

2D Lists

adapted from material by Mike Scott and Bill
Young at the University of Texas at Austin

Lists of Lists

In Python, list elements can be *anything*.

This includes other lists!

```
1 list1 = [[1, 4, 7], [2, 5, 8], [3, 6, 9]]
```

We can think of this as rows and columns
of a matrix:

`list1[i][j]`

	$j = 0$	$j = 1$	$j = 2$
$i = 0$	1	4	7
$i = 1$	2	5	8
$i = 2$	3	6	9

List Comprehension

```
1 table2 = [[0] * 12] * 10
2
3 flips = ['H' if random.random() < 0.5 else 'T'
4           for x in range(12)] for x in range(10)]
```

Applications

In science and engineering, multiple-dimensional lists
are *everything*.

Want to simulate water flowing? 3D list. Want to do
finite element modeling? 2D list. Want to simulate a
bridge span? 2D list with some extra mathematics.

You'll see other examples in homework.

Game of Life

- An automaton designed by mathematician John Conway
- not really a game
- simulation taking place on a 2D grid
- each element of the grid is occupied or not

A cell's neighbors are the 8 cells surrounding it (unless it is in a corner or edge, in which case it has fewer neighbors).

The simulation proceeds in generations.

If a cell is occupied:

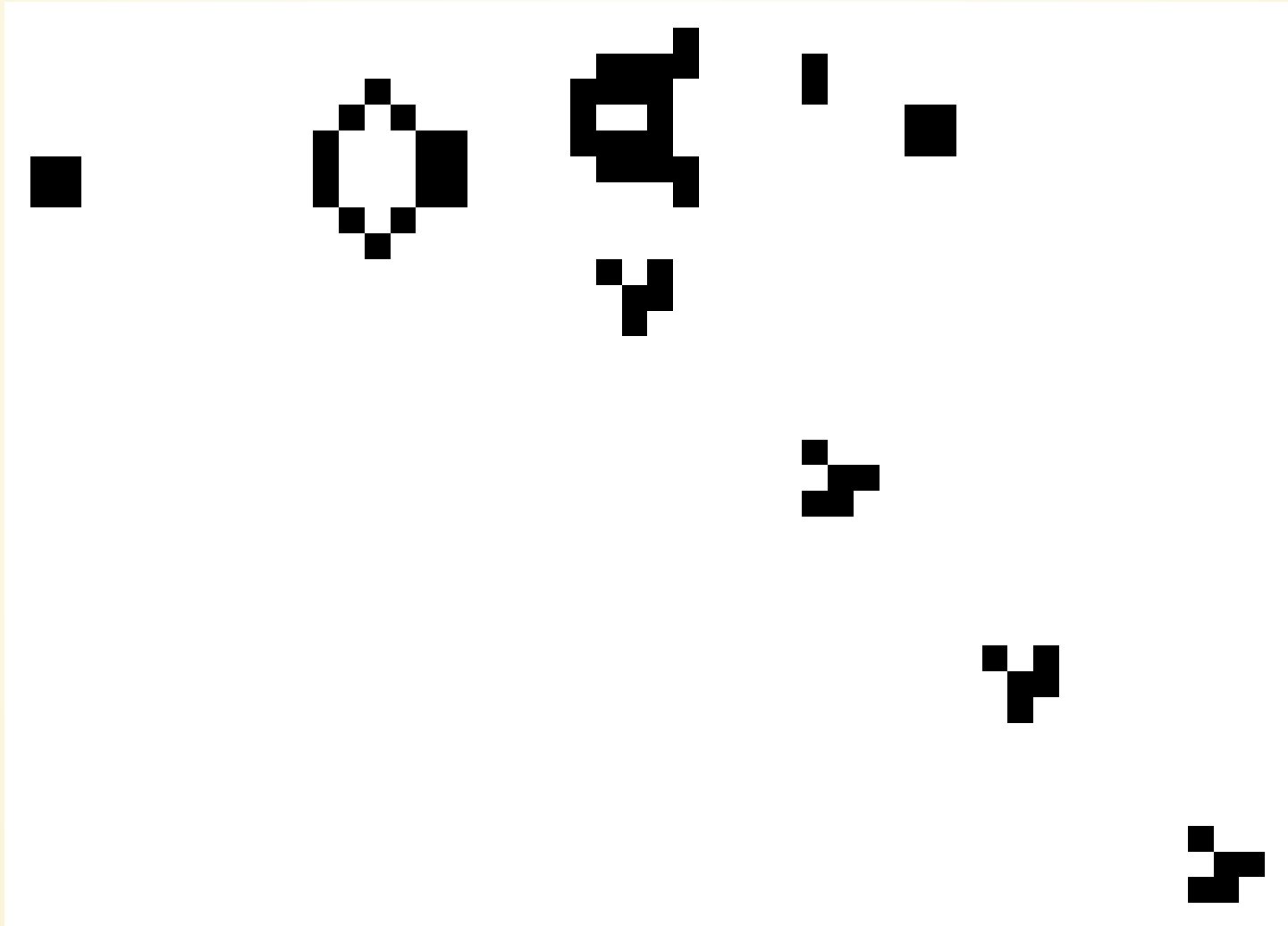
- It survives if it has 2 or 3 neighbors
- It dies if it has 0 or 1 neighbors
- It dies if it has 4 or more neighbors

If a cell is unoccupied:

- it becomes occupied if there are exactly three neighboring cells.

Simple, right?

Simple rules can often lead to complex behavior!



Design and implement a python program to play Conway's game of life

Starting state should be read from file.

Let's talk about the other components!

