

# Εισαγωγή στην Python

Σημειώσεις μαθήματος

## ΜΑΘΗΜΑ 2

Δρ.Ν.Μανδέλλος, Δρ.Φ.Δογάνης

# Δομές δεδομένων – Data structures

Εισαγωγή στις Λίστες (Lists)

Μια λίστα είναι μια συλλογή στοιχείων που **μπορεί να αλλάξει** (mutable) και μπορεί να περιέχει διαφορετικούς τύπους δεδομένων.

**Παράδειγμα Λίστας:**

```
# Δημιουργία λίστας με αριθμούς και strings
```

```
devices = ["Αποστακτική στήλη", "Χημικός αντιδραστήρας", "Διαχωριστής"]
```

```
print(devices)
```

```
# Πρόσβαση σε στοιχεία λίστας
```

```
print(devices[0]) # Πρώτο στοιχείο
```

```
print(devices[-1]) # Τελευταίο στοιχείο
```

# Δομές δεδομένων – Data structures

Βασικές Λειτουργίες στις Λίστες

- Προσθήκη στοιχείων: `append()`
- Αφαίρεση στοιχείων: `remove()`
- Μήκος λίστας: `len()`

**Παράδειγμα:**

*# Προσθήκη στοιχείου*

```
devices.append("Χημικός αντιδραστήρας")
```

```
devices.append(" Αποστακτική στήλη")
```

```
print(devices)
```

*# Αφαίρεση στοιχείου*

```
devices.remove("Αποστακτική στήλη")
```

```
print(devices)
```

*# Μήκος λίστας*

```
print("Το μήκος της λίστας είναι:", len(devices))
```

# Δομές δεδομένων – Data structures

## Πλειάδες (Tuples)

Οι πλειάδες είναι παρόμοιες με τις λίστες, αλλά είναι **αμετάβλητες** (immutable). Αυτό σημαίνει ότι δεν μπορούμε να αλλάξουμε τα στοιχεία τους μετά τη δημιουργία τους.

### Παράδειγμα Πλειάδας:

```
# Δημιουργία πλειάδας  
coordinates = (10, 20)  
print(coordinates)  
# Πρόσβαση σε στοιχεία  
print(coordinates[0])
```

Οι πλειάδες χρησιμοποιούνται όταν δεν θέλουμε να αλλάξουμε τα δεδομένα (π.χ., συντεταγμένες).

# Δομές δεδομένων – Data structures

Σε μια πλειάδα μπορώ να ορίσω ανάμεικτους τύπους μεταβλητών:

## Παραδείγματα:

```
address1 = ("Πατησίων", 128, "104-81")
```

```
address2 = ("Σταδίου", 33, "104-18")
```

```
print(address1)
```

```
print(address2)
```

```
print(address1[0])
```

*Μπορούμε να κάνουμε πράξεις με αυτές:*

```
print("Οδός {} αριθμός {} ΤΚ {}".format(address1[0], address1[1], address1[2]))
```

```
print(address1[0]+ " και " + address2[0])
```

```
print(address1[1]+address2[1])
```

# Δομές δεδομένων – Data structures

## Άσκηση:

Έστω ότι έχουμε 4εις τάξεις A1-A4 στον πρώτο όροφο ενός σχολείου, 3εις τάξεις B1-B2 στον δεύτερο όροφο ενός σχολείο, μια τάξη στον τρίτο όροφο C1 και καμία τάξη στον τέταρτο. Να δημιουργηθεί μια λίστα που να συνδέει όλους τους όροφους με τη λίστα των τάξεων.

