
Files
Chris Piech and Mehran Sahami

Who we are today

Galileo Galilei

Review

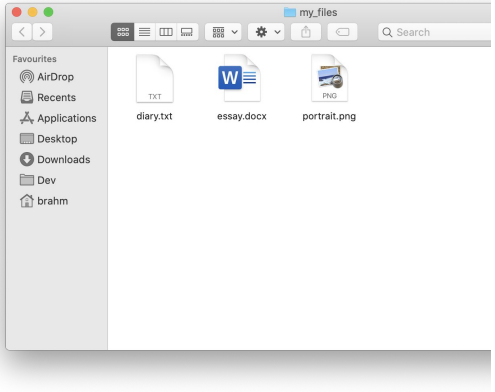
Dictionaries

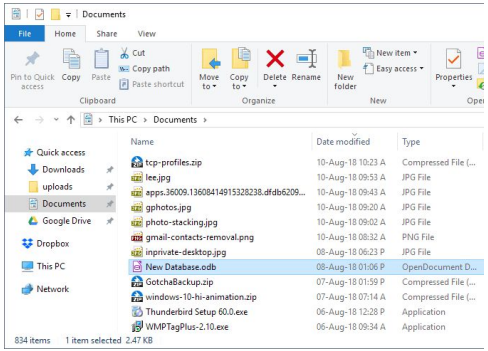
Dictionaries allow us to build one-way associations between one kind of data (which we call the key) and another (which we call the value). A common metaphor for them is a phone book, whose keys are people's names and whose values are their phone numbers. It's super easy to look someone's number up if you know their name, but harder to do it the other way around.

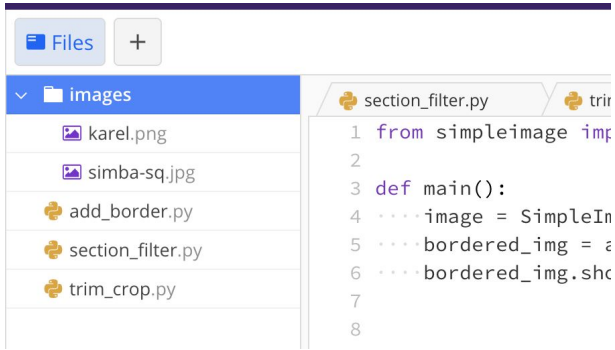
```
>>> d = {}                  # make an empty dictionary
>>> d['brahm'] = 42         # associate the key 'brahm' with the value 42
>>> d['julie'] = 5          # associate the key 'julie' with the value 5
>>> d['julie'] = 8          # change the value for 'julie' to be 8
                            # since keys need to be unique

>>> d['brahm']              # get the value associated with 'brahm'
42
>>> 'python' in d           # check whether a particular key is in the map
False
```

What’s a file?

Files on your computer







Files on your computer

When you edit a word document…

…the program *Microsoft Word* is opening the file `essay.docx`

Files on your computer

When you edit a word document…

…the program *Microsoft Word* is opening the file `essay.docx`

When you watch a video on your computer

…the program *VLC* is opening the file `Avengers: Endgame.mp4`

Files on your computer

When you edit a word document…

…the program *Microsoft Word* is opening the file `essay.docx`

When you watch a video on your computer

…the program *VLC* is opening the file `Avengers: Endgame.mp4`

When you listen to a song…

…the program *iTunes* is opening the file `Thriller.mp3`

Files in Python

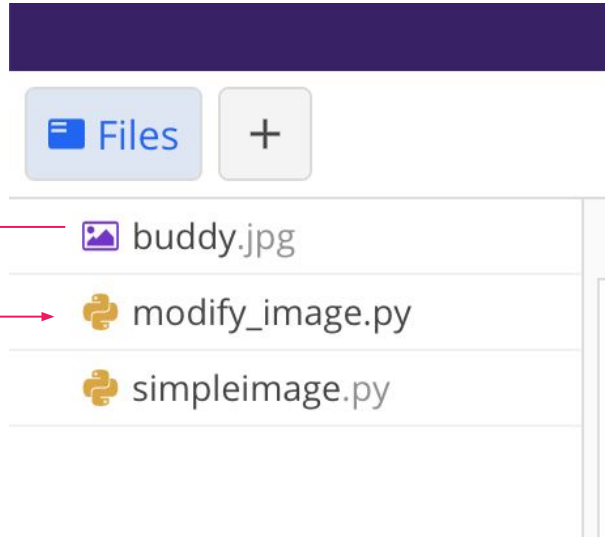
```
img = SimpleImage("buddy.jpg")
```

buddy.jpg

Files in Python

```
img = SimpleImage("buddy.jpg")
```

The program `modify_image.py` is opening the file `buddy.jpg`



Files in Python

```
python3 program.py
```

program.py

```
def main():
    # Python code here

if __name__ == "__main__":
    main()
```

Files in Python

```
python3 program.py
```

The program `python3` is opening the file `program.py`

program.py

```
def main():
    # Python code here

if __name__ == "__main__":
    main()
```

How files are stored

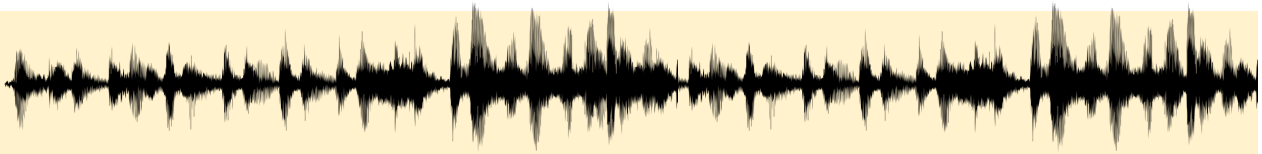
foundation.txt

```
His name was Gaal Dornick and he was just a
country boy who had never seen Trantor before...
```

program.py

```
def main():
    print("Hello world!")
```

thechain.mp3



spiderverse.mp4

How files are stored

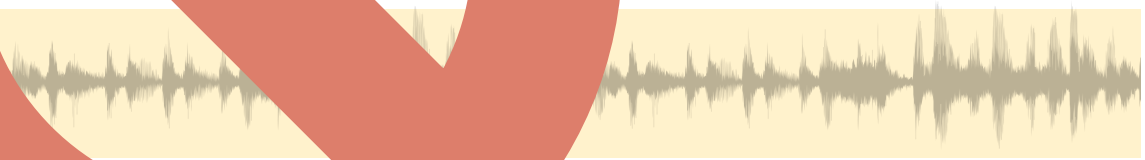
foundation.txt

```
His r                    ick and he was just a
co                       seen Trantor before...
```

program.py

```
pri        o world!
```

thechain.mp3



spiderverse.mp4

How are files stored?

