

What are the data types in JS?

JavaScript has dynamic typing, meaning that data types are determined automatically during program execution. Here are some of the main data types in JavaScript:

- * **Primitive Types:**
 - String:** Used to represent textual information. Example: "Hello, World!".
 - Number:** Used to represent numeric values, either integer or floating-point. Example: 42 or 3.14.
 - Boolean:** Represents true or false values.
 - Null:** Represents the absence of a value or absence of an object.
 - Undefined:** Indicates that a value has not been assigned.
- * **String:** Used to represent textual information. Example: "Hello, World!".
- * **Number:** Used to represent numeric values, either integer or floating-point. Example: 42 or 3.14.
- * **Boolean:** Represents true or false values.
- * **Null:** Represents the absence of a value or absence of an object.
- * **Undefined:** Indicates that a value has not been assigned.
- * **Object Types:**
 - Object:** Used to represent complex data and structures. Objects in JS can have properties and methods. Example: `javascriptCopy codelet person = { name: "John", age: 30, isStudent: false };`
 - Array:** Used for an ordered list of values. Arrays in JS are indexed starting from 0. Example: `javascriptCopy codelet numbers = [1, 2, 3, 4, 5];`
 - Function:** Functions in JS are objects and can be assigned to variables, passed as arguments, and returned from other functions. Example: `javascriptCopy codefunction add(a, b) { return a + b; }`
- * **Object:** Used to represent complex data and structures. Objects in JS can have properties and methods. Example: `javascriptCopy codelet person = { name: "John", age: 30, isStudent: false };`
- * **Array:** Used for an ordered list of values. Arrays in JS are indexed starting from 0. Example: `javascriptCopy codelet numbers = [1, 2, 3, 4, 5];`
- * **Function:** Functions in JS are objects and can be assigned to variables, passed as arguments, and returned from other functions. Example: `javascriptCopy codefunction add(a, b) { return a + b; }`

* Special Types:Symbol: Introduced in ECMAScript 6, it is an immutable primitive that can be used as an identifier for object properties.BigInt: Introduced in ECMAScript 2020, this is a special type for representing arbitrarily large integers.

* Symbol: Introduced in ECMAScript 6, it is an immutable primitive that can be used as an identifier for object properties.

* BigInt: Introduced in ECMAScript 2020, this is a special type for representing arbitrarily large integers.

These types interact and complement each other during the execution of JavaScript programs.