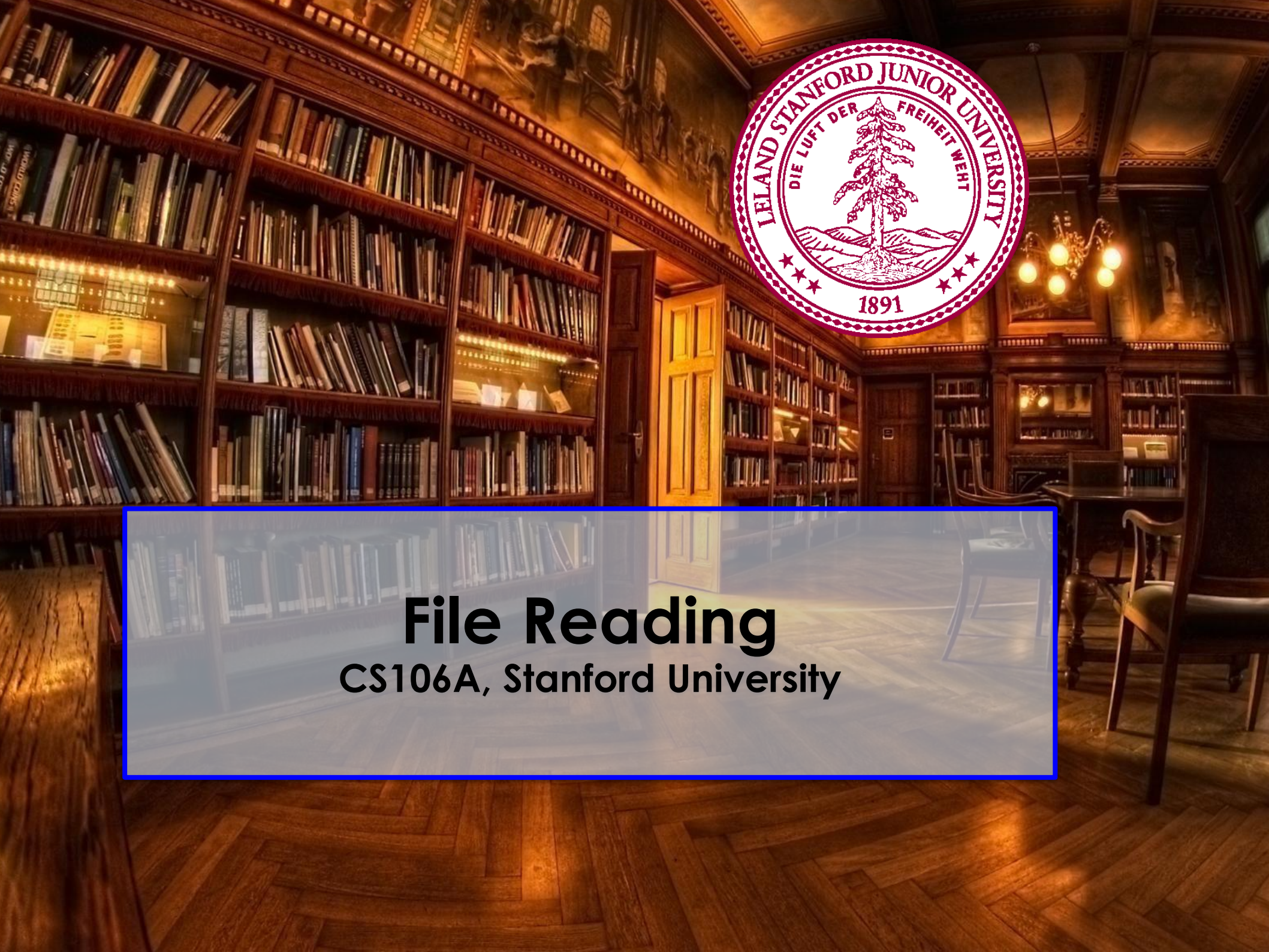
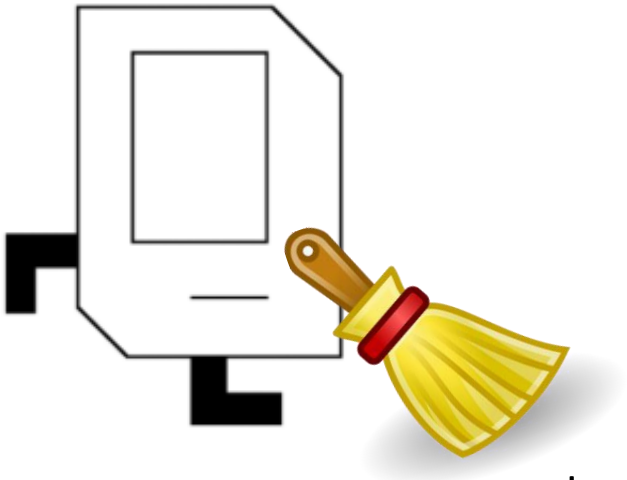


File Reading

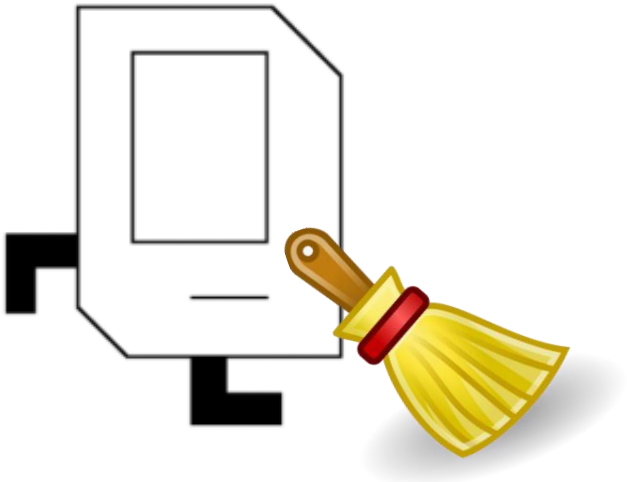
CS106A, Stanford University

Housekeeping I



- Midterm will be on Tuesday, May 3rd from 7pm-9pm
  - Location by first letter of last name:
    - Last name: A-H in room: 420-040
    - Last name: I-Z in room: CEMEX Aud
  - Midterm covers material through class on April 27
  - Open book/notes, but will need to bring printed reference material
  - Students with OAE accommodations should have gotten email from Juliette with exam information
  - Students with conflicts should have gotten email from me with exam information

