



Ανακεφαλαίωση

Final Part



Σου έχει τύχει ποτέ κάποια περίπτωση όπου θα  
κάνεις κάτι μόνο αν ισχύει κάτι άλλο πρώτα;

Εδώ αυτό το λέμε “Δομή Επιλογής”

Και πώς δουλεύει αυτό στον προγραμματισμό;

Στον προγραμματισμό έχουμε ειδικές δομές (if) οι οποίες ελέγχουν αν μια συνθήκη είναι αληθής. Ενώ, αν δεν ισχύει κάτι τέτοιο, η συνθήκη είναι ψευδής

```
vathmos_kourashs = 7 # poso kourasmenos eimai apo to 1 mexri to 10?  
  
if vathmos_kourashs >= 6:  
    print("as kanw ena dialeimma")  
  
if vathmos_kourashs < 4:  
    print("exw toso pollh energeia, prepei kati na kanw!")  
  
# ti tha emfanisei o parapanw kwdikas?
```

Αυτό είναι όλο όμως; δεν γίνεται ο κώδικας πολύ μονότονος;

- ◆ if...else...
- ◆ if...elif...else...
- ◆ if...if... (Εμφωλευμένα if)

Παρακάτω μπορούμε να τα δούμε λίγο πιο αναλυτικά μαζί με κάποια παραδείγματα.

If...else...

Ουσιαστικά είναι σαν την απλή if, απλά έχεις ακόμα ένα κομμάτι κώδικα, σε περίπτωση που θες κάτι συγκεκριμένο αν η συνθήκη βγεί ψευδής

```
variemai = True  
  
if variemai:  
    print("as fugw, yolo!")  
else:  
    print("ma ti wraia poy einai edw, pernew kala kai mathainw tautoxrona!")
```

If...elif...else

Είναι το ίδιο ακριβώς με την if...else... απλά εδώ είσαι έτοιμος για όλα

```
moneyz = 2500

if moneyz > 500000:
    print("ferrari")
elif moneyz > 4000:
    print("mini cooper")
else:
    print("palio, kalo, lewforeio")
```

If...if...

Έχει τύχει σε όλους εκεί που νομίζουμε ότι τελείωσαμε με κάτι και επιτέλους θα γευτούμε τους καρπούς της προσπάθειάς μας, έρχεται η πραγματικότητα και σε χτυπάει στη μούρη, έχει άλλο τόσο

Double the ifs, double the fun!

```
anoixto_kulikeio = True
kalos_kafes = True
energy_drinks = False

if anoixto_kulikeio:
    if kalos_kafes:
        print("ena cappuccino me lmish dosh espresso, 2 koutalies maurh zaxarh, afrogala
            xtuphmeno sto xeri kai ligo tsimphmenh th kanela")
    elif not energy_drinks:
        print("mia tsikla parakalw")
else:
    print("kakh sunennohsh")
```

# Επανάληψη





Όλοι ξέρουμε τι είναι μια επανάληψη, μια σειρά  
απο βήματα που γίνεται ξανά και ξανά μέχρι  
να γίνει κάτι που θα σπάσει αυτό το μοτίβο  
(πχ ο βηματισμός μας, εισπνοή-εκπνοή)

Έτσι υπάρχει και η “Δομή Επανάληψης” εδώ

Πώς/Γιατί όμως θα τη χρησιμοποιήσω  
σε ένα πρόγραμμα;

Η δομή αυτή είναι πολύ χρήσιμη σε προγράμματα τα  
οποία εκτελούν την ίδια πράξη όσο η συνθήκη μας  
είναι αληθής. Γλιτώνουμε έτσι και γραμμές κώδικα και  
πολύ χρόνο

Πχ1 Μέχρι να τελειώσει το παγωτό, τρώγε

Πχ2 Όσο έχω φωνή, θα σου τραγουδάω

Ναί αλλά πώς το αναπαριστώ όλο αυτό σε κώδικα;

- ◆ For loop
- ◆ While loop
- ◆ Εμφωλευμένα loops

Ακολουθούν περισσότερα παραδείγματα με κακά αστεία.

