# File Reading!

Let’s be data scientists!

Frankie Cerkvenik, CS106A, 2023

Stanford | ENGINEERING
Computer Science

# Housekeeping

- Fill out the midquarter feedback form!

- Breakout is due this Sunday :)

- Midterm review session **Monday 3:30-4:30 in STLC 114**!

# Today

- **Strings review**
  - **The key string functions**
  - **A code demo!**

- File reading
  - How to read text from a file
  - How to use that text to do cool things

# String functions (the important ones)

- Function: ***string1*** + ***string2***     # concatenation
  - Returns a new string that is the concatenation of ***string1*** and ***string2***

```
def copy_string(str):
    """
    Makes and returns a copy of str
    """
    copy = ''
    for letter in str:
        # copy become a new string equal to
        # the old letters in copy + letter
        copy = copy + letter
        # same as copy += letter
    return copy
```

# String functions (the important ones)

- Functions: ***string***`.isalpha()`/***string***`.isdigit()`
  - Returns true if all characters in ***string*** are letters in alphabet/digits

```
def separate_alphas(str):
  """
  Makes + returns a string with all the
  alphabetical characters in str
  """
  alphas = ''
  for letter in str:
    # check to see if the letter (not str)
    if letter.isalpha():
      alphas += letter
  return alphas
```

# General format of a string-building function

```
def separate_alphas(str):
  """
  Given str, make a new str with some
  transformation
  """
  result_str = '' # empty result to start
  for letter in str:
    # loop through each letter in input
    # possibly more code
    if some_condition:
      result += letter
  return result
```

# General format of a string-building function

```python
def separate_alphas(str):
  """
  Given str, make a new str with some
  transformation
  """
  result_str = '' # empty result to start
  for i in range(len(str)):
    # loop through each letter in input
    letter = str[i]
    # possibly more code
    if some_condition:
      result += letter
  return result
```

# String functions (the important ones)

- Function: ***string***`.find(string_to_find)`
  - Returns index of first occurrence of ***string_to_find*** in ***string***
  - Returns -1 if ***string_to_find*** is not found in the original ***string***

```
def find_barbie(str):
    """
    Returns true if the string barbie is in
    str
    """
    return str.find(‘barbie’) != -1
```

# String functions (the important ones)

- Function: ***string***`.find(string_to_find)`
  - Returns index of first occurrence of ***string_to_find*** in ***string***
  - Returns -1 if ***string_to_find*** is not found in the original ***string***
- Function: `string_to_find in string`
  - Returns **`True`** if **`string_to_find`** is in **string**
  - **`False`** otherwise

```
def find_barbie(str):
   """
   Returns true if the string barbie is in
   str
   """
   return ‘barbie’ in str
```

# String functions (the important ones)

- Function: ***string***`.lower()` / ***string***`.upper()`
  - Returns a new string with the same letters in **string,** but all lower/uppercase

```
def find_barbie(str):
  """
  Returns true if the string barbie is in
  str, case insensitive
  >>> find_barbie(‘BarBiE’)
  True
  """
  return ‘barbie’ in str.lower()
```

# String functions - Use your references!

- You are not expected to memorize every string function
- We can't even show you every string function!
- Goal: understand how to use strings enough that you can pick up a new function when you need it
- Don't be afraid to google something like "Python get index of a letter in a string" - increase your programming literacy! Cite your sources in assignment code
- **Don't** Google "how to do CS106A assn1"

# Strings are Immutable

```
str = 'abc'

str[1] = 'z'
```

Error!

Traceback (most recent call last):

...

TypeError: 'str' object does not support item assignment

```
str = str[0] + str[2:]
```

**Need to assign a new string**

# String Playground

In lecture code, we will do a few- you do the rest on your own!

Also check out string exercises on the experimental server for extra practice!

# Today

- ~~Strings review~~
    - ~~The key string functions~~
    - ~~A code demo!~~

- **File reading**
    - **How to read text from a file**
    - **How to use that text to do cool things**

# Getting Data into Programs

- **Put it directly in the program:**
  - Define constants holding your values.

- **Get it from the user:**
  - Mouse location, arguments to program, etc.

