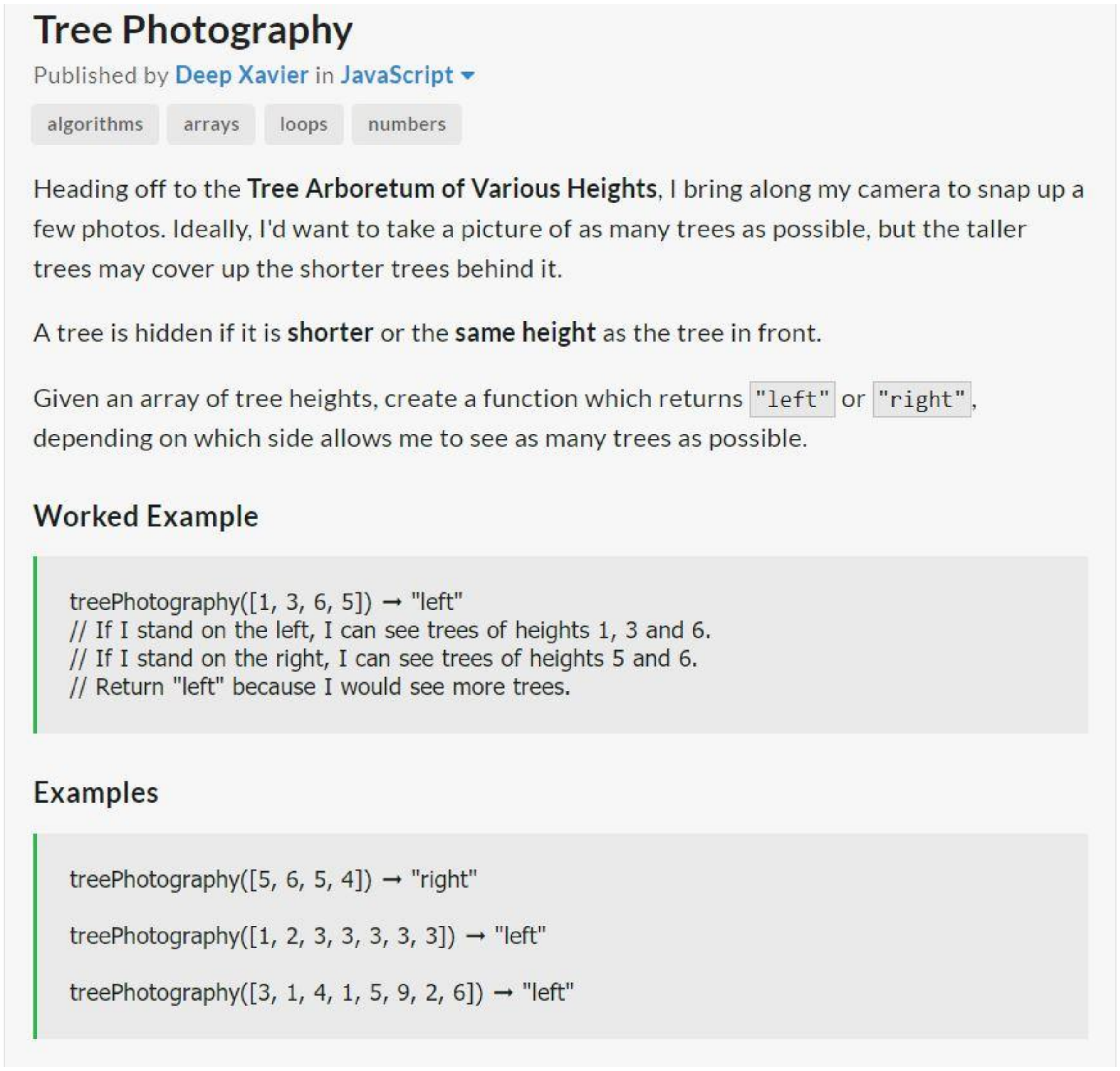
ข้อ 8



Tree Photography

Published by Deep Xavier in JavaScript

algorithms arrays loops numbers

Heading off to the **Tree Arboretum of Various Heights**, I bring along my camera to snap up a few photos. Ideally, I'd want to take a picture of as many trees as possible, but the taller trees may cover up the shorter trees behind it.

