3.2 Stacks and Queues

"A pile of objects stacked up"

THE STACK ADT:

- Contains zero or multiple items (*simple but powerful*)
- Add an item -> top of the stack -> "pushing" onto the stack
- Remove an item -> remove from top -> "popping" from the stack
- Last-In-First-Out: LIFO Behaviour -> first item added to the stack is the last item removed (*Net Effect*)

- **Data:**
  - A collection of items
- **Operations:**
  - Determine whether the stack is empty
  - Add an item (*push*)
  - Remove most recently-added item (*pop*)

List - based implementation of the Stack ADT:

Using the back of the list as the top of the stack

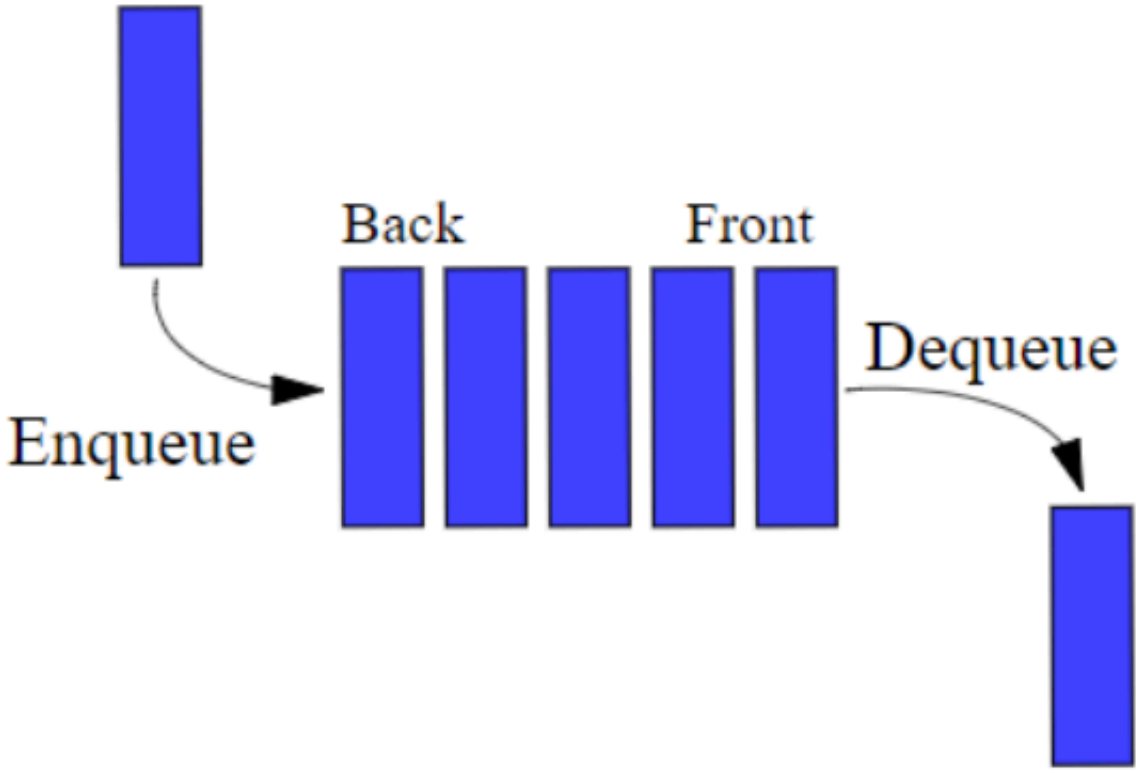
```
class Stack:
    """A last-in-first-out (LIFO) stack of items.

    Stores data in first-in, last-out order. When removing an item from the
    stack, the most recently-added item is the one that is removed.
    """
    # === Private Attributes ===
    # _items:
    #     The items stored in the stack. The end of the list represents
    #     the top of the stack.
    _items: List

    def __init__(self) -> None:
        """Initialize a new empty stack.
        """
        self._items = []

    def is_empty(self) -> bool:
        """Return whether this stack contains no items.

        >>> s = Stack()
        >>> s.is_empty()
```

```
        True
        >>> s.push('hello')
        >>> s.is_empty()
        False
        """
        return self._items == []

    def push(self, item: Any) -> None:
        """Add a new element to the top of this stack.
        """
        self._items.append(item)

    def pop(self) -> Any:
        """Remove and return the element at the top of this stack.

        >>> s = Stack()
        >>> s.push('hello')
        >>> s.push('goodbye')
        >>> s.pop()
        'goodbye'
        """
        return self._items.pop()
```

THE QUEUE ADT:

- Contains zero or more items
- Items come in order in which they entered.
- First-In-First-Out: FIFO behaviour

- **Data:**
  - A collection of items
- **Operations:**
  - Determine whether the stack is empty
  - Add an item (enqueue)
  - Remove the least recently-added item (dequeue)


Back
Front
Dequeue
Enqueue