DS2000

10/18 - Fri :)

Admin

- HW4 due 10/19 9pm
- HW5 out, due 10/25 9pm
- mini exam #2 10/29

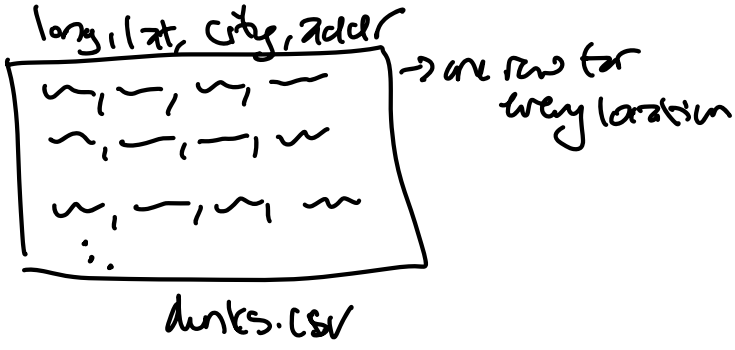
Agenda

1. Fixing tues code
2. 2D lists
3. Python

1. Perfect Tues. Code

tues data file: dunks in US

goals: plot locations
count # in given city
(split() is messy)



Improvements:

↳ plot everything at once

↳ organize code better?

↳ read in file better (csv library)

What do we need to plot everything all at once?

Where should we have functions?

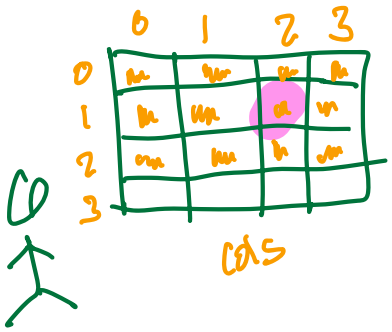
```
def main():
    # prompt the user for a city
    city = input("What city to look for Dunks in?\n")

    # start a counter varialbe at 0
    count = 0

    # step one - gather data by reading from the file
    with open(FILENAME, "r") as infile:
        infile.readline()

        for line in infile:
            # turn line into a list, because line is a string right now
            lst = process_str(line, LONG_POS, LAT_POS)

            # Is this dunks in the city the user gave us?
            if find_city(lst, city, CITY_POS):
                print(lst)
                count += 1

            # communication - plot the lat/long
            plt.plot(lst[LONG_POS], lst[LAT_POS], "x", color = "orange")

    # How many did I find in a given city?
    print(count, "Dunks in", city)

    # more communication, final touches and show the plot
    plt.title("Dunkin Donuts Locations in the US")
    plt.xlabel("Longitude")
    plt.ylabel("Latitude")
    plt.show()

main()
```

↳ create a list of lats/longs
cant # locs in a city
plotting?

↳ list of lats
list of longs
~ plt.scatter()
→ therefore, need to stop overwriting every time

2. 2D Lists

↳ table, matrix, spreadsheet...



raw 1, col 2

(need 2 positions to access a single value)

; rows first, column second

[[_, _, _, _],
[_, _, _, _],
[_, _, _, _]]

every raw is a list
lst[0] is a 1D list
lst[1][2] is a value
raw col

2D list is a data structure

- create
  - empty: lst = [ ]
  - non empty: lst = [[2,4], [6,8]] — 2 rows, 2 cols
- add something
  - ↳ add a list!
  - lst.append([10,12]) — adds a raw
- look at one thing
  - [ ] one list
  - [ ][ ] one value
  - lst[0] — raw 0 — [2,4]
  - lst[0][1] — raw, col — 4
- look at all the things
  - for loop to iterate (nested)
  - for raw in lst: → every raw
  - for num in raw: → every value in raw
  - by value →

Qs on 2D lists:

```
lst = [[1,3],
       [2,4]]
```

- to append a row

```
lst.append([-1,-3])
```

```
[[1,3],
 [2,4],
 [-1,-3]]
```

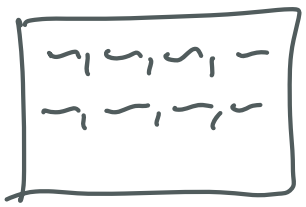
- rows don't need to be same length - but proceed cautiously! :)

- to append a value to a row

```
lst[0].append(5)
```

list!

today's code improvements



ducks.csv