DAY 3#

Question :- Implementing getAverageCaloriesByColor() function

Data :-

const inventory = [

{ name: "Banana", color: "Yellow", calories: 105 },

{ name: "Apple", color: "Red", calories: 95 },

{ name: "Lemon", color: "Yellow", calories: 37 },

{ name: "Strawberry", color: "Red", calories: 33 },

{ name: "Avocado", color: "Green", calories: 160 },

{ name: "Grapes", color: "Purple", calories: 62 },

{ name: "Kiwi", color: "Green", calories: 42 }

];

Answer :-

## **🔍 Calculating Average Calories by Color — The Power of JavaScript**

Recently, I built a simple yet effective fruit calorie calculator using nothing but **HTML, CSS, and JavaScript**—proving that powerful outcomes don't always need complex frameworks.

### **🧩 The Goal:**

To allow users to input a **fruit color** and get the **average calorie count** for all fruits of that color from a predefined inventory.

### **🛠️ Technical Breakdown:**

**Data Handling** A structured array of fruit objects:  
  
  
const Inventory = [

{ name: "Banana", color: "Yellow", calories: 105 },

{ name: "Apple", color: "Red", calories: 95 },

...

];

**Filtering Logic** Extract fruits by matching the input color:  
  
  
const filteredItems = Inventory.filter(item => item.color === color);

**Average Calculation** Total calories / number of filtered fruits:  
  
  
const totalCalories = filteredItems.reduce((acc, item) => acc + item.calories, 0);

const average = count > 0 ? totalCalories / count : 0;

**Event Binding & Output** On button click, the result updates dynamically in the DOM:  
  
  
document.querySelector('button').addEventListener('click', function() {

...

document.getElementById("answer").innerHTML = `The average calories for ${color} fruits is ${averageCalories}`;

});

1. **UI & UX** Styled using basic CSS for:  
   * A responsive table
   * Input field for color
   * Real-time result display

### **✅ Outcome:**

Users can instantly see the average calorie value of fruits by simply typing a color. No page reloads, no backend—just clean logic and fast results.

### **💡 Why This Matters:**

This project is a great example of how **core JavaScript** features like filter(), reduce(), and DOM manipulation can create meaningful, lightweight solutions.

#JavaScript #FrontendDevelopment #WebApp #VanillaJS #CodingProject #DOM