

**Τίτλος δραστηριότητας: Η Γλώσσα Logo ως παράδειγμα δομητιστικής μάθησης**

**Τάξη: Ε – ΣΤ**

**Μάθημα: ΜΑΘΗΜΑΤΙΚΑ**

**Δημιουργός : ΝΗΣΙ ΤΩΝ ΦΑΙΑΚΩΝ (Ειρήνη Βιδάκη – Τακούλη Ελένη)**

**Λογισμικό: MSW LOGO**

## **Το Νησί των Φαιάκων**

# **Η Γλώσσα Logo ως παράδειγμα δομητιστικής μάθησης**

**ΟΚΤΩΒΡΙΟΣ 1999**

## ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Το πακέτο της Logo είναι μια εφαρμογή που μας δίνει τη δυνατότητα να δημιουργούμε προγράμματα μόνοι μας και να τα εκτελεί ο υπολογιστής. Μας επιτρέπει να αναπτύσσουμε τις δικές μας εφαρμογές.

Μια γλώσσα προγραμματισμού είναι μια γλώσσα επικοινωνίας με τον υπολογιστή. Είναι ένα σύνολο από εντολές (δεσμευμένες λέξεις) οι οποίες λένε στον υπολογιστή τι να κάνει. Όπως η γλώσσα που χρησιμοποιούμε για την καθημερινή μας επικοινωνία, έτσι και οι γλώσσες προγραμματισμού διαθέτουν γραμματική και κανόνες για το συνδυασμό αυτών των εντολών, ώστε να συντάσσουμε προγράμματα κατανοητά στον υπολογιστή για να μπορεί να τα εκτελέσει.

**Στον υπολογιστή απευθυνόμαστε μέσω εντολών – πρέπει να είμαστε ακριβείς και να προσέχουμε οι εντολές αυτές να είναι σωστές συντακτικά. Σε διαφορετική περίπτωση ο υπολογιστής δεν καταλαβαίνει τι του λέμε να κάνει.**

## Γιατί Logo;

Υπάρχουν πολλές γλώσσες προγραμματισμού, με αρκετές ομοιότητες, αλλά και διαφορές μεταξύ τους. Άλλες έχουν σχεδιαστεί για ειδικές επιστημονικές εφαρμογές (Fortran, C), άλλες χρησιμοποιούνται στην Τεχνική Νοημοσύνη (Lisp, Prolog), ενώ άλλες θεωρούνται **γλώσσες γενικής χρήσης** (Pascal, Basic, **Logo**).

Η Pascal για παράδειγμα είναι ίσως η πιο κατάλληλη γλώσσα για να μάθει κανείς δομημένο προγραμματισμό, αλλά στόχος μας δεν είναι να γίνουμε προγραμματιστές ηλεκτρονικών υπολογιστών.

Οι λόγοι που επιλέχθηκε η Logo είναι ουσιαστικά δύο:

**1)** Η γλώσσα προγραμματισμού Logo δημιουργήθηκε από τον Seymour Papert όχι για την ανάπτυξη πακέτων εφαρμογών, αλλά ως ένα **ισχυρό εργαλείο** στα χέρια όλων εκείνων που εμπλέκονται με τη διαδικασία της μάθησης. Δημιουργήθηκε ειδικά για την εκπαίδευση και μπορεί να αξιοποιηθεί εύκολα και σε πολλές άλλες σχολικές δραστηριότητες. Η Logo έχει κατασκευαστεί με σκοπό ο μαθητής να μαθαίνει/διδάσκει τον υπολογιστή και όχι το αντίθετο. Δημιουργώντας έναν αλγόριθμο (λέγοντας αλγόριθμο εννοούμε τα βήματα που ακολουθούμε για την επίλυση ενός προβλήματος) ο προγραμματιστής – μαθητής προσπαθεί να διδάξει τον υπολογιστή πως να αντιμετωπίζει ένα πρόβλημα.

Έτσι ανακαλύπτει τον τρόπο με τον οποίο ο ίδιος θα έλυne το πρόβλημα και προσπαθώντας να τον περιγράψει με μια γλώσσα προγραμματισμού, συνειδητοποιεί το δικό του τρόπο σκέψης.

Η Logo έχει απλές εντολές και με την χρήση αυτών ο μαθητής αποκτά μια αίσθηση κυριαρχίας σ'αυτού του είδους την τεχνολογία.

2) Ο δεύτερος λόγος που επιλέχθηκε η Logo, έχει να κάνει με τη φύση και τη δομή της. Με βάση τις εντολές της που πλησιάζουν τη φυσική γλώσσα, η Logo ανοίγει ορίζοντες για τη δημιουργία διαδικασιών με τις οποίες αναπτύσσονται περιβάλλοντα με καθορισμένες ιδιότητες, χαρακτηριστικά και περιορισμούς, οι **μικρόκοσμοι**.

## ΓΝΩΡΙΜΙΑ ΜΕ ΤΟ ΠΕΡΙΒΑΛΛΟΝ ΤΗΣ Logo

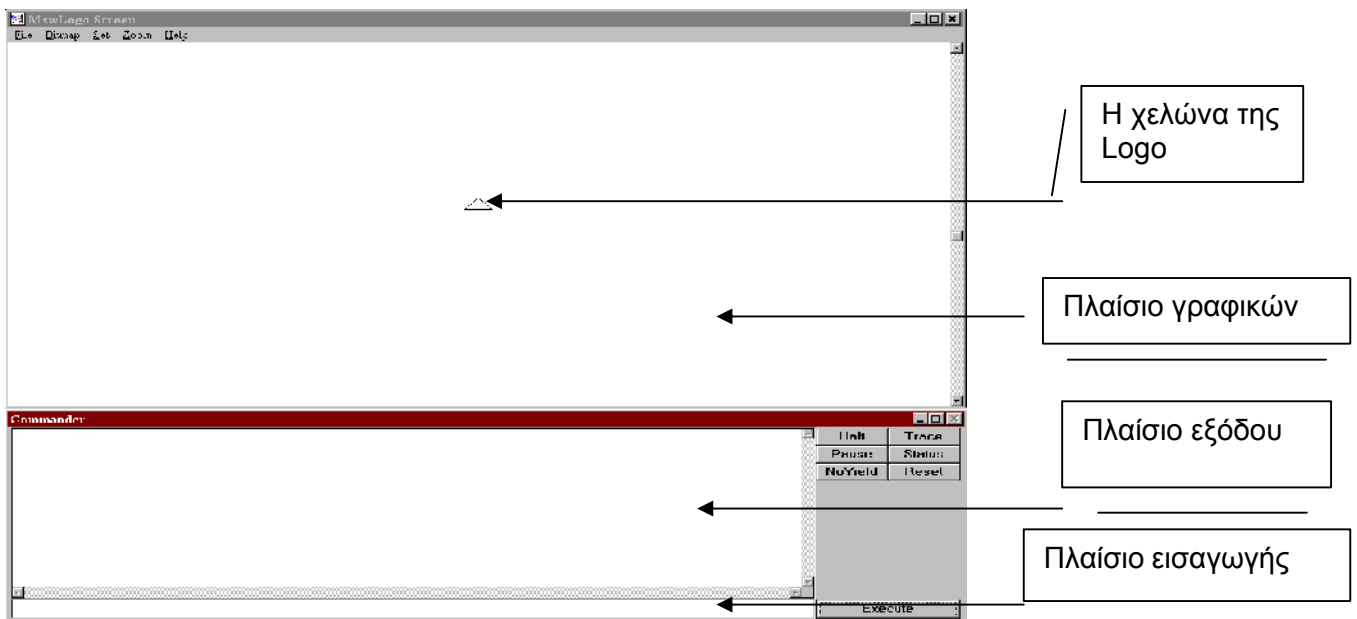
Για να ανοίξουμε την εφαρμογή της Logo είτε πηγαίνουμε από τη γραμμή εκκίνησης :**Start – Programs – Microsoft Windows Logo – MSWLogo** ή διπλοπατάμε στο εικονίδιό της,



MSWLogo .lnk

που βρίσκεται στην επιφάνεια εργασίας.

Στην οθόνη εμφανίζεται το περιβάλλον που φαίνεται στην εικόνα:



Παρατηρούμε ότι υπάρχουν δύο ξεχωριστά παράθυρα. Ένα μεγάλο που έχει τίτλο **MSWLogo Screen** και είναι το κύριο παράθυρο της εφαρμογής. Στο μέσον αυτού του παραθύρου βρίσκεται η χελώνα μας η οποία στη συγκεκριμένη έκδοση απεικονίζεται με τη μορφή ενός ισοσκελούς τριγώνου και η οποία έχει θέση και προσανατολισμό. Αρχικά η χελώνα έχει κατεύθυνση κατακόρυφα προς τα πάνω. Στη θέση αυτή βρίσκεται το «σπίτι» (home) της. Εδώ εμφανίζονται τα γραφικά της, γι'αυτό ονομάζεται παράθυρο γραφικών.

