**Simple Lists Kata**

**Description**

Simple lists of data may have more magic behind the scenes than you might think. We are going to experiment with some basic processing of lists.

**Rules of the Game**

Lists are one of the first data structures that we learn as programmers. But the fact that we are familiar with them, doesn't mean that we cannot learn a little from them. In this kata, we are going to code up an implementation of a list that has the following basic interface:

* A list consists of nodes. Each node has a value of type string.
* The new nodes are added at the end of the list.
* You can ask the list if it contains a particular string. If so, it will return the node that contains the string.
* You can remove any node from the list.
* You can ask the list to return an array with all the values of its nodes.

Remember, you cannot simply use data lists/collection features of most modern programming languages; build the list features by hand through separate functions. Implement the requirements through a singly linked list (each node has a reference to the next node) or a doubly linked list (each node has a reference to both the next and previous nodes).

There’s nothing magical or surprising in list implementations, but there are a fair number of boundary conditions. For example, when deleting from the singly-linked list, did you have to deal with the case of deleting the first element in the list specially?

For this kata, concentrate on ways of removing as many of these boundary conditions as possible. Then ask yourself: Is the resulting code, which will contain fewer special cases, easier to read and maintain?

And don’t forget your tests. Have fun!