**💡 What is JavaScript?**

**JavaScript** is a **programming language** used to make websites **interactive** and **dynamic**.

When you:

* Click a button and something happens ➡️ JavaScript
* Type in a form and get instant feedback ➡️ JavaScript
* See sliders, popups, drop-downs, or animations ➡️ JavaScript

**✅ Why Do We Use JavaScript?**

We use JavaScript to:

1. **Add interactivity** to websites (buttons, forms, sliders, etc.)
2. **Manipulate HTML & CSS** dynamically (change text, style, etc.)
3. **Communicate with servers** (send or receive data using APIs)
4. **Create games, animations, and single-page apps**
5. **Build full-stack applications** (with Node.js for backend)

**🌐 Where is JS used?**

* **Websites** (frontend)
* **Mobile apps** (React Native)
* **Desktop apps** (Electron.js)
* **Backend servers** (Node.js)

**🧠 Phase 1: JavaScript Fundamentals (Core Concepts)**

**✅ Topics to Learn:**

1. **Variables** (var, let, const)
2. **Data Types** (String, Number, Boolean, Null, Undefined, Object)
3. **Operators** (+, -, ==, ===, etc.)
4. **Conditionals** (if, else if, else, switch)
5. **Loops** (for, while, do while, for...of, for...in)
6. **Functions** (Regular & Arrow functions)
7. **Arrays & Objects** (push, pop, shift, unshift, map, filter, etc.)
8. **Events** (click, input, mouseover)
9. **DOM Manipulation** (getElementById, querySelector, innerHTML, etc.)
10. **Basic Form Validation**

------------------------------------------------------------------------------------------------------------------------------------

1. **Variables** (var, let, const):

JavaScript Variables can be declared in 4 ways:

* **Automatically**

A close-up of a math problem

AI-generated content may be incorrect.

* Using **var**

A screenshot of a math problem

AI-generated content may be incorrect.

* Using **let**

A screenshot of a math problem

AI-generated content may be incorrect.

* Using **const**

A screenshot of a math equation

AI-generated content may be incorrect.A screenshot of a math problem

AI-generated content may be incorrect.

A computer screen shot of a computer code

AI-generated content may be incorrect.

**\*\*Hoisting and Temporal Dead Zone (Important for Interviews)\*\***

**a. What is Hoisting?:**

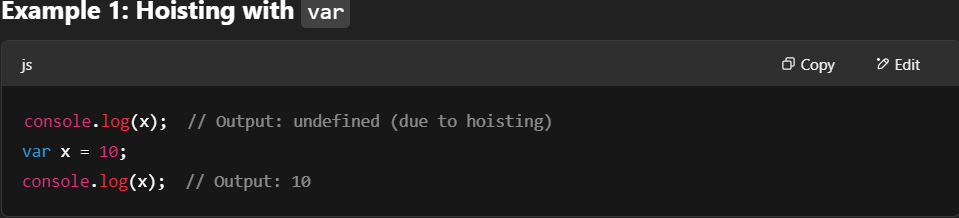
Hoisting is a JavaScript mechanism where variable and function declarations are moved to the top of their containing scope during the compile phase, before the code has been executed.

In simpler terms:

* JavaScript moves (or "hoists") the declarations of variables and functions to the top of their scope.
* However, only the declarations are hoisted, not the initializations.

**Hoisting with var**

* **var** declarations are hoisted to the top of the function or global scope, but the **initialization** (the value assignment) does **not** happen until the code reaches that point.
* Variables declared with var are **hoisted** and initialized to undefined.



**Explanation:**

* The declaration var x; is hoisted to the top of the scope.
* However, the initialization x = 10 happens at the point where the code reaches it.
* Therefore, when console.log(x) is called before initialization, it prints undefined because the variable is hoisted but not yet assigned a value.

