

ΓΝΩΡΙΜΙΑ ΜΕ ΤΗ LOGO ΚΑΙ

ΤΟ ΠΕΡΙΒΑΛΛΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ MSWLOGO.

1.1 Εισαγωγικά

Η Logo είναι μια διαδικαστική γλώσσα προγραμματισμού. Σχηματίστηκε ως διασκευή προς το απλούστερο της γλώσσας LISP από τους Wally Freurzeig και Seymour Papert γι’ αυτό και έχει ονομασθεί και Lisp χωρίς παρενθέσεις. Σήμερα είναι γνωστή κυρίως για τα «γραφικά χελώνας» αν και έχει σημαντικές δυνατότητες για την διαχείριση λιστών, αρχείων εισόδου/εξόδου (I/O) κ.α. Η Logo μπορεί να χρησιμοποιηθεί για την διδασκαλία των περισσότερων θεμάτων της ίδιας της Πληροφορικής αλλά και για την διδασκαλία ζητημάτων σε διάφορα αντικείμενα μέσω των «μικρόκοσμων».

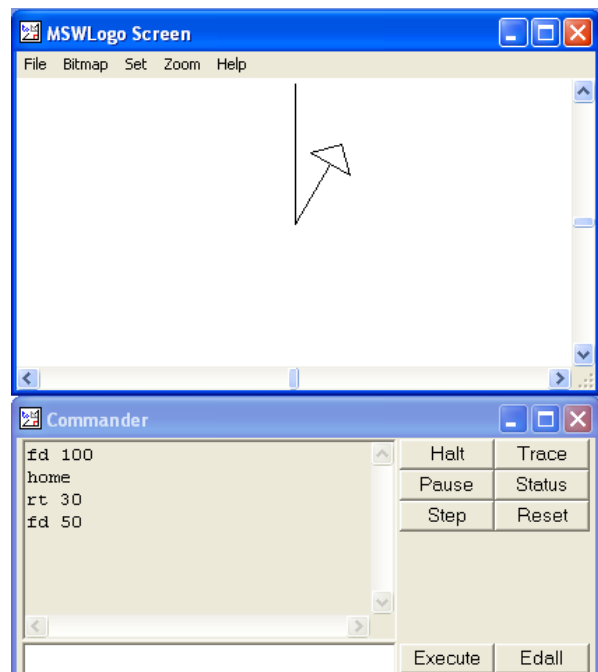
Η γλώσσα χρονολογείται από το 1966 και έχει περισσότερες από 130 διαφορετικές υλοποιήσεις. Το βασικό της πλεονέκτημα ως εκπαιδευτικού περιβάλλοντος είναι εκτός από την χρήση ελκυστικών γραφικών, η παροχή άμεσης μη λεκτικής πληροφορίας ανάδρασης στον χρήστη ώστε να εμπλέκεται προσωπικά και υπεύθυνα σε μια διαδικασία διερευνητικής-ανακαλυπτικής μάθησης.

1.2 Ο προγραμματισμός σε Logo

Η ιδέα είναι: Υπάρχει μια χελώνα που φέρει μολύβι και η οποία μπορεί να διαταχθεί να εκτελεί απλές πράξεις όπως να κινηθεί 100 βήματα εμπρός ή να στρίψει. Από αυτά τα δομικά στοιχεία μπορούν να οικοδομηθούν πολυπλοκότερες πράξεις που θα έχουν ως αποτέλεσμα πολύπλοκα σχήματα όπως τετράγωνα, τρίγωνα, κύκλους ή σχέδια σπιτιών, πλοίων κ.α. Η χελώνα κινείται σχετικά με την τρέχουσα θέση της, δηλαδή η εντολή «LEFT 90» σημαίνει να περιστραφεί η χελώνα κατά 90 μοίρες σε σχέση με τον τρέχοντα προσανατολισμό της. Ο μαθητής μπορεί να καταλάβει, να προβλέψει και να συλλογιστεί την κίνηση της χελώνας, να φανταστεί τι θα έκανε ο ίδιος αν ήταν η χελώνα. Ο Papert ονόμασε αυτό το είδος της σκέψης σωματοσυντονισμένο «body syntonic».