Housekeeping II

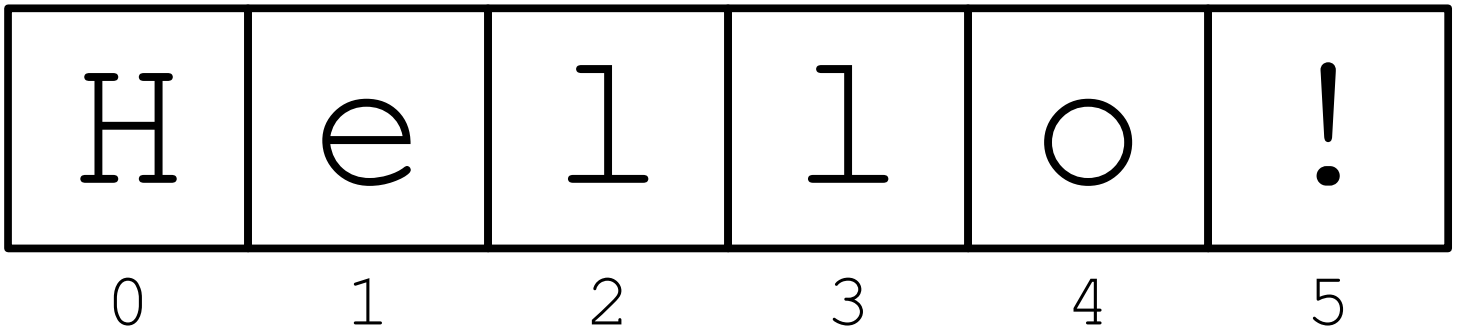


- Midterm review: TODAY, 1:30-2:30pm in Nvidia Aud
- Will be using BlueBook software to take the exam
  - BlueBook will not allow you to access other applications/files on your computer during the exam
- Practice midterm available on class website
  - Will give you practice with topics on actual exam
  - Great way to prepare and also test your BlueBook set-up
- Reminder: Please wear a mask in class

Strings review

Strings are similar to a list of characters

```
def main():
    text = "hello!"
```



***text*[*index*]**

reverse_string

```
def reverse_string(str):
    result = ""
    for i in range(len(str)):
        result = str[i] + result
    return result


def reverse_string_v2(str):
    result = ""
    for ch in str:
        result = ch + result
    return result


def reverse_string_v3(str):
    return str[::-1]
```

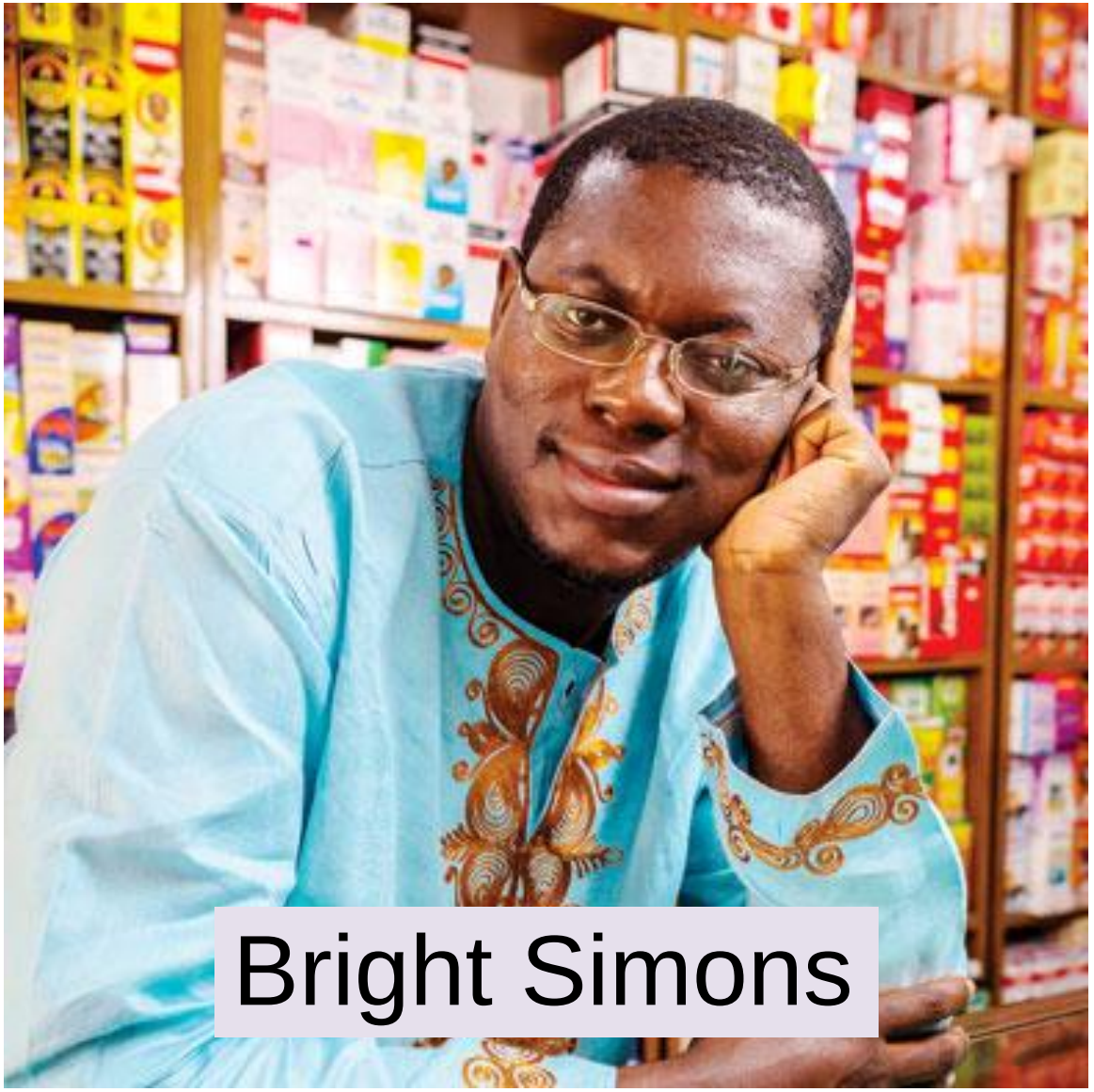
Palindrome

- A ***palindrome*** is a string that reads the same forwards and backwards (ignoring punctuation)
- For example:
  - Abba
  - Racecar
  - Kayak
  - Mr. Owl ate my metal worm.
  - Go hang a salami! I'm a lasagna hog.
  - Elu par cette crapule

Let's take it out for a spin!