**b. What is Temporal Dead Zone (TDZ)?**

Temporal Dead Zone (TDZ) is a term used to describe the time between the hoisting of a variable and the point where it's actually initialized (with let and const).

* TDZ applies to let and const variables.
* During this period, if you try to access a variable that’s hoisted but not yet initialized, JavaScript throws an error.

To summarize:

* Variables declared with var are hoisted and initialized with undefined, meaning they can be accessed even before their declaration, though with an undefined value.
* Variables declared with let and const are hoisted, but they stay in the TDZ until the line where they are initialized. Trying to access them before initialization leads to an error.

A screenshot of a computer error

AI-generated content may be incorrect.

**Explanation:**

* The declaration let x; is hoisted, but it stays in the **TDZ** until the initialization x = 10 is reached.
* When console.log(x) is executed, the variable x has not been initialized yet, so it throws an error (Cannot access 'x' before initialization).

A black screen with white text

AI-generated content may be incorrect.

**Explanation:**

* Similar to let, const is also hoisted, but it remains in the TDZ until it's initialized.
* Accessing country before it is initialized results in an error (Cannot access 'country' before initialization)

1. **Data Types**:

There are two types of data types, primitive and non-primitive

**i) Primitive Data Types**

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

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

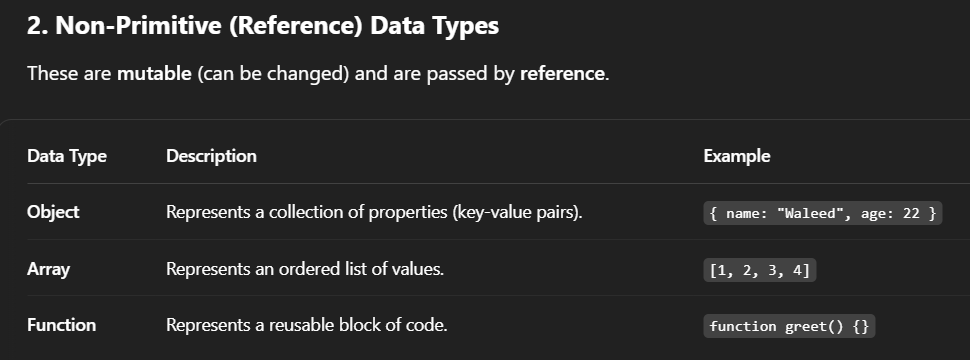
AI-generated content may be incorrect.A computer screen shot of a black background

AI-generated content may be incorrect. **Output:**

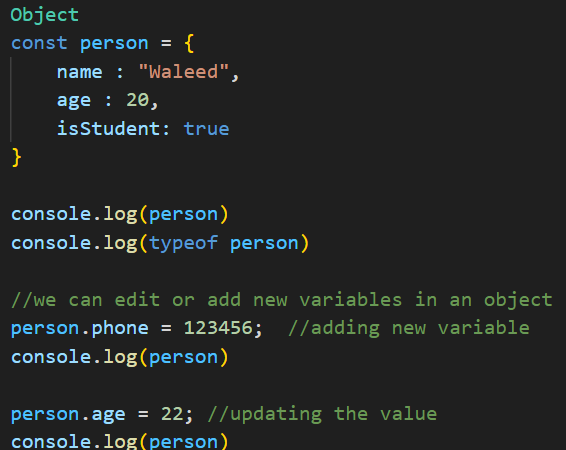
**Interview Question:**

**Why Data Type of null is object:** The null data type in JavaScript is classified as an object due to a historical design decision. This means that using the typeof operator on null returns "object", which can be confusing. Despite this classification, null is used to represent the intentional absence of any object value and should be treated as a distinct value indicating an empty or non-existent reference. This behavior is maintained for backward compatibility with the existing code.

