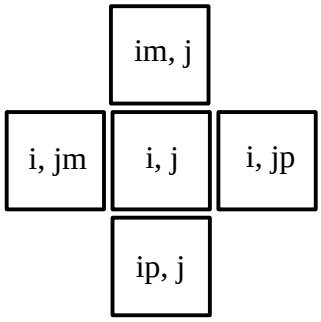
Index Notation for Neighbor Cells

by Sven Nilsen, 2020

When describing the algorithms for Self-Organizing Fractal Noise, I tried to come up with an index notation that is easy to understand and implement from just looking at the expression:



For example:

```
count := map[ip][j] + map[i][jp] + map[im][j] + map[i][jm]
```

The use of `i` and `j` is motivated by the storing columns of cells per row.
Translated into coordinates, `i` represents `y` and `j` represents `x`.

The reason `x` and `y` are not used, is because the order is reversed and this might seem confusing. Algorithms are symmetric, so the interpretation does not matter anyway, but using `i, j, im, jm, ip, jp` explains how to implement the algorithm no matter how the expression is interpreted.

Usually, the indices maps around to the opposite edge:

```
im := (i + h – 1) % h
ip := (i + 1) % h
jm := (j + w – 1) % w
jp := (j + 1) % w
```

This forms a topological surface of a torus in two dimensions. This has the benefit that the surface has no boundary, avoiding the need for specifying boundary conditions.

The width and height is obtained the following way:

```
w := len(map[0])
h := len(map)

map : [[]]
```