foundation.txt

```
01001000 01101001 01110011 00100000 01101110
01100001 01101101 01100101 00100000 01110111...
```

program.py

```
01100100 01100101 01100110 00100000 01101101
01100001 01101001 01101110 00101000 00101001...
```

thechain.mp3

```
01101001 01010110 01000010 01001111 01010010
01110111 00110000 01001011 01000111 01100111...
```

spiderverse.mp4

```
00101111 00111001 01101010 00101111 00110100
01010010 01001100 01101011 01010010 01011000...
```

How files are stored

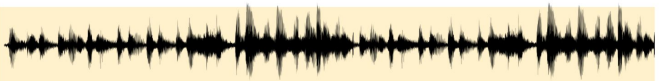
foundation.txt

```
His name was Gaal Dornick and he was just a
country boy who had never seen Trantor before...
```

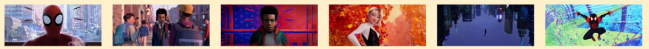
program.py

```
def main():
    print(“Hello world!”)
```

thechain.mp3



spiderverse.mp4



How files are stored

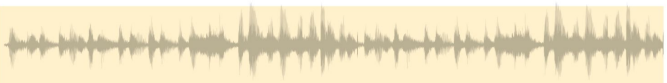
foundation.txt

```
His name was Gaal Dornick and he was just a
country boy who had never seen Trantor before...
```

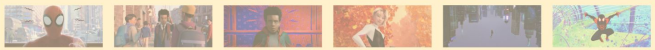
program.py

```
def main():
    print(“Hello world!”)
```

thechain.mp3



spiderverse.mp4



Text Files

foundation.txt

```
His name was Gaal Dornick and he was just a
country boy who had never seen Trantor before...
```

Text Files

foundation.txt

```
His name was Gaal Dornick and he was just a
country boy who had never seen Trantor before...
```

Text files store data that you can read

Text Files

foundation.txt

```
His name was Gaal Dornick and he was just a
country boy who had never seen Trantor before...
```

countries.csv

```
Argentina,Buenos Aires,45M,South America
Singapore,Singapore,5.7M,Asia
New Zealand,Wellington,4.9M,Oceania
```

Text files store data that you can read

Text Files

foundation.txt

```
His name was Gaal Dornick and he was just a
country boy who had never seen Trantor before...
```

countries.csv

```
Argentina,Buenos Aires,45M,South America
Singapore,Singapore,5.7M,Asia
New Zealand,Wellington,4.9M,Oceania
```

locations.json

```
{"Chris": "Stanford", "Brahm": "New York"}
```

Text files store data that you can read

Text Files

foundation.txt

```
His name was Gaal Dornick and he was just a
country boy who had never seen Trantor before...
```

countries.csv

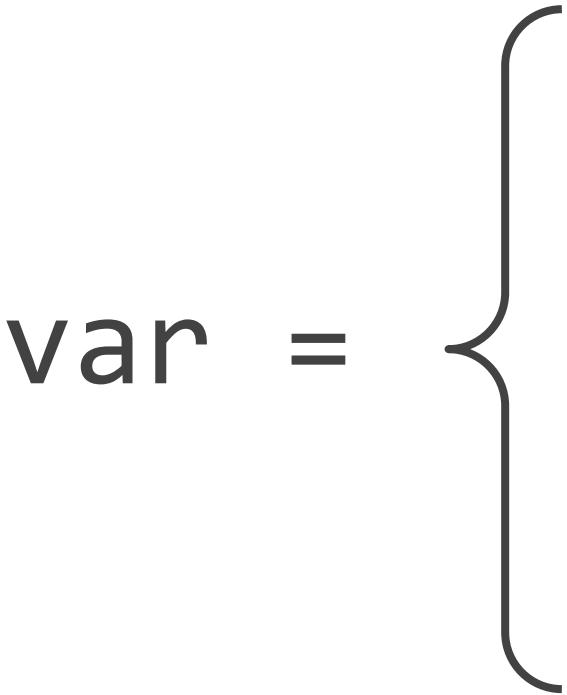
```
Argentina,Buenos Aires,45M,South America
Singapore,Singapore,5.7M,Asia
New Zealand,Wellington,4.9M,Oceania
```

locations.json

```
{"Chris": "Stanford", "Brahm": "New York"}
```

Text files store data that a Python program can read

Where do values come from in Python?



Where do values come from in Python?

`var =` { A literal like 42

Where do values come from in Python?

`var =` {

- A literal like 42
- User `input`

Where do values come from in Python?

`var =`

- A literal like `42`
- User `input`
- A `random` value

Where do values come from in Python?

`var =`

- A literal like 42
- User `input`
- A `random` value

All of these only exist while the program is running

Where do values come from in Python?

`var =`

- A literal like `42`
- User `input`
- A `random` value
- Data from a file!