1.3 Προγραμματίζοντας τη χελώνα

Στην ενότητα αυτή θα παρουσιάσουμε παραδείγματα προγραμματισμού σε Logo. Για την εκτέλεση των παραδειγμάτων θα χρησιμοποιήσουμε το περιβάλλον MSW Logo που διανέμεται και μέσω το ιστοχώρου <http://www.e-yliko.gr>. Στην επόμενη εικόνα εμφανίζεται το περιβάλλον και ξεχωρίζουν τα δύο βασικά του τμήματα. Το κεντρικό παράθυρο επάνω περιέχει την χελώνα (τρίγωνο για να φαίνεται και η κατεύθυνση) και τον χώρο που μπορεί αυτή να σχεδιάζει. Στο κάτω παράθυρο μπορούμε να πληκτρολογούμε τις εντολές μία-μία ή



και περισσότερες κάθε φορά. Η εκτέλεση ξεκινά με το «enter» ή με κλικ στο κουμπί «execute». Πάνω από το λευκό πλαίσιο κειμένου που εισάγουμε τις εντολές υπάρχει μια γκριζα περιοχή με τις εντολές που έχουν ήδη εκτελεστεί. Ένα κλικ πάνω σε μια τέτοια εντολή την επαναφέρει στο πλαίσιο εντολής για διόρθωση, επανάληψη κλπ.

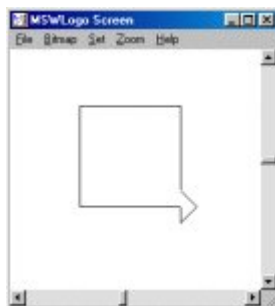
1.3.1 Οι βασικές εντολές της LOGO

Οι βασικές εντολές της Logo εμφανίζονται στον επόμενο πίνακα.

Εντολή	Αποτέλεσμα
FORWARD ή FD K	Εμπρός K βήματα
BACK ή BK K	Πίσω K βήματα
RIGHT ή RT X	Στροφή δεξιά κατά X μοίρες
LEFT ή LT X	Στροφή αριστερά κατά X μοίρες
PENUP ή PU	Σηκώνει το μολύβι από το χαρτί.
PENDOWN ή PD	Κατεβάζει το μολύβι στο χαρτί.
CLEARSCREEN ή CS	Καθαρίζει την οθόνη.
HIDETURTLE ή HT	Κρύβει την χελώνα.
SHOWTURTLE ή ST	Εμφανίζει την χελώνα.
REPEAT 3 [Εντολή]	Επαναλαμβάνει την εντολή ν φορές

1.3.2 Παράδειγμα 1. Τετράγωνο

```
FORWARD 100  
LEFT 90  
FORWARD 100  
LEFT 90  
FORWARD 100  
LEFT 90  
FORWARD 100  
LEFT 90
```



Η παραπάνω ακολουθία εντολών σχηματίζει ένα τετράγωνο 100 μονάδων. Οι εντολές είναι απλές, αλλά η δυνατότητα ομαδοποίησης τους σε διαδικασίες-υποπρογράμματα και η μαζική εκτέλεσή τους μόνο με την επίκληση του ονόματος της διαδικασίας, επιτρέπει την ανάπτυξη πολύπλοκων βιβλιοθηκών εντολών. Οι εντολές μπορούν επίσης να δοθούν σε συντομογραφία π.χ. LT 90 κλπ. Παρατηρώντας το πρόγραμμα βλέπουμε ότι το τετράγωνο σχηματίζεται με την επανάληψη για τέσσερις φορές του ζεύγους των εντολών

```
FORWARD 100  
LEFT 90
```

Αυτό μπορεί να εκφραστεί συντομότερα στην Logo με τη βοήθεια της εντολής επανάληψης REPEAT ως εξής:

```
REPEAT 4 [FORWARD 100 LEFT 90] ή REPEAT 4 [FD 100 LT 90]
```

