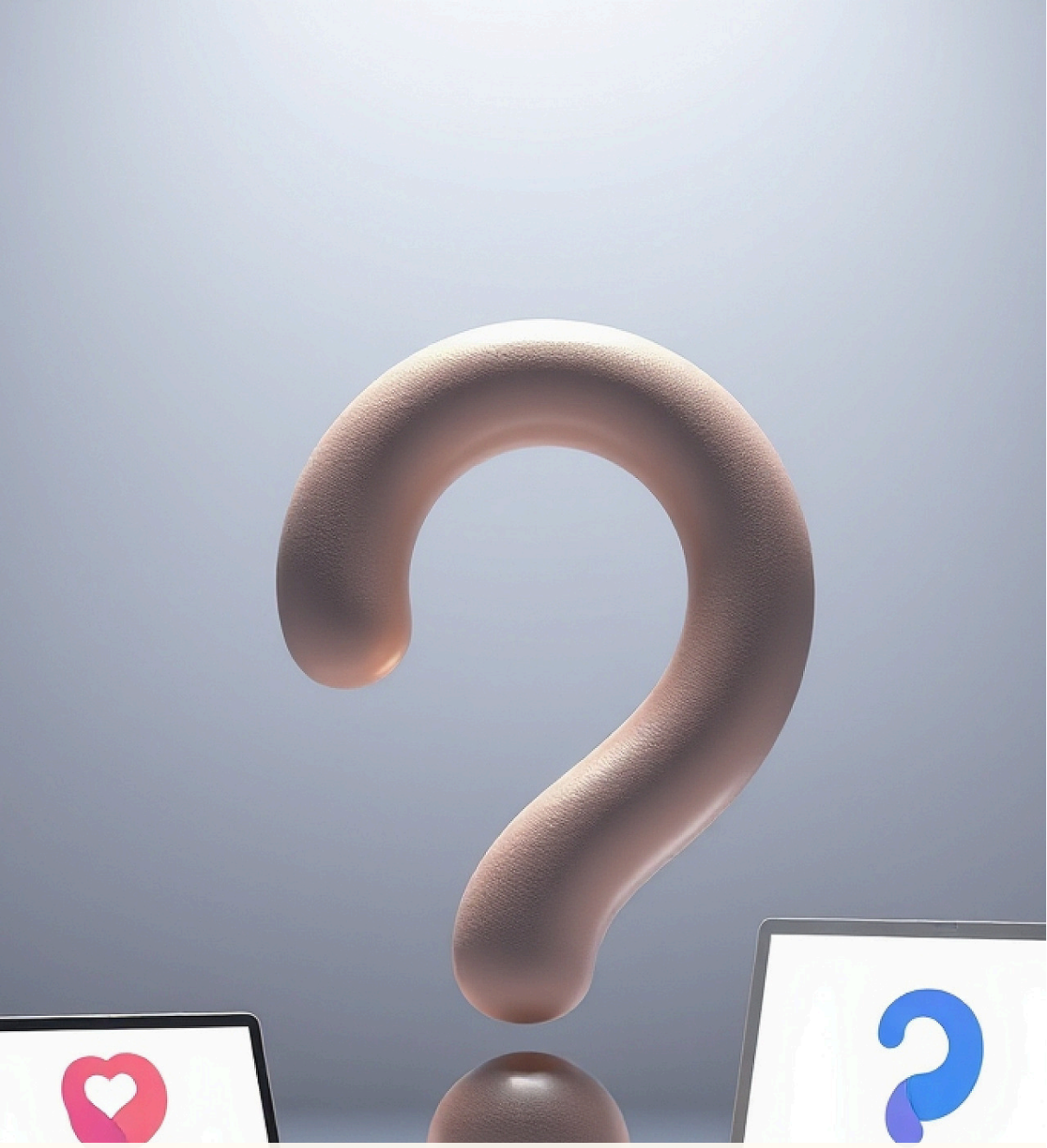
What is a nested loop used for?