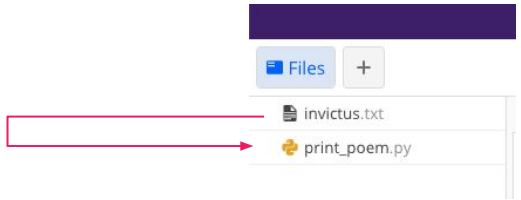
Reading a file

Reading a file in Python

`invictus.txt`

```
Out of the night that covers me,
      Black as the pit from pole to pole,
I thank whatever gods may be
      For my unconquerable soul.
.
.
.
I am the master of my fate,
      I am the captain of my soul.

 'Invictus', by William Ernest Henley
```

The program `print_poem.py` is opening the file `invictus.txt`



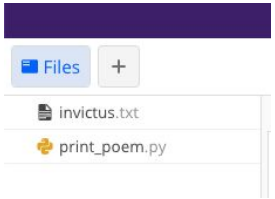
Reading a file in Python

print_poem.py

```
f = open('invictus.txt')
for line in f:
    # process the line
f.close()
```

invictus.txt

```
Out of the night that covers me,
      Black as the pit from pole to pole,
I thank whatever gods may be
      For my unconquerable soul.
.
.
.
I am the master of my fate,
      I am the captain of my soul.

 'Invictus', by William Ernest Henley
```



Reading a file in Python

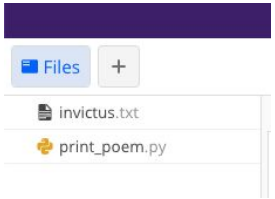
print_poem.py

```
f = open('invictus.txt')
for line in f:
    # process the line
f.close()
```

f

invictus.txt

```
Out of the night that covers me,
      Black as the pit from pole to pole,
I thank whatever gods may be
      For my unconquerable soul.
.
.
.
I am the master of my fate,
      I am the captain of my soul.

 'Invictus', by William Ernest Henley
```



Reading a file in Python

print_poem.py

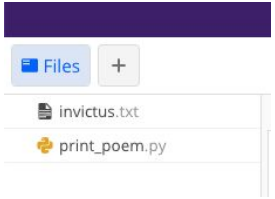
```
f = open('invictus.txt')
for line in f:
    # process the line
f.close()
```

invictus.txt

```
Out of the night that covers me,
      Black as the pit from pole to pole,
I thank whatever gods may be
      For my unconquerable soul.
.
.
.
I am the master of my fate,
      I am the captain of my soul.

 'Invictus', by William Ernest Henley
```

f

line → "Out of the night that covers me,"



Reading a file in Python

print_poem.py

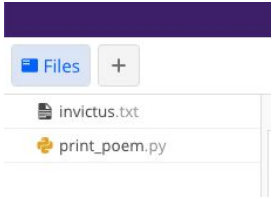
```
f = open('invictus.txt')
for line in f:
    # process the line
f.close()
```

invictus.txt

```
Out of the night that covers me,
      Black as the pit from pole to pole,
I thank whatever gods may be
      For my unconquerable soul.
.
.
.
I am the master of my fate,
      I am the captain of my soul.

 'Invictus', by William Ernest Henley
```

f

line → "Out of the night that covers me,"



Reading a file in Python

print_poem.py

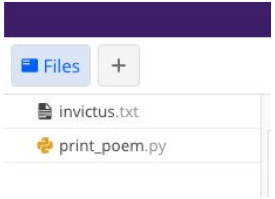
```
f = open('invictus.txt')
for line in f:
    # process the line
f.close()
```

invictus.txt

```
Out of the night that covers me,
      Black as the pit from pole to pole,
I thank whatever gods may be
      For my unconquerable soul.
.
.
.
I am the master of my fate,
      I am the captain of my soul.

 'Invictus', by William Ernest Henley
```

f

line → " Black as the pit from pole to pole,"



Reading a file in Python

print_poem.py

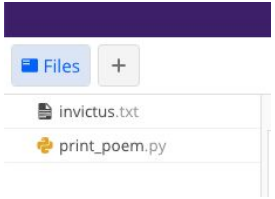
```
f = open('invictus.txt')
for line in f:
    # process the line
f.close()
```

invictus.txt

```
Out of the night that covers me,
      Black as the pit from pole to pole,
I thank whatever gods may be
      For my unconquerable soul.
.
.
.
I am the master of my fate,
      I am the captain of my soul.

 'Invictus', by William Ernest Henley
```

f

line → " Black as the pit from pole to pole,"



3.5 stanzas later...

Reading a file in Python

print_poem.py

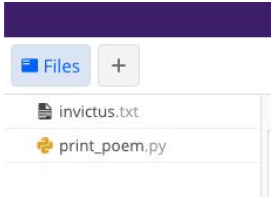
```
f = open('invictus.txt')
for line in f:
    # process the line
f.close()
```

invictus.txt

```
Out of the night that covers me,
      Black as the pit from pole to pole,
I thank whatever gods may be
      For my unconquerable soul.
.
.
.
I am the master of my fate,
      I am the captain of my soul.

'Invictus', by William Ernest Henley
```

f

line → ""



Reading a file in Python

print_poem.py

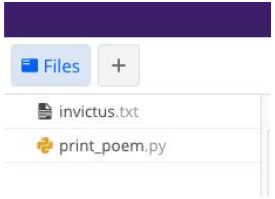
```
f = open('invictus.txt')
for line in f:
    # process the line
f.close()
```

invictus.txt

```
Out of the night that covers me,
      Black as the pit from pole to pole,
I thank whatever gods may be
      For my unconquerable soul.
.
.
.
I am the master of my fate,
      I am the captain of my soul.

'Invictus', by William Ernest Henley
```

f

line → "'Invictus', by William Ernest Henley"



Reading a file in Python

print_poem.py

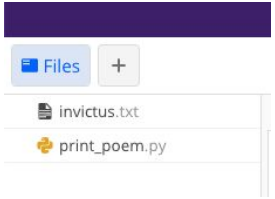
```
f = open('invictus.txt')
for line in f:
    # process the line
f.close()
```

invictus.txt

```
Out of the night that covers me,
      Black as the pit from pole to pole,
I thank whatever gods may be
      For my unconquerable soul.
.
.
.
I am the master of my fate,
      I am the captain of my soul.

 'Invictus', by William Ernest Henley
```

f

line → "'Invictus', by William Ernest Henley"



