

Stacks and Trees

Binary Tree

Write a constructor function called `BinaryTree` with no arguments. It should initialize one attribute:

- * ``head`` with ``null`` o
- * ``root`` with ``null``

The methods that you will implement are:

- * `add` (create a new Node)
- * `traverseDFS`
- * `traverseBFS`

```
```javascript
const tree = new BinaryTree();
tree.add(4);
tree.add(2);
tree.add(7);
tree.add(1);
tree.add(3);
```
```

```
```markdown
 4
 / \
 2 7
 / \
 1 3
```
```

```
```javascript
tree.traverseDFS(function(e) { console.log(e); });
// 4
// 2
// 1
// 3
// 7
```

```
tree.traverseBFS(function(e) { console.log(e); });
// 4
// 2
// 7
// 1
// 3
```
```

Balanced Parenthesis

Write a function called `isBalanced` that receives a string and returns `true` if the parenthesis are balanced, `false` otherwise:

```
```javascript
isBalanced("((()))"); // true
```

```
isBalanced("((()))"); // false
isBalanced("()()()"); // false
``,`
```