DS2000
2/21 - Tues.

Admin

- HW5 due Fri
- Quiz 6 out 4:30pm → Fri 9:50AM
- mini uiz due 3/3

Agenda

1. Tuples
2. utility Functions
3. Nearest neighbor
4. Python

1. Tuples

- data structure: one label, many values
- Similar to lists, except...
  - create with ( ) not [ ]
  - tuples are immutable

Any new data structure:

1. Create —— tup = (4, 5, 6) val
   0 1 2 pos
2. Add → can't! ~~tup.append(7)~~ ✗✗✗✗ error
   tup[1] = 12 ✗✗✗✗ error
3. Look at one thing
   ↳ print(tup[1]) ✓ num = tup[2] + 4 ✓
4. Look at all the things
   ↳

```
for item in tup:
    print(item)
```

Usefulness of Tuples

1. protect our data
2. more efficient than lists
3. combine data in useful ways

```
def get_max(nn):
    ~~~~~~~~
    ~~~~~~~~
    ~~~~~~~~
    return thing1, thing2
```

```
val1, val2 = get_max(nn)
```

Secret Python:

```
tup = (thing1, thing2)
return tup
```

Secret python

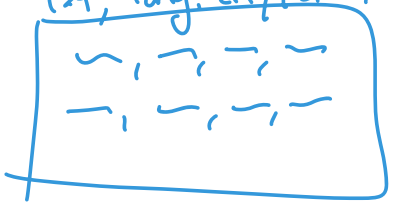
```
val1 = tup[0]
val2 = tup[1]
```

2. Utility Functions

- Tweak code from Friday
- Also in HW5 starter code

Last time...

lat, long, city, state, address



~~~.csv

→ 2D list

[lat, long, city, state, address]

Search: "Boston"

get: Boston, MA
Boston, IN

want: "Boston, MA"
~~~

Fix: use csv library

new python:

```
import csv
```

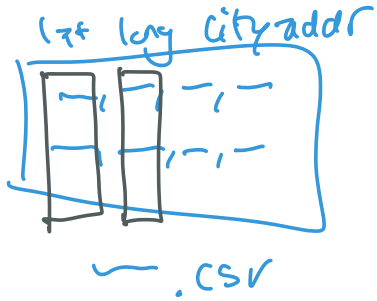
→ above main

→ read function

```
csvfile = csv.reader(infile)
next(csvfile)
```

} instead of readlines(), split()

Last time



want:

list of lats

list of longs

```
def col_to_lst(lst, col):
    new_lst = []
    for row in lst:
        new_lst.append(row[col])
    return new_lst
```

lst: 2D list; col: position

} save the same pos. in every row

→ 1D list

Nearest neighbor algorithm

- distance measure (ex: euclidean)
- Find closest physical thing (HW2)
- Find most similar thing (HW4)

Building block for: nearest neighbor classifier (ML)