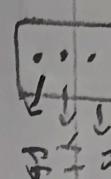


Stack & Queues (Chap 10)

Stack → Last in first out (LIFO)

Real world use case:

assume we visit in our browser first facebook and next youtube and
instagram



it saves like this, when we click back button from
instagram page it goes yt and pressing back again get to next item fb.

Stack → First in Last out (LIFO)

Stack constructed. (push, pop)

```
class Node:  
    def __init__(self, value):  
        self.value = value  
        self.next = None  
  
class Stack:  
    def __init__(self, stack):  
        self.top = None  
        self.length = 0  
  
    def push(self, value):  
        new_node = Node(value)  
        if self.length == 0:  
            self.top = new_node  
            self.length += 1  
        return  
  
    else:  
        current = self.top  
        new_node.next = current  
        self.top = new_node
```

stack.push(3) → None

Top → ②
None

③ ← Top
↓
②
↓
None

queue : i) Enqueue → insertion → names worse

ii) Dequeue → removal → first.

queue only supports enqueue (insert at last) and dequeue (remove from first)

iii) instead of head similar to linked list . instead head , tail here its called first and last.

iii) LL allows insertion , del at any position , while queue follows FIFO .

class Node :

def __init__(self, value) :

self.value = value

self.next = None

first
None
last

initially looks ⇒

None
first
last

after some enqueue ⇒

① → ② → ③ → None

class Queue :

def __init__(self) :

self.first = None

self.last = None

self.length = 0,

: