**JAVASCRIPT**

JavaScript is a programming language that lets you make web pages interactive. Imagine you're building a house. HTML is like the structure of the house, while CSS is like the paint and decoration. JavaScript, on the other hand, is like the electricity that powers everything and makes it work.

**1. Variables:** Think of variables as containers. They store different types of information, like numbers or words, that your program needs to remember and use later.

**2. Data Types**: There are different types of data in JavaScript. Numbers are for math, strings are for text, and booleans are for true/false values.

**3. Control Structures:** These are like directions telling your program what to do. "If" statements help your program make decisions based on conditions, while loops repeat tasks until a condition is met.

**4. Functions:** Functions are like mini-programs within your program. They take inputs (called parameters) and perform specific tasks. For example, you could have a function that adds two numbers together or changes the color of a webpage.

**5. Syntax:** Syntax is like the grammar of JavaScript. Just like you need to follow grammar rules to write sentences correctly, you need to follow JavaScript's syntax rules to write code that the computer can understand.

**6. Debugging:** Debugging means finding and fixing mistakes in your code. It's like being a detective, searching for clues to solve a problem in your program.

**7. Algorithmic Thinking:** This is about breaking down big problems into smaller, manageable steps. It's like following a recipe when you cook – you take one step at a time until you finish.

**8. Version Control:** Version control helps you keep track of changes you make to your code. It's like having a time machine for your code – you can go back to previous versions if something goes wrong.

**9. Software Development Life Cycle (SDLC):** This is the process of building software from start to finish. It's like following a roadmap – you plan, create, test, and deploy your code in different stages.

**10. Object-Oriented Programming (OOP):** OOP is a way of organizing your code into objects, which are like little bundles of data and functions. It helps you write code that's easier to understand and reuse.

**How to declare the variable**

**1. Declaration**: Variables in JavaScript are declared using the var, let, or const keywords.

**var:** Historically used for variable declaration in JavaScript. Variables declared with var are function-scoped or globally scoped.

**let**: Introduced in ECMAScript 6 (ES6). Variables declared with let have block scope, meaning they are limited to the block (enclosed by curly braces) in which they are defined.

**const:** Also introduced in ES6. Constants declared with const have block scope like let, but their values cannot be reassigned once they are initialized.

1. **Initialization**: Variables can be declared and initialized at the same time.

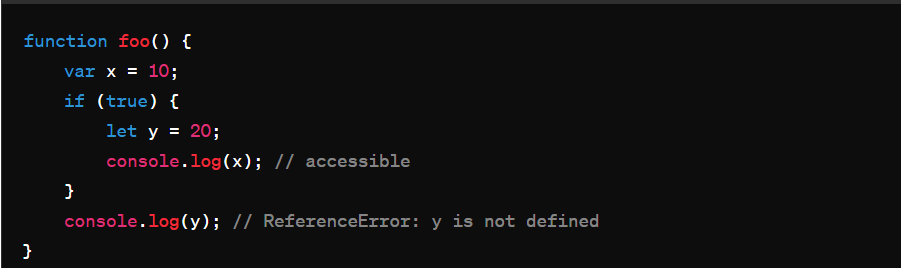
**Examples**:-



1. **Variable Dynamic Typing:** JavaScript is dynamically typed, meaning you don't have to specify the data type of a variable when declaring it. The type of the variable can change during the execution of the program.



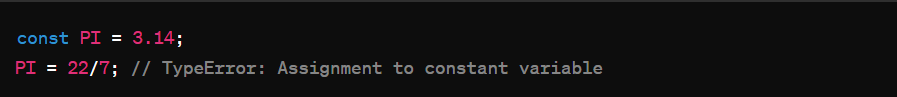
1. **Naming:** Variable names in JavaScript must begin with a letter (a-z, A-Z), underscore (\_), or dollar sign ($). They can also contain numbers (0-9), but cannot start with a number. Variable names are case-sensitive.
2. **Scope:** Variables in JavaScript have function-level scope (if declared with **var**) or block-level scope (if declared with **let** or **const**).



1. **Hoisting:** Variable declarations (not initialization) are hoisted to the top of their containing scope. This means you can access a variable before it's declared, but it will have an initial value of undefined.