- **Generate it randomly:**
  - Use `random` library

- **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)

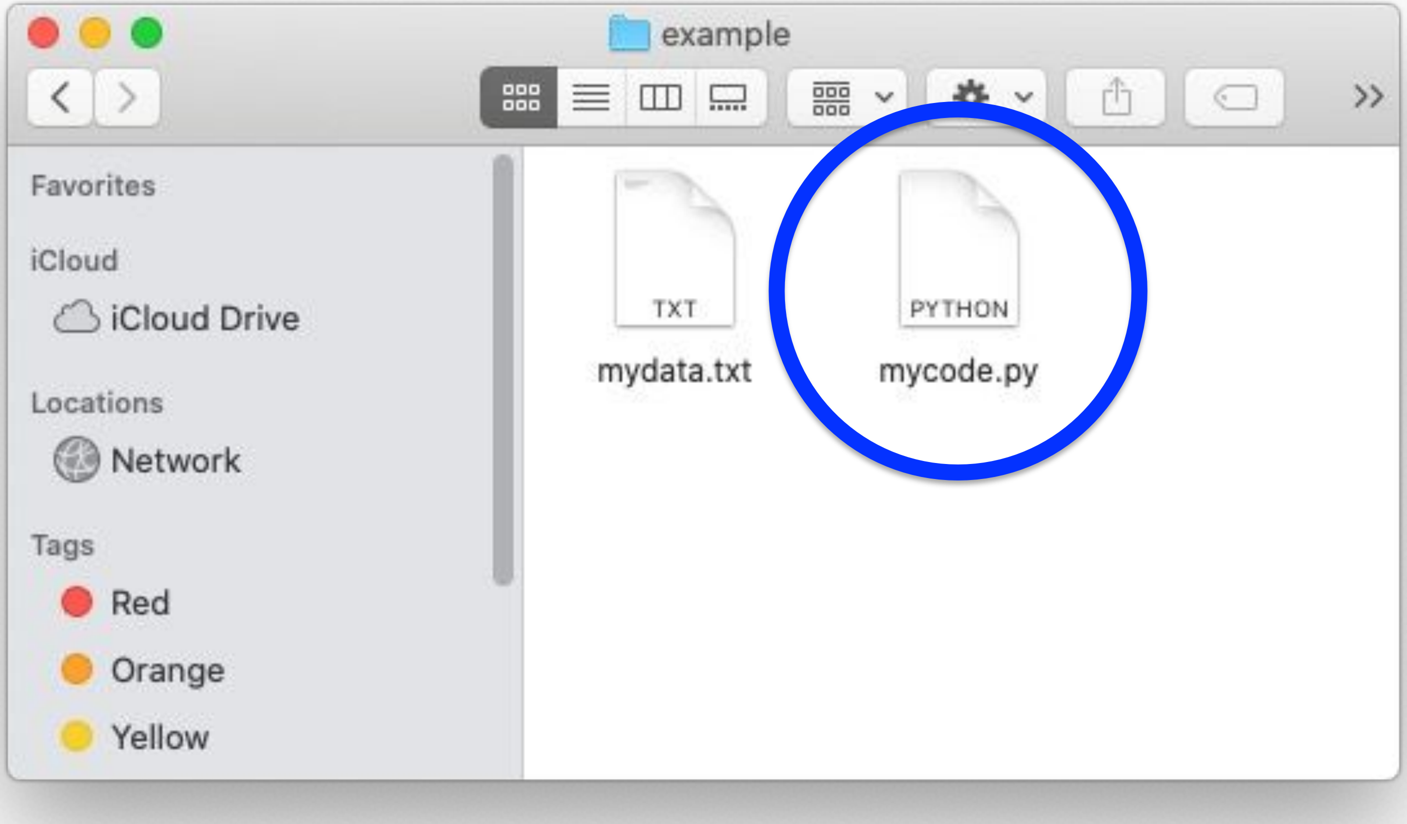
# The structure of files

- A file is just a series of ***bits*** (ones and zeros).
- Those bits can have structure:
  - Plain-text: Bits represent characters.
  - JPEG: Bits encode information about the structure of an image.
  - MP3: Bits encode frequency information about music.
  - etc.

