Queue using List

**Building a Queue in Python**

Before we start let us reiterate the key components of a queue.

A queue is a data structure that consists of two main operations: enqueue and dequeue.

Enqueue is when you add a element to the tail of the queue and a pop is when you remove an element from the head of the queue.

Python 3.x conviently allows us to demonstate this functionality with a list. When you have a list such as [2,4,5,6] you can decide which end of the list is the front and the back of the queue respectively.

Once you decide that, you can use the append, pop or insert function to simulate a queue.

We will choose the first element to be the front of our queue and therefore be using the append and pop functions to simulate it. Give it a try by implementing the function below!

**Try Building a Queue**

In [1]:



**class** Queue:

**def** \_\_init\_\_(self):

self.storage **=** []

**def** size(self):

**return** len(self.storage)

**def** enqueue(self, item):

self.storage.append(item)

​

**def** dequeue(self):

**return** self.storage.pop(0)

**Test the Queue**

In [2]:



*# Setup*

q **=** Queue()

q.enqueue(1)

q.enqueue(2)

q.enqueue(3)

​

*# Test size*

print ("Pass" **if** (q.size() **==** 3) **else** "Fail")

​

*# Test dequeue*

print ("Pass" **if** (q.dequeue() **==** 1) **else** "Fail")

​

*# Test enqueue*

q.enqueue(4)

print ("Pass" **if** (q.dequeue() **==** 2) **else** "Fail")

print ("Pass" **if** (q.dequeue() **==** 3) **else** "Fail")

print ("Pass" **if** (q.dequeue() **==** 4) **else** "Fail")

q.enqueue(5)

print ("Pass" **if** (q.size() **==** 1) **else** "Fail")

Pass

Pass

Pass

Pass

Pass

Pass