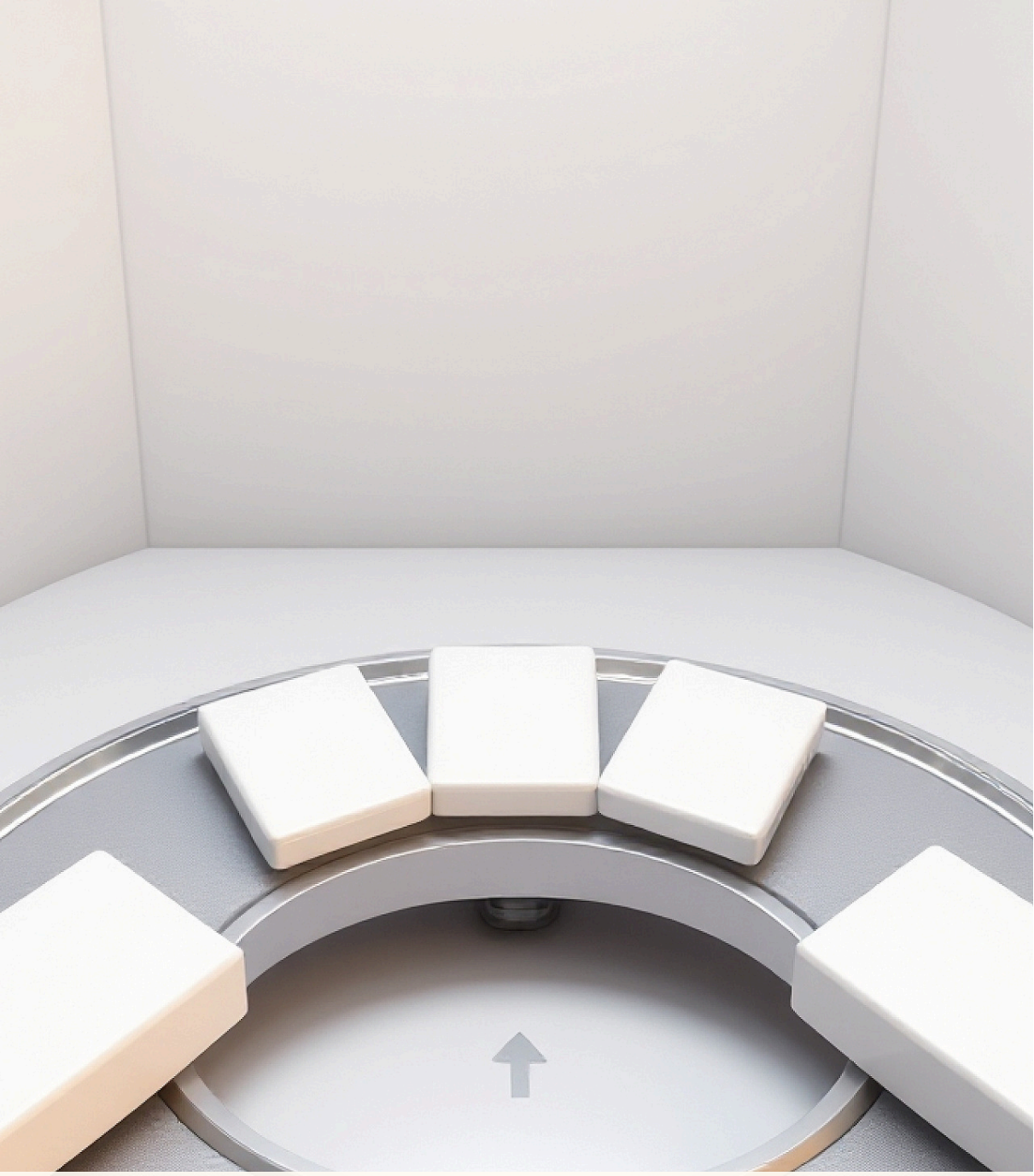


www.yzokul.com.tr

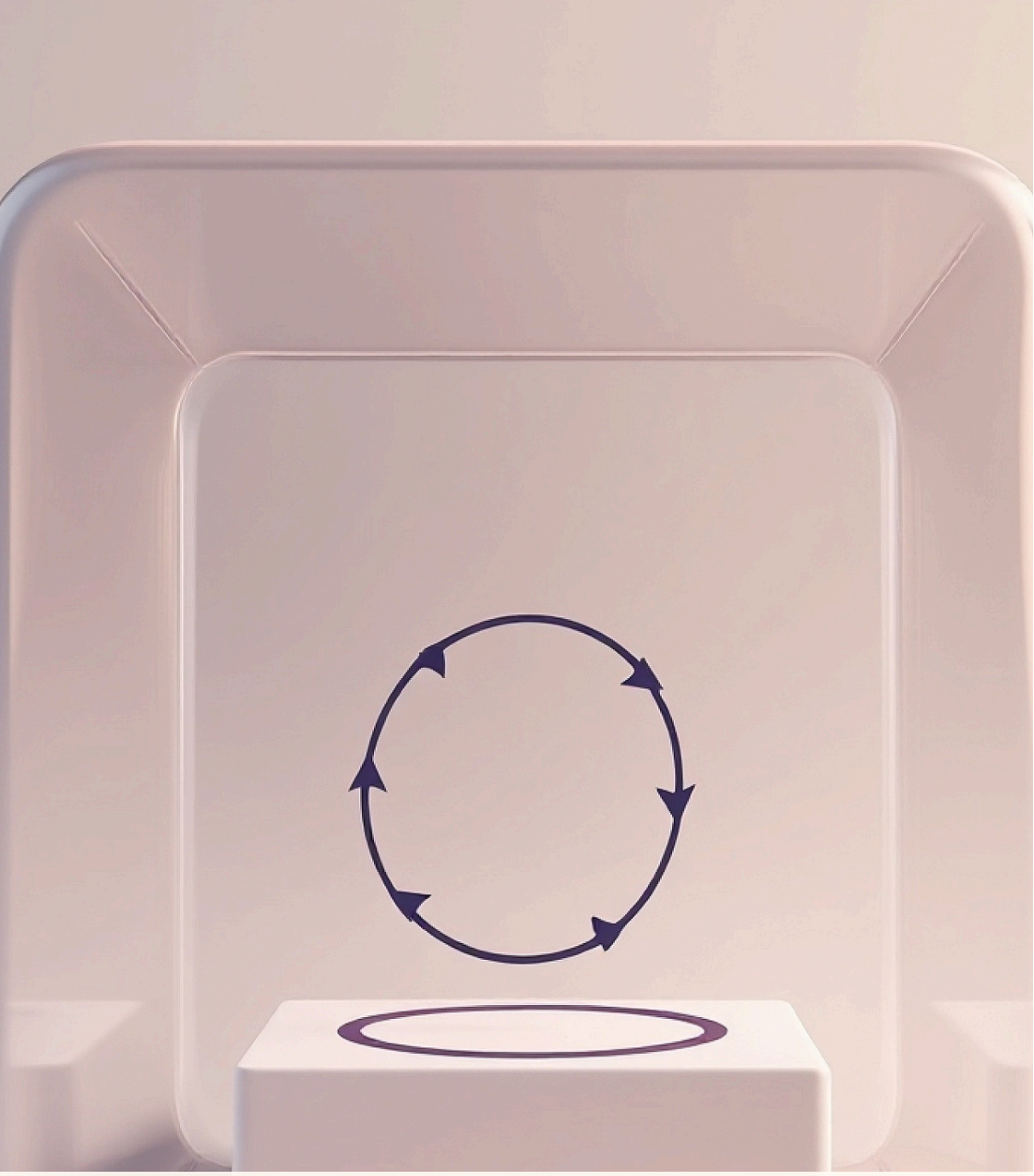
What is a nested loop used for?



You read the question: what is a nested loop used for? The short answer is that a nested loop is a loop inside another loop, and it is used to repeat work over combinations of things, like rows and columns. It lets you handle two directions of data in one structure. This book explains that idea step by step, starting from the very beginning so you need no prior knowledge.



A loop simply repeats a set of steps many times. For example, a loop can print the word hello five times without you writing it five times. You write the steps once, and the loop runs them again and again. Loops save effort and make repeated work clear and short. Every loop has a start, a set of steps, and a rule for when to stop.



A nested loop means one loop sits inside another loop. The outer loop runs its steps, and one of those steps is the whole inner loop. So the inner loop is part of the outer loop's body. This structure is called a loop inside a loop. The outer loop controls the big repetitions, and the inner loop handles the small repetitions inside each one.



Watch the outer loop take its first step. At that single step, the inner loop starts and does all of its repetitions. The outer loop pauses while the inner loop works. This means one step of the outer loop can contain many steps of the inner loop. The inner loop is not skipped; it runs completely before anything else in the outer loop continues.



The inner loop completes all of its repetitions before the outer loop moves on. Only then does the outer loop take its next step. For example, if the inner loop repeats four times, those four repetitions finish first. The outer loop waits until the inner loop is done. This order is important: the inner loop always finishes fully at each outer step.



