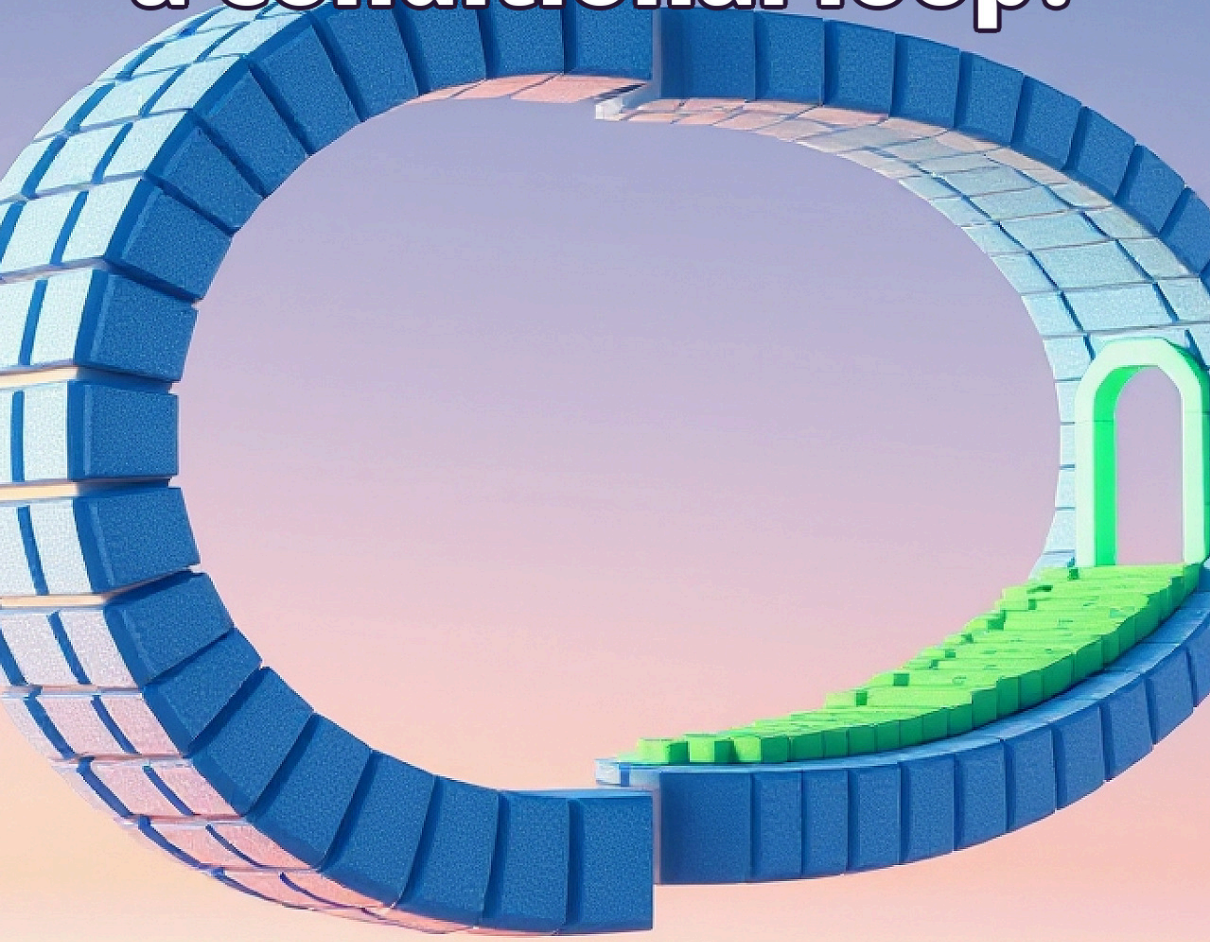
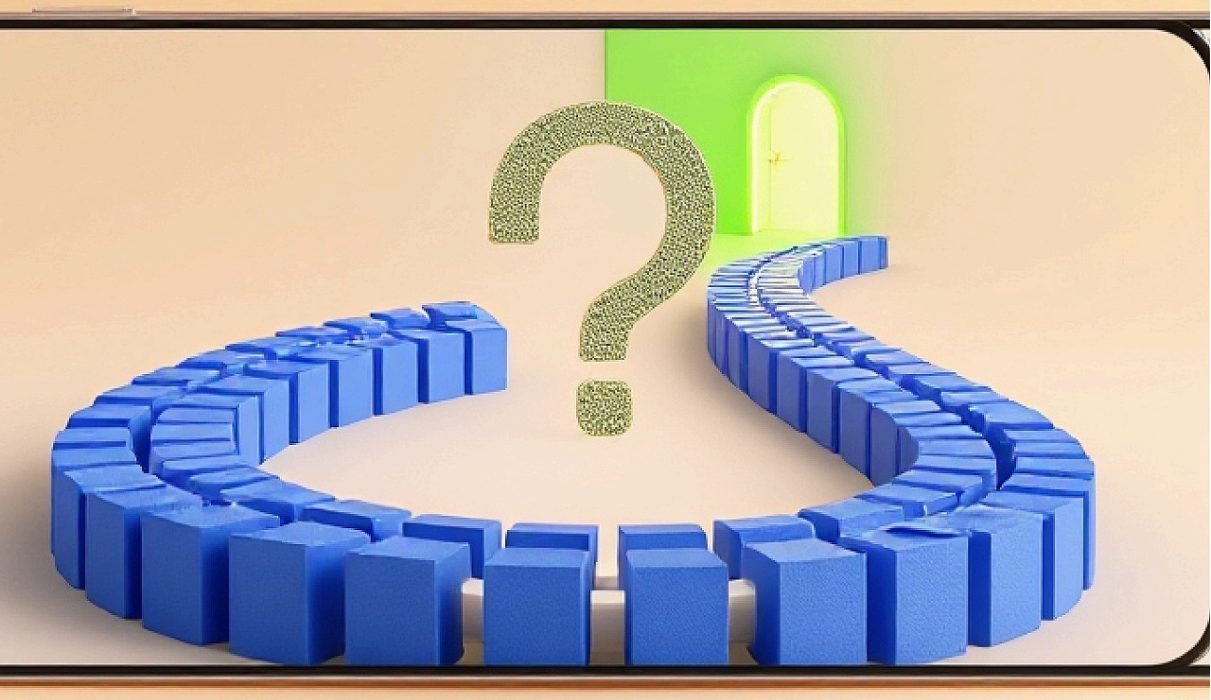
What is the difference between a counted loop and a conditional loop?



