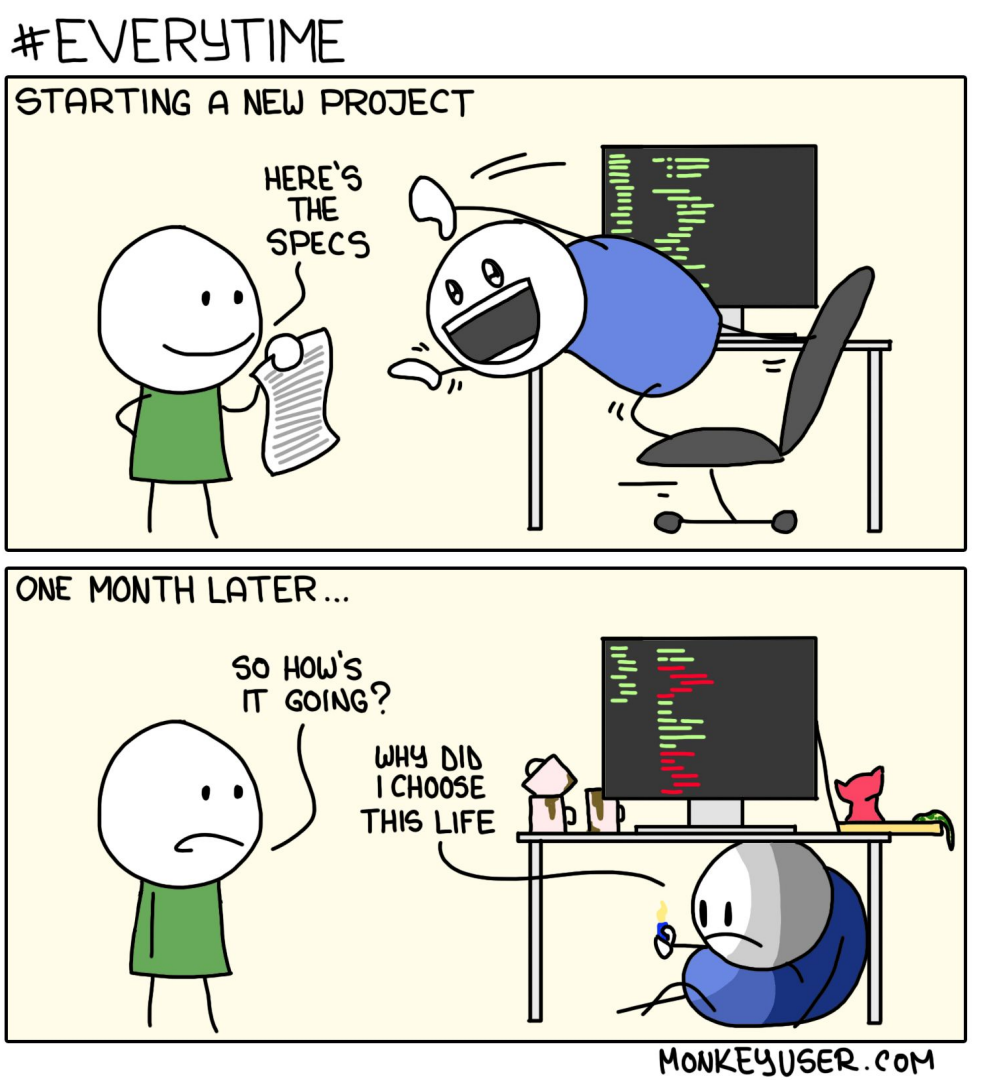
CSc 110

Tuples

Benjamin Dicken



Tuple

Tuples are similar to lists, except for 1 key difference.

What is that difference?

Why tuples?

What will happen?

```
employees = {}

e = ['USA', 'Engineering', 105234]

employees[e] = ['John', 'West', 60000]

print(employees)
```

What will happen?

```
employees = {}

e = ('USA', 'Engineering', 105234)

employees[e] = ['John', 'West', 60000]
```

What will happen?

```
employees = set()

e = ['USA', 'Engineering', 105234, 'Ian James']

employees.add(e)
```

What will happen?

```
employees = set()

e = ('USA', 'Engineering', 105234, 'Ian James')

employees.add(e)
```

