15-112 Lecture 2

Animations with Lists

Instructor: Pat Virtue

Tuesday Logistics

Announcements

HW7

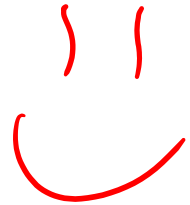
- Tetris! Plan ahead

CMU Graphics Installation

- Keep an eye on Piazza updates if you run into issues
- Come find Pat if you bump into issues that pinned Piazza update doesn’t address
- Don’t stress. You can do ALL the rest of the homework without this installation

Thursday Logistics

Announcements



HW7

- Tetris!

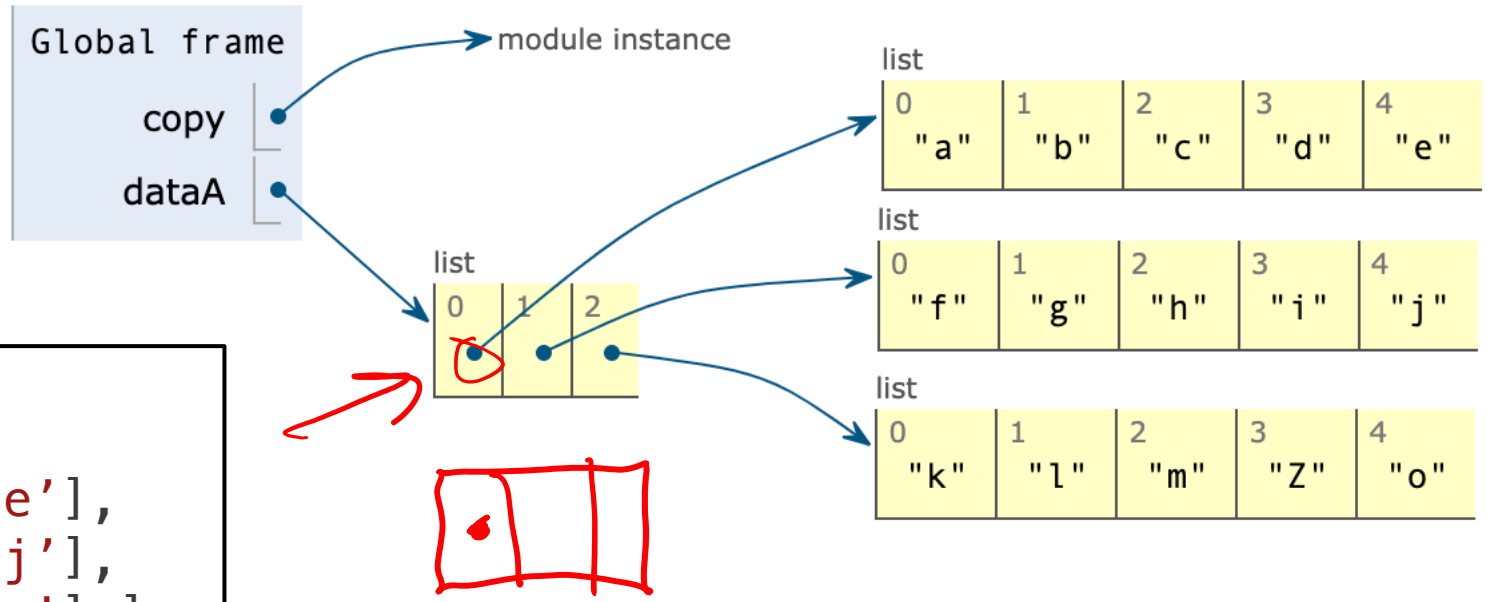
CMU Graphics Installation

- Keep an eye on Piazza updates if you run into issues
- Come find Pat if you bump into issues that pinned Piazza update doesn’t address
- Don’t stress. You can do ALL the rest of the homework without this installation

2D Lists

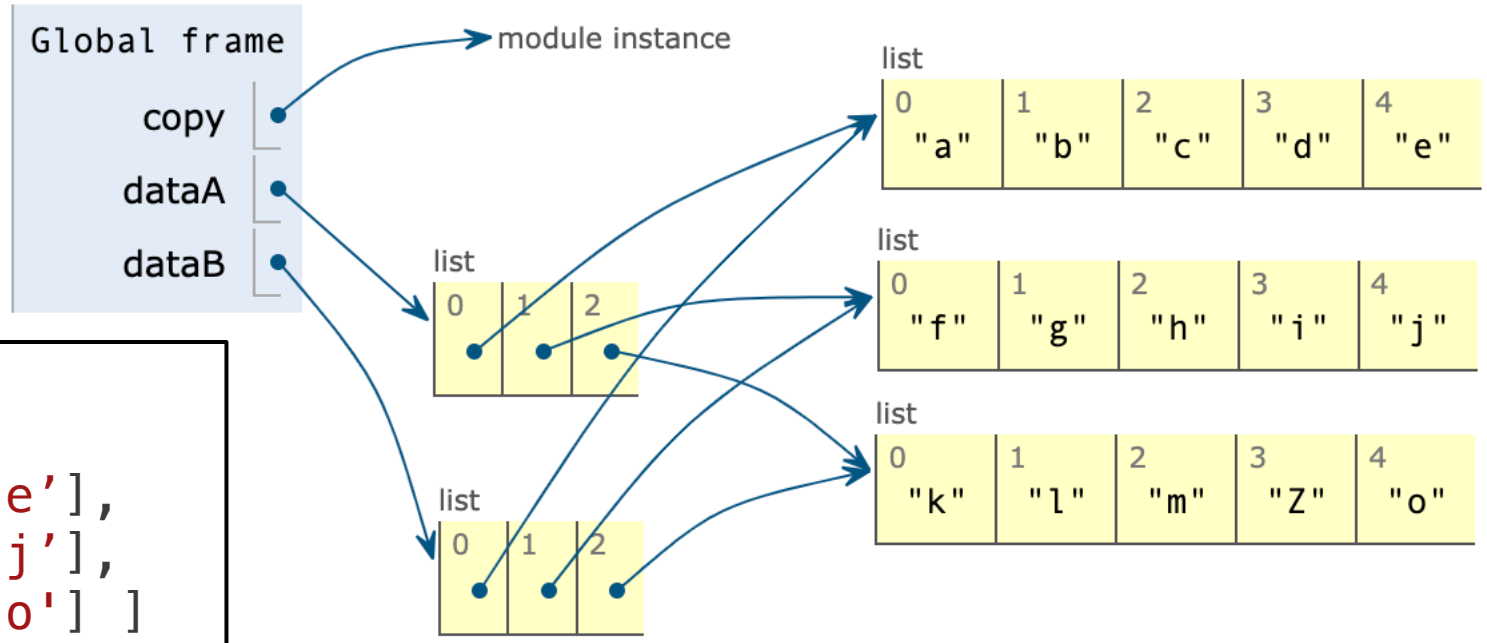
Copy vs Deepcopy

```
import copy

dataA = [ ['a','b','c','d','e'],
          ['f','g','h','i','j'],
          ['k','l','m','n','o'] ]
dataB = copy.copy(dataA)

dataB[2][3] = 'Z'
```



