

Exercise 6 - Scroll-Activated Animations

🔗 Objective

In this exercise, you will work with JavaScript to create an engaging scroll-activated animation effect. As the user scrolls down the page, specific elements will animate into place, providing a dynamic and interactive user experience. You will practice utilizing the `addEventListener` method along with scroll events, DOM manipulation, and transitions.

HTML and CSS Provided

The HTML and CSS code is already provided to you. You can find the structure and styling in the provided HTML file. A video demonstrating how the final solution should work can be viewed [here](#).

Instructions

1. **Obtain the Text Elements:** Start by selecting all the text elements you want to animate.
2. **Create Scroll Event Listener:** You'll need to set up an event listener for the scroll event on the window object. Inside the event listener, you'll write the logic to determine if an element is in view.
3. **Determine Element Position:** Inside the scroll event, use the `getBoundingClientRect` method to determine the position of the text elements within the viewport. You may use the `top` property of the result to get the distance from the top of the viewport.
4. **Apply Activation Logic:** Determine if the text element is within the desired range of the viewport. If it is, add a class that activates the animation (in the provided HTML/CSS, this class is called "active").
5. **Test the Solution:** Test your solution to ensure that the text elements animate into place as you scroll down the page. They should smoothly transition as described in the demonstration video.
6. **Extra Challenge:** If you want an extra challenge, try adding different animations or conditions to the elements as they come into view. You can also try to add the behaviour

that when the text element is out of the viewport, it animates back to the initial state.

Resources

- [MDN - Window scroll event](#)
- [MDN - Element.getBoundingClientRect](#)

Tips

- Pay attention to the relationship between the window's scroll position and the element's position within the viewport.
- Experiment with different threshold values to trigger the animation at just the right moment.
- Ensure that the class you are adding matches the one defined in the CSS for the desired animation effect.