1.3.3 Παράδειγμα 2. Ορισμός διαδικασίας

Το επόμενο βήμα είναι να ορίσουμε μια νέα λέξη στο λεξιλόγιο της χελώνας στο άκουσμα της οποίας η χελώνα θα σχηματίζει ένα τετράγωνο. Με άλλα λόγια θα ορίσουμε μια νέα διαδικασία.

```
EDIT "square
```

```
<Εμφανίζεται ο συντάκτης κειμένου notepad και πληκτρολογούμε το επόμενο κείμενο - διαδικασία>
```

```
TO SQUARE
```

```
REPEAT 4 [FD 100 RT 90]
```

```
END
```

```
<Αποθηκεύουμε και τερματίζουμε το συντάκτη. Επιστρέφουμε στη Logo. Η χελώνα τώρα, όταν της δίνουμε την εντολή square, σχηματίζει ένα τετράγωνο>
```

1.3.4 Παράδειγμα 3. Ορισμός διαδικασίας με όρισμα

Η διαδικασία που ορίσαμε προηγουμένως έχει ένα μειονέκτημα. Σχηματίζει ορθογώνιο με σταθερή πλευρά. Στο παράδειγμα αυτό θα τροποποιήσουμε τη διαδικασία square ώστε να δέχεται ως παράμετρο το μήκος της πλευράς του ορθογωνίου. Αυτό μπορεί να γίνει με τις επόμενες εντολές:

```
EDIT "square
```

```
TO SQUARE :length
```

```
REPEAT 4 [FD :length RT 90]
```

```
END
```

Με το κλείσιμο του συντάκτη μπορούμε να χρησιμοποιήσουμε την διαδικασία όπως στα παρακάτω παραδείγματα.

```
SQUARE 100
```

```
SQUARE 200
```

1.3.5 Παράδειγμα 4. Περισσότερες διαδικασίες

Στο παράδειγμα αυτό θα φτιάξουμε διαδικασίες και για άλλα σχήματα όπως το ισόπλευρο τρίγωνο, το πεντάγωνο και τελικά θα γενικεύσουμε με την διαδικασία πολύγωνο.

```
TO TRIANGLE :length
```

```
REPEAT 3 [FD :length RT 120]
```

```
END
```

```
TO PENTAGON :length
```

```
REPEAT 5 [FD :length RT 72]
```

```
END
```

```
TO POLYGON :length :sides
```

```
REPEAT :sides [FD :length RT 360.0/:sides]
END
```

Με τις παραπάνω διαδικασίες η επόμενη τριάδα εντολών έχει το ίδιο αποτέλεσμα με την μεθεπόμενη.

```
TRIANGLE 100
SQUARE 100
HEXAGON 100
```

```
POLYGON 100 3
POLYGON 100 4
POLYGON 100 5
```

Με τα παραπάνω φαίνεται η δυνατότητα χρήσης της Logo για την αποσύνθεση ενός προβλήματος σε υπό-προβλήματα αλλά και η γενίκευση των λύσεων με διαδοχικά νοητικά πειράματα.

Όλα τα υποπρογράμματα που έχουμε ορίσει αποτελούν ένα ενιαίο πρόγραμμα που πρέπει να αποθηκευθεί σε αρχείο στον δίσκο προκειμένου να μπορεί να ξαναχρησιμοποιηθεί σε περίπτωση που κλείσουμε την εφαρμογή. Για τον σκοπό αυτό δίνουμε την επόμενη εντολή ή επιλέγουμε FILE|SAVE.

```
SAVE "SHAPES.LGO
BYE
LOAD "SHAPES.LGO
```

1.3.6 Παράδειγμα 5. Σχήματα από περιστροφή

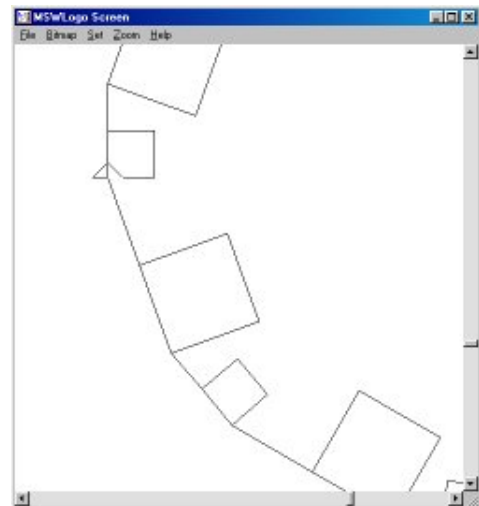
Ορίστε την επόμενη διαδικασία που σχηματίζει μια καρέκλα

```
TO CHAIR :thesize
REPEAT 4 [FD :thesize RT 90] FD :thesize
FD :thesize
END
```

Δοκιμάστε την επόμενη εντολή:

```
REPEAT 9 [CHAIR 50 RT 20 CHAIR 100 WAIT
50 RT 20]
```

Θα πρέπει να δείτε ένα σχήμα όπως στην διπλανή εικόνα.