www.yzokul.com.tr

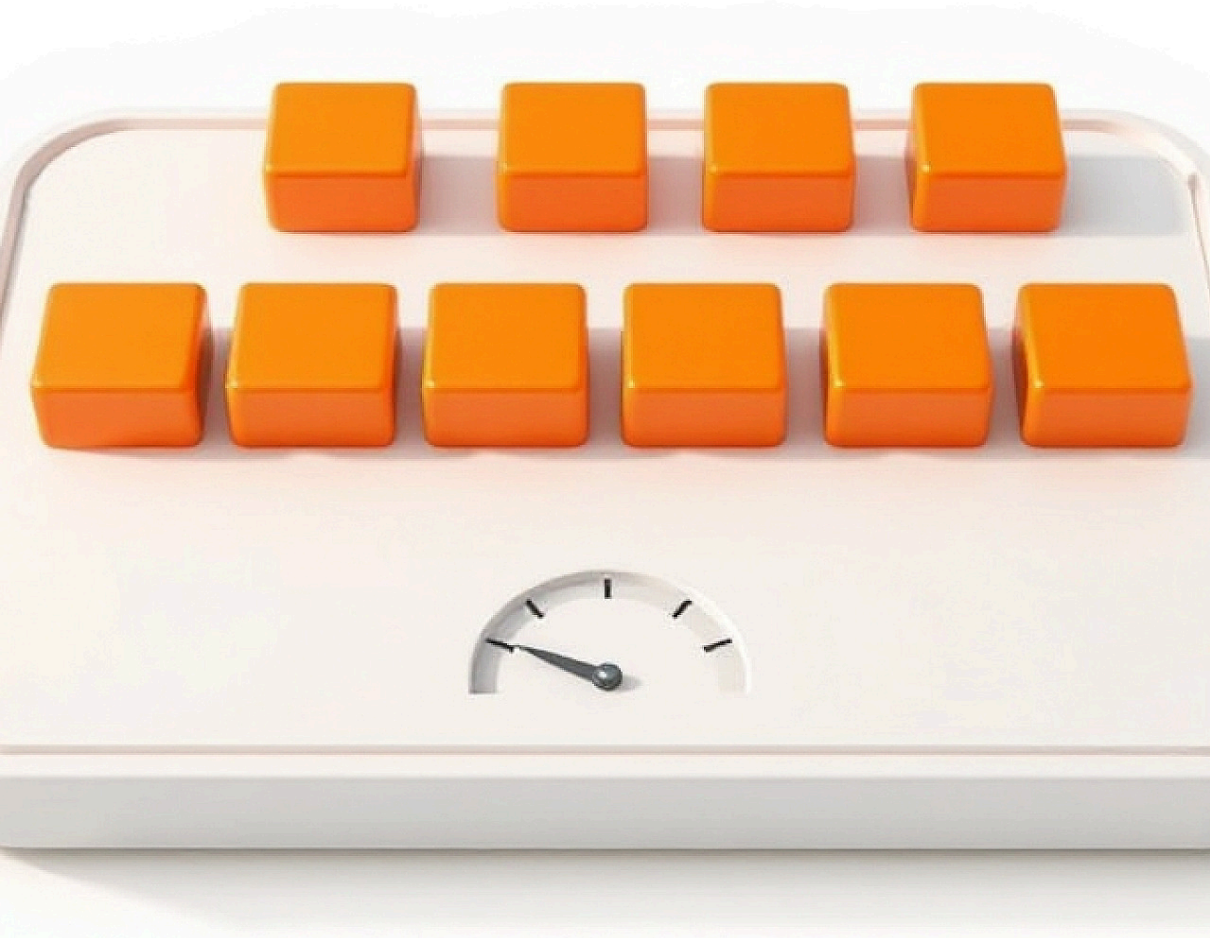
What is the difference between a counted loop and a conditional loop?