Copy vs Deepcopy

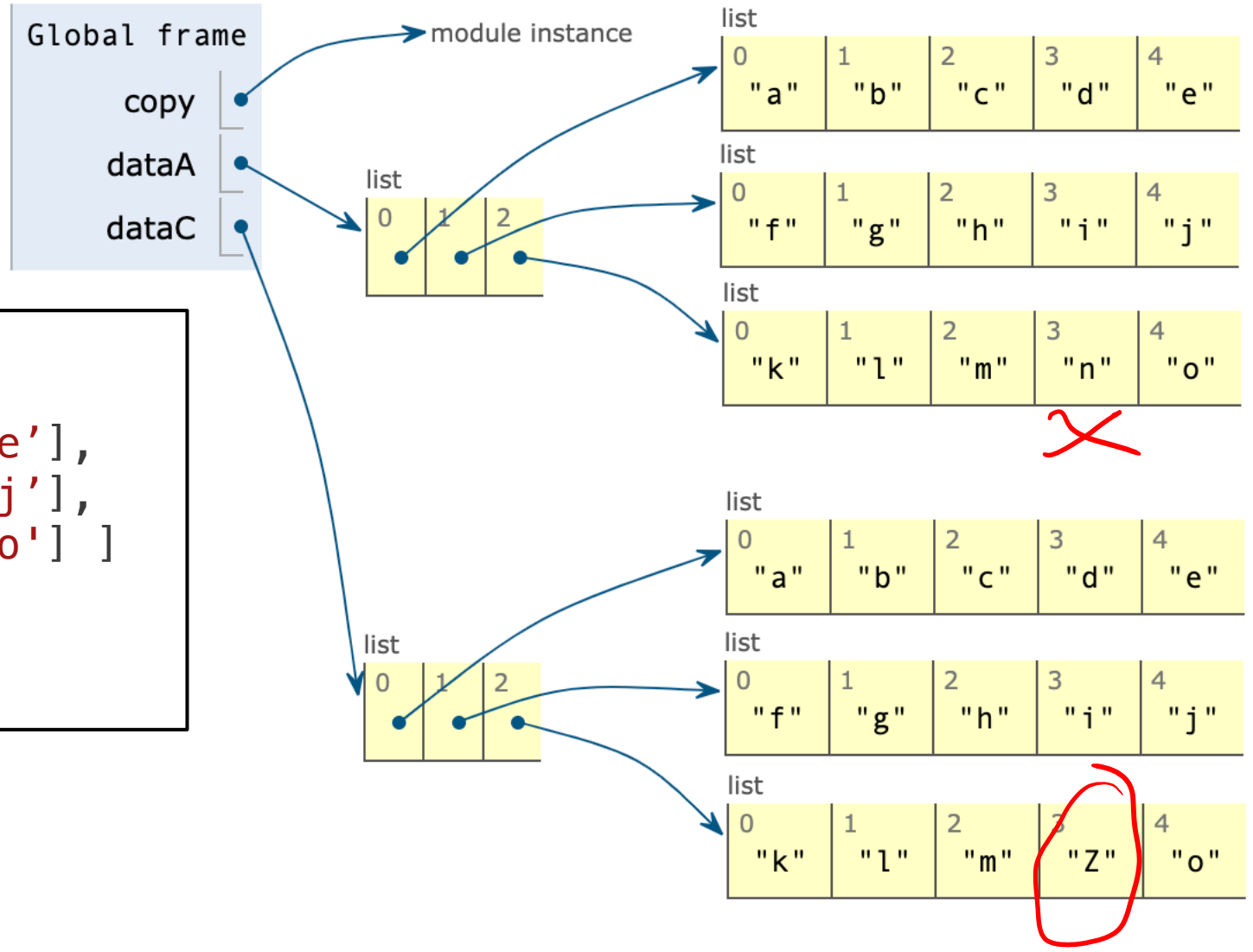
```
import copy

dataA = [ ['a','b','c','d','e'],
          ['f','g','h','i','j'],
          ['k','l','m','n','o'] ]
dataB = copy.copy(dataA)

dataB[2][3] = 'Z'
```



Deep

Copy vs Deepcopy

```
import copy

dataA = [ ['a','b','c','d','e'],
          ['f','g','h','i','j'],
          ['k','l','m','n','o'] ]
dataC = copy.deepcopy(dataA)

dataC[2][3] = 'Z'
```

