

Centro de Tecnología y Artes Visuales

Desarrollo de Sitios Web Móviles y Adaptables

Proyecto #2

Profesor: Andrés Díaz Soto

Estudiante: Yeth Penado

Técnicas de Lazy Loading

1. Evitar que la imagen cargue por adelantado. Si el navegador obtiene el atributo "src" de una etiqueta , activará la descarga de la imagen, independientemente de si está dentro o fuera del viewport actual. Para diferir la carga, debe colocarse la URL de la imagen en un atributo que no sea "src", por ejemplo, un atributo "src-data". Ahora que "src" está vacío, el navegador no activará la carga de la imagen. Sin embargo, no se debe olvidar decirle al navegador cuándo debe cargarla o de lo contrario, nunca se mostrará. Para esto, se debe indicar que tan pronto como la imagen ingrese al viewport, activamos la carga.
2. Mediante el **manejo de eventos** en Javascript, es posible evitar cargar imágenes por adelantado. Event Listeners como **scroll**, **resize** y **orientationChange** son importantes para trabajar Lazy Loading. El evento de scroll, evidentemente va a dispararse cuando el usuario haga scroll en la página. El evento resize ocurre cuando el tamaño del viewport del navegador cambia, y el orientationChange se dispara cuando el dispositivo gira en posición horizontal o vertical. Cuando ocurre cualquiera de estos eventos, encontramos que todas las imágenes en la página que están en modo Lazy Load y aún no se han cargado. A partir de estas imágenes, se debe verificar cuáles están ahora en el viewport. Si ha entrado en el viewport, se selecciona la URL del atributo "data-src" y la colocamos en el atributo "src". Esto dispara

la carga de la imagen. A su vez, debe eliminarse la clase "lazy" que identifica las imágenes que se cargarán en modo Lazy Load para los eventos que se activarán más adelante. Una vez que todas las imágenes están cargadas, se eliminan los Event Listeners. **Ejemplo de código:**

https://codepen.io/imagekit_io/pen/MBNwKB

3. Existen muchas bibliotecas y plugins dedicadas al Lazy Load que han adoptado dentro de su catálogo esta funcionalidad. Algunas de ellas son: jQuery, Lozad.js, LazySizes, Vue.js, Magento Lazy Image Loader, WordPress A3 Lazy Load, entre otras.
4. Mediante la API Intersection Observer (véase la siguiente sección).

Nota: Una buena práctica, es que las primeras imágenes presentes en el primer viewport, sí se carguen por adelantado.

Intersection Observer

Esta API hace que sea mucho más sencillo detectar cuándo un elemento ingresa al viewport, lo que genera una acción en el momento en el que dicho elemento hace su ingreso. El Intersection Observer (cuyo funcionamiento es asíncrono) deja al código registrar un callback function que se ejecuta si un elemento que se desea monitorizar entra o sale de otro elemento (o del viewport), o cuando la cantidad por la que ambos elementos se intersecan cambia en una cantidad requerida.

Intersection Observer – ¿Cómo funciona?

Se adjunta el 'observador' en todas las imágenes para que se carguen en modo Lazy Load. Una vez que la API detecta que el elemento ha entrado en el viewport, utilizando la propiedad **isIntersecting**, se selecciona la URL del atributo "data-src" y se desplaza al atributo "src", para que el navegador active la carga de la imagen. Una vez hecho esto, se debe eliminar la clase "lazy" de la imagen y también eliminamos el observador de dicha imagen. **Ejemplo de código:**

https://codepen.io/imagekit_io/pen/BPXQZZ

Nota: Intersection Observer es relativamente nuevo, por lo tanto, no está soportado en todos los navegadores.

IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Blackberry Browser	Opera Mobile	Chrome for Android	Firefox for Android	IE Mobile	UC Browser for Android
			4-50											
	12-14	2-51	51-57		10-37									
	15	52-54	58-64		38-44									
6-10	16-17	55-67	65-74	3.1-12	45-60	3.2-12.1		2.1-4.4.4	7	12-12.1			10	
11	18	68	75	12.1	62	12.3	all	67	10	46	75	67	11	11.8
	76	69-70	76-78	13-TP		13								