Ακριβώς κάτω απ' αυτό το παράθυρο υπάρχει ένα μικρότερο το οποίο έχει τίτλο **Commander** και ονομάζεται παράθυρο εντολών γιατί σ'αυτό πληκτρολογούμε τις εντολές τις οποίες καλείται να εκτελέσει η χελώνα.

Το παράθυρο εντολών αποτελείται από δύο πλαίσια (βλέπε παραπάνω σχήμα). Το πλαίσιο εισαγωγής στο οποίο εισάγουμε από το

πληκτρολόγιο τις οδηγίες προς τον υπολογιστή για να δώσουμε εντολή να εκτελεστούν, και το πλαίσιο εξόδου στο οποίο εμφανίζονται οι εντολές όταν εκτελεστούν τα μηνύματα του συστήματος, καθώς και τα λεκτικά ή αριθμητικά αποτελέσματα μιας επεξεργασίας. Η ενεργοποίηση των πλαισίων γίνεται με το ποντίκι. Το πλαίσιο εξόδου περιλαμβάνει διάφορα πλήκτρα που θα μάθουμε να τα χρησιμοποιούμε σιγά σιγά.

## Η ΚΙΝΗΣΗ ΤΗΣ ΧΕΛΩΝΑΣ ΣΤΗΝ ΟΘΟΝΗ

Η χελώνα μπορεί να εκτελέσει μόνο απλές κινήσεις. Προχωράει ευθύγραμμα εμπρός ή πίσω και στρίβει επί τόπου (χωρίς να μετατοπίζεται) δεξιά ή αριστερά. Με αυτές τις απλές κινήσεις όμως, αν την καθοδηγήσουμε κατάλληλα, δημιουργεί πολύπλοκα γραφικά, με καταπληκτική ακρίβεια και ταχύτητα.

Αυτές τις κινήσεις τις κατευθύνουμε με εξίσου απλές εντολές, που εκφράζουν με επιγραμματικό τρόπο τις λειτουργίες που κάνουν. Για τις περισσότερες από αυτές προβλέπεται και σύντομη έκφραση, με δύο ή τρία μόνο γράμματα, για να μην είναι κουραστική η πληκτρολόγηση (είναι το ίδιο αν γράφω με κεφαλαία ή με μικρά):

ΕΝΤΟΛΕΣ	ΣΥΝ/ΓΡΑΦΙΑ	ΑΠΟΤΕΛΕΣΜΑ	ΠΑΡΑΔΕΙΓΜΑ
<b>FORWARD</b>	<b>FD</b>	Κινεί τη χελώνα <b>μπροστά</b>	<b>FD</b> □ <b>50</b> ↵
<b>BACK</b>	<b>BK</b>	Η χελώνα <b>οπισθοχωρεί</b>	<b>BK</b> □ <b>30</b> ↵
<b>RIGHT</b>	<b>RT</b>	Στρίβει τη χελώνα <b>δεξιά</b>	<b>RT</b> □ <b>90</b> ↵
<b>LEFT</b>	<b>LT</b>	Στρίβει τη χελώνα <b>αριστερά</b>	<b>LT</b> □ <b>180</b> ↵
<b>PENUP</b>	<b>PU</b>	Η χελώνα σηκώνει το μολύβι και <b>δεν γράφει</b>	<b>PU</b> □ <b>FD</b> □ <b>60</b> ↵
<b>PENDOWN</b>	<b>PD</b>	Η χελώνα κατεβάζει το μολύβι και <b>ξαναγράφει</b>	<b>PD</b> □ <b>FD</b> □ <b>40</b> ↵
<b>PENERASE</b>	<b>PE</b>	Η χελώνα αφήνει το μολύβι και παίρνει τη <b>γόμα</b>	<b>PE</b> □ <b>BK</b> □ <b>30</b> ↵
<b>PENPAINT</b>	<b>PPT</b>	Η χελώνα <b>ξαναπιάνει το μολύβι ζωγραφικής</b>	<b>PPT</b> □ <b>FD</b> □ <b>100</b> ↵
<b>HOME</b>		Η χελώνα επιστρέφει στο <b>σπίτι της</b> , χωρίς να σβηστούν τα γραφικά	Για να μην αφή-νει το ίχνος της <b>PU</b> □ <b>home</b> □ <b>PD</b> ↵
<b>CLEARSCREEN</b>	<b>CS</b>	<b>Καθαρίζει η οθόνη</b> από όλα τα γραφικά και η χελώνα επιστρέφει	

		<b>στο σπίτι της</b>	
<b>CLEAN</b>		<b>Καθαρίζει η οθόνη από τα γραφικά, χωρίς να κινηθεί η χελώνα</b>	
<b>CLEARTEXT</b>	<b>CT</b>	<b>Καθαρίζει ότι υπάρχει στο πλαίσιο εξόδου εντολών</b>	

Όπως παρατηρούμε στα παραδείγματα, όλες οι εντολές κίνησης πρέπει να συνοδεύονται **απαραίτητα** από ένα **αριθμητικό στοιχείο εισόδου**, το οποίο πληροφορεί τη χελώνα πόσο ακριβώς θέλουμε να μετακινηθεί και πόσο να στρίψει. Αν δηλαδή πληκτρολογήσω μόνο:

**FD** ↵

Αμέσως θα εμφανιστεί το μήνυμα:

**Not enough inputs to FD**  
**(Όχι αρκετά στοιχεία στοFD)**

Επίσης απαραίτητο είναι **να μην υπάρχουν λάθη πληκτρολόγησης ή συντακτικά λάθη** όπως και **να αφήνω κενά μεταξύ της εντολής και του αριθμού**.

Αν δηλαδή αντί HOME, πληκτρολογήσω:

**HME** ↵

Αμέσως θα εμφανιστεί το μήνυμα:

**I don't know how to hme**  
**(Δεν ξέρω πώς να κάνω hme)**

Το ίδιο αν γράψω:

**FD80** ↵

Αμέσως θα εμφανιστεί το μήνυμα:

**I don't know how to FD80**  
**(Δεν ξέρω πώς να κάνω FD80)**

Ο αριθμός ο οποίος συνοδεύει τις δύο πρώτες εντολές, (FD και BK) εκφράζει το **μήκος σε "βήματα χελώνας" (pixel της οθόνης)** γι' αυτό και αν δώσουμε

**FD 1** ↵

δεν θα φανεί γιατί ισοδυναμεί με βήμα ενός pixel, δηλαδή μιας κουκίδας στην οθόνη.

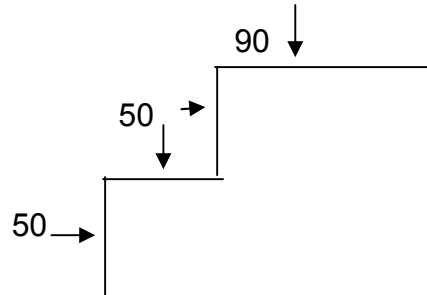
Ενώ ο αριθμός ο οποίος συνοδεύει τις δύο επόμενες εντολές, (RT και LT) εκφράζει τη **στροφή της χελώνας σε μοίρες**.

Ας δοκιμάσουμε μερικά παραδείγματα για να δούμε πως κινείται η χελώνα. Κάντε κλικ στο πλαίσιο εισόδου εντολών έτσι ώστε να αναβοσβήνει ο δρομέας εκεί, για να μπορέσουμε να εισάγουμε τις εντολές.

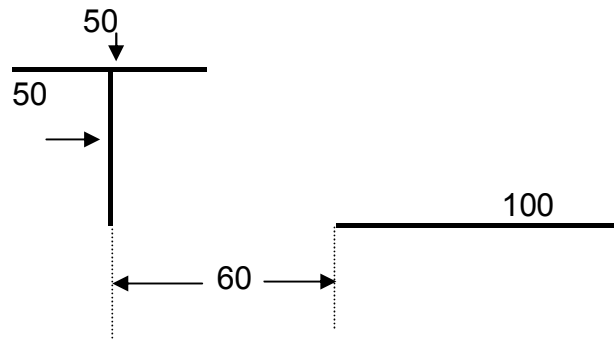


### ΕΞΑΣΚΗΣΗ:

\*Δοκίμασε με τη βοήθεια των εντολών, να κατασκευάσεις την παρακάτω σκάλα:



\*Δοκίμασε με τη βοήθεια των εντολών, να κατασκευάσεις το παρακάτω σχήμα:



\*Δοκίμασε με τη βοήθεια των εντολών, να παίξεις με τη χελώνα μόνος/η σου.

## ΣΧΕΔΙΑΖΟΥΜΕ ΕΝΑ ΤΕΤΡΑΓΩΝΟ ΜΕ ΤΟ ΜΟΛΥΒΙ ΤΗΣ ΧΕΛΩΝΑΣ

Ας προσπαθήσουμε να φτιάξουμε ένα τετράγωνο, αφού πρώτα καθαρίσουμε την οθόνη. Εδώ δε μπορούμε να προχωρήσουμε τυχαία, όπως πριν. Χρειάζεται μέθοδος.

