

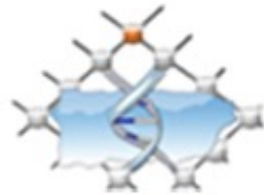


PennState
Huck Institutes of
the Life Sciences

University of Puerto Rico

Puerto Rico - IDeA Networks of Biomedical Research

Excellence



PRINBRE
IDeA Network of Biomedical Research Excellence

Python3 Part 3 – Crash Course in Python3 for Future STEM Coders

Judith S. Rodriguez-Martinez, MSc, PhD

Penn State University – University Park Campus

jzr5814@psu.edu



June 1-6, 2025



- The following material is the result of PR-INBRE research and curriculum development effort to provide a set of educational materials for research training and curriculum changes in biology programs across the island to support PR-INBRE efforts to establish a Community of Practice in Bioinformatics that offers a fruitful environment to increase computational and bioinformatics skills among traditional researchers and students (undergraduate and graduate) in the island. They have been developed as a part of the NIH funded project “**Puerto Rico IDeA Network Biomedical Research Excellence (PRINBRE)**” (Award Number 5P20GM103475).
- Unless otherwise specified, all the information contained within is Copyrighted © by University of Puerto Rico. Permission is granted for use, modify, and reproduce these materials for research and teaching purposes. A copy of the modified material should be sent to help@hpcf.upr.edu.
- Most recent versions of these presentations can be found at <http://inbre.hpcf.upr.edu/>.





Competencies

June 2, 2025 - Record and write simple and common **Python scripts to deal with Bioinformatic needs for biological data analyses** using a Jupyter Notebook

June 3, 2025 – Exploit **datatypes that can record objects**, as well as, reason **logical statements**

Today – **create your own dataset using Python**





Objectives

- **Create** a dataset
- **Open** and **Read data** of different file formats using either a **for loop** or a **Python Library**
 - TXT File
 - CSV File
- **Write** to different file formats
 - TXT File
 - CSV File
- **Import** a **Python Library** to your code
- Using Panda Library, create a **Pandas DataFrame** with columns and **export the DataFrame as a CSV File**





Objectives

- Continue to use **Google Colab as an environment** to practice and learn common **Python** lines of code.
- **Create** a new file using the function `open()`
- **Write** information to a file using the `write()` function
- Using **for Loops** analyze **lines of a file**
- **import** python libraries





Target Audience

- This training is addressed to beginners, highly motivated (eager to learn what is needed in order to be competitive without the tendency to self-limit when learning computational skills by saying that is difficult) wanting to become familiar with the **Python** programming language and become the Script Master of their bioinformatic analysis.





The importance of Python

- R is great; however, you'll eventually run into the Python language
- No matter your field, python will be a necessity (chemistry, biology, engineering, ecology, mathematics, etc)
- Knowing Python gives you an edge when job searching and navigating graduate research
- The majority bioinformatic tools are python-based and will lead you to understand under the hood code in order to successfully execute said program's purpose.





PennState
Huck Institutes of
the Life Sciences

Lesson 3.0.0

Create Files



open() function facilitates creating a new file

1

Identify the name
of your new file

```
[21] with open('first_file.txt', 'w') as f:
```



open() function facilitates creating a new file

Indicate with **'w'**
that you want to
write to this file

2

```
[21] with open('first_file.txt', 'w') as f:
```



open() function facilitates creating a new file

Assign a variable
to this new file

3

```
[21] with open('first_file.txt', 'w') as f:
```



open() function facilitates creating a new file

```
[21] with open('first_file.txt', 'w') as f:  
      f.write('hola mundo')
```

4

write() will **add**
content to a file



open() function facilitates creating a new file

```
[21] with open('first_file.txt', 'w') as f:  
      f.write('hola mundo')
```

5

Use **write()** to **write** 'hola mundo' to your new file called **f**



Success! Check out your first file!





Exercise 3.0.0

Create a file with
gene names found in
a List



How can we create this txt file?

input is a list

```
[28] crtam_genes_list = ['CRTAM_HUMAN', 'CRTAM_MOUSE', 'CADM1_HUMAN']  
     print(crtam_genes_list)
```



How can we create this txt file?

input is a list

```
[28] crtam_genes_list = ['CRTAM_HUMAN', 'CRTAM_MOUSE', 'CADM1_HUMAN']  
     print(crtam_genes_list)
```

```
['CRTAM_HUMAN', 'CRTAM_MOUSE', 'CADM1_HUMAN']
```



output the following txt file



crtam_genes.txt

```
CRTAM_HUMAN  
CRTAM_MOUSE  
CADM1_HUMAN
```



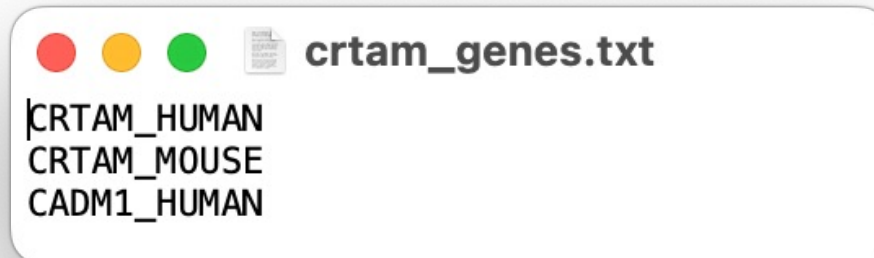
How can we create this txt file?

input is a list

```
[31] crtam_genes_list = ['CRTAM_HUMAN', 'CRTAM_MOUSE', 'CADM1_HUMAN']  
print(crtam_genes_list)
```

```
['CRTAM_HUMAN', 'CRTAM_MOUSE', 'CADM1_HUMAN']
```

Answer



How can we create this txt file?

input is a list

```
[31] crtam_genes_list = ['CRTAM_HUMAN', 'CRTAM_MOUSE', 'CADM1_HUMAN']  
print(crtam_genes_list)
```

```
['CRTAM_HUMAN', 'CRTAM_MOUSE', 'CADM1_HUMAN']
```

Answer

Tip: Use `'\n'` to indicate a new line



```
crtam_genes.txt  
CRTAM_HUMAN  
CRTAM_MOUSE  
CADM1_HUMAN
```

