# T11: Matrices and Lists

* Assignment T11 should be completed in a team of two.
* To begin, go to “File” and Select “Make a Copy...”
* Move your copy of the document to the folder for your section.
* Rename the file to **T11: Matrices and Lists - usernames** (replace usernames with your usernames). To do this, click the label in the top left corner of your browser.

## Learning Objectives

* Learn how to work with Matrices in Python
* Learn how memory is allocated for lists, and lists of lists

## Member Roles

* If you have only three people, also combine Recorder & Spokesperson
* If you have five people, add a Process Analyst
* Make up a team name which suits your team!

|  |  |
| --- | --- |
| **Team Roles** | **Member Name** |
| **Facilitator:**  Reads the questions aloud, keeps track of time, and makes sure everyone contributes appropriately. |  |
| **Recorder:**  Records all answers to each question after the team has discussed each question and reached a consensus. |  |
| **Spokesperson:**  Talks on behalf of team to the instructor, TAs and other teams. Compiles and runs programs when applicable. |  |
| **Quality Control Officer:**  Considers how the answers could be deeper, and how the team could work and learn more effectively. |  |
|  |  |
| **Team Name:** |  |

## Matrices

A matrix is a rectangular array of numbers, symbols, or expressions which are arranged in rows and columns. Nested lists are often used to represent matrices by making a list of rows.

For example, the matrix you see to the right might be represented in Python as:

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

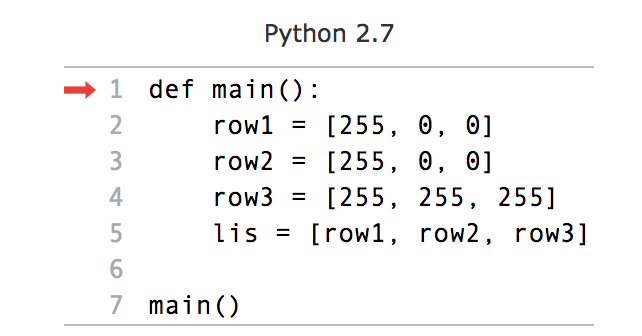
|  |  |
| --- | --- |
| Describe two examples where a matrix would be an appropriate data structure to represent something in a program. | |
| 1. | 2. |

|  |  |
| --- | --- |
| Given the following Python code:  row1 = [255, 0, 0]  row2 = [0, 127, 0]  row3 = [200, 100, 50]  lis = [row1, row2, row3]  What would the following code produce as an output? | |
| print(row1[0]) | 2.a. |
| print(row2[1]) | 2.b. |
| print(lis[2]) | 2.c. |
| print(lis[2][2]) | 2.d. |

### Matrices in Memory

Changing or copying lIsts of lists can be tricky to understand because of what happens in memory. To help you to more fully understand this, we want you to use a very cool free tool called Online Python Tutor which helps to develop understanding of what happens as the computer executes each line of a program's source code. Using this tool, you can better visualize what the computer is doing step-by-step as it executes your program.

Follow [this link](http://pythontutor.com/iframe-embed.html#code=def+main%28%29%3A%0D%0A++++row1+%3D+%5B255,+0,+0%5D%0D%0A++++row2+%3D+%5B255,+0,+0%5D%0D%0A++++row3+%3D+%5B255,+255,+255%5D%0D%0A++++lis+%3D+%5Brow1,+row2,+row3%5D%0D%0A++++%0D%0Amain%28%29&origin=opt-frontend.js&cumulative=false&heapPrimitives=false&textReferences=false&py=2&rawInputLstJSON=%5B%5D&curInstr=0&codeDivWidth=350&codeDivHeight=400), and step through the code. Pay attention to how memory is being allocated.