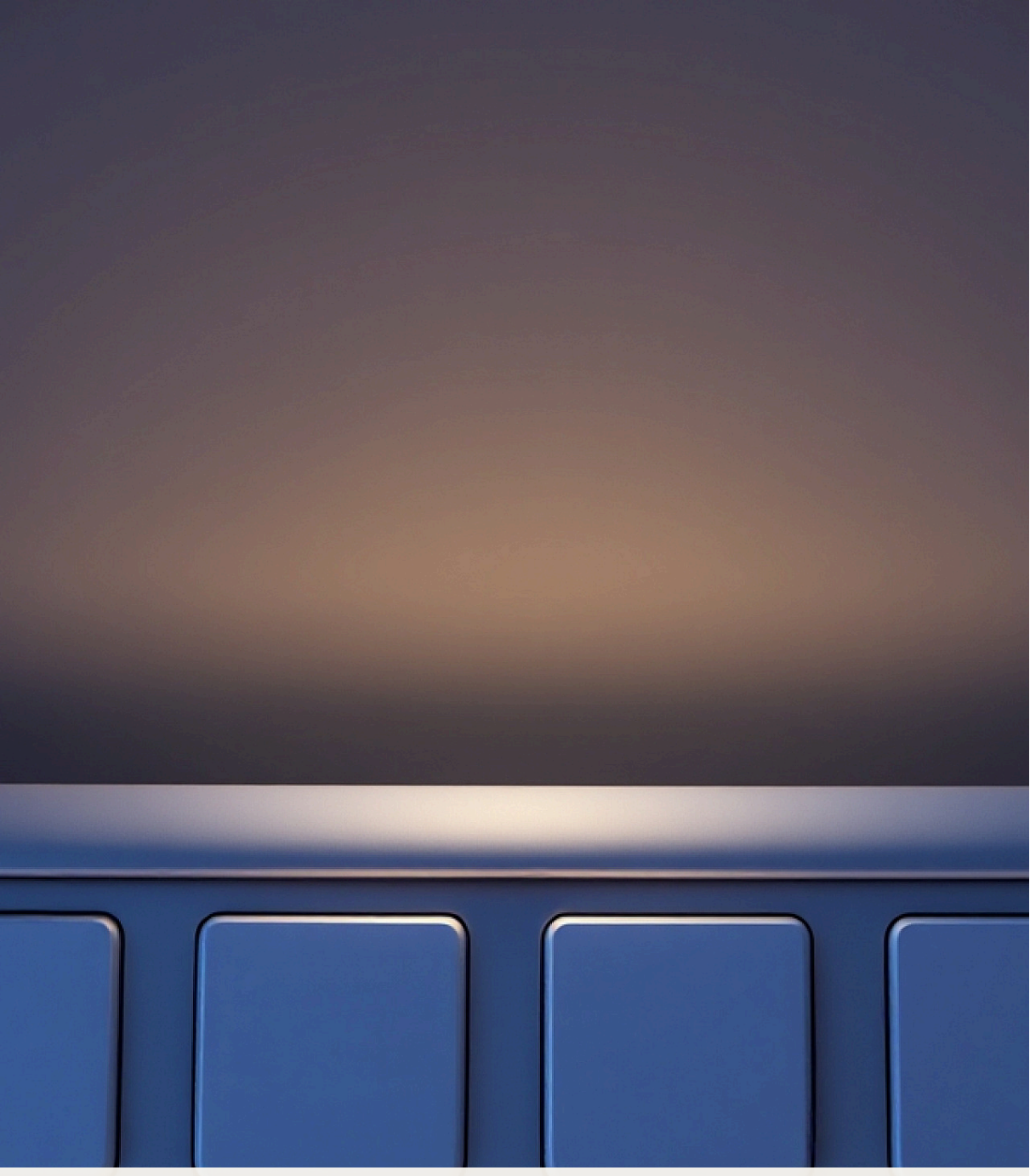
Now watch the outer loop take its second step. The inner loop runs fully again from the start. This repeats for every step of the outer loop. So if the outer loop has three steps, the inner loop runs its full set of repetitions three separate times. Each outer step gets a complete run of the inner loop, not a partial one.



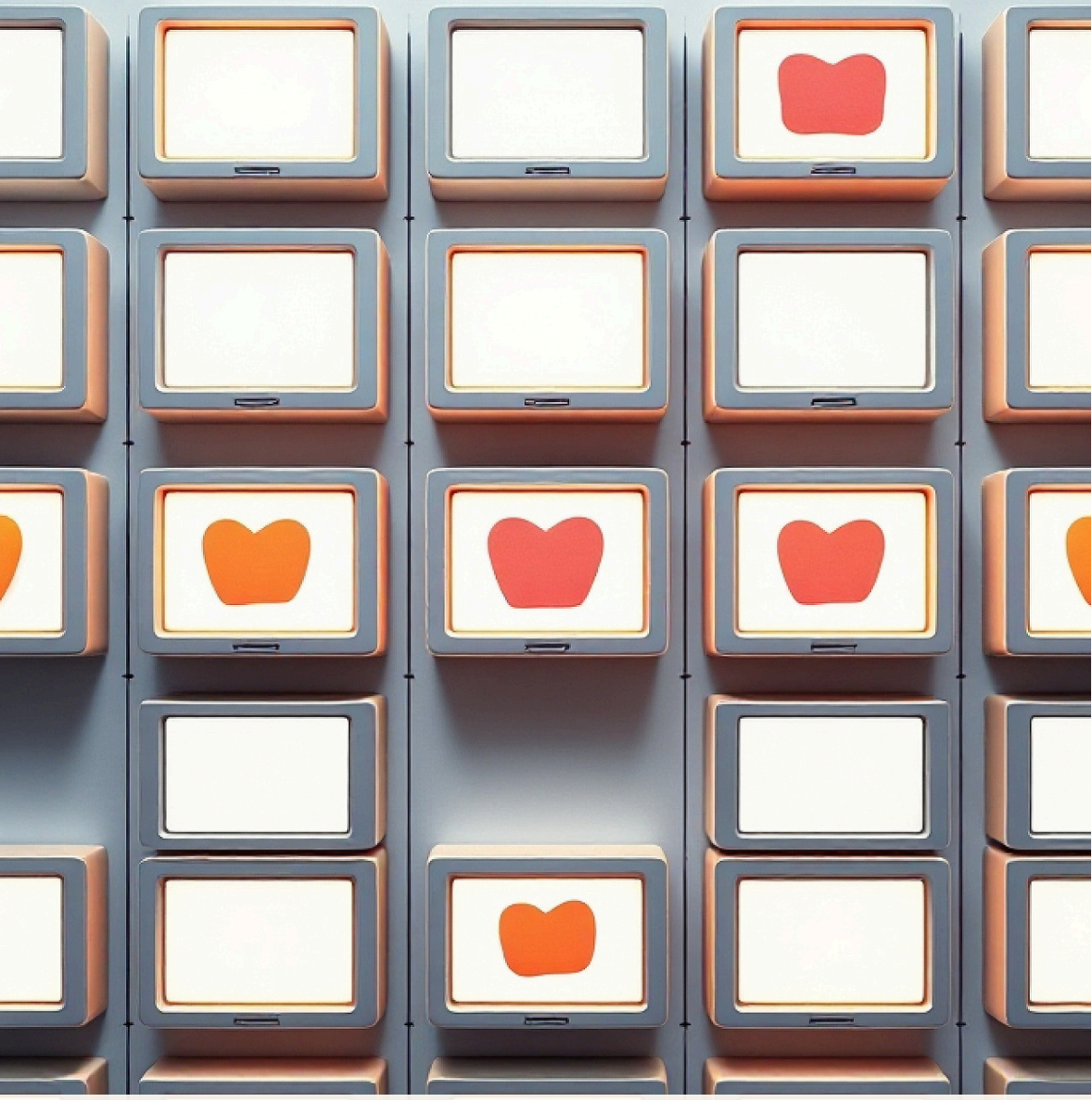
Count the total work. If the outer loop runs 3 times and the inner loop runs 4 times, the inner steps happen 3 times 4, which is 12 times. The counts multiply, not add. This multiplication is the key fact about nested loops. It tells you how much total repetition happens. Remember this rule whenever you see a loop inside a loop.