**ii) Non-Primitive Data Types**

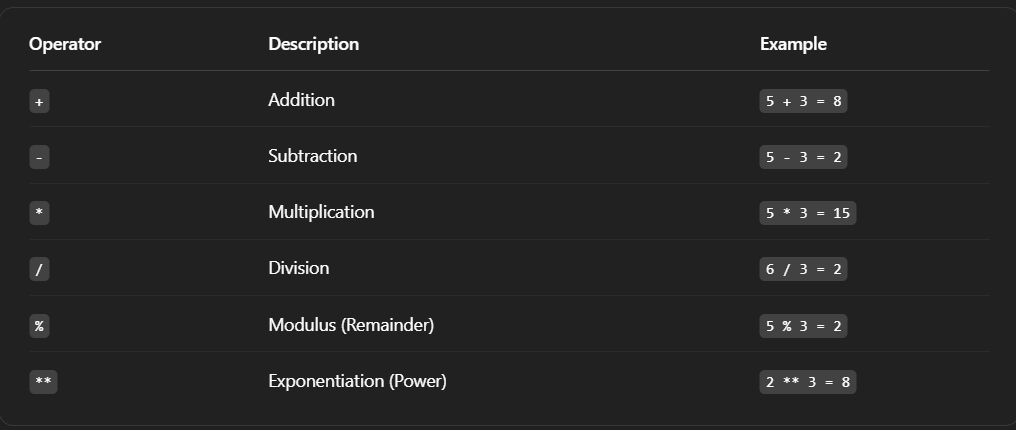


**Object**:

 A screen shot of a computer program

AI-generated content may be incorrect.

1. **Operators:**

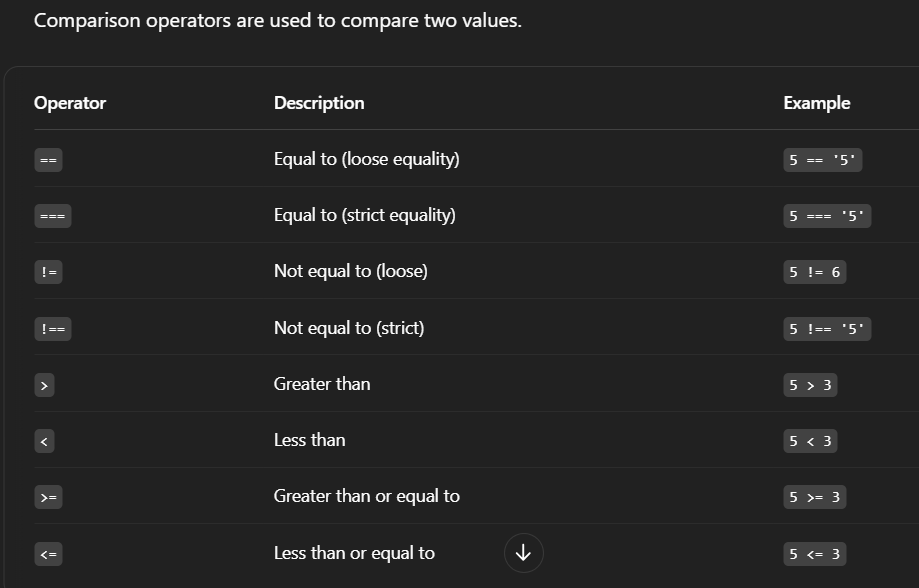


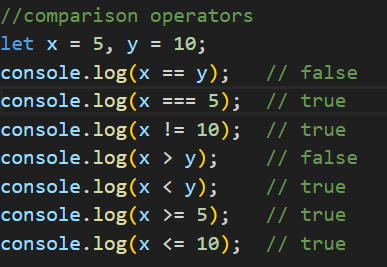
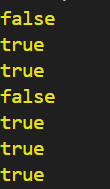
A screen shot of a computer program

AI-generated content may be incorrect.**Output**:A black background with yellow numbers

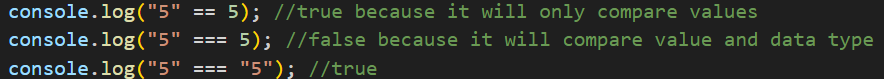
AI-generated content may be incorrect.

