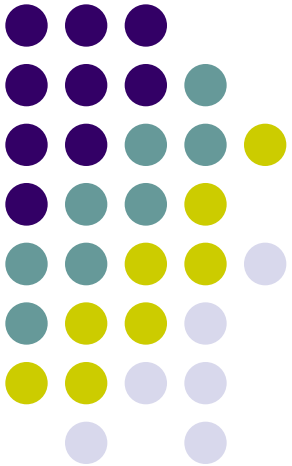
8. Δομές Δεδομένων II

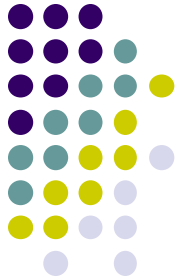
Συμβολοσειρές(strings)
Γ’ Πληροφορική



Εισαγωγή

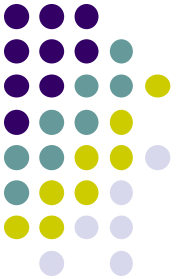
- Μια **δομή δεδομένων** είναι ένα σχήμα οργάνωσης σχετικών μεταξύ τους στοιχείων δεδομένων.
- Οι *ενσωματωμένες* **δομές δεδομένων** της Python που θα εξετάσουμε στο κεφάλαιο αυτό είναι τα **αλφαριθμητικά**, οι **λίστες**, οι **πλειάδες** και τα **λεξικά.**
- Επίσης με χρήση των ενσωματωμένων δομών της Python θα παρουσιάσουμε υλοποιήσεις σύνθετων δομών δεδομένων, όπως είναι η στοίβα και η ουρά.
- Η **λίστα** αποτελεί το δομικό λίθο σχεδόν όλων των δομών δεδομένων στην Python, ενώ η δομή του **λεξικού** χρησιμεύει όταν θέλουμε να κάνουμε γρήγορη αναζήτηση και ανάκληση πληροφορίας σχετική με κάποια συμβολοσειρά.

Στατικές και Δυναμικές Δομές Δεδομένων (Επανάληψη από Β' Λυκείου)



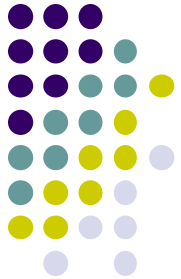
- Οι δομές δεδομένων χωρίζονται σε δυο βασικές κατηγορίες τις **στατικές** και **τις δυναμικές**, ανάλογα με τη φάση της ανάπτυξης του προγράμματος κατά την οποία καθορίζεται το μέγεθος της δομής.
- Το μέγεθος των στατικών δομών καθορίζεται κατά τη φάση του προγραμματισμού, δηλαδή κατά τη φάση της συγγραφής του προγράμματος. Είναι φανερό ότι
- Η πιο γνωστή **στατική** δομή δεδομένων είναι ο **πίνακας**. Στην Python στατικές δομές δεδομένων είναι οι **πλειάδες** (tuples).

Δυναμικές δομές δεδομένων(1) (Επανάληψη από Β' Λυκείου)



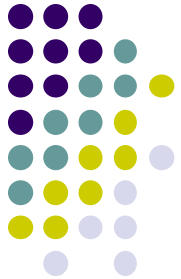
- Από την άλλη οι **δυναμικές** δομές δεδομένων στηρίζονται σε μια λειτουργία που λέγεται **δυναμική εκχώρηση μνήμης**. Κατά τη δυναμική εκχώρηση μνήμης, το πρόγραμμα μπορεί να ζητήσει από το λειτουργικό σύστημα όση μνήμη απαιτείται για τη δημιουργία της δομής δεδομένων κατά την εκτέλεση του προγράμματος.
- Οι δυναμικές δομές δεδομένων μπορούν να μεταβάλλουν το μέγεθός τους, προσθέτοντας ή αφαιρώντας αντικείμενα.

Δυναμικές δομές δεδομένων (2) (Επανάληψη από Β' Λυκείου)



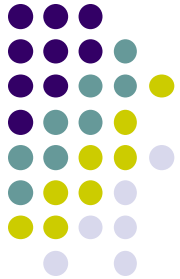
- Η πιο γνωστή δυναμική δομή δεδομένων είναι η **λίστα** η οποία είναι και η βασική δομή δεδομένων της python.
- Ουσιαστικά η λίστα της python δεν είναι τίποτα παραπάνω από ένα δυναμικό πίνακα, δηλαδή έναν πίνακα του οποίου το μέγεθος μπορεί να αυξομειώνεται κατά την εκτέλεση του προγράμματος.