### Σκεφτόμαστε πρώτα και πληκτρολογούμε μετά

#### Ανάλυση του προβλήματος

Θέλω να δημιουργήσω ένα τετράγωνο, μήκους βήματος χελώνας 60.

Βάζω τον εαυτό μου στη θέση της χελώνας και σκέφτομαι τι θα έκανα με το σώμα μου για να διαγράψω ένα τετράγωνο.

Τις ίδιες κινήσεις θα οδηγήσω τη χελώνα να κάνει με τις κατάλληλες εντολές.

Πληκτρολογούμε λοιπόν:

**FD □ 60 ↵**

**RT □ 90 ↵**

**FD □ 60 ↵**

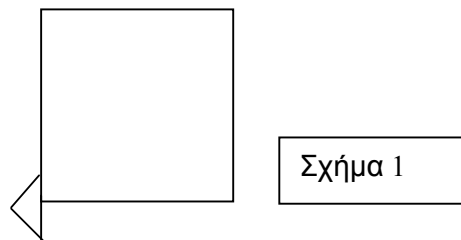
**RT □ 90 ↵**

**FD □ 60 ↵**

**RT □ 90 ↵**

**FD □ 60 ↵**

Και το τετράγωνο είναι έτοιμο (σχήμα 1).

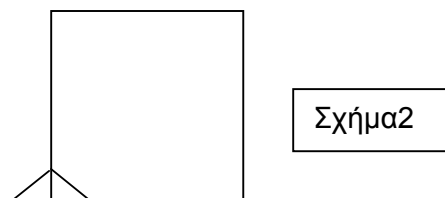


Όμως τα βήματα που ακολουθήσαμε δεν είναι σύμφωνα με το γεωμετρικό ορισμό του τετραγώνου: "Το τετράγωνο έχει 4 πλευρές ίσες και 4 γωνίες ορθές".

Κινήσαμε 4 φορές κατά 60 βήματα τη χελώνα, αλλά την στρίψαμε μόνο 3 φορές κατά 90 μοίρες.

Την οδηγούμε λοιπόν να στρίψει άλλη μια φορά κατά 90 μοίρες.

**RT □ 90 ↵**

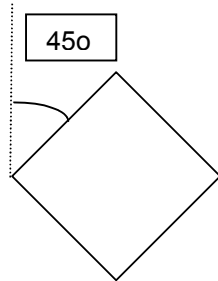


Παρατηρούμε ότι το τετράγωνο (σχήμα 2), δεν μεταβάλλεται αλλά η χελώνα στρέφεται άλλη μια φορά κατά 90 μοίρες και έτσι επανέρχεται στην αρχική της θέση και με τον ίδιο προσανατολισμό που είχε στην αρχή. (Διέγραψε δηλαδή ένα κλειστό πολύγωνο και στράφηκε συνολικά κατά  $90 \cdot 4 = 360$  μοίρες).

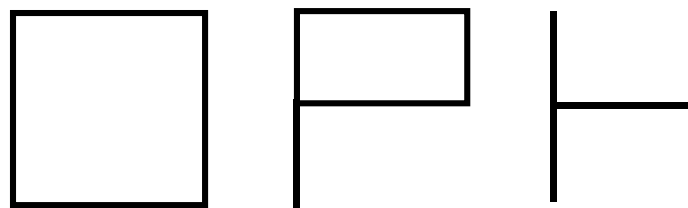
Αυτό είναι σημαντικό να το θυμόμαστε, να επαναφέρουμε δηλαδή τη χελώνα στην αρχική της θέση, για να μπορούμε αργότερα όταν θα σχεδιάζουμε πιο πολύπλοκα σχήματα να προβλέπουμε με ακρίβεια τη πορεία της στη συνέχεια.

## ΕΞΑΣΚΗΣΗ:

\*Δοκίμασε με τη βοήθεια των εντολών να φτιάξεις το παρακάτω τετράγωνο:

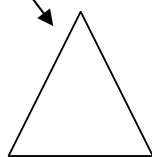


\*Δοκίμασε με τη βοήθεια των εντολών να σχηματίσεις στην οθόνη τη λέξη "ΟΡΗ", χωρίς καμπυλότητες:

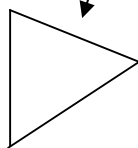


\*Δοκίμασε να φτιάξεις ένα ισόπλευρο τρίγωνο:

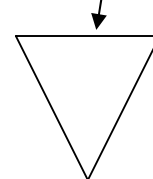
σαν και αυτό



ή αυτό



ή αυτό



## ΔΙΑΔΙΚΑΣΙΕΣ ΣΤΗ Logo

Τα προγράμματα σε μια οποιαδήποτε γλώσσα προγραμματισμού είναι μια σειρά από οδηγίες που βασίζονται στις εντολές αυτής της γλώσσας.

Τα προγράμματα της LOGO λέγονται **διαδικασίες (procedures)** και έχουν **ορισμένο τρόπο σύνταξης**. Κάθε διαδικασία έχει **το δικό της όνομα** και συνήθως επιλέγουμε ένα όνομα που να μας θυμίζει το αποτέλεσμα που θα έχουμε κατά την εκτέλεση.

Στην προηγούμενη ενότητα μάθαμε πως να σχεδιάζουμε ένα τετράγωνο χρησιμοποιώντας μια σειρά από οδηγίες που βασίζονται στις εντολές της γλώσσας LOGO.(π.χ. **fd**, **rt** οι οποίες είναι εντολές που έχουν οριστεί από τον κατασκευαστή της LOGO και γι'αυτό ονομάζονται πρωταρχικές διαδικασίες). Στη Logo μας δίνεται η δυνατότητα να «εκπαιδεύσουμε» τη χελώνα να εκτελεί εντολές τις οποίες έχουμε δημιουργήσει εμείς οι ίδιοι. Μπορούμε δηλαδή να την εκπαιδεύσουμε ώστε μόλις γράψουμε την εντολή **tetragono** ή κάτι ανάλογο, να καταλαβαίνει και να θυμάται τι πρέπει να κάνει, χωρίς να χρειάζεται κάθε φορά να της υπαγορεύουμε τις γνωστές πια σ'εμάς οδηγίες.

Ο τρόπος για να επιτύχουμε κάτι τέτοιο στη LOGO είναι να συμπεριλάβουμε αυτές τις οδηγίες σε μια διαδικασία (πρόγραμμα) με δικό της όνομα.

### Πώς συντάσσουμε μια διαδικασία

Για τη σύνταξη (αλλά και διόρθωση) μιας διαδικασίας προβλέπεται ένα άλλο παράθυρο. Για να το ανοίξουμε επιλέγουμε από την **γραμμή μενού FILE - EDIT**. Εμφανίζεται ένα πλαίσιο διαλόγου με τον τίτλο **Edit Procedure**. Στη συνέχεια πατάμε το κουμπί **OK** και εμφανίζεται ένα άλλο παράθυρο με τίτλο **EDITOR**, το οποίο καλούμε παράθυρο σύνταξης και εδώ ορίζουμε τις διαδικασίες. Στο παράθυρο αυτό παρατηρούμε ότι υπάρχουν έτοιμες οι λέξεις :

**to**  
**end.**

Η λέξη **to** ορίζει την αρχή της διαδικασίας (επικεφαλίδα) και η λέξη **end** το τέλος της.

- Δίπλα στη λέξη **to** δίνουμε το όνομα της διαδικασίας και πατάμε το πλήκτρο **enter** (↵).

- Στη νέα γραμμή πληκτρολογούμε σε μία ή περισσότερες γραμμές τις οδηγίες με την σειρά που θέλουμε να εκτελεστούν.

### Παράδειγμα

Εάν θέλουμε να ορίσουμε τη διαδικασία για το σχεδιασμό ενός τετραγώνου έχουμε την ακόλουθη σύνταξη.

```
to tetragono ↵  
fd 70 rt 90 fd 70 rt 90 fd 70 rt 90 fd 70 rt 90  
end
```

Οι οδηγίες που παρεμβάλλονται μεταξύ των λέξεων **to** και **end** αποτελούν τον κορμό της διαδικασίας.

Αυτές οι οδηγίες θα αντιπροσωπεύονται στο εξής από τη λέξη **tetragono**. Έτσι εκπαιδεύσαμε τη χελώνα (δηλ. τον υπολογιστή) σε νέες έννοιες.

Για να κλείσουμε το παράθυρο σύνταξης επιλέγουμε από τη γραμμή menu του συγκεκριμένου παραθύρου **FILE\_EXIT**.

Τότε εμφανίζεται ένα πλαίσιο διαλόγου το οποίο έχει τίτλο Contents Changed και μας ενημερώνει ότι τα περιεχόμενα έχουν αλλάξει και αν θέλουμε να τα σώσουμε (**Contents have changed.Save?**).

