

First thing we do is check our input array:

```
2  const unsortedArray = [40,30,20,10,4,3,2,1]
3  for(let x of unsortedArray) {
4      if(!Number.isInteger(x)){
5          console.log("found non-integer value inside the input array.")
6          return
7      } else if(x < 0) {
8          console.log("found negative value inside the input array.")
9          return
10     }
11 }
```

... more specifically, we are checking to see if it is an integer, greater than zero, and not negative.

Then we identify 40 as the max value.

The length of 40 as the iterator will be 2 (ie, 2 digits). This means we will iterate 2 times:

```
17  // first digit:
18  // bucketArray(buckets) = [ [], [], [], [], [], [], [], [], [], [] ]
19  //           0  1  2  3  4  5  6  7  8  9
20  // bucketArray(buckets) = [ [40,30,20,10], [1], [2], [3], [4], [], [], [], [], [] ]
21  //           0           1  2  3  4  5  6  7  8  9
22  // inputArray = [40,30,20,10,1,2,3,4]
23
24  // second digit:
25  // bucketArray(buckets) = [ [], [], [], [], [], [], [], [], [], [] ]
26  //           0  1  2  3  4  5  6  7  8  9
27
28  // bucketArray(buckets) = [ [], [], [], [], [], [], [], [], [], [] ]
29  //           0  1  2  3  4  5  6  7  8  9
30  // inputArray =[1,2,3,4,10,20,30,40]
```

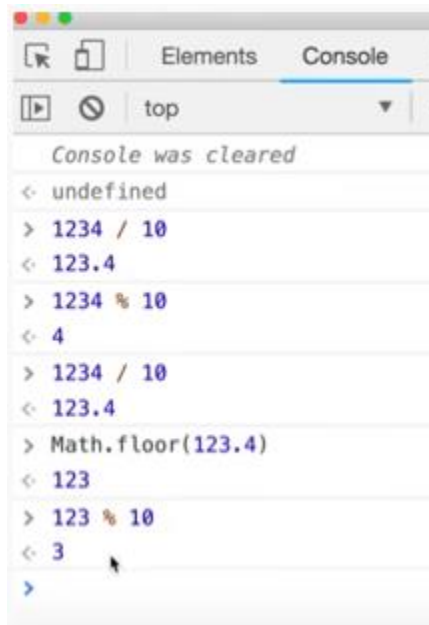
Notice how the list is sorted in each step based on digits:

```
17 // first digit:
18 // bucketArray(buckets) = [ [], [], [], [], [], [], [], [], [], [] ]
19 //           0  1  2  3  4  5  6  7  8  9
20 // bucketArray(buckets) = [ [40,30,20,10], [1], [2], [3], [4], [], [], [], [], [] ]
21 //           0           1  2  3  4  5  6  7  8  9
22 // inputArray = [40,30,20,10,1,2,3,4]
23
24 // second digit:
25 // bucketArray(buckets) = [ [], [], [], [], [], [], [], [], [], [] ]
26 //           0  1  2  3  4  5  6  7  8  9
27
28 // bucketArray(buckets) = [ [], [], [], [], [], [], [], [], [], [] ]
29 //           0  1  2  3  4  5  6  7  8  9
30 // inputArray =[1,2,3,4,10,20,30,40]
```

If we want the smallest digit of, for example, the integer 1234, we use 10:

```
> 1234 % 10  
< 4
```

Note steps within the console:



```
Console was cleared  
< undefined  
> 1234 / 10  
< 123.4  
> 1234 % 10  
< 4  
> 1234 / 10  
< 123.4  
> Math.floor(123.4)  
< 123  
> 123 % 10  
< 3  
>
```



We take the input array and check the input of the array: at least values in the... otherwise we return input array.

```
31 function radixSort(inputArray){
32     if(inputArray.length < 2) return inputArray
33     let maxValue = inputArray[0]
34     for(let i=1;i<inputArray.length;i++){
35         if(inputArray[i] > maxValue){
36             maxValue = inputArray[i]
37         }
38     }
```

At the end of this, our max value will be obtained and compared against the input array.

Then we parse in the length,

add another for-loop,

and bucketArray is declared with an empty array,

And digitValue will find the remainder using 10:

...after the array's for-loops blocks are finished, our bucketArray will be used to rearrange our inputArray:

```
39     const iterationCount = maxValue.toString().length
40     for(let digit=0; digit<iterationCount; digit++){
41         const bucketArray = Array.from({length:10}, ()=> [])
42         for(let i=0; i<inputArray.length; i++){
43             const digitValue = Math.floor(inputArray[i] / Math.pow(10,digit)) % 10
44             bucketArray[digitValue].push(inputArray[i])
45         }
46         inputArray = [].concat(...bucketArray)
47     }
48     return inputArray
49 }
```

```

31 function radixSort(inputArray){
32     if(inputArray.length < 2) return inputArray
33     let maxValue = inputArray[0]
34     for(let i=1;i<inputArray.length;i++){
35         if(inputArray[i] > maxValue){
36             maxValue = inputArray[i]
37         }
38     }
39     const iterationCount = maxValue.toString().length
40     for(let digit=0; digit<iterationCount; digit++){
41         const bucketArray = Array.from({length:10}, ()=> [])
42         for(let i=0; i<inputArray.length; i++){
43             const digitValue = Math.floor(inputArray[i] / Math.pow(10,digit)) % 10
44             bucketArray[digitValue].push(inputArray[i])
45         }
46         inputArray = [].concat(...bucketArray)
47     }
48     return inputArray
49 }

```

... now let's view it in the console:

```

50 console.log(radixSort(unsortedArray.slice()))

```