Note how the code can be stepped through one line at a line, so we can see what it means that the named variable lis is storing aliases to the named variables row1, row2, and row3.

|  |  |
| --- | --- |
| Next consider if instead we had:  row1 = [255, 0, 0]  row2 = [0, 127, 0]  row3 = [200, 100, 50]  lis2 = [[255, 0, 0], [0, 127, 0], [200, 100, 50]] | |
| 1. How would the new definition change the way data is stored in memory? In other words, would Python return True or False if you ran the following command at the end of the main() function:   print(lis2[0] == row1) | 3.a: |
| 1. Would Python return True or False if you ran the following command at the end of the main() function?   print(lis2[0] is row1) | 3.b: |

### Matrices and Images

One common way in which matrices are used is to represent images. Consider the following “image”:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

Each cell of the image, commonly referred to as a pixel, is an element in a matrix. Typically, the numbering for each pixel starts in the top left and works it’s way left-to-right, top-to-bottom. The image below shows every pixel’s 2-dimension index in the matrix given by row, column:

|  |  |  |
| --- | --- | --- |
| 0,0 | 0,1 | 0,2 |
| 1,0 | 1,1 | 1,2 |
| 2,0 | 2,1 | 2,2 |

|  |  |
| --- | --- |
| Assuming this image is stored in a variable called image, how would you access each of the black pixels? | 4.a. |
| Assuming this image is stored in a variable called image, how would you access each of the white pixels? | 4.b. |

Inside of each cell would be the value, or color, of that pixel. For this assignment, we are going to represent a **binary image** (e.g., an image that only has black or white pixels, similar to the ones above). To represent white pixels, we will use a value of 255. To represent black pixels, we will use a value of 0. Tying this all together, to put the color black into the top left cell, we would write:

image[0][0] = 0 # image[row idx][column idx] = color

|  |  |
| --- | --- |
| Given  image = [[255, 0, 0], [255, 0, 0], [255, 255, 255]]  answer the following questions. | |
| What value is stored in the matrix element image[2][1]? | 5.a. |
| What value is stored in the matrix element image[1][2]? | 5.b. |
| By using the cell background color of the table properties (which can be accessed by right clicking the grid), “draw” the binary image for the named variable image. | 5.c.   |  |  |  | | --- | --- | --- | |  |  |  | |  |  |  | |  |  |  | |

