DOM Exercise Questions

# 1. Basic DOM Manipulation

Write a function that changes the background color of a <div> with the id 'colorBox' to a random color each time a button is clicked. Use JavaScript to:  
1. Select the <div> and button.  
2. Add a click event listener to the button.  
3. Change the background color of the <div> on each button click.

## Expected Output:

Each time the button is clicked, the <div>'s background color changes randomly.

# 2. Dynamic Content Creation

Write a JavaScript function that takes an array of strings as input and creates an unordered list (<ul>) with each string as a list item (<li>). Append the generated list to an empty <div> with the id 'listContainer'.  
Example:  
const items = ['Apple', 'Banana', 'Cherry'];

## Expected Output:

A <ul> with <li> items for each fruit in the 'items' array displayed inside the 'listContainer' <div>.

# 3. Event Delegation

Given a list of items generated dynamically, add an event listener using event delegation to log the text content of each clicked item. Use an empty <ul> element with the id 'itemList', then populate it with 10 <li> items in JavaScript, each labeled 'Item 1' through 'Item 10.' Use event delegation to handle clicks on any item.

## Expected Output:

When an <li> item is clicked, its content ('Item 1', 'Item 2', etc.) is logged in the console.

# 4. DOM Traversal

Write a function that takes an element ID as an argument and logs all child elements of that element. The function should also list each child's tag name and ID (if present). Use an HTML structure with nested <div> elements to test the function.  
Example HTML:  
<div id='parent'>  
 <div id='child1'>  
 <span id='grandchild1'>Hello</span>  
 </div>  
 <div id='child2'></div>  
</div>

## Expected Output:

Logs each child and grandchild, their tag names, and IDs.

# 5. Form Handling

Create a simple form with an input field for entering a name and a button to submit. When the button is clicked, display the name entered by the user in an alert box and clear the input field. Use JavaScript to handle the button click and to prevent the default form submission behavior.

## Expected Output:

The entered name is shown in an alert box, and the input field is cleared after the button is clicked.

# 6. Updating and Removing Elements

Write JavaScript code that does the following:  
1. Creates a list of five items and appends it to a <div> with the id 'mainList'.  
2. Adds a 'Remove' button next to each list item.  
3. When the 'Remove' button is clicked, the corresponding list item is deleted.

## Expected Output:

Each list item has a 'Remove' button that deletes the item from the list when clicked.

# 7. Interactive CSS Styles

Write a JavaScript function that adds a hover effect to all <p> elements within a <div> with the id 'textContainer'. When a <p> element is hovered over, change its font size to 20px and text color to blue; when the mouse leaves, revert to the original style.

## Expected Output:

Each <p> element changes style on hover, then reverts when the mouse is no longer over it.