Introduction: Nodes in JavaScript

Now that you have an understanding of what nodes are, let's see one way they can be implemented using Javascript.

We will create a basic node that contains data and one link to another node. The node's data will be specified when creating the node and immutable (can't be updated). Remember that a node's link to the next node is `null` when there are no nodes to traverse.

Go ahead and take a look at the starter code in the editor. You will find the `Node` class defined. `module.exports` allows the `Node` class to be used outside this module. Make sure to always leave this line of code. We'll need it when we use this class to implement more complex data structures.

Instructions

1.

Let's begin by setting up the constructor for our `Node` class. Since nodes contain data, we want the constructor to expect a `data` argument of any type to be passed in. The constructor can assign the given argument to the `data` property of the `Node` instance.

Be sure to set the arguments to the appropriate properties in `this` class (i.e. `this.data`).

Hint

Remember, classes are created using the `class` expression, and `constructor` is a method that will run when an instance of the class is first created.

```
class MyClass {
  constructor(someValue) {
    this.someValue = someValue;
  }
}
```

2.

Let's check that our `Node` class has the correct data.

Underneath the `Node` class, instantiate a `Node` with any value and set it to `firstNode`. Then use `console.log()` to print out the instance's `data` property. We should see the value that the node was instantiated with in the terminal.

Hint

Create a `Node` instance using the `new` operator and assign it to the variable `firstNode`:

```
const firstNode = new Node();
```

3.

We also know that a node is aware of the other node it links to. We will represent this with the `next` property on the Node class.

Similar to how we created the `data` property in the constructor, let's do the same with the `next` node property. When the node is first created, it is an orphan node (a node with no links). Set the `next` node to `null`.

4.

Let's check that `next` node property was successfully set in the constructor.

Underneath the node we previously created, use `console.log()` to print out the instance's `next` node property. Check that `null` is output in the terminal.

Node.js

```
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }
}

module.exports = Node;

const firstNode = new Node('hi');
console.log(firstNode.data);
console.log(firstNode.next);
```

Node Methods: Set Next Node

Currently when a node is created, the `next` node is initially set to `null`, and we do not have a formal way to change it. However, we want to allow the `next` node to be updated so it can be traversed and used in more complex data structures. For this, we will use a setter to modify `this.next` node property.

Instructions

1.

Implement the `.setNextNode()` method in the `Node` class.

It should take `node` as an argument and update the `next` node property appropriately.

Hint

Make sure that you pass in a `node` argument and set it to the corresponding property.

2.

To verify that our `.setNextNode()` performs as intended, let's call the method on our `Node` instance. Create a second `Node` instance and set it to `secondNode`. Link it to the `firstNode` by passing `secondNode` to the call to `setNextNode`.

Now let's print out `firstNode` so we can see it in its entirety using `console.log()`. We should see the second node instance set to the `next` node property instead of the default `null` value.

Hint

To call a method on an instance of a class, you can access it similarly to a class property.

```
classInstance.method();
```

Node.js

```
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    this.next = node;
  }
}

const firstNode = new Node('I am an instance of a Node!');

module.exports = Node;
firstNode.setNextNode();
const secondNode = new Node();
firstNode.setNextNode(secondNode);
console.log(firstNode);
```

Node Methods: Set Next Node Validation

We arbitrarily set our `next` node to any argument that gets passed in. This can be problematic. Imagine if the `next` node accidentally gets set to a different data type, like a string. We run the risk of mistakenly confusing the string for a node, and we would have to build out special support to avoid traversing anything that is not a node.

To prevent these unnecessary complications, let's add in a check that only allows arguments that are `instanceof` nodes or `null`. We want to allow `null` values as an argument in the event that we want to break the link between a node and its next node.

Instructions

1.

Inside `.setNextNode()`, check if the `node` argument is an `instanceof` the `Node` class. If the argument is a `Node` or `null`, set `this.next` equal to `node`. Otherwise, `throw` an error.

Hint

When we check if a variable is an `instanceof` a class, it will evaluate to a boolean, like so:

```
const x = new String('this is a string');
// this will evaluate to true
if (x instanceof String || x === null) {
  console.log('x is a string!');
}
```

We can also throw an instance of an Error and pass it an informational message.

```
throw new Error('message');
```

2.

We know that we can set the `next` node to another node, so let's verify that `.setNextNode()` will not accept an argument that is not a node.

