

Σημειώσεις Matlab

Γενικά

<code>a = 2</code>	Εκχώρηση της τιμής 2 στη μεταβλητή a.
<code>b = 3;</code>	Εκχώρηση της τιμής 3 στη μεταβλητή b, χωρίς να εμφανίζεται το αποτέλεσμα.
<code>c = a + b</code>	Δίνει το άθροισμα των a και b.
<code>x = 1, y = -2</code>	Δύο εντολές σε μία γραμμή, χωρισμένες με κόμμα (εμφανίζονται και τα δύο αποτελέσματα).
<code>b</code>	Εμφανίζει την τιμή που έχουμε δώσει προηγουμένως: 3.
<code>clear b</code>	Σβήνει τη μεταβλητή b από τη μνήμη.
<code>b</code>	Μήνυμα λάθους, γιατί έχουμε σβήσει το b.
<code>clear</code>	Σβήνει όλες τις μεταβλητές από τη μνήμη.

Εισαγωγή Πινάκων

<code>v = [7 -2 5]</code>	Δημιουργεί ένα διάνυσμα, αποτελούμενο από μία γραμμή.
<code>v = [7,-2, 5]</code>	Ομοίως με προηγουμένως.
<code>w = [1; 2; 3]</code>	Δημιουργεί ένα διάνυσμα, αποτελούμενο από μία στήλη.
<code>A = [4 -2 5; -1 7 -6; 2 3 5]</code>	Εισαγωγή πίνακα 3x3.
<code>A = [4 -2 5 -1 7 -6 2 3 5]</code>	Εισαγωγή του ίδιου πίνακα (πατάμε Enter στο τέλος της γραμμής).

Βασικές Συναρτήσεις

<code>va = abs(v)</code>	Δημιουργεί το διάνυσμα va του οποίου κάθε στοιχείο ισούται με την απόλυτη
--------------------------	---

τιμή του αντίστοιχου στοιχείου του ορίσματος v .

- $\text{sq} = \text{sqrt}(v)$ Δημιουργεί το διάνυσμα sq του οποίου κάθε στοιχείο ισούται με την τετραγωνική ρίζα του αντίστοιχου στοιχείου του ορίσματος v . Αν κάποιο στοιχείο είναι αρνητικό, το αντίστοιχο στοιχείο του sq θα είναι μιγαδικός.
- $y = \text{sin}(\pi/6)$ Επιστρέφει το ημίτονο του $\pi/6$. Όμοια για cos (συνημίτονο), tan (εφαπτομένη), asin (τόξο ημ.), acos (τόξο συν.), atan (τόξο εφ.).
- $y = \text{exp}(4)$ Επιστρέφει το e^4 , όπου e η βάση των νεπέριων λογαρίθμων.
- v' Επιστρέφει τον ανάστροφο του πίνακα v .
- $C1 = \text{eye}(4)$ Δίνει το μοναδιαίο τετραγωνικό πίνακα με διάσταση 4.
- $C2 = \text{eye}(5,3)$ Δίνει το μοναδιαίο πίνακα με 5 γραμμές και 3 στήλες.
- $C3 = \text{ones}(3)$ Δίνει τον τετραγωνικό πίνακα με διάσταση 3 και όλα τα στοιχεία μονάδες.
- $C4 = \text{eye}(2,3)$ Δίνει τον πίνακα με 5 γραμμές και 3 στήλες και όλα τα στοιχεία μονάδες.
- $C5 = \text{zeros}(2)$ Δίνει τον τετραγωνικό πίνακα με διάσταση 2 και όλα τα στοιχεία μηδέν.
- $A1 = \text{diag}(v)$ Επιστρέφει το διαγώνιο (τετραγωνικό) πίνακα με στοιχεία διαγωνίου τα στοιχεία του v , αφού το όρισμα v είναι διάνυσμα.
- $A2 = \text{diag}(v,k)$ Επιστρέφει τον πίνακα με όλα τα στοιχεία 0, εκτός αυτών που ανήκουν στην k -διαγώνιο και που θα παίρνουν τις τιμές από τα στοιχεία του διανύσματος v . Για $k = 0$ τα στοιχεία του v μπαίνουν στην κύρια διαγώνιο, για $k = 1$ μπαίνουν στη διαγώνιο που βρίσκεται ακριβώς πάνω από την κύρια διαγώνιο, για $k = -1$ μπαίνουν στη διαγώνιο που βρίσκεται ακριβώς κάτω από την κύρια διαγώνιο κοκ. Οι διαστάσεις του πίνακα θα είναι οι ελάχιστες δυνατές ώστε να χωρούν το διάνυσμα v .
- $w = \text{diag}(A,k)$ Παίρνει τα στοιχεία της k -διαγωνίου του τετραγωνικού πίνακα A και σχηματίζει με αυτά ένα διάνυσμα-στήλη. Το k ορίζεται όπως πριν. Η συνάρτηση diag είναι η ίδια με τις δύο προηγούμενες περιπτώσεις, απλώς συμπεριφέρεται διαφορετικά ανάλογα με τη μορφή του πρώτου ορίσματος.