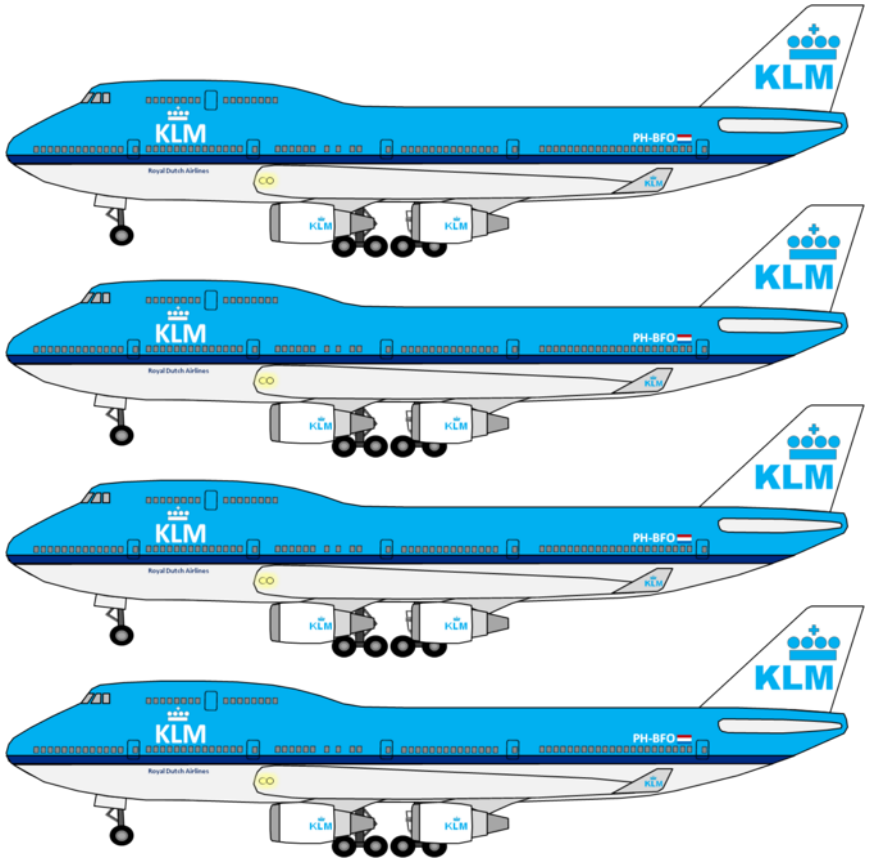
palindrome.py

Saving Lives with Strings



Fake Medicine was a Problem

700,000 deaths a **year** from **fake** malaria and tuberculosis drugs [1]



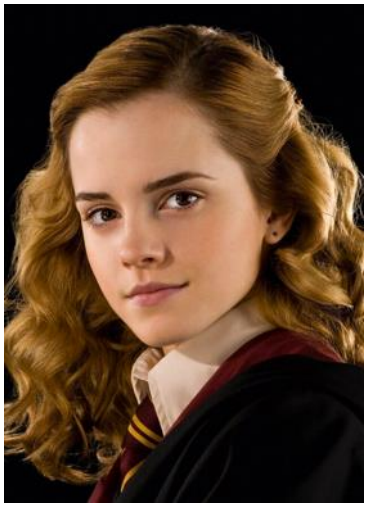
Equivalent of this many crashes per day

[1] http://www.un.org/africarenewal/magazine/may-2013/counterfeit-drugs-raise-africa%E2%80%99s-temperature

Underlying Puzzle

Counterfeiter

User

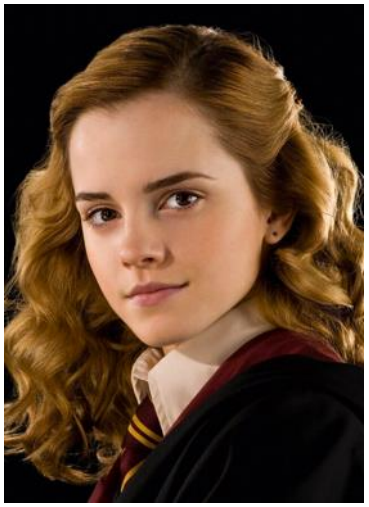


You (Distributor)

Underlying Puzzle

Counterfeiter

User



You (Distributor)

