

JavaScript Datatypes Demonstartion

- JavaScript is a dynamic type language, means you don't need to specify type of the variable because it is dynamically used by JavaScript engine.
- There are two types of data types in JavaScript.
 - Primitive data type
 - Non-primitive (reference) data type
- JavaScript has dynamic types. This means that the same variable can be used to hold different data types.

No	Data Type Name	Description	Example
10	Array	<ul style="list-style-type: none">Arrays are used to store multiple values in a single variable.They are a special type of object, designed for storing sequences of values.JavaScript arrays are written with square brackets.Array items are separated by commas.Array indexes are zero-based, which means the first item is [0], second is [1], and so on.They are a special type of object, designed for storing ordered collections of values, which can be of any data type.They can grow or shrink dynamically and can contain elements of different data types within the same array.	let colors = ["Red", "Green", "Blue"];

Array

This is an example 1 of Array Definition:

Eg: let colors = ["Red", "Green", "Blue"];

Array Result is : Red,Green,Blue and the type is : object

The Characteristics of the Array is in below diagram :

JavaScript arrays have several characteristics that make them versatile and widely used in programming. Here are some key characteristics of JavaScript arrays:

- Ordered Collection:** Arrays in JavaScript maintain the order of elements, meaning the elements are stored in a specific sequence and can be accessed by their index.
- Dynamic Size:** JavaScript arrays can dynamically grow or shrink in size. You can add or remove elements from an array without explicitly specifying its size.
- Mixed Data Types:** Arrays can contain elements of different data types. You can store numbers, strings, objects, functions, or even other arrays within the same array.
- Zero-based Indexing:** JavaScript arrays use zero-based indexing, meaning the index of the first element is '0', the index of the second element is '1', and so on.
- Length Property:** Arrays have a built-in 'length' property that indicates the number of elements in the array. This property can be used to iterate over the array or to dynamically manage its size.
- Mutability:** Elements in a JavaScript array can be modified, added, or removed after the array is created. This allows for flexible manipulation of data stored in the array.
- Heterogeneous:** JavaScript arrays can store elements of different types within the same array. For example, an array can contain a mix of strings, numbers, objects, and other arrays.
- Supports Iteration:** Arrays support various iteration methods such as 'for' loops, 'for...of' loops, 'forEach()' method, 'map()' method, 'filter()' method, etc., which makes it easy to iterate over the elements of an array and perform operations on them.
- Methods for Manipulation:** JavaScript arrays provide built-in methods for manipulation, such as adding elements ('push()', 'unshift()'), removing elements ('pop()', 'shift()'), slicing and splicing ('slice()', 'splice()'), concatenation ('concat()'), joining elements into a string ('join()'), etc.