1. **Global Variables**: Variables declared outside of any function or block have global scope and can be accessed from anywhere in the code.
2. **Constants**: Constants declared with **const** cannot be reassigned a new value once initialized.

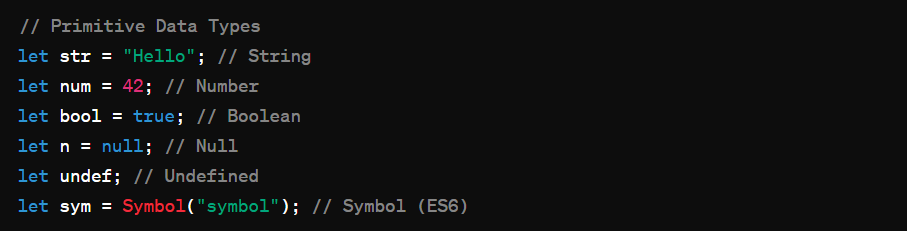


1. **Mutable vs. Immutable**: While variables declared with **var** and **let** are mutable (their values can be changed), constants declared with **const** are immutable (their values cannot be changed after initialization)

**How to store the value in variables and how many types of value stored in variable**

1. **JavaScript has 6 Datatypes (Primitive data types)**

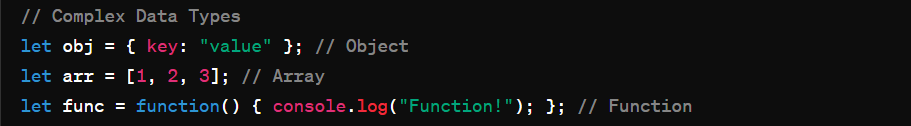
* **String:** Represents textual data, enclosed within single ('') or double ("") quotes.
* **Number:** Represents numeric data, including integers and floating-point numbers.
* **Boolean:** Represents a logical value, either true or false.
* **Null:** Represents the intentional absence of any object value.
* **Undefined**: Indicates that a variable has been declared but has not been assigned a value.
* **Symbol (ES6):** Represents a unique and immutable value, often used as property keys in objects.

****

1. **The Object Datatype (Secondary data types)**

**Complex Data Types:**

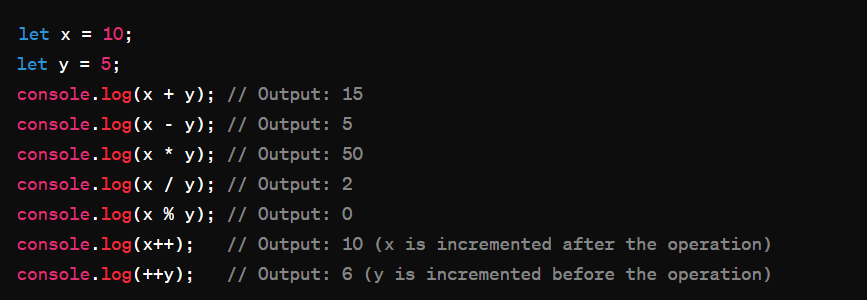
* **Object:** A collection of key-value pairs, also referred to as properties and methods.
* **Array:** An ordered list of values, accessible by numeric indices.
* **Function:** A reusable block of code that performs a specific task.

