



BOOLEANS

Boolean represents a logical
value ‘true’ or ‘false’

CREATING BOOLEAN VALUES

We can create Boolean Values using
the literals ‘true’ or ‘false’



EVERYTHING WITHOUT A VALUE IS FALSE

Values like 0, and empty string (''), null, and undefined are considered falsy. Everything else is considered truthy.

BOOLEAN USING COMPARISON OPERATORS

Comparison operators (`==`, `!=`, `<`, `>`, `<=`, `>=`) compares two values and return a boolean result. They are often used to determine equality and relative values





BOOLEAN USING LOGICAL OPERATORS

Logical Operators (AND - &&, OR - ||, NOT - !) are used to combine and modify boolean values.



IF STATEMENT

If statement allows us to execute a block of code if the given condition is true.

IF ELSE STATEMENT

The if...else statement extends the basic if by providing an alternative block of code to execute when the condition is false.



ELSE IF STATEMENT

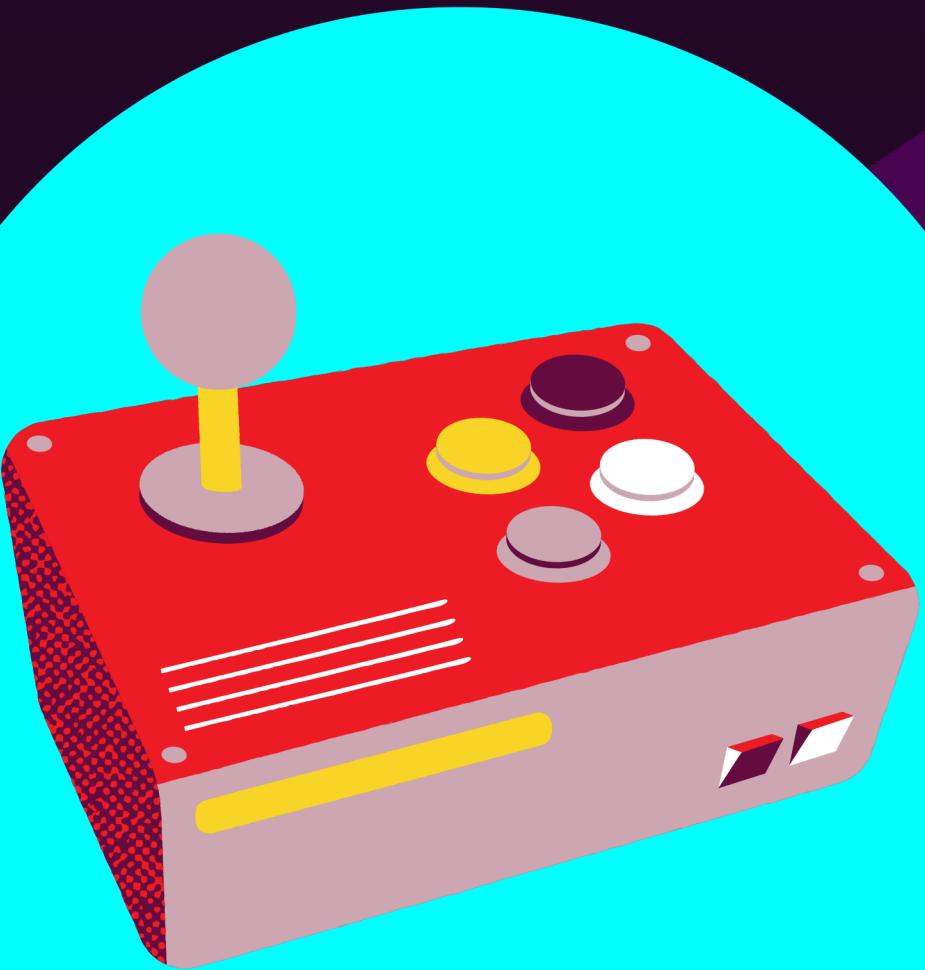
The else if statement allows us to have multiple possible outcomes. In else if statement, the first condition that evaluates to true from the top to bottom is the block that gets executed.



