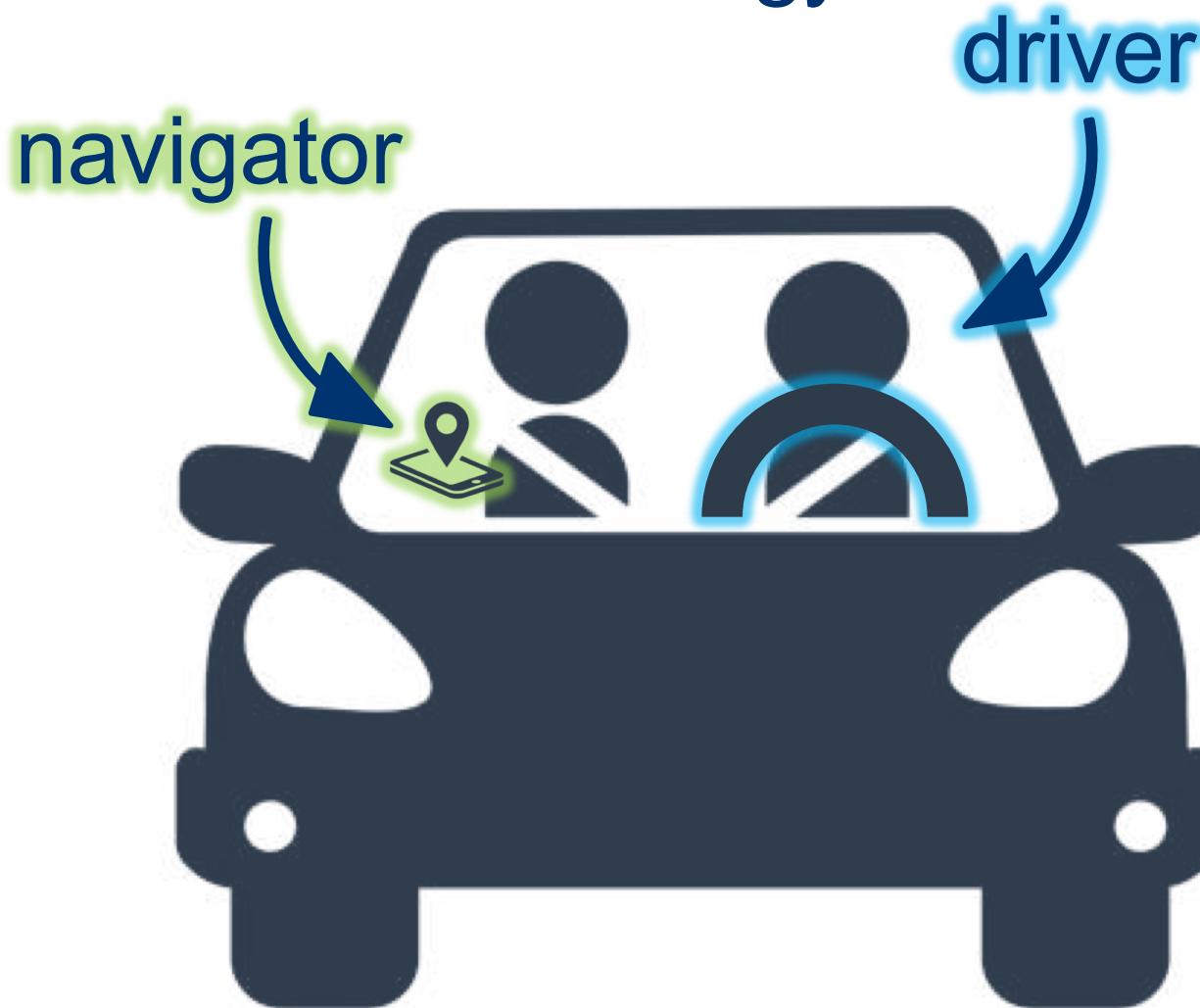
Student Day @ the NERD on Monday 9/24

- Tour the facility, visit with Boston-area Smith alums, see cool demos, check out the maker space, win cool prizes!
- Travel support / carpooling available
- Interested? Fill out the form posted to Slack or talk to Asmita (your CS Department Liaison)

A common misconception



Remember the analogy?



