

JAVASCRIPT

JavaScript is a programming language used to create dynamic content for websites.

In HTML, JavaScript code is inserted between `<script>` and `</script>` tags.

Scripts can be placed in the `<body>`, or in the `<head>` section of an HTML page, or in both.

Scripts can also be placed in external files.

JavaScript Can Change HTML Content

One of many JavaScript HTML methods is `getElementById()`.

`document.getElementById(id)` is a method in the Document Object Model (DOM) used in JavaScript to access an HTML element by its unique id attribute. The id attribute is a unique identifier assigned to an element within an HTML document.

We can use `document.getElementById(id)` to manipulate the element in various ways, such as changing its content, style, or attributes.

JavaScript can "display" data in different ways:

- 1) The `innerHTML` property defines the HTML content
- 2) For testing purposes, use `document.write()`. Writing into the HTML.
- 3) Use an alert box to display data. Writing into an alert box, using `window.alert()`.
- 4) For debugging purposes, you can call the `console.log()` method in the browser to display data.

JavaScript Values:

The JavaScript syntax defines two types of values:

- Fixed values
- Variable values

Fixed values are called **Literals**.

Variable values are called **Variables**.

JavaScript Literals

The two most important syntax rules for fixed values are:

1. **Numbers** are written with or without decimals:
2. **Strings** are text, written within double or single quotes:

JavaScript Variables

In a programming language, **variables** are used to **store** data values.

JavaScript uses the keywords var, let and const

to **declare** variables.

An **equal sign** is used to **assign values** to variables.

JavaScript Identifiers

All JavaScript **variables** must be **identified** with **unique names**.

These unique names are called **identifiers**.

Identifiers can be short names (like x and y) or more descriptive names (age, sum, totalVolume).

The general rules for constructing names for variables (unique identifiers) are:

- Names can contain letters, digits, underscores, and dollar signs.
- Names must begin with a letter.
- Names can also begin with \$ and _.
- Names are case sensitive (y and Y are different variables).
- Reserved words (like JavaScript keywords) cannot be used as names.

var : Variable can be re-declared & updated. A global scope variable.

let : Variable cannot be re-declared but can be updated. A block scope variable.

const : Variable cannot be re-declared or updated. A block scope variable.

Global Scope

- **Definition:** Variables declared outside of any function or block are said to have global scope. These variables can be accessed from anywhere in the code, both inside and outside of functions or blocks.

Block Scope

- **Definition:** Block scope refers to the scope of variables defined within a block, such as those declared with let or const inside a pair of curly braces {}. These variables are only accessible within the block they are defined in.

JavaScript Operators

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

Types of JavaScript Operators

There are different types of JavaScript operators:

- Arithmetic Operators
- Assignment Operators
- Comparison Operators
- String Operators
- Logical Operators
- Bitwise Operators
- Ternary Operators
- Type Operators

Arithmetic Operators

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
**	Exponentiation (ES2016)
/	Division
%	Modulus (Remainder)
++	Increment
--	Decrement

Assignment Operators

Operator	Example	Same As
=	x = y	x = y
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
%=	x %= y	x = x % y
**=	x **= y	x = x ** y

Comparison Operators

Operator	Description
==	equal to
===	equal value and equal type
!=	not equal
!==	not equal value or not equal type
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
?	ternary operator

JavaScript Data Types

- String
- Number
- BigInt
- Boolean
- Undefined
- Null
- Object

The **typeof** Operator

We can use the JavaScript **typeof** operator to find the type of a JavaScript variable.

The **typeof** operator returns the type of a variable or an expression