

Pascal's Triangle

Objective

Compute [Pascal's triangle](#) up to a given number of rows. In Pascal's Triangle each number is computed by adding the numbers to the right and left of the current position in the previous row.

The rows of Pascal's triangle are conventionally enumerated starting with row $n = 0$ at the top (the 0th row). The entries in each row are numbered from the left beginning with $k = 0$ and are usually staggered relative to the numbers in the adjacent rows. The triangle may be constructed in the following manner: In row 0 (the topmost row), there is a unique nonzero entry 1. Each entry of each subsequent row is constructed by adding the number above and to the left with the number above and to the right, treating blank entries as 0. For example, the initial number in the first (or any other) row is 1 (the sum of 0 and 1), whereas the numbers 1 and 3 in the third row are added to produce the number 4 in the fourth row.

Examples

```
  1
 1 1
1 2 1
1 3 3 1
1 4 6 4 1
# ... etc
```

For more information: https://en.wikipedia.org/wiki/Pascal%27s_triangle

Technical requirements

The code should be in a public Github repository. When you make a repository you should send us the link. Commit and push to Github as often as possible. In the code use only English language.

The code should be covered by RSpec tests.

Source: <http://exercism.io/exercises/ruby/pascals-triangle/readme>