Discussion

So how is it really
supposed to work?



D E M O

T Y M E

Overview of the week

- ✓ Crash course in computers
 - ✓ A little history
 - ✓ 4 key components
 - ✓ Quick hardware demo
 - ✓ Boolean logic
- ✓ Introduction to Python
- ✓ Life skill #1: pair programming
- Lab: Getting Started with Python (**debrief**)
- User Input

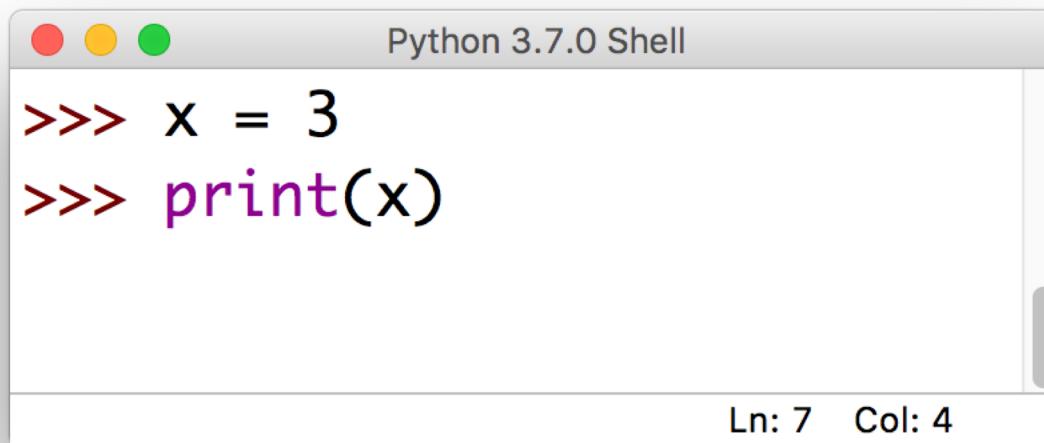
Lab 1: debrief

What do you remember
from this week's lab?



Recall: the **print()** function

- **print()** outputs information to the console ("the shell")
- Works on lots of different **data types** (strings, integers, floats, and many more!)
- When **print()** is called on ("passed") a **variable**, it outputs the **contents**



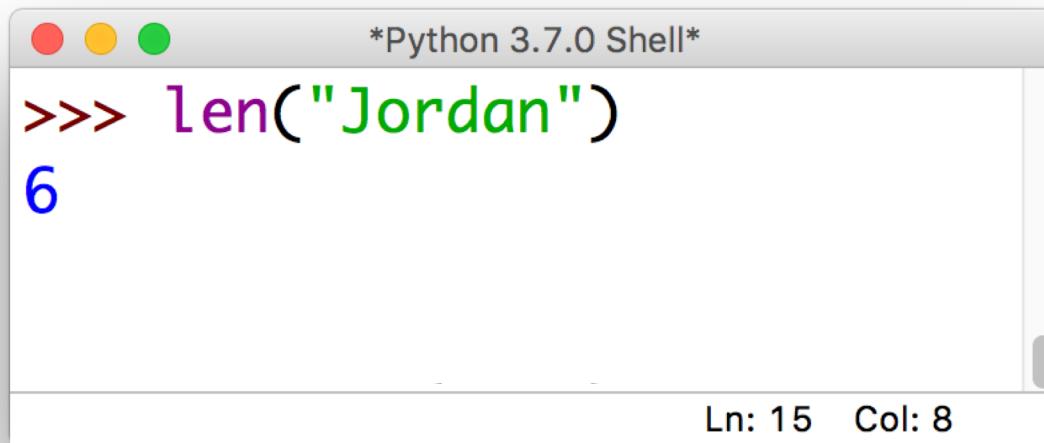
A screenshot of the Python 3.7.0 Shell window. The title bar says "Python 3.7.0 Shell". The main area contains the following code:

```
>>> x = 3
>>> print(x)
```

The status bar at the bottom right shows "Ln: 7 Col: 4".