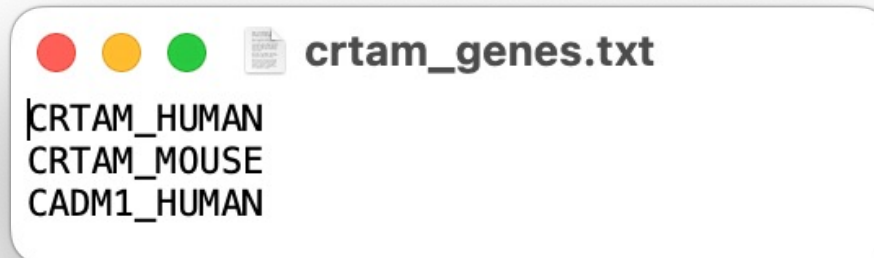
How can we create this txt file?

input is a list

```
[31] crtam_genes_list = ['CRTAM_HUMAN', 'CRTAM_MOUSE', 'CADM1_HUMAN']  
print(crtam_genes_list)
```

```
['CRTAM_HUMAN', 'CRTAM_MOUSE', 'CADM1_HUMAN']
```

```
[32] with open('crtam_genes.txt', 'w') as f:  
    for gene in crtam_genes_list:  
        f.write(gene)  
        f.write('\n')
```





PennState
Huck Institutes of
the Life Sciences

Lesson 3.1.0

Reading in your newly created files





Lesson 3.1.1

Read in the lines of a
file using a For Loop



Read in your file using a for loop

```
▶ crtam_genes_file = open('crtam_genes.txt', 'r')
```

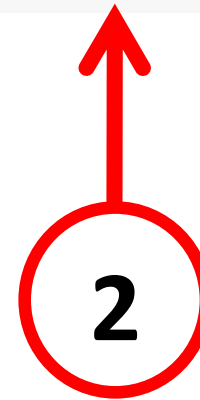
1

Use `open()`
function



Read in your file using a for loop

```
▶ crtam_genes_file = open('crtam_genes.txt', 'r')
```



Identify your
file of
interest



Read in your file using a for loop

```
▶ crtam_genes_file = open('crtam_genes.txt', 'r')
```

Indicate that
you want to
read the file
with **'r'**

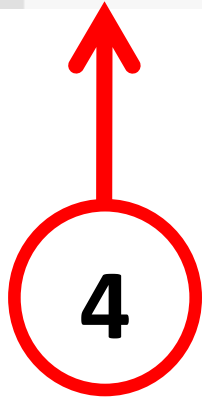
3



Read in your file using a for loop



```
crtam_genes_file = open('crtam_genes.txt', 'r')  
for crtam_genes_line in crtam_genes_file:
```



Use a for
loop to loop
through the
lines in a file



Read in your file using a for loop



```
crtam_genes_file = open('crtam_genes.txt', 'r')  
for crtam_genes_line in crtam_genes_file:  
    print(crtam_genes_line)
```

print() each lines

5



Read in your file using a for loop



```
crtam_genes_file = open('crtam_genes.txt', 'r')  
for crtam_genes_line in crtam_genes_file:  
    print(crtam_genes_line)  
crtam_genes_file.close()
```



6

close() a file is
important to
prevent xyz



Read in your file using a for loop



```
crtam_genes_file = open('crtam_genes.txt', 'r')  
for crtam_genes_line in crtam_genes_file:  
    print(crtam_genes_line)  
crtam_genes_file.close()
```

CRTAM_HUMAN

CRTAM_MOUSE

CADM1_HUMAN



How many lines are being printed out?



```
crtam_genes_file = open('crtam_genes.txt', 'r')  
for crtam_genes_line in crtam_genes_file:  
    print(crtam_genes_line)  
crtam_genes_file.close()
```

CRTAM_HUMAN

CRTAM_MOUSE

CADM1_HUMAN



How many lines are being printed out?



```
crtam_genes_file = open('crtam_genes.txt', 'r')  
for crtam_genes_line in crtam_genes_file:  
    print(crtam_genes_line)  
crtam_genes_file.close()
```

```
0 CRTAM_HUMAN  
1  
2 CRTAM_MOUSE  
3  
4 CADM1_HUMAN
```



Remove these blank lines using strip()



```
crtam_genes_file = open('crtam_genes.txt', 'r')  
for crtam_genes_line in crtam_genes_file:  
    print(crtam_genes_line)  
crtam_genes_file.close()
```

```
0 CRTAM_HUMAN  
1  
2 CRTAM_MOUSE  
3  
4 CADM1_HUMAN
```



Remove these blank lines using strip()

```
[25] crtam_genes_file = open('crtam_genes.txt', 'r')
      for crtam_genes_line in crtam_genes_file:
          print(crtam_genes_line.strip('\n'))
      crtam_genes_file.close()
```

```
0 CRTAM_HUMAN
1 CRTAM_MOUSE
2 CADM1_HUMAN
```

Step
5

Add `strip('\n')`
to line variable
to **remove**
blank lines
from a file





Lesson 3.1.2

Read in the lines of a
file using a While
Loop



Read in your file using a while loop

```
▶ crtam_genes_file = open('crtam_genes.txt', 'r')
```

1

Use `open()`
function



Read in your file using a while loop

```
▶ crtam_genes_file = open('crtam_genes.txt', 'r')  
crtam_genes_lines = crtam_genes_file.readline()
```

Using
`readline()`
function will
allow you to
while loop the
lines of a file

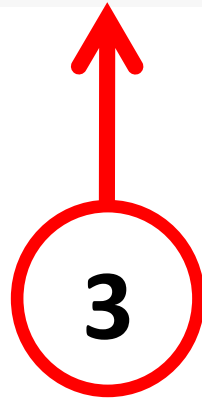
2



Read in your file using a while loop

```
▶ crtam_genes_file = open('crtam_genes.txt', 'r')  
  crtam_genes_lines = crtam_genes_file.readline()  
  while crtam_genes_lines:
```

Perform **while**
loop through
lines of this file



Read in your file using a while loop

```
▶ crtam_genes_file = open('crtam_genes.txt', 'r')  
crtam_genes_lines = crtam_genes_file.readline()  
while crtam_genes_lines:  
    print(crtam_genes_lines.strip('\n'))
```

4

**Strip and print
each line**



Read in your file using a while loop

```
▶ crtam_genes_file = open('crtam_genes.txt', 'r')  
crtam_genes_lines = crtam_genes_file.readline()  
while crtam_genes_lines:  
    print(crtam_genes_lines.strip('\n'))  
    crtam_genes_lines = crtam_genes_file.readline()
```