Reading a file in Python

print_poem.py

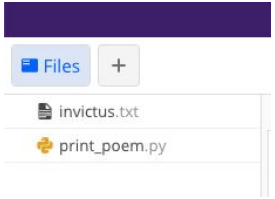
```
f = open('invictus.txt')
for line in f:
    # process the line
f.close()
```

invictus.txt

```
Out of the night that covers me,
      Black as the pit from pole to pole,
I thank whatever gods may be
      For my unconquerable soul.
.
.
.
I am the master of my fate,
      I am the captain of my soul.

 'Invictus', by William Ernest Henley
```

f ⟶ ❌

line ⟶ "'Invictus', by William Ernest Henley"



Reading a file in Python

print_poem.py

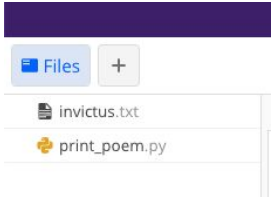
```
f = open('invictus.txt')
for line in f:
    # process the line
f.close()
```

invictus.txt

```
Out of the night that covers me,
      Black as the pit from pole to pole,
I thank whatever gods may be
      For my unconquerable soul.
.
.
.
I am the master of my fate,
      I am the captain of my soul.

 'Invictus', by William Ernest Henley
```

f ⟶ ¯_(ツ)_/¯

line ⟶ "'Invictus', by William Ernest Henley"



Reading a file in Python

print_poem.py

```
f = open('invictus.txt')
for line in f:
    # process the line
f.close()
```

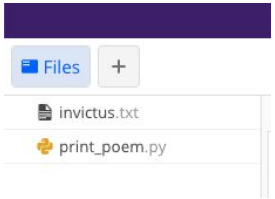
invictus.txt

```
Out of the night that covers me,
      Black as the pit from pole to pole,
I thank whatever gods may be
      For my unconquerable soul.
.
.
.
I am the master of my fate,
      I am the captain of my soul.

 'Invictus', by William Ernest Henley
```

f ⟶ ¯_(ツ)_/¯

line ⟶ "'Invictus', by William Ernest Henley"

You can't reread the file without reopening it



Reading a file in Python: Newlines

Computers don’t know that ‘lines’ are shown below one another in a file.

```
Out of the night that covers me,

  Black as the Pit from pole to pole,

I thank whatever gods may be

  For my unconquerable soul.

In the fell clutch of circumstance

  I have not winced nor cried aloud.

Under the bludgeonings of chance

  My head is bloody, but unbowed.

Beyond this place of wrath and tears

  Looms but the Horror of the shade,

And yet the menace of the years

  Finds, and shall find, me unafraid.

It matters not how strait the gate,

  How charged with punishments the scroll,

I am the master of my fate:

  I am the captain of my soul.


'Invictus', by William Ernest Henley
```

Reading a file in Python: Newlines

They store all the lines in a file in one continuous block...

```
Out of the night that covers me,  Black as the Pit from pole to pole,I thank whatever gods may be  For my unconquerable soul.In the fell clutch of circumstance  I have not winced nor cried aloud.Under the bludgeonings of chance   My head is bloody, but unbowed.Beyond this place of wrath and tears  Looms but the Horror of the shade,And yet the menace of the years  Finds, and shall find, me unafraid.It matters not how strait the gate,  How charged with punishments the scroll,I am the master of my fate:  I am the captain of my soul.'Invictus', by William Ernest Henley
```

Reading a file in Python: Newlines

...and separate the lines with a special newline character, \n.

```
Out of the night that covers me,\n  Black as the Pit from pole to pole,\nI thank whatever gods may be\n  For my unconquerable soul.\nIn the fell clutch of circumstance\n  I have not winced nor cried aloud.\nUnder the bludgeonings of chance\n My head is bloody, but unbowed.\nBeyond this place of wrath and tears\n  Looms but the Horror of the shade,\nAnd yet the menace of the years\n  Finds, and shall find, me unafraid.\nIt matters not how strait the gate,\n  How charged with punishments the scroll,\nI am the master of my fate:\n  I am the captain of my soul.\n\n'Invictus', by William Ernest Henley
```

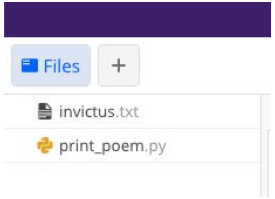
Reading a file in Python

print_poem.py

```
f = open('invictus.txt')
for line in f:
    print(line)
f.close()
```

invictus.txt

```
Out of the night that covers me\n,
      Black as the pit from pole to pole\n,
I thank whatever gods may be\n
      For my unconquerable soul.\n
.
.
.
I am the master of my fate,\n
      I am the captain of my soul.\n
 \n
 'Invictus', by William Ernest Henley
```



Reading a file in Python

print_poem.py

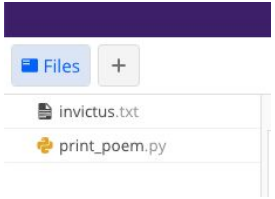
```
f = open('invictus.txt')
for line in f:
    print(line)
f.close()
```

invictus.txt

```
Out of the night that covers me\n,
      Black as the pit from pole to pole\n,
I thank whatever gods may be\n
      For my unconquerable soul.\n
.
.
.
I am the master of my fate,\n
      I am the captain of my soul.\n
 \n
 'Invictus', by William Ernest Henley
```

f

line → "Out of the night that covers me,\n"



Reading a file in Python

print_poem.py

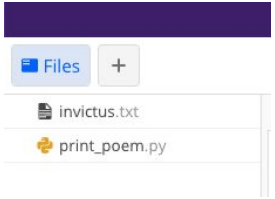
```
f = open('invictus.txt')
for line in f:
    print(line)
f.close()
```

invictus.txt

```
Out of the night that covers me\n,
      Black as the pit from pole to pole\n,
I thank whatever gods may be\n
      For my unconquerable soul.\n
.
.
.
I am the master of my fate,\n
      I am the captain of my soul.\n
 \n
 'Invictus', by William Ernest Henley
```

f

line → "Out of the night that covers me,\n"



Reading a file in Python

```
        print(line)
```

line ⟶ "Out of the night that covers me,\n"

Reading a file in Python

```
print("Out of the night that covers me,\n")
```

Reading a file in Python

```
print("Out of the night that covers me,\n")
print("  Black as the Pit from pole to pole,\n")
```

