Intro to JS - Part 2 - Day 4: Sept 23

Intersection Observer API

An observer can be configured to watch for changes in the visibility of one or more target elements, and to trigger a callback function whenever the intersection ratio of a target element changes.

The intersection ratio is a measure of how much of the target element is currently visible within the viewport. (0: not visible, 1: completely visible)

1. create an instance of the IntersectionObserver class

   ```
   const observer = new IntersectionObserver(callback);
   ```

2. use its **observe** method to start watching for changes in the intersection of a target element

   ```
   const element = document.querySelector('#first-observer-target');
   observer.observe(element);
   ```

3. Define the callback function

   ```
   function logIntersection(entries, observer) { entries.forEach(entry
   => { // Do something with the entry data const entryObject = {
   isIntersecting: entry.isIntersecting, intersectionRatio:
   entry.intersectionRatio, target: { element: entry.target.tagName,
   id: entry.target.id } } if (entry.isIntersecting) {
   console.log(`${entry.target.id} is intersecting`) } else {
   console.log(`${entry.target.id} is NOT intersecting`) }
   console.log('Entry: ', entryObject) console.log('\n')
   observer.unobserve(entry.target) }) }
   ```

Thresholds

set one or more thresholds (minimum intersection ratio) at which the callback function should be triggered

```
const secondObserverOptions = { threshold: 0.5 };
const secondObserver = new IntersectionObserver(logIntersection,
secondObserverOptions);
const element2 = document.querySelector('#second-observer-target');
secondObserver.observe(element2);
```

Viewport Margins

Add margin values to the **rootMargin** property of the options object.
rootMargin property : a string that contains four margin values
(pixels, percentages, or a combination of both)

```
const thirdObserverOptions = { rootMargin: '0px 0px 100px 0px' }; const
thirdObserver = new IntersectionObserver(logIntersection,
thirdObserverOptions) const element3 =
document.querySelector('#third-observer-target')
thirdObserver.observe(element3)
//callback function will be triggered when the target element is 100px
from the bottom of our viewport.
```