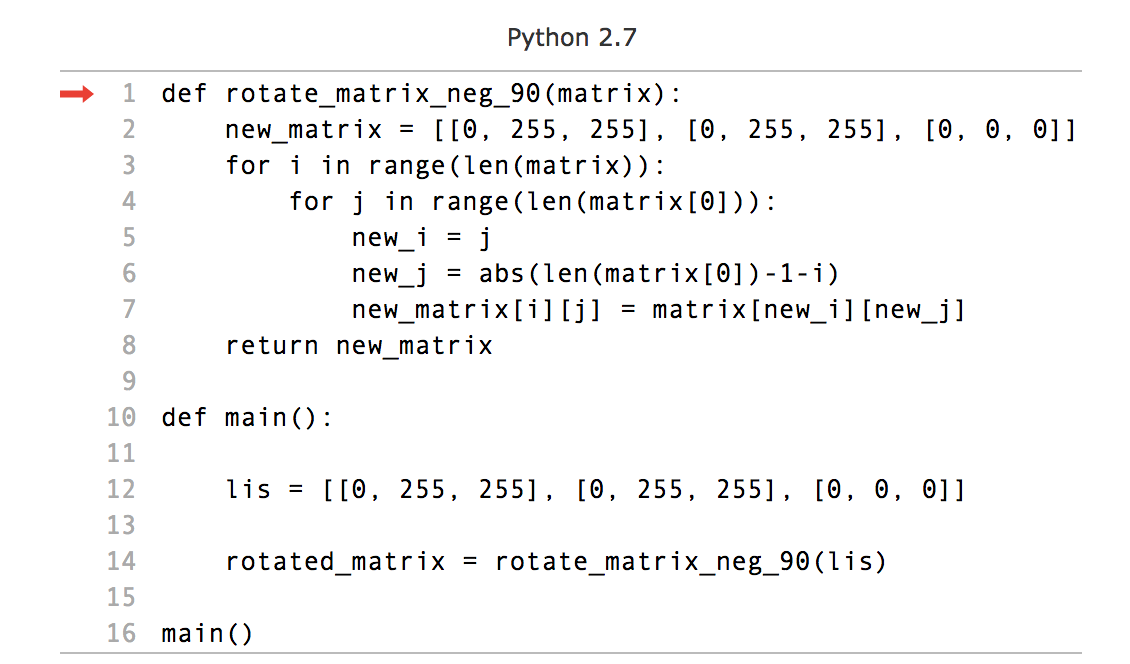
## Manipulating Images

There are many things we might want to do with images like color correction, red eye reduction, and rotation. Does it bother you when your friends post “sideways” images in Facebook?

Rotating the image from Figure 1 and 2 by -90 degrees (counterclockwise) will result in:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

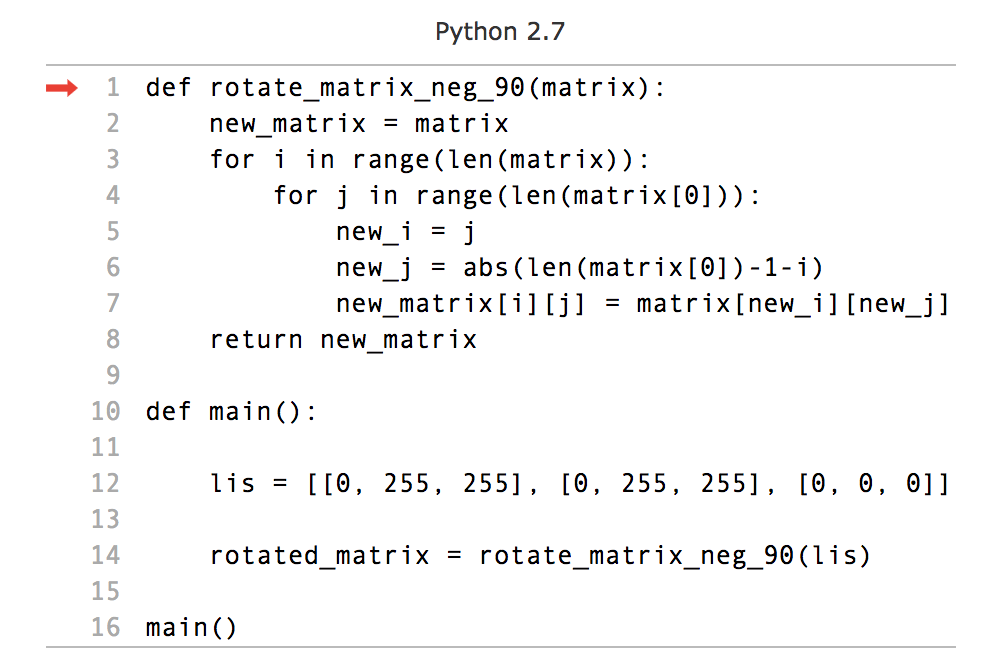
The following Online Python Tutor code example rotates a matrix by -90 degrees. It should be used to answer the next questions. Click on [this link](http://pythontutor.com/iframe-embed.html#code=def%20rotate_matrix_neg_90%28matrix%29%3A%0A%20%20%20%20new_matrix%20%3D%20%5B%5B0,%20255,%20255%5D,%20%5B0,%20255,%20255%5D,%20%5B0,%200,%200%5D%5D%0A%20%20%20%20for%20i%20in%20range%28len%28matrix%29%29%3A%0A%20%20%20%20%20%20%20%20for%20j%20in%20range%28len%28matrix%5B0%5D%29%29%3A%0A%20%20%20%20%20%20%20%20%20%20%20%20new_i%20%3D%20j%0A%20%20%20%20%20%20%20%20%20%20%20%20new_j%20%3D%20abs%28len%28matrix%5B0%5D%29-1-i%29%0A%20%20%20%20%20%20%20%20%20%20%20%20new_matrix%5Bi%5D%5Bj%5D%20%3D%20matrix%5Bnew_i%5D%5Bnew_j%5D%0A%20%20%20%20return%20new_matrix%0A%0Adef%20main%28%29%3A%0A%0A%20%20%20%20lis%20%3D%20%5B%5B0,%20255,%20255%5D,%20%5B0,%20255,%20255%5D,%20%5B0,%200,%200%5D%5D%0A%20%20%20%20%0A%20%20%20%20rotated_matrix%20%3D%20rotate_matrix_neg_90%28lis%29%0A%20%20%20%20%0Amain%28%29&codeDivHeight=400&codeDivWidth=350&cumulative=false&curInstr=0&heapPrimitives=false&origin=opt-frontend.js&py=2&rawInputLstJSON=%5B%5D&textReferences=false) to get to the active code.



