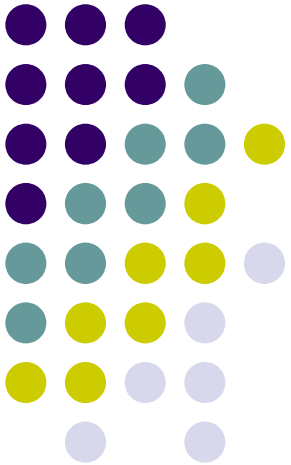
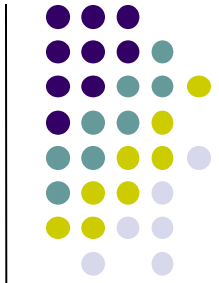
Στοίβα

Κ. Φλώρος



Εισαγωγή – Στοίβα

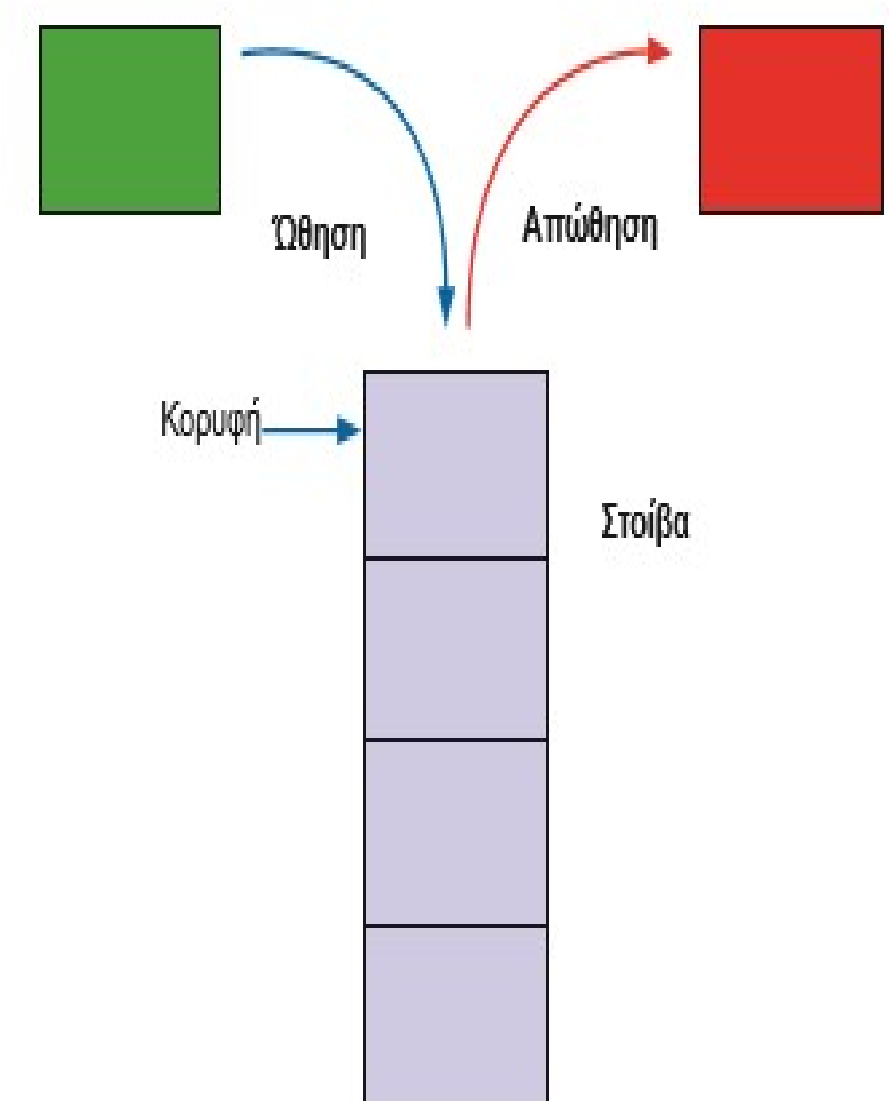


Η **στοίβα** είναι μια δομή δεδομένων τύπου λίστας στην οποία οι εισαγωγές και οι διαγραφές στοιχείων γίνονται από το πάνω άκρο μόνο.

