Code Academy

Stack

Operations and applications

Lecture Goals

- Understand the meaning of **stacks**.
- Know the **operations** and **application** of stacks.
- Know the **implementation** of stacks.
- Know the **advantages and disadvantages** of stacks.

What is Stack?

- Abstract Data Type (ADT).
- Can be implemented using different languages.
- It is called stack because it has the similar operations as the real-world stacks: a pack of cards or a pile of plates, etc.
- The stack follows the LIFO (Last in - First out) structure where the last element inserted would be the first element deleted.

The stack operation

- Push ()
- Pop()
- Peek()
- isFull()
- isEmpty()

Code Academy

Insertion: push

- The *push* is an operation that inserts elements into the stack.
- The push algorithm:

  1) Start
  2) Add element to the stack (list).
  5) End

- To add an element to a stack (list) in python:
  use: list.append()

Deletion: pop()

- The *pop()* is a data manipulation operation which removes elements from the stack.
- The pop() algorithm:

  1) Start
  2) Remove element from the stack (list).
  5) End

- To remove an element from stack (list) in python use: list.pop()

Peek

- The *peak* is an operation retrieves the topmost element within the stack, without deleting it.
- The peak algorithm:

```
1) Start
2) return element at the top of the stack.
3) End
```

- To get the top most element in the stack using python use: list[-1]

full()

- The *full()* operation checks whether the stack is full. This operation is used to check the status of the stack with the help of top pointer.
- The full() algorithm:

```
1) Start
2) If the size of the stack is equal to the top position of the stack,
the stack is full. Return 1.
3) otherwise, return 0.
4) End
```

empty()

- The *empty()* operation verifies whether the stack is empty. This operation is used to check the status of the stack with the help of top pointer.
- The empty() algorithm:

```
1) Start
2) If the top value is -1, the stack is empty. Return 1.
3) otherwise, return 0.
4) End
```

Code Academy

Let's Code together ☺

python

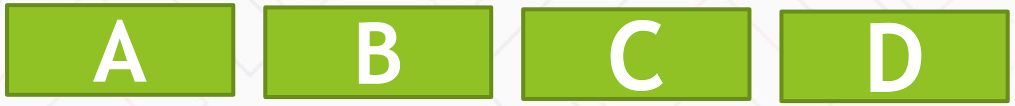
Stack Application

- Reverse a data.
- Evaluation of Arithmetic expression.
- Backtracking.
- Delimiter Checking
- Processing Function Calls

Reverse a data

- To reverse a given set of data, we need to reorder the data so that the first and last elements are exchanged, the second and second last element are exchanged, and so on for all other elements.

- For Example: the reversed string of “ABCD” is “DCBA”



