



CETAV

Jimmy Alvarez Mendoza

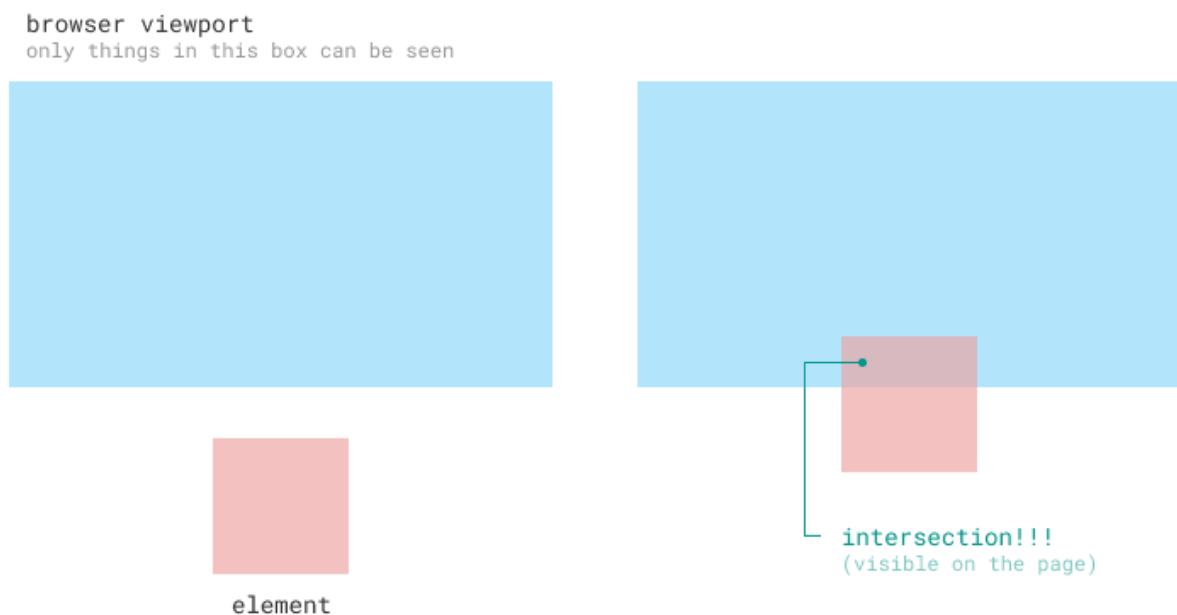
<https://github.com/JgeovanniAm/Laboratorio/tree/master/lazyload>

Diseño de Sitios Web Móviles



Intersection Observer API

Intersection observer API es una técnica en la cual el browser observa si hay intersecciones entre elementos padres e hijos. Lo cual nos puede ayudar en diferentes situaciones y una de tantas es facilitar mucho el performance o en la rapidez de nuestro sitio. Se utiliza mucho en “infinite scroll” y para hacer lazyload de img o files.



Detectamos si un elemento entró en las medidas del viewport lo que es visible para el usuario.

¿Cómo usarlo?

Tenemos una Class de javascript que recibe un callback de la lógica de la intersección de mis elementos.

```
const observerClass = new IntersectionObserver(observerMyitem, {  
  root:wrapper,  
  threshold:0
```

```
}}
```

root: es un key de un objeto que recibe como valor el elemento padre o “viewport” lo que el usuario va estar viendo.

threshold: indican el porcentaje de visibilidad del elemento y es donde yo le indico donde se va ejecutar mi lógica

```
function observerMyitem( entries , observer){
  entries.forEach(item => {
    if(item.isIntersecting){
      const Result = item.target.innerText + ' / ' +
item.intersectionRatio;
      output.innerHTML = Result;
    }
  });
}
```

Después tenemos propiedades que nos ayudan a identificar la intercesión de mis ítems

como: `isIntersecting`

Y propiedades que nos permite conocer el resultado del radio en donde interseca nuestro item entre 0 y 1:

```
divs.forEach(element => {
  myobserverClass.observe(element)
});
```

