

Memoria T3Actividad5

Ejercicio 5: Generación de un círculo anidado con promesa

Enunciado

El quinto ejercicio consiste en crear un círculo en la pantalla que vaya incrementando su tamaño de forma gradual hasta alcanzar una dimensión específica. Al finalizar la animación, se debe mostrar un mensaje dentro del círculo con el texto "Hola, mundo". La funcionalidad debe activarse mediante un evento de botón, y la implementación debe utilizar promesas para gestionar la animación.

Análisis del Problema

Para este ejercicio, se requiere crear una animación de crecimiento de un círculo, implementando la lógica con JavaScript. La animación se controlará utilizando una promesa para manejar el tiempo de ejecución y asegurar que el mensaje aparezca cuando la animación haya terminado. La estructura del código debe permitir que la animación sea fluida y que el mensaje se muestre en el momento adecuado.

Solución Propuesta

La solución consta de los siguientes pasos:

1. **Interfaz HTML:** Se crea una estructura HTML simple con un botón que activa la animación del círculo al hacer clic.
2. **Estilos CSS:** Se definen los estilos para el círculo y el mensaje, así como las transiciones de la animación.
3. **Lógica en JavaScript:**
 - Una función `showCircle()` crea el círculo y lo anima para que crezca hasta alcanzar el tamaño especificado.
 - La función retorna una promesa que se resuelve cuando la animación termina.
 - La función `go()` maneja el evento de clic del botón y utiliza la promesa para mostrar el mensaje "Hola, mundo" dentro del círculo al finalizar la animación.

Código Fuente

HTML

El código HTML se encuentra en el archivo `T3Actividad5.html` del proyecto.

JavaScript

El código JavaScript está adjunto en el archivo `T3Actividad5.js` del proyecto.

Explicación del Código

1. **Interfaz HTML:**
 - La estructura HTML incluye un botón que activa la animación del círculo. Cuando se hace clic en el botón, se llama a la función `go()` para iniciar el proceso.
 - Un archivo CSS en línea define los estilos necesarios para la animación del círculo y el mensaje.
2. **Estilos CSS:**
 - Se utiliza la clase `.circle` para definir el aspecto del círculo, incluyendo propiedades como el color de fondo, el radio de la esquina (para que sea redondo), y las transiciones para animar el cambio de tamaño.
 - La clase `.message-ball` se aplica para centrar el texto dentro del círculo y ajustar la fuente del mensaje.
3. **Función `go()`:**
 - Esta función se ejecuta al hacer clic en el botón. Llama a la función `showCircle()` pasándole las coordenadas del centro del círculo y su radio inicial.
 - Una vez que la promesa se resuelve (cuando la animación ha terminado), se añade el texto "Hola, mundo" al círculo.
4. **Función `showCircle(cx, cy, radius)`:**
 - La función crea un nuevo elemento `<div>` que representa el círculo, y lo posiciona en la página utilizando las coordenadas `cx` y `cy`.
 - Inicialmente, el tamaño del círculo es cero, y se expande a través de una transición CSS hasta alcanzar el tamaño especificado (doble del radio).
 - Se utiliza `setTimeout` para iniciar la transición de crecimiento. Cuando la animación de la transición termina, el evento `transitionend` resuelve la promesa, pasando el elemento del círculo (`div`) para su posterior manipulación.

Pruebas Realizadas

Se realizaron diversas pruebas para garantizar el correcto funcionamiento del programa:

1. **Crecimiento del círculo:** Se verificó que el círculo se expandiera de forma gradual hasta alcanzar el tamaño especificado.
 2. **Aparición del mensaje:** Se comprobó que el mensaje "Hola, mundo" aparecía dentro del círculo solo después de que la animación había terminado.
 3. **Respuesta al clic del botón:** Se probó el comportamiento del botón para asegurar que la animación se iniciara correctamente al hacer clic.
-

Conclusión

La implementación cumple con los requisitos del enunciado, utilizando promesas para controlar la secuencia de la animación del círculo y mostrar el mensaje una vez finalizada. La combinación de HTML, CSS y JavaScript asegura una animación fluida y una experiencia de usuario satisfactoria.