- ♦ Εάν επιλέξουμε το κουμπί  κλείνει το παράθυρο σύνταξης αλλά το περιεχόμενο του δεν σώζεται.
- ♦ Εάν επιλέξουμε το κουμπί  τότε ξαναεπιστρέφουμε στο παράθυρο του editor και μπορούμε να συνεχίσουμε τη σύνταξη διαδικασιών.
- ♦ Εάν επιλέξουμε το κουμπί  κλείνει το παράθυρο σύνταξης και το περιεχόμενο της διαδικασίας που μόλις ορίσαμε καταχωρείται στο χώρο εργασίας στη μνήμη του υπολογιστή, και το όνομά της συγκαταλέγεται στο λεξιλόγιο της Logo που την αναγνωρίζει στο εξής όπως ακριβώς και τις πρωταρχικές διαδικασίες. **Έτσι η χελώνα γνωρίζει στο εξής τη σημασία της λέξης tetragono.**

Δοκιμάζουμε πληκτρολογώντας στο πλαίσιο εισόδου το όνομα της διαδικασίας που έχουμε ορίσει δηλ. **tetragono** ↵ Πράγματι η χελώνα σχεδιάζει αστραπιαία ένα τετράγωνο στην οθόνη γραφικών. Το σύστημα εκτέλεσε την ακολουθία εντολών που περιλαμβάνονται στην αντίστοιχη διαδικασία.

Την νέα εντολή αυτή μπορούμε να τη χρησιμοποιήσουμε, όπως και τις πρωταρχικές. Έτσι μπορούμε να τη συνδυάσουμε με άλλες εντολές, ώστε να φτιάξουμε όμορφα γραφικά.

### **Δοκιμάστε:**

```
cs tetragono rt 90 tetragono rt 90 tetragono rt 90 tetragono rt 90 ↵
```

Με τον παραπάνω τρόπο στη LOGO μας δίνεται η δυνατότητα να εμπλουτίσουμε το "λεξιλόγιό" μας με έννοιες νέες, πιο σύνθετες οι οποίες βασίζονται σε προηγούμενες γνώσεις. Αυτές οι έννοιες τις οποίες ορίζουμε

εμείς ονομάζονται απλά διαδικασίες σε αντίθεση με τις εντολές που διαθέτει αρχικά το περιβάλλον της LOGO (fd, bk, rt, lt, pe, ppt, pu, pd...) οι οποίες όπως έχουμε ήδη αναφέρει ονομάζονται πρωταρχικές διαδικασίες. Το όνομα μιας πρωταρχικής διαδικασίας δεν μπορεί να χρησιμοποιηθεί ως όνομα σε μια διαδικασία που θα ορίσουμε εμείς.

Δοκιμάζουμε να ορίσουμε στον editor μια διαδικασία με το όνομα right. Παρατηρούμε ότι όταν στο πλαίσιο διαλόγου Contents Changed επιλέξουμε yes εμφανίζεται ένα μήνυμα το οποίο μας πληροφορεί ότι δεν μπορεί η διαδικασία με το όνομα που δώσαμε να καταχωρηθεί στη μνήμη γιατί η λέξη είναι ήδη ορισμένη από τον κατασκευαστή της Logo.



## ΜΙΑ ΔΙΑΔΙΚΑΣΙΑ ΚΑΛΕΙ ΚΑΠΟΙΑ ΑΛΛΗ - (ΥΠΟΔΙΑΔΙΚΑΣΙΕΣ ΥΠΕΡΔΙΑΔΙΚΑΣΙΕΣ)

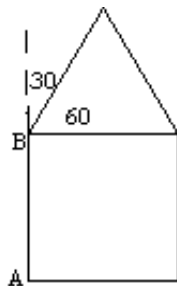
Όπως έχουμε αναφέρει το όνομα μιας διαδικασίας είναι μία εντολή που τη χρησιμοποιούμε όπως και τις πρωταρχικές. Έτσι μπορούμε να τη συμπεριλάβουμε στη σύνταξη μιας νέας διαδικασίας, όπως μπορούμε να συμπεριλάβουμε και οποιαδήποτε άλλη εντολή που είναι απαραίτητη.

Στην προηγούμενη ενότητα συντάξαμε τη διαδικασία για την κατασκευή ενός τετραγώνου. Με παρόμοιο τρόπο μπορούμε να συντάξουμε μια διαδικασία για την κατασκευή ενός ισόπλευρου τριγώνου με πλευρά ίση με αυτή του τετραγώνου.

Π.χ. **to trigono**

```
fd 70 rt 120 fd 70 rt 120 fd 70 rt 120  
end
```

Και αν θέλουμε μία διαδικασία για το σχεδιασμό του παρακάτω σχήματος:



### Ανάλυση του προβλήματος

Όπως παρατηρούμε το σχήμα αυτό αποτελείται από ένα τετράγωνο και ένα ισόπλευρο τρίγωνο, με ίδιο μήκος πλευράς. Μια καλή ιδέα θα ήταν να χρησιμοποιήσουμε κατάλληλα τις ήδη έτοιμες διαδικασίες `teragono` και `trigono`. Αυτές τις δύο διαδικασίες μπορούμε να τις χρησιμοποιήσουμε στον ορισμό μιας άλλης διαδικασίας που θα λέγεται `SPITI`. Φροντίζουμε μόνο να έχουν και οι δύο το ίδιο μήκος πλευράς.

Όπως φαίνεται στο παραπάνω σχήμα, αν θεωρήσουμε αφετηρία για το τετράγωνο το σημείο `A`, στο σημείο αυτό θα ξαναβρεθεί η χελώνα, μετά την κατασκευή του με κλίση `0`. Ο σχεδιασμός του τριγώνου λοιπόν θα

προβλέψουμε, πριν κληθεί η διαδικασία `trigono`, να έχει η χελώνα διανύσει την κατακόρυφη απόσταση `AB`. Όταν η χελώνα φτάσει στο σημείο `B` θα πρέπει να τη στρίψουμε δεξιά 30 μοίρες.

Έτσι η διαδικασία θα είναι:

```
to SPITI
tetragono
fd 70 rt 30
trigono
end
```

Ας παρακολουθήσουμε νοερά την εκτέλεση, κάνοντας σε ένα χαρτί το σχήμα που πιστεύουμε ότι θα έκανε η χελώνα με αυτές τις εντολές.

Με το κάλεσμα της διαδικασίας **SPITI**, το σύστημα αναζητά τον ορισμό της και εκτελεί με τη σειρά τις εντολές που αυτή περιλαμβάνει. Συναντά πρώτα τη λέξη **tetragono**. Την αναγνωρίζει και αναζητά τον ορισμό της για να τον εκτελέσει. **Ο έλεγχος δηλαδή περνά στη διαδικασία tetragono**. Την εκτελεί και σχεδιάζεται το τεράγωνο, ενώ η χελώνα επιστρέφει στο σημείο `A`.

Μόλις φτάσει στο τέλος της διαδικασίας `tetragono` (`end`), την εγκαταλείπει και ο έλεγχος επιστρέφει στη διαδικασία από την οποία κλήθηκε, δηλ. το `SPITI`, όπου η ροή εκτέλεσης συνεχίζεται στην αμέσως επόμενη εντολή που είναι η **fd 70**. Η χελώνα μετακινείται κατά 70 βήματα και φθάνει στο σημείο `B`. Στρίβει δεξιά κατά 30 μοίρες, **rt 30** (οι γωνίες ενός ισόπλευρου τριγώνου είναι 60 μοίρες)

Η ροή συνεχίζεται στην επόμενη λέξη που είναι η `trigono`. Όπως προηγουμένως, το σύστημα εγκαταλείπει προσωρινά τη διαδικασία `SPITI` και ο έλεγχος περνά στην **trigono**. Την εκτελεί, σχηματίζεται το τρίγωνο και επιστρέφει πάλι στη `SPITI`, στην αμέσως επόμενη εντολή που είναι η **end**.

Η διαδικασία `SPITI` έχει ολοκληρωθεί και το σπίτι είναι έτοιμο.

Βλέπουμε λοιπόν ότι μια διαδικασία μπορεί να κληθεί μέσα από μια άλλη. Γι' αυτό είναι σημαντικό να οργανώνουμε σωστά τη δουλειά μας ώστε να μπορούμε να την αξιοποιήσουμε.

Οι διαδικασίες `tetragono` και `trigono` αποτελούν ανεξάρτητες διαδικασίες αλλά συγχρόνως αποτελούν και **υποδιαδικασίες** της `SPITI`, μέσα στην οποία καλούνται.

Μια διαδικασία μπορεί να είναι υποδιαδικασία σε περισσότερες από μία διαδικασίες, όπως μια πρωταρχική μπορεί να καλείται από πολλές άλλες διαδικασίες. Στο συγκεκριμένο πρόβλημα η διαδικασία `SPITI` αποτελεί την κύρια διαδικασία ή **υπερδιαδικασία** για τις άλλες δύο.