A tree is hidden if it is **shorter** or the **same height** as the tree in front.

Given an array of tree heights, create a function which returns `"left"` or `"right"`, depending on which side allows me to see as many trees as possible.

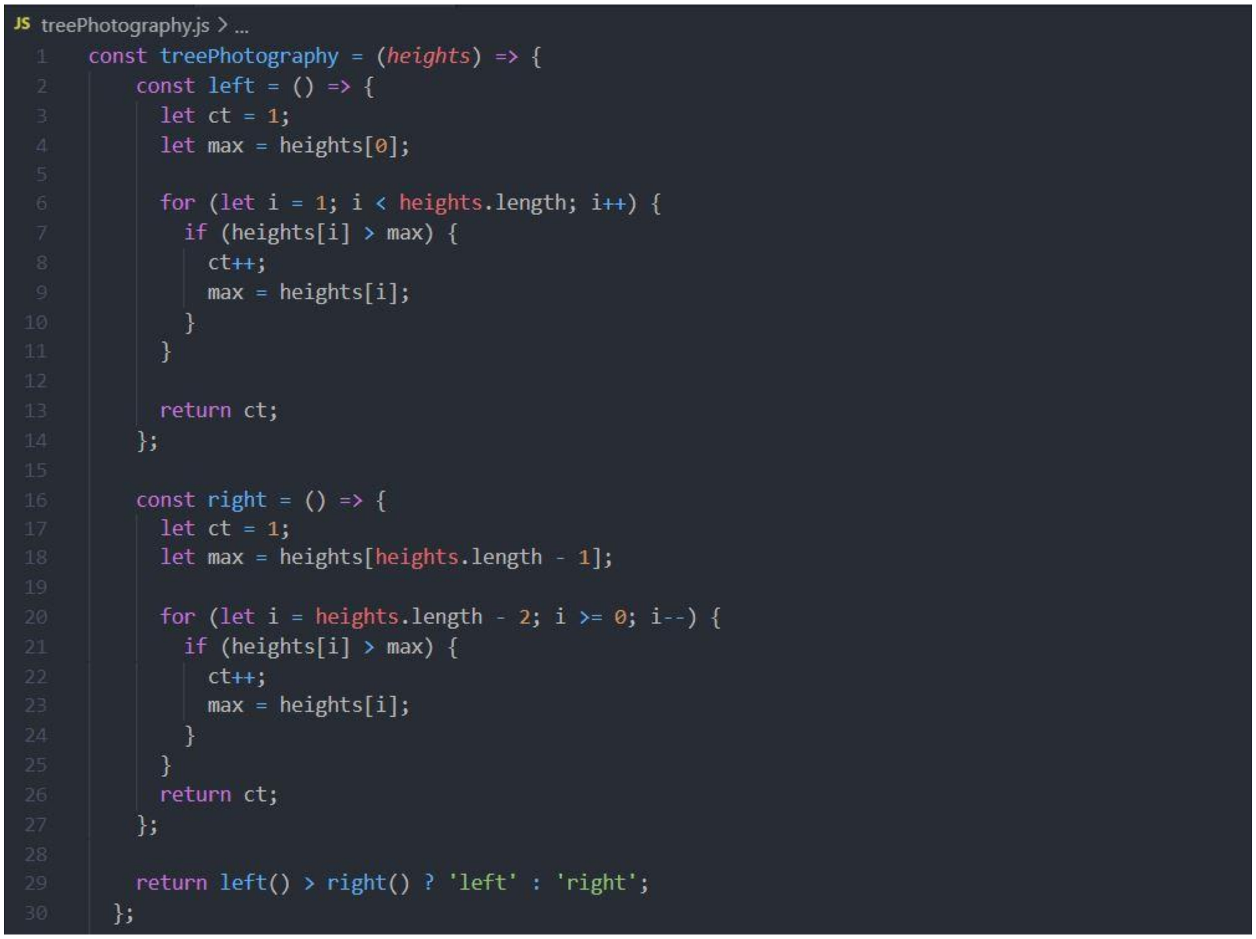
Worked Example

```
treePhotography([1, 3, 6, 5]) → "left"
// If I stand on the left, I can see trees of heights 1, 3 and 6.
// If I stand on the right, I can see trees of heights 5 and 6.
// Return "left" because I would see more trees.
```