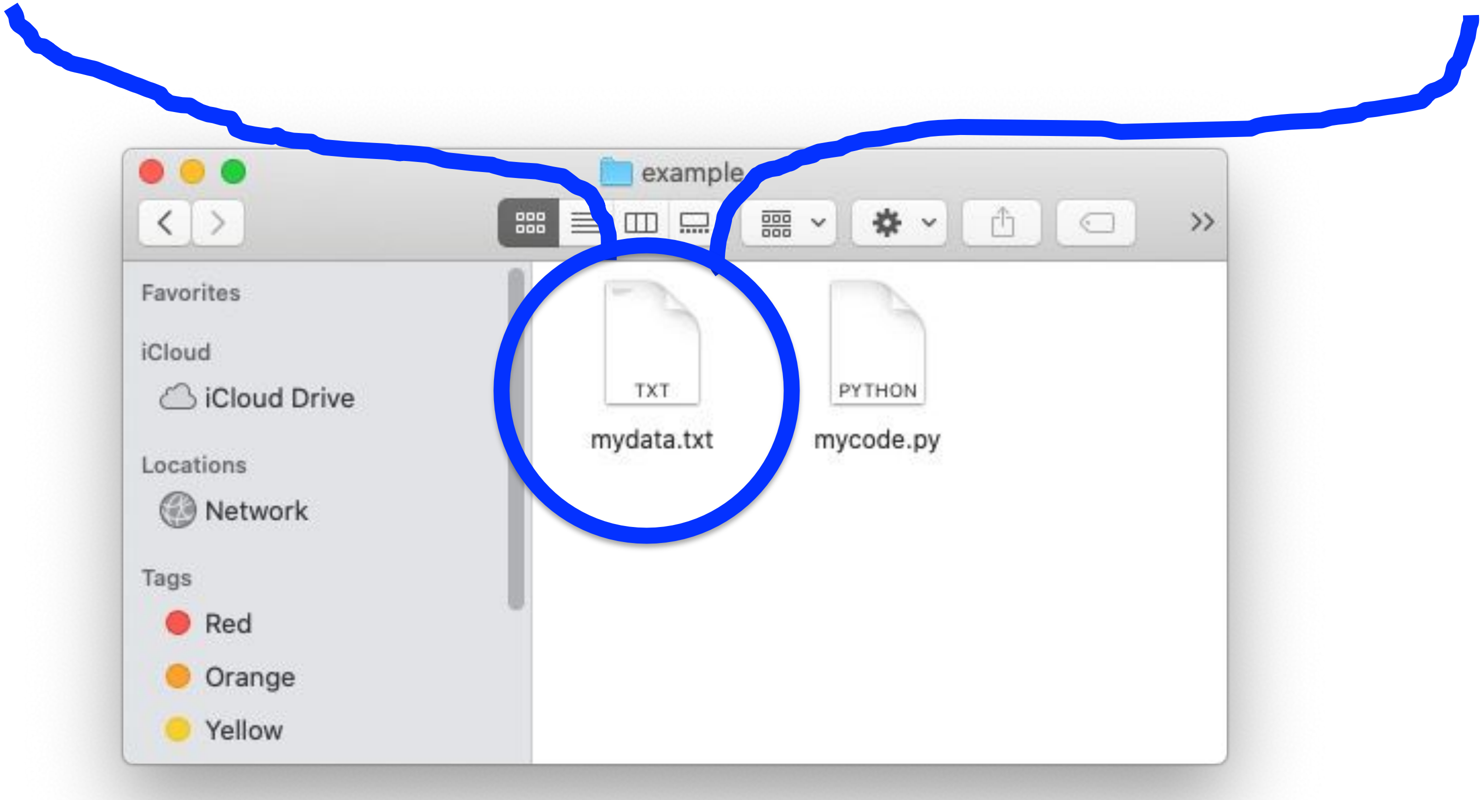
# The structure of files

- A file is just a series of ***bits*** (ones and zeros).
- Those bits can have structure:
  - **Plain-text: Bits represent characters.**
  - JPEG: Bits encode information about the structure of an image.
  - MP3: Bits encode frequency information about music.
  - etc.