Συμβολοσειρές(strings)



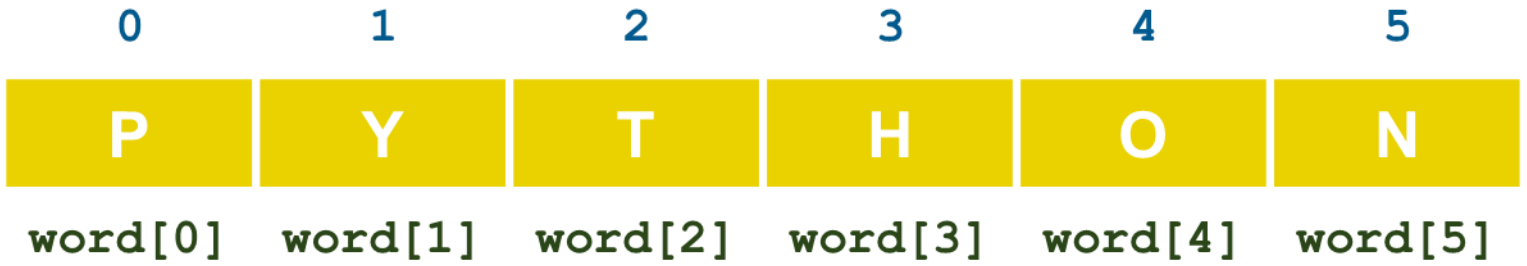
- Τα **αλφαριθμητικά** ή **συμβολοσειρές** είναι ακολουθίες από χαρακτήρες που έχουν σταθερό μέγεθος και μη μεταβαλλόμενα περιεχόμενα.
- Δηλαδή, δεν μπορούμε να προσθέσουμε ή να αφαιρέσουμε χαρακτήρες, ούτε να τροποποιήσουμε τα περιεχόμενα του αλφαριθμητικού. Γι αυτό λέμε ότι η δομή ανήκει στις μη μεταβαλλόμενες(immutable) δομές της Python.
- Μια συμβολοσειρά την ορίζουμε με εισαγωγικά αμφίπλευρα μονά ή διπλά.

Συμβολοσειρές - Παράδειγμα



- Η αρίθμηση των χαρακτήρων σε ένα αλφαριθμητικό ξεκινάει από το 0. Για παράδειγμα:

αν **word='PYTHON'**



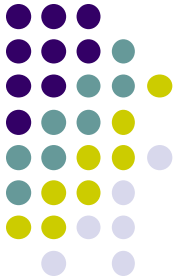
- Μια συμβολοσειρά την ορίζουμε με εισαγωγικά αμφίπλευρα μονά ή διπλά.

Άσκηση 1- Παράδειγμα

- a="Καλημέρα "
- b="ηλιόλουστη μέρα"
- z=a+b
- d=a*2
- print a
- print b
- print z
- print d

Ο τελεστής + όταν εφαρμόζεται σε αντικείμενα τύπου string, έχει σαν αποτέλεσμα τη συνένωση τους σε μια συμβολοσειρά

Άσκηση 2- Παράδειγμα



- a="Καλημέρα "
- b="ηλιόλουστη μέρα"
- z=a+b
- d=a*2
- print a
- print b
- print z
- print d
- print len(b)
- print b[1]
- print b[11:15]

Άσκηση 3 - Παράδειγμα

```
word="python"
letter=word[1]
print letter
print word[0]
print len(word)
print len('12345678')
```

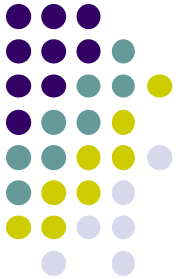
Η συνάρτηση len επιστρέφει το μήκος, δηλαδή το πλήθος των χαρακτήρων

Άσκηση 3α - Παράδειγμα

```
word="python"
print word.upper()
```

Άσκηση 3b - Παράδειγμα

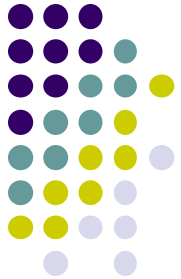
```
print "monty".upper()
```



Άσκηση 3c - Παράδειγμα

```
print "PASCAL".lower()
```

Συμβολοσειρές–Συνάρτηση str

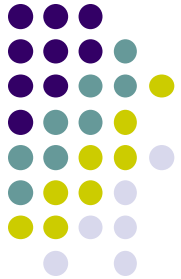


- Η συνάρτηση **str** μετατρέπει μια τιμή σε συμβολοσειρά.

Παράδειγμα:

```
>>> str(28) == "28"
True
```

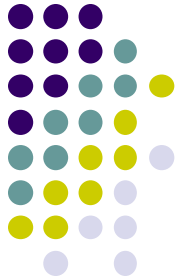
Συμβολοσειρές–Συνάρτηση int



- Η συνάρτηση **int** μετατρέπει ένα αλφαριθμητικό στον ακέραιο αριθμό που αναπαριστά.
- Παράδειγμα:

```
>>> print int("6") + 4
10
```

Συμβολοσειρές–Έλεγχος ύπαρξης



Ο τελεστής ***in*** *ελέγχει αν ένα αντικείμενο ανήκει σε ένα σύνολο αντικειμένων.* Δεδομένου ότι οι συμβολοσειρές μπορούν να θεωρηθούν ως σύνολα χαρακτήρων, μπορούμε να τον χρησιμοποιήσουμε:

Παραδείγματα:

```
>>> "Py" in "Python"
True
>>> "a" in "Python"
False
>>> "a" not in "Python"
True
```

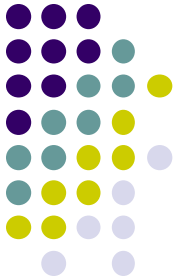
Άσκηση 4 - Παράδειγμα

- Να γραφεί πρόγραμμα το οποίο:
- Α) Να διαβάζει μια λέξη
- Β) και να εμφανίζεται σε κάθε γραμμή ένα γράμμα αυτής της λέξης.

