**Justin\_Stack\_0225.**  **Implement Stack using Queues**

**Concept:**

\_\_init\_\_ ==> 設一個 stack []

push ==> 直接 append

pop ==> [-1]先存在 temp，再刪除，最後在return temp

top ==> return [-1]

empty ==> 看是否 = []

**Code:**

class MyStack:

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

"""

Initialize your data structure here.

"""

self.data= []

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

"""

Push element x onto stack.

"""

self.data.append(x)

def pop(self) -> int:

"""

Removes the element on top of the stack and returns that element.

"""

temp = self.data[-1]

del self.data[-1]

return temp

def top(self) -> int:

"""

Get the top element.

"""

return self.data[-1]

def empty(self) -> bool:

"""

Returns whether the stack is empty.

"""

return self.data == []

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

# obj = MyStack()

# obj.push(x)

# param\_2 = obj.pop()

# param\_3 = obj.top()

# param\_4 = obj.empty()