## Τα δύο είδη επαναλήψεων

### For Loop

Σε ένα for loop ελέγχεται αρχικά μια συνθήκη. Έπειτα, **για (for)** κάθε φορά που η συνθήκη είναι αληθής, εκτέλεσε τον κώδικα που υπάρχει στη δομή

### While Loop

Σε ένα while loop, ελέγχεται αρχικά μια συνθήκη, όπως και στη for. **Όσο (while)** αυτή η συνθήκη είναι αληθής, εκτέλεσε τον κώδικα που υπάρχει στη δομή

## Τα δύο είδη επαναλήψεων 2

### For Loop

```
moneyz = 0

for wres_ergasias in range(20):
    print("ligo akoma kai meta mporw na paw spiti na dw metabots")
    moneyz += 7

print("\n" + str(moneyz))
```

### While Loop

```
kafes = 10 # poso polu kafe exw?

while kafes > 0:
    print("as piw ligo apo to dwro tou theou pros tous anthrwpous")
    kafes -= 1
    print("posos exei meinei akoma? hmm.. " + str(kafes) + "\n")

print("wp! fainetai prepei na paw na parw ena kafe!")
```

Μα δάσκαλε, τι είναι το *in range* που είχε η `for`;  
Γιατί δεν αλλάζαμε τον μετρητή;

το *in* είναι μια ειδική λέξη που κάνει την μεταβλητή  
πίσω της να μετατρέπεται σε χαμαιλέοντας!  
Ενώ το *range* σου δίνει ένα πλήθος αριθμών,

Μα δάσκαλε, τι είναι το *in range* που είχε η `for`;  
Γιατί δεν αλλάζαμε τον μετρητή;

Πιο συγκεκριμένα, το *range* θα αρχίσει να μετράει για  
τόσους αριθμούς όσο το νούμερο στην παρένθεση

Μα δάσκαλε, τι είναι το *in range* που είχε η *for*;  
Γιατί δεν αλλάζαμε τον μετρητή;

Το *in* όπως είπαμε, μετατρέπει την προηγούμενη  
μεταβλητή σε χαμαιλέοντα, δηλαδή παίρνει ό,τι  
υπάρχει μπροστά του και το βάζει στη μεταβλητή  
πίσω του.



## Range example

```
for dekalapta in range(10):  
    print("poso anupomoneis na kanoume dialeimma meta apo auto to dekalapto?")  
    if dekalapta < 5:  
        print("oxi polu, antexw akoma, exei endiaferon, " + str(dekalapta) + "/10\n")  
    elif dekalapta < 8:  
        print("ntaksei, thelw na kanw ena dialeimma alla yolo, " + str(dekalapta) + "/10\n")  
    else:  
        print("niwthw nikhmenos, thelw to arkoudaki mou, " + str(dekalapta)) + "/10\n")
```

*In example*

```
favorite_artists = ["eminem", "50 cent", "Bobby Tarantino", "pantelidis?" , "Steven Cooper"]  
  
for artist in favorite_artists:  
    print(artist)  
  
# ti einai auto to favorite_artists? de moiazei me kanonikh metavlhth..  
# skefteite to kai tha to analusoume se ligo
```

Τα δύο είδη επαναλήψεων 3  
Η επιστροφή της επανάληψης

Μη ξεχνάμε όμως και την νούμερο 1 πηγή των  
πονοκεφάλων των προγραμματιστών

Μπορούμε να έχουμε δομή επανάληψης μέσα σε μια  
άλλη δομή επανάληψης

Πχ Όσο διαρκεί το μάθημα, για κάθε άτομο που  
μιλάει, θα βάζω μια μονόωρη...έκαστος!

## Εμφωλευμένα loops πχ1

```
counter = 0

for i in range(10):
    for j in range(10):
        if i > j:
            counter += 1
            print(counter)

# τι τιμή πιστεύετε θα έχει ο counter όταν τελειώσει το programma?
```

## Εμφωλευμένα loops πχ2

```
rich = False
dead = False
grind = 0

while not rich and not dead:
    print("get rich or die tryin'")
    for attempt in range(10):
        print("it's all about the grind")
        grind += 10

    if grind > 80:
        rich = True

# se ti katastash vriskomai meta to programma?
```

# Λίστες

Τι είναι λίστα; πως τη δημιουργώ και σε τι χρησιμεύει;

Πώς μπορώ να χειριστώ και να δαμάσω μια λίστα;



Η λίστα είναι μια οντότητα η οποία περιέχει  
έναν αριθμό στοιχείων που συνδέονται μεταξύ  
τους (to-do list, λίστα για ψώνια, λίστα του  
σίντλερ, λυσ' τα κορδόνια σου, Liss Taylor)

Πώς αναπαριστώ μια λίστα όμως;

Πώς δημιουργώ μία λίστα;

Για να δημιουργήσω μια λίστα, πρέπει απλά να την αρχικοποιήσω όπως θα έκανα με μια μεταβλητή, όμως αντί για κάποια τιμή, θα την έκανα ίση με αγκύλες, δηλαδή αυτό -> [ ]. Έτσι δημιουργώ μια λίστα που περιμένει, έχοντας κανένα στοιχείο.



Πώς δημιουργώ μια λίστα;

```
lista = [] # edw blepoume mia lista sto fusiko ths perivallon  
psonia = ["giaourti", "gala", "egw", "makaronia"]  
  
oi_vathmoi_mou = [3, 2, 4, 0, 9999, (7*3.14)**2.7]  
  
print(oi_vathmoi_mou)  
print(psonia)
```

Προσοχή νεαροί πάνταγουανς!