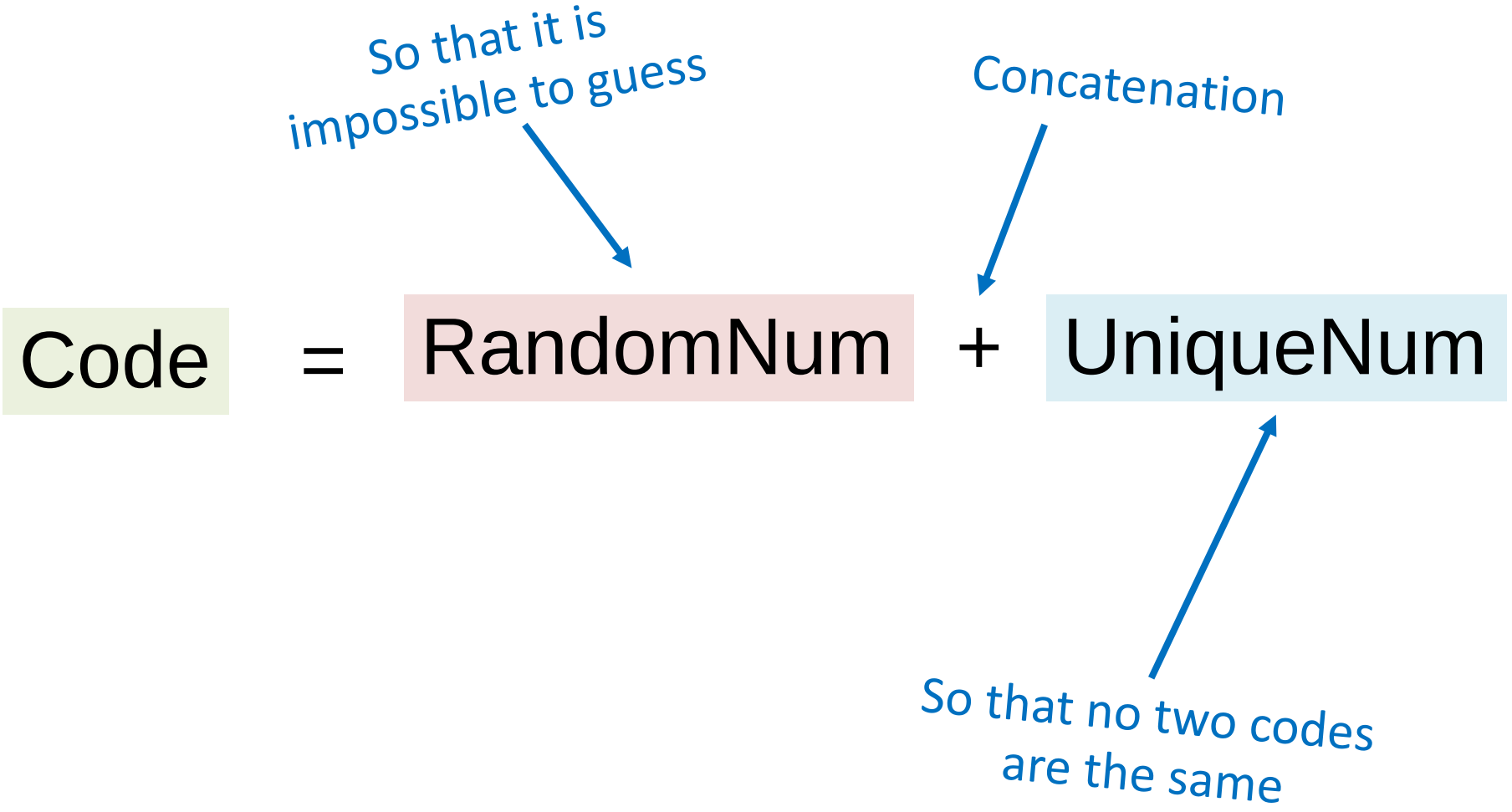
Make a code to
put on every box


Artelum®
Artemether Lumefantrin
40 /240 mg
12 Tablets
Anti-malarial agent
M&B
342302033
BlackBerry
151 / 1 of 6
To: 1393
To:
342302033

1. Unique

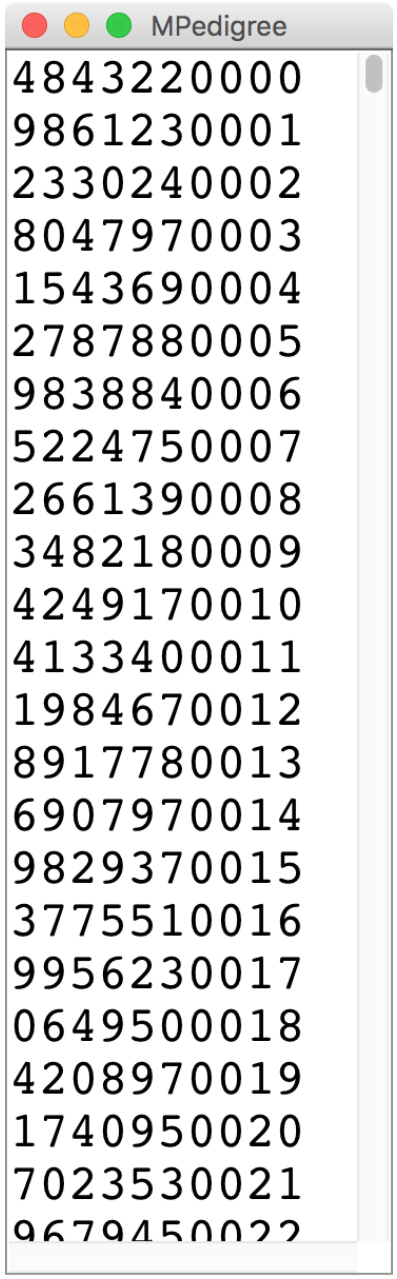
2. Impossible to guess

Insight


So that it is impossible to guess
Concatenation
Code
=
RandomNum
+
UniqueNum
So that no two codes are the same

M-Pedigree

Every string should be the same length




```
N_LABELS = 100
MAXNUM = 999999


def main():
  # prints a set of unique labels
  for i in range(N_LABELS):
    rand_value = random.randint(0, MAXNUM)
    rand_part = pad(rand_value, 6)
    unique_part = pad(i, 4)
    print(rand_part + unique_part)


# prepends 0s to string version of number
# until the str has length goal_length
def pad(number, goal_length):
  number_string = str(number)
  while len(number_string) < goal_length:
    number_string = '0' + number_string
  return number_string
```

End Review

Learning Goals

1. Know how to read a file line by line.

Getting Data into Programs

- Put it directly in the program:
  - Define constants holding your values.
- Get it from the user:
  - **`input`**, etc.
- Generate it randomly:
  - Use random.
- Get it from an external source.
  - Store it in a file and read it later.

