**Data Structure**

**&**

**Algorithm**

**Class 8**

**Lab 10**

|  |
| --- |
| **Lab Objectives:** Linear structuresStackStack abstract data typeImplementation of stack |

## We will begin our study of data structures by considering four simple but very powerful concepts.

## Stacks, queues, deques, and lists are examples of data collections whose items are ordered depending on how they are added or removed.



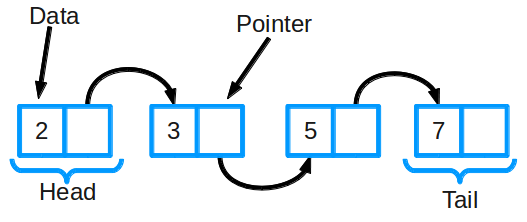
# What Are Linear Structures?

Once an item is added, it stays in that position relative to the other elements that came before and came after it. Collections such as these are often referred to as linear data structures.

## Linear structures can be thought of as having two ends. Sometimes these ends are referred to as the “left” and the “right” or in some cases the “front” and the “rear.” You could also call them the “top” and the “bottom” or “head” and “tail”.

## IMG_256

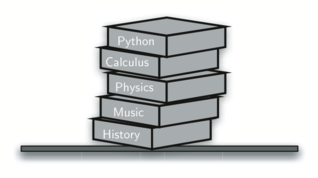
## For example, a structure might allow new items to be added at only one end. Some structures might allow items to be removed from either end.



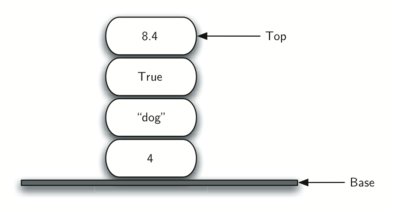
## These variations give rise to some of the most useful data structures in computer science.

# What is a Stack?

A stack (sometimes called a “push-down stack”) is an ordered collection of items where the addition of new items and the removal of existing items always takes place at the same end.



## This end is commonly referred to as the “top.” The end opposite the top is known as the “base.”



## This ordering principle is sometimes called LIFO, last-in first-out. Newer items are near the top, while older items are near the base.

## Imagine a stack of books on a desk. The only book whose cover is visible is the one on top. To access others in the stack, we need to remove the ones that are sitting on top of them.

## The order of insertion is the reverse of the order of removal.The picture below shows the Python data object stack as it was created and then again as items are removed. Note the order of the objects.

## IMG_256

## For example, every web browser has a Back button. As you navigate from web page to web page, those pages are placed on a stack (actually it is the URLs that are going on the stack). The current page that you are viewing is on the top and the first page you looked at is at the base. If you click on the Back button, you begin to move in reverse order through the pages.

# The Stack Abstract Data Type

## IMG_256The stack abstract data type is defined by the following structure and operations. A stack is structured, as described above, as an ordered collection of items where items are added to and removed from the end called the “top.” Stacks are ordered LIFO. The stack operations are given below.

## ***Stack()*** creates a new stack that is empty. It needs no parameters and returns an empty stack.

## ***push(item)*** adds a new item to the top of the stack. It needs the item and returns nothing.

## ***pop()*** removes the top item from the stack. It needs no parameters and returns the item. The stack is modified.

## ***peek()*** returns the top item from the stack but does not remove it. It needs no parameters. The stack is not modified.

## ***isEmpty()*** tests to see whether the stack is empty. It needs no parameters and returns a boolean value.

## ***size()***returns the number of items on the stack. It needs no parameters and returns an integer.

## Selection_005For example, if s is a stack that has been created and starts out empty, then *[Table 1](https://interactivepython.org/runestone/static/pythonds/BasicDS/TheStackAbstractDataType.html" \l "tbl-stackops)* shows the results of a sequence of stack operations. Under stack contents, the top item is listed at the far right.

# Implementing a Stack with Python

|  |
| --- |
| from pythonds.basic.stack import Stack  s=Stack()  print(s.isEmpty())  s.push(4)  s.push('dog')  print(s.peek())  s.push(True)  print(s.size())  print(s.isEmpty())  s.push(8.4)  print(s.pop())  print(s.pop())  print(s.size()) |

## Pythonds is the module of Python programming language for implementation of all data structure.

## The code shows above will give the same output as showed in previous table.