DS2500
1/14-Tues.

Admin

- lab 1 yesterday!
- policy reminders: lab late policy
  due 9pm on same day
  late deadline (labs 1-3) Feb 7th 9pm ← hard deadline
- lecture question: bit.ly/ds2500_lecture_9
- please say your name!

Agenda

1. Classes + Objects (OOO)
2. Class syntax + vocab
3. Python

0. Lab Review

↳ problem 1, lab 1

given a list of strings
convert to ints, empty string to 0

```
def clean_empty(lst):
    ''' given a 1d list of strings, convert all values to ints,
        except empty string converts to zero'''
    new_lst = []
    for item in lst:
        val = int(item) if item != '' else 0
        new_lst.append(val)
    return new_lst
```

for item in lst: → iterate by value

val = int(item) if item != '' else 0 → shortcut for:

```
if item != '':
    val = int(item)
else
    val = 0
```

new_lst.append(val) ↳ add to list

return new_lst ↳ after for loop completed

return executes once per function

↳ def func():

return x ✓ then function ends

return y

1. Classes and Objects

↳ OOP is a programming paradigm

procedural ✓ ✓ ✓

object oriented
- group like elements together
- define how they interact

Python is object-oriented! everything is an object

How do we get there?

(ex) start with a list of dunkin coffee prices

[2.51, 2.99, 3.51]

{'s', 'm', 'l'}

⇝ dictionary of sizes:prices

{'s': 2.51,
'm': 2.99,
'l': 3.51}

⇝ How better?
- meaning behind the #s
- labels are attached to prices
- extensible for more kinds of drinks
- protection from accidental dupes

OOP builds on same idea ⇐

us coffee class

attributes (variables)
- size
- price
- type
- holiday_cup?
- temp
- sugar
- milk_alt

methods (function)
- ring up
- exchange

In Pycharm Project:

one file per class

class.py
- attributes
- methods

driver

w.py
- create objects
- call methods
- def main()

2. Class Syntax and Vocab

What we need to know:

- every class has exactly one **constructor**
  - create an instance of the class

```
def __init__(self, ....):
    # establish attributes
```

- self refers to current instance of class
  - first param to every method

Inside a class:

```
self.attr = 3
self.attr2 = 185
```

- every method has access to attributes we established in constructor

class: blueprint
object: active thing
w/ one class, make many objs

self.w attribute
w variable (not part of a class)

ex coffee.py

```
class Coffee:
    def __init__(self, size):
        self.size = size
        self.price = 2.50
    def meth(self):
        self.price += .50
```

driver.py

```
from coffee import Coffee

def main():
    c1 = Coffee('m')
    c2 = Coffee('l')
    c2.meth()
```

10:41

3. Python

this week's data: PWHL

↳ for each team, data for one season

Class HockeyTeam:

Attributes / methods?

- # wins
- # losses
- # ties?
- ticket sales
- # secs possession
- # shots, # points
- Home/Away games
- # power plays
- city
- starting lineup
  - ↳ Player class?

→ attrs

methods:

- scoring efficiency
- unit conversions
- plot the team