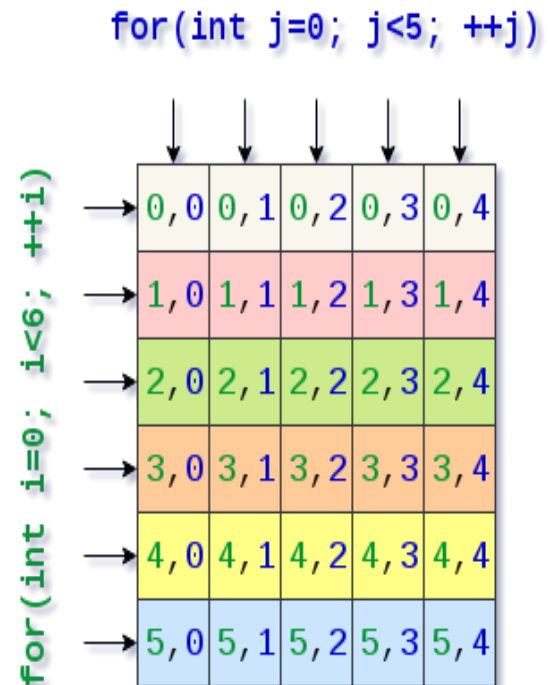


Nested For Loops

Nested **for** loops may be used when displaying a calendar, drawing shapes, or altering elements of two-dimensional arrays.

In the picture, the outer **for** loop is denoted in **seafoam green** and the inner **for** loop is denoted in **blue**. The outer loop controls the number of rows and the inner loop controls the number of columns. The inner loop also does the work. In other words, the code that is to be displayed or the code to be used to alter elements of a two-dimensional array will go inside of the inner loop. If you want to display something in a grid-like fashion, you must **cout << endl;** or **cout << '\n';** after the inner loop, but still inside of the outer loop. This makes each complete looping cycle on its own line. Each colored row in the picture denotes one complete cycle of both loops.



Seafoam green = part of outer loop
Blue = part of inner loop

```
for(int i = 0; i < 6; ++i) {  
    for(int j = 0; j < 5; ++j) {  
        cout << i << ", " << j << ' ';  
    }  
    cout << endl; //Displays each colored row on their own line  
}
```

This example shows how to display what the picture above indicates. Whatever is in the curly brackets of the outer **for** loop will execute six times, including the inner **for** loop. Each of those six times, the inner **for** loop will execute five times. During the first iteration, **i** will be zero. **j** will start at zero and increment until it is five. Then, the **cout << endl;** executes after the inner **for** loop and ensures that the row prints on its own separate line. This creates the first row of the picture. The second iteration now begins with **i** equalling one and **j** restarting at zero and incrementing until five. The newline is displayed. This creates the second row of the picture. Repeat the iterative process until **i** equals six. Each iteration is the same process, but **i** will be a different number.