# What’s a file?


example
Favorites
iCloud
iCloud Drive
Locations
Network
Tags
Red
Orange
Yellow
TXT
mydata.txt
PYTHON
mycode.py

```
Neither be cynical about love;
for in the face of all aridity
and disenchantment it is as
perennial as the grass.
- Max Ehrmann "Desiderata"
```



# Coding with files

```
Neither be cynical about love;
for in the face of all aridity
and disenchantment it is as
perennial as the grass.
- Max Ehrmann "Desiderata"
```

# Coding with files

```
Neither be cynical about love;
for in the face of all aridity
and disenchantment it is as
perennial as the grass.
- Max Ehrmann "Desiderata"
```

```
file = open('mydata.txt')
for line in file:
    print(line)
```

Step One:

Open the file

```
Neither be cynical about love;
for in the face of all aridity
and disenchantment it is as
perennial as the grass.
- Max Ehrmann "Desiderata"
```

**Terminal**

```
file = open('mydata.txt')
for line in file:
    print(line)
```

## Step Two:

## Read the file one line at a time

```
Neither be cynical about love;
for in the face of all aridity
and disenchantment it is as
perennial as the grass.
- Max Ehrmann "Desiderata"
```

**Terminal**

```
Neither be cynical about love;
```

```python
file = open('mydata.txt')
for line in file:
    print(line)
```

## Step Two:

## Read the file one line at a time

```
Neither be cynical about love;
for in the face of all aridity
and disenchantment it is as
perennial as the grass.
- Max Ehrmann "Desiderata"
```

```python
file = open('mydata.txt')
for line in file:
    print(line)
```

**Terminal**

```
Neither be cynical about love;
```

Step Two:

Read the file one line at a time

```
Neither be cynical about love;
for in the face of all aridity
and disenchantment it is as
perennial as the grass.
- Max Ehrmann "Desiderata"
```

```python
file = open('mydata.txt')
for line in file:
    print(line)
```

**Terminal**

```
Neither be cynical about love;

for in the face of all aridity
```

## Step Two:

## Read the file one line at a time

```
Neither be cynical about love;
for in the face of all aridity
and disenchantment it is as
perennial as the grass.
- Max Ehrmann "Desiderata"
```

```python
file = open('mydata.txt')
for line in file:
    print(line)
```

Terminal

```
Neither be cynical about love;

for in the face of all aridity
```

## Step Two:

## Read the file one line at a time

```
Neither be cynical about love;
for in the face of all aridity
and disenchantment it is as
perennial as the grass.
- Max Ehrmann "Desiderata"
```

```python
file = open('mydata.txt')
for line in file:
    print(line)
```

**Terminal**

```
Neither be cynical about love;

for in the face of all aridity

and disenchantment it is as
```

## Step Two:

## Read the file one line at a time

```
Neither be cynical about love;
for in the face of all aridity
and disenchantment it is as
perennial as the grass.
- Max Ehrmann "Desiderata"
```

```python
file = open('mydata.txt')
for line in file:
    print(line)
```

**Terminal**

```
Neither be cynical about love;

for in the face of all aridity

and disenchantment it is as
```

## Step Two:

## Read the file one line at a time

```
Neither be cynical about love;
for in the face of all aridity
and disenchantment it is as
perennial as the grass.
- Max Ehrmann "Desiderata"
```

```python
file = open('mydata.txt')
for line in file:
    print(line)
```

**Terminal**

```
Neither be cynical about love;

for in the face of all aridity

and disenchantment it is as

perennial as the grass.
```

## Step Two:

## Read the file one line at a time

```
Neither be cynical about love;
for in the face of all aridity
and disenchantment it is as
perennial as the grass.
- Max Ehrmann "Desiderata"
```

```python
file = open('mydata.txt')
for line in file:
    print(line)
```

**Terminal**

```
Neither be cynical about love;

for in the face of all aridity

and disenchantment it is as

perennial as the grass.
```

## Step Two:

## Read the file one line at a time

```
Neither be cynical about love;
for in the face of all aridity
and disenchantment it is as
perennial as the grass.
- Max Ehrmann "Desiderata"
```

```python
file = open('mydata.txt')
for line in file:
    print(line)
```

