Lecture 15: Nested Lists

CS 51P

October 30, 2019

Lists

- a list is an ordered set of elements:

```
[3, 6, 2, 1]
```

- many ways to create a list including:

```
a_list = [3, 6, 2, 1]
b_list = []
c_list = "a b c d".split()
d_list = open("temp.txt","r").readlines()
```

- a list is a sequence, so can index into, loop over, check for membership, slice, etc
- operators: + and *
- lists are mutable

adding to a list

- a_list.extend(*list*)
- a_list.append(*elem*)
- a_list.insert(*index*, *elem*)

other

- min(a_list), max(a_list), len(a_list)
- *elem* in a_list
  - returns bool
- a_list.index(*elem*)
  - returns int or error

removing from a list

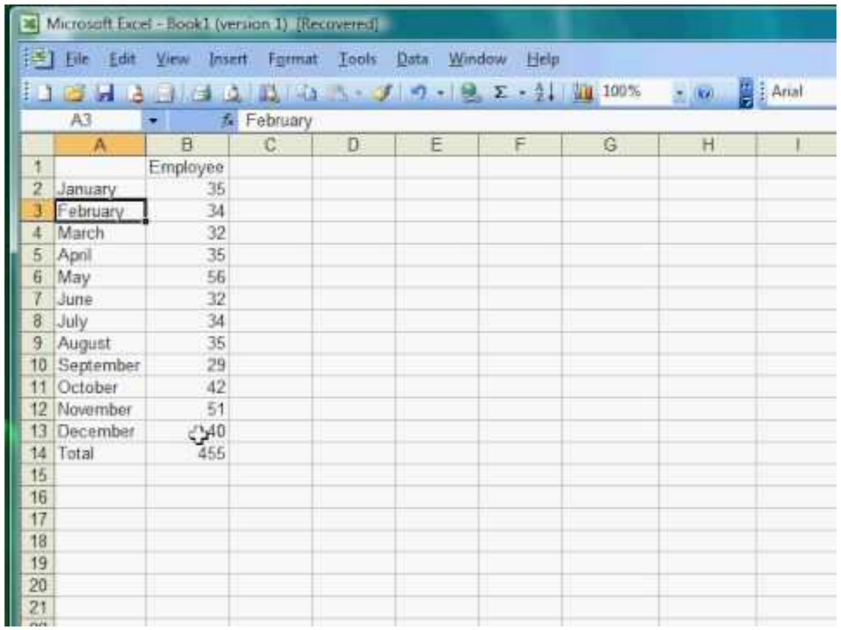
- del(a_list[*slice*])
- a_list.remove(*elem*)
  - error if *elem* not in a_list
- a_list.pop()
  - returns (and removes) a_list[-1]
- a_list.pop(*index*)
  - returns (and removes) a_list[index]

modifying a list

- direct assignment

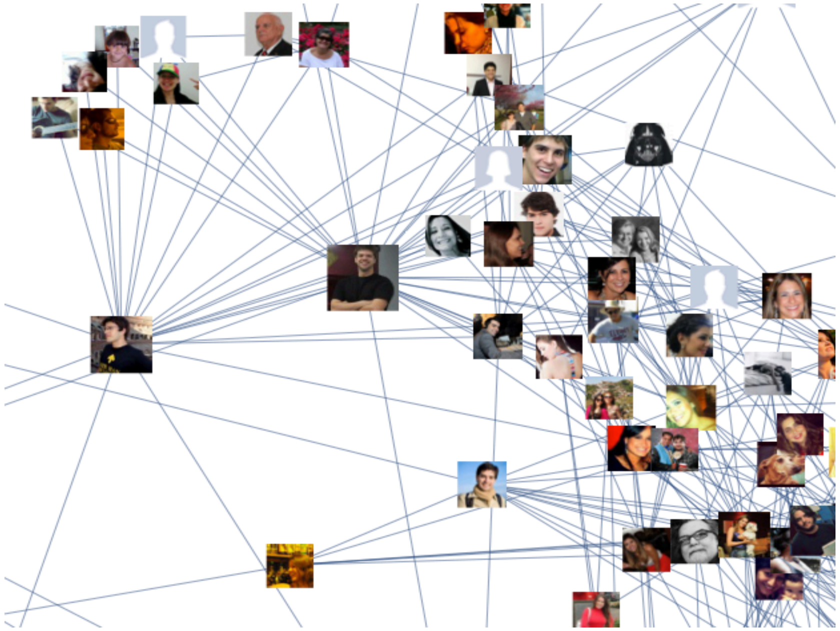
Matrices

- Can think of lists as a one-dimensional matrix
- What if you want to use a two-dimensional matrix?
- Can create a **list of lists** aka a **nested list**!