Reading a file in Python

```
print("Out of the night that covers me,\n")
print("  Black as the Pit from pole to pole,\n")
```

```
Out of the night that covers me,\n
\n
```

This \n comes from the string we're printing

This \n comes from the print function

Reading a file in Python

```
print("Out of the night that covers me,\n")
print("  Black as the Pit from pole to pole,\n")
```

```
Out of the night that covers me,\n
\n
  Black as the Pit from pole to pole,\n
\n
```

A file reading loop

```
f = open(filename)
for line in f:
    line = line.strip()  # or strip("\n"), lstrip(), rstrip()
    # process line
f.close()
```

The file reading loop

```
with open(filename) as f:
    for line in f:
        line = line.strip()  # or strip("\n"), lstrip(), rstrip()
        # process line
```

The file reading loop

This line creates `f` for us, and automatically `close`s it when we finish reading the file

```
with open(filename) as f:
    for line in f:
        line = line.strip()  # or strip("\n"), lstrip(), rstrip()
        # process line
```

Skipping lines in a file

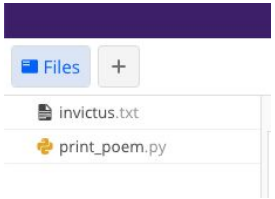
print_poem.py

```
f = open('invictus.txt')
next(f)
for line in f:
    # process the line
f.close()
```

f

invictus.txt

```
Out of the night that covers me,
      Black as the pit from pole to pole,
I thank whatever gods may be
      For my unconquerable soul.
.
.
.
I am the master of my fate,
      I am the captain of my soul.

 'Invictus', by William Ernest Henley
```



Skipping lines in a file

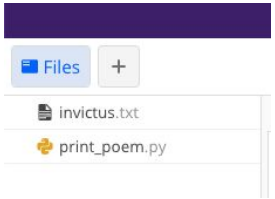
print_poem.py

```
f = open('invictus.txt')
next(f)
for line in f:
     # process the line
f.close()
```

f

invictus.txt

```
Out of the night that covers me,
      Black as the pit from pole to pole,
I thank whatever gods may be
      For my unconquerable soul.
.
.
.
I am the master of my fate,
      I am the captain of my soul.

 'Invictus', by William Ernest Henley
```



Skipping lines in a file

print_poem.py

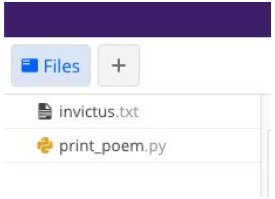
```
f = open('invictus.txt')
next(f)
for line in f:
    # process the line
f.close()
```

invictus.txt

```
Out of the night that covers me,
      Black as the pit from pole to pole,
I thank whatever gods may be
      For my unconquerable soul.
.
.
.
I am the master of my fate,
      I am the captain of my soul.

 'Invictus', by William Ernest Henley
```

f

line → " Black as the pit from pole to pole,"



Another way to read files

```
with open(filename) as f:
    file_contents = f.read()
```

This reads the entire contents of the files into a string, including the \n newline characters

Another way to read files

```
with open(filename) as f:
    lines = f.readlines()
```

This reads each line of a file (ending with a `\n` character) into a list

A gentle warning

`f.read()` and `f.readlines()` are handy, but shouldn't be your first resort. If you try to read really big files, that takes up a lot of your computer's memory, which you should try to avoid. You generally won't need the contents of the entire file as a string or list, anyway.

Writing a file

Writing to files

```
with open(filename) as f:
    for line in f:
        line = line.strip()  # or strip("\n"), lstrip(), rstrip()
        # process line
```

This opens the file in read mode, meaning that we can read it, but can't modify it.

Writing to files

```
with open(filename, 'w') as f:
```

This opens the file in write mode, meaning that we can write to it, but can't read it.

Writing to files

```
with open(filename, 'w') as f:
```

If there isn't a file with the name stored in `filename` already, this creates it

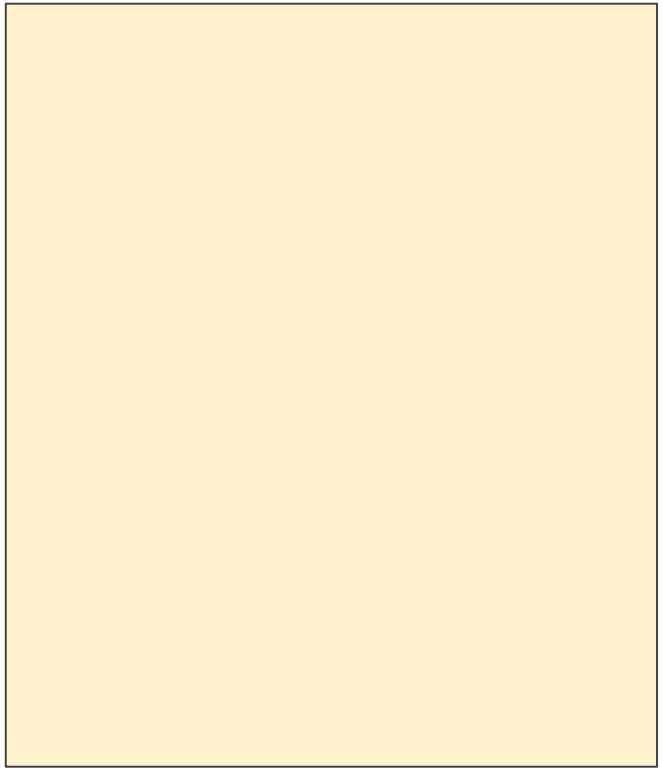
Writing to files

```
with open(filename, 'w') as f:
```

If there is a file with the name stored in filename already, this clears that file