**Comparison Operators:**



Output: 

**Important Concept:**

**Output**:



1. **Functions**

A screen shot of a computer

AI-generated content may be incorrect.

Function with parameters:  
A computer screen with text

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

1. **Strings:**

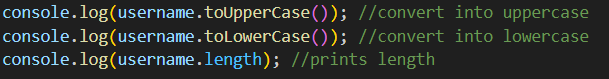
A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.Output: A black background with white letters

AI-generated content may be incorrect.

 A black background with white text

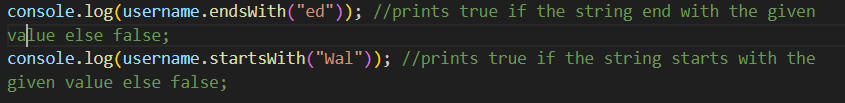
AI-generated content may be incorrect.



**Template Literal Backtricks:**

**A black screen with white text

AI-generated content may be incorrect.**

****

1. **Arrays:**

**A computer screen shot of text

AI-generated content may be incorrect. 🡺 A number on a black background

AI-generated content may be incorrect.**

**Console.log(num.pop(x))** and **console.log(num.push(x))** 🡪 will return the total length of the updated array

**A black screen with green and white text

AI-generated content may be incorrect.🡺 **

A close up of a screen

AI-generated content may be incorrect.🡺A number on a black background

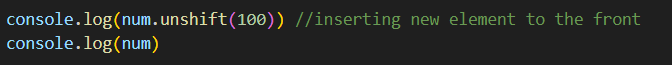
AI-generated content may be incorrect.

**Console.log(num.shift())** 🡪 will pop and print the first element for array

A group of colorful lines on a black background

AI-generated content may be incorrect. 🡺 

**Console.log(num.unshift(x))** 🡪 will print the total length after updation

 🡺 A number on a black background

AI-generated content may be incorrect.

**Concatenating two Arrays:**

A black background with white text

AI-generated content may be incorrect. 🡺 A number on a black background

AI-generated content may be incorrect.

**Reverse the array:**

 🡺 

**Sort the array:**

**🡺 **

**Array into strings:**

** 🡺 **

**For Loop in array:**

**A group of colorful text

AI-generated content may be incorrect.A screenshot of a computer screen

AI-generated content may be incorrect.**

**For Each Loop in array:**

**A black background with colorful text

AI-generated content may be incorrect.🡺 A number on a black background

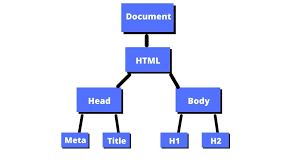
AI-generated content may be incorrect.**

**For Of Loop in array:**

**A black background with white text

AI-generated content may be incorrect. 🡺 A number on a black background

AI-generated content may be incorrect.**



1. **DOM**

**What is the DOM?**

DOM (Document Object Model) is a programming interface provided by the browser that allows JavaScript to interact with and manipulate HTML & CSS.

Think of the DOM as a tree structure (nodes) created from your HTML page.

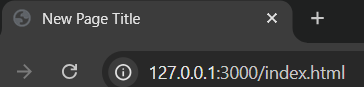
**🧠 Why Learn DOM?**

With DOM, you can:

* Access and change HTML elements.
* Add or remove elements dynamically.
* Respond to user input and events.
* Build interactive UIs.