Όταν καλείται μια υποδιαδικασία, ο έλεγχος περνά σ' αυτή, και αφού ολοκληρωθεί η εκτέλεσή της, επιστρέφει στην κύρια διαδικασία όπου συνεχίζεται η εκτέλεση της αμέσως επόμενης εντολής.

## ΑΠΟΘΗΚΕΥΣΗ ΤΗΣ ΕΡΓΑΣΙΑΣ ΜΑΣ ΣΤΗ LOGO

Αν κλείσουμε την εφαρμογή της Logo και την επανεκκινήσουμε την ίδια ή κάποια άλλη μέρα και γράψουμε tetragono στο πλαίσιο εισαγωγής θα δούμε ότι δεν εκτελείται αν και είχαμε ορίσει και "σώσει" τη διαδικασία.

Αυτό συμβαίνει διότι όσο είναι ανοικτό το περιβάλλον της Logo, και ορίζω διαδικασίες στο παράθυρο σύνταξης αυτές αποθηκεύονται στη μνήμη RAM του υπολογιστή. Αν όμως σβήσουμε τον υπολογιστή ή εγκαταλείψουμε τη Logo το περιεχόμενο της μνήμης σβήνεται. Πρέπει λοιπόν προηγουμένως να το αποθηκεύσουμε σε δίσκο.

Με τη γνωστή επιλογή **FILE\_SAVE** από το κύριο παράθυρο, εμφανίζεται το επίσης γνωστό στο περιβάλλον των WINDOWS πλαίσιο διαλόγου **Save As**.

Επιλέγουμε το δίσκο **C:\** και τον κατάλογο στον οποίο θέλουμε να αποθηκευτεί το αρχείο. Στη συνέχεια πληκτρολογούμε το όνομα π.χ. **sximata** και όταν πατήσουμε **OK** αποθηκεύονται εκεί όλες οι διαδικασίες που έχω ορίσει ως εκείνη τη στιγμή (στο παράθυρο σύνταξης) και οι οποίες προσωρινά βρίσκονταν στη μνήμη.

## ΑΝΑΚΤΗΣΗ ΠΡΟΗΓΟΥΜΕΝΗΣ ΕΡΓΑΣΙΑΣ

Για να ξαναέχω όλες τις διαδικασίες που είχα ορίσει, όταν ξαναανοίξω τον υπολογιστή μου ή τη Logo, θα πρέπει να φορτώσω το αρχείο που τις περιέχει αποθηκευμένες και το οποίο βρίσκεται στο δίσκο (αφού εκεί το αποθήκευσα).

Γι'αυτή την εργασία υπάρχει η επιλογή **FILE\_LOAD** από το κύριο παράθυρο. Με αυτήν την επιλογή εμφανίζεται το πλαίσιο διαλόγου **File Open**.

Επιλέγουμε το δίσκο, όπου είναι αποθηκευμένο το αρχείο π.χ. **sximata** το οποίο περιλαμβάνει τις διαδικασίες οι οποίες μας ενδιαφέρουν. Με τη μπάρα κύλισης μπορώ να εντοπίσω το αρχείο μου- αν τυχόν δεν φαίνεται- και με το ποντίκι να το επιλέξω. Τότε βλέπω ότι εμφανίζεται το όνομα του αρχείου **sximata** και στο πλαίσιο εισαγωγής ονόματος. Πατώντας **OK** το αρχείο φορτώνεται στη μνήμη. Τώρα το σύστημα μπορεί να αναγνωρίσει όλες τις διαδικασίες που υπάρχουν σε αυτό το αρχείο και να τις εκτελέσει. Έτσι αν εκεί υπάρχει, π.χ. η διαδικασία **tetragono** και την πληκτρολογήσω στο πλαίσιο εισόδου εντολών θα εκτελεστεί.

Με τον ίδιο τρόπο μπορώ να φορτώσω συγχρόνως και άλλες διαδικασίες τις οποίες τυχόν έχω αποθηκεύσει σε άλλα αρχεία.

## ΑΠΟΘΗΚΕΥΣΗ ΑΡΧΕΙΩΝ ΕΙΚΟΝΑΣ ΚΑΙ ΑΝΑΚΤΗΣΗ ΑΥΤΩΝ

Προηγουμένως είδαμε πώς αποθηκεύουμε σε αρχείο τις διαδικασίες. Τα γραφικά όμως που δημιουργούνται στην οθόνη δεν αποθηκεύονται συγχρόνως. Έτσι αν θέλουμε να φυλάξουμε ένα σχήμα για να το χρησιμοποιήσουμε και σε άλλες εφαρμογές ή για να το συμπληρώσουμε αργότερα, **πρέπει να το αποθηκεύσουμε ιδιαίτερα στο δίσκο ως αρχείο εικόνας.**

Έτσι με την επιλογή **BITMAP\_SAVE** από το κύριο παράθυρο, αποθηκεύονται σε αρχείο τα γραφικά που εκείνη τη στιγμή είναι στην οθόνη.

Αν θέλω να τα φορτώσω κάποια άλλη στιγμή, αυτό μπορεί να γίνει με την επιλογή **BITMAP\_LOAD**, όπου ανοίγει το γνωστό πλαίσιο διαλόγου **File Open** και όπου αν συμπληρώσω ή επιλέξω το αρχείο εικόνας που θέλω θα το φορτώσει.

Ένα αρχείο εικόνας το οποίο έχω δημιουργήσει στη Logo, μπορώ να το ανοίξω και από την εφαρμογή της **Ζωγραφικής**, και από εκεί να το μεταφέρουμε με **COPY/PASTE** σε ένα κείμενο που γράφουμε.

## Η ΔΟΜΗ ΤΗΣ ΕΠΑΝΑΛΗΨΗΣ ΣΤΗ LOGO (REPEAT)

Προκειμένου να σχεδιάσουμε ένα τετράγωνο χρησιμοποιούμε την εξής ακολουθία εντολών στο παράθυρο σύνταξης :

```
To tetragono ↵  
Fd 70 rt 90 ↵  
Fd 70 rt 90 ↵  
Fd 70 rt 90 ↵  
Fd 70 rt 90 ↵  
end
```

Σ'αυτή τη διαδικασία παρατηρούμε ότι το ζεύγος των εντολών [fd 70 rt 90] επαναλαμβάνεται 4 φορές. Πρόκειται λοιπόν για μια **επαναληπτική δομή**, όπου κάποιες ενέργειες επαναλαμβάνονται με την ίδια σειρά για **ένα ορισμένο πλήθος φορών**.

Στη γλώσσα προγραμματισμού Logo για τις περιπτώσεις επανάληψης μιας ομάδας εντολών προβλέπεται η εντολή **REPEAT**, που μας επιτρέπει να τις εκφράσουμε με πιο σύντομο τρόπο. Μια τέτοια επανάληψη στην περίπτωση του τετραγώνου εκφράζεται με τη διατύπωση:

```
To tetragono ↵  
repeat 4[fd 70 rt 90]  
end
```

που σημαίνει, επανάλαβε (REPEAT) 4 φορές τη λίστα ενεργειών (αυτές που περιλαμβάνονται στις αγκύλες).

Εκτελείται λοιπόν πρώτα η fd 70 ,στη συνέχεια η rt 90 και η ροή της εκτέλεσης επιστρέφει πάλι στην fd 70 ώσπου να εκτελεστεί η λίστα αυτή τόσες φορές όσες αναφέρει ο αριθμός μπροστά της. Δημιουργείται λοιπόν ένας βρόγχος, ο οποίος στο συγκεκριμένο παράδειγμα εκτελείται 4 φορές.

Όσον αφορά τον τρόπο σύνταξης της REPEAT, είμαστε υποχρεωμένοι να περιβάλουμε την ακολουθία των εντολών που επαναλαμβάνονται με αγκύλες, διαφορετικά δεν θα γίνει κατανοητή από τον υπολογιστή.

Με την εντολή REPEAT μπορούμε να καθοδηγούμε τη χελώνα και κατ'επέκταση τον υπολογιστή, με σύντομο αλλά δυναμικό τρόπο. Σε διαφορετική περίπτωση θα ήμασταν υποχρεωμένοι να πληκτρολογούμε μία-μία τις εντολές για το σχηματισμό ενός δωδεκαγώνου ή ενός εικοσαγώνου...

## Παραδείγματα με την εντολή repeat

360



Στη διατύπωση `REPEAT 4[FD 70 RT 90]` παρατηρούμε ότι ο αριθμός των επαναλήψεων (4) και η γωνία στροφής (90) μέσα στη λίστα έχουν γινόμενο 360 ( $4 \cdot 90 = 360$ ).

Αυτό συμβαίνει γιατί όταν η χελώνα διαγράφει ένα κυρτό πολύγωνο, στρέφεται συνολικά κατά μία πλήρη γωνία (δηλ. 360 μοίρες).

Το ίδιο θα ισχύει και αν πρέπει να διαγράψει ένα ισόπλευρο τρίγωνο. Αφού θα χρειαστεί να στρίψει τρεις φορές συνολικά, η γωνία στροφής θα είναι κάθε φορά  $360:3=120$ .