SWITCH

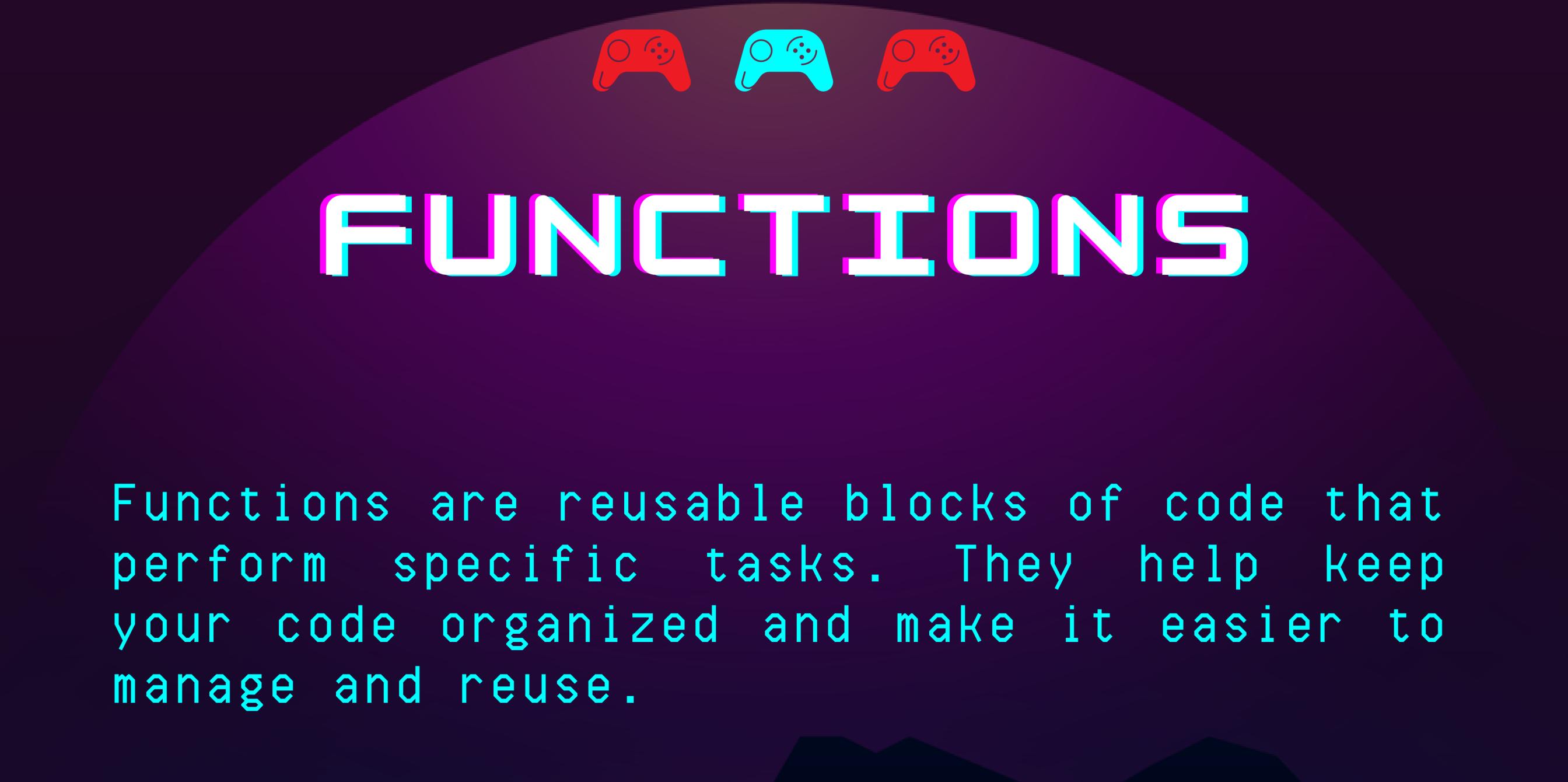
The switch statement is used for multi-way branching based on the value of an expression. It's an alternative to multiple if...else if... statements.





TERNARY OPERATOR

The ternary operator, also known as conditional operator, is a concise way to write conditional statements



FUNCTIONS

Functions are reusable blocks of code that perform specific tasks. They help keep your code organized and make it easier to manage and reuse.



PARAMETERS AND ARGUMENTS

Parameters are placeholders in a function definition, and arguments are the actual values you pass to the function when calling it. Parameters allow functions to be flexible and work with different inputs.



DEFAULT PARAMETER

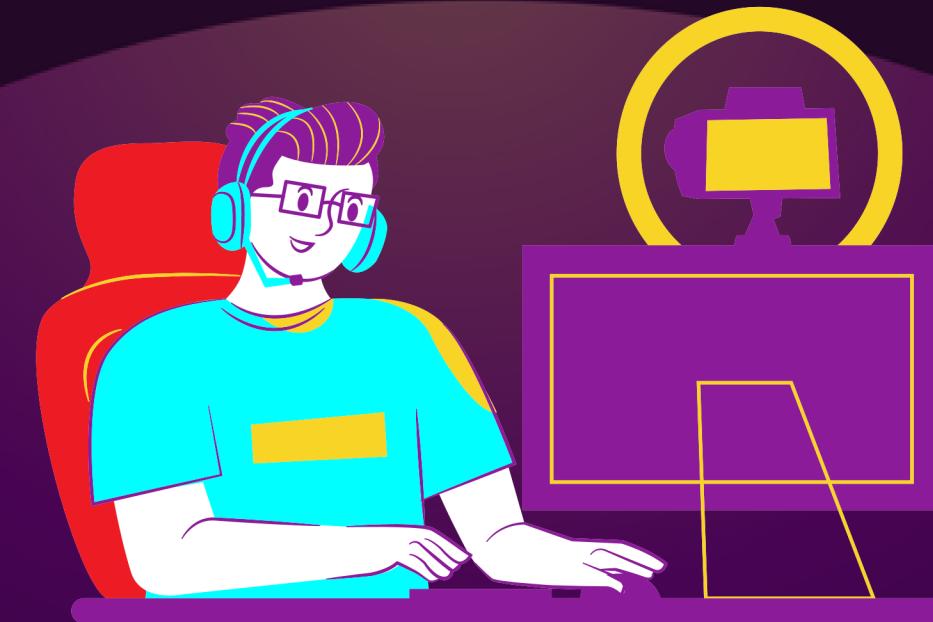
We can set default values for function parameters, making them optional to provide when calling the function.

RETURN

Functions can return values using the `return` statement. The returned value can then be used in other parts of the code.



STEP 04



FUNCTION SCOPE

Function Scope refers to the visibility and accessibility of variables defined within a function.



FUNCTION EXPRESSION

A function expression is a way of defining a function that involves creating a function as part of an expression. In a function expression, you don't give the function a name (anonymous function) or you can provide a name (named function expression).

ARROW FUNCTIONS

Arrow Functions remove the need to type out the keyword function every time you need to create a function. Instead, you first include the parameters inside the () and then add an arrow => that points to the function body surrounded in { }