Call the `.setNextNode()` method on our `Node` instance and pass it any argument that is not a node. We expect to see an error in the terminal because you didn't set the next node to a `Node` instance.

Hint

For the argument, you can choose anything, like a string, boolean, number, array, etc.

Node.js

```
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
        throw new Error('node has to be an instance of the Node class');
      }
    }
```

```
}

const firstNode = new Node('I am an instance of a Node!');

firstNode.setNextNode(rana);

module.exports = Node;
```

Node Methods: Get Next Node

We could continue accessing the `next` node property directly, like how we have been doing so far. However, if we ever want it to be given in a special way, we would want to use a getter to handle the preprocessing.

Let's go ahead and create a simple `.getNextNode()` method that just returns the `next` node property.

Instructions

1.

Implement the `.getNextNode()` method in the `Node` class.

Hint

The method should return the node's `next` node property.

2.

Using the `Node` instance that we have already created, verify that the `.getNextNode()` method returns the correct `next` node property by logging the call from `firstNode`.

Node.js

```
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
```

```
      this.next = node;
    } else {
      throw new Error('Next node must be a member of the Node class.');
    }
  }

  getNextNode(node) {
    return this.next;
  }
}

const firstNode = new Node('I am an instance of a Node!');
const secondNode = new Node('I am the next Node!');
firstNode.setNextNode(secondNode);
console.log(firstNode.getNextNode(secondNode));

module.exports = Node;
```

Review: Nodes in JavaScript

Let's see all of our `Node` class methods together in action!

Imagine we are working at an ice cream shop that sells three flavors: strawberry, vanilla, and coconut. The signature sundae is made of these three flavors, but our new hires have a hard time remembering the order.

To help them remember, our JavaScript nodes may do just the trick. Let's get started...

Instructions

1.
Outside of `Node`, instantiate three more nodes.

- The first will represent our strawberry ice cream with the name, `'Berry Tasty'`. Assign it to the variable, `strawberryNode`
- The second will represent our vanilla ice cream with the value, `'Vanilla'`. Assign it to the variable, `vanillaNode`

- The third will represent our coconut ice cream with the value, `'Coconuts for Coconut'`. Assign it to the variable, `coconutNode`

2.

Now let's put these ice cream nodes in order. vanilla goes first, followed by strawberry. Finally, coconut goes after strawberry.

Below the newly created nodes, use your `.setNextNode()` method so that:

- `strawberryNode` is the `next` node of `vanillaNode`
- `coconutNode` the `next` node of `strawberryNode`

3.

Let's walk through our ice cream nodes to make sure that they are linked in the correct order. Create a new `currentNode` and set it `vanillaNode`. We will use `currentNode` to iterate through the nodes, so declare it using `let`. Create a `while` loop that will only iterate when the `currentNode` is not `null`.

Inside the `while` loop, log out the `currentNode`'s `data`, and update `currentNode` to the next node.

We should see the ice cream flavors in order of vanilla, strawberry, and coconut in the terminal.

When your code is passing, take a moment to consider:

- What other ways do you think nodes could be helpful for keeping track of and storing information?
- What could happen if we added another link to the `Node` class?
- What if we created an instance of a `Node` with another `Node` instance?

Hint

We can iterate through the nodes in a similar way that we would an array.

```
const sampleArray = [1, 2, 3];

let currentIndex = 0;
while(currentIndex < sampleArray.length) {
  currentIndex += 1;
}
```

`currentIndex` is what we use to iterate through our nodes, so we set it to the very first position. In the `while` condition, we check to make sure there are no more positions left to iterate on, and inside the loop, we update the current position to the next position.

Node.js

```
class Node {
  constructor(data) {
    this.data = data;
    this.next = null;
  }

  setNextNode(node) {
    if (node instanceof Node || node === null) {
      this.next = node;
    } else {
      throw new Error('Next node must be a member of the Node class.');
    }
  }

  getNextNode() {
    return this.next;
  }
}

const strawberryNode = new Node('Berry Tasty');
const vanillaNode = new Node('Vanilla');
const coconutNode = new Node('Coconuts for Coconut');

vanillaNode.setNextNode(strawberryNode);
strawberryNode.setNextNode(coconutNode);

let currentNode = vanillaNode;

while (currentNode) {
  console.log(currentNode.data);
  currentNode = currentNode.getNextNode();
}
```

```
module.exports = Node;
```