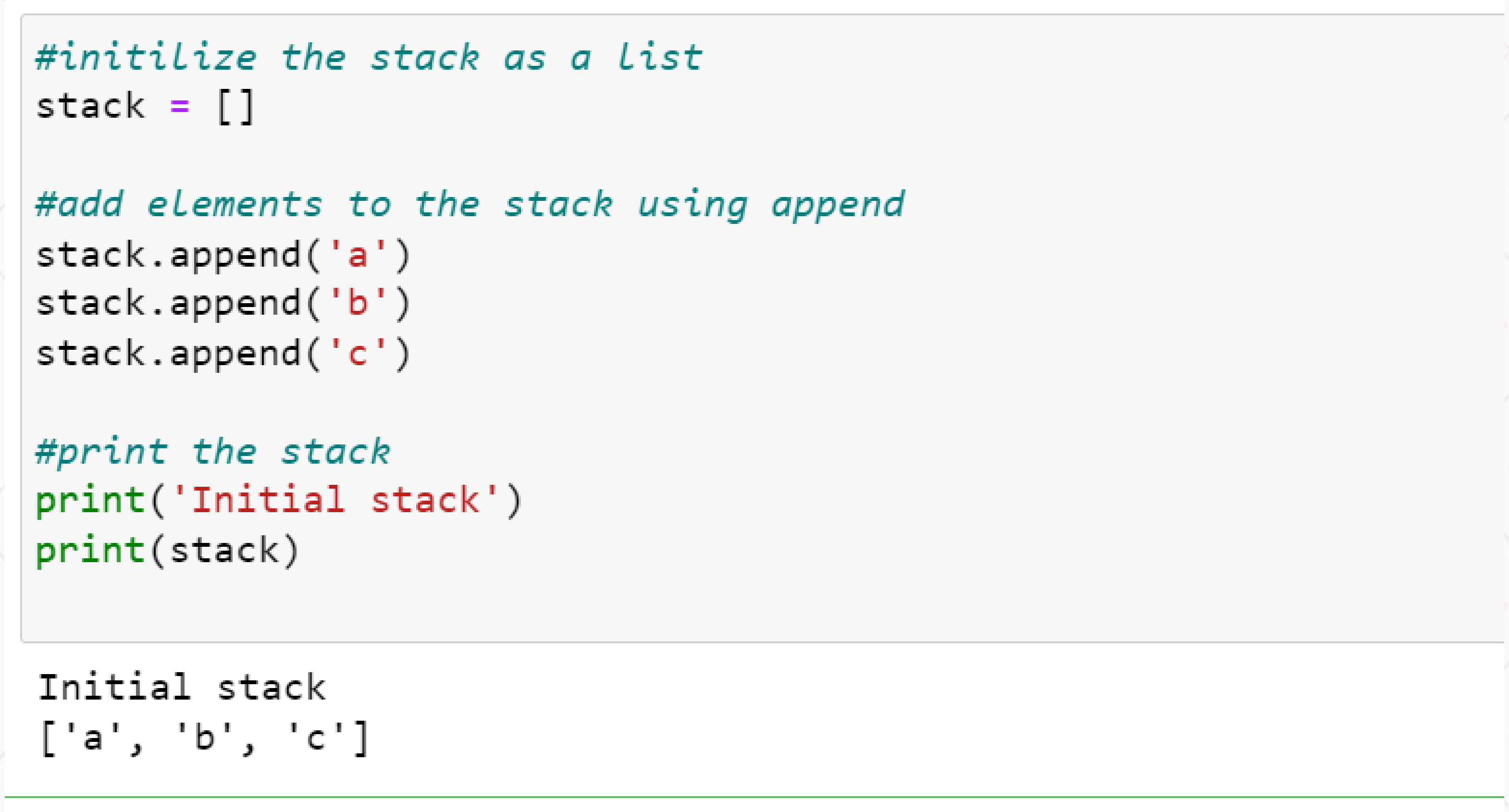
A	B	C	D

Code Academy

Implementation

How to implement stack in python?

push implementation



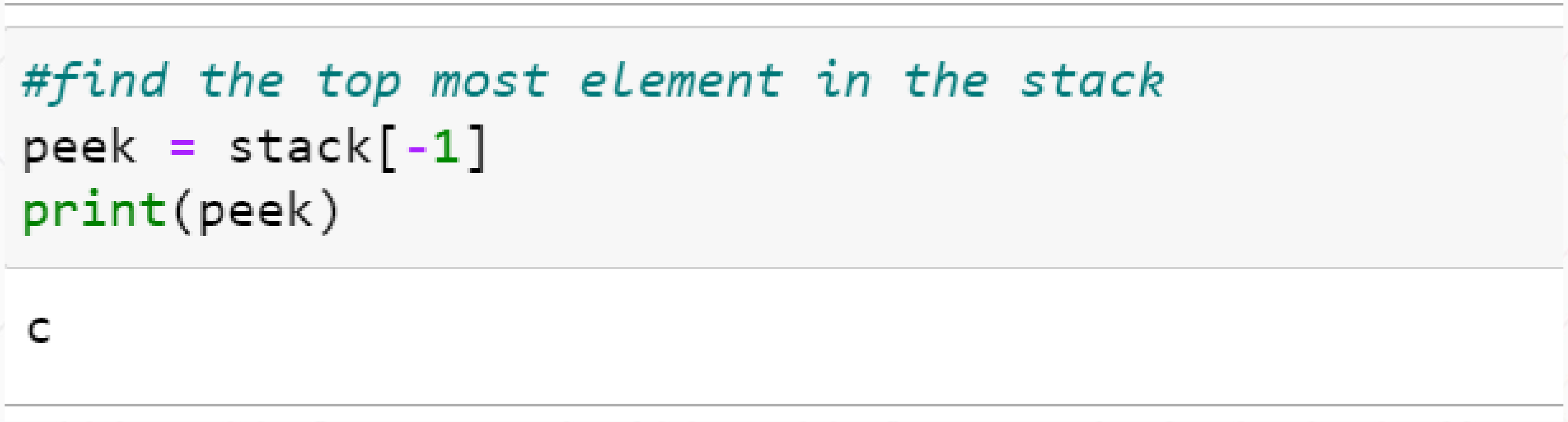
```
#initilize the stack as a list
stack = []

#add elements to the stack using append
stack.append('a')
stack.append('b')
stack.append('c')

#print the stack
print('Initial stack')
print(stack)
```

```
Initial stack
['a', 'b', 'c']
```