Στο παράθυρο σύνταξης γράφουμε τη διαδικασία για την κατασκευή ισόπλευρου τριγώνου:

```
To trigon ↵  
Repeat 3[fd 70 rt 120]  
end
```

Στο πλαίσιο εισαγωγής εισάγουμε την εντολή :

```
cs trigon
```

και όταν πατήσουμε **enter** ( ↵ ) στην οθόνη σχηματίζεται ένα τρίγωνο.

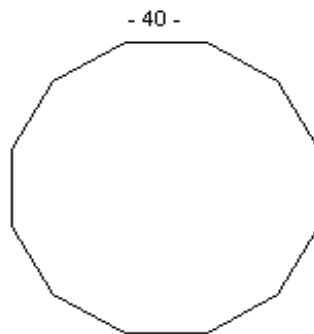
Γενικά λοιπόν για να σχηματίσει η χελώνα ένα κανονικό κυρτό πολύγωνο με ***n* πλευρές**, θα πρέπει να στραφεί ***n* φορές κατά τη γωνία  $360:n$  κάθε φορά**, όσο δηλαδή είναι η εξωτερική (αλλά και η κεντρική) του γωνία.

<b>Ο αριθμός των επαναλήψεων και η γωνία στροφής της χελώνας για το σχηματισμό ενός κανονικού πολυγώνου, έχουν γινόμενο 360 μοίρες.</b>
---

## ΕΞΑΣΚΗΣΗ

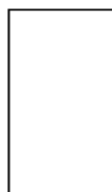
1) Δοκίμασε να φτιάξεις ένα δωδεκάγωνο με τη βοήθεια της εντολής repeat.

```
To dodekagono  
repeat 12[fd 30 rt 360/12]  
end
```



2) Δοκίμασε με την εντολή repeat να ορίσεις διαδικασία χρησιμοποιώντας τη για ένα ορθογώνιο με πλευρές 50 και 100.

```
To parallilo  
repeat 2[fd 100 rt 90 fd 50 rt 90]  
end
```

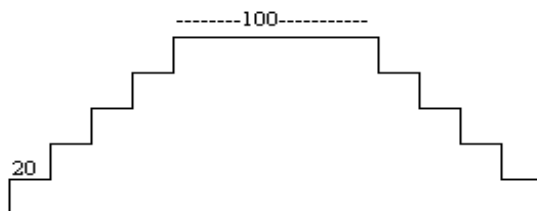


**3)** Να ορίσετε διαδικασία για το παρακάτω σχήμα (exedra) (Ορίστε μια διαδικασία askala για την αριστερή σκάλα χρησιμοποιώντας τη REPEAT, και με παρόμοιο τρόπο μία διαδικασία dskala για την δεξιά σκάλα.-Να ορίσετε μια υπερδικασία που να καλεί τις δύο προηγούμενες διαδικασίες καλώντας ενδιάμεσα εντολές για τον σχεδιασμό της βάσης και του δαπέδου της εξέδρας.

```
To askala  
Repeat 5[fd 20 rt 90 fd 20 lt 90]  
end
```

```
To dskala  
Repeat 5[fd 20 rt 90 fd 20 lt 90]  
end
```

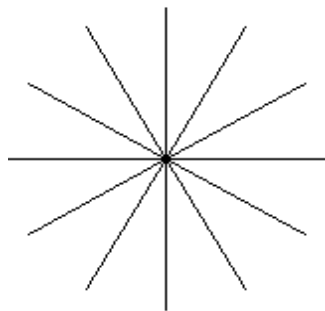
```
To exedra  
askala  
Rt 90 fd 60  
dskala  
Rt 180 fd 260  
end
```



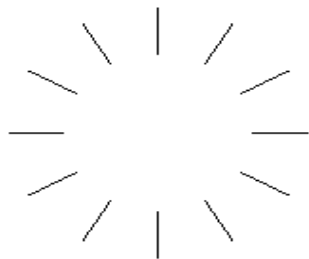


**4) και 5)** Δοκιμάστε να σχεδιάσετε τα παρακάτω σχήματα. (Στο δεύτερο σχήμα είναι απαραίτητη η χρήση των εντολών `pu` και `pd`)

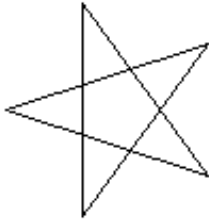
```
to aktina  
repeat 12[fd 100 bk 100 rt 360/12]  
end
```



```
to aktinab  
repeat 12[pu fd 70 pd fd 30 pu bk 100 rt 360/12 pd]  
end
```

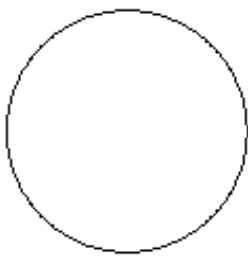


6) Γράψτε τη διαδικασία για την κατασκευή του παρακάτω σχήματος. (Το σχήμα αυτό δεν είναι κυρτό πολύγωνο. Παρατηρούμε όμως ότι οι κορυφές του είναι κορυφές κανονικού πενταγώνου εγγεγραμμένου σε κύκλο. Τα τόξα λοιπόν που ορίζονται από τις κορυφές αυτές είναι τόξα 72 μοιρών. Άρα η γωνία ΑΒΓ είναι 36 μοιρών, οπότε η εξωτερική γωνία είναι 144 μοίρες).



```
to asteri  
repeat 5[fd 100 rt 144]  
end
```

7) Δοκιμάστε να σχεδιάσετε έναν κύκλο χρησιμοποιώντας τις εντολές FD και RT. Τι θα κάνετε αν τον φτιάχνετε στο πάτωμα χρησιμοποιώντας τα βήματά σας; Πόσες φορές θα χρειαστεί να προχωρήσετε μπροστά και να στρίψετε δεξιά; Η εντολή repeat θα σας διευκολύνει να μην επαναλαμβάνεται συνέχεια τις συγκεκριμένες εντολές τόσες φορές όσες χρειάζεται προκειμένου να σχεδιαστεί ένας κύκλος(kyklos).



```
to kyklos  
repeat 360[fd 1 rt 1]  
end
```

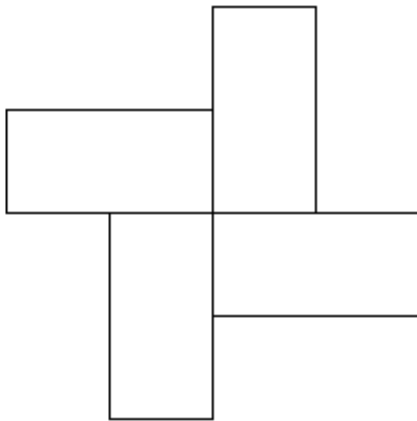
{Ο πιο πιστός κύκλος : repeat 720[fd 1 rt 0.5]}

## **ΜΙΑ ΕΠΑΝΑΛΗΨΗ ΜΠΟΡΕΙ ΝΑ ΕΠΑΝΑΛΑΜΒΑΝΕΤΑΙ**

Ορίζουμε μια διαδικασία για το σχεδιασμό ενός παραλληλογράμμου με πλευρές 50 και 100 αντίστοιχα.

```
to parallilo  
repeat 2 [fd 100 rt 90 fd 50 rt 90]  
end
```

Μπορούμε χρησιμοποιώντας αυτή τη διαδικασία να φτιάξουμε σύνθετα σχήματα όπως το παρακάτω.



Για τη δημιουργία αυτού του σχήματος επαναλάβαμε 4 φορές το ζεύγος των εντολών PARALLILO LT 90. (LT 90 γιατί πρέπει να αλλάξουμε τον προσανατολισμό της χελώνας )

```
PARALLILO LT 90 PARALLILO LT 90 PARALLILO LT 90 PARALLILO  
LT 90
```

Την παραπάνω ακολουθία μπορούμε να την συμπύξουμε με τη βοήθεια της repeat.

```
REPEAT 4 [PARALLILO LT 90]
```

Αυτή η διατύπωση ορίζει την επανάληψη της λίστας των εντολών PARALLILO και LT 90 τέσσερις φορές. Η εντολή PARALLILO όμως αντιπροσωπεύει κι αυτή μια επαναληπτική δομή την

```
Repeat 2[fd 100 rt 90 fd 50 rt 90].
```

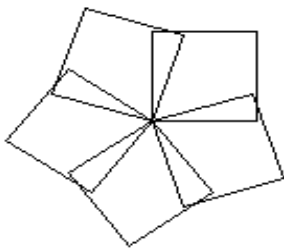
Αν γράψουμε αναλυτικά την προηγούμενη διατύπωση θα έχουμε:

```
REPEAT 4 [REPEAT 2[FD 100 RT 90 FD 50 RT 90] LT 90]
```

Είναι φανερό ότι στην παραπάνω δομή **επαναλαμβάνεται μια επαναληπτική διαδικασία**. Έχουμε λοιπόν μια επανάληψη μέσα στην επανάληψη. Η εσωτερική επανάληψη λέγεται **ένθετος βρόγχος**.