ΣΗΜΕΙΩΣΗ: Όλες οι συναρτήσεις (εκτός αν αναφέρεται διαφορετικά), όταν παίρνουν σαν όρισμα πίνακα αντί για αριθμό, επιστρέφουν ένα πίνακα με ίδιες διαστάσεις, του οποίου το κάθε στοιχείο ισούται με την τιμή της συνάρτησης του αντίστοιχου στοιχείου του αρχικού πίνακα. Ισχύει π.χ. για sqrt , abs , sin κτλ.

Ακολουθίες Αριθμών

- $u1 = [4:10]$ Δίνει το διάνυσμα με στοιχεία τους αριθμούς 4,5,6,7,8,9,10 (βήμα 1).
- $v1 = [3:2:8]$ Δίνει το διάνυσμα με στοιχεία τους αριθμούς 3,5,7 (βήμα 2).
- $t1 = [1:-1/2:-1]$ Δίνει το διάνυσμα με στοιχεία τους αριθμούς 1,1/2,0,-1/2,-1 (βήμα -1/2).
- $u2 = \text{linspace}(4,10,7)$ Δίνει το διάνυσμα με τους 7 αριθμούς από το 4 έως και το 10 - δίνει

ό,τι και το u1.

t1 = logspace(0,3,4) Δίνει την ακολουθία των 4 αριθμών από 10^0 έως 10^3 .

Διαστάσεις και Δείκτες Στοιχείων

n = length(t1) Επιστρέφει το μήκος του διανύσματος t1 ή για πίνακα τη μέγιστη διάστασή του.

[nr,nc] = size(A) Εκχωρεί στο nr τις γραμμές του πίνακα A και στο nc τις στήλες του A.

A(1,2) Επιστρέφει το στοιχείο που ανήκει στην πρώτη γραμμή και δεύτερη στήλη.

A(4,5) = 3 Θέτει ως τιμή του στοιχείου 4,5 την τιμή 3 (αν υπολείπονται γραμμές έως την 4^η ή στήλες έως την 5^η, τα υπόλοιπα στοιχεία των επιπλέον γραμμών συμπληρώνονται με μηδενικά).

B = [1:5, 2:6] Δημιουργεί τον πίνακα $\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}$.

B = [1:5; 2:6] Δημιουργεί τον πίνακα $\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 6 \end{bmatrix}$.

B(2,:) Επιστρέφει ως διάνυσμα γραμμή τη 2^η γραμμή του B.

B(:,5) Επιστρέφει ως διάνυσμα στήλη τη 5^η στήλη του B.

B(1:2, end-1:end) Επιστρέφει ένα πίνακα με τα 4 στοιχεία των 2 πρώτων γραμμών και των 2 τελευταίων στηλών, δηλαδή τον πίνακα $\begin{bmatrix} 4 & 5 \\ 5 & 6 \end{bmatrix}$.

B(:, 2:4) = [] Διαγράφει τη 2^η, 3^η και 4^η στήλη, εμφανίζει το νέο πίνακα.

Παραγωγή Πινάκων με Τυχαία Στοιχεία

v = rand(1,3) Σχηματίζει έναν πίνακα με 1 γραμμή και 3 στήλες και στοιχεία τυχαίας πραγματικής τιμής από το 0 έως και το 1.

A = rand(3) Σχηματίζει έναν τετραγωνικό πίνακα με διάσταση 3 και τυχαία στοιχεία.

B = rand(3) Σχηματίζει έναν πίνακα 3x3 με τυχαία στοιχεία (εν γένει διαφορετικά του A).

Πράξεις Πινάκων

S = A+B Επιστρέφει το άθροισμα των A και B, που έχουν κοινές διαστάσεις.

P = A*B Εκτελεί το γινόμενο του A(m x n) επί τον B(n x t) και το εκχωρεί στον P(m x t).

X1.*X2 Για τους πίνακες X1 και X2, που έχουν κοινές διαστάσεις, δημιουργεί πίνακα του οποίου κάθε στοιχείο ισούται με το γινόμενο των αντίστοιχων των X1, X2.

