JavaScript Challenge 2 - Character Carousel

Objective:

Create a character carousel that rotates characters when clicked, providing a smooth transition animation. Each character's position, scale, and styling should change accordingly.

Resources:

- HTML and CSS code (provided).
- Demo video of the solution

Instructions:

1. **Initialize Characters and Positions:**

   - Create a constant that holds all the characters inside the carousel.
   - Define an array called `positions` to hold the initial order of the characters (e.g. `[0, 1, 2]`).

2. **Create a Function to Update the Carousel:** Create an `updateCarousel` function that is responsible for updating the positions, scaling, and z-index of the characters within the carousel based on their current position in the `positions` array.

   Here is a detailed breakdown of how this function should operate:

   1. **Iterate Through Characters:**

      - Use a `forEach` loop to iterate through each character in the `characters` collection.
      - Inside the loop, you'll work with each character's position and apply corresponding transformations.
      - Pass to the `forEach` loop the character and also the index. You'le need it later.

   2. **Retrieve Current Position:**

    - Store in a variable the current position of the character in the carousel using `positions[index];`.

3. **Apply Transformations:**

    - Utilize the `transform` CSS property to apply translation and scaling based on the character's position.
    - Translate the character horizontally using `translateX`. Use a calculation with `(pos - 1) * 180` to set the correct translation based on the position.
    - Apply scaling for the central character using a ternary based on if the `pos === 1`. If the position is 1, the character is in the center, and apply a scale of 1.2. If not, apply `''`.

4. **Adjust z-index and Additional Styling:**

    - Use an `if` conditional statement to check if the character is in the center (position 1).
    - If it's the central character, set the `z-index` to 2, making it appear above the other characters. If not, set the `z-index` to 1.
    - Add or remove the class `character-center` to apply additional styling to the central character.

5. **Transition Effects:**

    - The transition effect is defined in the CSS with a 1-second duration. This ensures a smooth animation when the transformations are applied.

Tips and Challenges:

- Experiment with different translation and scaling values to understand how they affect the appearance and behavior of the carousel.
- Make sure to call the `updateCarousel` function whenever the `positions` array is modified.
- Consider adding more effects or animations to enhance the user experience.
- Debugging with browser developer tools will allow you to inspect elements and see how transformations are applied in real time.

---

3. **Handle Character Clicks:**

- Use a `forEach` loop to add click event listeners to each character. Pass the character and aslo the `index` to this loop.
- If the clicked character is in the center, exit the function. Think of what is the conditon you need to check for this.
- Determine the clicked position and how many times you need to rotate the `positions` array (either once or twice).
- Use a loop and `positions.push(positions.shift())` in each iteration, to rotate the positions array.
- After exiting the loop, call the `updateCarousel` function to reflect these changes.

4. **Finalize the Carousel:**
    - Call the `updateCarousel` function initially to set up the carousel when the page loads.

Challenges and Tips:

- Be mindful of the transformations and how they interact with each other. The translate and scale functions should work together smoothly.
- Use the browser's developer tools to debug the transformations and alignments.
- Understanding CSS Transitions may further help you grasp the smooth transition effect.

Evaluation Criteria:

- Functionality: Carousel works as demonstrated in the video.
- Code Quality: Code is well-organized, efficient, and follows best practices.
- Creativity: Bonus points for any additional creative features or improvements.