# Simple List class

# Recap

- HW2 - recap.
- Reading was due
   Any questions?

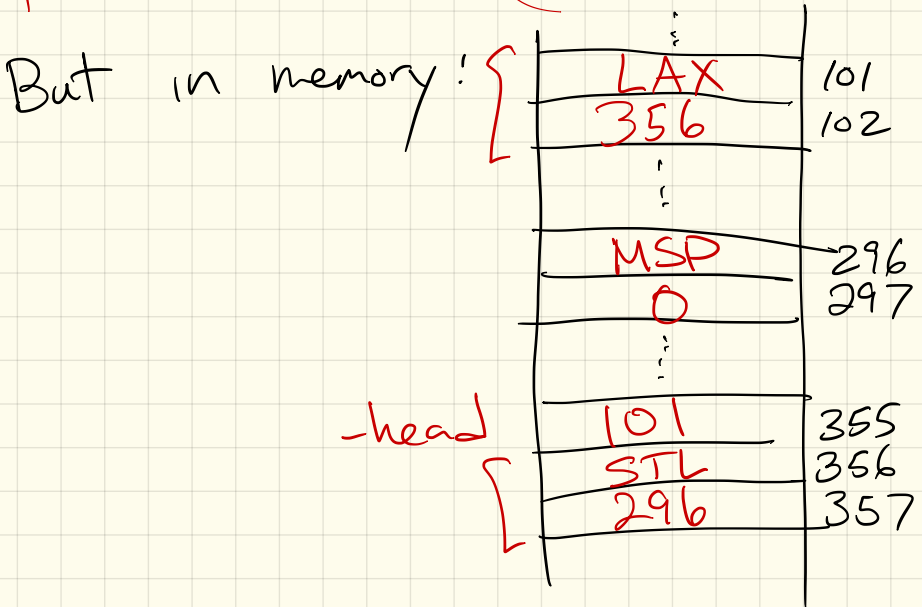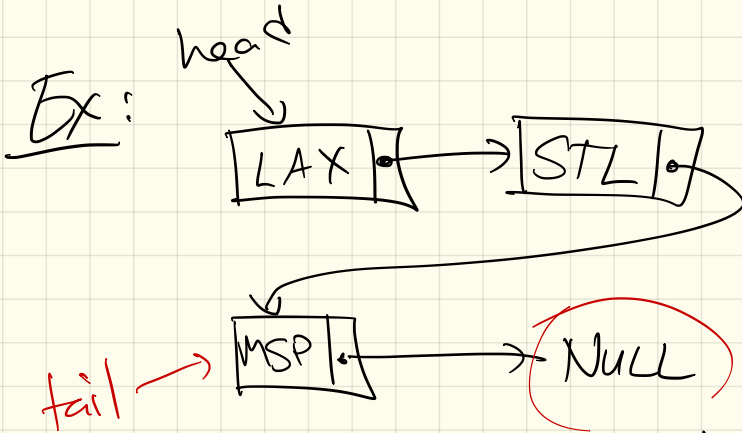- Reading assignment for
   Wednesday                    ;

- No office hours Thursday -
   will make up Wed. or fr


- Next HW: on lists
   will be on ZyBooks

Now: A first data structure

Singly linked lists:
A collection of nodes that
have a linear ordering

Ex:

head

| LAX | • | → | STL | • |

tail → | MSP | • | → ( NULL )

But in memory!

| LAX | 101 |
| 356 | 102 |
| ⋮ | |
| MSP | 296 |
| 0 | 297 |
| ⋮ | |
| 101 | 355 |
| STL | 356 |
| 296 | 357 |

head

# Why this structure?

Note: Not the same as C++'s list class (or Python's, for that matter)

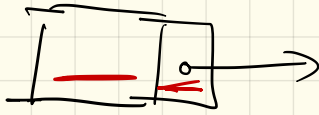However, this linked structure is useful in a number of data structures.

Why not use an array?
- fixed size

downside:
- getting $i^{th}$ element is slow

# Implemention: Nodes



Huh?

We'll need, a node struct (or class).

```
struct SNode {
    Object data;
    SNode * next;
}
```

Then, in the class, have:

```
private:
    SNode * head;
    SNode * tail;
```

Functons?

append
insertAfter / Before
Prepend
empty

# Code

```cpp
template <typename Object>
class SLinkedList {
private:
    struct SNode {
        Object data;
        SNode * next;
    }
    int s
    SNode * ; head;
public:
    SLinkedList();
    ~SLinkedList();
}
```

# Today: Code!

(on webpage)