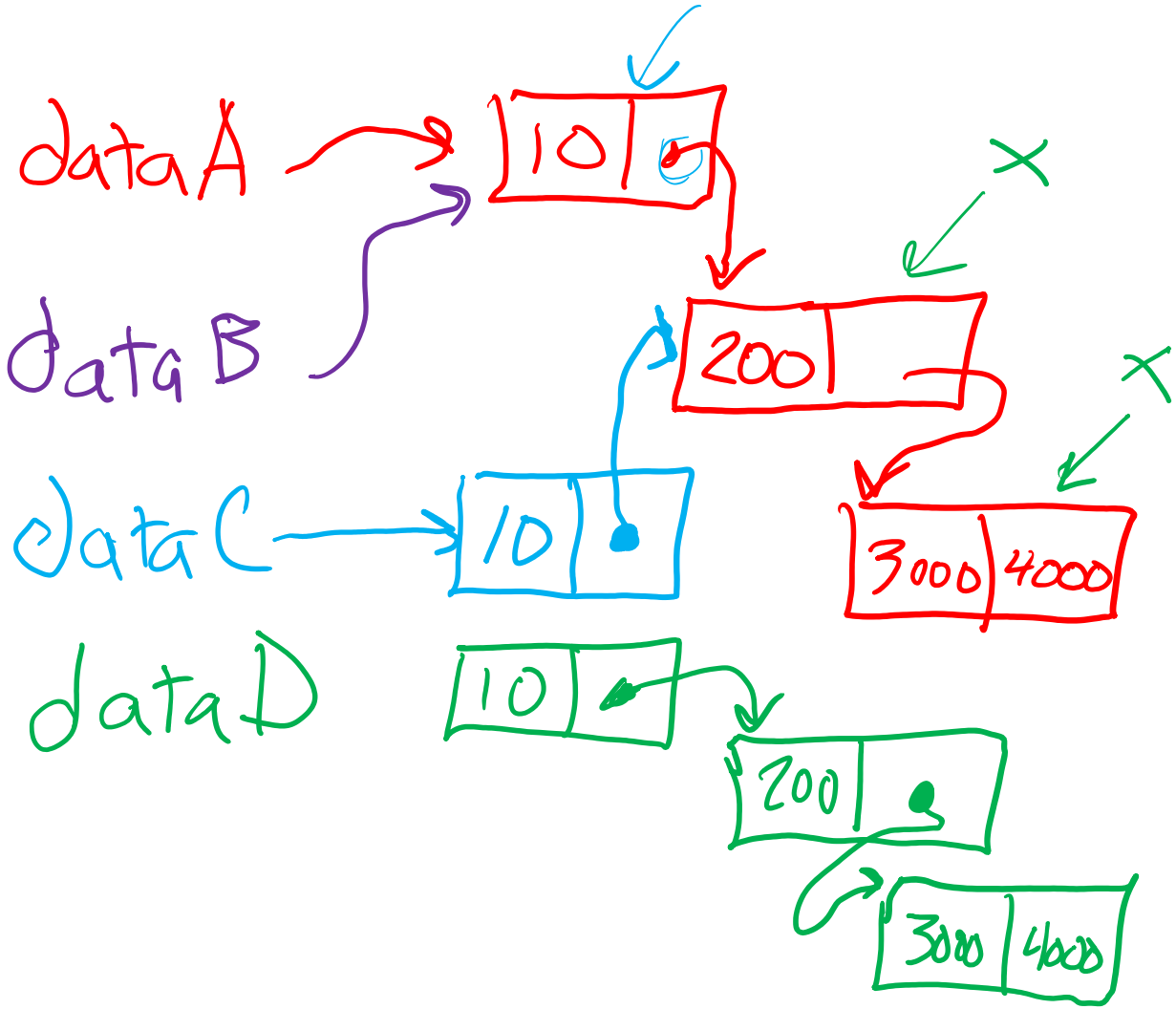


Poll 1

How many total list objects exist after running this code?

```
import copy

dataA = [10, [200, [3000, 4000] ] ]
dataC = copy.copy(dataA)

dataD = copy.deepcopy(dataA)
```

dataB = dataA



A. 3

B. 4

C. 5

D. 6

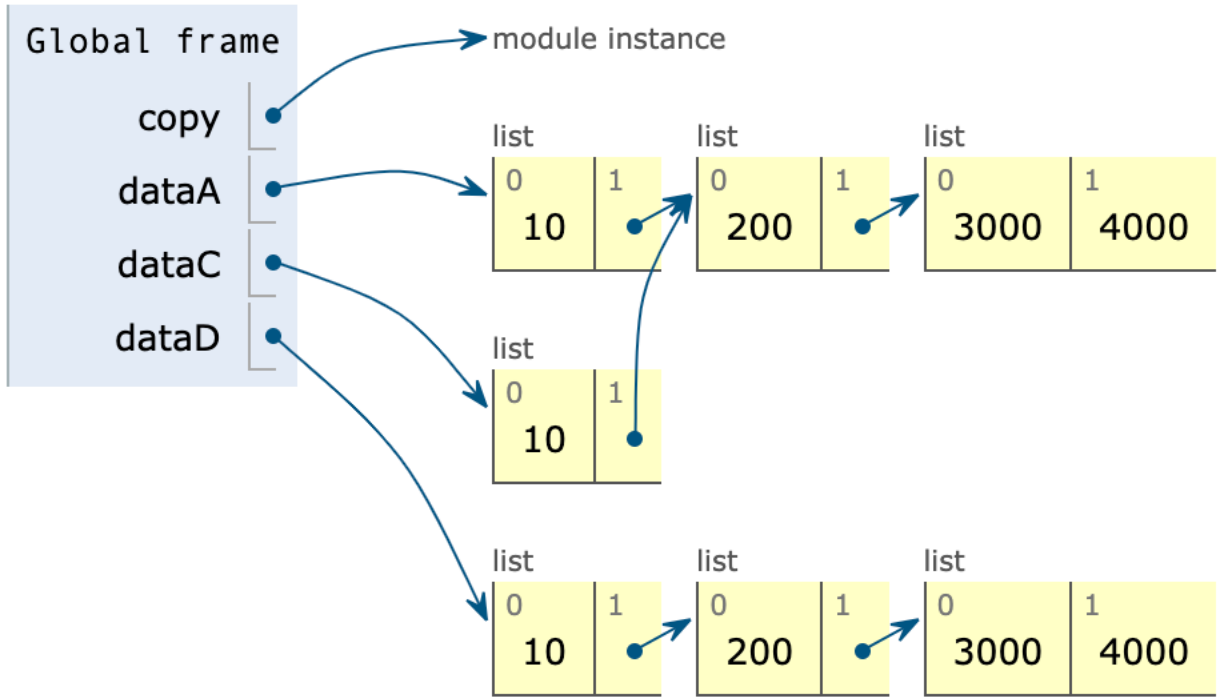
E. 7

F. 9

Poll 1

How many total list objects exist after running this code?

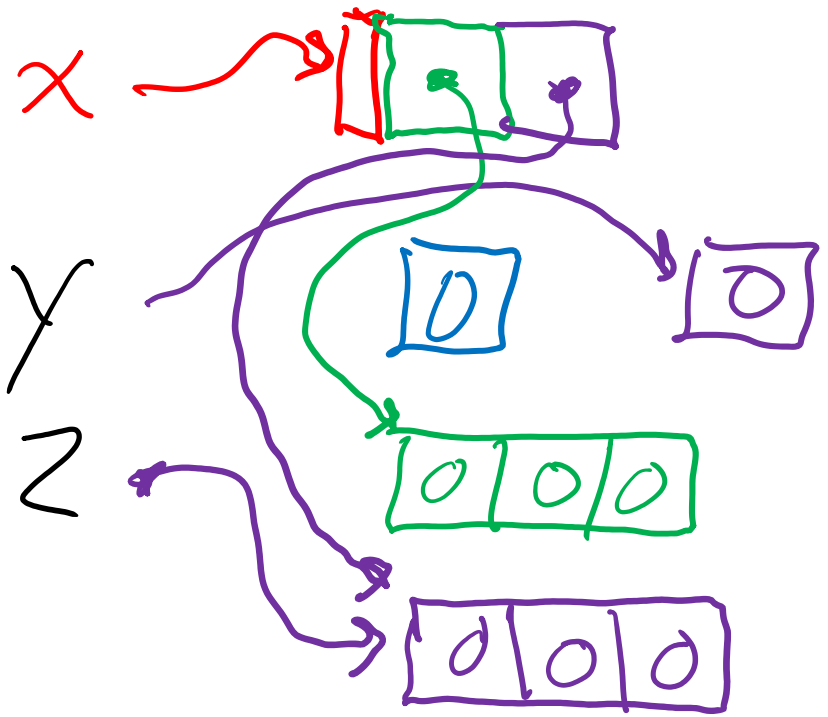
```
import copy

dataA = [10, [200, [3000, 4000] ] ]
dataC = copy.copy(dataA)

dataD = copy.deepcopy(dataA)
```