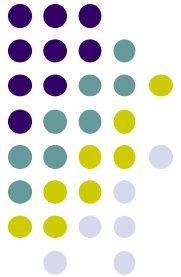
Στη λίστα αυτή το στοιχείο που προστέθηκε τελευταίο είναι και το πρώτο που θα εξαχθεί, έχουμε δηλαδή μια λειτουργία τύπου **lifo (last in first out)**, δηλαδή ο τελευταίος που εισέρχεται στη λίστα, είναι και ο πρώτος που θα εξαχθεί.

Οι λειτουργίες εισαγωγής και εξαγωγής είναι γνωστές στη βιβλιογραφία ως **ώθηση** και **απώθηση**.

Λειτουργίες – Ώθηση και Απώθηση

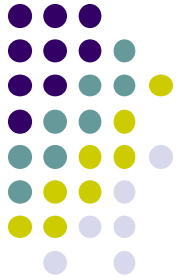


Βασικές λειτουργίες υλοποίησης στοίβας (1)



Όταν η στοίβα είναι άδεια, είναι προφανές ότι δεν μπορεί να γίνει απώθηση. Επομένως, όταν απωθούμε ένα στοιχείο από τη στοίβα, θα πρέπει προηγουμένως να έχουμε εξασφαλίσει ότι η στοίβα δεν είναι κενή. Για αυτό το λόγο, εκτός από την ώθηση και την απώθηση, πρέπει να υλοποιήσουμε και τον έλεγχο, αν η στοίβα είναι κενή.

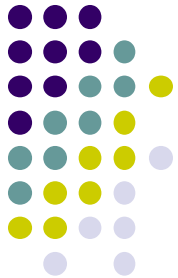
Βασικές λειτουργίες υλοποίησης στοίβας (2)



άρα οι βασικές λειτουργίες που πρέπει να υποστηρίζει η υλοποίηση μιας στοίβας είναι:

- **δημιουργία** μιας κενής στοίβας.
- **έλεγχος**, αν η στοίβα είναι κενή.
- **ώθηση** (προσθήκη) ενός στοιχείου στη στοίβα.
- **απώθηση** (διαγραφή) ενός στοιχείου από τη στοίβα.

Υλοποίηση της στοίβας



- Η δομή της στοίβας μπορεί να υλοποιηθεί στην python με μια λίστα στην οποία οι εισαγωγές και οι εξαγωγές στοιχείων γίνονται μόνο από το ένα άκρο είτε είναι στο πίσω μέρος της λίστας, είτε στο εμπρός μέρος της λίστας.

Υλοποίηση στοίβας με δύο τρόπους

Υλοποίηση Στοίβας σε Python με δύο τρόπους

```
def push(stack, item) :
    stack.append(item)
def pop(stack) :
    return stack.pop( )
def isEmpty(stack) :
    return len(stack) == 0
def createStack( ) :
    return [ ]
```

```
def push(stack, item) :
    stack.insert(0,item)
def pop(stack) :
    return stack.pop( 0 )
def isEmpty(stack) :
    return len(stack) == 0
def createStack( ) :
    return [ ]
```

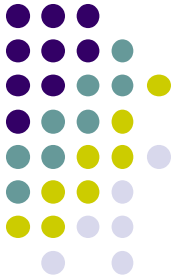
Οι δύο υλοποιήσεις που δίνονται παραπάνω, διαφέρουν μόνο ως προς το σημείο της λίστας στο οποίο γίνονται οι εισαγωγές/διαγραφές των στοιχείων.

Παράδειγμα

Να γραφεί πρόγραμμα το οποίο:

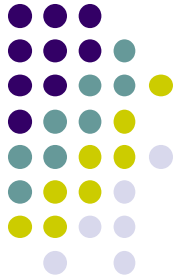
- να δέχεται από το χρήστη αριθμούς μέχρι να δοθεί το 0 και
- να τους εμφανίζει σε αντίστροφη σειρά από αυτή με την οποία δόθηκαν.
- Δηλαδή θέλουμε κάθε φορά να εμφανίσουμε πρώτο τον αριθμό που δόθηκε τελευταίος.

