

Advanced Java Script	Modern JavaScript (ES6): Javascript - Fetch API , Arrow functions, Template literals, Destructuring, Spread and rest operators
	Date and Time Objects, Closures, Promises and Async Programming, Closures and Scoping,
	DOM Manipulation: DOM Objects, Accessing elements in the DOM, Modifying HTML content, Adding and removing elements
	Functional Programming, Error Handling client side scripting
	JSON, JSON - Objects, arrays etc, JSON-Ajax Advanced DOM Manipulation,
	jQuery Plugins, Event Delegation, Ajax and Deferred Objects, Optimization and Performance

What is laziness ? hile learning something basics as HTML,CSS and Js
How to overcome laziness ?

Task	Outcome
Completing training module	Reducing the onboarding errors by 20% Score 10% more than first attempt Completing it with 100% efficiency
Preparing slides	Enabling client decision in first meeting Getting lesser doubts after presentation
Attending session	<ul style="list-style-type: none"> ● Building Project deployable skills within 30-45 days ● Creating a real use case based project for better visibility ● To have a clear pathway for career planning. ● Mastering professional skills for being resourceful.
finishing project documentation	preventing rework and client escalation

“This task exists so that _____ outcome is achieved.”

“Your manager doesn’t track effort.
They track risk, revenue, quality, and timelines”.

How to Kill Laziness (Actionable Pointers):

1. Replace all your “task” with “outcome”(What business problems does it solve ?)
2. Tie everything to a consequence : (No Consequence = No urgency)
 - a. What breaks, if I don't do this ?
 - b. Who gets blocked ?
 - c. What risk increases ?
3. Shrink my first step (Think in terms of a small actionable steps)
4. Work in time boxes (using 25-45 mins focus block)
 - a. Decide stop time before starting for better clarity.
 - b. Open - ended work invites procrastination.
 - c. Deadlines create energy.
5. Make Progress visible
 - a. Track :
 - i. What moved forward today ?
 - ii. What decision got easier ?
 - iii. What risk is reduced ?

visibility -> ownership -> action

JavaScript is a scripting language.

- To add logic, interactivity and dynamic **behavior**.

What JS does ?

It allows web pages to respond to user actions, process data and communicate with servers without reloading pages.

Key Responsible Areas : KRA

1. Enable real time validation
2. Dynamic UI updates
3. Interactivity user experiences

key features :

1. Interpreted & Light weight.

- a. Runs directly on browsers without compilation, enabling faster development cycles.

2. Dynamically types language:

- a. variables do not require fixed data types, making coding flexible but requires discipline.

3. Event - driven language : Executed codes based on some user action - button clicks, input, page loads events

4. Functions are First-class citizens:

- a. Functions can be stored, passed and returned, enabling modular and reusable code.

5. Built in support for Arrays and strings : hence simplifying UI logic for data handling and client side processing.

6. Cross- browser compatibility: - Supported by all modern browser making it an industry standard

7. Foundation of Moderns frameworks :

- a. It forms the base for react, Angular, Vue and Node.js

Note: Without Js Modern web application(forms, dashboards, single page Apps (SPA) would not be possible).

DEmo : Create a script that does following :

1. Accepts the user name and marks
2. Determine pass/fail using conditions
3. Store subjects in an array
4. Displays result using string methods.

Step 1: Declaring a variable & taking input from user

Step 2: Function creation

Step 3: Conditional logic

Step 4: Loping

Step 5: Arrays & String implementation

Pros: Easy to learn, Widely supported and flexible , faster to develop.

Cons: runtime errors, Loosely typed nature.

Simple calculator using HTML , CSS and Js:

index.html	style.css	script.js
Buttons Input field onClick()	input button .calculator	Read display value append clicked value