Picture a table with rows. The outer loop goes through each row, one row at a time. Each step of the outer loop handles one row. This is a natural way to use a nested loop: the outer loop moves down the rows. For example, a table with five rows means the outer loop takes five steps, one for each row.



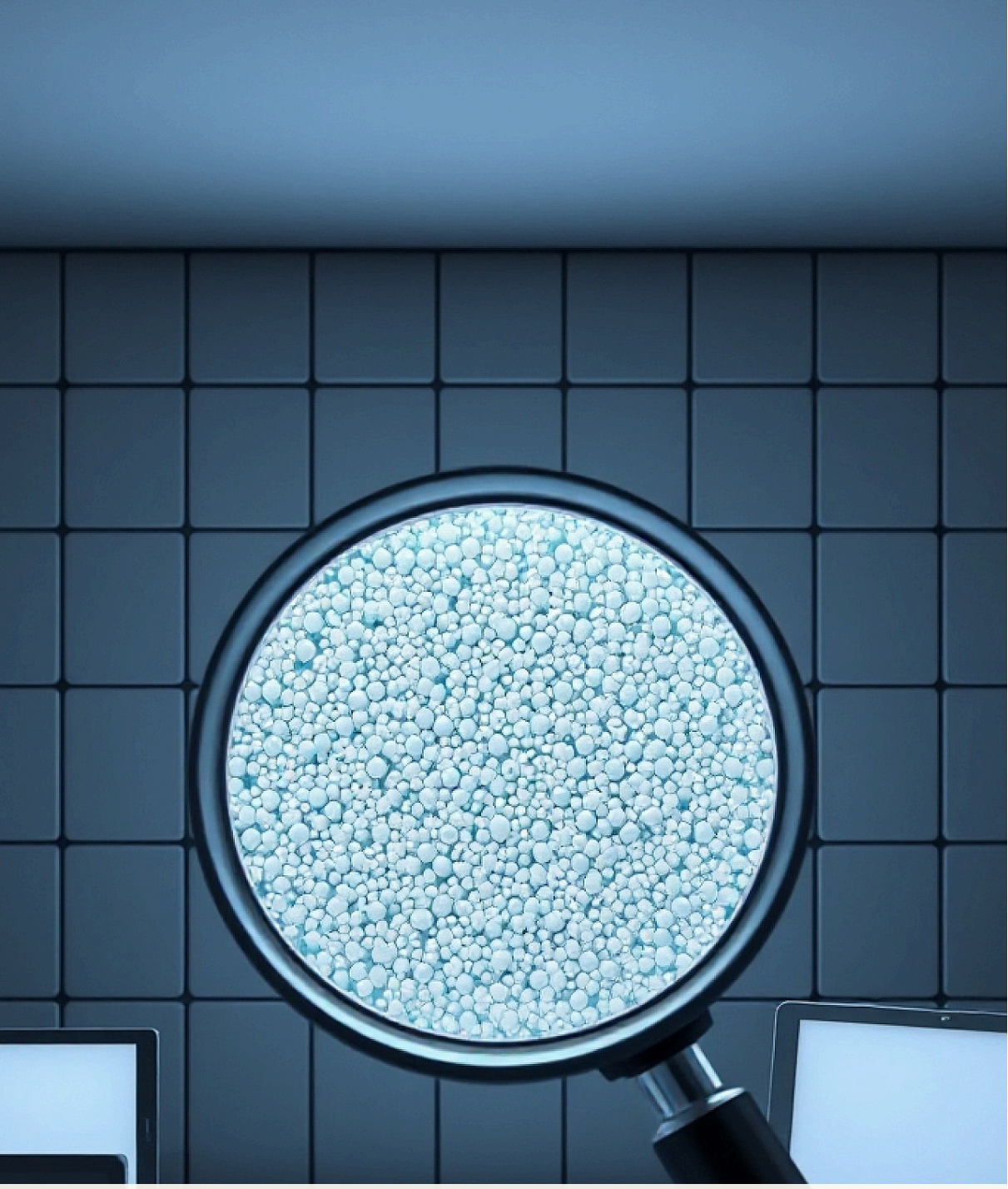
Inside each row there are columns. The inner loop goes through each column of the current row. At every outer step, the inner loop visits all the columns in that row. So the inner loop handles the horizontal direction, and the outer loop handles the vertical direction. Together they cover the whole row, column by column.