**Terminal**

```
Neither be cynical about love;

for in the face of all aridity

and disenchantment it is as

perennial as the grass.

- Max Ehrmann “Desiderata”
```

## Step Two:

## Read the file one line at a time

```
Neither be cynical about love;
for in the face of all aridity
and disenchantment it is as
perennial as the grass.
- Max Ehrmann "Desiderata"
```

```python
file = open('mydata.txt')
for line in file:
    print(line)
```

**Terminal**

```
Neither be cynical about love;

for in the face of all aridity

and disenchantment it is as

perennial as the grass.

- Max Ehrmann “Desiderata”
```

## Step Two:

## Read the file one line at a time

# Why all the empty lines?

line

```
'perennial as the grass.\n'
```

```python
file = open('mydata.txt')
for line in file:
    print(line)
```

Terminal

```
Neither be cynical about love;

for in the face of all aridity

and disenchantment it is as

perennial as the grass.

- Max Ehrmann “Desiderata”
```

# Take two: use strip()

line 'perennial as the grass.'

```
file = open('mydata.txt')
for line in file:
   line = line.strip()
   print(line)
```

# Take two: use strip()

```
line    'perennial as the grass.'
```

```
file = open('mydata.txt')
for line in file:
    line = line.strip()
    print(line)
```

**Terminal**

```
Neither be cynical about love;
for in the face of all aridity
and disenchantment it is as
perennial as the grass.
- Max Ehrmann “Desiderata”
```

# What does this do?

```
f = open('mydata.txt')
for line in f:
  print(line.strip())
print('-----')
for line in f:
   print(line.strip())
```

Option A

```
Neither be cynical about love;
for in the face of all aridity
and disenchantment it is as
perennial as the grass
- Max Ehrmann "Desiderata"
----
```

Option B

```
Neither be cynical about love;
for in the face of all aridity
and disenchantment it is as
perennial as the grass
- Max Ehrmann "Desiderata"
-----
Neither be cynical about love;
for in the face of all aridity
and disenchantment it is as
perennial as the grass
- Max Ehrmann "Desiderata"
```

# What does this do?

```
f = open('mydata.txt')
for line in f:
  print(line.strip())
print('-----')
for line in f:
   print(line.strip())
```

Option A

```
Neither be cynical about love;
for in the face of all aridity
and disenchantment it is as
perennial as the grass
- Max Ehrmann "Desiderata"
----
```

Option B

```
Neithe      nical a      ove;
for i        e of        ty
and
perenn
- Max Eh                 "
-----
Neither b              love;
for in                  dity
and d
pere          e g
- Max          "Desi
```

# What does this do?

```python
f = open('mydata.txt')
for line in f:
  print(line.strip())
print('-----')
for line in f:
   print(line.strip())
```

Option A

```
Neither be cynical about love;
for in the face of all aridity
and disenchantment it is as
perennial as the grass
- Max Ehrmann "Desiderata"
----
```

**Key idea: files (the f object) don’t reset after reading!**

# Round 2: What does this do?

```
f = open('mydata.txt')
next(f)
for line in f:
  print(line.strip())
```

# Round 2: What does this do?

```
f = open('mydata.txt')
next(f)
for line in f:
  print(line.strip())
```

```
for in the face of all
aridity and disenchantment it
is as perennial as the grass
- Max Ehrmann "Desiderata"
```

**`next(f)` skips (and returns) a line in f!**

# Aside: `with`

*This is the proper way to do it in the old days…*

```
with open('mydata.txt') as f:
    for line in file:
        line = line.strip()
        print(line)
```

**`with`** *recruits the "file manager" to help python know when it can "close" the file*

# Aside: `with`

*But we can do it like this now…*

```
for line in open('mydata.txt'):
    line = line.strip()
    print(line)
```

You actually don't need **`with`**.
The garbage collector will close your file for you…

# Aside: `with`

```
with open('mydata.txt') as f:
    for line in file:
        line = line.strip()
        print(line)
```

But some old versions of Python don't garbage collect and then the file is open till the program ends wasting resources

# Let’s be edgy, no need for `with!`

```python
for line in open('mydata.txt'):
    line = line.strip()
    print(line)
```

The 90s called.