You have likely heard the terms counted loop and conditional loop and want to know how they differ. The short answer is that a counted loop repeats over a specific range or collection of items, while a conditional loop repeats as long as a condition stays true. This book walks you through both ideas step by step, starting from zero.



A loop is a way to make a program repeat some steps instead of writing them again and again. Seeing repetition as a single instruction is the first idea you need. For example, instead of writing a jump command ten times, you write a loop that says repeat the jump ten times. The program then handles the repetition for you.



Picture a jumping animation that must play exactly ten times. You know the number ten before the repeating starts. This is different from a situation where you must wait for something to happen, because here the count is fixed and known up front. The program can simply repeat the jump ten times and then stop.



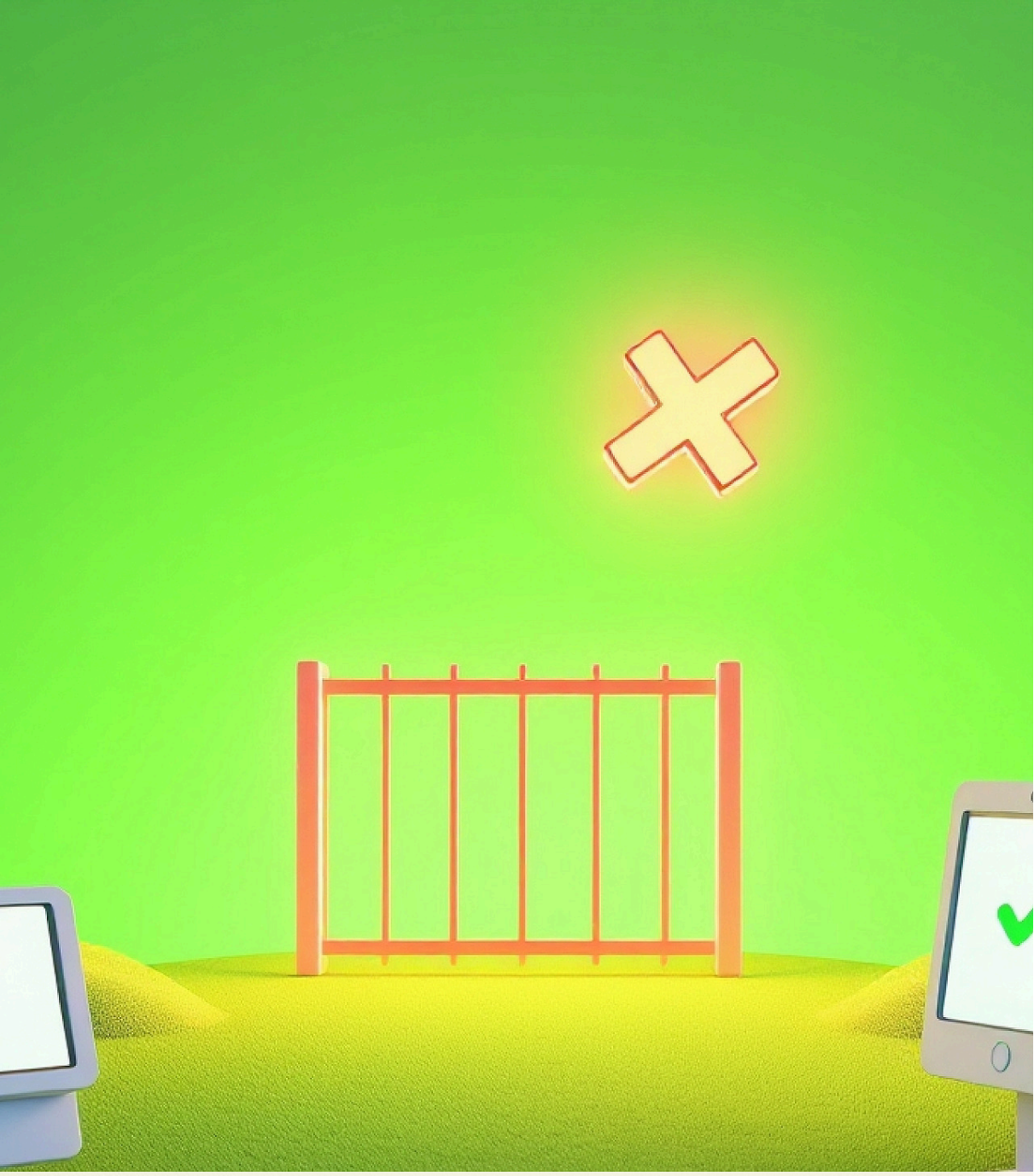
A counted loop goes through a specific range or group of items, one by one. The count or collection is decided before the loop starts, so the loop knows exactly when to stop. For a jumping animation, the counted loop repeats the jump ten times, then moves on. The range is fixed, so the loop does not need to check anything else.