X1./X2 Παρόμοια με πριν, αλλά το κάθε στοιχείο προκύπτει με διαίρεση του

	αντίστοιχου του X1 δια του X2.
A^3	Εκτελεί τον πολ/μό A*A*A (ισχύει για τετραγωνικό μόνο).
X3 = X1.^3	Δημιουργεί τον πίνακα X3 ίδιων διαστάσεων με τον X1 και κάθε στοιχείο του ισούται με την 3 ^η δύναμη του αντίστοιχου του X1.
X4 = X1.^X2	Για τους πίνακες X1 και X2, που έχουν κοινές διαστάσεις, δημιουργεί πίνακα X4 του οποίου κάθε στοιχείο ισούται με το αντίστοιχο του X1 υψωμένο στο αντίστοιχο του X2.
A^(-1)	Επιστρέφει τον αντίστροφο πίνακα του A.
inv(A)	Επίσης επιστρέφει τον αντίστροφο πίνακα του A.
A\B	Αντίστροφη διαίρεση: εκτελεί την πράξη inv(A)*B, όπου ο B δεν πρέπει απαραίτητα να είναι τετραγωνικός πίνακας, αρκεί όμως το πλήθος των γραμμών του να είναι ίσος με τη διάσταση του τετραγωνικού A. Η αντίστροφη διαίρεση είναι χρήσιμη για την επίλυση του συστήματος $Ax = b$.
max(A)	Αν το A είναι πίνακας, επιστρέφει το διάνυσμα-γραμμή του οποίου κάθε στοιχείο ισούται με το μέγιστο από τα στοιχεία της στήλης που ανήκει. Αν το A είναι διάνυσμα (στήλη ή γραμμή), επιστρέφει το μέγιστο αριθμό.
max(A,[],1)	Αν το A είναι πίνακας, επιστρέφει το διάνυσμα-γραμμή του οποίου κάθε στοιχείο ισούται με το μέγιστο από τα στοιχεία της στήλης που ανήκει.
max(A,[],2)	Αν το A είναι πίνακας, επιστρέφει το διάνυσμα-στήλη του οποίου κάθε στοιχείο ισούται με το μέγιστο από τα στοιχεία της γραμμής που ανήκει.
min(A)	Ίδια σύνταξη με το max, αλλά επιστρέφει τα ελάχιστα.
sum(A)	Αν το A είναι πίνακας, επιστρέφει το διάνυσμα-γραμμή του οποίου κάθε στοιχείο ισούται με το άθροισμα των στοιχείων της στήλης που ανήκει. Αν το A είναι διάνυσμα (στήλη ή γραμμή), επιστρέφει το άθροισμα όλων των στοιχείων.
sum(A,1)	Αν το A είναι πίνακας, επιστρέφει το διάνυσμα-γραμμή του οποίου κάθε στοιχείο ισούται με το άθροισμα των στοιχείων της στήλης που ανήκει.
sum(A,2)	Αν το A είναι πίνακας, επιστρέφει το διάνυσμα-στήλη του οποίου κάθε στοιχείο ισούται με το άθροισμα των στοιχείων της γραμμής που ανήκει.
prod(A)	Ίδια σύνταξη με το sum, αλλά επιστρέφει τα γινόμενα.

Μετατροπές Πινάκων

C = reshape([1 2 3;
4 5 6;
7 8 9;
10 11 12], 2,6) Ανασχηματίζει τον 4x3 πίνακα να έχει 2 γραμμές και 6 στήλες δημιουργώντας τον $C = \begin{bmatrix} 1 & 7 & 2 & 8 & 3 & 9 \\ 4 & 10 & 5 & 11 & 6 & 12 \end{bmatrix}$, δηλαδή διαβάζει ανά στήλη τα στοιχεία.

`E = C(:)` Μετατρέπει τον πίνακα C σε διάνυσμα-στήλη.

Μιγαδικοί Αριθμοί και Συναρτήσεις Μιγαδικών

`z1 = 1-2*i` Μιγαδικός αριθμός με πραγματικό μέρος 1 και φανταστικό -2.

`z2 = 1-2*j` Ο ίδιος με τον z1 (τα i και j να μη χρησιμοποιούνται σαν ακέραιοι).

`real(z1)` Επιστρέφει το πραγματικό μέρος.

`imag(z1)` Επιστρέφει το φανταστικό μέρος.

`angle(z1)` Επιστρέφει τη γωνία του διανύσματος που δημιουργείται στο σύστημα των μιγαδικών

Συναρτήσεις Συμβολοσειρών

`s = 'A string example'` Οτιδήποτε μέσα σε μονές αποστρόφους θεωρείται συμβολοσειρά.

`s = num2str(pi,3)` Αποθηκεύει στο s την τιμή '3.142'. Μετατρέπει έναν αριθμό σε συμβολοσειρά με όσα σημαντικά ψηφία ζητάμε μέσω του 2^{ov} ορίσματος.

`n = str2num('2.34')` Αποθηκεύει στο n την τιμή 2.3400... Μετατρέπει μια συμβολοσειρά σε αριθμό με όση ακρίβεια του έχουμε δώσει.

Πολυώνυμα

`polyval([1 0 -2 12], 1.5)` Βρίσκει την τιμή του πολυωνύμου $x^3 - 2x + 12$ για $x = 1.5$.