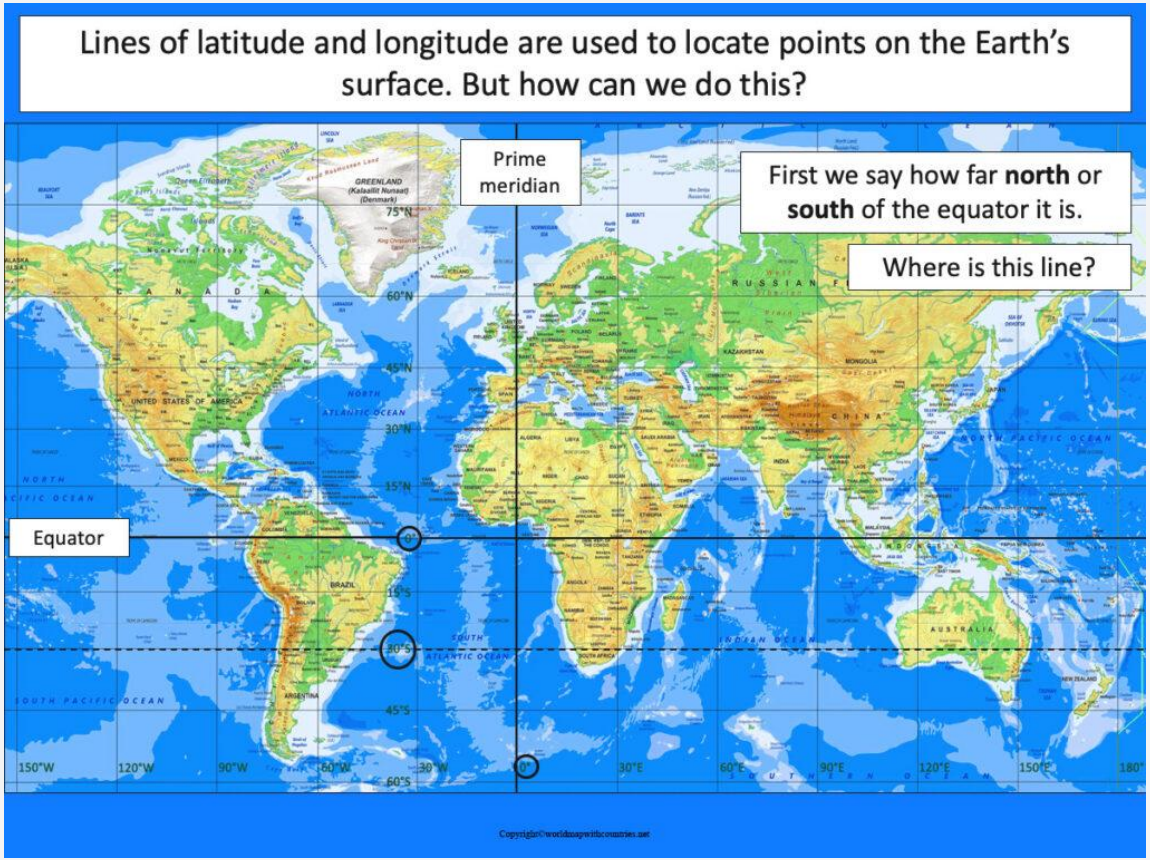
Mutability!

- Lists are **mutable**
- Tuples are **immutable**
- Only immutable types can be a dictionary key
- Only immutable types can be in a set

Why tuples then?

- Can be used as keys in a dictionary
- Can be added to a set

Latitude and Longitude



Latitude and Longitude

- Can be used to specify location on the map
- A lat/long coordinate is composed of (at least) two parts
  - The lat and long angles
- For instance:
  - Enter **34°S 151°E** in the search on https://www.google.com/maps

Latitude and Longitude

- Can be used to specify location on earth!
- A lat/long coordinate is composed of (at least) two parts
  - The lat and long angles
- For instance:
  - Enter **34°S 151°E** in the search on https://www.google.com/maps
  - Can we provide more specificity?

Latitude and Longitude

- Can be used to specify location on earth!
- A lat/long coordinate is composed of (at least) two parts
  - The lat and long angles
- For instance:
  - Enter **32°N 111°W** in the search on https://www.google.com/maps
  - Can we provide more specificity?
    - Try: **32° 13' 47" N 110° 57' 18" W**

What data structure(s) would you use to store the relationship between a city name and a lat/long specification?

City name to lat/long

```
city_to_ll = {
   'Aberdeen': [57, 9, 'N', 2, 9, 'W'],
   'Beijing' : [39, 55, 'N', 116, 25, 'E'],
   . . .
   'Osaka' : [34, 32, 'N', 135, 30, 'E'] }
```

What data structure(s) would you use to store the relationship between lat/long and city name?

Lat/long to city name

```
ll_to_city = {
  [57, 9, 'N', 2, 9, 'W'] : 'Aberdeen',
  [39, 55, 'N', 116, 25, 'E'] : 'Beijing',
  . . .
  [34, 32, 'N', 135, 30, 'E'] : 'Osaka'}
```

Lat/long to city name

```
ll_to_city = {
   '57 9 N 2 9 W' : 'Aberdeen',
   '39 55 N 116 25 E' : 'Beijing',
   . . .
   '34 32 N 135 30 E' : 'Osaka'}
```

Lat/long to city name

```
ll_to_city = {
   (57, 9, 'N', 2, 9, 'W') : 'Aberdeen',
   (39, 55, 'N', 116, 25, 'E') : 'Beijing',
   . . .
   (34, 32, 'N', 135, 30, 'E') : 'Osaka'}
```

Implement add_ll

```
def add_ll(ll_to_city, ll_string, city_name):
    '''
    Add a new mapping of a lat/long to a city. Assume that:
      ll_to_city: a dictionary mapping tuples of
                  lat/long to city names
      ll_string: a lat/long string. For instance:
                 '40 50 N 30 12 E'
      city_name: the name of the city to map to
    '''
```

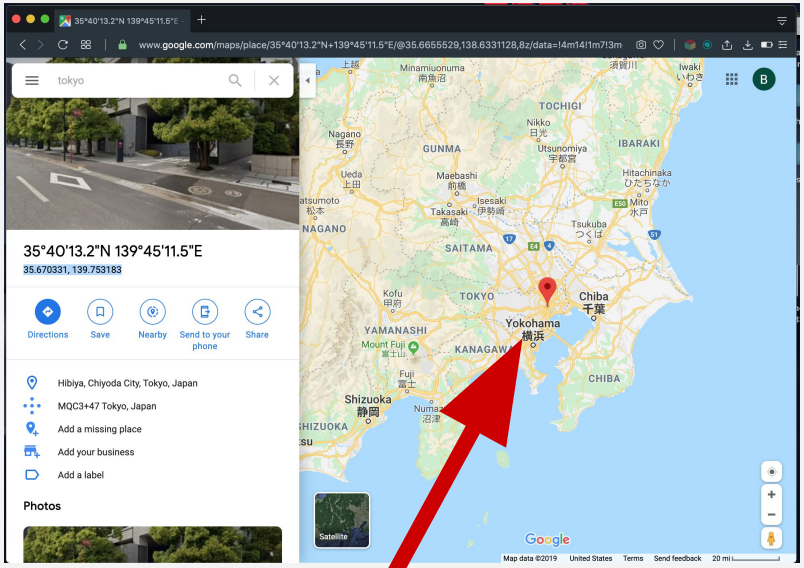
Implement get_city

```
def get_city(ll_to_city, ll_string):
    '''
    Get the city name that corresponds to the lat/long
    that the ll_string parameter variable references.
    If ll_to_city doesn't have the lat/long info, say
    "not sure".
      ll_to_city: a dictionary mapping tuples of
                  lat/long to city names
      ll_string: a lat/long string. For instance:
                 '40 50 N 30 12 E'
    '''
```

Program to take to a city

locations.csv

```
Sydney,-33.866402,151.056558
Tucson,32.217219,-110.946265
Rome,41.894099,12.497515
Jerusalem,31.764265,35.212615
Tokyo,35.670331,139.753183
```



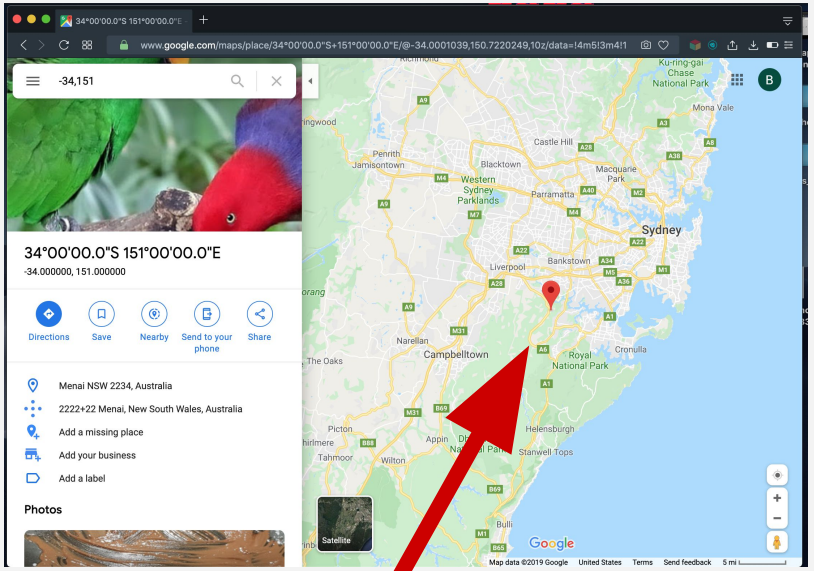
locations.py

```
What city would you like to see on a map? Tokyo
```

Program to take to a city

The webbrowser module

```
import webbrowser
. . .
webbrowser.open(url)
```



Google maps URL format

```
https://www.google.com/maps/search/?api=1&query=-34,150
```

Implement get_location_info

```
def get_location_info():
    '''
     (A) Open a file named locations.csv, formatted like:
           Sydney,-33.866402,151.056558
           Tucson,32.217219,-110.946265
           . . .
     (B) Loads them into a dictionary that maps city name to
         a tuple of the lat and long
     (C) Return the dictionary
    '''
```

Implement get_location_info

```
def get_location_info():
    location_info = {}
    location_file = open('locations.csv', 'r')
    for line in location_file:
        values = line.split(',')
        location_info[values[0]] = (values[1], values[2])
    return location_info
```

```
import webbrowser

def get_location_info():
    location_info = {}
    location_file = open('locations.csv', 'r')
    for line in location_file:
        values = line.split(',')
        location_info[values[0]] = (values[1], values[2])
    return location_info

def main():
    locations = get_location_info()
    url_base = 'https://www.google.com/maps/search/?api=1&query='
    city = input('What city would you like to see on a map? ')
    if city in locations:
        url = url_base + locations[city][0] + ',' + locations[city][1]
        webbrowser.open(url)
    else:
        print('I don\'t know where that city is.')

main()
```