1.3.7 Παράδειγμα 6. Ο κύκλος

Ο σχεδιασμός του κύκλου γίνεται με το επόμενο υποπρόγραμμα.

```
to circle :r
make "d :r*pi/180
repeat 360 [fd :d rt 1]
end
```

1.3.8 Παράδειγμα 7. Αναδρομικά υποπρογράμματα – το σπирάλ

Αναδρομικά ονομάζονται τα υποπρογράμματα που επικαλούνται τον εαυτό τους. Για την αποφυγή της ατέρμονης επανάληψης υπάρχει μια απλή συνθήκη τερματισμού που εισάγει και την απλή δομή επανάληψης if. Ορίστε το επόμενο υποπρόγραμμα:

```
to spiral :size
  if :size > 30 [stop] ; a stop condition
  fd :size rt 15      ; many lines of action
  spiral :size *1.02  ; the tailend recursive
call
end
Δώστε: spiral 10
```



1.3.9 Παράδειγμα 8. Οι μεταβλητές στη LOGO

Στη Logo υπάρχουν τρεις τύποι δεδομένων: word, list και array. Οι αριθμοί είναι ειδικές περιπτώσεις της λέξης (word). Ο τύπος των μεταβλητών δεν είναι απαραίτητο να οριστεί αλλά συνάγεται από τα συμφραζόμενα. Δύο σύμβολα που είναι σημαντικά σχετικά με τις μεταβλητές είναι η άνω κάτω τελεία «:» και τα εισαγωγικά «"». Οι άνω κάτω τελείες σημαίνουν την τιμή μιας μεταβλητής και τα εισαγωγικά σημαίνουν κυριολεκτικές τιμές (εκφράσεις που αποτιμούνται ακριβώς όπως αναγράφονται). Για παράδειγμα το επόμενο τμήμα θα τυπώσει 10.

```
make "x 4
make "y 6
make "sum :x+:y
print :sum
```

Η εντολή make παίρνει δύο ορίσματα ένα όνομα μεταβλητής και μια τιμή. Το αποτέλεσμα της είναι η δημιουργία μιας μεταβλητής με το δεδομένο όνομα (εφόσον αυτή δεν υπάρχει ήδη) και η απόδοση του δεύτερου ορίσματος ως αρχική τιμή. Σε περίπτωση που υπάρχει ήδη μια μεταβλητή με το ίδιο όνομα, τότε η εντολή make λειτουργεί ως εντολή ανάθεσης τιμής.

Το όνομα της μεταβλητής μπορεί να είναι ό,τιδήποτε η LOGO βλέπει ως λέξη. Είναι επίσης case-insensitive.

1.4 Περισσότερες πηγές

Η Logo διαθέτει πληθώρα δυνατοτήτων που δεν παρουσιάστηκαν στα παραπάνω παραδείγματα όπως η δυνατότητα παραγωγής μουσικής, γεννήτριες τυχαίων αριθμών, επεξεργασίας πολυμέσων κ.α. Για περισσότερη κατανόηση της γλώσσας ο ενδιαφερόμενος αναγνώστης παραπέμπεται στην τεκμηρίωση των περιβαλλόντων ανάπτυξης και στις πλούσιες πηγές του διαδικτύου π.χ.:

LOGO FOUNDATION <http://el.media.mit.edu/logo-foundation/>

MSWLogo <http://www.softronix.com/logo.html>

<http://www.dapontes.gr/>

<http://users.sch.gr/glezou/>

<http://blogs.sch.gr/glezou/>

<http://logogreekworld.ning.com/>