## Context: You are an accessibility expert reviewing HTML for accessibility issues. Analyze the following code in the context of WCAG (Web Content Accessibility Guidelines) 2.2, listing any issues that may violate these guidelines. You will only be analyzing the HTML, ignoring any included or missing styles and scripts.

Code:

```<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="styles.css">

<script defer src="script.js"></script>

<title>Calvin Kelvin Products</title>

</head>

<body>

<!-- Navigation Bar -->

<nav>

<ul>

<li><a href="#">Home</a></li>

<li><a href="#">Products</a></li>

<li><a href="#">About Us</a></li>

<li><a href="#">Contact</a></li>

</ul>

</nav>

<!-- Header Section -->

<header>

<h1>Calvin Kelvin Products</h1>

<p>Your go-to destination for quality products.</p>

</header>

<!-- Featured Products Section -->

<section id="featured-products">

<!-- Product 1 -->

<div class="product-card">

<img src="product1.jpg" alt="Product 1">

<h2>ComfortFit T-Shirt</h2>

<p>Experience unmatched comfort with our premium cotton T-shirt.</p>

<button onmousedown="addToCart('ComfortFitTShirt')">Add to Cart</button>

</div>

<!-- Product 2 -->

<div class="product-card">

<img src="product2.jpg" alt="Product 2">

<h2>Classic Denim Jeans</h2>

<p>Stay stylish and comfortable with our classic denim jeans.</p>

<button onclick="addToCart('ClassicDenimJeans')">Add to Cart</button>

</div>

<!-- Add more products as needed -->

</section>

<!-- Video Section -->

<section id="video-container" onmouseout="PauseVideo()">

<h2>Our Story</h2>

<div id="video-container"></div>

</section>

<!-- Contact Section -->

<section id="contact">

<h2>Contact Us</h2>

<form>

<div class="form-group">

<input type="text" id="name" name="name" placeholder="Your name">

</div>

<div class="form-group">

<label for="email">Email:</label>

<label for="email">Email:</label>

<input type="email" id="email" name="email" placeholder="Your email">

</div>

<div class="form-group">

<label for="message">Message:</label>

<textarea id="message" name="message" placeholder="Your message"></textarea>

</div>

<button type="submit">Submit</button>

</form>

</section>

<!-- Footer Section -->

<footer>

<p>&copy; 2023 Calvin Kelvin Products</p>

</footer>

</body>

</html>

```

Response format:

Snippet with error: …

Location of error: …

WCAG Guideline violated: …

Detection Algorithm: …

Explanation of Error: …

### Snippet with error:

```html

<button onmousedown="addToCart('ComfortFitTShirt')">Add to Cart</button>

```

### Location of error:

In the "Featured Products Section", specifically within the product card for "ComfortFit T-Shirt".

### WCAG Guideline violated:

\*\*2.1.1 Keyboard\*\*: Ensure that all functionality of the content is available from a keyboard interface.

### Detection Algorithm:

The algorithm checks for interactive elements that rely solely on mouse events (`onmousedown`, `onclick`) without providing alternative keyboard event handlers (`onkeydown`, `onkeyup`).

### Explanation of Error:

The button for adding a product to the cart uses the `onmousedown` event handler, which means that users who cannot use a mouse (e.g., those using keyboards or assistive technologies) would not be able to interact with this button. This violates WCAG 2.1.1, which requires that all functionality should be accessible via a keyboard interface.

To fix this issue, add a `tabindex` attribute to the button to make it focusable via the keyboard and implement keyboard event handlers (`onkeydown`, `onkeyup`) to handle the addition to cart action. Here's how you could modify the button:

```html

<button tabindex="0" onkeydown="handleKeyDown(event)" onkeyup="handleKeyUp(event)">Add to Cart</button>

```

And then define the JavaScript functions `handleKeyDown` and `handleKeyUp` to manage the addition to cart logic based on keyboard input.

Citations:

## Are there any more issues? List all that you find.

Based on the provided code and the WCAG 2.2 guidelines, here are additional accessibility issues identified:

### Snippet with error:

```html

<label for="email">Email:</label>

<label for="email">Email:</label>

```

### Location of error:

In the "Contact Section", specifically within the form under the "Email:" label.

### WCAG Guideline violated:

\*\*3.3.2 Labels or Instructions\*\* (Level A): Properly associate labels with form controls.

### Detection Algorithm:

The algorithm checks for duplicate `for` attributes on `<label>` elements, indicating a potential misuse or misunderstanding of the `for` attribute's purpose.

### Explanation of Error:

The provided code contains two `<label>` elements with identical `for` attributes pointing to the same form control (`id="email"`). This is likely an oversight and does not serve the intended purpose of labeling form controls uniquely and accurately. Each form control should have a unique label associated with it to ensure that screen readers and other assistive technologies can properly convey the purpose of each form element to users.

To fix this issue, remove one of the duplicate `<label>` elements or ensure that each form control has a unique and descriptive label. Here's the corrected code snippet:

```html

<div class="form-group">

<label for="email">Email:</label>

<input type="email" id="email" name="email" placeholder="Your email">

</div>

```

### Additional Issue Identified:

#### Snippet with error:

```html

<section id="video-container" onmouseout="PauseVideo()">

<h2>Our Story</h2>

<div id="video-container"></div>

</section>

```

### Location of error:

In the "Video Section", specifically within the section designated for video content.