## ΕΞΑΣΚΗΣΗ

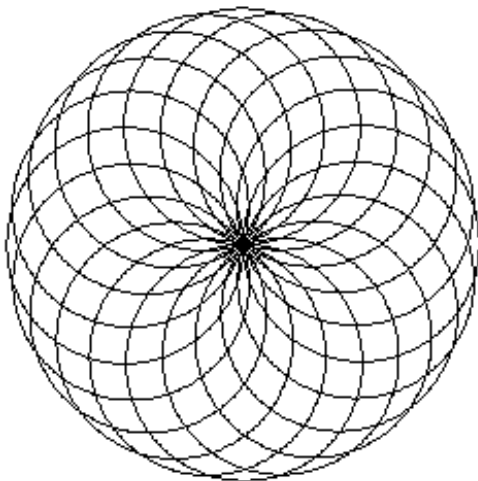
**1)** Γράψτε τη διαδικασία για την κατασκευή του παρακάτω σχήματος. (Πρώτα όρισε μία διαδικασία για τον σχεδιασμό του τετραγώνου (tetragono). Στη συνέχεια όρισε μια υπερδιαδικασία (polytetrag) η οποία θα καλεί την προηγούμενη 5 φορές. Η στροφή μετά από την ολοκλήρωση του τετραγώνου



```
To polytetr  
Repeat 5[repeat 4[fd 50 rt 90 ] rt 360/5]  
end
```

θα πρέπει να είναι 360:5).

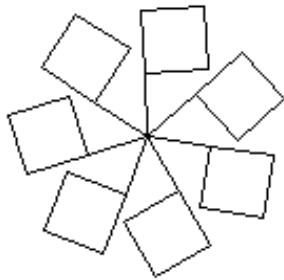
**2)** Σχεδιάστε επαναλαμβανόμενους κύκλους. Χρησιμοποιώντας την διαδικασία repeat.



```
to rodakas  
repeat 20[repeat 360[fd 1 rt 1] rt 360/20]  
end
```

3) Δοκιμάστε να σχεδιάσετε το παρακάτω σχήμα χρησιμοποιώντας δύο φορές την REPEAT. (πρέπει να οριστεί αρχικά μία διαδικασία για τη μία σημαία (χρησιμοποιώντας την διαδικασία tetragono) και στη συνέχεια μία διαδικασία για τον μύλο η οποία θα επαναλαμβάνει την διαδικασία σημαία – πρέπει να στρέψουμε τη χελώνα κατά 360μοίρες/ n φορές).

1ος τρόπος



```
To tertagono  
Repeat 4[fd 50 rt 90]  
end
```

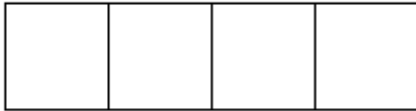
```
To simaia  
Fd 50 tertagono bk 50  
end
```

```
To milos  
Repeat 7[simaia rt 360/7]  
end
```

ή 2ος τρόπος

```
To milos  
Repeat 7[fd 50 repeat 4[fd 50 rt 90] bk 50 rt 360/7]  
end
```

4) Δοκιμάστε να σχεδιάσετε το παρακάτω σχήμα χρησιμοποιώντας 2 φορές την εντολή REPEAT.



```
To tetra  
Repeat 4[repeat 4[fd 50 rt 90] rt 90 fd 50 lt 90]  
end
```

## ΧΡΩΜΑΤΑ ΚΑΙ ΠΑΧΗ ΓΡΑΜΜΩΝ ΣΤΟ ΠΕΡΙΒΑΛΛΟΝ ΤΗΣ LOGO

Στη Logo έχουμε τη δυνατότητα να επιλέξουμε χρώμα με δύο τρόπους:

**A) Πληκτρολογώντας εντολές** και **B) Από τη γραμμή Μενού**

**A)**

ΕΝΤΟΛΕΣ	ΣΥΝΤ/ΓΡΑΦΙΑ	ΑΠΟΤΕΛΕΣΜΑ	ΠΑΡΑΔΕΙΓΜΑ
<b>SETPENCOLOR</b>	<b>SETPC</b>	Αλλάζει το χρώμα του μολυβιού με το οποίο σχεδιάζει η χελώνα, σε πράσινο και φτιάχνει <b>πράσινο σπίτι</b>	<b>Setpc [0 255 0]</b> ↵ <b>Repeat 4[fd 50 rt 90]</b> ↵
<b>SETFLOODCOLOR</b>  <b>FILL</b>	<b>SETFC</b>  <b>FILL</b>	Επιλέγουμε το χρώμα (κίτρινο) με το οποίο θα γεμίσουμε μια <u>κλειστή περιοχή</u> και αφού πάω με <b>PU</b> μέσα, κατεβάζω το μολύβι με <b>PD</b> και δίνουμε εντολή να γελίσει (fill)	<b>PU rt 45 fd 20 PD</b> ↵ <b>Setfc [255 255 0]</b> ↵ <b>fill</b> ↵
<b>SETSCREENCOLOR</b>	<b>SETSC</b>	Αλλάζει το χρώμα στο φόντο της οθόνης σε κόκκινο, αρκεί η χελώνα να είναι κάπου μέσα στην οθόνη.	<b>Setsc [255 0 0]</b> ↵
<b>SETPENSIZE</b>		Ορίζω το πάχος της γραμμής με 2 αριθμούς για ύψος και πλάτος κουκίδας	<b>Setpensize [3 3]</b> ↵

Όπως παρατηρούμε δίπλα σε κάθε εντολή είναι απαραίτητο να βάλουμε και μια **τριάδα αριθμών μέσα σε αγκύλη**. Αυτή είναι η απαραίτητη σύνταξη για τα χρώματα.

Έτσι:

	<b>ΚΟΚΚΙΝΟ</b>	<b>ΠΡΑΣΙΝΟ</b>	<b>ΜΠΛΕ</b>
<b>Μαύρο</b>	0	0	0
<b>Μπλε</b>	0	0	255
<b>Πράσινο</b>	0	255	0
<b>Γαλάζιο</b>	0	255	255
<b>Κόκκινο</b>	255	0	0
<b>Μωβ</b>	255	0	255
<b>Κίτρινο</b>	255	255	0
<b>Λευκό</b>	255	255	255

Με την αντίστοιχη εντολή και την αντίστοιχη τριάδα χρωμάτων μπορώ να έχω το χρώμα που θέλω.

**B)** Το ίδιο αποτέλεσμα μπορώ να έχω αν επιλέξω από τη γραμμή Μενού SET :

**SET\_PENSIZE**

μολυβιού

για το πάχος της γραμμής του

**SET\_PENCOLOR**

μολύβι

για το χρώμα που θα γράφει το

**SET\_FLOODCOLOR**

για το χρώμα μιας κλειστής περιοχής

**SET\_SCREENCOLOR**

για το χρώμα στο φόντο της οθόνης

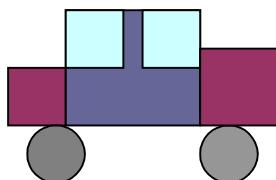


## Η ΕΝΝΟΙΑ ΤΗΣ ΜΕΤΑΒΛΗΤΗΣ

### Παραμετρικές διαδικασίες με μία είσοδο

Στο κεφάλαιο με τις διαδικασίες αναφέραμε ότι ορίζοντας μια διαδικασία μπορούμε να την καλέσουμε, και η χελώνα - υπολογιστής την αναγνωρίζει ως εντολή και την εκτελεί. Αυτό μας δίνει την δυνατότητα να μην χρειάζεται να της υπαγορεύουμε κάθε φορά τι πρέπει να κάνει.

Οι διαδικασίες αυτές όμως έχουν σταθερές τιμές, και δεν είναι πολύ ευέλικτες γιατί τα σχήματα που σχεδιάζονται έχουν πάντα τις ίδιες διαστάσεις. Έτσι αν προσπαθήσουμε να φτιάξουμε ένα σχήμα σαν το παρακάτω, θα πρέπει να τροποποιούμε κάθε φορά τη διαδικασία **tetragono**, αλλάζοντας το μήκος της πλευράς του.



Με τη γλώσσα Logo μας δίνεται η δυνατότητα να δημιουργούμε γενικές διαδικασίες και να μπορούμε καλώντας τις να σχεδιάζουμε σχήματα π.χ. τετράγωνα οποιοδήποτε μήκους, αρκεί στη θέση του σταθερού αριθμού με τον οποίο δηλώνουμε την πλευρά να δηλώσουμε τώρα μια **μεταβλητή**.

Μια μεταβλητή έχει ένα όνομα το οποίο μπορεί να αποτελείται από έναν ή περισσότερους χαρακτήρες, χωρίς ενδιάμεσα κενά. Για να ορίσουμε μια διαδικασία με μεταβλητή θα πρέπει να σημειώσουμε μπροστά από το όνομα της μεταβλητής **άνω-κάτω τελεία (:)**. (π.χ. **:x** ή **:a** ή **:plevra** ή **:plevra.tetragonou** Με τον τρόπο αυτό δηλώνεται στον υπολογιστή ότι η λέξη αυτή αντιπροσωπεύει μια μεταβλητή και όχι το όνομα μιας διαδικασίας.