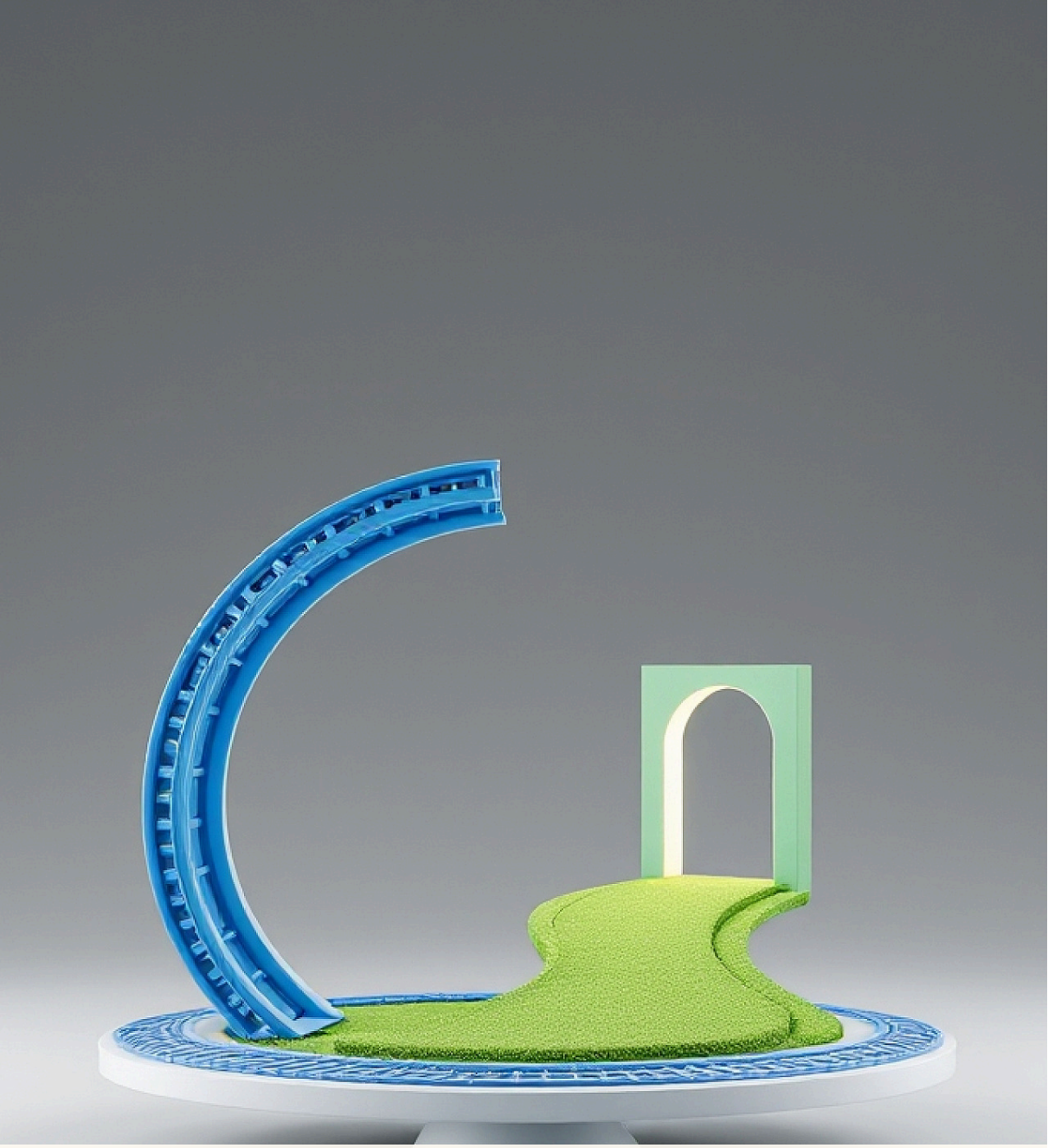
A counted loop also works to visit each item in a collection, like each name on a list. Each item gets its turn exactly once. You might use it to print every name, greet every student, or check every photo. The collection is known in advance, so the loop simply goes through each item until none are left.