### WCAG Guideline violated:

\*\*2.4.7 Focus Visible\*\* (Level AA): Ensure that any focusable element can be reached and navigated using a keyboard.

### Detection Algorithm:

The algorithm checks for interactive elements that lack proper focus management, particularly in sections that might contain embedded media or interactive components.

### Explanation of Error:

The `onmouseout` event handler on the `<section>` element is used to pause a video, but there's no indication of how the video is embedded or managed within the `<div id="video-container">`. Without proper keyboard navigation support or visible focus indicators for the video player, users relying on keyboard navigation might face difficulties interacting with the video content.

To address this issue, ensure that the video player within the `#video-container` div is accessible via keyboard controls and that focus management is implemented correctly. This might involve adding keyboard event listeners to the video player and ensuring that it can receive focus and be controlled using the keyboard. Additionally, consider adding visual cues or styles to indicate focus state for improved accessibility.

Citations:

[1] https://webaim.org/standards/wcag/checklist

[2] https://www.w3.org/TR/WCAG22/

[3] https://www.digitala11y.com/wcag-checklist/

[4] https://www.levelaccess.com/blog/wcag-2-2-aa-summary-and-checklist-for-website-owners/

[5] https://www.wuhcag.com/wcag-checklist/

[6] https://www.w3.org/TR/2006/WD-WCAG20-20060427/appendixB.html

[7] https://adabook.medium.com/wcag-2-2-aa-guide-checklist-for-2021-web-accessibility-66c6fdaea034

[8] https://help.blackboard.com/Ally/Ally\_for\_WCM/Teacher\_Editor/Accessibility\_Report/Checklist

[9] https://accessible.org/wcag/

[10] https://userway.org/blog/ada-compliance-checklist/

## Are there any more issues? List all that you find.

After reviewing the provided code against the WCAG 2.2 guidelines and considering the insights from the sources, here are additional accessibility issues identified:

### Snippet with error:

```html

<script defer src="script.js"></script>

```

### Location of error:

At the end of the `<head>` section, just before the closing `</head>` tag.

### WCAG Guideline violated:

\*\*4.1.1 Parsing\*\* (Level A): In content implemented using markup languages, ensure that elements have complete start and end tags, are properly nested, that specify changes in semantics and/or theme do not occur in a way that would be confusing to a user.

### Detection Algorithm:

The algorithm checks for the use of the `defer` attribute in script tags without proper consideration of its implications on parsing and rendering.

### Explanation of Error:

The `defer` attribute in the `<script>` tag indicates that the script execution should be deferred until after the page has finished parsing. While this is generally useful for performance reasons, especially for external scripts, it can potentially introduce accessibility issues if not handled carefully. Specifically, if the script modifies the DOM or manipulates content in a way that affects accessibility features (like ARIA roles, properties, or states), it might cause confusion or accessibility problems for users relying on assistive technologies.

To mitigate this issue, ensure that any script using the `defer` attribute does not alter the DOM in a way that would negatively affect accessibility. Additionally, consider providing fallback content or ensuring that the script's modifications are accessible even if the script fails to load or execute. This might involve checking for script errors and providing alternative content or functionality as a safeguard.

### Additional Issue Identified:

#### Snippet with error:

```html

<img src="product1.jpg" alt="Product 1">

```

### Location of error:

Within the "Featured Products Section", specifically in the product card for "ComfortFit T-Shirt".

### WCAG Guideline violated:

\*\*1.1.1 Non-text Content\*\* (Level A): All non-text content that is presented to the user has a text alternative that senses a purpose of the non-text content.

### Detection Algorithm:

The algorithm checks for images without sufficient alternative text descriptions.

### Explanation of Error:

The image lacks a detailed description that conveys the purpose or function of the image to users who cannot see it. While the current `alt` text provides a basic description ("Product 1"), it does not offer enough detail about the product itself, such as its features, benefits, or how it relates to the surrounding content.

To address this issue, enhance the `alt` attribute with a more descriptive text that accurately represents the image content. This helps users with visual impairments understand the context and significance of the image. For example:

```html

<img src="product1.jpg" alt="A high-quality ComfortFit T-Shirt made from premium cotton fabric, featuring a modern design and available in various sizes.">

```

This revised `alt` text provides a clearer picture of what the image represents, aiding in accessibility for users relying on screen readers or other assistive technologies.

Citations:

[1] https://www.w3.org/TR/WCAG22/

[2] https://www.w3.org/WAI/WCAG22/Techniques/

[3] https://www.accessibility.works/blog/wcag-2-2-guide/

[4] https://github.com/w3c/wcag/issues/2820

[5] https://www.milespartnership.com/how-we-think/article/tackling-most-common-web-accessibility-issues-2024

[6] https://accessibility.psu.edu/guidelines/wcaglist/

[7] https://tetralogical.com/blog/2023/10/05/whats-new-wcag-2.2/

[8] https://adasitecompliance.com/wcag-2-2-decoding-latest-web-accessibility-guidelines/

[9] https://dequeuniversity.com/resources/wcag-2.2/

[10] https://www.w3.org/WAI/WCAG22/Understanding/consistent-identification.html