# JS Assignment

## Introduction to JavaScript

1. Open any website → right-click → Inspect → Console → type: document.title
2. Try: alert(“Hello from Sheryians!”)
3. Write one line in your own words: “If HTML is skeleton, CSS is clothes — what is JS?”
4. Write one example of where you’ve seen JS being used (like popup, animation, etc.)

**Goal: Observe what JS does on a webpage.**

## Linking JavaScript Files

1. Create an HTML file and link JS file like this:
2. In app.js, write: console.log(“Connected!”)
3. Move the script tag to the  section and see what happens.
4. Try using defer:

```
<script defer src="script1.js"></script>
<script defer src="script2.js"></script>
```

**Goal: Understand how to connect JS with HTML.**

## Running JS in Browser Console

1. Open console and type: 2 + 2
2. Type: alert(“Hi”)
3. Try: prompt(“Your name?”)
4. Type: let city = “Bhopal”; city

## Variables and Keywords (var, let, const)

1. Declare your name using all three: var a = “Harsh”; let b = “Sheryians”; const c = “School”
2. Try reassigning them: a = “Updated”; b = “Updated”; c = “Updated”
3. Create a variable inside curly braces using let and log it outside.
4. Predict output before running.
5. Write 3 examples where const is useful (like PI, baseURL, etc.)

## Logging and Interaction (console, alert, prompt)

1. Log name, age, and city using console.log, console.info, console.warn.
2. Use prompt to take user’s name → alert a welcome message.
3. Log typeof of user’s input.
4. Try: let age = prompt(“Enter age:”); console.log(age + 5); observe what happens.

## Working with Strings

1. let msg = “I love Sheryians”;

2. Try msg.slice(2, 6) and predict the result.
3. Try msg.split(” “) and count words.
4. Try msg.replace(“love”, “study at”).
5. Template string example: let name = “Harsh”; console.log(Hey ${name}, welcome to JS!)
6. Check if msg.includes(“love”)

## Statements and Semicolons

1. Remove semicolon and check if code still runs.
2. Combine two statements in one line and see if it breaks.
3. Write 3 console.log statements without semicolons and predict output.

## Comments

1. Write your name as a single-line comment.
2. Write a 3-line comment explaining what your code does.
3. Hide one console.log using comment and check output.

## Expressions vs Statements

1. Type 5 + 10 (expression).
2. Type let x = 10; (statement).
3. Which one gives a value immediately?

4. Try: let y = (5 + 10) * 2; console.log(y)
5. Write one line explaining the difference between both.

## Data Types

1. let age = 25; let name = “Harsh”; let isStudent = true; let skills = [“JS”, “HTML”]; let user = { city: “Bhopal” }; let x = null; let y; let z = Symbol(“id”)
2. Log typeof each variable.
3. Change one value and recheck typeof.
4. Try adding a number and string together.

## Special Values

1. Log 1 / 0, 0 / 0, Number(“abc”), undefined + 1
2. Write what appears (Infinity, NaN, etc.)
3. Write one line explaining when to use null vs undefined.

## Primitive vs Reference

1. let x = 5; let y = x; y = 10; console.log(x, y)
2. let obj1 = { name: “Harsh” }; let obj2 = obj1; obj2.name = “Sheryians”; console.log(obj1.name)
3. Observe which one changes together.
4. Draw memory boxes on paper to visualize difference.