Picture a program that waits until the user gives the correct answer. Nobody knows in advance how many tries it will take. The user might get it right on the first try, or need many attempts. The program cannot count a fixed number of rounds, because the number depends on what the user does. It must keep repeating until the answer is correct.



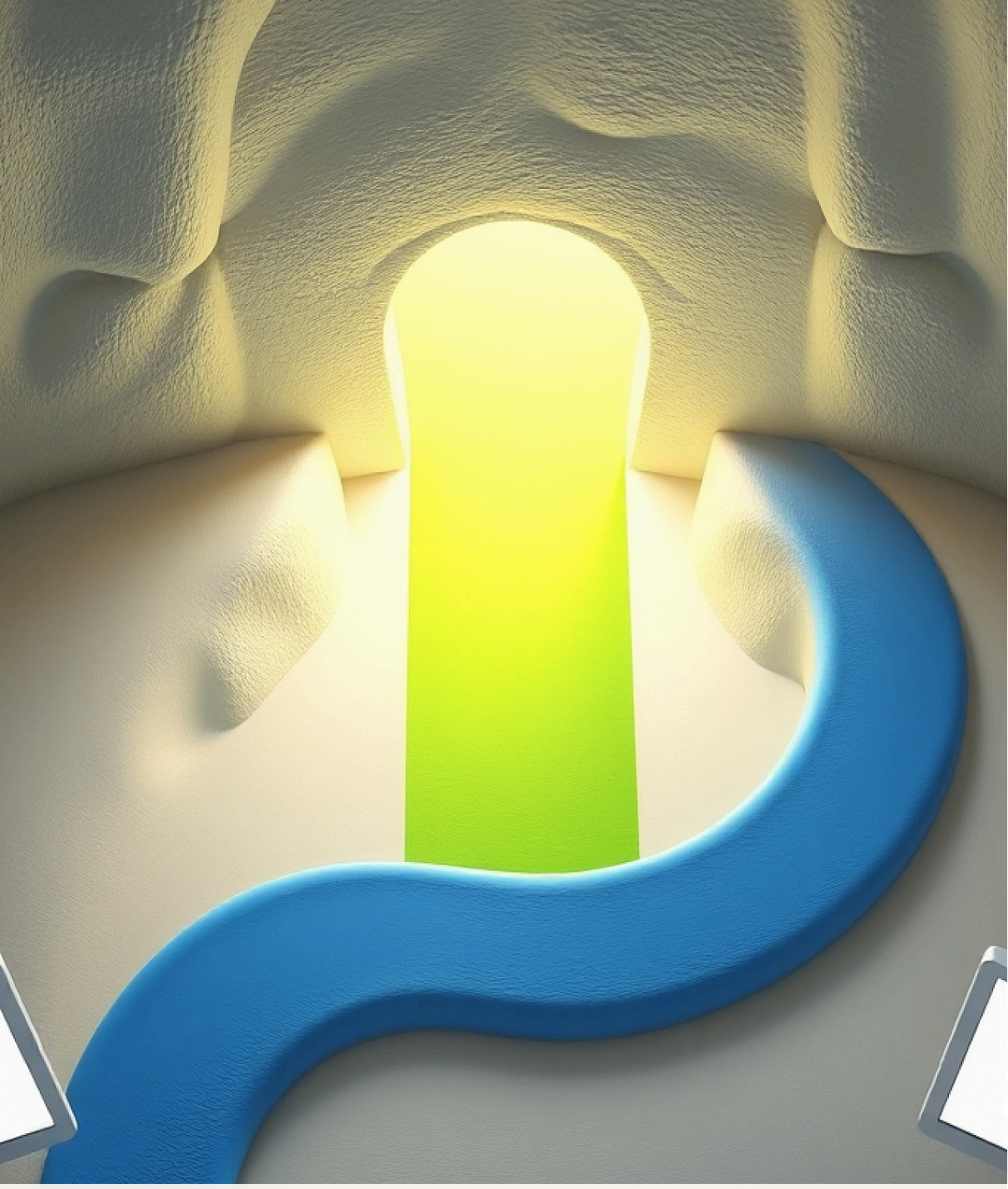
A conditional loop repeats as long as a condition continues to be true. It stops only when the condition changes, not after a fixed number of rounds. For the waiting program, the condition might be the answer is not correct. While that condition is true, the program keeps asking. When the answer becomes correct, the condition turns false and the loop stops.