Reuse
`readline()` to
move on to the
next line

5



Read in your file using a while loop

```
▶ crtam_genes_file = open('crtam_genes.txt', 'r')  
crtam_genes_lines = crtam_genes_file.readline()  
while crtam_genes_lines:  
    print(crtam_genes_lines.strip('\n'))  
    crtam_genes_lines = crtam_genes_file.readline()  
crtam_genes_file.close()
```

close() your file

6



Read in your file using a while loop

```
▶ crtam_genes_file = open('crtam_genes.txt', 'r')
  crtam_genes_lines = crtam_genes_file.readline()
  while crtam_genes_lines:
      print(crtam_genes_lines.strip('\n'))
      crtam_genes_lines = crtam_genes_file.readline()
  crtam_genes_file.close()
```

```
0 CRTAM_HUMAN
1 CRTAM_MOUSE
2 CADM1_HUMAN
```





Lesson 3.2.0

Manipulate files the Panda way





PennState
Huck Institutes of
the Life Sciences

Lesson 3.2.1

Import the Panda Library



How to import a library

```
[39] import pandas as pd
```

Call **import**
to use a
library



How to import a library

```
[39] import pandas as pd
```

Call **import**
to use a
library

Name your
library of
interest



How to import a library

```
[39] import pandas as pd
```

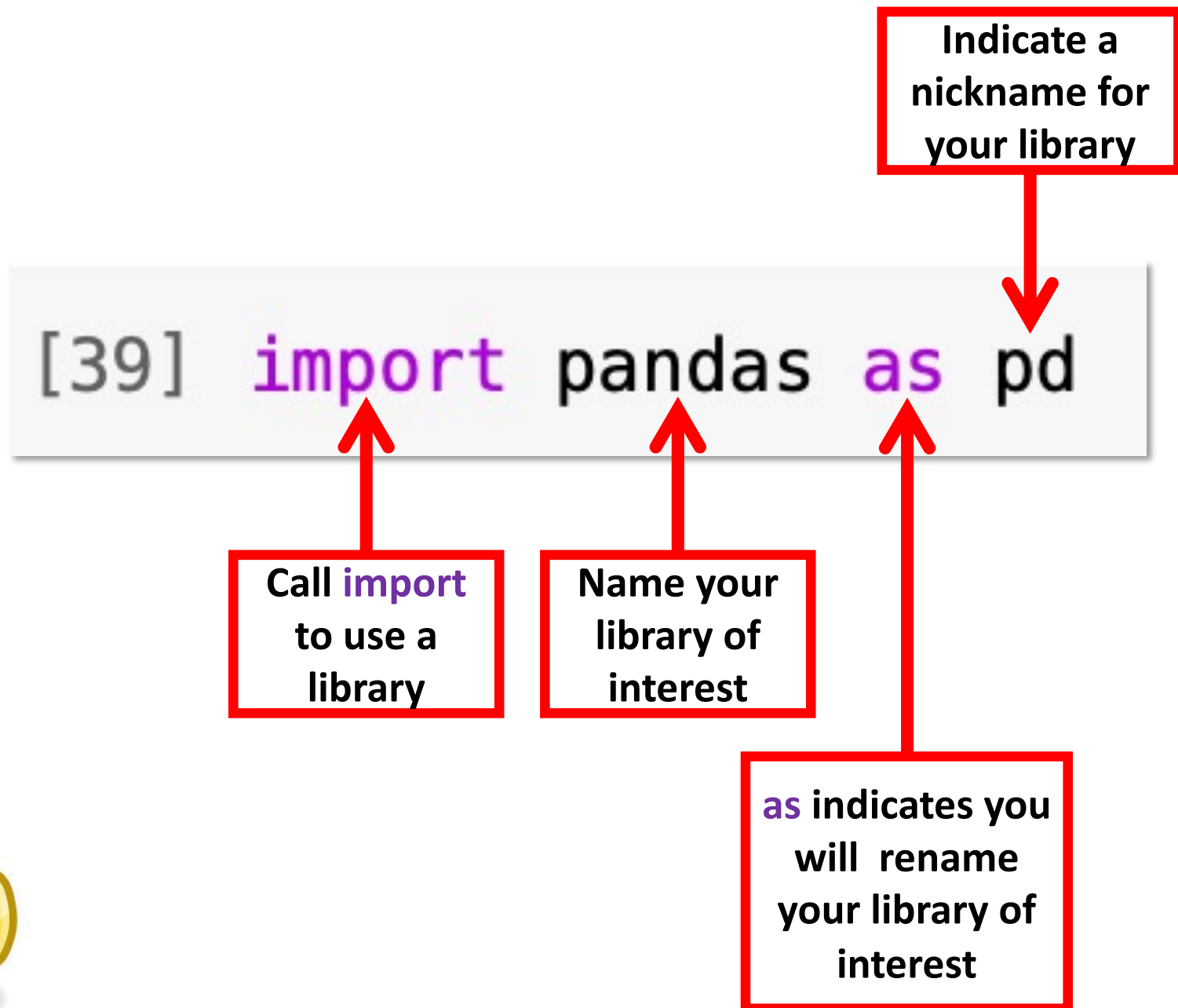
Call **import**
to use a
library

Name your
library of
interest

as indicates you
will rename
your library of
interest



How to import a library





Lesson 3.2.2

Read in a CSV file using Panda Library






Lesson 3.2.2a

Reading in a CSV file with a header using Panda Library



CSV file with header



crtam_with_header.csv

```
gene,length  
CRTAM_HUMAN,393  
CRTAM_MOUSE,781  
CADM1_HUMAN,442
```



CSV file with header


 crtam_with_header.csv

```
gene,length  
CRTAM_HUMAN,393  
CRTAM_MOUSE,781  
CADM1_HUMAN,442
```

```
[66] pd.read_csv('crtam_with_header.csv',header=0,delimiter=',')
```



CSV file with header



crtam_with_header.csv

gene	length
CRTAM_HUMAN	393
CRTAM_MOUSE	781
CADM1_HUMAN	442

```
[66] pd.read_csv('crtam_with_header.csv',header=0,delimiter=',')
```