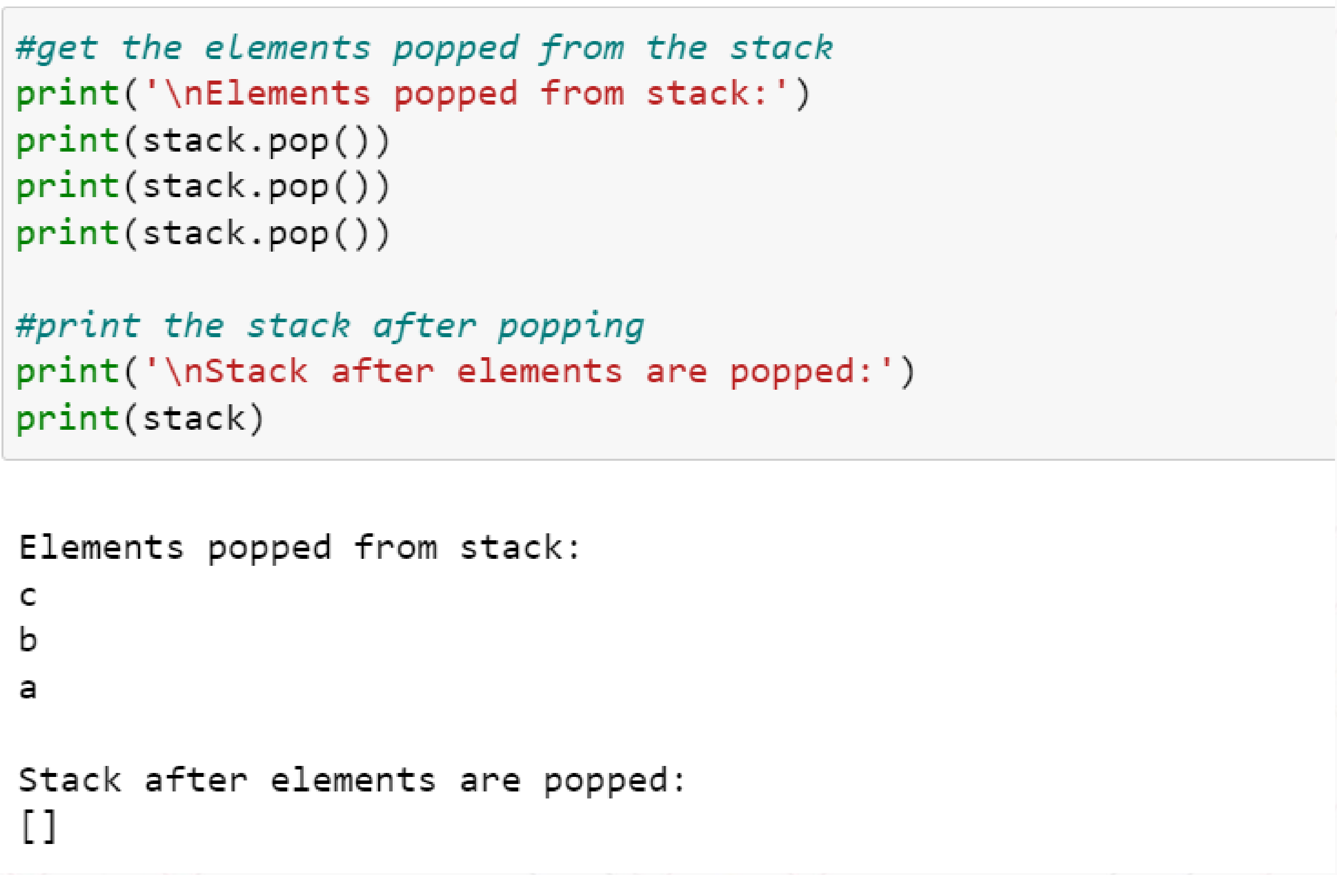
peek implementation



```
#find the top most element in the stack
peek = stack[-1]
print(peek)
```

```
c
```

pop implementation



```
#get the elements popped from the stack
print('\nElements popped from stack:')
print(stack.pop())
print(stack.pop())
print(stack.pop())

#print the stack after popping
print('\nStack after elements are popped:')
print(stack)
```

```

Elements popped from stack:
c
b
a

Stack after elements are popped:
[]
```

Advantages of stacks

- A Stack helps to manage the data in the ‘Last in First out’ method.
- When the variable is not used outside the function in any program, the Stack can be used.

Code Academy

Disadvantages of stacks

- It is difficult in Stack to create many objects as it increases the risk of the Stack overflow.
- It has very limited memory.
- In Stack, random access is not possible.

Code Academy

Let’s Code Together ☺

python

Code Academy

Exercise:

- Write a python class called Stack with the following functions:
  - push(element)
  - pop()
  - peek()
  - isEmpty()
  - get_stack()

Then, create an object of the class and do the following:
- check is the stack empty
- push the following elements into the stack (1,2,3,4,5)
- peek the topmost element is the stack.
- get the stack elements.
- delete two elements from the stack.
- show the remaining elements of the stack.