

Apuntes para Examen: Integración de Contenido Interactivo y Animaciones en JavaScript

1. Temporizadores en JavaScript

setTimeout y clearTimeout

- **setTimeout(función, tiempo)** ejecuta una función tras un tiempo en milisegundos.
- **clearTimeout(id)** cancela un **setTimeout** activo.

Ejemplo:

```
let id = setTimeout(() => alert("Hola después de 3 segundos"), 3000);  
clearTimeout(id); // Cancela el timeout
```

setInterval y clearInterval

- **setInterval(función, tiempo)** ejecuta repetidamente una función cada cierto intervalo.
- **clearInterval(id)** detiene un **setInterval** activo.

Ejemplo:

```
let contador = 0;  
let id = setInterval(() => {  
  console.log("Contador:", ++contador);  
  if (contador === 5) clearInterval(id);  
}, 1000);
```

2. Animaciones eficientes

requestAnimationFrame y cancelAnimationFrame

- **requestAnimationFrame(función)** ejecuta animaciones sincronizadas con la tasa de refresco del navegador.
- **cancelAnimationFrame(id)** cancela la animación.

Ejemplo:

```
let start = null;
let element = document.getElementById("animado");

function mover(timestamp) {
  if (!start) start = timestamp;
  let progreso = timestamp - start;
  element.style.transform = `translateX(${Math.min(progreso / 10, 200)}px)`;
  if (progreso < 2000) requestAnimationFrame(mover);
}

requestAnimationFrame(mover);
```

3. Transiciones CSS

- Permiten animar cambios de propiedades CSS.

Ejemplo:

```
<div class="caja"></div>
<style>
.caja {
  width: 100px;
  height: 100px;
  background-color: red;
  transition: width 2s ease, height 4s ease-in;
}
.caja:hover {
  width: 200px;
  height: 200px;
}
</style>
```

4. Promesas en JavaScript

Creación y encadenamiento con **then** y **catch**

- Las promesas gestionan operaciones asíncronas.

Ejemplo:

```
let promesa = new Promise((resolve, reject) => {
  setTimeout(() => resolve("¡Éxito!"), 2000);
});
```

```
promesa
  .then(resultado => console.log(resultado))
  .catch(error => console.error(error));
```

Promise.all y Promise.race

- **Promise.all([])** espera que todas las promesas se resuelvan.
- **Promise.race([])** devuelve la primera promesa resuelta o rechazada.

Ejemplo:

```
let p1 = new Promise(res => setTimeout(res, 4000, "Una"));
let p2 = new Promise(res => setTimeout(res, 1000, "Dos"));
```

```
Promise.race([p1, p2])
  .then(valor => console.log("Primera resuelta:", valor));
```

5. Async/Await

- Simplifica el uso de promesas haciendo el código más legible.

Ejemplo:

```
async function ejemploAsync() {
  try {
    let resultado = await new Promise((resolve) => setTimeout(() => resolve("Listo"), 2000));
    console.log(resultado);
  } catch (error) {
    console.error("Error:", error);
  }
}
```

```
ejemploAsync();
```

6. Librerías útiles

- **jQuery** (menos usado actualmente)
- **animate.css**, **move.js**, **velocity.js** (para animaciones avanzadas)

Consejo: Usa herramientas nativas siempre que sea posible para optimizar el rendimiento.

Con estos ejemplos y explicaciones puedes tener una referencia rápida para aplicar animaciones y manejar asincronía en tus proyectos. ¡Suerte en el examen! 🚀