Linked List in Python

Learning Objectives

- Understand ADT List
- Implement ADT List as Linked List
- Explore ADT List as Doubly Linked List

ADT List

A list is a collection of items which are arranged in linear order. The terminology across programming languages is not always consistent. Both vectors and lists are sequential ***collections*** (Java) or ***containers*** (C++) which allow adding and removing items. A usual distinction is that vectors allow random access with an index (position) while lists store elements at non-contiguous memory location (using doubly linked-lists internally) and need an extra step to access items by index, usually by traversing the list from start to index. That is an implementation detail which should be addressed in the implementation stage.

Lists could be implemented using arrays (possibly requiring resizing and copying), vectors, or something new.

ADT List Interface

In this interface, items are referenced by their position within the list. Other implementations may use values for identification.

`createList()`	Create an empty list
`isEmpty()`	Determine whether a list is empty
`size()`	Determine the number of items in a list
`insert(index, value)`	Add an **`item`** at a given **`index`** in the list
`remove(index)`	Remove the item at a given **`index`** in the list
`clear()`	Remove all items from the list
`get(index)`	Retrieve (get) the item at a given **`index`** in the list

```
x.next = x.next.next
```

Linked List Implementation

Goodbye

Auf Wiedersehen

再见