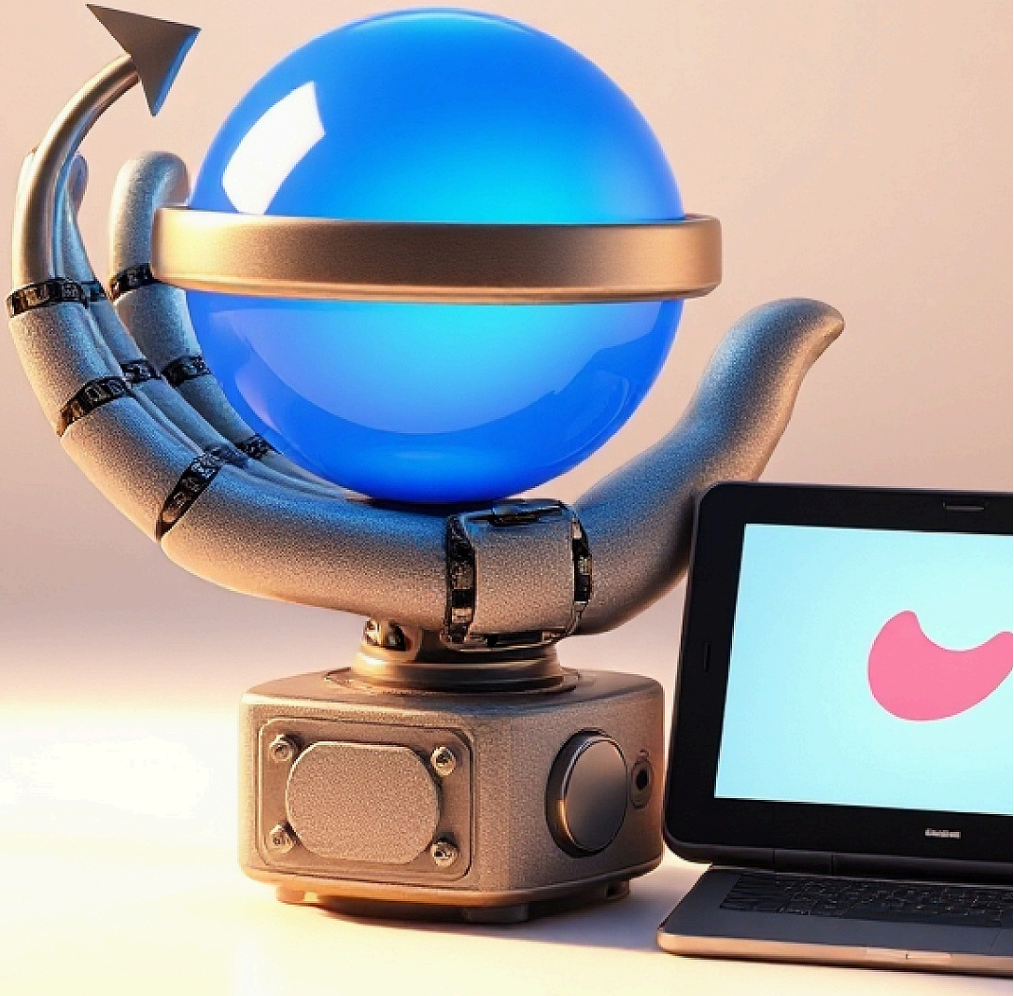
What does it mean for a function to return a value?



www.yzokul.com.tr

What does it mean for a function to return a value?



You may wonder what it means for a function to return a value. The short answer is that a function can hand its result back to the place that called it. That place is called the caller. The function does its job and then gives the result to the caller. This handing back is what returning a value means.



A function is a named block of instructions that does one job. You give the function a name so you can use it later. When you want that job done, you call the function by its name. The function then runs its instructions. Think of it as a tool you can use whenever you need it.