Microsoft Excel - Book1 (version 1) [Recovered]
File Edit View Insert Format Tools Data Window Help
100%
Arial

A3 February

	A	B
1		Employee
2	January	35
3	February	34
4	March	32
5	April	35
6	May	56
7	June	32
8	July	34
9	August	35
10	September	29
11	October	42
12	November	51
13	December	40
14	Total	455



Example

```
a_list = [ [4, [True, False], 6, 8], [888, 999] ]

if alist[0][1][0]:
    print(alist[1][0])
else:
    print(alist[1][1])
```

Example

- Define a function nested_total that takes a list of lists of ints and returns the sum of all the values.

```
list = [[1,2], [3], [4,5,6]]
sum = nested_total(list)
print(sum)
```

```
21
```

Exercise

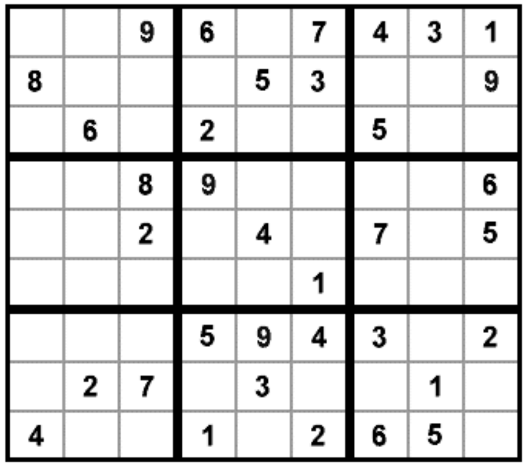
- Define a function nested_avg that takes a list of lists of ints and returns a list with each sublist averaged

```
list = [[1,2], [3], [4,5,6]]
list_avg = nested_avg(list)
print(list_avg)
```

```
[1.5, 3.0, 5.0]
```

Example

LEVEL: Beginner



		9	6		7	4	3	1
8				5	3			9
	6		2			5		
		8	9					6
		2		4		7		5
					1			
			5	9	4	3		2
	2	7		3			1	
4			1		2	6	5	

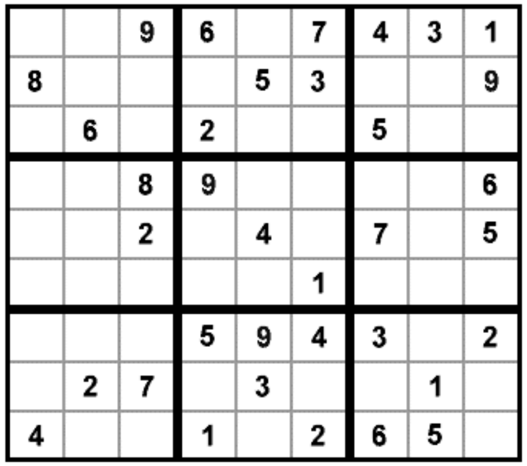
www.dctech.com/sudoku/

```
board = [[0,0,9,6,0,7,4,3,1],
         [8,0,0,0,5,3,0,0,9],
         [0,6,0,2,0,0,5,0,0],
                   ...
         [4,0,0,1,0,2,6,5,0]]
```

- write a function `set_value` that takes a nested list board and ints i, j, n and updates the (i,j)th entry of board to be the value n
- write a function `check_row_i` that takes an int i and a nested list board. The function should return True if and only if row i contains each integer from 1 through 9 exactly once.

Exercise

LEVEL: Beginner



		9	6		7	4	3	1
8				5	3			9
	6		2			5		
		8	9					6
		2		4		7		5
					1			
			5	9	4	3		2
	2	7		3			1	
4			1		2	6	5	

www.dctech.com/sudoku/

```
board = [[0,0,9,6,0,7,4,3,1],
         [8,0,0,0,5,3,0,0,9],
         [0,6,0,2,0,0,5,0,0],
                   ...
         [4,0,0,1,0,2,6,5,0]]
```

- write a function `check_column_i` that takes an int i and a nested list board. The function should return True if and only if column i contains each integer from 1 through 9 exactly once.
- write a function `check_block_ij` that takes ints i and j and a nested list board. The function should return True if and only if the 3x3 block starting at row i, column j contains each integer from 1 through 9 exactly once
- write a function `check_solution` that takes a nested list board and returns True if and only if board represents a correctly solved puzzle