

DS2000

2/17 - Fri

### Admin

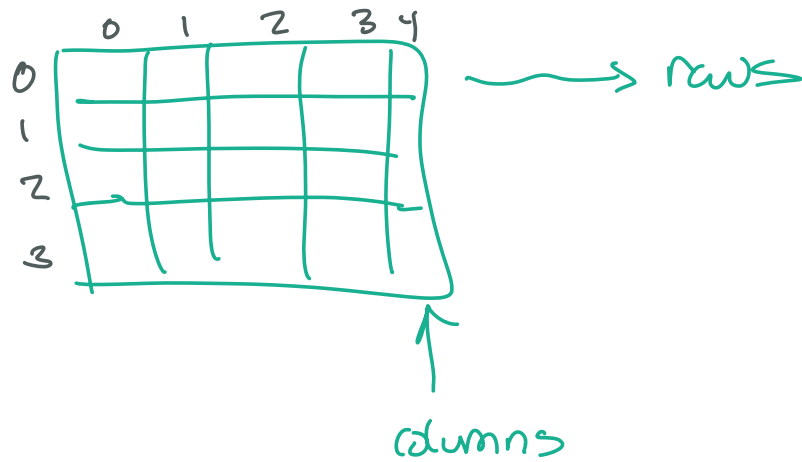
- HW 4 due 4pm
- HW 5 out, due 2/24
- mini vtz due 3/3 (~~due~~ no late ones)

### Agenda

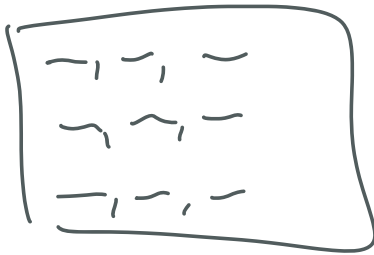
1. 2D lists
2. Functions for today's data
3. Python

#### 1. 2-Dimensional list

- data structure - one label, multiple values
- 2D list - list where every element is a list
- like a spreadsheet



Start: CSV File



~ .csv

2D list  
every row is one element

$[[~,~,~],$   
 $[~,~,~],$   
 $[~,~,~]]$

Every data structure, we need to know:

1. Create ~  $lst = []$

2. Add ~ add a list to the list

$lst.append([4, 5, 6])$

$lst.append([7, 8, 9])$

$print(len(lst))$  # 2

$\begin{matrix} & 0 & 1 & 2 & \text{cols} \\ \text{rows} \rightarrow & 0 & [4, 5, 6] \\ & 1 & [7, 8, 9] \end{matrix}$

3. Look at one thing

`print(lst[0])` # a row `[4, 5, 6]`

`print(lst[0][0])` # row 0 col 0 4  
row col

`print(lst[1][2])` # row 1 col 2 9

4. Look at all the things

• nested for loops,

```
for row in lst:
    for item in row:
        print(item)
```

→ `row = [4, 5, 6]`  
`item = 4`  
`print(item)`  
`item = 5`  
`print(item)`  
`item = 6`  
`print(item)`

`row = [7, 8, 9]`  
`item = 7`  
`print(item)`  
...

• the inner loop runs all iterations for every time the outer loop runs

```

for i in range(len(lst)):
    for j in range(len(lst[i])):
        print(lst[i][j])

```

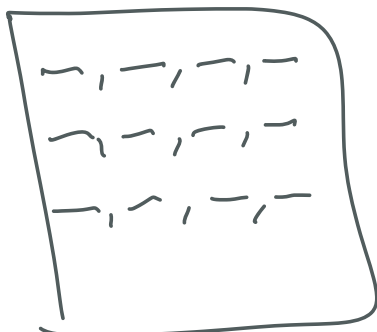
*i* = 0  
*j* = 0  
*j* = 1  
*j* = 2

print(lst[0][0])  
 print(lst[0][1])  
 print(lst[0][2])

*i* = 1  
*j* = 0  
*j* = 1  
*j* = 2  
 print(lst[1][0])  
 print(lst[1][1])  
 print(lst[1][2])

## 2. Functions for CSV File

Today's file: PO1 - GPS file



DR\_US.csv

(at long city address  
 {[-,-,-,-],  
 [-,-,-,-],  
 [-,-,-,-]}  
 ...

What Dunks are  
 in ~~Boston~~?  
 city

## Functions:

name: read-csv

param: filename

returns: 2D list of strings

name: get-local

params: 2D list, name of city, city column

returns: smaller 2D list

(everything that matches the city)