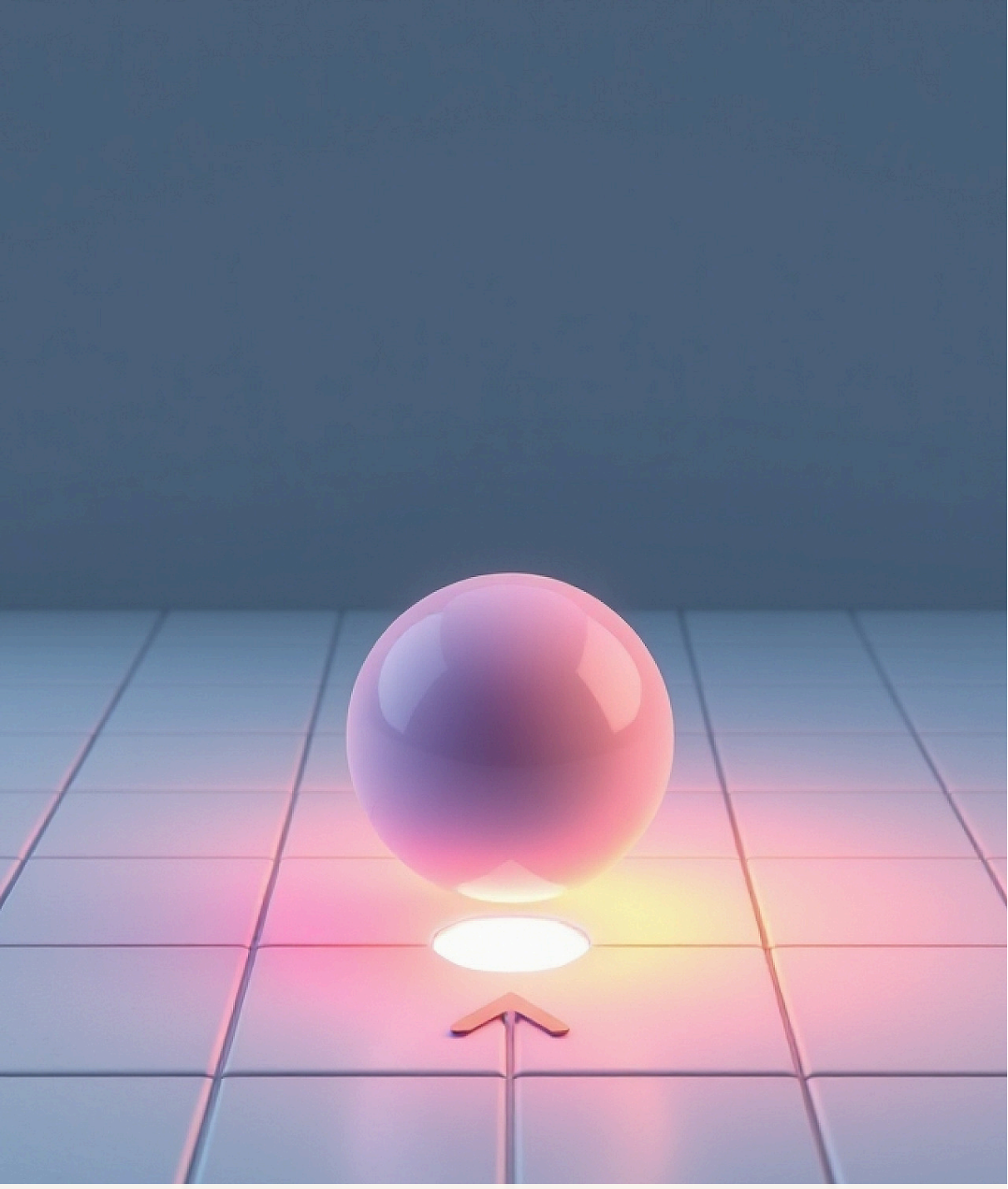
The place in your code that calls the function is called the caller. The caller asks the function to do its job by writing the function's name. For example, if you have a function named add, you call it by writing add. The caller is simply the code that makes the call. It waits for the function to finish.



Look at a simple function that adds two numbers. Its job is to produce the sum of those numbers. You give it two numbers, and it calculates the total. For example, if you give it 2 and 3, it produces 5. The function's purpose is to make that sum available. The sum is the result of its work.



The function gives the sum back to the caller. This handing back is exactly what returning a value means. The function does not keep the sum for itself; it returns it. The caller receives the sum as the result of the call. So the value travels from the function back to the caller.



Picture the caller receiving the result at the exact spot where it made the call. When the function finishes, the value flows back to that spot. For example, if you write add(2, 3) in your code, the value 5 appears there. The caller can then use that value right away. The value is now available at the call site.



