Solution Review 3: Print a Right-Angled Triangle

This lesson gives a detailed solution review to the challenge in the previous lesson.

We'll cover the following

- Solution
  - Explanation

Solution

```
fn test(n:i32) {
    // define a nested for loop
    for i in 0..n { //outer loop
        for j in 0..i + 1 { // inner loop
            print!("&");
    }
    println!("");
    }
}
fn main(){
    println!("Right angled triangle when n = 5 ");
    test(5);
    println!("Right angled triangle when n = 6 ");
    test(6);
}
```

Explanation

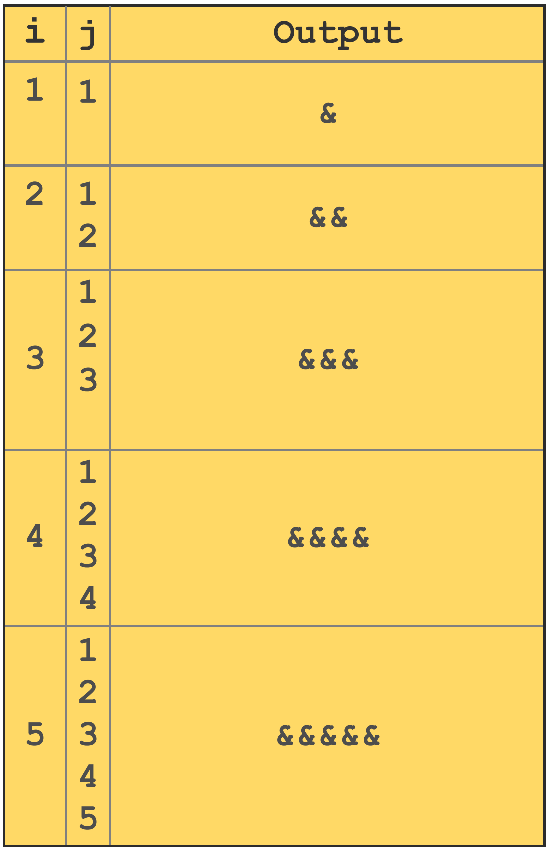
> The value `n` is given to you for which the right-angled triangle needs to be printed.

nested `for` loop

- On **line 3**, in the **outer `for` loop**
  - An iterator `i` iterates over the **range 0 to `n`**.
- On **line 4**, for each `i`, within the **inner `for` loop**, an iterator `j` iterates over the **range 0 to `i` + 1**

the **range 0 to `i` + 1**

  - In each iteration, it prints the character `&` within the `print!` macro on **line** 5.
  - When `j` **equals** `i+1`**, the inner loop breaks** and the control goes to outer `for` loop.
- When the inner `for` loop terminates `println!("")` appends **a new line**.
- When the value of `i` **equals** `n`**, the outer loop terminates**.
- The output is a right-angled triangle.



i	j	Output
1	1	&
2	1 2	&&
3	1 2 3	&&&
4	1 2 3 4	&&&&
5	1 2 3 4 5	&&&&&

Now you have learned about loops in Rust, but what if you want to make a reusable piece of code in your program and call it whenever desired? Let’s study “Functions” in the next chapter.