Εάν θέλουμε για παράδειγμα να δηλώσουμε μια γενική διαδικασία για το σχεδιασμό τετραγώνων με διαφορετικά μήκη πλευρών εργαζόμαστε ως εξής:

Από την γραμμή **menu** επιλέγουμε **FILE / EDIT** και ανοίγει το παράθυρο **Edit Procedure**. Επιλέγουμε **OK** και ανοίγει το παράθυρο σύνταξης όπου θα ορίσουμε τη διαδικασία για το σχεδιασμό τετραγώνου με μεταβλητή. Το όνομα της μεταβλητής δηλώνεται στην επικεφαλίδα, δίπλα στο όνομα της διαδικασίας.

Π.χ.

```
to tetragono :x  
Repeat 4[fd :x rt 90]  
End
```

Έτσι η διαδικασία έχει είσοδο. Αυτό σημαίνει ότι για να εκτελεστεί όταν την καλέσουμε στο πλαίσιο εισαγωγής εντολών, θα πρέπει να πληκτρολογήσουμε το όνομα της διαδικασίας και να δώσουμε μία τιμή για τη μεταβλητή. Αν θέλουμε να σχεδιαστεί ένα τετράγωνο, που οι πλευρές του έχουν μήκος 30, πληκτρολογούμε **tetragono 30**.(με τον ίδιο τρόπο εργαζόμασταν και με τις πρωταρχικές διαδικασίες, όπως rt 90 ή fd 70).

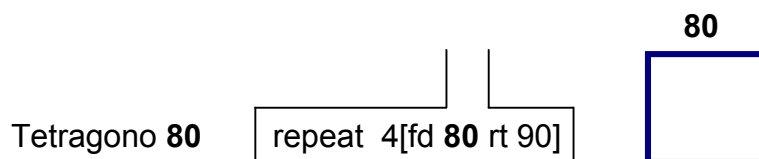
Αν θέλουμε να σχεδιάσουμε ένα τετράγωνο πλευράς 80 θα πληκτρολογήσουμε **tetragono 80** κ.ο.κ.. Θα πρέπει κι εδώ να είμαστε σαφείς και να ακολουθούμε τους κανόνες σύνταξης, κα να δίνουμε τιμή στην διαδικασία.

Σε διαφορετική περίπτωση, αν π.χ. πληκτρολογήσουμε μόνο tetragono χωρίς τιμή η διαδικασία δεν θα εκτελεστεί και θα εμφανιστεί το μήνυμα

```
: not enough inputs to tetragono
```

Οι διαδικασίες που έχουν είσοδο ονομάζονται παραμετρικές και ο τρόπος με τον οποίο εκτελούνται είναι ο εξής:

Η τιμή που πληκτρολογούμε εισάγεται μέσα στον κορμό της διαδικασίας και δίνεται στη μεταβλητή x. Μετά την απόδοση της τιμής στη μεταβλητή εκτελούνται οι πράξεις – όπου υπάρχουν – και η χελώνα σχεδιάζει το συγκεκριμένο σχήμα.

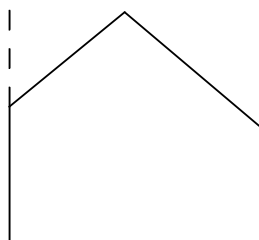


Το μήκος των πλευρών του τετραγώνου είναι ίσο με την τιμή που έχουμε δώσει στη μεταβλητή.

Μπορούμε να δημιουργήσουμε παραμετρικές διαδικασίες με τη βοήθεια της **repeat** αντίστοιχες με την tetragono, για ισόπλευρα τρίγωνα, κανονικά πεντάγωνα εξάγωνα κ.λ.π. Το μήκος της πλευράς θα είναι μεταβλητό, αλλά το μέγεθος της γωνίας ορισμένο, ανάλογα με το είδος του πολυγώνου.

## ΕΞΑΣΚΗΣΗ

1) Ας προσπαθήσουμε να ορίσουμε μια παραμετρική διαδικασία για την κατασκευή ισόπλευρου τριγώνου. Γνωρίζουμε ότι οι γωνίες ενός ισόπλευρου τριγώνου είναι 60 μοίρες η κάθε μία. Αν όμως δώσουμε την οδηγία `repeat 3[fd 70 rt 60]` δεν σχηματίζεται τρίγωνο αλλά το παρακάτω ημιτελές σχήμα.



Θα πρέπει να τονίσουμε ότι τα σχήματα στη Logo δημιουργούνται με την κίνηση της χελώνας. Αν προσπαθήσουμε εμείς οι ίδιοι να κινηθούμε, πόσο θα στρίψουμε; Ποιά γωνία διαγράφουμε με την κίνησή μας; Την εσωτερική ή την εξωτερική γωνία του τριγώνου; Αφού η γωνία του τριγώνου είναι 60 μοίρες, η εξωτερική θα είναι  $180-60=120$ .

2) Προσπαθήστε να ορίσετε παραμετρικές διαδικασίες για τον σχεδιασμό πενταγώνου αλλά και κάθε άλλου κανονικού πολυγώνου.

3) Αν στις προηγούμενες διαδικασίες για κανονικά πολύγωνα χρησιμοποιήσουμε `lt` θα σχεδιαστούν αριστερόστροφα πολύγωνα. Ορίστε παραμετρικές διαδικασίες για τον σχεδιασμό αριστερόστροφων πολυγώνων.

4) Αν δημιουργήσουμε πολλά πολύγωνα με όλο και περισσότερες πλευρές, θα παρατηρήσουμε ότι από ορισμένο πλήθος πλευρών και πάνω, τα πολύγωνα αυτά μοιάζουν με κύκλο.

Έτσι οι διαδικασίες που θα οριστούν παραπάνω, δημιουργούν, για διάφορες τιμές του  $x$ , κύκλους διαφόρων μεγεθών στην οθόνη.

```
to polygon.30 :x
repeat 30[fd :x rt 12]
end
```

```
to polygon.180 :x
repeat 180[fd :x rt 2]
end
```

```
to polygon.60 :x
repeat 60[fd :x rt 6]
end
```

```
to polygon.360 :x
repeat 360[fd :x rt 1]
end
```

### Κύκλος ακτίνας r

Με όσα έχουμε μάθει ως τώρα στη Logo μπορούμε να σχεδιάζουμε κύκλους συγκεκριμένης κάθε φορά ακτίνας δίνοντας στη χελώνα τις εξής οδηγίες :

**repeat 360[fd :x rt 1]**

Ο κύκλος αυτός έχει μήκος  $360 \cdot x$ , γιατί η χελώνα προχωράει 360 φορές από  $x$  βήματα κάθε φορά.  
Πώς μπορούμε να σχεδιάσουμε κύκλο ακτίνας που εμείς θέλουμε;

Γνωρίζουμε ότι ένας κύκλος ακτίνας  $r$  έχει μήκος  $6,28 \cdot r$ . Έτσι η σχέση που συνδέει τα βήματα που διανύει σε κάθε επανάληψη η χελώνα και την ακτίνα του κύκλου που δημιουργείται είναι  $360 \cdot x = 6,28 \cdot r$ . Αν λοιπόν θέλουμε να δημιουργείται κύκλος συγκεκριμένης ακτίνας λύνουμε την εξίσωση ως προς  $x$  για να βρούμε πόσο πρέπει να προχωράει σε κάθε επανάληψη.

Δηλ.  **$x = fd :r \cdot 6,28 / 360$**

Επομένως για να δημιουργήσουμε κύκλο ακτίνας  $r$  θα πρέπει να ορίσουμε την παραμετρική διαδικασία

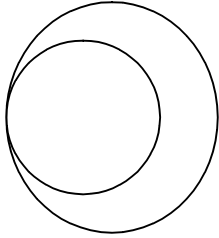
```
to kyklos :r  
repeat 360[fd :r*6.28/360 rt 1]  
end
```

Πειραματιστείτε ορίζοντας παραμετρικές διαδικασίες για κύκλους διαφορετικής ακτίνας.

Τι διαδικασία θα ορίσουμε αν θέλουμε να σχεδιάσουμε ένα μέρος, μόνο του κύκλου δηλαδή ένα **ημικόκλιο** ή **τόξο ακτίνας**;

## **ΕΞΑΣΚΗΣΗ**

1) Ορίστε παραμετρική διαδασία για τον σχεδιασμό επαπτόμενων κύκλων.



2) Δημιουργείστε ομόκεντρους κύκλους, χρησιμοποιώντας παραμετρική διαδικασία με διαφορετικές τιμές ακτίνας. Η χελώνα θα πρέπει να καθοδηγηθεί με τις κατάλληλες εντολές στη νέα αφετηρία.