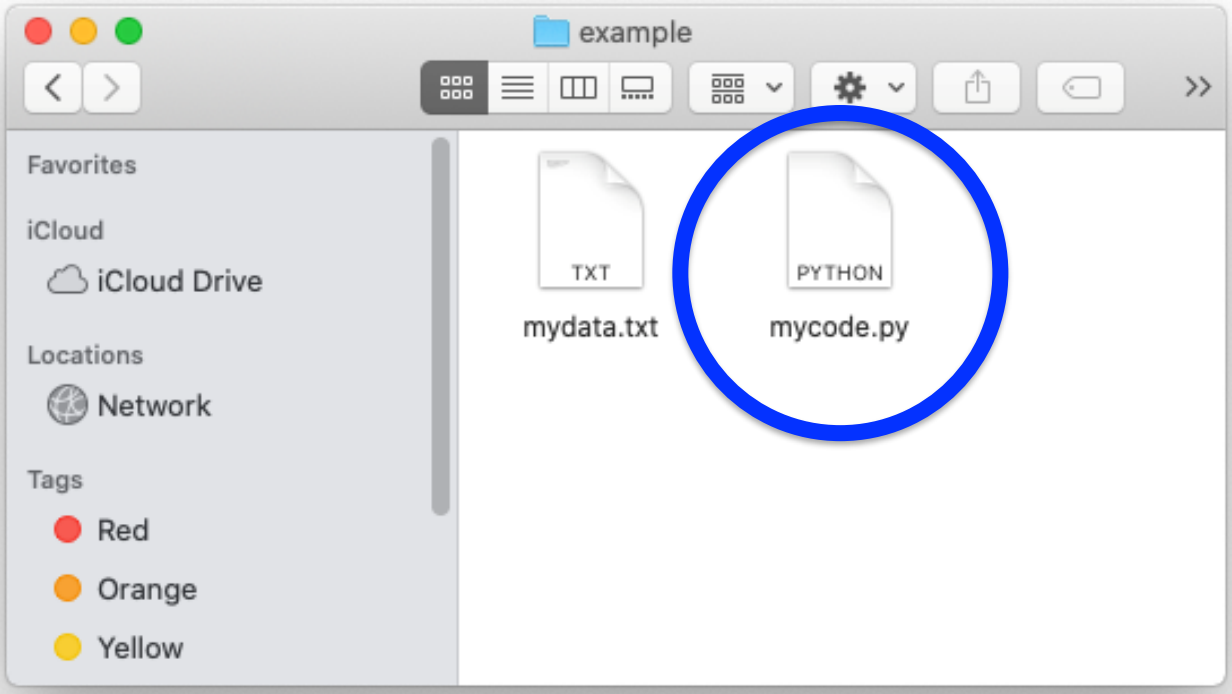
Reading Files

- Virtually all programs that you've used at some point read files from disk:
  - Word processing (documents)
  - Web browser (cookies)
  - Games (saved progress)
  - PyCharm (Python files)
  - Music player (songs)

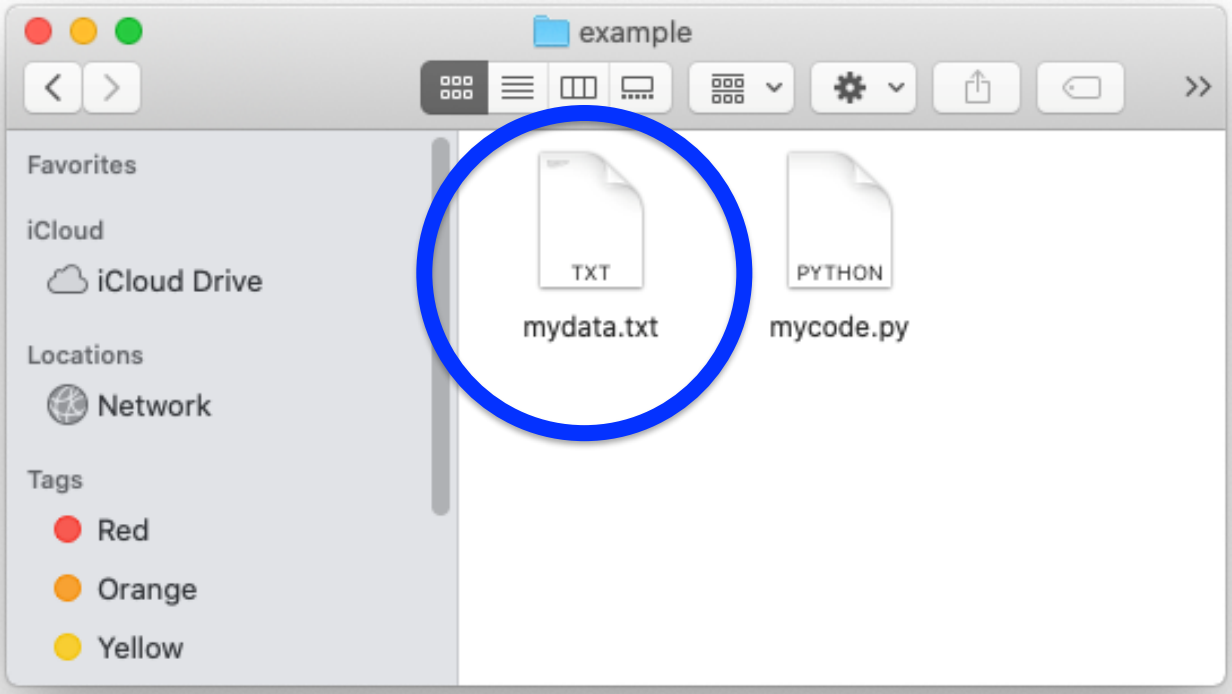
The structure of files

- A file is just a series of ***bits*** (ones and zeros).
  - "bit" is short for "binary digit"
- Those bits can have structure:
  - Plain-text: Bits represent characters (e.g., ASCII)
  - JPEG: Bits encode information about the structure of an image.
  - MP3: Bits encode frequency information about music.
  - etc.

What is a file?