Άσκηση 4 - Λύση

```
a=raw_input('Δώσε μια λέξη: ')
num=len(a)
for i in range(num):
    print a[i]
```

Άσκηση 5 - Παράδειγμα



- Να γραφεί πρόγραμμα το οποίο:
- Α) Να διαβάζει μια λέξη
- Β) και να εμφανίζει σε κάθε γραμμή ένα γράμμα αυτής της λέξης αντίστροφα.

Άσκηση 5 - Λύση

```
a=raw_input('Δώσε μια λέξη: ')
num=len(a)
for i in range(num-1,-1,-1):
    print a[i]
```

Άσκηση 6- Παράδειγμα

- Να γραφεί πρόγραμμα το οποίο:
- Α) Να διαβάζει μια λέξη με πεζά
- Β) και να την εμφανίζει με κεφαλαία.

Άσκηση 6 - Λύση

```
a=raw_input('Δώσε μια λέξη: ')
print a.upper()
```

Άσκηση 7- Παράδειγμα

- Να γραφεί πρόγραμμα το οποίο:
- Α) Να διαβάζει μια λέξη με κεφαλαία
- Β) και να την εμφανίζει με πεζά.

Άσκηση 7 - Λύση

```
a=raw_input('Δώσε μια λέξη: ')
print a.lower()
```

Άσκηση 8- Παράδειγμα

- Να γραφεί πρόγραμμα το οποίο:
- Α) Να διαβάζει μια λέξη με πεζά ή κεφαλαία
- Β) και να την εμφανίζει με κεφαλαία ή πεζά αντίστροφα.

Άσκηση 8 - Λύση

```
a=raw_input('Δώσε μια λέξη: ')
print a.swapcase()
```

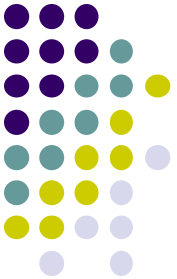
Άσκηση 9

- Να γραφεί πρόγραμμα το οποίο:
- Α) Να διαβάζει μια πρόταση
- Β) και να δημιουργεί και να εμφανίζει μια νέα συμβολοσειρά όμοια με την αρχική, αλλά χωρίς κενά ανάμεσα στις λέξεις.

Άσκηση 9 - Λύση

```
a=raw_input('Δώσε πρόταση:')
x=''
for i in a:
    if i!=' ':
        x=x+i
print x
```

Άσκηση 9 – Λύση (με συνάρτηση)



```
def  trimSpaces(sentence):
    result=''
    for i in sentence:
        if i!=' ':
            result=result+i
    return result
a=raw_input('Δώσε πρόταση: ')
print trimSpaces(a)
```

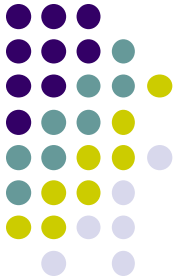
Άσκηση 10

- Να γραφεί πρόγραμμα το οποίο:
- Α) Να διαβάζει μια πρόταση
- Β) και να υπολογίζει και να εμφανίζει πόσα φωνήεντα έχει η πρόταση.

Άσκηση 10 - Λύση

```
a=raw_input('Δώσε πρόταση:')
vowels='AEIOUaeiou'
count=0
for letter in a:
    if letter in vowels:
        count=count+1
print "Το πλήθος είναι: ", count
```

Άσκηση 10 – Λύση (με συνάρτηση)



```
def countvowels(word)
    vowels='AEIOUaeiou'
    count=0
    for letter in word:
        if letter in vowels:
                count=count+1
    return count
a=raw_input('Δώσε πρόταση:')
print "Το πλήθος είναι: ", countvowels(a)
```

Άσκηση 11

- **Να γραφεί πρόγραμμα το οποίο:**
- Να διαβάζει λέξεις από το πληκτρολόγιο,
- να μετράει και να εμφανίζει πόσες λέξεις ξεκινούν από το γράμμα Α (κεφαλαίο ή μικρό).
- Όταν δοθεί λέξη που να τελειώνει σε «Ω» ή «ω», τότε το πρόγραμμα να τερματίζει.

Άσκηση 11 - Λύση

```
count=0
lexi=raw_input('Δώσε μια λέξη: ')
while lexi[-1]!='ω' and lexi[-1]!='Ω':
    if lexi[0]=='A' or lexi[0]=='α':
        count=count+1
    lexi=raw_input('Δώσε μια λέξη: ')
print 'Το πλήθος είναι: ' , count
```