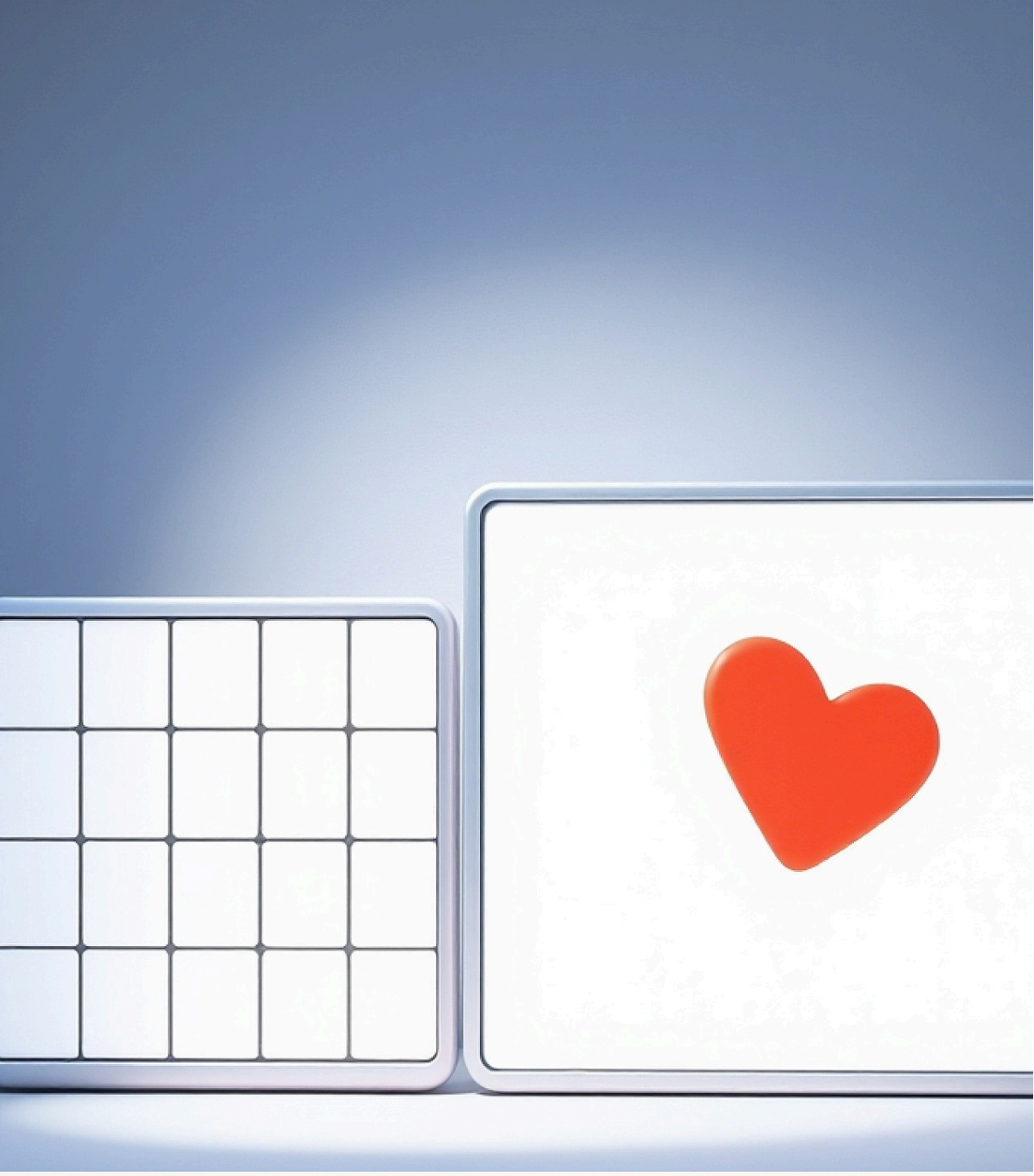
Together the two loops visit every cell of the table. For each row, the inner loop touches every column, so no cell is missed. If the table has 3 rows and 4 columns, the inner steps happen 12 times, once for each cell. This is exactly why nested loops are used: to reach every combination of row and column.



