/\*

  - Set vs WeakSet

  "

  the WeakSet is Weak,

  meaning references to object in a WeakSet are Held weakly.

  if no other references to an Object stored in the WeakSet exist,

  those objects can be garbage collected.

  "

  --

  Set     => Can Store Any Data Values

  WeakSet => collection of Objects Only

  --

  Set     => Have Size Property

  WeakSet => Does Not Have Size property

  --

  Set     => Have Keys, Values, Entries

  WeakSet => Does Not Have Clear, Key, Values And Entries

  --

  Set     => Can Use forEach

  WeakSet => Cannot Use ForEach

  Usage: Store objects and removes them once they become inaccessible

\*/

// Size

console.log(`Size Of Elmenet Inside Set Is: ${mySet.size}`);

// Values + Keys [Alias For Values]

let iterator = mySet.keys();

// the word key and the word value are the same thing

console.log(iterator.next().value); // it would keep picking up the next after the other so it will be 1 2 3 A

console.log(iterator.next().value);

console.log(iterator.next().value);

console.log(iterator.next().value);

console.log(iterator.next());// here when u remove the value it will tell u that theres no more index in the array so it will give u a true

// forEach

mySet.forEach((el) => console.log(el));

console.log("#".repeat(20));

// type Of Data

let mySet = new WeakSet([{A: 1, B: 2}]);

console.log(mySet);

// WeakSet Use Cases