Seguido de esto nos encontramos con una function que nos permite observar el

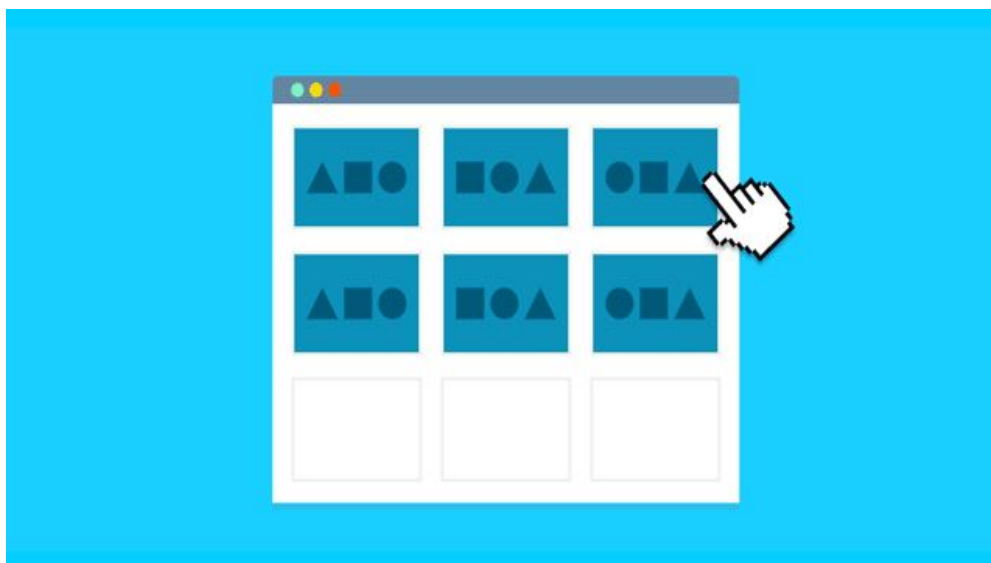
comportamiento de nuestros elementos con `MYCLASS.observe(element)//mis`

`elementos`

Lazy Loading

Lazy Load Remastered / [lazyload](#)

Es una librería o plugin de Js , que su función es dividir el código en partes y cargarlos en tiempos distintos, esto acelera la carga inicial de la aplicación y aligera su peso general. Y los eventos los “dispara por medio del scroll” lo cual solo cargará lo que el usuario esté apunto de ver y no cargue lo que no esté necesitando.



¿Cómo usarlo?

Descargue el plugin o use el CDN y aplíquelo en la etiqueta script

```
<script  
src="https://cdn.jsdelivr.net/npm/lazyload@2.0.0-beta.2/lazyload.js"></script>
```

y declara las clases de sus imágenes que desea aplicar el lazyload, y en su script usa la función lazyload y como parámetro sus [imágenes]

```
let images = document.querySelectorAll('.myImages');  
console.log(images)  
lazyload(images)
```

Y en su html usar el atributo **data-src**

```
img class "myImages" data-src "https://placeimg.com/400/200/tech" width "765"
height "574"
```

Lozad

Es una librería o plugin de js .

Está escrito con el objetivo de cargar de forma perezosa imágenes, marcos, anuncios, videos o cualquier otro elemento mediante la API de Intersection Observer recientemente agregada con enormes beneficios de rendimiento.

Podemos obtener la librería mediante npm o por un CDN

```
script type "text/javascript"
src "https://cdn.jsdelivr.net/npm/lozad/dist/lozad.min.js" script
```

```
npm install --save lozad
```

¿Cómo usarlo?

Primero set nuestras imágenes con una clase llamada lozad o la que nos guste,

```
img class "lozad" data-src "https://placeimg.com/400/200/animals" width "765"
height "574"
```

Luego en nuestro javascript

declaramos una variable con nuestra function lozad y podemos pasar como parámetro nuestro array de img o si no ponemos un parámetro el automáticamente buscará una clase llamada "lozad".

```
const observer = lozad() // so no ponemos nada como parametro el automaticamento
buscará una clase llamada lozad
observer.observe
```

y le decimos a nuestra variable que observe nuestras img , la cual en otras palabras estamos usando un intersection observer por detrás.