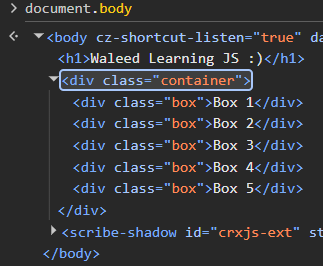
**🡪 Changing Page Title Dynamically:**

**document.title = "New Page Title";**



**🡪 Targeting body using DOM**

**A purple rectangular object with black text

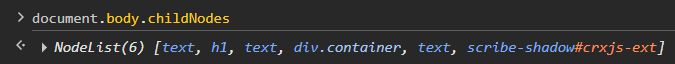
AI-generated content may be incorrect.**

**A black screen with yellow text

AI-generated content may be incorrect.🡪 Changing Background Color of Body:**

**🡪 Children, Parent and Sibling Nodes:**

****

****

**this will return first child even if it is not HTML element {text/comment}A black screen with yellow and white text

AI-generated content may be incorrect.**

**this will return on HTML elements placed on that positionA black screen with yellow and white text

AI-generated content may be incorrect.**

**Can also access any element using indexes**

**A black background with yellow and red text

AI-generated content may be incorrect. 🡪 A screenshot of a computer

AI-generated content may be incorrect.**

**Sub Children/Grand ChildrenA computer screen with text

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**First element of the parent**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Last Element of the parent**

**A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a box

AI-generated content may be incorrect.🡪 Styling Elements dynamically**

**A screenshot of a computer

AI-generated content may be incorrect.A black screen with yellow and green text

AI-generated content may be incorrect.**

**A screenshot of a box

AI-generated content may be incorrect.**

**🡪 Parent of an Element:**

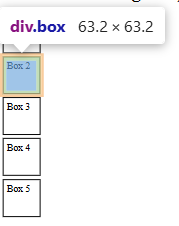
**A screen shot of a computer

AI-generated content may be incorrect.** ==============🡺

**A screenshot of a computer

AI-generated content may be incorrect.**

**🡪Previous Element Sibling of an element:**

** =========🡺**

**🡪Next Element Sibling of an element:**

** ===============🡺**

**🡺Targeting Elements Using Classes, ID, Tags  
  
🡪 Targeting Elements Using Classes:**

**A screenshot of a computer

AI-generated content may be incorrect.A black background with white text

AI-generated content may be incorrect.**



**🡪 Targeting Elements Using ID**



For Targeting Specific Elements:

HTML: JS:

A white and purple squares with black text

AI-generated content may be incorrect.A black screen with white text

AI-generated content may be incorrect.A screen shot of a computer code

AI-generated content may be incorrect.

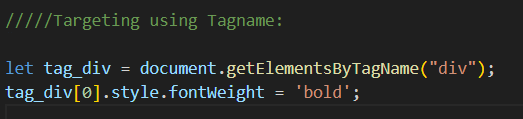


**🡪 Targeting Elements Using Tagname:**



Targeting wrt to tag name such as div, footer, header, etc…

A white square with black text

AI-generated content may be incorrect.****here it will target all boxes in container div because we have selected 0th index of tag div which is “container div”.

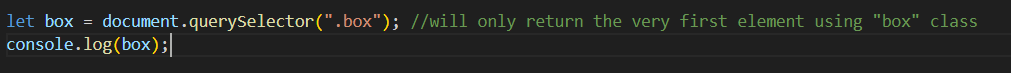
A close-up of a sign

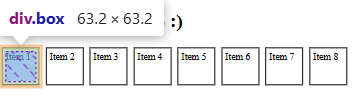
AI-generated content may be incorrect.A screen shot of a computer code

AI-generated content may be incorrect.here it will target all boxes in container div and we also have used for loop to style all of the boxes.

**🡪 Query Selector**

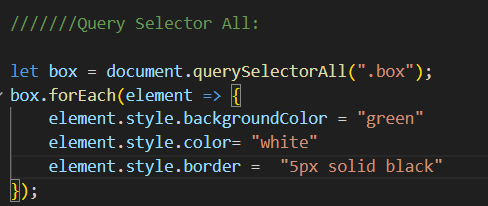
Will return the very first class using the respective class :





**🡪 Query Selector All**

Will target all elements using the class and return a node list / html list:



A green square with white text

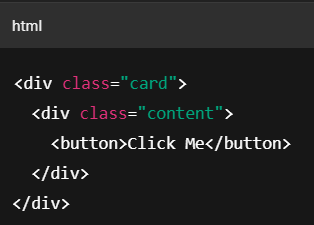
AI-generated content may be incorrect.

