/\*

- Reduce

--- method executes a reducer function on each element of the array

--- resulting in a single output value.

syntax

reduce(callBAckFunc(Accumualtor, Current val, Current Index source Array) { }, initialVaule)

- Accumulator => the accumulated value perviously returned in the last invocation

- current val => the current element being processed in the array

- index => the index of the current element being processed in the array

-------- starts from index 0 if an initialVaule is provided

-------- Otherwise, it starts from index 1

- Array +> the Current Array

\*/

let nums = [10, 20, 15, 30];

let add = nums.reduce(function (acc, current, index, arr) {

  // acc is the first first value which is 10 and current is the one after it (second one) which is 20

  console.log(`Acc => ${acc}`); //  acc is the first first value which is 10

  console.log(`Current Element => ${current}`); // current is the one after it (second one) which is 20

  console.log(`Current Element Index => ${index}`); // the index of the current here which is 1

  console.log(`Array => ${arr}`); // array is the amount of elements display: Array => 10,20,15,30

  console.log(acc + current);

  console.log(`############`);

  return acc + current;

}, 5); // this is the initialVaule if u put any number here it will start from it and then it would go to the elements so it will count here the 5 + 10 then the current will be 10 and the index of the current will be 0, also keep in mind that it wont count it in the array

console.log(add);