MERN STACK DEVELOPMENT

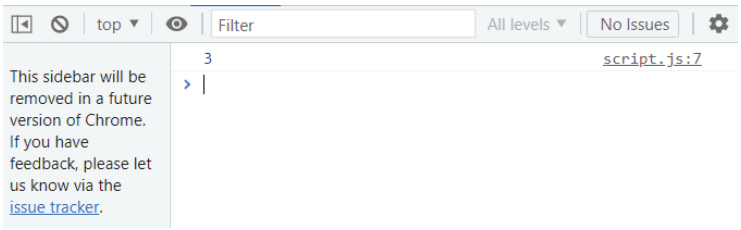
ES5 vs ES6 - Task 6

3. Part 3: Find the culprits and nail them – debugging javascript

Fix the code to get the largest of three.

```
aa = (f, s, t) => {
  if (f > s && f > t) {
    console.log(f);
  } else if (s > f && s > t) {
    console.log(s);
  } else {
    console.log(t);
  }
};
aa(1, 2, 3);
```

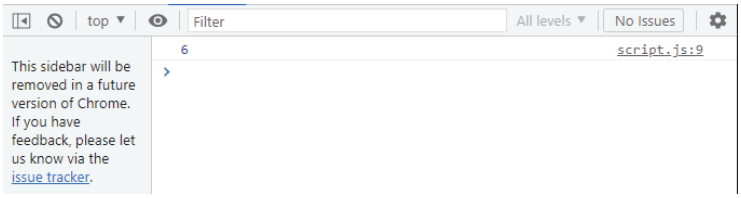
Input: (1, 2, 3) Output: 3



Fix the code to Sum of the digits present in the number

```
let n = "123";
function add(n) {
  let sum = 0;
  for (var i = 0; i < n.length; i++) {
    sum += parseInt(n[i]);
  }
  return sum;
}
console.log(add(n));
```

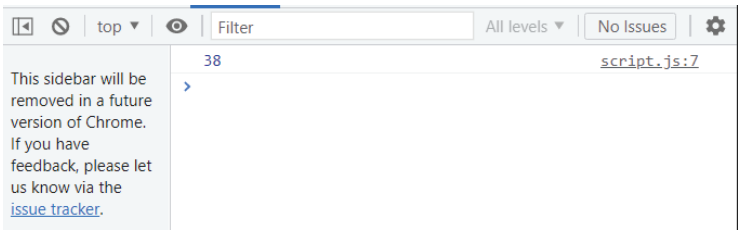
Input: 123 Output: 6



Fix the code to Sum of all numbers using IIFE function

```
const arr = [9, 8, 5, 6, 4, 3, 2, 1];
(function(array){
  let sum = 0;
  for (let i = 0; i < array.length; i++) {
  sum += arr[i];
}
  console.log(sum);
})(arr);
```

Input: [9,8,5,6,4,3,2,1] Output: 38



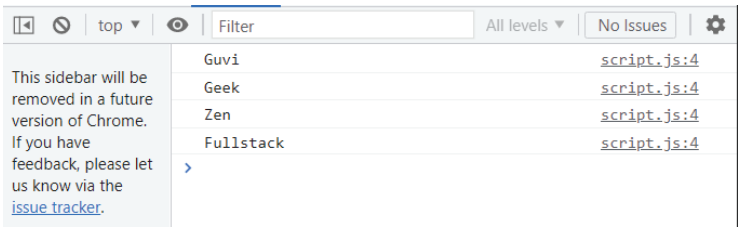
MERN STACK DEVELOPMENT

ES5 vs ES6 – Task 6

Fix the code to gen Title caps.

```
var arr = ["guvi", "geek", "zen", "fullstack"];
var ano = function(arro){
 for(var i = 0; i < arro.length; i++){
 console.log(arro[i][0].toUpperCase() +
arro[i].substr(1));
 }
}
ano(arr);
```

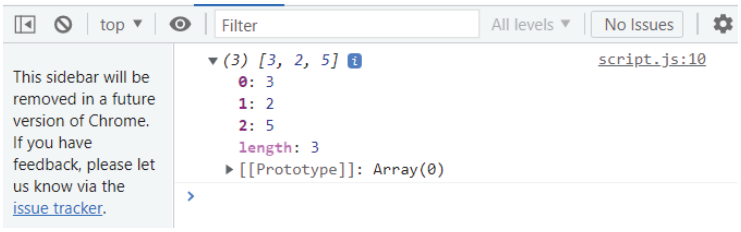
Input: ["guvi", "geek", "zen", "fullstack"]



Fix the code to return the Prime numbers

```
const newArray = [1, 3, 2, 5, 10];
const myPrime = newArray.filter((num) => {
  for (let i = 2; i < num; i++) {
    if (num % i === 0) {
      return false;
    }
  }
  return num > 1;
});
console.log(myPrime);
```

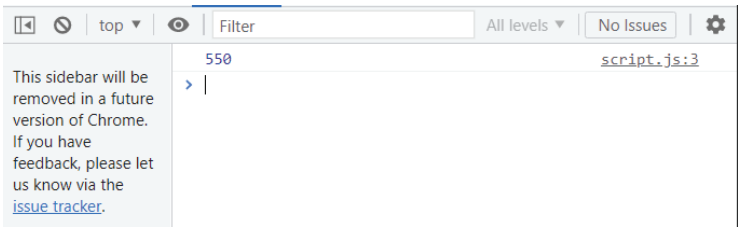
Input: [1,3,2,5,10] Output: [3, 2, 5]



Fix the code to sum the number in that array

```
const num =
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100];
const sum = (a, b) => a + b;
console.log(num.reduce(sum));
```