Code tracing with lists

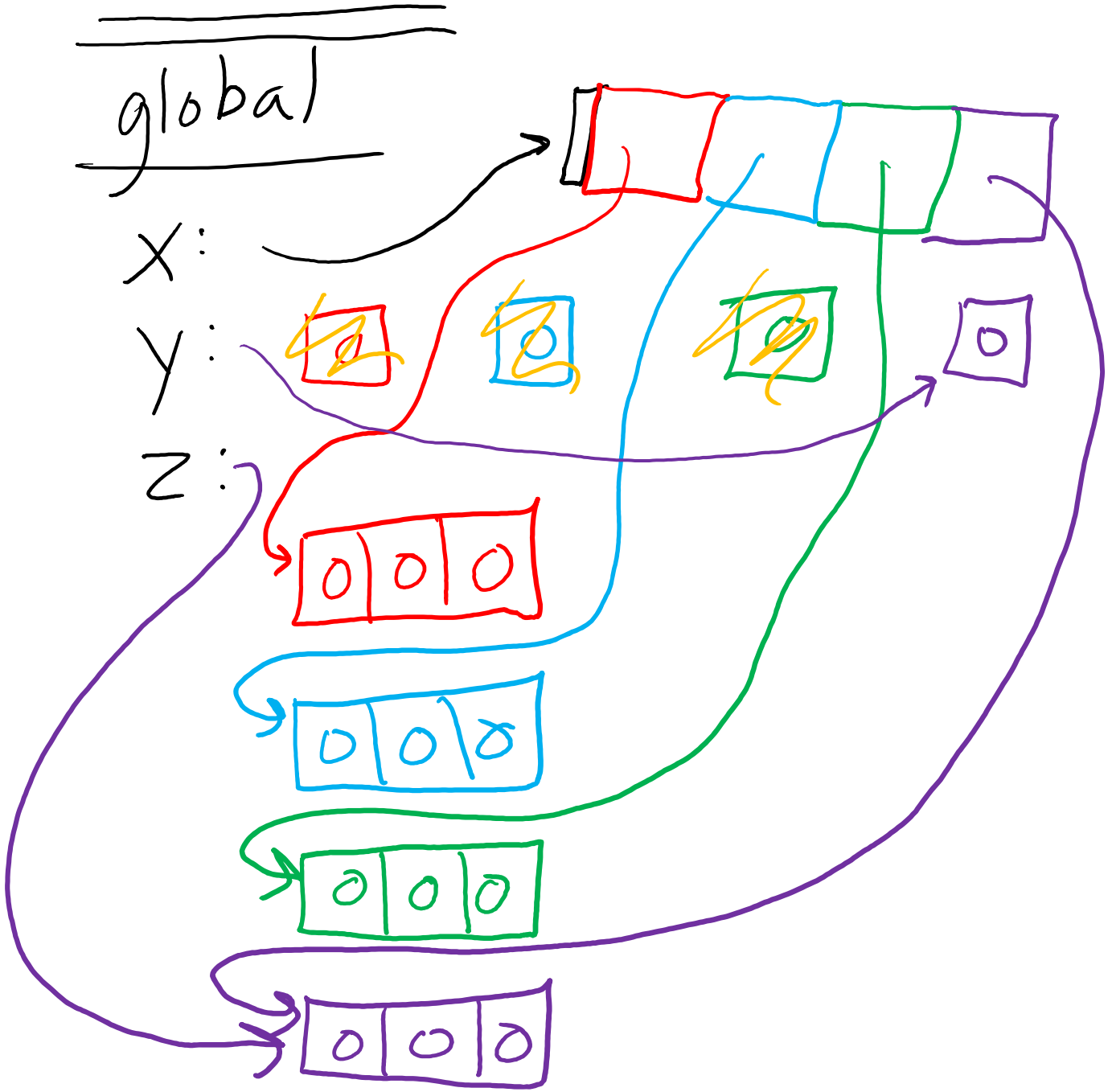
```
x = []
for i in range(4):
    y = [0]
    z = y * 3
    x.append(z)
```

one



Code tracing with lists

```
x = []
for i in range(4):
    y = [0]
    z = y * 3
    x.append(z)
```