The `len()` function

- `len()` takes in a string and gives back the string's length (number of characters, including spaces)
- Can be called on string **literals** (`"stuff in quotes"`) or on **variables** whose contents are strings
- Unlike `print()`, `len()` **returns** a value



A screenshot of the Python 3.7.0 Shell window. The title bar says "*Python 3.7.0 Shell*". The code input area shows the command `>>> len("Jordan")` followed by the output `6`. The status bar at the bottom right shows "Ln: 15 Col: 8".

```
>>> len("Jordan")
6
```

Refresher exercise

- Let's write a short program that prints the following:

#####

CSC 111

#####

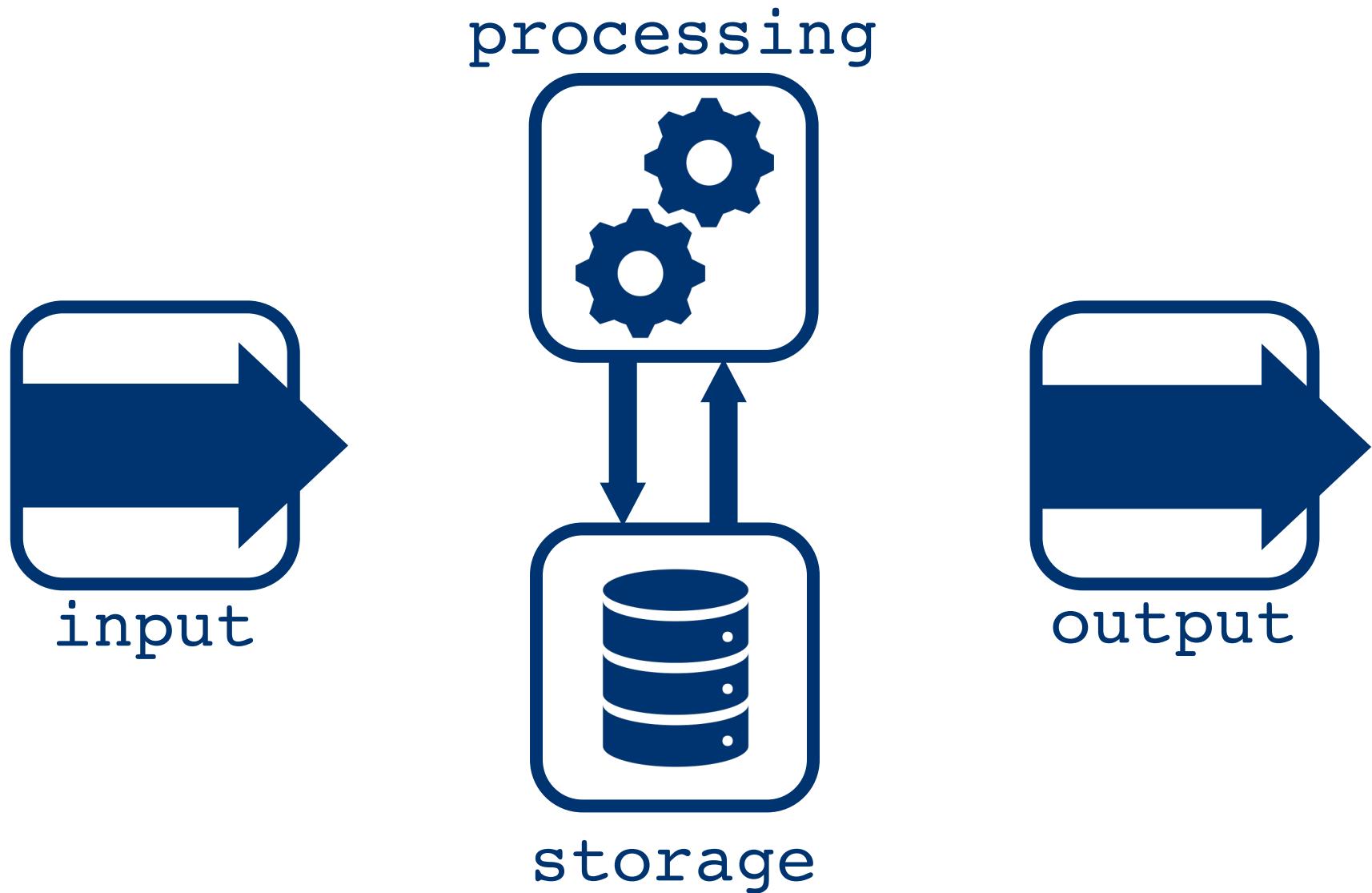
Discussion

What if we wanted to be able to
print a banner around ANY word?

What would we need?