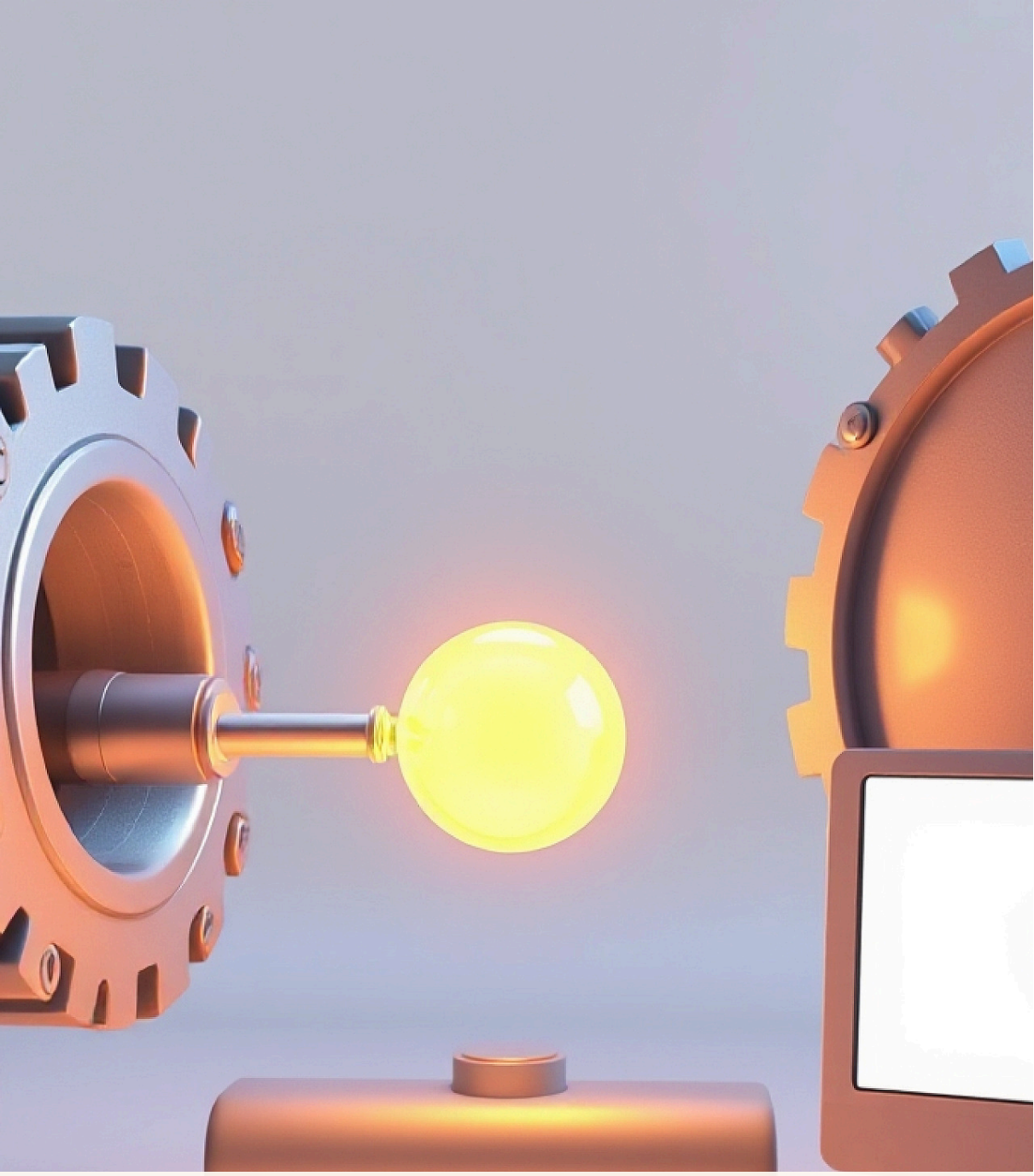
Printing shows text on the screen for a person to read. Returning gives the result back to the program for further use. These are two different operations. A function can print something without returning a value. It can also return a value without printing anything. You need to know which one you want.



A function can print a number to the screen and still return nothing useful. Printing only displays information; it does not hand a result back to the caller. For example, a function might print the sum but return nothing. The caller sees the printed number but gets no value to use. Printing is not returning.



The returned value can be assigned to a variable. A variable is a named storage location that holds a value. For example, you can write result = add(2, 3). The variable result then holds the value 5. You can use that variable later in your program. Storing the result lets you keep it for later.



The returned value can be sent straight into another function. One function's result becomes another function's input. For example, you can write print(add(2, 3)). Here, the add function returns 5, and that 5 is passed to the print function. This chaining lets you combine functions easily. The result flows from one function to the next.



The returned value can be checked in a condition. A condition is a test that decides what the program does next. For example, you can write if add(2, 3) > 4: and then do something. The program checks whether the returned value is greater than 4. Based on that, it chooses a path. This makes decisions using the result.



