






Create a Queue Class




Like stacks, queues are a collection of elements. But unlike stacks, queues follow the FIFO (First-In First-Out) principle. Elements added to a queue are pushed to the tail, or the end, of the queue, and only the element at the front of the queue is allowed to be removed.

We could use an array to represent a queue, but just like stacks, we want to limit the amount of control we have over our queues.

The two main methods of a queue class is the enqueue and the dequeue method. The enqueue method pushes an element to the tail of the queue, and the dequeue method removes and returns the element at the front of the queue. Other useful methods are the front, size, and isEmpty methods.

Write an `enqueue` method that pushes an element to the tail of the queue, a `dequeue` method that removes and returns the front element, a `front` method that lets us see the front element, a `size` method that shows the length, and an `isEmpty` method to check if the queue is empty.

	Your <code>Queue</code> class should have a <code>enqueue</code> method.
	Your <code>Queue</code> class should have a <code>dequeue</code> method.
	Your <code>Queue</code> class should have a <code>front</code> method.
	Your <code>Queue</code> class should have a <code>size</code> method.
	The <code>dequeue</code> method should remove and return the front element of the queue.

	The <code>front</code> method should return value of the front element of the queue.
	The <code>size</code> method should return the length of the queue.
	The <code>isEmpty</code> method should return false if there are elements in the queue.