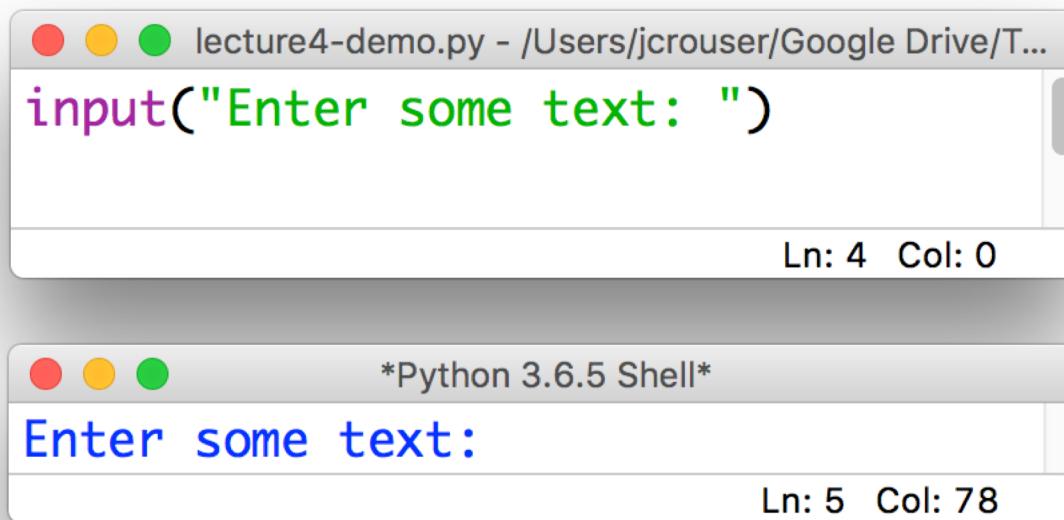


Back to the 4 basic tasks



The `input()` function

- Python has a built-in `input()` function that allows us to ask the user to type in information
- The `input()` function takes in a value, which will be printed to the console as a prompt:



A screenshot showing a Python code editor and a terminal window. The code editor window has a title bar with three colored dots (red, yellow, green) and the text "lecture4-demo.py - /Users/jcrouser/Google Drive/T...". It contains the following code:

```
input("Enter some text: ")
```

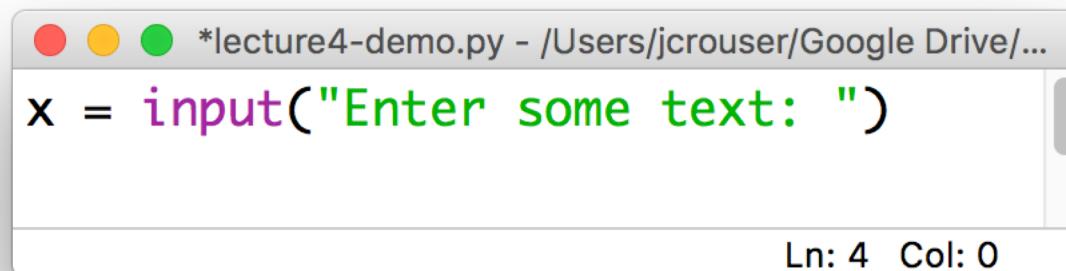
The terminal window below it has a title bar with three colored dots and the text "*Python 3.6.5 Shell*". It shows the output of the code execution:

```
Enter some text:
```

