# Challenge 4: Pascal's Triangle

In this challenge, you have to implement Pascal's triangle using a two-dimensional Array.

**We'll cover the following**


- Problem statement
  - Function prototype
  - Sample input
  - Sample output
  - Pictorial representation


## Problem statement

Implement a Java method that takes an *integer* size as input and **displays** a table that represents a Pascal's triangle using a *two-dimensional* array.

### Function prototype

```
void printPascalTri(int size)
```

### Sample input

```
int size = 5;
```

### Sample output

*Print* Pascal's triangle of the size 5.
Pascal's triangle is filled from the top towards the bottom. In *Pascal's triangle*:

- **first** and the **second** *rows* are set to **1**.
- Each *element* of the *triangle* (from the **third** row downward) is the **sum** of the element directly above it and the *element* to the **left** of the *element* directly **above** it.

> **Note:** In order to move a value to the next line you can use `\n`.

## Pictorial representation

See the example *Pascal triangle*(**size=5**) below:

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

Try to implement the solution by *yourself* once you have understood the *problem statement* clearly. Referring to the solution part should be your last resort.
**Good luck!**

```
class PrintTri {
    public static void printPascalTri(int size) { //define your function

        //write your code here for making and displaying pascals triangle
        //use can use \n to move numers to next line in the triangle
        //use " " to add space between numbers in triangle
        System.out.println(); //comment out this line when you write your code
    }
}
```

Printing the Pascal's triangle

---

Let’s discuss the solution of the above challenge in the next lesson.