Examples

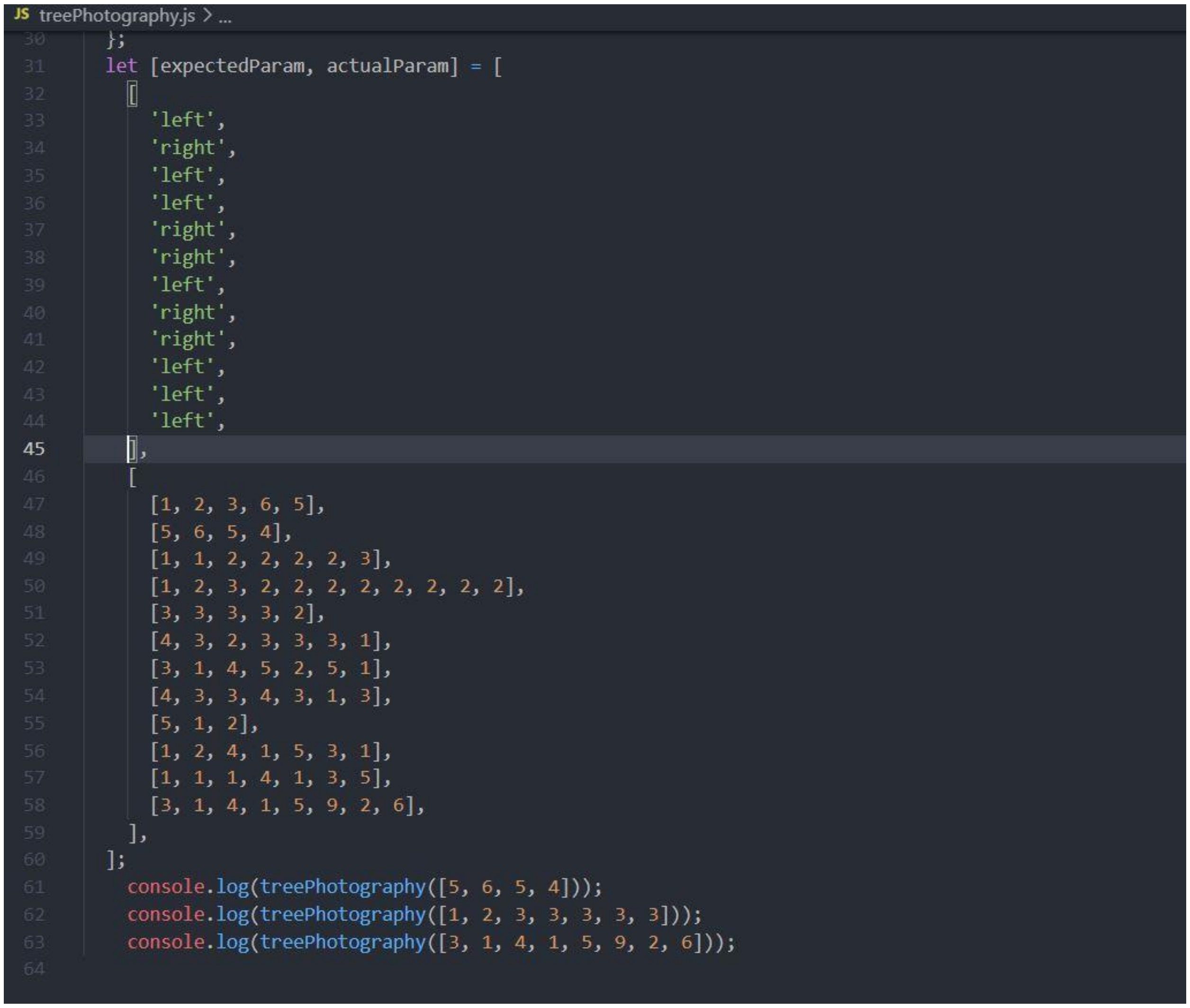
```
treePhotography([5, 6, 5, 4]) → "right"

treePhotography([1, 2, 3, 3, 3, 3, 3]) → "left"

treePhotography([3, 1, 4, 1, 5, 9, 2, 6]) → "left"
```