Μια λίστα ξεκινάει απο τον αριθμό 0 και συνεχίζει  
προς τα πάνω, όχι απο το 1, προσοχη! Ξαφνικά το  
range βγάζει πολύ περισσότερο νόημα

Προσοχή νεαροί πάνταγουανς!

Αν λοιπόν έχουμε μια λίστα με τους μέσους όρους ενός τμήματος, για να πάρουμε τον βαθμό του πρώτου παιδιού, θα πρέπει να γράψουμε στο πρόγραμμά μας “vathmoi[0]”, για το δεύτερο άτομο “vathmoi[1]” κ.ο.κ.

## Ανακαλύπτοντας τα μυστικά της λίστας

```
# ta diaforetika tmhmata se ena tuxaio lukeio
tmhmata = ["a1", "a2", "a3", "b1", "b2", "b3", "c1", "c2", "c3"]

'''
.....
.....
.....
.....
print(tmhmata[9])    Auta kalo einai na mh ginontai
.....
.....
'''

print(tmhmata[1])
# ti pisteuete tha emfanisei h othoni?
```

Γιατί είναι τόσο κακό να ξεκινήσω απ το 1;  
τι θα συμβεί;

Αν ποτέ κάνεις το λάθος και ξεκινήσεις να μετράς τα στοιχεία απ' το 1, τότε απλά θα πάρεις άλλο αποτέλεσμα απο αυτό που θα έπρεπε. Υπάρχει όμως και μια εξαίρεση. Ο θρύλος λέει πως αν ψάξεις για ένα σημείο της λίστας που δεν υπάρχει, τότε θα νιώσεις την πραγματική οργή του μηχανήματος!

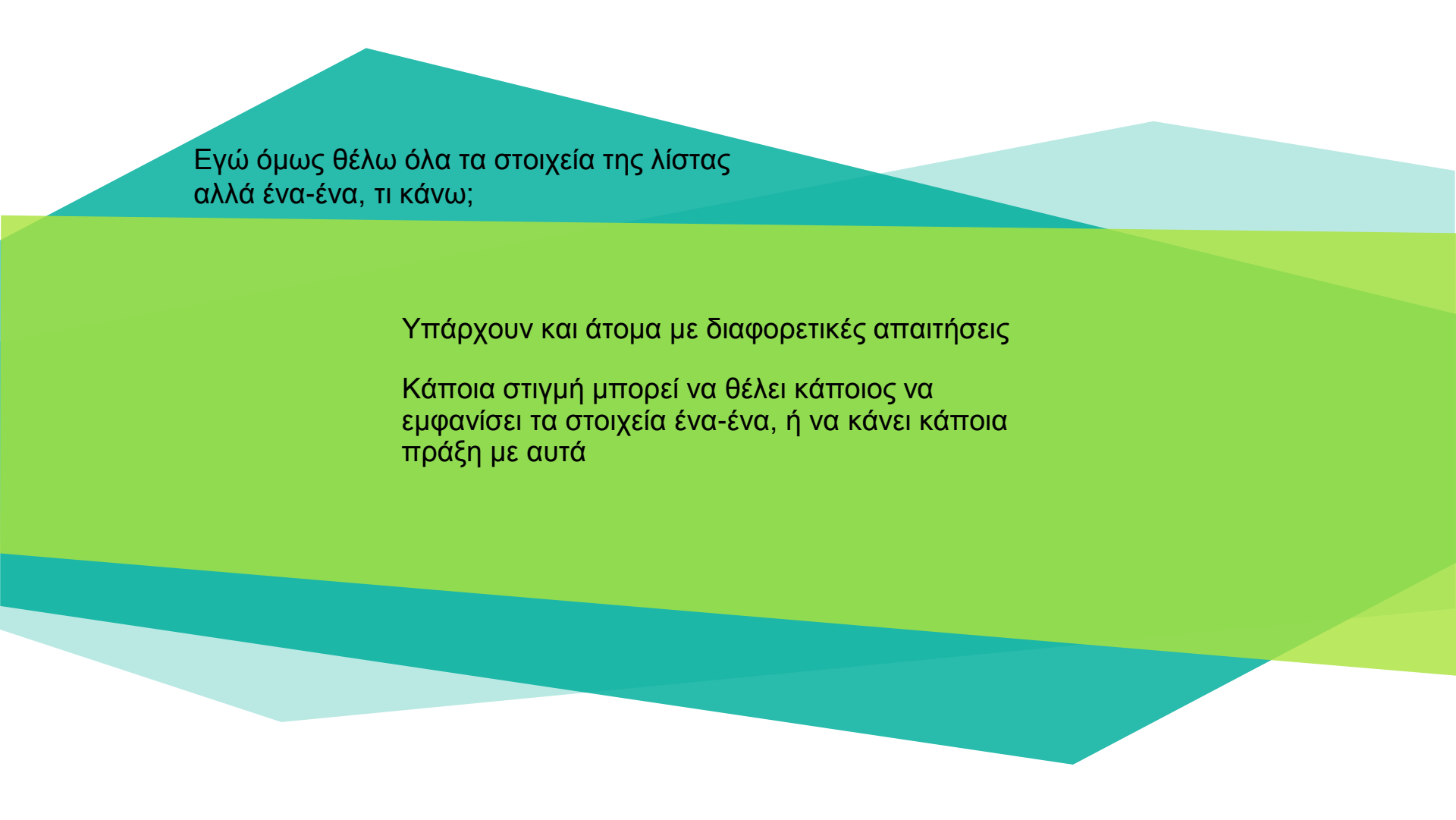
## Ανακαλύπτοντας κι άλλα μυστικά της λίστας

```
# ta διαφορεtika tmhmata se ena tuxaio lukeio
tmhmata = ["a1", "a2", "a3", "b1", "b2", "b3", "c1", "c2", "c3"]

#eipame an vgw ektos panw ap to orio (px tmhmata[9]) den tha exei plaka
#ti tha ginei omws an vgw ektos katw ap to orio? as to anakalupsoume mazi

print(tmhmata[-1])
print(tmhmata[-4])

#ti parathreite?
#epishs, ti tha kanw an thelw na emfanisw olh th lista?
#de tha einai kourastiko na emfanisw ola ta stoixeia gia megales listes?
```



