Week 5

CSC119

Inputs and Outputs

Inputs:

- From the keyboard, file, network, etc.

Outputs:

- Print to screen, write to file, network etc.

String Manipulation

- ```
  my_string = "hello there, i'm a python program, thanks"
  ```
- Concatenation - adding one string to the end of another
  - ```
    new_string = my_string + " and keep learning"
    ```
- Substring - selecting a portion of the string
  - ```
    new_string = my_string[0:5] # just hello
    ```
- Split - make list of strings split on a certain character
  - ```
    string_parts = my_string.split(",")
    #makes a list ['hello there', 'i'm a python program', 'thanks']
    ```
- Contains - test if one string is in another
  - ```
    if "python" in my_string:
        print("thanks for learning python")
    ```
    ```

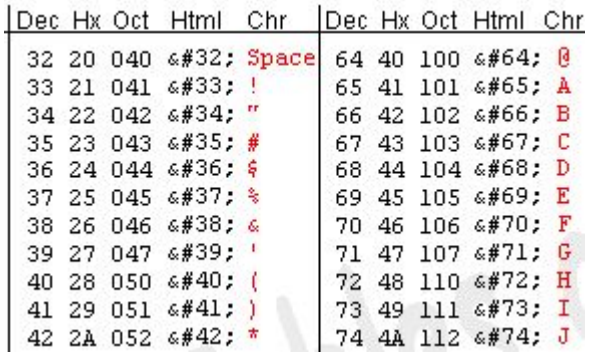
Reading from a file

- File is a sequence of bytes on the hard drive
- To access data contained in the file, it must be parsed by our program
- Open a file, read one byte at a time until end of file
- How the bytes are organized is the format of the file
  - JSON
  - XML
  - HTML

Text Files

- Text files contain text readable characters
- Text files are organized by lines
- Read text files line by line

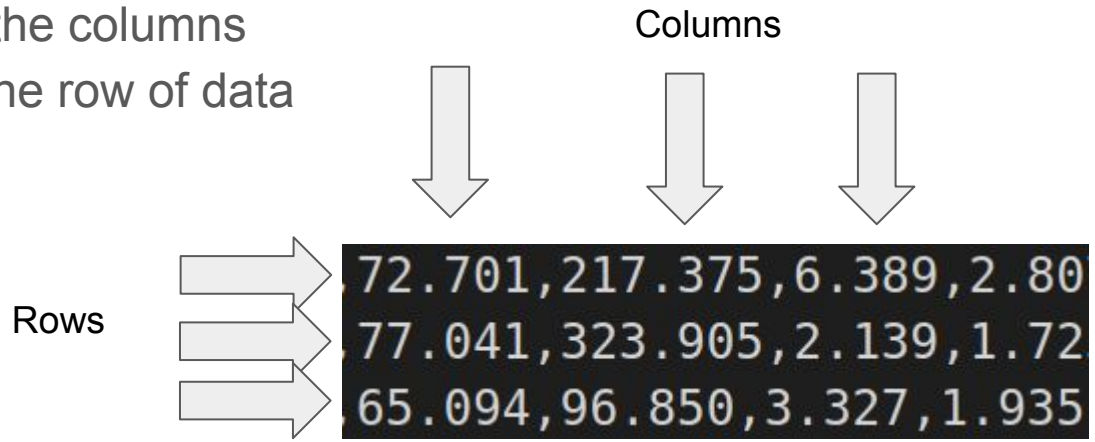
```
#read each line in the file
with open("mydata.csv") as f:
    for line in f.readlines():
        print(line)
```



Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
32	20	040		Space	64	40	100	@	@
33	21	041	!	!	65	41	101	A	A
34	22	042	"	"	66	42	102	B	B
35	23	043	#	#	67	43	103	C	C
36	24	044	$	$	68	44	104	D	D
37	25	045	%	%	69	45	105	E	E
38	26	046	&	&	70	46	106	F	F
39	27	047	'	'	71	47	107	G	G
40	28	050	(	(	72	48	110	H	H
41	29	051	)	)	73	49	111	I	I
42	2A	052	*	*	74	4A	112	J	J

Comma Separated Value files

- Values in each line are separated by commas
- Each line is a piece of data
- First line in file defines the columns
- Each line in the file is one row of data



Parsing Data

- Need to bring data into our program so we can use it
- Using lists and/or dictionaries, the data can be organized in a useful manner
- How the data from the file is brought into the program is called parsing

```
csv_data = {} #empty dict to store our data
with open("US_points_hourly_CST.csv") as f:
    header = f.readline() # read first line as header, column names separated by comma
    header_names = header.split(",") # list of columns
    num_columns = len(header_names)
    for line in f.readlines(): # read the rest of the file
        values = line.split(",") # split each line by commas
        for i in range(num_columns): # now loop over each value
            column_name = header_names[i] #
            value = values[i]
            if not column_name in csv_data:
                # if this column_name isn't in our dictionary,
                # add it as an empty list
                csv_data[column_name] = []
            csv_data[column_name].append(value)

print(csv_data['WindSpeed'])
```

Easier way

In [7]:

```
import pandas as pd

data = pd.read_csv("US_points_hourly_CST.csv")

print(data['WindSpeed'])
```

```
0       2.807
1       1.725
2       1.935
3       6.872
4       2.795
        ...
2203    5.311
2204    4.114
2205    3.435
2206    2.877
2207    5.640
Name: WindSpeed, Length: 2208, dtype: float64
```