

JavaScript Challenge 3 - Custom Context Menu

🔗 Objective

Your task is to create a custom context menu (right-click menu) using JavaScript, HTML, and CSS. The HTML document and video have been provided to demonstrate how the final solution should work. Your challenge is to write the JavaScript code to make the context menu functional.

Video Demonstration

[Click here to watch the video demonstration of how the final result should look and behave.](#)

Requirements

1. **Display Context Menu:** When the user right-clicks on the page, a custom context menu should appear at the cursor's location.
2. **Hide Context Menu:** If the user left-clicks anywhere else on the page or selects an option from the context menu, it should be hidden.
3. **Options:** The context menu should have at least two options. In this exercise, we'll include "View Details" and "Copy Text."
4. **Functionality:** Each option in the context menu should perform an action. For this exercise, you can use an alert to indicate that the option was clicked.

Instructions

1. Prevent Default Right-Click Menu:

- Inside the event listener, use `preventDefault()` to stop the default right-click menu from appearing.

2. Display Custom Menu:

- Inside the `contextmenu` event listener, use JavaScript to display the custom context menu.
- Utilize the event properties `e.pageX` and `e.pageY` to position the menu at the cursor's location.

3. Hide Custom Menu:

- Add a `click` event listener to the document.
- Inside the event listener, write the code to hide the custom context menu when clicked elsewhere.

4. Option Functionality:

- Attach click event listeners to the options in the context menu.
- Within each event listener, perform the required actions (e.g., display an alert).

Tips and Guidance

- Use the `addEventListener` method to efficiently handle various mouse events.
- Pay attention to how to obtain the cursor's position using `e.pageX` and `e.pageY` to place the context menu accurately.
- Feel free to explore and read more about the [contextmenu event](#) and [mouse events](#) on "w3schools" for further understanding.
- Practice good code organization by grouping related functionality together. This aids in readability and maintainability.
- Ensure to test your code across different browsers to check compatibility.
- Creativity: **Bonus points** for implementing functioning copying of selected text.

Note: This exercise aims to sharpen your skills in handling mouse events in JavaScript and manipulating the DOM.