

Pretty Print

2 min

Wouldn't it be nice to be able to display the structure of our tree in a captivating visual way? We have provided a helpful

Preview: Docs Loading link description

[method](#)

, `.print()` that will give you a formatted text display of our tree.

For example, a tree with 3 levels starting with root node 15, children 3, 12, 0, and grandchildren 6, 9, 19, 8, 10, 19 is displayed below:

```
15
-- 3
-- -- 6
-- -- 9
-- 12
-- -- 19
-- -- 8
-- 0
-- -- 10
-- -- 19
```

This method takes one

Preview: Docs Loading link description

[parameter](#)

, `level`, which is initialized to 0, to enable printing the entire tree structure from the top to the bottom.

Instructions

1. Checkpoint 1 Passed

1.

Open **script.js**, and study how we add data in a sample tree. Then call `.print()` on the sample tree to see the output on the terminal.

TreeNode.js

```
class TreeNode {  
  constructor(data) {  
    this.data = data;  
    this.children = [];  
  }  
  
  addChild(child) {  
    if (child instanceof TreeNode) {  
      this.children.push(child);  
    } else {  
      this.children.push(new TreeNode(child));  
    }  
  }  
  
  removeChild(childToRemove) {  
    const length = this.children.length;  
    this.children = this.children.filter(child => {  
      return childToRemove instanceof TreeNode  
        ? child !== childToRemove  
        : child.data !== childToRemove;  
    });  
  
    if (length === this.children.length) {  
      this.children.forEach(child => child.removeChild(childToRemove));  
    }  
  }  
  
  print(level = 0) {
```

```
let result = "";
for (let i = 0; i < level; i++) {
  result += '-- ';
}
console.log(`${result}${this.data}`);
this.children.forEach(child => child.print(level + 1));
}
};
```

```
module.exports = TreeNode;
```

script.js

```
const TreeNode = require('./TreeNode');
const tree = new TreeNode(1);
const randomize = () => Math.floor(Math.random() * 20);
```

```
// add first-level children
for (let i = 0; i < 3; i++) {
  tree.addChild(randomize());
}
```

```
// add second-level children
for (let i = 0; i < 3; i++) {
  for (let j = 0; j < 2; j++) {
    tree.children[i].addChild(randomize());
  }
}
```

```
// add third-level children
```

```

for (let i = 0; i < 3; i++) {
  for (let j = 0; j < 2; j++) {
    for (let k = 0; k < 2; k++) {
      tree.children[i].children[j].addChild(randomize());
    }
  }
}

```

```

// pretty-print the tree
tree.print();

```

>>Output

```

1
-- 19
-- -- 6
-- -- -- 12
-- -- -- 4
-- -- 3
-- -- -- 6
-- -- -- 10
-- 7
-- -- 15
-- -- -- 4
-- -- -- 3
-- -- 7
-- -- -- 1
-- -- -- 13
-- 17
-- -- 1

```

-----12

-----9

---14

-----2

-----12