This row-and-column pattern is called two-dimensional data. Two-dimensional data has two directions, like a grid or a table. A nested loop is a natural fit whenever data has two directions. The outer loop moves along one direction, and the inner loop moves along the other. So nested loops are used to process tables, grids, and similar structures.



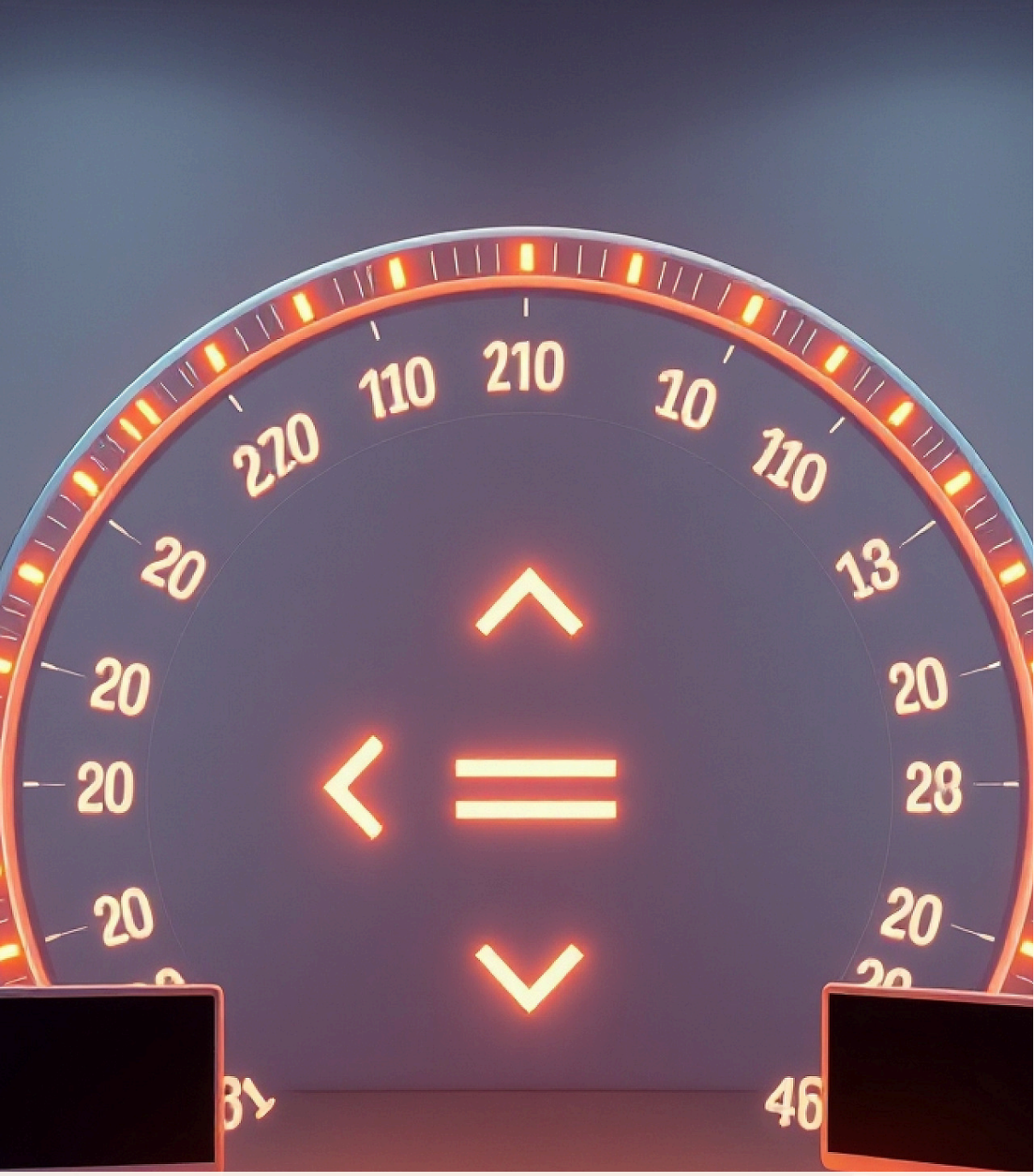
The repetition counts multiply. A 10 by 10 table means the inner steps happen 100 times, not 20. You might guess 10 plus 10, but it is 10 times 10. This multiplication is why nested loops can do much more work than they appear to. Always multiply the outer count by the inner count to find the total.



