Programming Foundations in Python
Adapted From: CMSC 201 Computer Science I for Majors

Lecture 12 – Lists Part 2

Last Class We Covered

- Value-returning functions
  - **`None`**
  - Common errors
- Function scope

#TBT

Infinite **`while`** loops

Any Questions from Last Time?

Today’s Objectives

- To review what we know about lists already

- To learn more about lists in Python
- To understand two-dimensional lists
  - (And more dimensions!)
- To practice passing lists to functions
- To learn about mutability and its uses

List Review

Vital List Algorithm: Iterating

- Write the code to iterate over and print out the contents of a list called **`classNames`**

```
index = 0
while index < len(classNames):
    print( classNames[index] )
    index += 1
```

Two-Dimensional Lists

Two-Dimensional Lists

- Lists can hold any type (int, string, float, etc.)
  - This means they can also hold another list
- We've looked at lists as being one-dimensional
  - But lists can also be two- (or three- or four- or five-, etc.) dimensional!

Two-Dimensional Lists: Syntax

- We use square brackets to indicate lists
  - 2D lists are essentially a list of lists
  - What do you think the syntax will look like?

```
twoD = [ ["first", "row"], ["second",
"row"], ["last", "row"] ]
```

```
twoD = [ ["first",  "row"],
         ["second", "row"],
         ["last",   "row"]   ]
```

Same code, just lined up to be more readable

Two-Dimensional Lists: A Grid

- It may help to think of 2D lists as a grid

```
twoD = [ [1,2,3], [4,5,6], [7,8,9] ]
```