## Λύση:

Δημιουργώ λίστες `classes1` – `classes4` για τις τάξεις σε κάθε όροφο:

```
classes1 = ["A1", "A2", "A3", "A4"]
```

```
classes2 = ["B1", "B2", "B3"]
```

```
classes3 = ["C1"]
```

```
classes4 = []
```

# Δομές δεδομένων – Data structures

Δημιουργώ πλειάδες όροφος – λίστα τάξεων ανά όροφο:

```
floor1 = (1, classes1)
```

```
floor2 = (2, classes2)
```

```
floor3 = (3, classes3)
```

```
floor4 = (4, classes4)
```

Δημιουργώ την μεταβλητή *school* ως λίστα όλων των ορόφων.

```
school = [floor1, floor2, floor3, floor4]
```

Εκτυπώσεις

```
print(school)
```

```
print(school[0])
```

```
print(school[0][1])
```

```
print(school[0][1][0])
```

# Δομές δεδομένων – Data structures

## Ευρετήριο (Dictionaries)

Τα ευρετήρια είναι συλλογές δεδομένων με τη μορφή **κλειδί-τιμή** (key-value pairs). Είναι χρήσιμα για την αποθήκευση ζευγών δεδομένων, όπως για παράδειγμα ονόματα και τηλέφωνα.

## Παράδειγμα Λεξικού:

```
# Δημιουργία λεξικού
```

```
phonebook = {"Μαρία": "6941234567", "Γιάννης": "6939876543"} print(phonebook)
```

```
# Πρόσβαση σε τιμή μέσω κλειδιού
```

```
print(phonebook["Μαρία"])
```

```
# Προσθήκη νέου ζεύγους
```

```
phonebook["Νίκος"] = "6912345678"
```

```
print(phonebook)
```



# Δομές δεδομένων – Data structures

## **Format** δεδομένων σε ευρετήριο

{ }: χρησιμοποιούμε τις αγκύλες για να ορίσουμε την αρχή και το τέλος ενός ευρετηρίου.

*Ορισμός μελών ενός ευρετηρίου. Κάθε μέλος έχει ένα όνομα και μια τιμή και ορίζεται ως:*

*“[όνομα πεδίου]” : [τιμή πεδίου]*

### *Παρατηρήσεις:*

- Αν η τιμή του πεδίου είναι αριθμός ή λογική μεταβλητή δεν χρειάζονται εισαγωγικά. πχ. { “productPrice” : 15.321 }*
- Η τιμή του πεδίου μπορεί να είναι επίσης μια άλλη σύνθετη δομή: πχ. πλειάδες { “127.11.35” : (10, 2.4, 12.4), “127.11.36” : (20, 4.8, 24.8) }, δηλαδή έχουμε αντιστοίχιση κωδικό προϊόντος με (τιμή προϊόντος χωρίς φόρο, ΦΠΑ, τιμή λιανικής)*
- Ακολουθεί το ίδιο format όπως στον δημοφιλή τύπο JSON για μεταφορά δεδομένων διαδικτυακά, το JSON (JavaScript Object Notation) χρησιμοποιείται για τη μεταφορά δεδομένων με το πρωτόκολλο http.*

# Δομές δεδομένων – Data structures

## Ευρετήριο (Dictionaries) αριθμών

*# Δημιουργία λεξικού*

```
chem_dic = {"H2O": 18, "H2": 2, "N2": 28}
```

```
print(chem_dic)
```

*# Πρόσβαση σε τιμή μέσω κλειδιού*

```
print(chem_dic["H2O"])
```

*# Προσθήκη νέου ζεύγους*

```
chem_dic["H2O"] = 18
```

```
print(chem_dic)
```

# Δομές δεδομένων – Data structures

## Σύνθετο παράδειγμα dictionary

*# Δημιουργία λεξικού με πλειάδες, κάθε πλειάδα δείχνει ΜΒ, Σημείο βρασμού, Κατάσταση σε Κ.Σ.*

```
chem_dic = {"H2O": (18, 100, "υγρό"), "H2": (2, -252.9, "αέριο")}
```