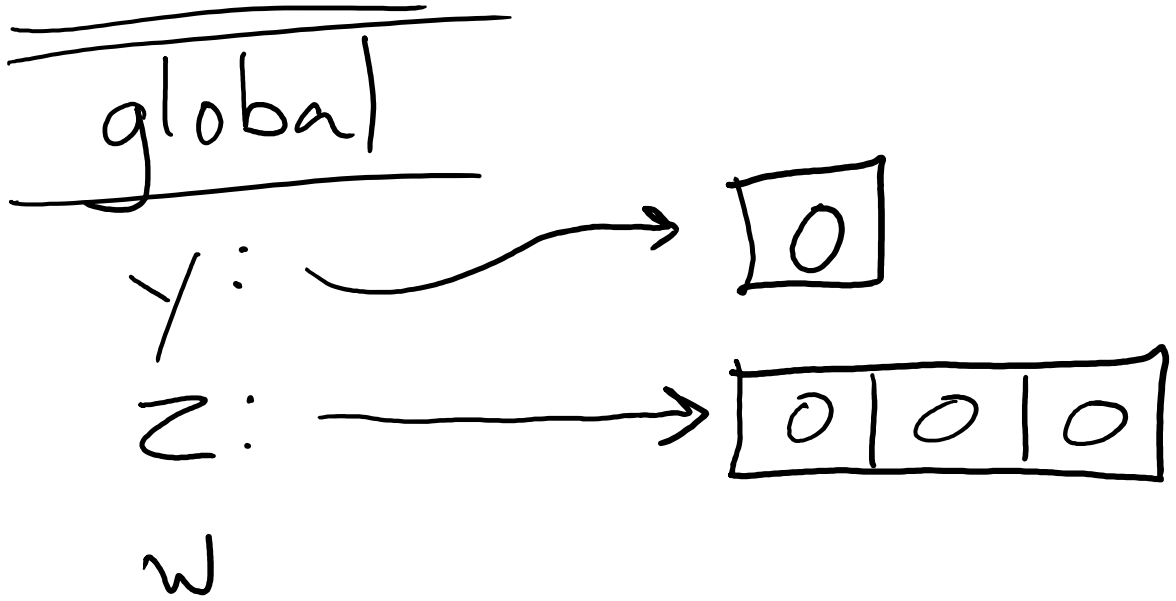


Code tracing with lists

```
x = [[0]*3] * 4
```

Step by step:

```
y = [0]
z = y * 3
w = [z]
x = w * 4
```

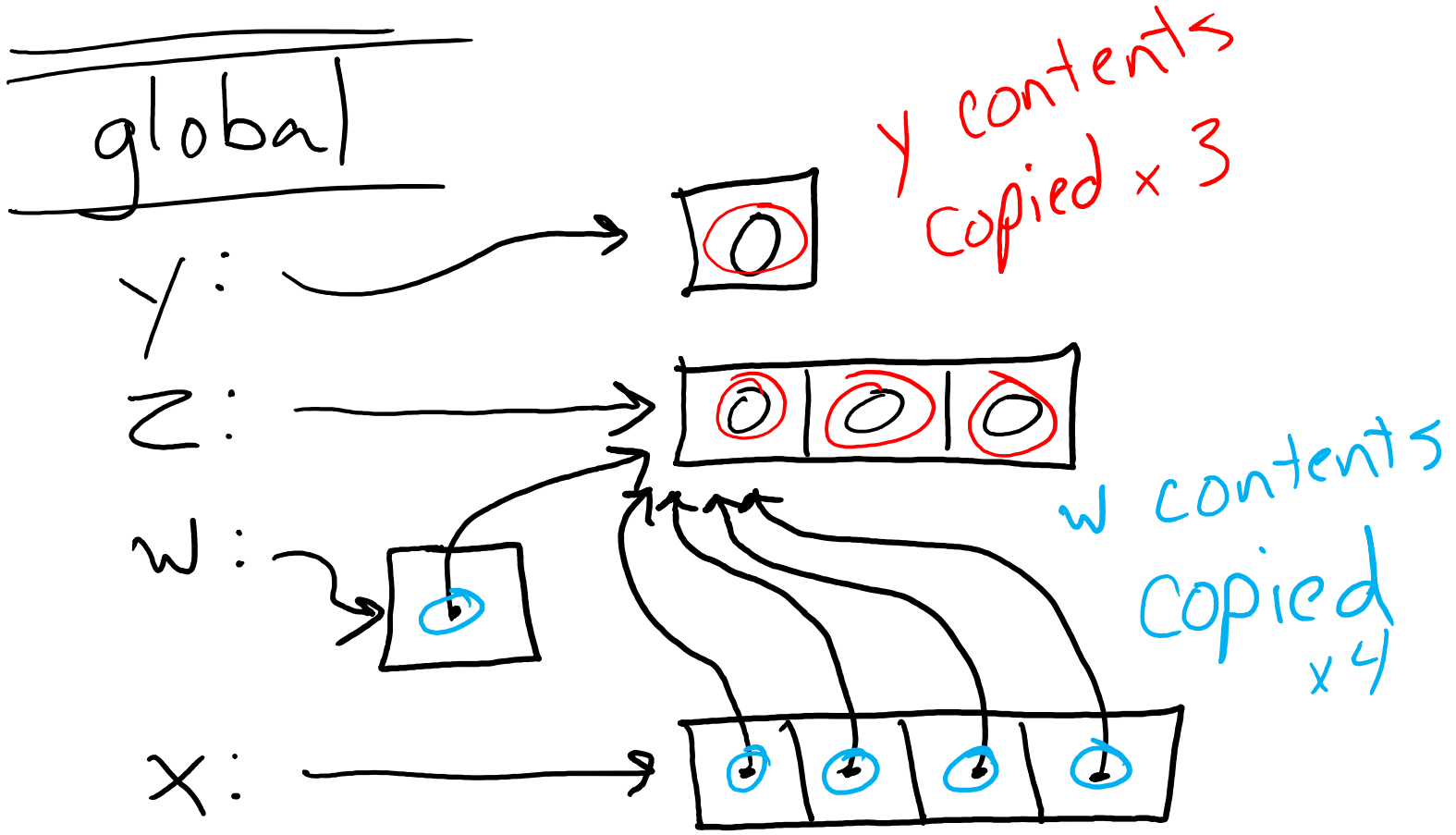


Code tracing with lists

```
x = [[0]*3] * 4
```

Step by step:

```
y = [0]
z = y * 3
w = [z]
x = w * 4
```



2D lists

Things to keep in mind

Watch out for aliasing!

Think before copying

- Shallow copy
- Deep copy

Helper functions for printing 

2D Board Graphics

112 Graphics Grid Worksheet

Given:

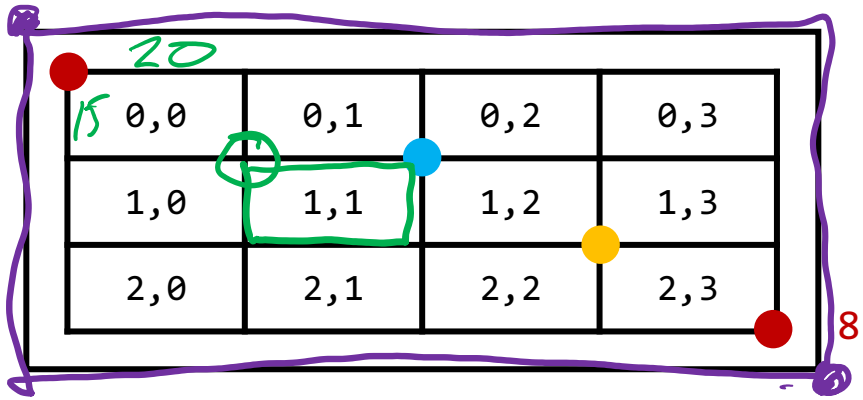
```
app.width: 90
app.height: 55
app.margin: 5
```

boardWidth = w - margin*2

boardHeight = h - margin*2

90 - 5*2 = 80

55 - 5*2 = 45



Fill in the following grid with the value return for each call to:

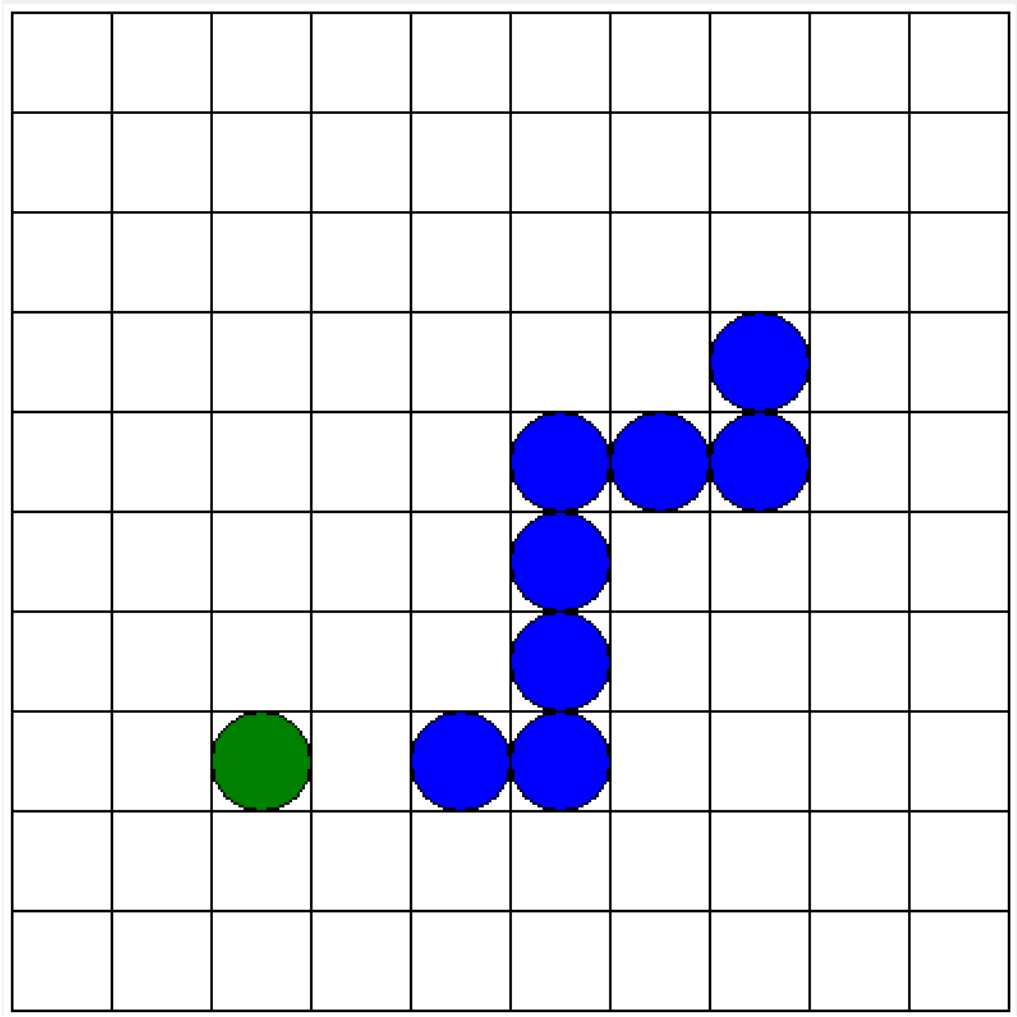
```
x0, y0 = getCellLeftTop(app, r, c)
```

getCellLeftTop(app,0,0)	getCellLeftTop(app,0,1)	getCellLeftTop(app,0,2)	getCellLeftTop(app,0,3)
5, 5	___, ___	___, ___	
getCellLeftTop(app,1,0)	**getCellLeftTop(app,1,1)**	**getCellLeftTop(app,1,2)**	**getCellLeftTop(app,1,3)**
	5+20, 5+15	___, ___	
getCellLeftTop(app,2,0)	**getCellLeftTop(app,2,1)**	**getCellLeftTop(app,2,2)**	**getCellLeftTop(app,2,3)**
			___, ___

Building an Application!