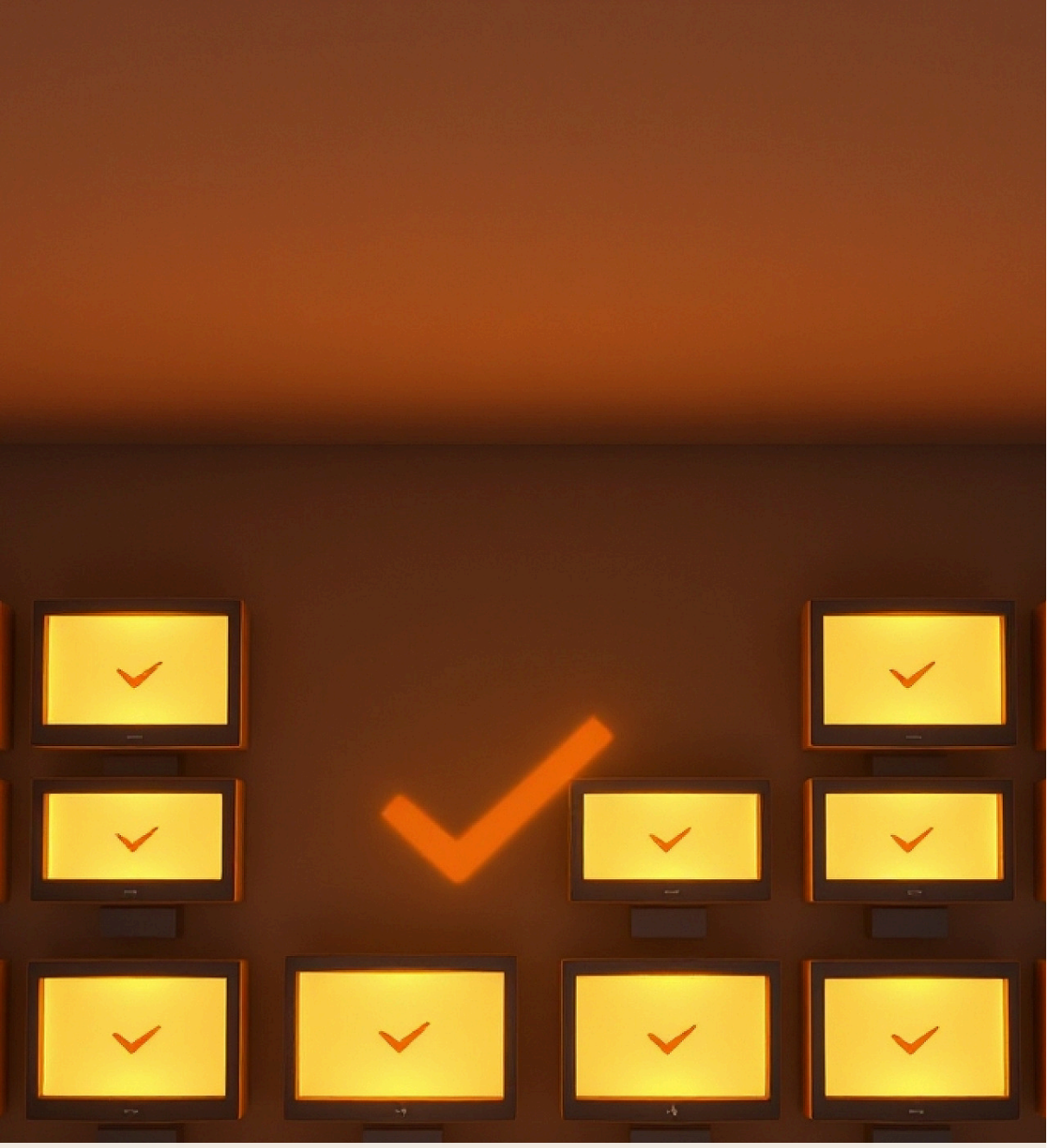
With large data the amount of processing can grow very fast. Doubling both sizes makes the work four times bigger. For example, a 20 by 20 table needs 400 inner steps, while a 10 by 10 table needs only 100. So nested loops are powerful but can become slow when the data is large. Keep this in mind when you use them.



A common mistake is thinking the inner loop runs only once. In fact it runs fully again at every step of the outer loop. If you forget this, you will misjudge how many times the inner steps happen. Always remember that the inner loop restarts from the beginning for each outer step. This is the heart of how a nested loop works.



Another common mistake is forgetting that the counts multiply. Beginners expect a small number of steps, but nested loops can do far more work than they guess. For example, 100 rows and 100 columns means 10,000 inner steps, not 200. This surprise can make a program slow. Always multiply the counts to see the real total.



A nested loop is a loop inside a loop, where the inner loop repeats fully at each outer step. It is used for two-dimensional data like rows and columns, because it visits every combination. But remember the repetition counts multiply, so processing grows quickly with large data. Use nested loops when you need to cover two directions, and check the total count before you run them.