1	2	3
4	5	6
7	8	9

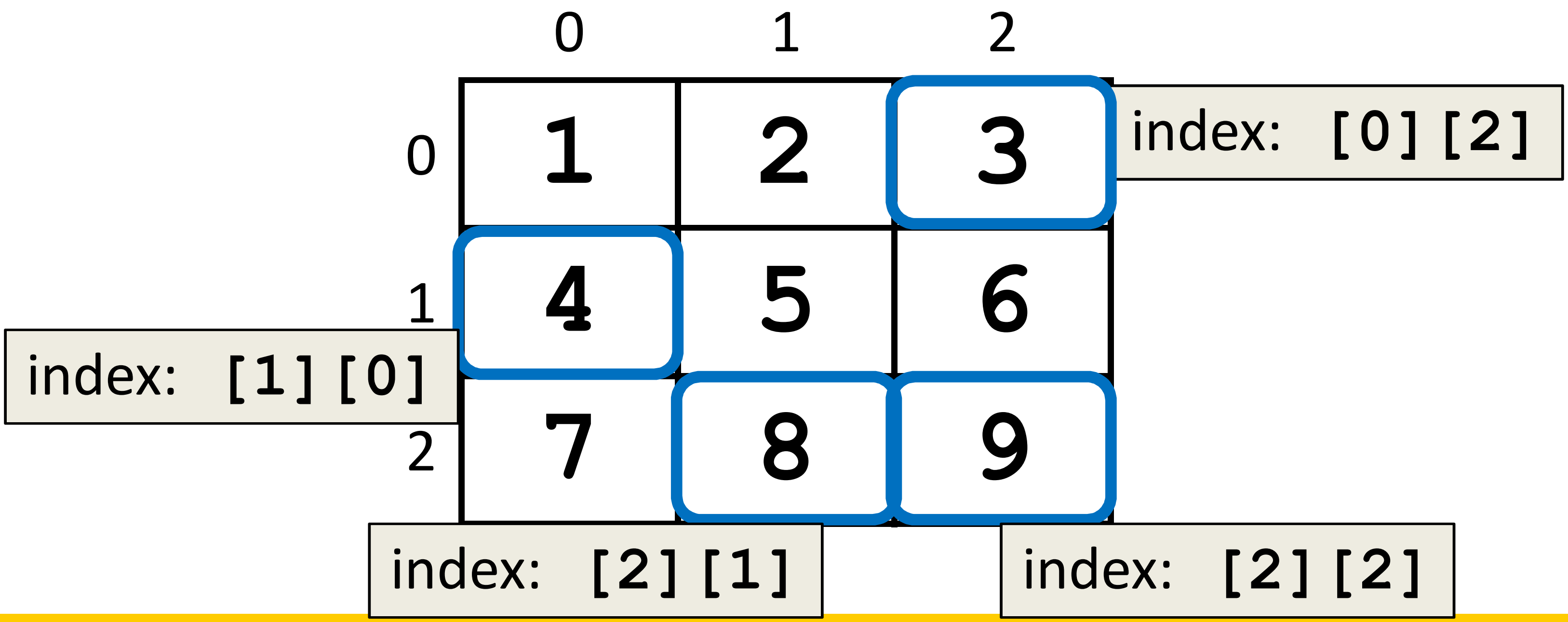
Two-Dimensional Lists: A Grid

- You access an element by the index of its <u>row</u>, and then the <u>column</u>
  - Remember – indexing starts at 0!

	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

Two-Dimensional Lists: A Grid

- You access an element by the index of its row, and then the column
  - Remember – indexing starts at 0!



Lists of Strings

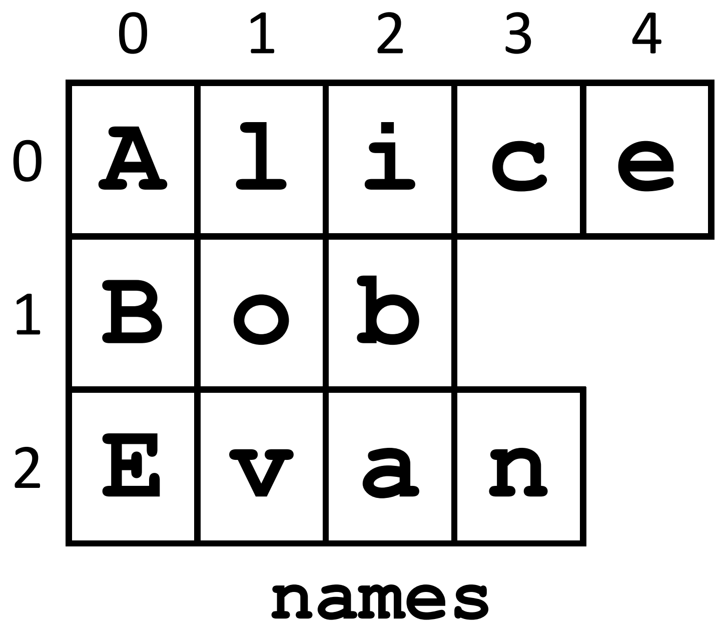
- Remember, a string is <u>like</u> a list of characters
- So what is a list of strings?
  - <u>Like</u> a two-dimensional list!

- We have the index of the string (the row)
- And the index of the character (the column)

UMBC
AN HONORS UNIVERSITY IN MARYLAND

Lists of Strings

- Lists in Python don't have to be rectangular
  - They can be jagged (rows of different lengths)
- Anything we could do with a one-dimensional list, we can do with a two-dimensional list
  - Slicing, index, appending



Vital List Algorithm: 2D Creating

- Write the code to create a 2D list of symbols called **`gameBoard`**, given **`width`** and **`height`**

```
gameBoard = []
while len(gameBoard) < height:
    boardRow = []
    while len(boardRow) < width:
        boardRow.append(".")
    gameBoard.append(boardRow)
```

Vital List Algorithm: 2D Iterating

- Write the code to iterate over and print out the contents of a 2D list called **`gameBoard`**

```
row = 0
while row < len(gameBoard):
    col = 0
    while col < len( gameBoard[row] ):
        print( gameBoard[row][col], end = " ")
        col += 1
    print()  # print a newline at end of each row
    row += 1
```

Mutability

Image Sources

- Tesseract:
  - https://commons.wikimedia.org/wiki/File:Tesseract.gif
- Cardboard box:
  - https://pixabay.com/p-220256/
- Wooden ship (adapted from):
  - https://pixabay.com/p-307603/
- Coconut island (adapted from):
  - https://pixabay.com/p-1892861/