Input: [10, 20, 30, 40,50,60,70,80,90,100] Output: 550



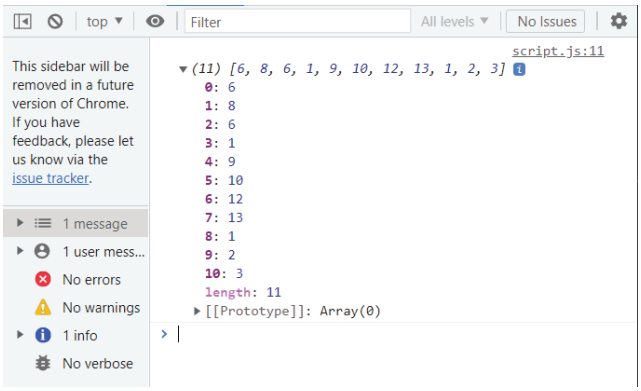
MERN STACK DEVELOPMENT

ES5 vs ES6 – Task 6

Fix the code to rotate an array by k times and return rotated array using IIFE function

```
var arr = [1, 2, 3, 6, 8, 6, 1, 9, 10, 12, 13];
var k = 3;
k = arr.length % k;
(function (array) {
  out = array.slice(k + 1, array.length);
  var count = out.length;
  for (var i = 0; i < k + 1; i++) {
    out[count] = array[i];
    count += 1;
  }
  console.log(out);
})(arr);
```

Input: [1, 2, 3, 6, 8, 6, 1, 9, 10, 12, 13]

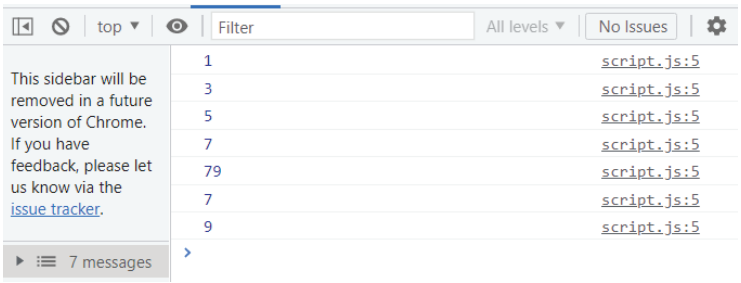


---

print all odd numbers in an array using IIFE function

```
var arr = [1, 2, 3, 5, 7, 79, 7, 2, 6, 9, 4];
(function() {
  for (var i = 0; i < arr.length; i++) {
  if (arr[i] % 2 !== 0) {
  console.log(arr[i]);
  }}
 })();
```

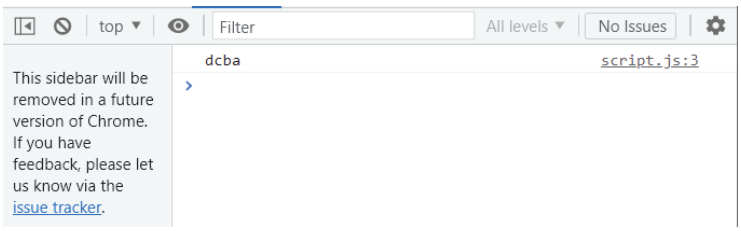
Input: [1, 2, 3, 5, 7, 79, 7, 2, 6, 9, 4]



Fix the code to reverse

```
(function(str){
  let str1 = str.split("").reverse().join("");
  console.log(str1);
 })("abcd")
```

Input: abcd Output: dcba



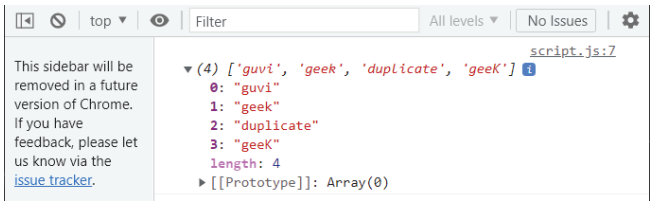
MERN STACK DEVELOPMENT

ES5 vs ES6 – Task 6

Fix the code to remove duplicates.

```
var res = function(arr){
 newArr = [ ];
 for(var i=0; i < arr.length; i++){
 if(newArr.indexOf(arr[ i ]) === -1) {
 newArr.push(arr[ i ]);
 } }
 console.log(newArr)
}
res(["guvi","geek","guvi","duplicate","geeK"]);
```

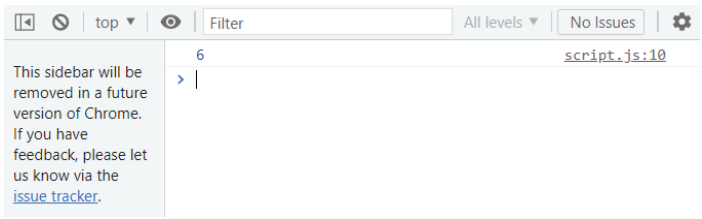
Input: ["guvi","geek","guvi","duplicate","geeK"]



Fix the code to give the Sum of odd numbers in an array

```
var as = [12, 34, 5, 6, 2, 56, 6, 2, 1];
var s = as
 .filter((x) => {
  if (x % 2 !== 0) {
   return x;
  } })
 .reduce(function (a, c) {
  return a + c;
 });
console.log(s);
```

Input: [12,34,5,6,2,56,6,2,1]



Fix the code to swap the odd and even digits

```
var aa = (data) => {var a = data;
 var l = "";
 for (i = 0; i < a.length - 1; i++) {
  var s = a[ i + 1];
  var b = a[ i ];
  l += s;
  l += b;
  i = i + 1;
  console.log(l);
 }};
aa("1234");
```

Input: 1234