

## Python 07 Challenge!

Conway's Game of Life is a cellular automaton that creates beautiful patterns. Given an  $N \times N$  grid of cells, each cell is either dead (0) or alive (1). Every cell interacts with its eight neighbours. At each step in time, the following transitions occur:

- ▶ Any live cell with fewer than two neighbours dies (underpopulation).
- ▶ Any live cell with two or three neighbours lives on (survival).
- ▶ Any live cell with more than three neighbours dies (overcrowding).
- ▶ Any dead cell with exactly three live neighbours becomes alive (reproduction).

Represent the  $N \times N$  grid with an  $(N + 2) \times (N + 2)$  numpy matrix bordered by zeros. Implement:

- ▶ a `neighbours` function that counts the live neighbours of each cell.
- ▶ an `iterate` function that applies the rules.
- ▶ call your code on the glider pattern `[[0, 0, 1], [1, 0, 1], [0, 1, 1]]`.