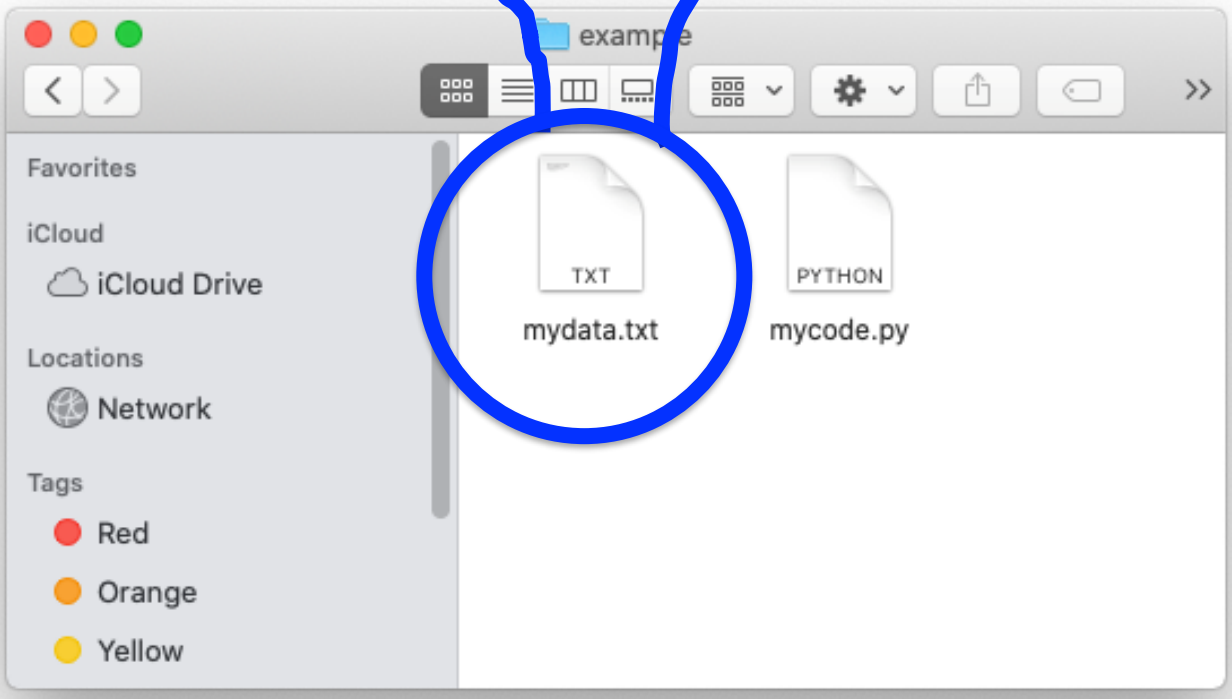


What is a file?



```
Happy day, CS106A!
You're awesome.
We believe in you!
```

mydata.txt



```
Happy day, CS106A!
You're awesome.
We believe in you!
```

mydata.txt

```
Happy day, CS106A!
You're awesome.
We believe in you!
```

mydata.txt

```
def main():
    with open('mydata.txt', 'r') as file:
        for line in file:
            print(line)
```

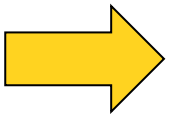
```
Happy day, CS106A!
You're awesome.
We believe in you!
```

mydata.txt

```
def main():
    with open('mydata.txt', 'r') as file:
        for line in file:
            print(line)
```

Step One:

Open the file in *reading* mode



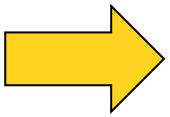
```
Happy day, CS106A!
You're awesome.
We believe in you!
```

mydata.txt

```
def main():
    with open('mydata.txt', 'r') as file:
        for line in file:
            print(line)
```

Step Two:

Read the file one line at a time

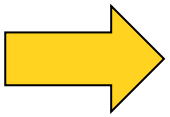


```
Happy day, CS106A!
You're awesome.
We believe in you!
```

mydata.txt

```
def main():
    with open('mydata.txt', 'r') as file:
        for line in file:
            print(line)
```

console



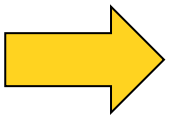
```
Happy day, CS106A!
You're awesome.
We believe in you!
```

mydata.txt

```
def main():
    with open('mydata.txt', 'r') as file:
        for line in file:
            print(line)
```

console

```
Happy day, CS106A!
```



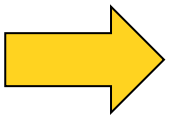
```
Happy day, CS106A!
You're awesome.
We believe in you!
```

mydata.txt

```
def main():
    with open('mydata.txt', 'r') as file:
        for line in file:
            print(line)
```

console

```
Happy day, CS106A!
```



```
Happy day, CS106A!
You're awesome.
We believe in you!
```

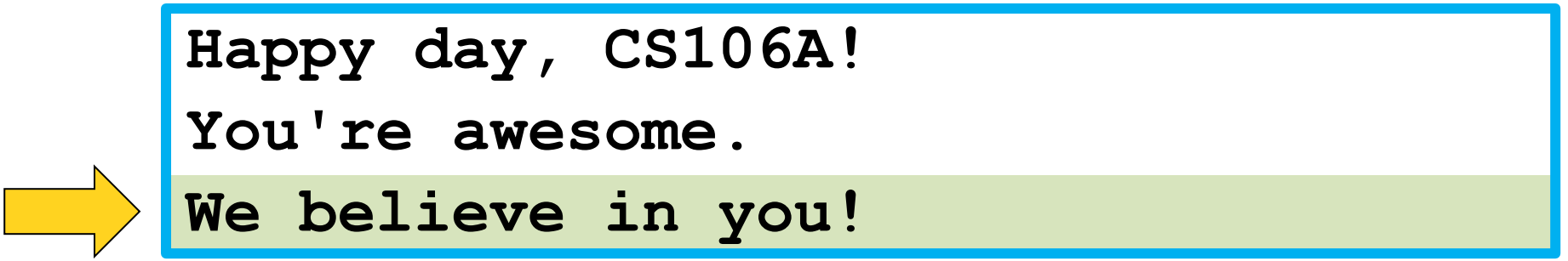
mydata.txt

```
def main():
    with open('mydata.txt', 'r') as file:
        for line in file:
            print(line)
```

console

```
Happy day, CS106A!

You're awesome.
```



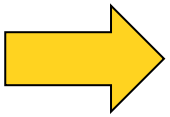
mydata.txt

```
def main():
    with open('mydata.txt', 'r') as file:
        for line in file:
            print(line)
```

console

```
Happy day, CS106A!

You're awesome.
```



```
Happy day, CS106A!
You're awesome.
We believe in you!
```

mydata.txt

```
def main():
    with open('mydata.txt', 'r') as file:
        for line in file:
            print(line)
```

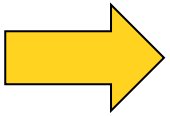
console

```
Happy day, CS106A!

You're awesome.

We believe in you!
```

```
Happy day, CS106A!
You're awesome.
We believe in you!
```



mydata.txt

```
def main():
    with open('mydata.txt', 'r') as file:
        for line in file:
            print(line)
```

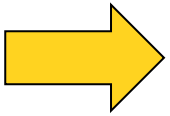
console

```
Happy day, CS106A!

You're awesome.

We believe in you!
```

```
Happy day, CS106A!
You're awesome.
We believe in you!
```



mydata.txt

```
def main():
    with open('mydata.txt', 'r') as file:
        for line in file:
            print(line)
```

Done!

console

```
Happy day, CS106A!

You're awesome.

We believe in you!
```

```
Happy day, CS106A!
You're awesome.
We believe in you!
```

mydata.txt

```
def main():
    with open('mydata.txt', 'r') as file:
        for line in file:
            print(line)
```

console

```
Happy day, CS106A!

You're awesome.

We believe in you!
```

```
Happy day, CS106A!
You're awesome.
We believe in you!
```

mydata.txt

line `'Happy day, CS106A!\n'`

```
def main():
    with open('mydata.txt', 'r') as file:
        for line in file:
            print(line)
```

console

```
Happy day, CS106A!

You're awesome.

We believe in you!
```

Take 2

```
Happy day, CS106A!
You're awesome.
We believe in you!
```

mydata.txt

```
def main():
    with open('mydata.txt', 'r') as file:
        for line in file:
            line = line.strip()
            print(line)
```

