* arr = [1, 2, 3, 4, 5, 6] ;

*

$$\text{* arr2} = [\overset{0}{1}, \overset{1}{2}, \overset{2}{3}, \overset{3}{4}, \overset{4}{[5, 6]}] ; \Rightarrow 1 \text{ (depth of the array)}$$

arr2[0]

arr2[1]

---

* arr[4][0]

* arr[4][1]

let arr3 = [1, [2,3,4], 5, [6,[7,8]]];

$$\underset{\text{depth } 2}{\text{arr3}} = [\underset{0}{1},\ \underset{1}{\underbrace{[2,3,4]}},\ \underset{0}{5},\ \underset{1}{\underbrace{[6,[7,8]]}}]$$

~~0~~ ~~1~~ 2

$$\boxed{[\text{an array having depth } 1] \Rightarrow 2}$$

[2, 3, 4] ⇒ depth 0

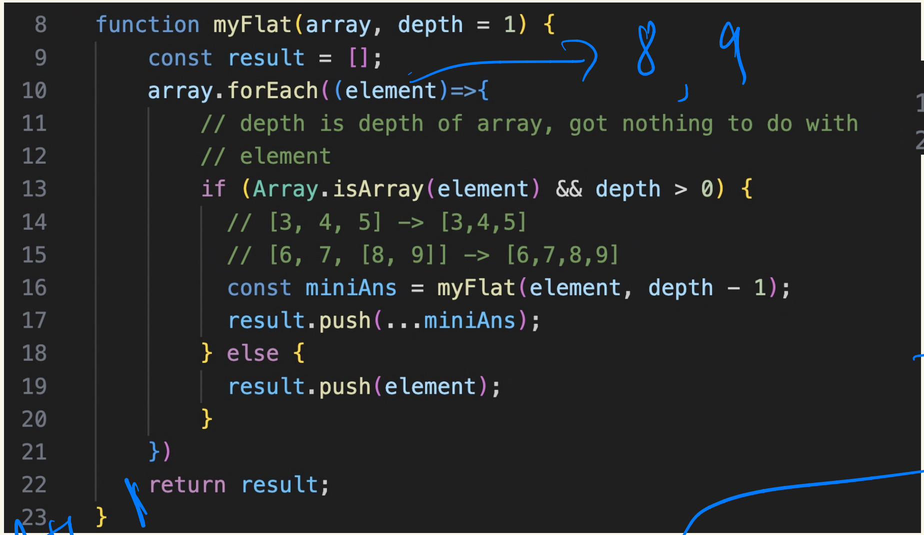
[[2, 3, 4]] ⇒ depth ⇒ 1

* let arr3 = [1, [2,3,4], 5, [6,[7,8]]] ⇒ depth is 2

* [1, 2, 3, 4, 5, 6, [7, 8]] ⇒ depth 1
*

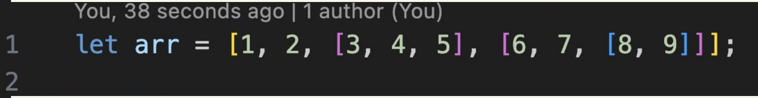
* [1, 2, 3, 4, 5, 6, 7, 8] ⇒ depth 0

* [ ] ⇒



```javascript
function myFlat(array, depth = 1) {
    const result = [];
    array.forEach((element)=>{
        // depth is depth of array, got nothing to do with
        // element
        if (Array.isArray(element) && depth > 0) {
            // [3, 4, 5] -> [3,4,5]
            // [6, 7, [8, 9]] -> [6,7,8,9]
            const miniAns = myFlat(element, depth - 1);
            result.push(...miniAns);
        } else {
            result.push(element);
        }
    })
    return result;
}
```

8, 9



You, 38 seconds ago | 1 author (You)

```javascript
let arr = [1, 2, [3, 4, 5], [6, 7, [8, 9]]];
```

result ⇒ [1, 2, 3, 4, 5, 6, 7, 8, 9];

depth

* result ⇒ [6, 7, 8, 9];

depth 0

result ⇒ [8, 9]

* dry run with depth 1

* dry run code without depth
(mentioned in reso)