Εγώ όμως θέλω όλα τα στοιχεία της λίστας  
αλλά ένα-ένα, τι κάνω;

Υπάρχουν και άτομα με διαφορετικές απαιτήσεις

Κάποια στιγμή μπορεί να θέλει κάποιος να  
εμφανίσει τα στοιχεία ένα-ένα, ή να κάνει κάποια  
πράξη με αυτά

Εμφάνιση κάθε στοιχείου της λίστας

**Μικρό, άμαθο παιδί**

```
# ta diaforetika tmhmata se ena tuxaio lukeio
tmhmata = ["a1", "a2", "a3", "b1", "b2", "b3", "c1", "c2", "c3"]

for arithmos in range(9):
    print(tmhmata[arithmos])

# uparxei omws kai pio omorfos tropos na kaneis to idio akriwvs pragma
```

**Like a mother-flippin' boss!**

```
# ta diaforetika tmhmata se ena tuxaio lukeio
tmhmata = ["a1", "a2", "a3", "b1", "b2", "b3", "c1", "c2", "c3"]

for tmhma in tmhmata:
    print(tmhma)

# de niwtheis ligo pio mpampas etsi?
```



Ξέχασα να βάλω το βούτυρο στη λίστα μου  
Τι κάνω τώρα;

Κι εσύ έχεις βαρεθεί να ξεχνάς να βάλεις το βούτυρο  
στη λίστα όταν πας για ψώνια;  
Κι εσύ θα' χεις κουραστεί να τρώς μπισσούκο No 42  
στη μούρη επειδή πάντα γυρνάς με τα μισά πράγματα  
απ' όσα χρειάζεσαι;

Τι είναι το *append()*;

```
psonia = ["giaourti", "gala", "egw", "makaronia"]  
# ksexasa to voutiro! kai twra? as to valw sto telos ths listas  
psonia.append("voutiro")  
print(psonia)
```

Τελικά έχω βούτυρο, δε θέλω να ξαναπάρω  
Τι κάνω τώρα;

Κι εσύ έχεις βαρεθεί να παίρνεις βούτυρο ενώ ήδη  
έχεις άλλες 7 συσκευασίες σπίτι;


Κι εσύ έχεις κουραστεί να τρώς μπισσούκο No 42 στη  
μούρη επειδή πάντα γυρνάς με περισσότερα  
πράγματα απ' όσα χρειάζεσαι;

Τι είναι το *remove()*;

```
psonia = ["giaourti", "gala", "egw", "makaronia", "voutiro"]  
  
psonia.remove("voutiro")  
  
print(psonia)  
# luthike to provlima?
```



## Credits

- ◆ Στο SlidesCarnival για το περίγραμμα των παρουσιάσεων
  - ◆ Στα παιδιά που δημιούργησαν το 1ο workshop ως πηγή έμπνευσης του περιεχομένου
- 

Αυτή η διαφάνεια σημαίνει το τέλος  
του επαναληπτικού μέρους του workshop  
ας κάνουμε ένα διάλειμμα

**Ευχαριστώ για τη προσοχή σας!**