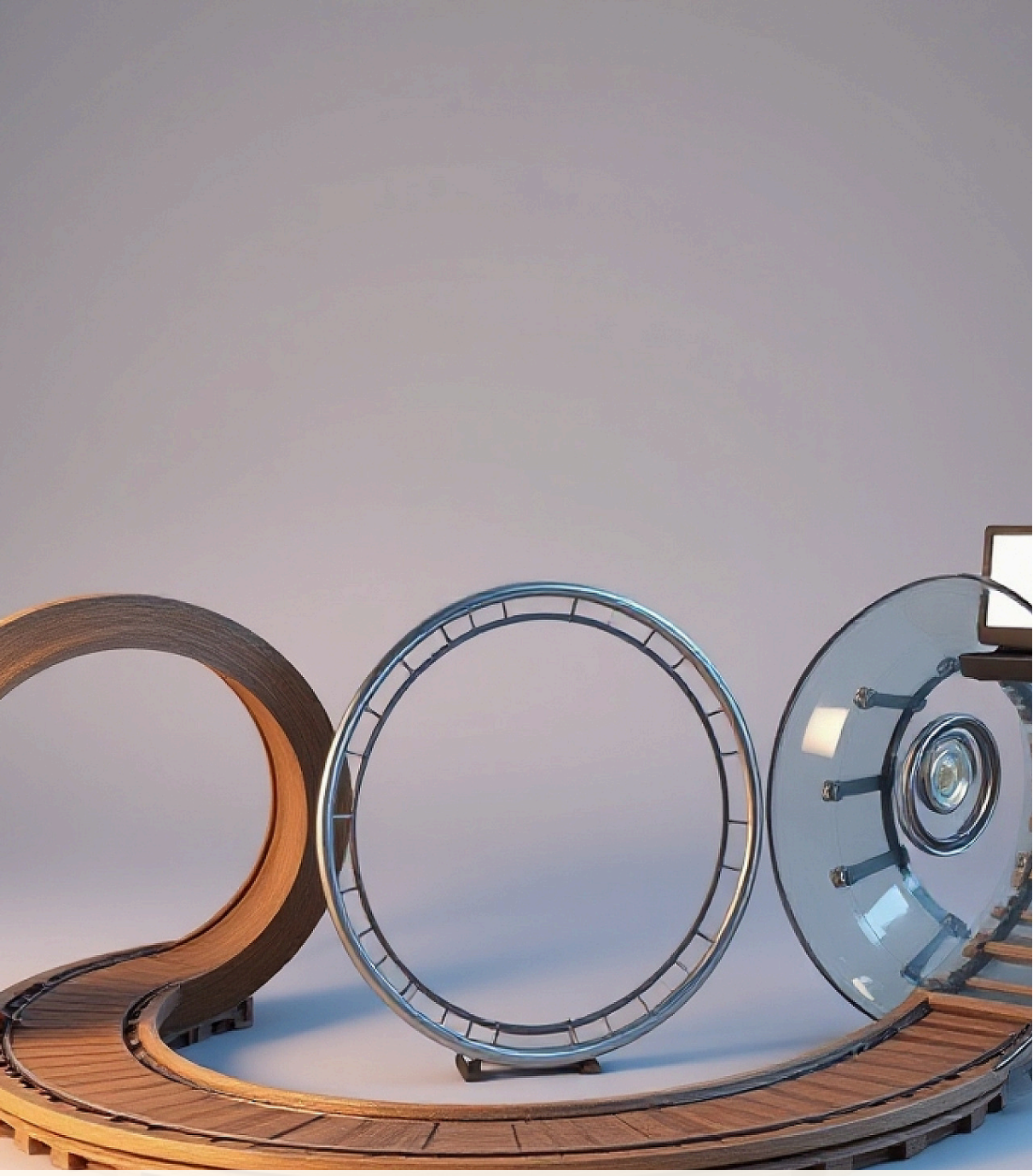
Compare the two side by side: a counted loop repeats over a count or collection, while a conditional loop repeats while a condition holds. One knows its number of rounds in advance; the other checks a condition each time. The need decides which one fits. If you know the count, a counted loop is natural; if you are waiting for something, a conditional loop fits.