Code



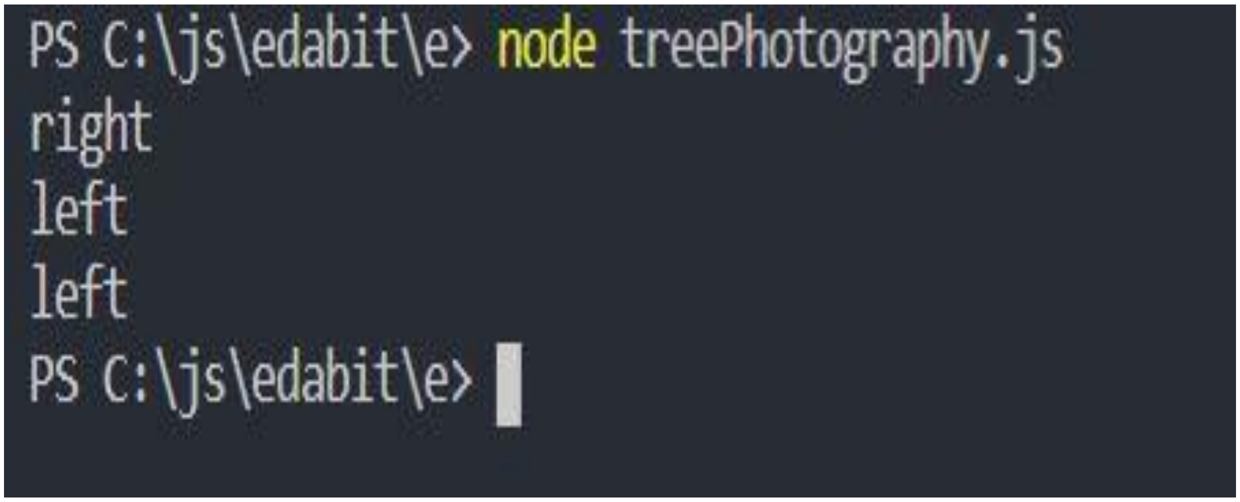
treePhotography.js > ...

```javascript
const treePhotography = (heights) => {
    const left = () => {
      let ct = 1;
      let max = heights[0];

      for (let i = 1; i < heights.length; i++) {
        if (heights[i] > max) {
          ct++;
          max = heights[i];
        }
      }

      return ct;
    };

    const right = () => {
      let ct = 1;
      let max = heights[heights.length - 1];

      for (let i = heights.length - 2; i >= 0; i--) {
        if (heights[i] > max) {
          ct++;
          max = heights[i];
        }
      }
      return ct;
    };

    return left() > right() ? 'left' : 'right';
  };
```


treePhotography.js > ...

```
  };
  let [expectedParam, actualParam] = [
    [
      'left',
      'right',
      'left',
      'left',
      'right',
      'right',
      'left',
      'right',
      'right',
      'left',
      'left',
      'left',
    ],
    [
      [1, 2, 3, 6, 5],
      [5, 6, 5, 4],
      [1, 1, 2, 2, 2, 2, 3],
      [1, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2],
      [3, 3, 3, 3, 2],
      [4, 3, 2, 3, 3, 3, 1],
      [3, 1, 4, 5, 2, 5, 1],
      [4, 3, 3, 4, 3, 1, 3],
      [5, 1, 2],
      [1, 2, 4, 1, 5, 3, 1],
      [1, 1, 1, 4, 1, 3, 5],
      [3, 1, 4, 1, 5, 9, 2, 6],
    ],
  ];
    console.log(treePhotography([5, 6, 5, 4]));
    console.log(treePhotography([1, 2, 3, 3, 3, 3, 3]));
    console.log(treePhotography([3, 1, 4, 1, 5, 9, 2, 6]));
```

Run



```
PS C:\js\edabit\e> node treePhotography.js
right
left
left
PS C:\js\edabit\e>
```