Call **pandas**
library by its
nickname



CSV file with header

 crtam_with_header.csv

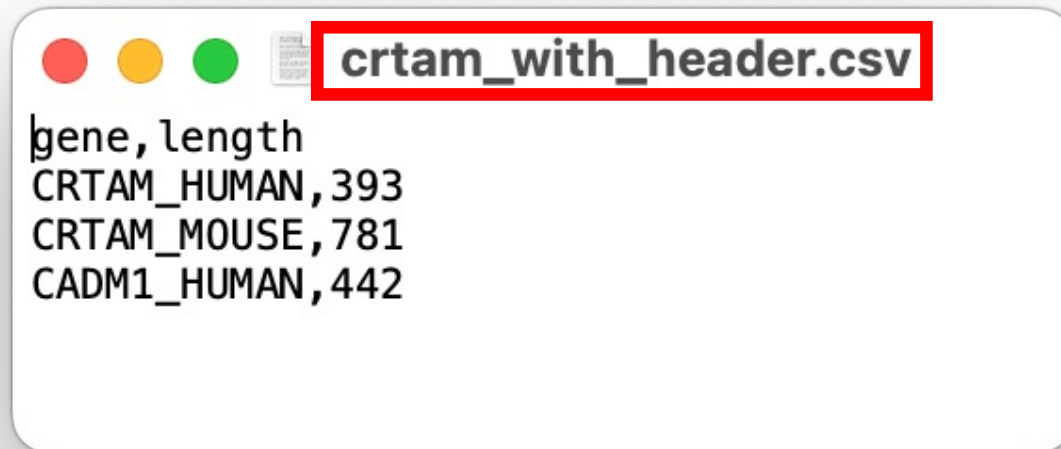
```
gene,length  
CRTAM_HUMAN,393  
CRTAM_MOUSE,781  
CADM1_HUMAN,442
```

```
[66] pd.read_csv('crtam_with_header.csv',header=0,delimiter=',')
```



Call
`read_csv()`
function for
a csv file

CSV file with header

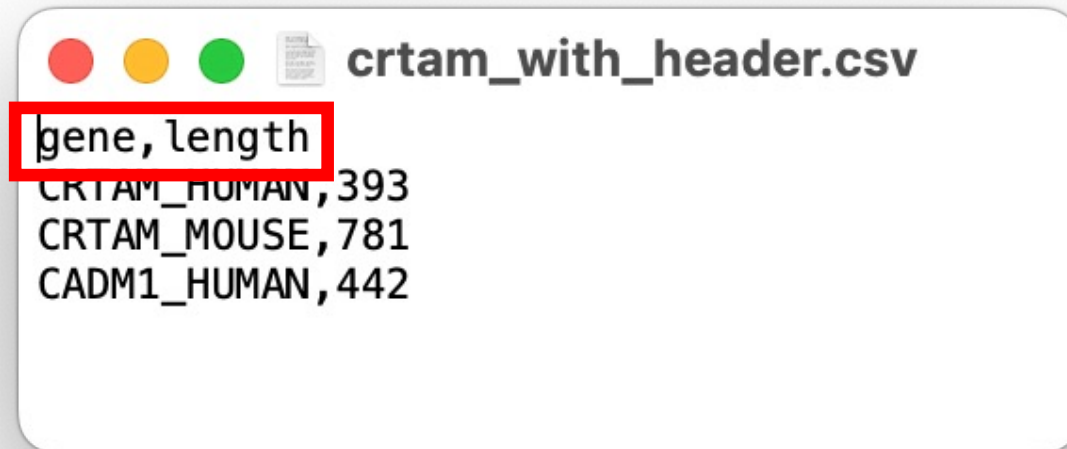


```
[66] pd.read_csv('crtam_with_header.csv', header=0, delimiter=',')
```

Name your
file of
interest



CSV file with header



The image shows a window titled 'crtam_with_header.csv'. Inside, the first row is a header: 'gene,length'. This header row is highlighted with a red box. Below it are three data rows: 'CRTAM_HUMAN,393', 'CRTAM_MOUSE,781', and 'CADM1_HUMAN,442'.

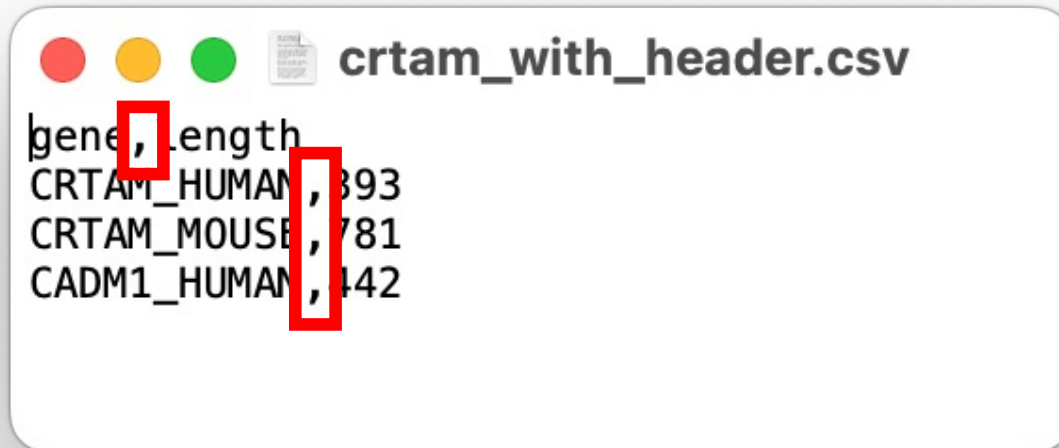
gene,length
CRTAM_HUMAN,393
CRTAM_MOUSE,781
CADM1_HUMAN,442

```
[66] pd.read_csv('crtam_with_header.csv',header=0,delimiter=',')
```

Indicate that
there is a
header exist



CSV file with header



gene	length
CRTAM_HUMAN	393
CRTAM_MOUSE	781
CADM1_HUMAN	442

```
[66] pd.read_csv('crtam_with_header.csv',header=0,delimiter=',')
```

Indicate the
type of
delimiter is

“
,”



CSV file with header

 crtam_with_header.csv

```
gene,length  
CRTAM_HUMAN,393  
CRTAM_MOUSE,781  
CADM1_HUMAN,442
```

```
[66] pd.read_csv('crtam_with_header.csv',header=0,delimiter=',')
```

