frameboxx 2.0®
animation | visual effects | programming | coding
Premier Academy for IT Development

+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

✅ Basic Tasks (Foundational understanding of variables)

1. **Declare a variable** using `let`, `const`, and `var`.
2. **Store your name** in a variable and log it to the console.
3. **Create two number variables** and output their sum.
4. **Swap two variables** using a third temporary variable.
5. **Store a boolean value** indicating if it's raining or not.
6. **Concatenate two strings** and store the result in a new variable.
7. **Use `typeof`** to log the type of various variables.
8. **Assign a value to a variable** without declaring it (and explain the consequence).
9. **Create a variable** to store your age and log "You are X years old."
10. **Reassign a variable** declared with `let` and log the change.

🟡 Intermediate Tasks (Working with logic and operations)

1. **Swap two variables without using a third variable**.
2. **Create a temperature converter** (Celsius to Fahrenheit) using variables.
3. **Create a program** that takes two variables and outputs the larger number.
4. **Check if a number is even or odd** using a variable.
5. **Create a calculator** that adds, subtracts, multiplies, and divides using two variables.
6. **Use `prompt()` to get user input**, store it in a variable, and output a message.
7. **Declare a variable inside a block**, and log it outside (demonstrate block scope).

+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

8. **Create a variable to hold a shopping cart total** and update it as items are added.
9. **Create a countdown timer** using a variable that decreases.
10. **Use template literals** to combine multiple variables in a sentence.

🔴 Advanced Tasks (Scope, hoisting, destructuring, and best practices)

1. **Demonstrate hoisting** with `var`, `let`, and `const`.
2. **Use destructuring** to extract values from an object into variables.
3. **Create a closure** that uses a variable to count how many times a function is called.
4. **Demonstrate variable shadowing** in a nested function.
5. **Create a function with default parameters** using variables.
6. **Use `const` to declare an array** and modify its contents (not reassignment).
7. **Implement a simple module pattern** using an IIFE and private variables.
8. **Track variable usage in a loop** and show how `let` vs `var` affects behavior.
9. **Dynamically generate variable names** using an object as a variable store.
10. **Create a real-world use case** (e.g., shopping cart or form handler) using multiple variables with proper naming, grouping, and updates.

+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

1. Identify Bad Variable Names

- Given a list of poorly named variables (e.g., `a`, `x1`, `temp2`), rewrite them with meaningful and descriptive names.

js

CopyEdit

```
// BAD:
let x = "John";
let y = 25;
let z = true;

// TASK: Rename them clearly and descriptively
```

---

2. Convert Snake Case to Camel Case

- Write code to convert variable names like `user_name` to `userName`.

js

CopyEdit

```
// Input: user_name
// Output: userName
```

---

3. Follow Camel Case for Variable Names

- Declare at least five variables using proper camelCase convention (e.g., `userAge`, `isLoggedIn`, `totalPrice`).

+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

4. Create Variables for a Real-World Object

- Create a set of variables to represent a car (`make`, `model`, `year`, `isElectric`, etc.), using clear and consistent naming.

---

5. Match Naming with Data Type

- Based on a variable's purpose, suggest an appropriate name.
  For example:
  - Boolean → `isAvailable`,
  - Array → `itemsList`,
  - Number → `userAge`

---

6. Rename Misleading Variables

- You are given variables like `list` (which is a string) and `isValid` (which holds a number). Rename them properly according to their content.

---

7. Use Prefixes for Boolean Variables

- Declare boolean variables for different features using prefixes like `is`, `has`, or `can` (e.g., `isActive`, `hasPermission`, `canEdit`).

---

8. Write Code Without Abbreviations

- Refactor code that uses abbreviations (e.g., `usr`, `cnt`, `tmp`) to use full, readable names (`user`, `count`, `temporary`).

+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

9. Consistent Naming in Loops and Functions

- Write a function that iterates over a list of products. Ensure all variable names (loop counters, function parameters, etc.) are consistent and meaningful.

---

10. Identify and Fix Case Conflicts

- Given variables like `UserName`, `userName`, `username`, and `user_name`, explain which one fits JavaScript conventions best and why. Then refactor to use only camelCase consistently.