Writing to files

```
with open(filename, 'w') as f:
```

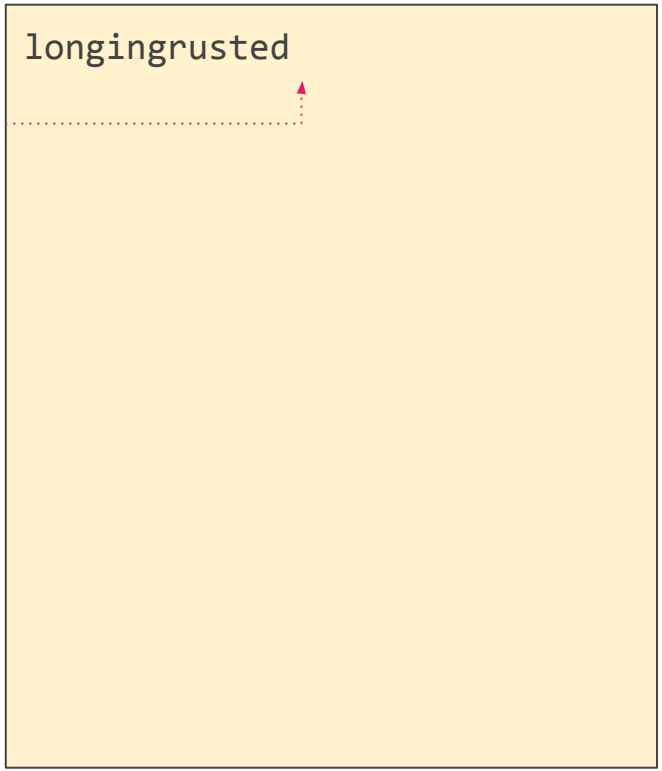


Writing to files

```
with open(filename, ‘w’) as f:
    f.write("longing")
```

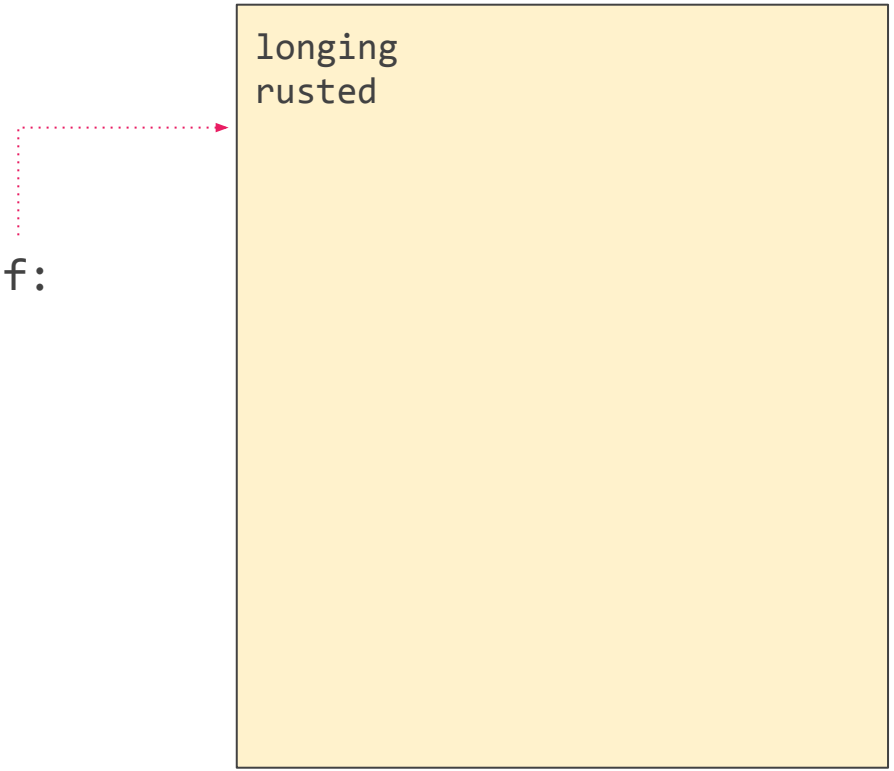
Writing to files

```
with open(filename, ‘w’) as f:
    f.write("longing")
    f.write("rusted")
```



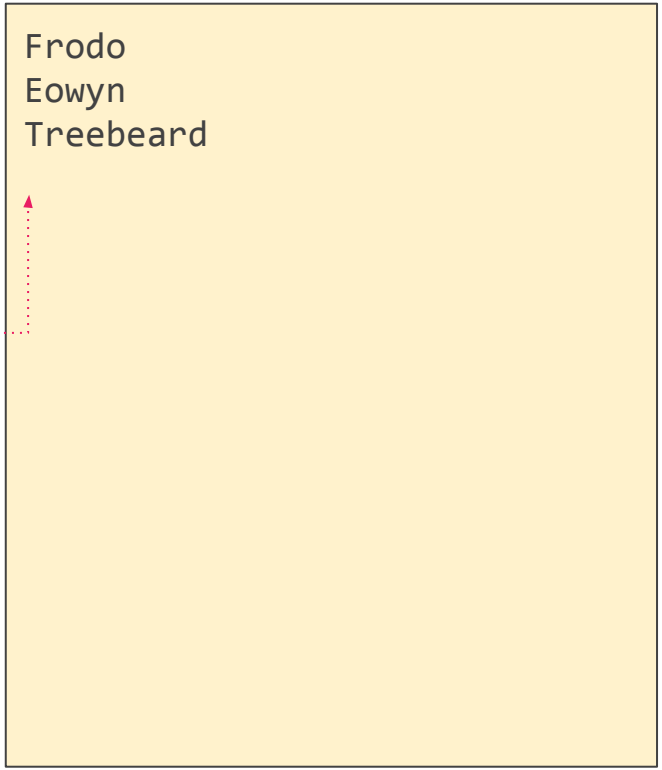
Writing to files

```
with open(filename, ‘w’) as f:
    f.write("longing\n")
    f.write("rusted\n")
```



