**Justin\_Stack\_Design\_0232.**  **Implement Queue using Stacks**

**Concept:**

\_\_init\_\_ ==> 設一個 stack []

push ==> 直接 append

pop ==> [0]先存在 temp，再刪除，最後在return temp

top ==> return [0]

empty ==> 看是否 = []

**Code:**

class MyQueue:

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

"""

Initialize your data structure here.

"""

self.data = []

def push(self, x: int) -> None:

"""

Push element x to the back of queue.

"""

self.data.append(x)

def pop(self) -> int:

"""

Removes the element from in front of queue and returns that element.

"""

temp = self.data[0]

del self.data[0]

return temp

def peek(self) -> int:

"""

Get the front element.

"""

return self.data[0]

def empty(self) -> bool:

"""

Returns whether the queue is empty.

"""

return self.data == []

# Your MyQueue object will be instantiated and called as such:

# obj = MyQueue()

# obj.push(x)

# param\_2 = obj.pop()

# param\_3 = obj.peek()

# param\_4 = obj.empty()