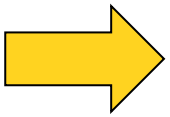
console

```
Happy day, CS106A!
You're awesome.
We believe in you!
```

mydata.txt

```
def main():
    with open('mydata.txt', 'r') as file:
        for line in file:
            line = line.strip()
            print(line)
```

console



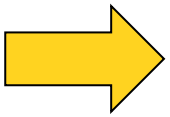
```
Happy day, CS106A!
You're awesome.
We believe in you!
```

mydata.txt

line `'Happy day, CS106A!\n'`

```
def main():
    with open('mydata.txt', 'r') as file:
        for line in file:
            line = line.strip()
            print(line)
```

console



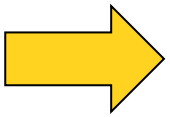
```
Happy day, CS106A!
You're awesome.
We believe in you!
```

mydata.txt

line `'Happy day, CS106A!'`

```
def main():
    with open('mydata.txt', 'r') as file:
        for line in file:
            line = line.strip()
            print(line)
```

console



```
Happy day, CS106A!
You're awesome.
We believe in you!
```

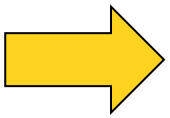
mydata.txt

line `'Happy day, CS106A!'`

```
def main():
    with open('mydata.txt', 'r') as file:
        for line in file:
            line = line.strip()
            print(line)
```

console

```
Happy day, CS106A!
```



```
Happy day, CS106A!
You're awesome.
We believe in you!
```

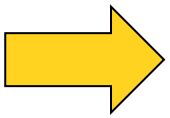
mydata.txt

line `'You're awesome.\n'`

```
def main():
    with open('mydata.txt', 'r') as file:
        for line in file:
            line = line.strip()
            print(line)
```

console

```
Happy day, CS106A!
```



```
Happy day, CS106A!
You're awesome.
We believe in you!
```

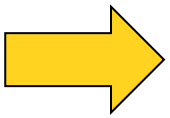
mydata.txt

line `'You're awesome.'`

```
def main():
    with open('mydata.txt', 'r') as file:
        for line in file:
            line = line.strip()
            print(line)
```

console

```
Happy day, CS106A!
```



```
Happy day, CS106A!
You're awesome.
We believe in you!
```

mydata.txt

line `'You're awesome.'`

```
def main():
    with open('mydata.txt', 'r') as file:
        for line in file:
            line = line.strip()
            print(line)
```

console

```
Happy day, CS106A!
You're awesome.
```

```
Happy day, CS106A!
You're awesome.
We believe in you!
```

mydata.txt

```
def main():
    with open('mydata.txt', 'r') as file:
        for line in file:
            line = line.strip()
            print(line)
```

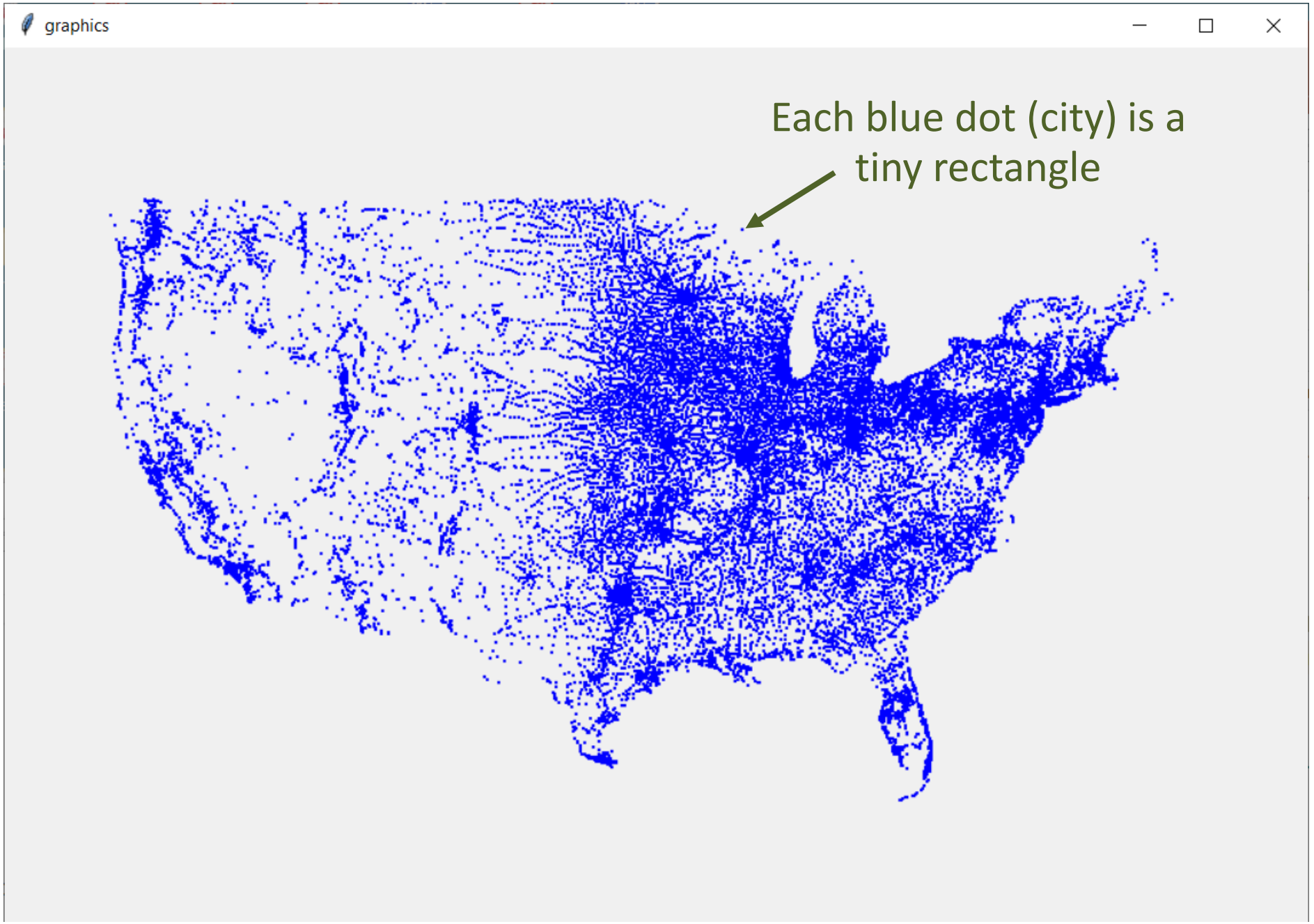
console

```
Happy day, CS106A!
You're awesome.
We believe in you!
```

