

# Introduction to Binary Trees: Takeaways



by Dataquest Labs, Inc. - All rights reserved © 2021

## Syntax

- Node implementation:

```
class Node:
    def __init__(self, value):
        self.value = value
        self.left_child = None
        self.right_child = None
```

- BST implementation:

```
class BST:
    def __init__(self):
        self.root = None
    def add(self, value):
        if self.root is None:
            # The root does exist yet, create it
            self.root = Node(value)
        else:
            # Find the right place and insert new value
            self._add_recursive(self.root, value)
    def _add_recursive(self, current_node, value):
        if value <= current_node.value:
            # Go to the left
            if current_node.left_child is None:
                current_node.left_child = Node(value)
            else:
                self._add_recursive(current_node.left_child, value)
        else:
            # Go to the right
            if current_node.right_child is None:
                current_node.right_child = Node(value)
            else:
                self._add_recursive(current_node.right_child, value)
    def _contains(self, current_node, value):
        if current_node is None:
            return False
        if current_node.value == value:
            return True
        if value < current_node.value:
            return self._contains(current_node.left_child, value)
        return self._contains(current_node.right_child, value)
```

```
def contains(self, value):  
    return self._contains(self.root, value)
```

## Concepts

- A binary tree is a tree data structure in which each node has at most two children, which we refer to as the left child and the right child.
- We call the top node of the tree the "root."
- We call a node a "leaf" if it has no children.
- We call a node that is neither the root nor a leaf an "internal node."
- A binary search tree is a binary tree such that, for each node, all values on the left are smaller than or equal to the node value, and all values on the right are larger than the node value.
- The height of a tree is the length of the longest path from the root to a leaf.
- The height of a binary tree is at most  $n - 1$  and at least  $\log_2(n)$  where  $n$  is the number of nodes in the tree.

## Resources

- [Binary trees](#)
- [Binary search tree](#)
- [Tree data structures for beginners](#)