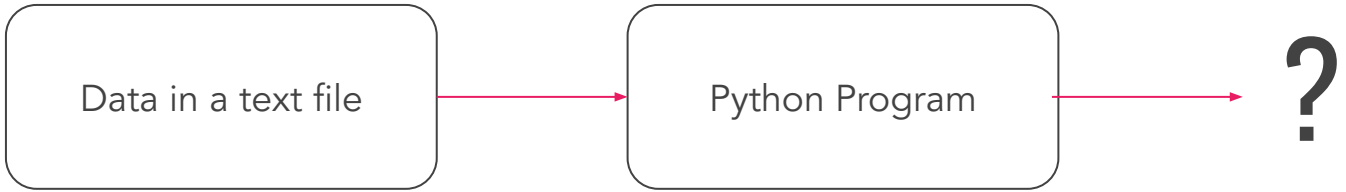
Writing to files

```
names = ["Frodo", "Eowyn", "Treebeard"]
with open(filename, 'w') as f:
    f.writelines(names)
```

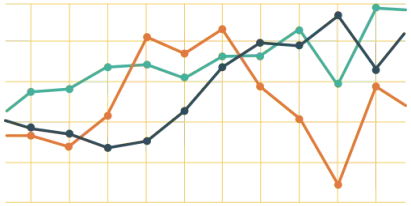


Visualizing Data

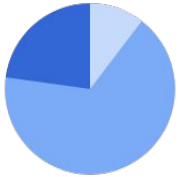
What can we do with data in a file?



What can we do with data in a file?



Data in a text file → Python Program →

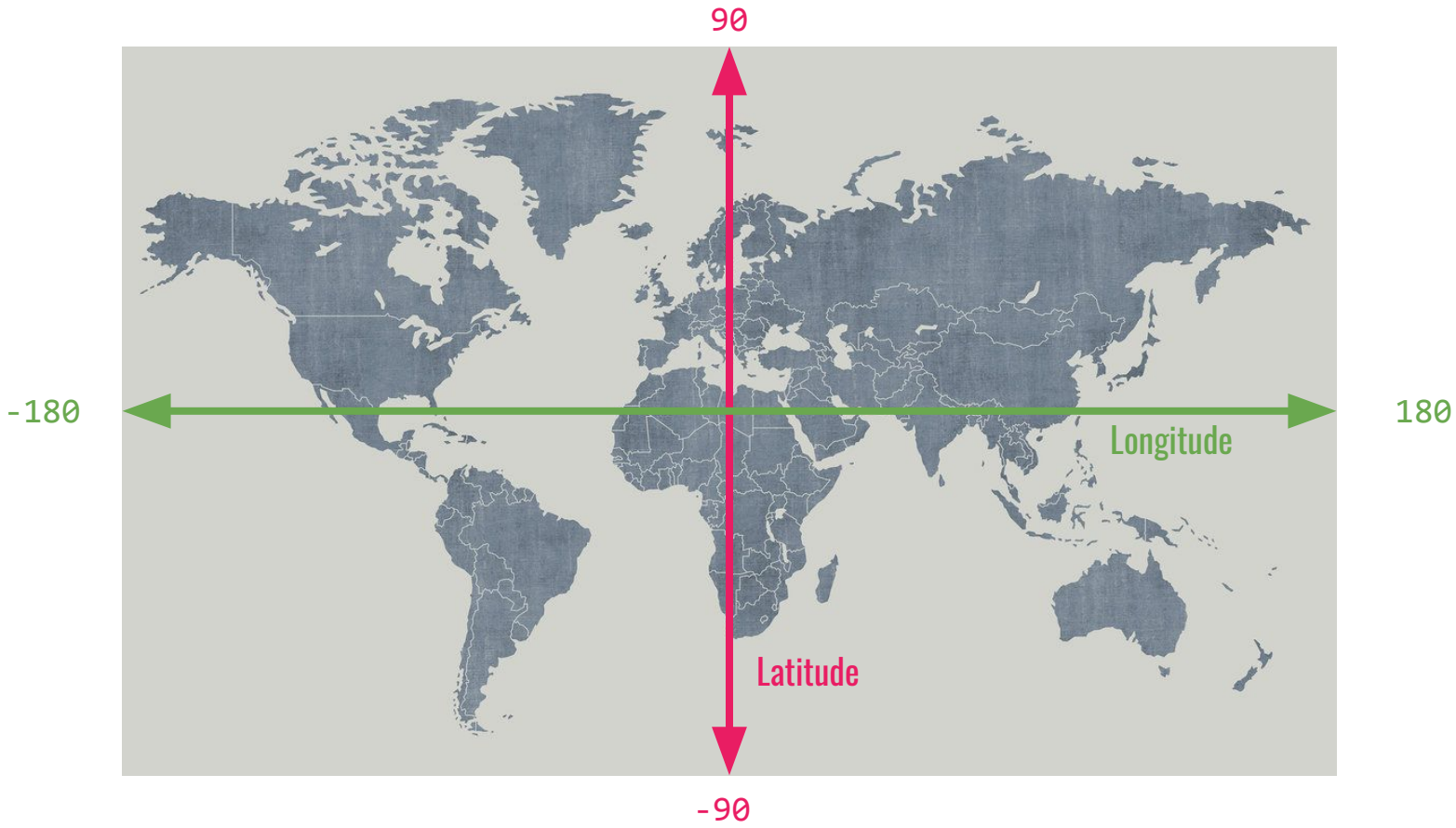


An interesting data file

United States.csv

```
City,Latitude,Longitude,Population
New York,40.6943,-73.9249,18713220
Los Angeles,34.1139,-118.4068,12750807
Chicago,41.8373,-87.6862,8604203
Miami,25.7839,-80.2102,6445545
Dallas,32.7936,-96.7662,5743938
Philadelphia,40.0077,-75.1339,5649300
Houston,29.7863,-95.3889,5464251
.
.
.
```

Defining some terms



An interesting data file

United States.csv

```
City,Latitude,Longitude,Population
New York,40.6943,-73.9249,18713220
Los Angeles,34.1139,-118.4068,12750807
Chicago,41.8373,-87.6862,8604203
Miami,25.7839,-80.2102,6445545
Dallas,32.7936,-96.7662,5743938
Philadelphia,40.0077,-75.1339,5649300
Houston,29.7863,-95.3889,5464251
.
.
.
```

