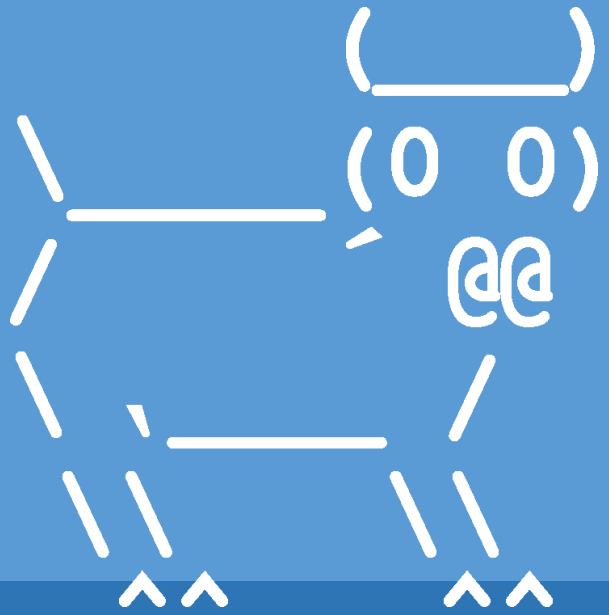
ASSORTED USEFUL CONCEPTS

WEEK-AT-A-GLANCE



Introduction to Computer Science

Iris Howley

LAST WEEK OF LECTURES!

Almost there!!

“Debug High, Debug Far, Your Goal the Script,
Your Aim the Star.”

Adapted from Hopkins Gate

HAPPENING THIS WEEK

- Quiz 3 is this **Friday, May 15**!
  - (*topics include up to, but not including recursion*)
  - Quiz 4 is Friday, May 22
- Homework 9 is due Monday, May 11 @11pm EST
  - The last homework!
- Lab 9 feedback was released last week
  - No more labs!
  - Lab 10 (extra credit) feedback will be posted later this week

THIS WEEK'S LESSON

Assorted Topics

(Some nifty things to know about python and being a computer scientist)

LECTURES THIS WEEK

- Monday
  - Week Overview
  - Persistent Data (pickling)
  - Pickling Twenty Questions
- Wednesday
  - Hashing
- Friday
  - Review
  - CS opportunities

Prior to lecture videos…

Complete:

1. POGIL Activities: Object Persistence
   - *available under Glow > Modules*
   - *also posted to the course website under Remote Lectures*

- Best done prior to watching lectures!
- Good for working with a partner (virtually, too!)
  - But will work without a partner, as well

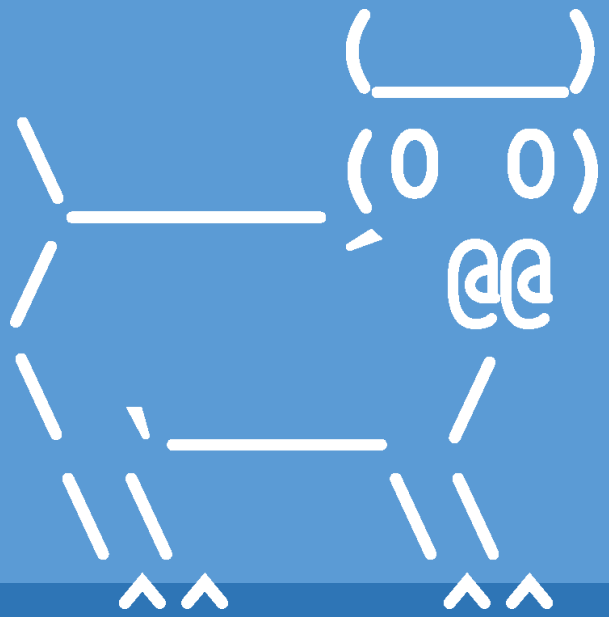
NO BOOK CHAPTERS THIS WEEK

Consult POGILs, slides, Lecture Notes

Highly recommended

Please contact me!