Call **pandas**
library by its
nickname

Name your
file of
interest

Indicate that
there is a
header exist

Indicate the
type of
delimiter is
,

Call
read_csv()
function for
a csv file



CSV file with header

 crtam_with_header.csv

```
gene,length  
CRTAM_HUMAN,393  
CRTAM_MOUSE,781  
CADM1_HUMAN,442
```

```
[66] pd.read_csv('crtam_with_header.csv',header=0,delimiter=',')
```

	gene	length
0	CRTAM_HUMAN	393
1	CRTAM_MOUSE	781
2	CADM1_HUMAN	442

header



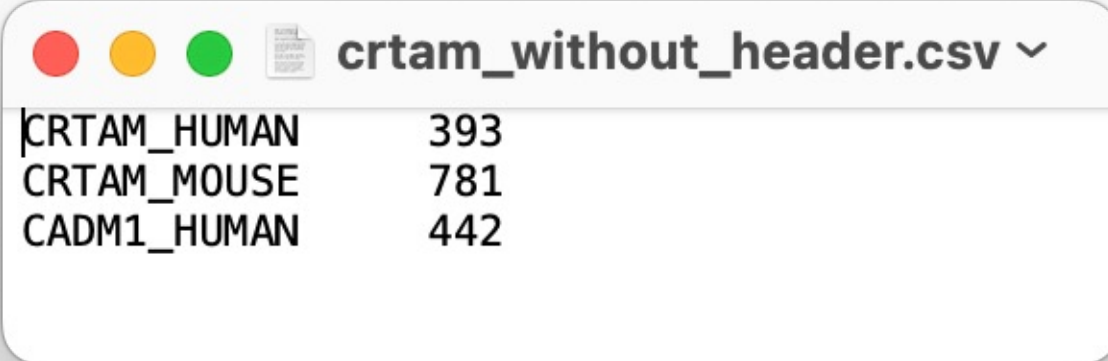


Lesson 3.2.2b

Reading in a CSV file
with no header using
Panda Library



CSV file with no header

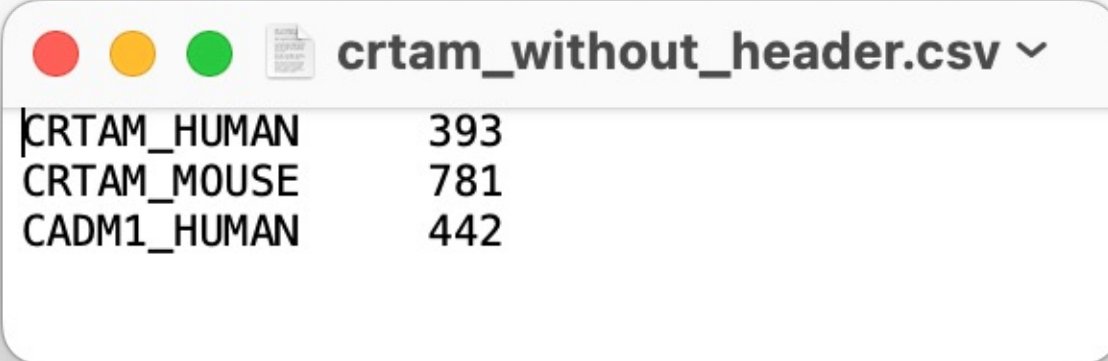


A window titled "crtam_without_header.csv" displays a CSV file. The window has three colored buttons (red, yellow, green) and a document icon. The content of the file is as follows:

CRTAM_HUMAN	393
CRTAM_MOUSE	781
CADM1_HUMAN	442



CSV file with no header

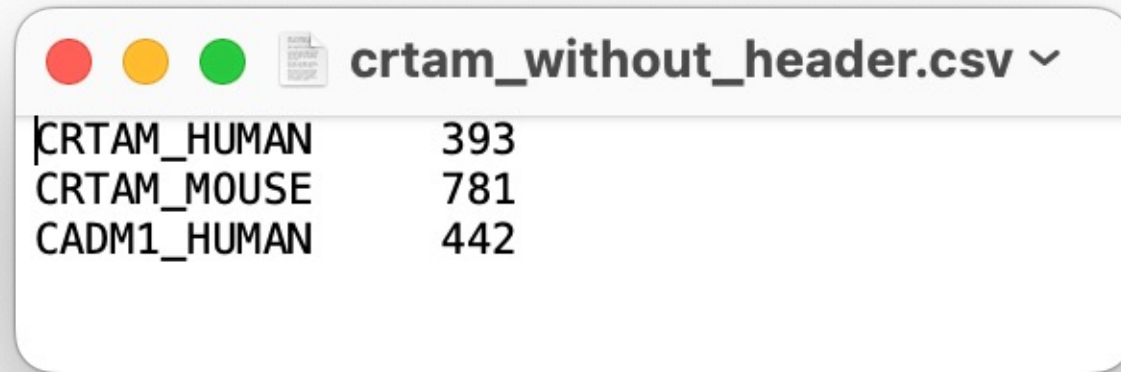


CRTAM_HUMAN	393
CRTAM_MOUSE	781
CADM1_HUMAN	442

```
[65] pd.read_csv('crtam_without_header.csv', header=None, delimiter='\t')
```



CSV file with no header



CRTAM_HUMAN	393
CRTAM_MOUSE	781
CADM1_HUMAN	442

```
[65] pd.read_csv('crtam_without_header.csv', header=None, delimiter='\t')
```

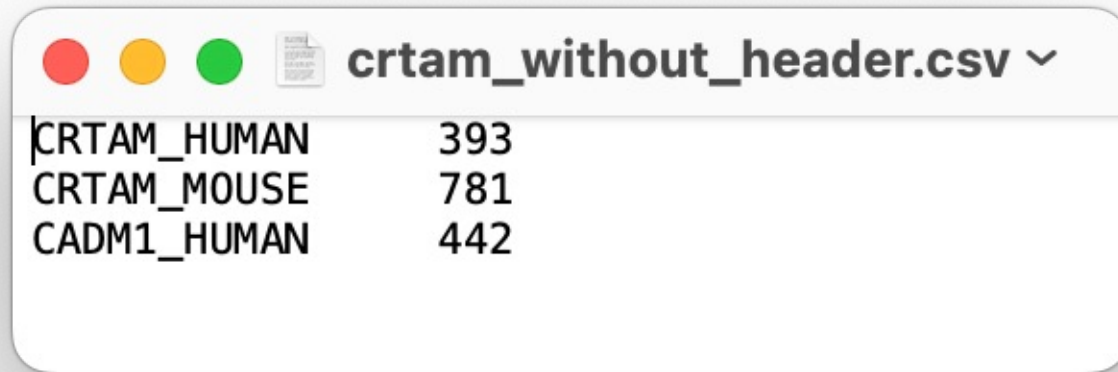
Call **pandas**
library by its
nickname

Name your
file of
interest

Call
read_csv()
function for
a csv file



CSV file with no header



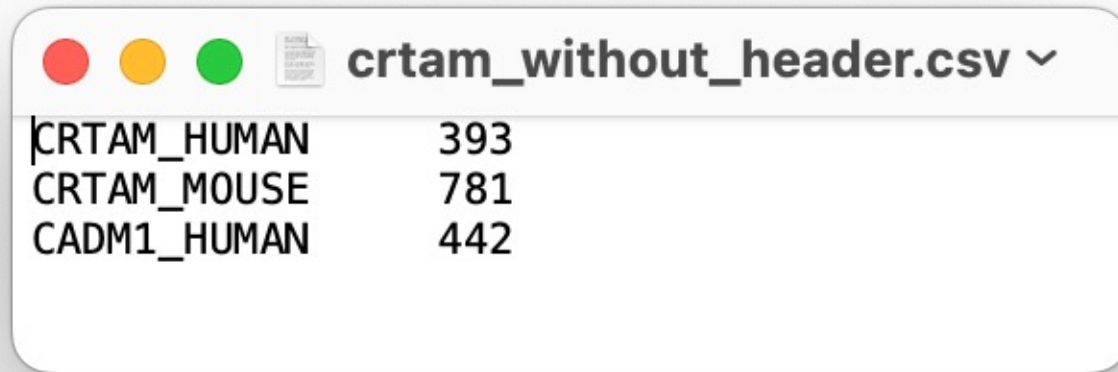
CRTAM_HUMAN	393
CRTAM_MOUSE	781
CADM1_HUMAN	442

```
[65] pd.read_csv('crtam_without_header.csv', header=None, delimiter='\t')
```

Indicate that
there is **no**
header exist



CSV file with no header



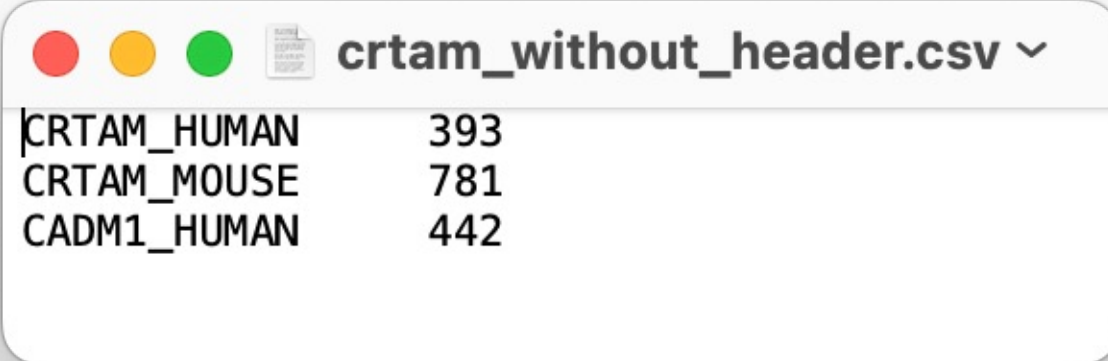
CRTAM_HUMAN	393
CRTAM_MOUSE	781
CADM1_HUMAN	442

```
[65] pd.read_csv('crtam_without_header.csv', header=None, delimiter='\t')
```

Indicate the
type of
delimiter is
'\t'



CSV file with no header



CRTAM_HUMAN	393
CRTAM_MOUSE	781
CADM1_HUMAN	442

```
[65] pd.read_csv('crtam_without_header.csv', header=None, delimiter='\t')
```

Call **pandas**
library by its
nickname

Name your
file of
interest

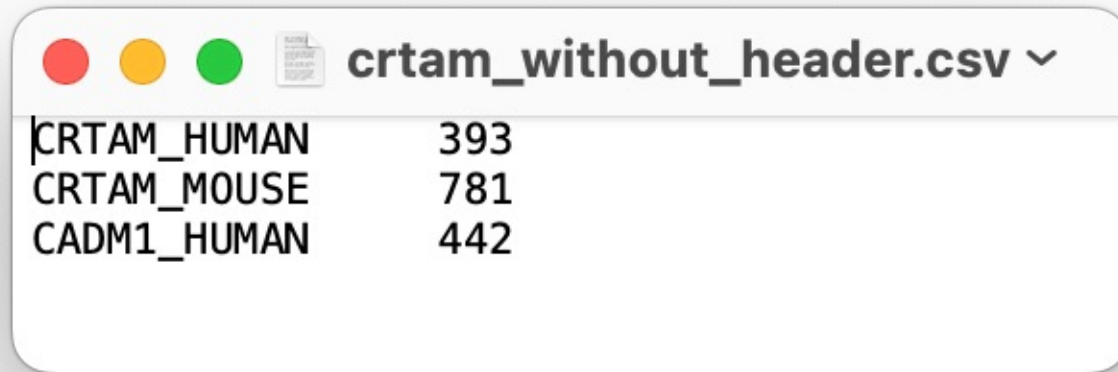
Indicate that
there is **no**
header exist

Indicate the
type of
delimiter is
'\t'

Call
read_csv()
function for
a csv file



CSV file with no header



CRTAM_HUMAN	393
CRTAM_MOUSE	781
CADM1_HUMAN	442

```
[65] pd.read_csv('crtam_without_header.csv', header=None, delimiter='\t')
```

	0	1
0	CRTAM_HUMAN	393
1	CRTAM_MOUSE	781
2	CADM1_HUMAN	442

← header





Lesson 3.2.3

Create a DataFrame
and export a new
CSV file using Panda
Library





Lesson 3.2.3a

Create a Panda DataFrame from a single List



Convert a simple list into a Pandas Dataframe

```
[80] crtam_genes_list = ['CRTAM_HUMAN', 'CRTAM_MOUSE', 'CADM1_HUMAN']
```



Convert a simple list into a Pandas Dataframe

```
[80] crtam_genes_list = ['CRTAM_HUMAN', 'CRTAM_MOUSE', 'CADM1_HUMAN']  
df = pd.DataFrame(crtam_genes_list, columns=['gene'])
```