*# Δες το στοιχείο "H2O"*

```
print(chem_dic["H2O"])
```

*# Πρόσθεσε το στοιχείο "N2" στο λεξικό --- έχω γράψει λάθος την κατάσταση*

```
chem_dic["N2"] = (28, -195.8, "στερεό")
```

```
print(chem_dic["N2"])
```

```
print(chem_dic)
```

*# Πως διορθώνω;*

```
chem_dic["N2"] = (18, -195.8, "αέριο")
```

```
print(chem_dic)
```

# Δομές δεδομένων – Data structures

## Σύνθετο παράδειγμα dictionary

*# Τι θα γίνει αν προσθέσω το Cl, για το οποίο δεν έχω στοιχεία;*

```
chem_dic["Cl2"] = ("αέριο")
```

```
print(chem_dic)
```

*# Πως θα αφαιρέσω στοιχείο;*

```
chem_dic.pop("Cl2")
```

```
print(chem_dic)
```

# Δομές δεδομένων – Data structures

## Statement in:

*Χρησιμοποιούμε το **in** για να μπορέσουμε να αναζητήσουμε εγγραφές που υπάρχουν σε μια λίστα, πλειάδα, λεξικό ή σε ένα `string`, όπως φαίνεται στα παρακάτω παραδείγματα:*

```
list1= [1,2,3,4,5]  
string1= "My name is Peter"  
tuple1=(11,22,33,44)
```

```
print(5 in list1) #True  
print("is" in string1) #True  
print(88 in tuple1) #False
```

# Δομές δεδομένων – Data structures

Άσκηση: Δημιουργία Λίστας Χημικών Στοιχείων

Γράψτε ένα πρόγραμμα που:

1. Δημιουργεί μια κενή λίστα χημικών στοιχείων.
2. Ζητά από τον χρήστη να προσθέσει 3 στοιχεία.
3. Εμφανίζει τη λίστα στο τέλος.

**Υπόδειξη:** Χρησιμοποιήστε τη `append()` για την προσθήκη στοιχείων.

**Παράδειγμα:**

```
# Άσκηση: Λίστα χημικών στοιχείων  
chem_list = []  
# Προσθήκη 3 χημικών στοιχείων από τον χρήστη  
for i in range(3):  
    item = input("Προσθέστε ένα χημικό στοιχείο: ")  
    chem_list.append(item)  
print("Η λίστα των στοιχείων είναι: ", chem_list)
```

# Άσκηση: Δομές δεδομένων

Άσκηση: Να φτιαχτεί ένα λεξικό που θα έχει μια λίστα από χημικά στοιχεία συνδεδεμένα με το μοριακό τους βάρος.

Στη συνέχεια να δωθεί σε μια μεταβλητή name τιμή από το χρήστη χρησιμοποιώντας το input, και σε μια άλλη μεταβλητή η τιμή του MB που αντιστοιχεί.

- a. Να γίνει έλεγχος αν είναι ήδη καταχωρημένο το στοιχείο αυτό
- b. Να προστεθεί το στοιχείο αυτό στο λεξικό
- c. Να γίνει μεταβολή της τιμής του

# Άσκηση: Δομές δεδομένων

+ Code + Text

✓  
9s

```
chem_dic = {"H2": 2, "N2": 28, "O2": 32}
# Ερώτημα a.
name = input("Give me a new element name: ")
value = float(input("Give me MB for element " + name + ": "))
exists = name in chem_dic
print("Exists ? ", exists)
# Ερώτημα b.
chem_dic[name] = value
print("chem_dic = ", chem_dic)
# Ερώτημα c.
value = float(input("Alter value for element " + name + ": "))
chem_dic[name] = value
print("chem_dic = ", chem_dic)
```

```
↳ Give me a new element name: H2O
Give me MB for element H2O: 18
Exists ? False
chem_dic = {'H2': 2, 'N2': 28, 'O2': 32, 'H2O': 18.0}
Alter value for element H2O: 18.01
chem_dic = {'H2': 2, 'N2': 28, 'O2': 32, 'H2O': 18.01}
```