Snake Example

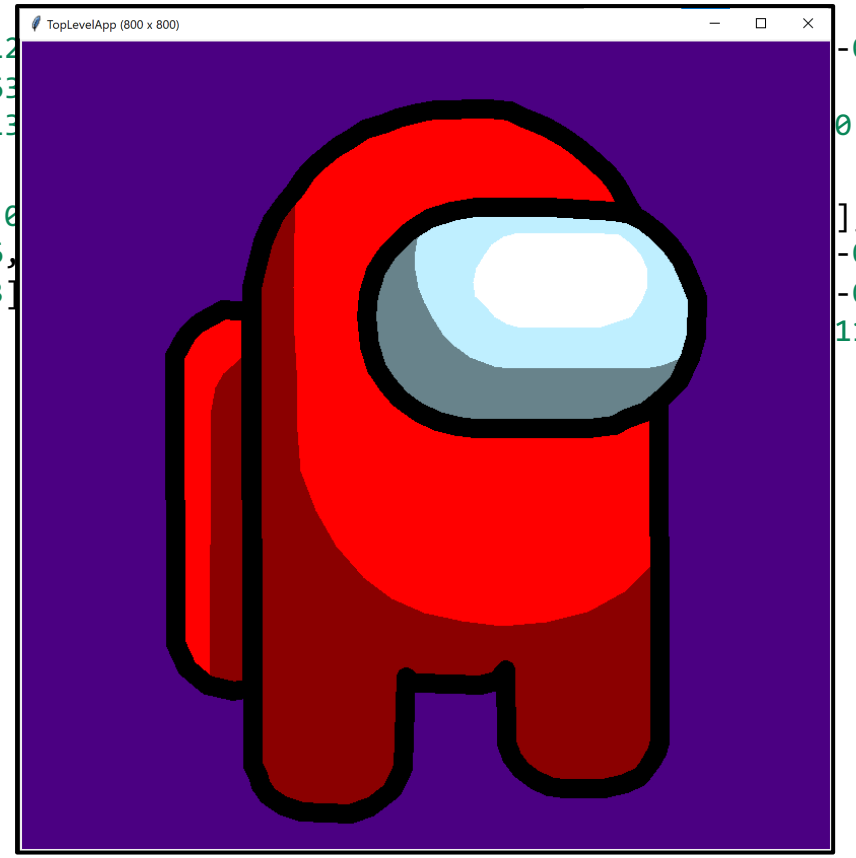
See Guided Exercise in notes



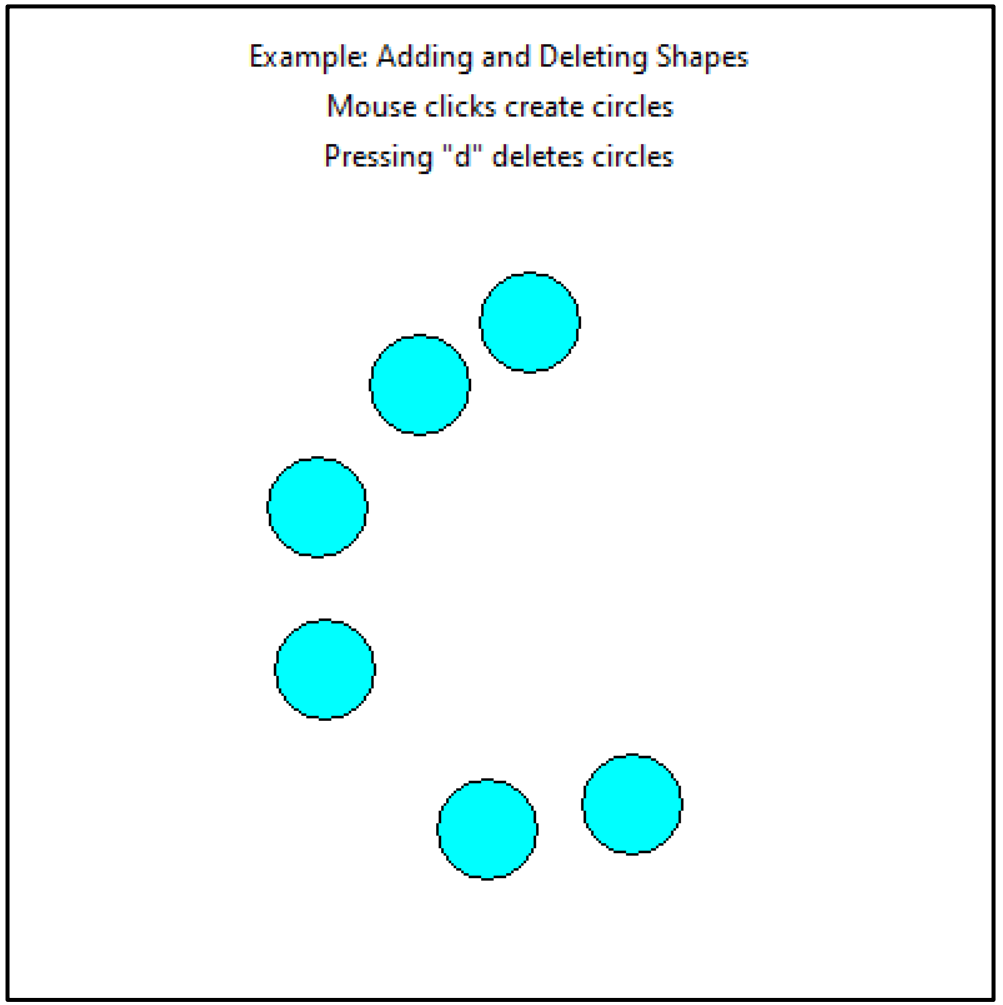
Polygons

Polygon Creator Example

```
def amongUsShape(usColor='red', usColor2='darkRed', usLineWidth=5):
    polys = [
        # Pack
        (usColor, None, 0, [[-0.48, -0.396], [-0.553, -0.4], [-0.616, -0.366], [-0.653, -0.33], [-0.683, -0.276], [-0.68, 0.49], [-0.65,
        (usColor2, None, 0, [[-0.476, -0.28], [-0.506, -0.27], [-0.553, -0.23], [-0.573, -0.193], [-0.586, -0.133], [-0.59, 0.586], [-0.
        (None, 'black', usLineWidth, [[-0.48, -0.396], [-0.553, -0.4], [-0.616, -0.366], [-0.653, -0.33], [-0.683, -0.276], [-0.68, 0.49
        # Body
        (usColor2, None, 0, [[0.01, -0.933], [-0.033, -0.926], [-0.086, -0.913], [-0.12
        (usColor, None, 0, [[-0.36, -0.686], [-0.363, -0.603], [-0.366, -0.46], [-0.363
        (None, 'black', usLineWidth, [[0.01, -0.933], [-0.033, -0.926], [-0.086, -0.913
        # Eyes
        ('steelBlue', None, 0, [[0.486, -0.663], [0.316, -0.673], [0.123, -0.673], [0.0
        ('lightCyan', None, 0, [[0.15, -0.67], [0.096, -0.666], [0.056, -0.66], [0.016,
        (None, 'black', usLineWidth, [[0.486, -0.663], [0.316, -0.673], [0.123, -0.673]
        ('white', None, 0, [[0.263, -0.606], [0.226, -0.603], [0.196, -0.596], [0.163,
    ]
    return polys
```