****

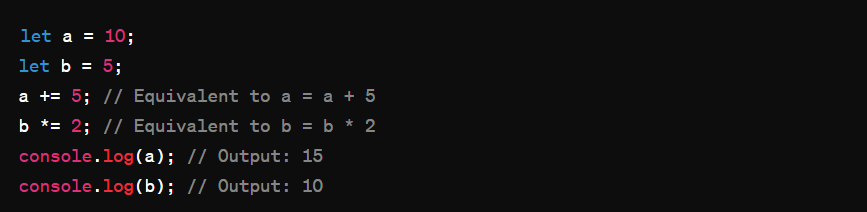
**Operators in JavaScript**

**JavaScript operators are used to perform different types of mathematical and logical computations.**

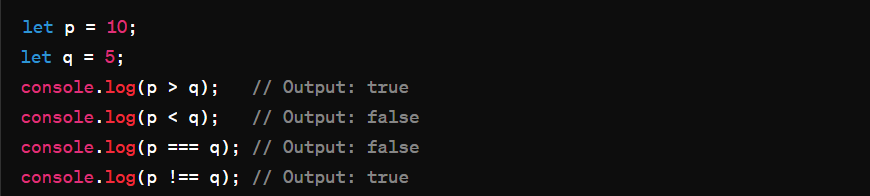
1. **Arithmetic Operators**
2. **Assignment Operators**
3. **Comparison Operators**
4. **Logical Operators**
5. **String Concatenation Operator**
6. **Arithmetic Operators:** Addition (+), Subtraction (-), Multiplication (\*), Division (/), Modulus (%), Increment (++), Decrement (--)



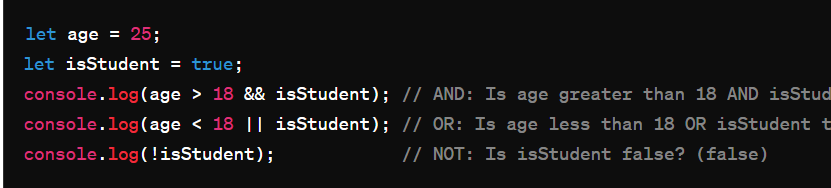
1. **Assignment Operators:** Assigns a value to a variable. It can also perform arithmetic operations with assignment.

****

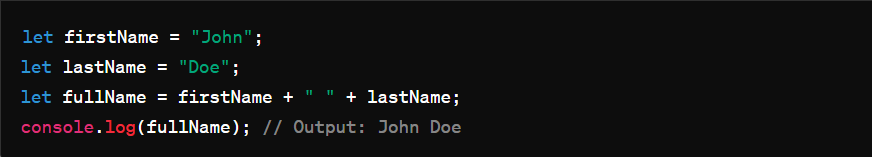
1. **Comparison Operators:** Compare two values and return a Boolean result (true or false).



1. **Logical Operators:** Used to combine conditional statements. (AND: &&, OR: ||, NOT: !).

****

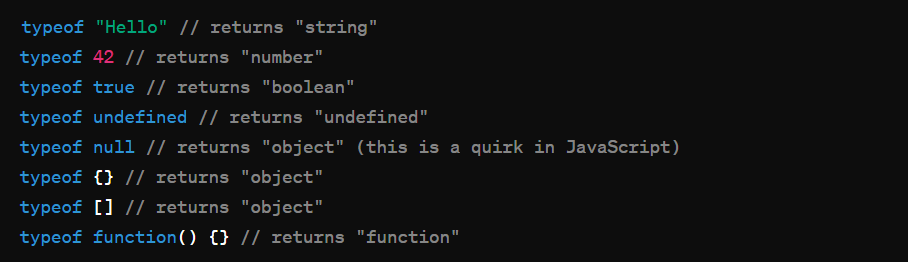
1. **String Concatenation Operator:** Used to concatenate two or more strings.

****

**How to know the type of operands:**

You can use the typeof operator to find the data type of a JavaScript variable.

In JavaScript, the typeof operator is used to determine the type of a variable or an expression. It returns a string indicating the type of the operand.

****

**Conditional Statements**

Conditional statements are like making decisions in programming. They help your code to behave differently based on certain conditions. The most common type of conditional statement in JavaScript is the "if" statement.

Here's how it works:

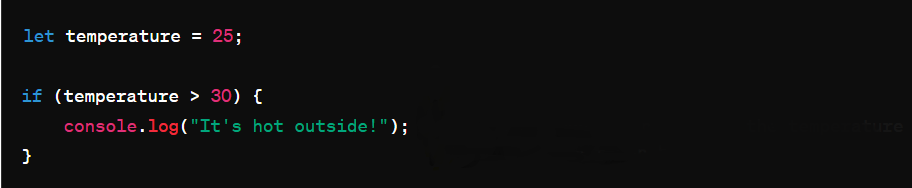
Sure, I'd be happy to explain conditional statements in JavaScript in simple terms.

Conditional statements are like making decisions in programming. They help your code to behave differently based on certain conditions. The most common type of conditional statement in JavaScript is the "if" statement.