Pay particular attention to how the code was written, especially with regard to creating the new matrix (line 2). Was there a surprising choice made there?

|  |  |
| --- | --- |
| Compare the variables lis and new\_matrix.  Are they the same object? (i.e., would:  print(matrix is new\_matrix)  print the value True)? | 6.a. |

Consider the following alternative implementation of this algorithm by following [this link](http://pythontutor.com/iframe-embed.html#code=def%20rotate_matrix_neg_90%28matrix%29%3A%0A%20%20%20%20new_matrix%20%3D%20matrix%0A%20%20%20%20for%20i%20in%20range%28len%28matrix%29%29%3A%0A%20%20%20%20%20%20%20%20for%20j%20in%20range%28len%28matrix%5B0%5D%29%29%3A%0A%20%20%20%20%20%20%20%20%20%20%20%20new_i%20%3D%20j%0A%20%20%20%20%20%20%20%20%20%20%20%20new_j%20%3D%20abs%28len%28matrix%5B0%5D%29-1-i%29%0A%20%20%20%20%20%20%20%20%20%20%20%20new_matrix%5Bi%5D%5Bj%5D%20%3D%20matrix%5Bnew_i%5D%5Bnew_j%5D%0A%20%20%20%20return%20new_matrix%0A%0Adef%20main%28%29%3A%0A%0A%20%20%20%20lis%20%3D%20%5B%5B0,%20255,%20255%5D,%20%5B0,%20255,%20255%5D,%20%5B0,%200,%200%5D%5D%0A%20%20%20%20%0A%20%20%20%20rotated_matrix%20%3D%20rotate_matrix_neg_90%28lis%29%0A%20%20%20%20%0Amain%28%29&codeDivHeight=400&codeDivWidth=350&cumulative=false&curInstr=0&heapPrimitives=false&origin=opt-frontend.js&py=2&rawInputLstJSON=%5B%5D&textReferences=false):



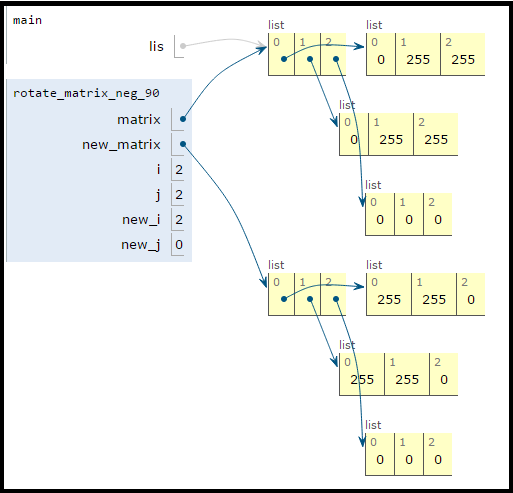
This change, while it looks correct, does something very wrong to the code. Run it and figure out where the code now breaks, and why.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| By using the cell background color of the table properties (which can be accessed by right clicking the grid), “draw” the binary image of what you think was the intended output of the code. | 7.a.   |  |  |  | | --- | --- | --- | |  |  |  | |  |  |  | |  |  |  | |
| By using the cell background color of the table properties (which can be accessed by right clicking the grid), “draw” the binary image of the actual output of the code. | 7.b.   |  |  |  | | --- | --- | --- | |  |  |  | |  |  |  | |  |  |  | |