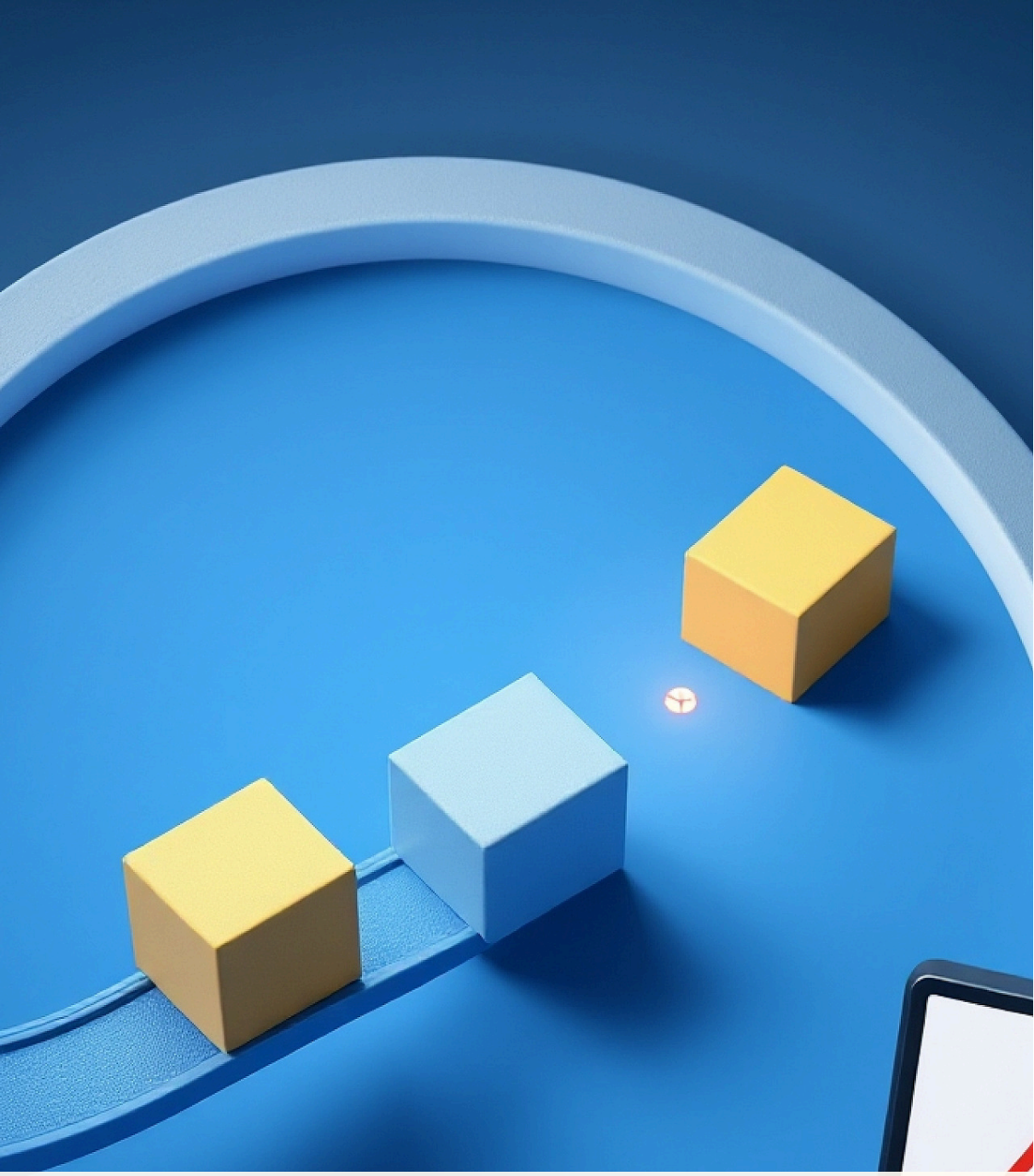
Match needs to loops to see the difference clearly. Ten jumps means a counted loop, because the number ten is known. Waiting for a right answer means a conditional loop, because the number of tries is unknown. The choice comes from the problem, not from habit. Ask what you know in advance and what you must wait for.



Many problems can be solved with either approach. A counted loop can track a condition by checking it inside the loop, and a conditional loop can count by adding one each round. So the two often overlap. Knowing both helps you pick the one that is clearer for your situation, but neither is always wrong.