Here's how it works:

1. **"if" Statement:**

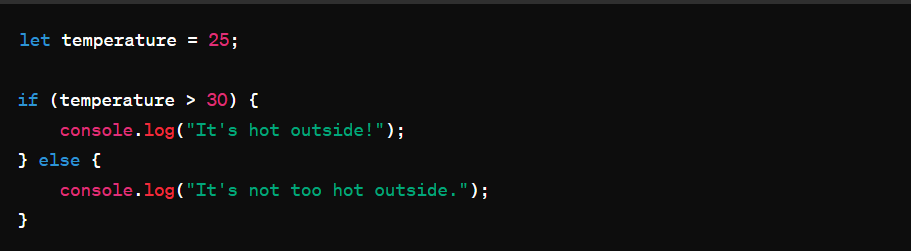
The "if" statement checks a condition. If the condition is true, it executes a block of code. If the condition is false, it skips that block of code.



This line won't run because the temperature is not greater than 30.

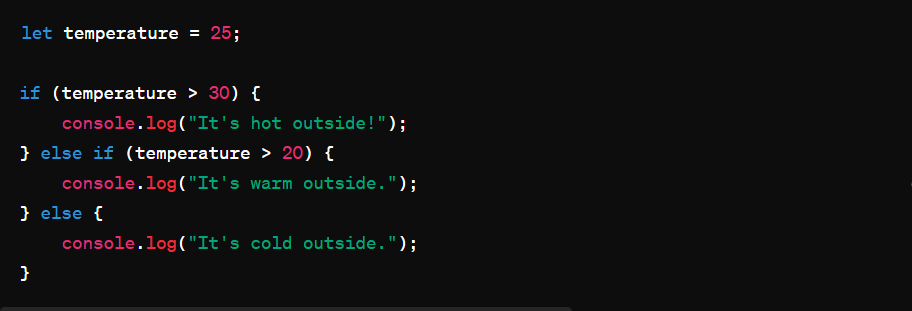
1. **"If-else" Statement:**

The "if-else" statement checks a condition. If the condition is true, it executes one block of code. If the condition is false, it executes another block of code



1. **"if-else if-else" Statement:**

This statement allows you to check multiple conditions. It starts with an "if" statement, followed by one or more "else if" statements, and ends with an "else" statement (optional).



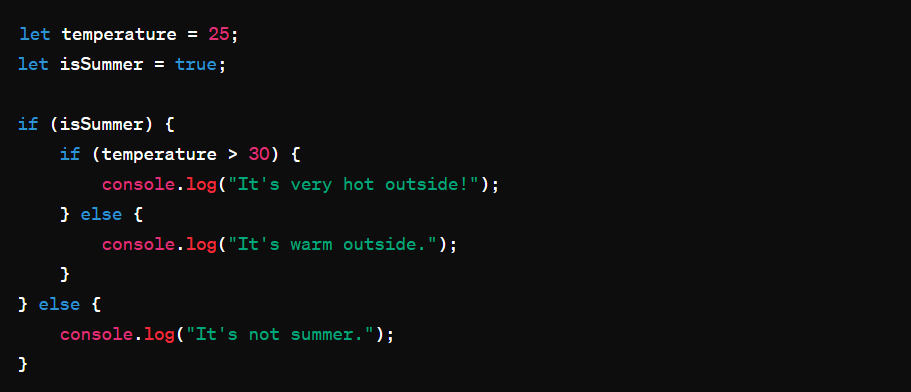
else if (temperature > 20) {

console.log("It's warm outside.");

// This line will be executed because the temperature is greater than 20 but not greater than 30.

1. **Nested Conditional Statements:**

You can also put conditional statements inside other conditional statements. This is called nesting.



let temperature = 25;

let isSummer = true;

if (isSummer) {

if (temperature > 30) {

console.log("It's very hot outside!");

} else {

console.log("It's warm outside.");

// This line will be executed because it's summer and the temperature is not greater than 30.

}

} else {

console.log("It's not summer.");

// This line won't be executed because the condition isSummer is true.

}

These are the basics of conditional statements in JavaScript. They allow your code to make decisions based on different conditions, making your programs more flexible and powerful.

**Loops**