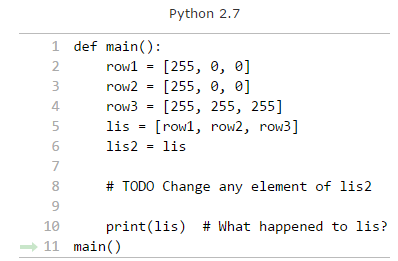
## 

Describe in your own words what caused this unintended outcome. Use the figure below, which represents the code **before** the change was made, to explain the problem.

|  |
| --- |
| 8. |



The root of the problem may not be immediately apparent, so let’s go back to the very [first example](https://goo.gl/06odZJ):



|  |  |
| --- | --- |
| Before running the code or making the change, what do you expect the output of line 10 to be? | 9.a. |
| Make the change noted in the comment, and run the altered code. Now what is the output of line 10? | 9.b. |
| Does you think that your original answer to Question 8 still fully explains the problem? If not, more fully explain the problem based on what you learned from the above code. | 9.c. |

|  |  |
| --- | --- |
| Imagine that we decided to store our matrix in a tuple of tuples instead of a list of lists:  tup = ((255, 0, 0), (255, 0, 0), (255, 255, 255)) | |
| Which problems would be solved? | 10.a. |
| Which new problems would likely be created? | 10.b. |

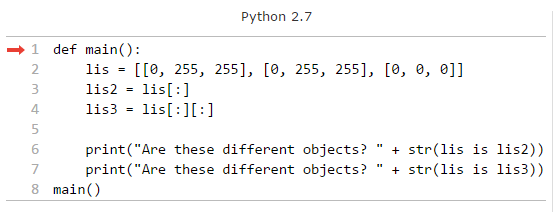
## 

Cloning a 1-dimensional list is straightforward:

lis1 = [1, 2, 3]

lis2 = lis1[:]

Cloning a 2-dimensional list, however, is not straightforward. Try the following code by clicking the following [link](http://pythontutor.com/iframe-embed.html#code=def%20main%28%29%3A%0A%20%20%20%20lis%20%3D%20%5B%5B0,%20255,%20255%5D,%20%5B0,%20255,%20255%5D,%20%5B0,%200,%200%5D%5D%0A%20%20%20%20lis2%20%3D%20lis%5B%3A%5D%0A%20%20%20%20lis3%20%3D%20lis%5B%3A%5D%5B%3A%5D%0A%20%20%20%20%0A%20%20%20%20print%28%22Are%20these%20different%20objects%3F%20%22%20%2B%20str%28lis%20is%20lis2%29%29%0A%20%20%20%20print%28%22Are%20these%20different%20objects%3F%20%22%20%2B%20str%28lis%20is%20lis3%29%29%0Amain%28%29&codeDivHeight=400&codeDivWidth=350&cumulative=false&curInstr=0&heapPrimitives=false&origin=opt-frontend.js&py=2&rawInputLstJSON=%5B%5D&textReferences=false):



Brainstorm ideas with your group on how to clone a matrix while avoiding the aliasing problem. You do not need to code the solution; a high-level, plain English response is sufficient (though code works too!)

|  |
| --- |
| 11. |

## Submission Instructions

1. Make sure everyone’s name is at the top of the document.
2. Make sure your file is saved in the correct Google Drive folder for your section of the course.
3. You do not need to submit anything to Moodle.