Persistent Data (pickling)



Introduction to Computer Science

Iris Howley

TODAY'S LESSON

Saving & Loading data between python sessions

(Sometimes you don't want to start from the beginning over and over again)

A typical python session

```
0 -> python3
1 >>> age = {'dizzy': 7, 'pixel': 1}
2 >>> age['dizzy']
3 7
4 >>> exit()

5 -> python3
6 >>> age['dizzy']
```

ERROR!

age not defined during this session!

Saving & Loading Data

```
0 -> python3
1 >>> age = {'dizzy': 7, 'pixel': 1}
2 >>> import pickle
3 >>> pickle.dump(age, open('save.pickle', 'wb'))
4 >>> exit()
5 -> python3
6 >>> import pickle
7 >>> newage = pickle.load( open('save.pickle',
'rb') )
8 >>> newage['dizzy']
9 7
```

Pickling – Saving to a file

```
pickle.dump(age, open('save.pickle', 'wb'))
```

- `pickle.dump(..)` will save the data, "dump" it to a file
- The first argument (`age`) is the data structure to write to a file
- Second argument is file to write out to
  - We've seen `open(..)` before!
  - First argument is the filename to save to
  - Second argument is the action, in this case **W**rite **B**ytecode

https://docs.python.org/3/library/pickle.html

Pickling – Loading from a file

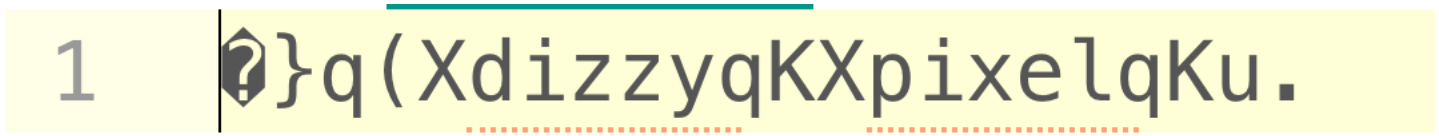
```
newage = pickle.load( open('save.pickle', 'rb') )
```

- `pickle.load(..)` will load data from a file
  - It returns the data, need to store it somewhere (`newage`)
- The first argument is file load from using `open(..)`
  - First argument is the filename to open
  - Second argument is the action, in this case **R**ead **B**ytecode

https://docs.python.org/3/library/pickle.html

Pickling – What does bytecode look like?

`'save.pickle'` opened in Atom:

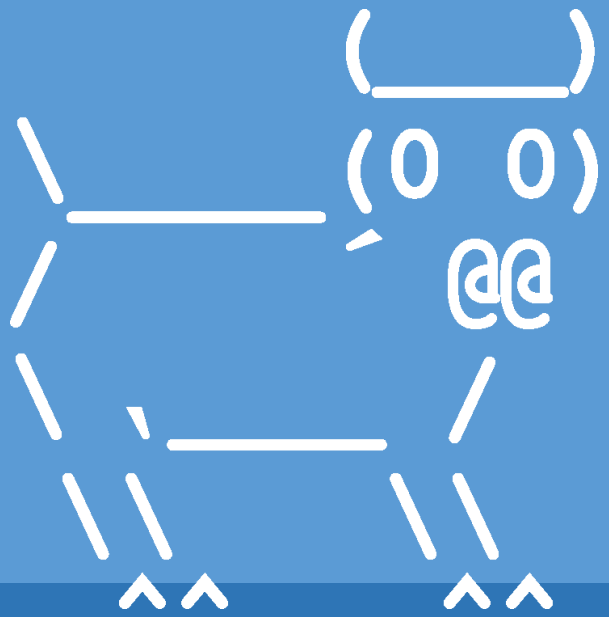


Pickling allows us to store objects by converting them to a byte stream for use later, much like placing a cucumber in a salt brine allows us to enjoy the pickle at a later time.

Persistent objects are those objects which survive between successive invocations of a program.

Please contact me!

Pickling a Binary Tree



Introduction to Computer Science

Iris Howley

TODAY'S LESSON

Pickling user-defined data types

(Saving & loading our Twenty Questions game from a previous lecture)

Modifying Twenty Questions

- See example code on the course website!

q20.py

Pickling – What does bytecode look like?