Polygon Creator Example

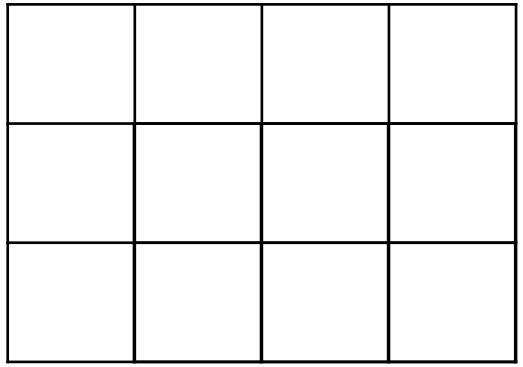


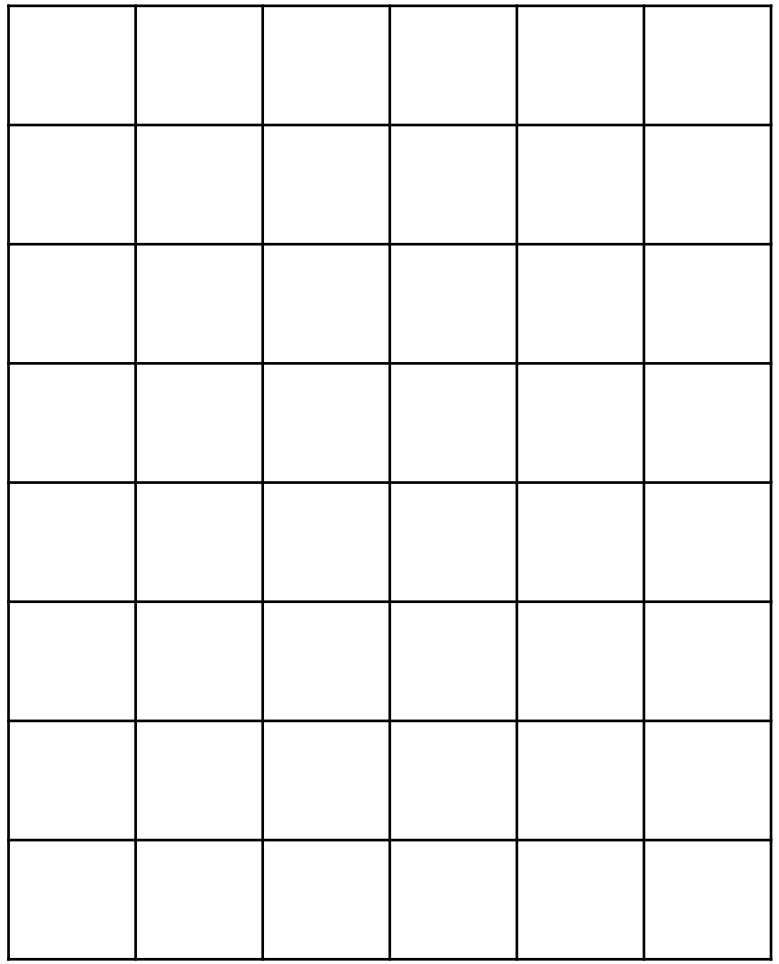
Tetris Hints

Draw Piece

app.pieceLeftCol

app.pieceTopRow





Draw Piece

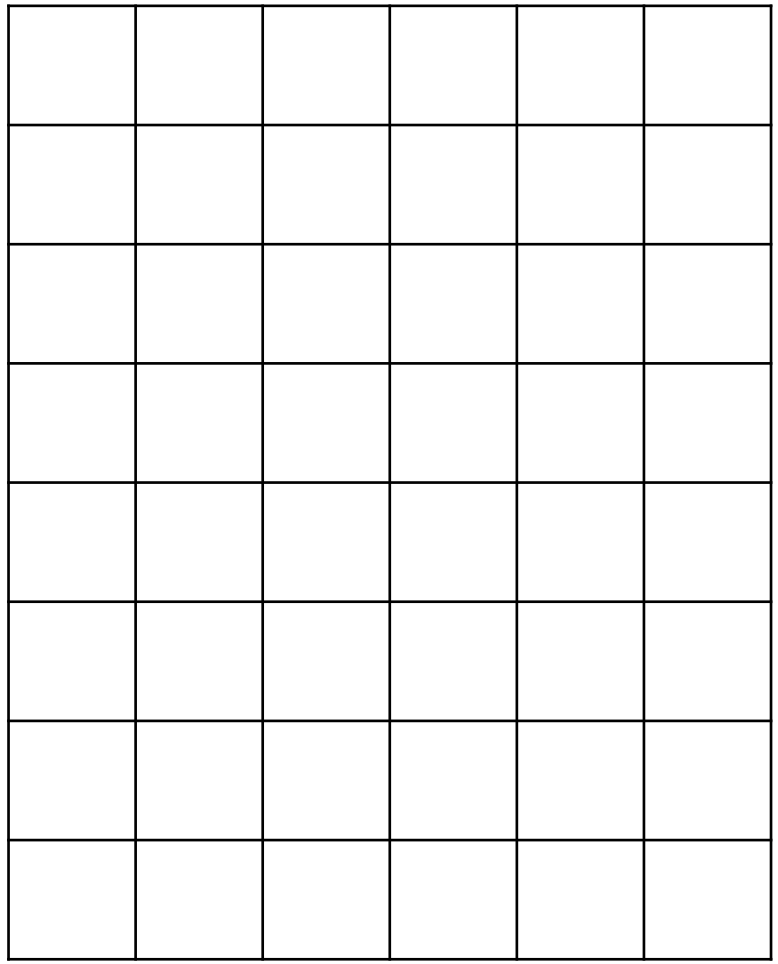
Example

`app.pieceLeftCol`

`app.pieceTopRow`

`app.piece`

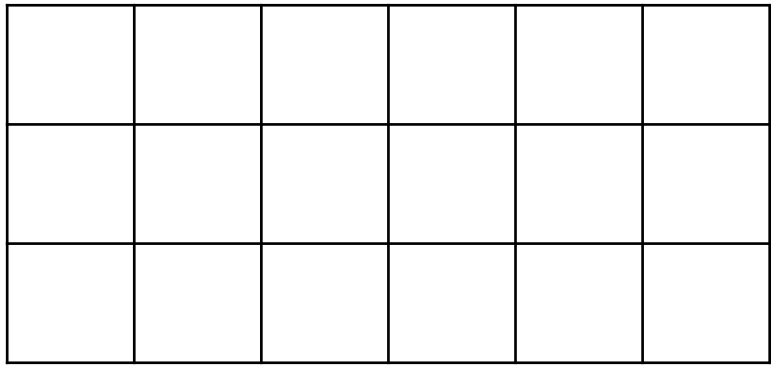
	0	1	2
0	F	T	F
1	T	T	T

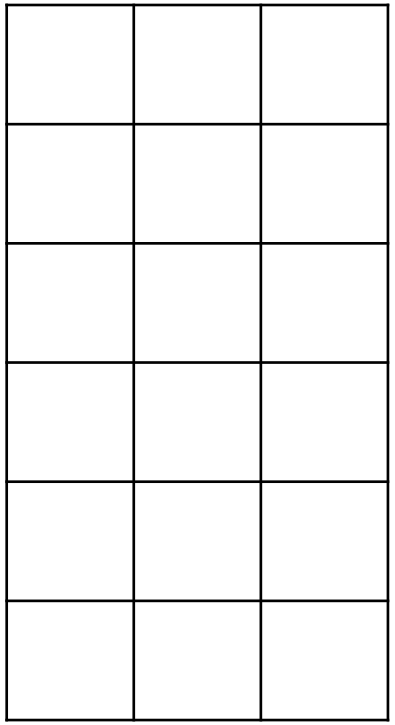


Rotate 2D List

app.pieceLeftCol

app.pieceTopRow





Term Project!