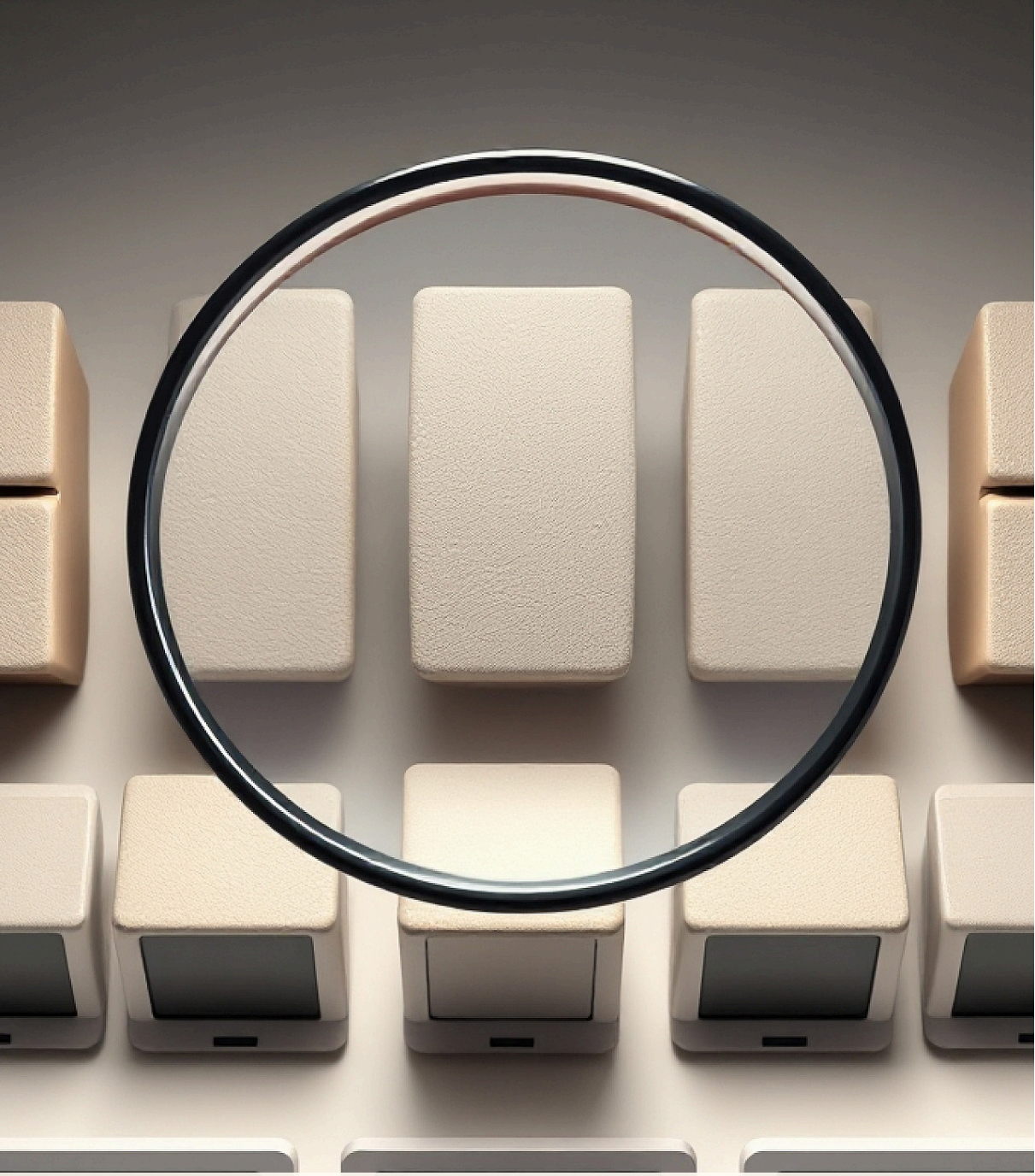
How each loop is written differs from one programming language to another. The idea stays the same even when the words change. Some languages use a for loop for counted repetition and a while loop for conditional repetition. Others use different names, but the concept of repeating over a range or while a condition holds remains the same.



A common mistake is using a counted loop when the number of repeats is unknown. The program then stops too early or too late. For example, if you use a counted loop to wait for a correct answer, you must guess how many tries the user needs. If you guess wrong, the program may stop before the user succeeds or keep asking after success.



Another mistake is a conditional loop whose condition never becomes false, so it runs forever. The condition must be able to change. For example, if the condition is the answer is not correct, the program must update the answer when the user responds. Otherwise the condition stays true and the loop never ends, which freezes the program.



A counted loop can run one time too many or too few. Check carefully where the count starts and when it stops. For example, repeating ten times might mean counting from zero to nine, not from one to ten. If you start at one and stop at ten, you get ten rounds; if you start at zero and stop at ten, you get eleven. Always check the start and end.



Practice a simple check: do I know how many times, or am I waiting for something? The answer points you to the counted or the conditional loop. If you know the number of repeats, use a counted loop. If you are waiting for a condition to become true, use a conditional loop. This question quickly guides your choice.



Sum up the key points: counted loops repeat over a known range or collection, conditional loops repeat while a condition stays true. Remember that the need shapes the loop. Many problems work either way, so focus on clarity. With this understanding, you can choose the loop that fits your task and avoid common mistakes.