Coding Question-2

FLATTERN

Interviewquestion-43

JS

Flattern

Input: let arr = [[1, 2], [3, 4], [5, 6],[7, 8, 9], [10, 11, 12, 13]];

Output: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13];

Interviewer:
The input depth can be infinite and you should able to flattern the input

Input   let arr = [[1, 2], [3, 4], [5, 6],[7, 8, 9], [10, 11, 12, 13]];

```
let flatArray = arr.reduce((prev, current) => {
   return prev.concat(current)
}, []);
```

Output: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]

Note

- we know **reduce function** reduces to **one single value**
- concat() method is used to combine two or more arrays or values into a new array

explanation

step:1

```
let flatArray = arr.reduce([ ],[1,2] ) => {
    return [ ].concat([1,2]) //[1,2]
}, [ ]);
```

step:2

```
let flatArray = arr.reduce([ 1,2],[3,4] ) => {
    return [1,2].concat([3,4]) //[1,2,3,4]
}, []);
```

step:3

```
let flatArray = arr.reduce([ 1,2,3,4],[5,6] ) => {
    return [1,2,3,4].concat([5,6]) //[1,2,3,4,5,6]
}, []);
```

This steps are repeated until the array is flattern

Easy way to handle the same problem

```
let flatArray = [].concat(...arr);
```

To handle infinite depth array to flattern

Input: let arr = [[1, 2], [3, 4], [5,[6]],
[7,[8, 9]], [10, 11, 12, 13]];

Output: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13];

code as follows

To handle infinite depth array to flattern

```javascript
const flatArray=(arr) =>{
  return arr.reduce((prev, current) => {
  if(Array.isArray(current)){
    console.log("value")
      prev=prev.concat(flatArray(current))
   }
  else{
   prev.push(current);
   }
  return prev
}, []);
}
```

short code with same output

```javascript
console.log(arr.flat(Infinity));
```

explanation with simple words

- This line **Array.isArray(current)** tells the elements are array or not

- Once it is an array it goes inside and checks whether in-depth array is present or not by once again calling the flatArray **(prev.concat(flatArray(current)))**

- As of second point its knocks **flatArray(current)** once again and checks array of array is present are not , if there is no array its goes to else statement ,where its pushing the elements into prev (empty array at start) .

This pattern goes of and gives us desired output