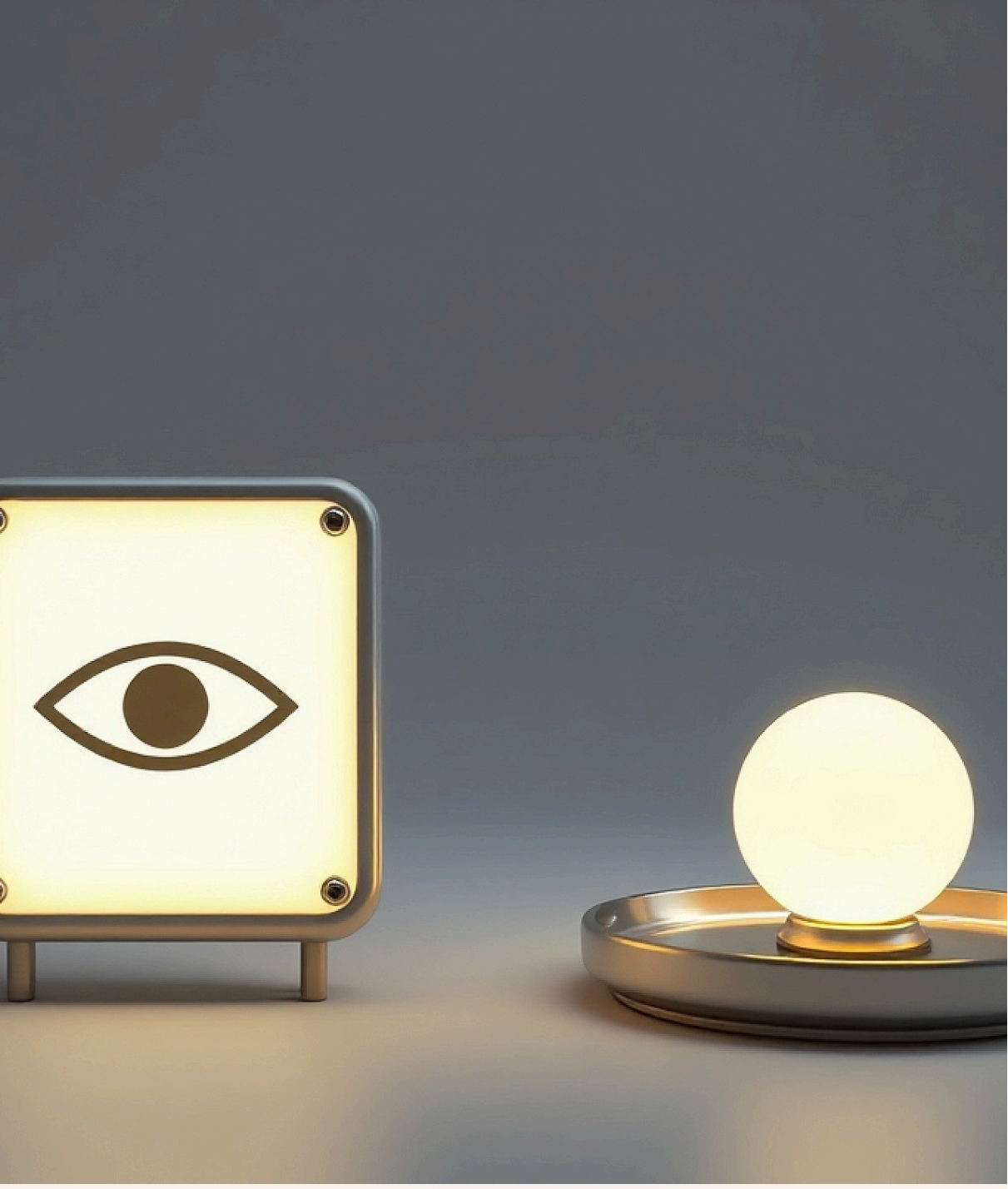
Returning a value makes a function reusable in many places. You can call the same function whenever you need its result. The result can be stored, passed on, or checked, as you have seen. This flexibility turns functions into building blocks. Instead of one-off actions, you get tools you can use again and again.



A common beginner mistake is expecting a printed number to be usable as a result. If a function only prints, the caller gets nothing back to use. For example, if add prints the sum but does not return it, you cannot store the sum in a variable. You must return the value to use it later.



Another mistake is calling a function that returns a value but never storing or using it. The value is handed back but then thrown away. For example, writing add(2, 3) without assigning it to a variable wastes the result. To keep the result, assign it to a variable or use it immediately. Otherwise, the returned value is lost.



Printing and returning serve different purposes. A function may print, return, or do both. You should know which one you need for your task. If you want to show a result to a person, print it. If you want to use the result in your program, return it. Confusing the two leads to errors.



Returning a value means giving the result back to the caller. Remember that the returned result can be stored in a variable, passed onward to another function, or checked in a condition. It is not the same as printing, which only shows text on the screen. Keep these ideas in mind as you write and use functions.