An interesting data file

United States.csv

```
City,Latitude,Longitude,Population
New York,40.6943,-73.9249,18713220
Los Angeles,34.1139,-118.4068,12750807
Chicago,41.8373,-87.6862,8604203
Miami,25.7839,-80.2102,6445545
Dallas,32.7936,-96.7662,5743938
Philadelphia,40.0077,-75.1339,5649300
Houston,29.7863,-95.3889,5464251
.
.
.
```

Key idea: Represent each city as a pixel on an image

Smarter Files

Smarter Files

foundation.txt

```
His name was Gaal Dornick and he was just a
country boy who had never seen Trantor before...
```

countries.csv

```
Argentina,Buenos Aires,45M,South America
Singapore,Singapore,5.7M,Asia
New Zealand,Wellington,4.9M,Oceania
```

locations.json

```
{"Chris": "Stanford", "Brahm": "New York"}
```

`.csv` and `.json` files are more structured than arbitrary text files

Smarter Files

foundation.txt

```
His name was Gaal Dornick and he was just a
country boy who had never seen Trantor before...
```

countries.csv

```
Argentina,Buenos Aires,45M,South America
Singapore,Singapore,5.7M,Asia
New Zealand,Wellington,4.9M,Oceania
```

locations.json

```
{"Chris": "Stanford", "Brahm": "New York"}
```

We *could* manually open a `csv` or `json` file and read and process it, but that would be annoying and lead to a lot of repeated code...

Smarter Files

foundation.txt

```
His name was Gaal Dornick and he was just a
country boy who had never seen Trantor before...
```

countries.csv

```
Argentina,Buenos Aires,45M,South America
Singapore,Singapore,5.7M,Asia
New Zealand,Wellington,4.9M,Oceania
```

locations.json

```
{"Chris": "Stanford", "Brahm": "New York"}
```

...so Python gives us utilities for working with them!

Reading csv files

USA.csv

```
City,Latitude,Longitude,Population
New York,40.6943,-73.9249,18713220
Los Angeles,34.1139,-118.4068,12750807
Chicago,41.8373,-87.6862,8604203
Miami,25.7839,-80.2102,6445545
Dallas,32.7936,-96.7662,5743938
Philadelphia,40.0077,-75.1339,5649300
Houston,29.7863,-95.3889,5464251
.
.
.
```

plot_countries.py

```
import csv

def plot_country():
     with open("USA.csv") as f:
          next(f)
          reader = csv.reader(f)

          for line in reader:
               lat = float(line[1])
               lon = float(line[2])
```

Looping through a `csv.reader` splits by commas for you

Reading csv files

USA.csv

```
City,Latitude,Longitude,Population
New York,40.6943,-73.9249,18713220
Los Angeles,34.1139,-118.4068,12750807
Chicago,41.8373,-87.6862,8604203
Miami,25.7839,-80.2102,6445545
Dallas,32.7936,-96.7662,5743938
Philadelphia,40.0077,-75.1339,5649300
Houston,29.7863,-95.3889,5464251
.
.
.
```

plot_countries.py

```
import csv

def plot_country():
     with open("USA.csv") as f:
          reader = csv.DictReader(f)

          for line in reader:
               lat = float(line['Latitude'])
               lon = float(line['Longitude'])
```

Looping through a `csv.DictReader` and gives you back each line as a dictionary whose keys are column names and whose values are that column's value for that particular row

Writing csv files

data.csv

```
x,y
1,2
2,4
4,6
```

write_data.py

```
import csv

def write_data():
    with open("data.csv", "w") as f:
        writer = csv.writer(f)
        writer.writerow(["x", "y"])
        writer.writerows([
            [1,2],
            [2,4],
            [4,6]
        ])
```

A `csv.writer` allows you to write rows as lists of data, without worrying about things like `\n` characters

Writing csv files

data.csv

```
x,y
1,2
2,4
4,6
```

write_data.py

```
import csv

def write_data():
     with open("data.csv", "w") as f:
          columns = [‘x’, ‘y’]
          writer = csv.DictWriter(f)
          writer.writeheader(fieldnames=columns)
          writer.writerow({‘x’: 1, ‘y’: 2})
          writer.writerow({‘x’: 2, ‘y’: 4})
          writer.writerow({‘x’: 4, ‘y’: 6})
```

A `csv.DictWriter` allows you to write rows as dictionaries of data, without worrying about things like `\n` characters

Why use csv files?

- Great for tabular data
- Understood by Excel and Google Sheets!

Reading json files

ages.json

```
{
  "Mehran": 18,
  "Julie": 18,
  "Brahm": 24,
  "Chris": 33
}
```

```
$ python3 read_ages.py
Brahm is 24
```

read_ages.py

```
import json

def get_ages():
    with open("ages.json") as f:
        ages = json.load(f)

        age = ages['Brahm']
        print("Brahm is " + str(age))
```

`json.load` allows you to read a dictionary from a json file without having to manually process the file

Reading json files

to_buy.json

```
["Sugar", "Flour", "Salt"]
```

shopping_list.py

```
import json

def get_shopping_list():
    with open("to_buy.json") as f:
        shopping_list = json.load(f)
        print(shopping_list)
        print(len(shopping_list))
```

```
$ python3 shopping_list.py
["Sugar", "Flour", "Salt"]
3
```

`json.load` allows you to read a list from a json file without having to manually process the file

Writing json files

data.json

```
{
    "Name": "Brahm",
    "Age": 24
}
```

shopping_list.py

```
import json

def write_data():
    data = {"Name": "Brahm", "Age": 24}
    with open("data.json", "w") as f:
        json.dump(data, f, indent=4)
```

`json.dump` allows you to easily write a dictionary to a `json` file. The `indent=4` parameter makes the file more readable)

Writing json files

data.json

```
["Rorschach", "Silk Spectre"]
```

shopping_list.py

```
import json

def write_data():
    data = ["Rorschach", "Silk Spectre"]
    with open("data.json", "w") as f:
        json.dump(data, f, indent=4)
```

`json.dump` allows you to easily write a list to a `json` file.

Why use json files?

- Great for structured data that *isn't* a table
- Will convert numbers to strings and back again
- It's the language of the internet
  - JSON stands for JavaScript Object Notation, and JavaScript is one of the languages that the internet uses
- Come back next time for more...