Below the terminal window, status bars indicate "Ln: 4 Col: 0" for the code editor and "Ln: 5 Col: 78" for the terminal.

The `input()` function

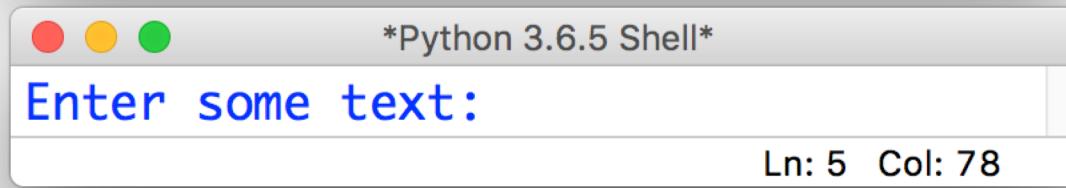
- In general, we will want to **save** what the user enters so we can do something with it
- This means we need to **assign** the value **returned** by the `input()` function to some variable, e.g.



A screenshot of a Python code editor window. The title bar says "*lecture4-demo.py - /Users/jcrouser/Google Drive/...*". The code in the editor is:

```
x = input("Enter some text: ")
```

In the bottom right corner of the editor window, it says "Ln: 4 Col: 0".



A screenshot of a Python shell window. The title bar says "*Python 3.6.5 Shell*". The prompt "Enter some text:" is displayed in blue text. In the bottom right corner of the shell window, it says "Ln: 5 Col: 78".

The **eval()** function

- The user's input is always returned as a **string**, even if they enter only numeric characters
- If we want Python to interpret it as a number, we can use the **eval()** function



A screenshot of a Python code editor window. The title bar shows the file name as "lecture4-demo.py - /Users/jcrouser/Google Drive/...". The main text area contains the following code:

```
x = eval(input("Enter some text"))
```

Below the code area, status information is displayed: "Ln: 3 Col: 0".

- Then we can manipulate **x** using mathematical operations

Quick exercise: putting it all together

- Find a partner, and write a program that asks the user to **input()** two strings:
 - a **word**
 - a **number**
- Store the user input in appropriate **variables** (remember: **eval()** will return the *numeric value* of a *string*)
- **print()** the **word** the user-specified **number** of times
- Want a **challenge**? Also ask the user to input a **character** (a single letter or symbol), and use that to print a banner around the repeated word

Quick exercise: putting it all together

- Find a partner, and write a program that asks the user to **input()** two strings:
 - a **word**
 - a **number**
- Store the user input in appropriate **variables** (remember: **eval()** will return the *numeric value* of a *string*)
- **print()** the **word** the user-specified **number** of times
- Want a **challenge**? Also ask the user to input a **character** (a single letter or symbol), and use that to print a banner around the repeated word like in lab

Discussion

What did you come up with?



Overview of the week

- ✓ Crash course in computers
 - ✓ A little history
 - ✓ 4 key components
 - ✓ Quick hardware demo
 - ✓ Boolean logic
- ✓ Introduction to Python
- ✓ Life skill #1: pair programming
- ✓ Lab: Getting Started with Python (debrief)
- ✓ User Input

Coming up next

- *A1: How Does it Do That?* is due **Sunday 11:55pm**
- **Mon 9/17:** more on numbers (ints vs. floats)
- **Weds 9/19:** mathematical operators
- **Lab:** Cash Machine
- **Fri 9/21:** Conditional (“if”) statements