'db.pickle' opened in Atom:

```
��D�tree��Tree���)��N}�(�_value��is it an animal��_left�h)��N}�
(h�does it bork?�hh)��N}�(h�dog�hN�_right�Nu��bhh)��N}�(h�does it
 ribbit?�hh)��N}�(h�frog�hNhNu��bhh)��N}�(h�cat�hNhNu��bu��bu��bhh)
 ��N}�(h�is it drinkable?�hh)��N}�(h�tea�hNhNu��bhh)��N}�(h� a toaster
 �hNhNu��bu��bu��b.
```

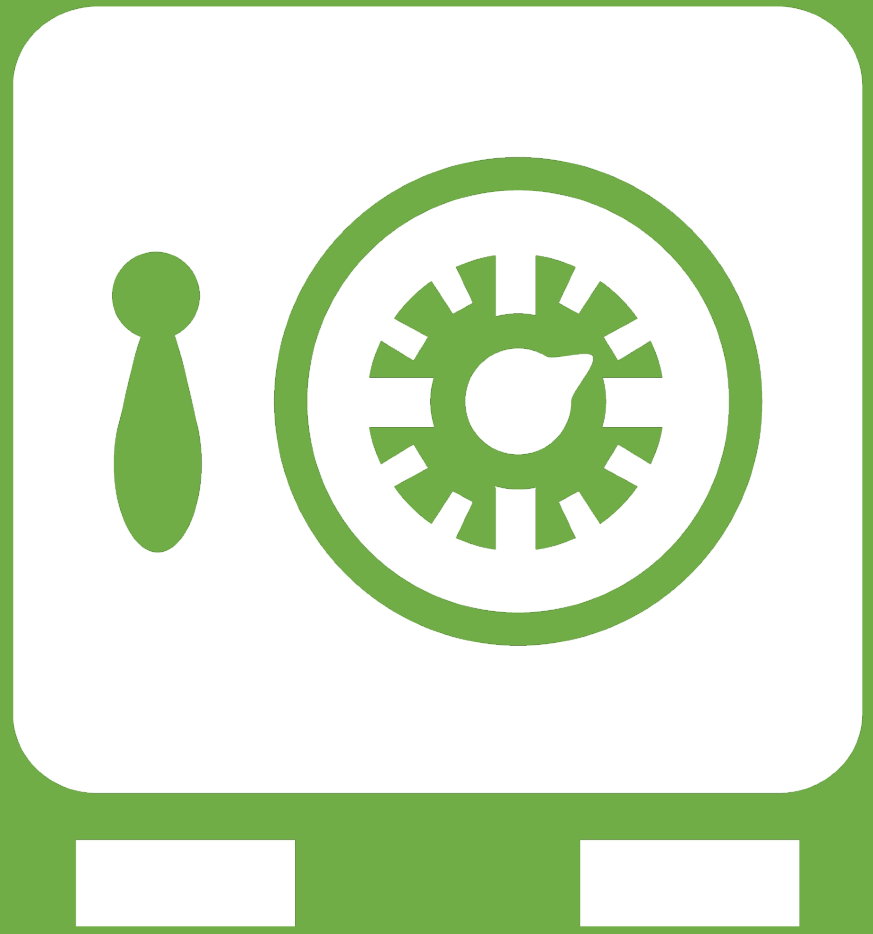
Terminal Input to Python

```
from sys import argv
filename = argv[1] if len(argv) > 1 else None
```

- `argv` is a special variable that stores the Terminal input
  - Whatever follows your 'python3' command is stored as a list of strings in `argv`
  - The first element of `argv` is the filename (i.e., `q20.py`)
- From Terminal: `-> python3 q20.py db`
  - `argv[0]` is `'q20.py'`
  - `argv[1]` is `'db'`

We check the length of `argv` to see if the user gave us a filename, or if we should use `None`

Please contact me!



Leftover Slides