Παράδειγμα - Λύση



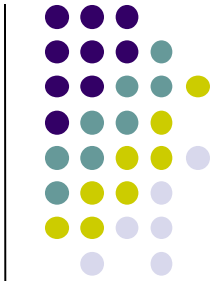
```
def createStack():
    return []
def push(stack,item):
    stack.append(item)
def pop(stack):
    return stack.pop()
def isEmpty(stack):
    return len(stack)==0
```

Παράδειγμα – Λύση (συνέχεια)



```
stack=createStack()
num=int(raw_input('Dose aritmo: '))
while num !=0:
    push(stack,num)
    num=int(raw_input('dose arithmo: '))
print stack
while not isEmpty(stack):
    num=pop(stack)
    print num
```

Παράδειγμα – Λύση (χωρίς συναρτήσεις)



```
Stack=[]
num=input('Δώσε αριθμό: ')
while num!=0:
    Stack.append(num)
    num=input('Δώσε αριθμό: ')
print Stack
while len(Stack)!=0:
    num=Stack.pop()
    print num
```

Άσκηση 1

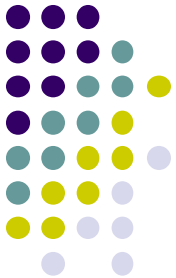
Να γραφεί πρόγραμμα το οποίο:

- να διαβάζει μια λέξη,
- να δημιουργεί μια στοίβα με όνομα LEXI,
- να τοποθετεί το κάθε γράμμα της λέξης στην παραπάνω στοίβα,
- και να εμφανίζει αντιστραμμένη την αρχική λέξη αφαιρώντας το κάθε γράμμα της από τη στοίβα

Άσκηση 1 - Λύση

```
def createStack():
    return []
def push(stack,item):
    stack.append(item)
def pop(stack):
    return stack.pop()
def isEmpty(stack):
    return len(stack)==0
```

Άσκηση 1 - Λύση (συνέχεια)



```
LEXI=createStack()
x=raw_input('Δώστε μια λέξη: '))
for i in x:
    push(LEXI, i)

print 'Τα γράμματα με αντίστροφη σειρά'
while not isEmpty(LEXI):
    y=pop(LEXI)
    print y
```

Άσκηση 2

Να γραφεί πρόγραμμα το οποίο:

- να διαβάζει 10 ονόματα ενός τμήματος και να τα τοποθετεί στη στοίβα stackONOMA,
- να εκτελεί τη λειτουργία της ώθησης και απώθησης επαναληπτικά ως εξής:

  να διαβάζει την επιλογή λειτουργίας, συγκεκριμένα “Ω” για ώθηση, “Α” για απώθηση και “Τ” για τέλος. Όταν εκτελείται η λειτουργία της ώθησης να διαβάζει ένα όνομα και να το τοποθετεί στη στοίβα, ενώ τη λειτουργία της απώθησης την εκτελεί αν υπάρχουν στοιχεία στη στοίβα. Η διαδικασία επαναλαμβάνεται μέχρι να διαβαστεί το γράμμα “Τ”, ή να αδειάσει η στοίβα.
- να εμφανίζει πόσες απωθήσεις έγιναν και πόσα ονόματα έχει μέσα η στοίβα αν δεν είναι άδεια

Άσκηση 2 - Λύση

```
def createStack():
    return []
def push(stack,item):
    stack.append(item)
def pop(stack):
    return stack.pop()
def isEmpty(stack):
    return len(stack)==0
stackONOMA=createStack()
for i in range(10):
    x=raw_input('Δώστεόνομα: ')
    push(stackONOMA, x)
```

Άσκηση 2 – Λύση (συνέχεια)

```
m=0
epil=raw_input('Δώστε επιλογή: ')
while epil !='T' and not isEmpty(stackONOMA):
    if epil=='Ω':
                x=raw_input('Δώστε όνομα: ')
                    push(stackONOMA, x)
    else:
                x=pop(stackONOMA)
        print x
        m=m+1
    epil=raw_input('Δώστε επιλογή: ')
print m, ' απωθήσεις '
if isEmpty(stackONOMA):
    print 'ΑΔΕΙΑ ΣΤΟΙΒΑ'
else: print 'Η Στοίβα έχει μέσα: ', len(stackONOMA)
```