🡺 **closest(), matches(), and contains()**:

**🡪 .closest():**

Finds the **nearest ancestor (or self)** of an element that **matches a selector**.

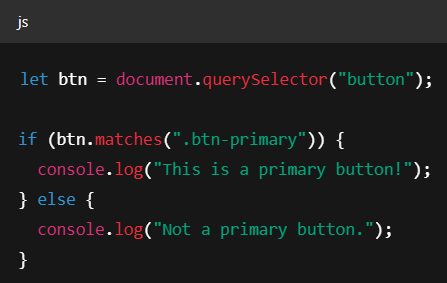
Used when you click an element and want to check if it's inside a certain container, or find its parent with a certain class.

 A screen shot of a computer

AI-generated content may be incorrect.

**🡪 .matches():**

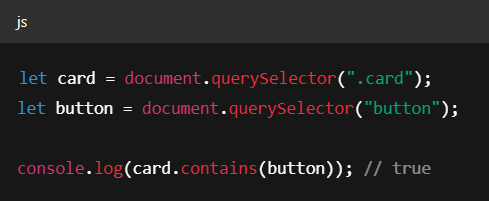
Checks if the element itself matches the given selector.  
Returns true or false.



**🡪 .contains():**

Checks if the parent element contains (or wraps around) the child element.

|| Returns true if the child is inside the parent (any depth), else false.

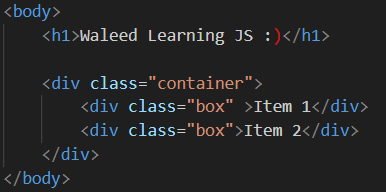


Used for checking DOM relationships, like:

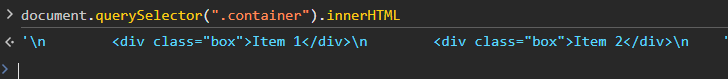
* Is a button inside a form?
* Is a div part of a section?

1. **Events :**

**🡪 Inner HTML:** Gets or sets the HTML content inside an element

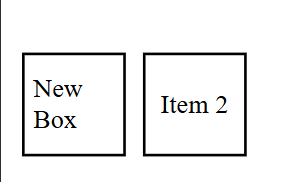
will select the very first element using .box class and then gets it inner HTML:





In this it will select the very first class having classname = container and then get its all of the inner HTML present in it.

‘\n’ – represent the blank space (the part highlighted with white color in the HTML code snippet)

****

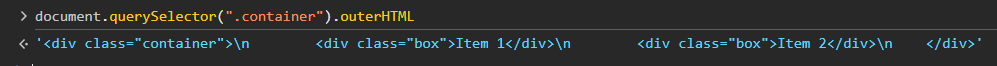
**Setting the Inner HTML:**

****

|  |
| --- |
| **🡪 Inner Text : Gets or sets the visible text inside the element (ignores hidden text)**    Setting Inner Text Is same as setting Inner HTML |

|  |
| --- |
|  |

**🡪 Outer HTML:** Gets or sets the entire element + its HTML content

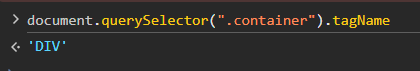


Setting it is also as same as Inner HTML/Text

**🡪 Text Content:** Gets or sets all the text content (even hiddden via css)



**🡪 Tag Name:** Will print the tag used in that element



**🡺 Attribute Methods:**

**🡪 getAttribute();**

Returns the value of specified attribute



**🡪 hasAttribute()**



**🡪 setAttribute()**

****

**🡪 removeAttribute()**