They want python2 back…

# Let’s be edgy, no need for `with!`

```
for line in open('mydata.txt'):
    line = line.strip()
    print(line)
```

The 90s called.

They want python2 back…

Just don’t worry if you see code with with!

# Should you use `with`?

People bother you less if you use the first one

```
with open('mydata.txt') as f:
    for line in file:
        line = line.strip()
        print(line)
```

But this is the future!

```
for line in open('mydata.txt'):
    line = line.strip()
    print(line)
```

# Files + strings + lists = data science

(more on this on Tuesday)

## dataset.csv

```
Kenya,100,50
Malaysia,50,100

...

Turkey,20,20
Spain,95,95
```

## dataset.csv

```
Kenya,100,50
Malaysia,50,100

...

Turkey,20,20
Spain,95,95
```

```
file = open(dataset.csv)
for line in file:
    line = line.strip()
    values = line.split(',')
    print(values[1])
```

# dataset.csv

```
Kenya,100,50
Malaysia,50,100

...

Turkey,20,20
Spain,95,95
```

```python
file = open(dataset.csv)
for line in file:
    line = line.strip()
    values = line.split(',')
    print(values[1])
```

line:

`'Kenya,100,50\n'`

## dataset.csv

```
Kenya,100,50
Malaysia,50,100

...

Turkey,20,20
Spain,95,95
```

```
file = open(dataset.csv)
for line in file:
    line = line.strip()
    values = line.split(',')
    print(values[1])
```

line:

**'Kenya,100,50'**

# dataset.csv

```
Kenya,100,50
Malaysia,50,100

...

Turkey,20,20
Spain,95,95
```

```python
file = open(dataset.csv)
for line in file:
    line = line.strip()
    values = line.split(',')
    print(values[1])
```

line:

'Kenya,100,50'

values:

| | | |
|---|---|---|
| 'Kenya' | '100' | '50' |
| 0 | 1 | 2 |

## dataset.csv

```
Kenya,100,50
Malaysia,50,100

...

Turkey,20,20
Spain,95,95
```

```
file = open(dataset.csv)
for line in file:
    line = line.strip()
    values = line.split(',')
    print(values[1])
```

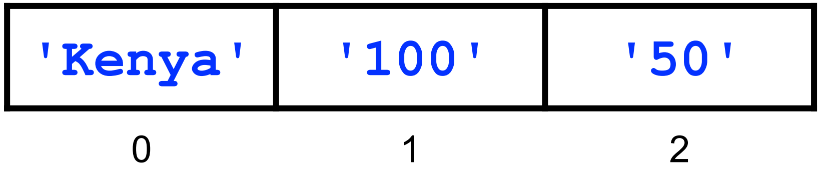
**Terminal**:

```
100
```

line:

'Kenya,100,50'

values:



| 'Kenya' | '100' | '50' |
|---|---|---|
| 0 | 1 | 2 |

# read_csv.py

- Let's write a program that can take in a filename of a .csv file OR a filename and a column number
- Print out every row in the csv file as a list (implement this first)
- If the user provided a column number, print each row's value in that column
- Example: Run

```
python3 read_csv_soln.py staff_info.csv 1
```

# (if time)Let’s do some data processing!

Graphs graphs graphs graphs

# (if time) Remember Assn0?

- Write a program that allows the user to specify the filename of a CSV, a column number in that CSV, a min_frequency and a max_frequency, and any number of string values
- Display a bar chart representing the frequency with which each string value appears in the specified column in the dataset
- (Demo in the started code)
- Use the pre-made make_bar_chart function
- Decompose logic to process the file
- Use it on our anonymized Assn0 answers!

# Key idea!

Test your data-processing functions on small files, where you can manually tally expected output!

# Recap

- Review string functions in Python! There are many, check out string slides and don't be afraid to Google!

- Key idea of building strings: loops + concatenate

- We can read in text files one line at a time just the **`open`** function

- We often need to process each line in some way with **`split()`** and/or **`strip()`**... this is the key to data processing!