

Data Types & Variables in JavaScript

What are Data Types?

A data type defines the kind of value a variable can store and how it behaves in operations.

JavaScript has two categories of data types:

1 Primitive Data Types

These are immutable (cannot be changed) and stored by value.

Type	Example	Description
String	<code>"Hello"</code>	Represents text enclosed in quotes.
Number	<code>10, 3.14</code>	Stores integers and floating-point numbers.
Boolean	<code>true, false</code>	Represents logical values (true or false).
Undefined	<code>let x;</code>	Variable declared but not assigned any value.
Null	<code>let y = null;</code>	Represents an empty or non-existent value.
BigInt	<code>123n</code>	Used for very large numbers beyond <code>Number</code> limit.
Symbol	<code>Symbol('id')</code>	Unique identifiers, often used for object properties.

💡 Primitive values are immutable, meaning their actual value cannot be changed once created.

2 Non-Primitive (Reference) Data Types

These are mutable (can be changed) and stored by reference in memory.

Type	Example	Description
Object	<code>{ name: "JS" }</code>	Stores collections of properties and methods.
Array	<code>[1, 2, 3]</code>	A special type of object used for ordered lists.
Function	<code>function() {...}</code>	A block of reusable code stored as an object.

💡 Non-primitive values are mutable, meaning they can be changed after creation.

What are Variables?

A variable is a named container that stores data values.

Declaring Variables in JavaScript

JavaScript allows variables to be declared using `var`, `let`, or `const`.

Keyword	Scope	Reassignable?	Hoisted?	Use Case
var	Function-scoped	✔ Yes	✔ Yes	Avoid using (old way)
let	Block-scoped	✔ Yes	✘ No	Recommended for variables that change
const	Block-scoped	✘ No	✘ No	Use for constants

Understanding Scope

Function Scope: Variables declared with var are only accessible inside the function where they are defined.

Block Scope: Variables declared with let and const are accessible only within the {} block they are declared in.