**Stack**

يعتبر stack أو ما يسمى بالمكدس إحدى بنى المعطيات الخطية تتبع مبدأ LIFO وهذا يعني أن أخر عنصر تمت أضافته إلى المكدس هو أول عنصر تتم إزالته



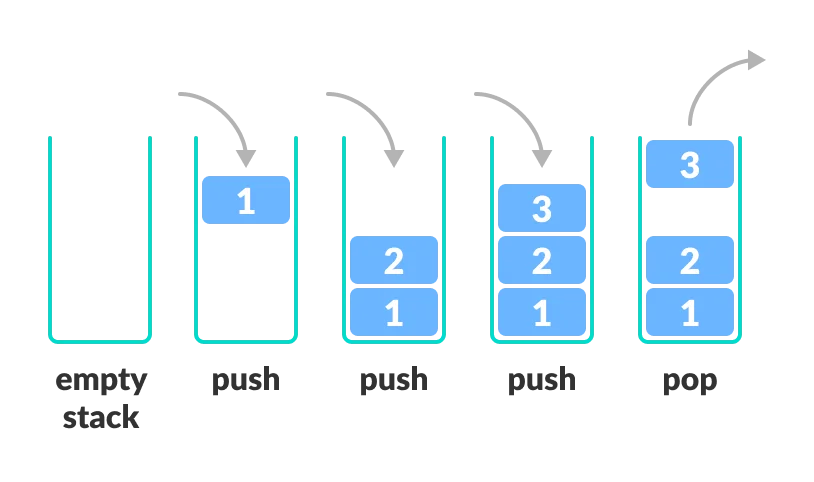
هنا يمكننا :

أصافة طبق في الأعلى

إزالة الطبق الموجود في الأعلى

إذا كنت تريد الطبق الموجود في الأسفل يجب عليك إزالة جميع الأطباق الموجود في الأعلى هذا بضبط يشرح كيفية عمل بنية المعطيات المكدس

من حيث البرمجة أضافة عنصر في أعلى المكدس يدعى push وحذف عنصر يدعى pop



في الصورة أعلاه العنصر الثالث تم أضافته في الأخير وتمت إزالته أولا وهذا بضبط الطريقة التي يعمل بها LIFO اخر عنصر تمت أضافته أول عنصر تتم إزالته

**هناك بعض العمليات الأساسية التي تسمح لنا تنفيذ إجراءات مختلفة على المكدس :**

إضافة عنصر في أعلى المكدس:Push

حذف عنصر من أعلى المكدس:Pop

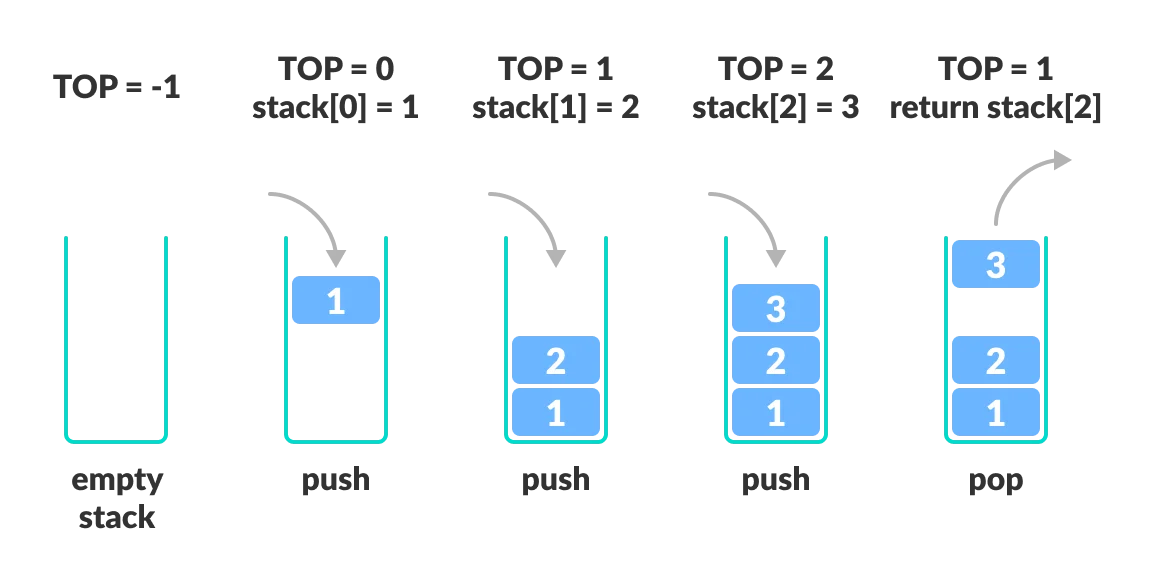
التحقق إذا كان المكدس فارغ:IsEmpty

التحقق إذا كان المكدس ممتلئ:IsFull

الحصول على قيمة أخر عنصر بدون حذفه:Peek

**كيفية عمل المكدس:**

* **المؤشر الذي يدعى top يستخدم لتعقب العنصر الموجود في القمة**
* **عند تهيئة المكدس نقوم بتعيين قيمةtop 1- حتى نتمكن من مما إذا كان المكدس فارغ عن طريق مقارنة top==-1**
* **عند أضافة عنصر نقوم بزيادة قيمة top ونقوم بوضع العنصر الجديد في الموقع الذي يشير له top**
* **عند حذف عنصر نقوم بإعادة العنصر الذي يشير له ال top وقوم بإنقاص قيمة top**
* **قبل الإضافة نتحقق إ1ا كان المكدس ممتلئ**
* **قبل الحذف نتأكد إذا كان المكدس فارغ**

****

تطبيق المكدس الأكثر شيوعا يستخدم المصفوفات ولكن يمكن أيضا تطبيقه باستخدام القوائم المترابطة

# Stack implementation in python

# Creating a stack

def create\_stack():

stack = []

return stack

# Creating an empty stack

def check\_empty(stack):

return len(stack) == 0

# Adding items into the stack

def push(stack, item):

stack.append(item)

print("pushed item: " + item)

# Removing an element from the stack

def pop(stack):

if (check\_empty(stack)):

return "stack is empty"

return stack.pop()

stack = create\_stack()

push(stack, str(1))

push(stack, str(2))

push(stack, str(3))

push(stack, str(4))

print("popped item: " + pop(stack))

print("stack after popping an element: " + str(stack))

التعقيد الزمني للمكدس:

تستغرق عمليات الحذف والإضافة زمن ثابت O(1) في المكدس الذي يعتمد في تطبيقه على المصفوفات

**تطبيقات المكدس :**

**على الرغم من أن المكدس من احدى بنى المعطيات سهلة التنفيد إلا أن تعتبر قوية جدا من أكثر الاستخدامات لها:**

* **لعكس الكلمة حيث نقوم بإضافة جميع الحروف إلى المكدس ثم نقوم بإخراجهم بسبب ألية عمل المكدس LIFO سوف نحصل على الحروف في ترتيب عكسي**
* **في المترجمات يستخدم المكدس لحساب قيمة التعبير التالي 2 + 4 / 5 \* (7 - 9) عن طريق تحويل هذا التعبير إلى** prefix  أو postfix
* **في المتصفحات يقوم يقوم الزر الخلفي في المتصفح بحفظ جميع عناوين URL التي زرتها سابقا في مكدس وفي كل مرة تقوم بزيارة صفحة جديدة تتم أضافتها أعلى المكدس عند الضغط على الزر تتم إزالة عنوان URL الحالي من المكدس ويتم الوصول إلى العنوان السابق**