Call **pandas**
library by its
nickname

Identify a
list of
interest

Indicate a
name for
the
information
of the single
column

Call **DataFrame()**
function to start
a Pandas
Dataframe



Convert a simple list into a Pandas Dataframe

```
[80] crtam_genes_list = ['CRTAM_HUMAN', 'CRTAM_MOUSE', 'CADM1_HUMAN']  
df = pd.DataFrame(crtam_genes_list, columns=['gene'])  
df
```

	gene
0	CRTAM_HUMAN
1	CRTAM_MOUSE
2	CADM1_HUMAN

← column name



Convert a simple list into a Pandas Dataframe

```
[80] crtam_genes_list = ['CRTAM_HUMAN', 'CRTAM_MOUSE', 'CADM1_HUMAN']  
df = pd.DataFrame(crtam_genes_list, columns=['gene'])  
df
```

	gene
0	CRTAM_HUMAN
1	CRTAM_MOUSE
2	CADM1_HUMAN

index





Lesson 3.2.3b

Create a Panda DataFrame from multiple Lists



Convert multiple lists into a Pandas Dataframe

```
[81] crtam_human_list = ['CRTAM_HUMAN', 393, 'human']  
      crtam_mouse_list = ['CRTAM_MOUSE', 781, 'mouse']  
      cadm1_human_list = ['CADM1_HUMAN', 442, 'human']
```



Convert multiple lists into a Pandas Dataframe

```
[81] crtam_human_list = ['CRTAM_HUMAN', 393, 'human']  
      crtam_mouse_list = ['CRTAM_MOUSE', 781, 'mouse']  
      cadm1_human_list = ['CADM1_HUMAN', 442, 'human']  
      df = pd.DataFrame([crtam_human_list, crtam_mouse_list, cadm1_human_list],  
                        columns=['gene', 'length', 'organisms'])
```

Call **pandas**
library by its
nickname

Call **DataFrame()**
function to start
a Pandas
Dataframe



Convert multiple lists into a Pandas Dataframe

```
[81] crtam_human_list = ['CRTAM_HUMAN', 393, 'human']  
      crtam_mouse_list = ['CRTAM_MOUSE', 781, 'mouse']  
      cadm1_human_list = ['CADM1_HUMAN', 442, 'human']  
      df = pd.DataFrame([crtam_human_list, crtam_mouse_list, cadm1_human_list],  
                          columns=['gene', 'length', 'organisms'])
```

List of Lists



Convert multiple lists into a Pandas Dataframe

```
[81] crtam_human_list = ['CRTAM_HUMAN', 393, 'human']  
      crtam_mouse_list = ['CRTAM_MOUSE', 781, 'mouse']  
      cadm1_human_list = ['CADM1_HUMAN', 442, 'human']  
      df = pd.DataFrame([crtam_human_list, crtam_mouse_list, cadm1_human_list],  
                          columns=['gene', 'length', 'organisms'])
```

Naming columns



Convert multiple lists into a Pandas Dataframe

```
[81] crtam_human_list = ['CRTAM_HUMAN', 393, 'human']  
      crtam_mouse_list = ['CRTAM_MOUSE', 781, 'mouse']  
      cadm1_human_list = ['CADM1_HUMAN', 442, 'human']  
      df = pd.DataFrame([crtam_human_list, crtam_mouse_list, cadm1_human_list],  
                        columns=['gene', 'length', 'organisms'])  
      df
```

	gene	length	organisms
0	CRTAM_HUMAN	393	human
1	CRTAM_MOUSE	781	mouse
2	CADM1_HUMAN	442	human

index



Convert multiple lists into a Pandas Dataframe

```
[81] crtam_human_list = ['CRTAM_HUMAN', 393, 'human']  
      crtam_mouse_list = ['CRTAM_MOUSE', 781, 'mouse']  
      cadm1_human_list = ['CADM1_HUMAN', 442, 'human']  
      df = pd.DataFrame([crtam_human_list, crtam_mouse_list, cadm1_human_list],  
                        columns=['gene', 'length', 'organisms'])  
      df
```

	gene	length	organisms
0	CRTAM_HUMAN	393	human
1	CRTAM_MOUSE	781	mouse
2	CADM1_HUMAN	442	human



Convert multiple lists into a Pandas Dataframe

```
[81] crtam_human_list = ['CRTAM_HUMAN', 393, 'human']  
      crtam_mouse_list = ['CRTAM_MOUSE', 781, 'mouse']  
      cadm1_human_list = ['CADM1_HUMAN', 442, 'human']  
      df = pd.DataFrame([crtam_human_list, crtam_mouse_list, cadm1_human_list],  
                        columns=['gene', 'length', 'organisms'])  
      df
```

	gene	length	organisms
0	CRTAM_HUMAN	393	human
1	CRTAM_MOUSE	781	mouse
2	CADM1_HUMAN	442	human

crtam_human_list



Convert multiple lists into a Pandas Dataframe

```
[81] crtam_human_list = ['CRTAM_HUMAN', 393, 'human']  
      crtam_mouse_list = ['CRTAM_MOUSE', 781, 'mouse']  
      cadm1_human_list = ['CADM1_HUMAN', 442, 'human']  
      df = pd.DataFrame([crtam_human_list, crtam_mouse_list, cadm1_human_list],  
                        columns=['gene', 'length', 'organisms'])  
      df
```

	gene	length	organisms
0	CRTAM_HUMAN	393	human
1	CRTAM_MOUSE	781	mouse
2	CADM1_HUMAN	442	human

crtam_mouse_list



Convert multiple lists into a Pandas Dataframe

```
[81] crtam_human_list = ['CRTAM_HUMAN', 393, 'human']  
      crtam_mouse_list = ['CRTAM_MOUSE', 781, 'mouse']  
      cadm1_human_list = ['CADM1_HUMAN', 442, 'human']  
      df = pd.DataFrame([crtam_human_list, crtam_mouse_list, cadm1_human_list],  
                        columns=['gene', 'length', 'organisms'])  
      df
```

	gene	length	organisms
0	CRTAM_HUMAN	393	human
1	CRTAM_MOUSE	781	mouse
2	CADM1_HUMAN	442	human

cadm1_human_list





Lesson 3.2.3c

Create a Pandas DataFrame from a Nested Dictionary



Convert a nested dictionary into a Panda DataFrame

```
[83] crtam_dictionary = {'CRTAM_HUMAN':{'length':393,'organism':'human'},  
                        'CRTAM_MOUSE':{'length':781,'organism':'mouse'},  
                        'CADM1_HUMAN':{'length':442,'organism':'human'}}
```



Convert a nested dictionary into a Panda DataFrame

```
[83] crtam_dictionary = {'CRTAM_HUMAN':{'length':393,'organism':'human'},  
                        'CRTAM_MOUSE':{'length':781,'organism':'mouse'},  
                        'CADM1_HUMAN':{'length':442,'organism':'human'}}  
df = pd.DataFrame(crtam_dictionary)
```



Convert a nested dictionary into a Panda DataFrame

```
[83] crtam_dictionary = {'CRTAM_HUMAN':{'length':393,'organism':'human'},  
                        'CRTAM_MOUSE':{'length':781,'organism':'mouse'},  
                        'CADM1_HUMAN':{'length':442,'organism':'human'}}  
  
df = pd.DataFrame(crtam_dictionary)  
df
```

	CRTAM_HUMAN	CRTAM_MOUSE	CADM1_HUMAN
length	393	781	442
organism	human	mouse	human



Convert a nested dictionary into a Panda DataFrame

```
[86] crtam_dictionary = {'CRTAM_HUMAN':{'length':393,'organism':'human'},  
                        'CRTAM_MOUSE':{'length':781,'organism':'mouse'},  
                        'CADM1_HUMAN':{'length':442,'organism':'human'}}  
df = pd.DataFrame(crtam_dictionary).T
```



Convert a nested dictionary into a Panda DataFrame

```
[86] crtam_dictionary = {'CRTAM_HUMAN':{'length':393,'organism':'human'},  
                        'CRTAM_MOUSE':{'length':781,'organism':'mouse'},  
                        'CADM1_HUMAN':{'length':442,'organism':'human'}}  
  
df = pd.DataFrame(crtam_dictionary).T  
df
```

	length	organism
CRTAM_HUMAN	393	human
CRTAM_MOUSE	781	mouse
CADM1_HUMAN	442	human



Convert a nested dictionary into a Panda DataFrame

```
[87] crtam_dictionary = {'CRTAM_HUMAN':{'length':393,'organism':'human'},  
                        'CRTAM_MOUSE':{'length':781,'organism':'mouse'},  
                        'CADM1_HUMAN':{'length':442,'organism':'human'}}  
df = pd.DataFrame(crtam_dictionary).T.reset_index()
```



Convert a nested dictionary into a Panda DataFrame

```
[87] crtam_dictionary = {'CRTAM_HUMAN':{'length':393,'organism':'human'},  
                        'CRTAM_MOUSE':{'length':781,'organism':'mouse'},  
                        'CADM1_HUMAN':{'length':442,'organism':'human'}}  
  
df = pd.DataFrame(crtam_dictionary).T.reset_index()  
df
```

	index	length	organism
0	CRTAM_HUMAN	393	human
1	CRTAM_MOUSE	781	mouse
2	CADM1_HUMAN	442	human



Convert a nested dictionary into a Panda DataFrame

```
[89] crtam_dictionary = {'CRTAM_HUMAN':{'length':393,'organism':'human'},  
                        'CRTAM_MOUSE':{'length':781,'organism':'mouse'},  
                        'CADM1_HUMAN':{'length':442,'organism':'human'}}  
df = pd.DataFrame(crtam_dictionary).T.reset_index().rename(columns={'index':'gene'})
```



Convert a nested dictionary into a Panda DataFrame

```
[89] crtam_dictionary = {'CRTAM_HUMAN':{'length':393,'organism':'human'},  
                        'CRTAM_MOUSE':{'length':781,'organism':'mouse'},  
                        'CADM1_HUMAN':{'length':442,'organism':'human'}}  
df = pd.DataFrame(crtam_dictionary).T.reset_index().rename(columns={'index':'gene'})  
df
```

	gene	length	organism
0	CRTAM_HUMAN	393	human
1	CRTAM_MOUSE	781	mouse
2	CADM1_HUMAN	442	human





Lesson 3.2.4

Export a Panda DataFrame as a CSV File





Export a nested dictionary to a CSV file

```
[89] crtam_dictionary = {'CRTAM_HUMAN':{'length':393,'organism':'human'},  
                        'CRTAM_MOUSE':{'length':781,'organism':'mouse'},  
                        'CADM1_HUMAN':{'length':442,'organism':'human'}}  
  
df = pd.DataFrame(crtam_dictionary).T.reset_index().rename(columns={'index':'gene'})  
df
```


	gene	length	organism
0	CRTAM_HUMAN	393	human
1	CRTAM_MOUSE	781	mouse
2	CADM1_HUMAN	442	human



  crtam_information.csv

```
,gene,length,organism  
0,CRTAM_HUMAN,393,human  
1,CRTAM_MOUSE,781,mouse  
2,CADM1_HUMAN,442,human
```

Export a nested dictionary to a CSV file



	gene	length	organism
0	CRTAM_HUMAN	393	human
1	CRTAM_MOUSE	781	mouse
2	CADM1_HUMAN	442	human


```
[90] df.to_csv('crtam_information.csv', sep=',')
```



Call variable
of your
dataframe



Export a nested dictionary to a CSV file




	gene	length	organism
0	CRTAM_HUMAN	393	human
1	CRTAM_MOUSE	781	mouse
2	CADM1_HUMAN	442	human

```
[90] df.to_csv('crtam_information.csv', sep=',')
```



Call `to_csv()`
function to start
a Pandas
Dataframe

Export a nested dictionary to a CSV file




	gene	length	organism
0	CRTAM_HUMAN	393	human
1	CRTAM_MOUSE	781	mouse
2	CADM1_HUMAN	442	human

```
[90] df.to_csv('crtam_information.csv', sep=',')
```

Name your
file of
interest



Export a nested dictionary to a CSV file




	gene	length	organism
0	CRTAM_HUMAN	393	human
1	CRTAM_MOUSE	781	mouse
2	CADM1_HUMAN	442	human

```
[90] df.to_csv('crtam_information.csv', sep=',')
```

Indicate the
type of
separator or
delimiter is
“
,”



Export a nested dictionary to a CSV file



	gene	length	organism
0	CRTAM_HUMAN	393	human
1	CRTAM_MOUSE	781	mouse
2	CADM1_HUMAN	442	human

```
[90] df.to_csv('crtam_information.csv', sep=',')
```

Call variable
of your
dataframe

Call `to_csv()`
function to start
a Pandas
Dataframe

Name your
file of
interest

Indicate the
type of
separator or
delimiter is
,



Check out your new CSV file!

 **crtam_information.csv**

```
,gene,length,organism  
0,CRTAM_HUMAN,393,human  
1,CRTAM_MOUSE,781,mouse  
2,CADM1_HUMAN,442,human
```





Proficiency assessment

Show off your new skills!

1. Read in the Fasta File called crtam.faa
2. Choose at least 3 of the following options below to create a dataframe with your columns of interest.
 - Gene_names
 - Length in amino acid
 - Length in it's DNA sequence
 - Total RSNNEETS
 - Total KAH
 - Total PGLRLRLLLL
3. Export a file to import into Excel

