

# Implementation of Deque

In this lecture we will implement our own Deque class!

## Methods and Attributes

- Deque() creates a new deque that is empty. It needs no parameters and returns an empty deque.
- addFront(item) adds a new item to the front of the deque. It needs the item and returns nothing.
- addRear(item) adds a new item to the rear of the deque. It needs the item and returns nothing.
- removeFront() removes the front item from the deque. It needs no parameters and returns the item. The deque is modified.
- removeRear() removes the rear item from the deque. It needs no parameters and returns the item. The deque is modified.
- isEmpty() tests to see whether the deque is empty. It needs no parameters and returns a boolean value.
- size() returns the number of items in the deque. It needs no parameters and returns an integer.

## Deque Implementation

```
In [1]: class Deque:
        def __init__(self):
            self.items = []

        def isEmpty(self):
            return self.items == []

        def addFront(self, item):
            self.items.append(item)

        def addRear(self, item):
            self.items.insert(0,item)

        def removeFront(self):
            return self.items.pop()

        def removeRear(self):
            return self.items.pop(0)

        def size(self):
            return len(self.items)
```

```
In [2]: d = Deque()
```

```
In [3]: d.addFront('hello')
```

```
In [4]: d.addRear('world')
```

```
In [5]: d.size()
```

Out[5]: 2

```
In [6]: print(d.removeFront() + ' ' + d.removeRear())  
hello world
```

```
In [7]: d.size()
```

Out[7]: 0

## Good Job!