Learning Goals

1. Know how to read a file line by line.

Files + Strings = Data Science!


graphics
Each blue dot (city) is a
tiny rectangle

us-cities.txt

```
City,State,Latitude,Longitude
Abbeville,AL,31.566367,-85.251300
Adamsville,AL,33.590411,-86.949166
Addison,AL,34.200042,-87.177851
Akron,AL,32.876425,-87.740978
```

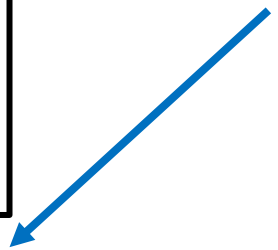
Strategy:

- Open file
- Skip header line
- Read/process each line
  - Strip ending `'\n'` from line
  - Split line into list (using `','` a delimiter)
  - list[2] is latitude
  - list[3] is longitude
  - plot city based on latitude and longitude

us-cities.txt

```
City,State,Latitude,Longitude
Abbeville,AL,31.566367,-85.251300
Adamsville,AL,33.590411,-86.949166
Addison,AL,34.200042,-87.177851
Akron,AL,32.876425,-87.740978
```

open uses reading mode ('r') by default



```
with open('us-cities.txt') as cities_file:
    next(cities_file)    # skip line in file
    for line in cities_file:
        line = line.strip()
        parts = line.split(',')
        lat = float(parts[2])
        lon = float(parts[3])
        plot_one_city(canvas, lat, lon)
```

us-cities.txt

```
City,State,Latitude,Longitude
Abbeville,AL,31.566367,-85.251300
Adamsville,AL,33.590411,-86.949166
Addison,AL,34.200042,-87.177851
Akron,AL,32.876425,-87.740978
```

```
with open('us-cities.txt') as cities_file:
    next(cities_file)    # skip line in file
    for line in cities_file:
        line = line.strip()
        parts = line.split(',')
        lat = float(parts[2])
        lon = float(parts[3])
        plot_one_city(canvas, lat, lon)
```

us-cities.txt

```
City,State,Latitude,Longitude
Abbeville,AL,31.566367,-85.251300
Adamsville,AL,33.590411,-86.949166
Addison,AL,34.200042,-87.177851
Akron,AL,32.876425,-87.740978
```

```
with open('us-cities.txt') as cities_file:
    next(cities_file)    # skip line in file
    for line in cities_file:
        line = line.strip()
        parts = line.split(',')
        lat = float(parts[2])
        lon = float(parts[3])
        plot_one_city(canvas, lat, lon)
```

us-cities.txt

```
City,State,Latitude,Longitude
Abbeville,AL,31.566367,-85.251300
Adamsville,AL,33.590411,-86.949166
Addison,AL,34.200042,-87.177851
Akron,AL,32.876425,-87.740978
```

```
with open('us-cities.txt') as cities_file:
    next(cities_file)    # skip line in file
    for line in cities_file:
        line = line.strip()
        parts = line.split(',')
        lat = float(parts[2])
        lon = float(parts[3])
        plot_one_city(canvas, lat, lon)
```

line `'Abbeville,AL,31.566367,-85.251300\n'`

us-cities.txt

```
City,State,Latitude,Longitude
Abbeville,AL,31.566367,-85.251300
Adamsville,AL,33.590411,-86.949166
Addison,AL,34.200042,-87.177851
Akron,AL,32.876425,-87.740978
```

```
with open('us-cities.txt') as cities_file:
    next(cities_file)    # skip line in file
    for line in cities_file:
        line = line.strip()
        parts = line.split(',')
        lat = float(parts[2])
        lon = float(parts[3])
        plot_one_city(canvas, lat, lon)
```

line `'Abbeville,AL,31.566367,-85.251300'`

us-cities.txt

```
City,State,Latitude,Longitude
Abbeville,AL,31.566367,-85.251300
Adamsville,AL,33.590411,-86.949166
Addison,AL,34.200042,-87.177851
Akron,AL,32.876425,-87.740978
```

```
with open('us-cities.txt') as cities_file:
    next(cities_file)    # skip line in file
    for line in cities_file:
        line = line.strip()
        parts = line.split(',')
        lat = float(parts[2])
        lon = float(parts[3])
        plot_one_city(canvas, lat, lon)
```

line `'Abbeville,AL,31.566367,-85.251300'`

parts `['Abbeville', 'AL', '31.566367', '-85.251300']`

us-cities.txt

```
City,State,Latitude,Longitude
Abbeville,AL,31.566367,-85.251300
Adamsville,AL,33.590411,-86.949166
Addison,AL,34.200042,-87.177851
Akron,AL,32.876425,-87.740978
```

```
with open('us-cities.txt') as cities_file:
    next(cities_file)    # skip line in file
    for line in cities_file:
        line = line.strip()
        parts = line.split(',')
        lat = float(parts[2])
        lon = float(parts[3])
        plot_one_city(canvas, lat, lon)
```

line `'Abbeville,AL,31.566367,-85.251300'`

parts `['Abbeville', 'AL', '31.566367', '-85.251300']`

lat `31.566367`

us-cities.txt

```
City,State,Latitude,Longitude
Abbeville,AL,31.566367,-85.251300
Adamsville,AL,33.590411,-86.949166
Addison,AL,34.200042,-87.177851
Akron,AL,32.876425,-87.740978
```

```
with open('us-cities.txt') as cities_file:
    next(cities_file)    # skip line in file
    for line in cities_file:
        line = line.strip()
        parts = line.split(',')
        lat = float(parts[2])
        lon = float(parts[3])
        plot_one_city(canvas, lat, lon)
```

line `'Abbeville,AL,31.566367,-85.251300'`

parts `['Abbeville', 'AL', '31.566367', '-85.251300']`

lat `31.566367`

lon `-85.251300`

us-cities.txt

```
City,State,Latitude,Longitude
Abbeville,AL,31.566367,-85.251300
Adamsville,AL,33.590411,-86.949166
Addison,AL,34.200042,-87.177851
Akron,AL,32.876425,-87.740978
```

```
with open('us-cities.txt') as cities_file:
    next(cities_file)    # skip line in file
    for line in cities_file:
        line = line.strip()
        parts = line.split(',')
        lat = float(parts[2])
        lon = float(parts[3])
        plot_one_city(canvas, lat, lon)
```

line	`'Abbeville,AL,31.566367,-85.251300'`
parts	`['Abbeville', 'AL', '31.566367', '-85.251300']`
lat	`31.566367`
lon	`-85.251300`

Let me see!
plotcities.py