`roots([1 0 -2 12])` Βρίσκει όλες τις ρίζες πραγματικές και μιγαδικές του πολυωνύμου.

`poly([0 -1 2])` Επιστρέφει το πολυώνυμο με ρίζες τα στοιχεία του ορίσματος (το πολυώνυμο επιστρέφεται σε μορφή διανύσματος των συντελεστών).

Γραφικές Παραστάσεις 2 Διαστάσεων

`t = [0: pi/20: 2*pi];`

`plot(t, cos(t))` Δημιουργούμε μια ακολουθία $t \in [0, 2\pi]$ με βήμα $\pi/20$ και φτιάχνουμε το διάγραμμα του συνημίτονου.

`grid on` Εμφανίζει τη διαγράμμιση στην τελευταία γραφική παράσταση.

`title('The Cos Function')` Εμφανίζει τον τίτλο στο γράφημα.

`xlabel('t'); ylabel('Cos')` Εμφανίζει τα ονόματα των αξόνων x και y αντίστοιχα.

`axis([0 2*pi -1 1])` Θέτει τα όρια του γραφήματος για το $t \in [0, 2\pi]$ και για τις συναρτήσεις $y \in [-1, 1]$.

`plot(t,cos(t),'k-', t,sin(t),'ro')` Δημιουργεί στο ίδιο γράφημα τη γραφική παράσταση του συνημίτονου με συνεχή μαύρη γραμμή και του ημίτονου με κόκκινους κύκλους.

`legend('Cos', 'Sin')` Εμφανίζει το υπόμνημα του γραφήματος.

Λογαριθμική Κλίμακα

`semilogy(t, abs(cos(t)))` Κάνει το διάγραμμα της απόλυτης τιμής του συνημίτονου, αλλά στον κατακόρυφο άξονα η κλίμακα είναι λογαριθμική.

`semilogx(t, abs(cos(t)))` Παρόμοια με πριν, αλλά για τον οριζόντιο άξονα.

`loglog(t, abs(cos(t)))` Λογαριθμική κλίμακα και στους δύο άξονες.

Πολλαπλά Γραφήματα

`subplot(2,2,1)` Εμφανίζει σε ένα παράθυρο 4 (2 γραμμές x 2 στήλες) χώρους για γραφήματα και θέτει σαν ενεργό το 1^ο, όπου και θα εμφανιστεί το επόμενο γράφημα, όταν δοθεί εντολή τύπου `plot`, `loglog` κτλ.

Γραφήματα 3 Διαστάσεων

`x = linspace(-20, 20, 80);`

`y = linspace(-2*pi, 2*pi, 160);`

`[X,Y] = meshgrid(x,y)` Από τα διανύσματα x και y δημιουργούνται τα X και Y που είναι διαστάσεων $\text{length}(y) \times \text{length}(x)$ και τα στοιχεία μιας στήλης του X είναι ίδια μεταξύ τους, όπως επίσης ίδια μεταξύ τους είναι και τα στοιχεία μιας γραμμής του Y . Δηλαδή ο πίνακας X είναι το διάνυσμα-γραμμή x επαναλαμβανόμενο 160 φορές κατακόρυφα και ο πίνακας Y είναι το διάνυσμα-στήλη y' επαναλαμβανόμενο 80 φορές οριζόντια.

`surf(X, Y, X.^2.*cos(Y))` Δίνει το τρισδιάστατο γράφημα του $x^2 \cos(y)$, για $x \in [-20, 20]$ και $y \in [-2\pi, 2\pi]$.

ΠΡΟΣΟΧΗ στη χρήση της τελείας `.` πριν από το σύμβολο μιας πράξης (`*` `/` `^` `\` κτλ.), όταν θέλουμε να δηλώσουμε πράξη στοιχείου προς στοιχείο.

Άλλες Εντολές

`clc` Σβήνει τις προηγούμενες εντολές και τα αποτελέσματά τους από το παράθυρο εντολών (command window), αλλά όχι τις μεταβλητές από

τη μνήμη.
help exp Μας δίνει πληροφορίες για τη σύνταξη της συνάρτησης exp.
lookfor exponential Επιστρέφει μια λίστα από θέματα που περιέχουν τη λέξη exponential, τα οποία μπορούμε να κοιτάξουμε το καθένα ξεχωριστά με την εντολή help.

Συνδυασμοί πλήκτρων

Ctrl+C ή **Ctrl+Break** Τερματισμός οποιας εντολής βρίσκεται σε εξέλιξη.

Δημιουργία συνάρτησης

File > New > M-file Δημιουργία νέου m-file

File > Open ή **Ctrl+O** Άνοιγμα αρχείου

- Πατάμε **F5** από τον m-file editor ή πληκτρολογούμε το όνομα του αρχείου χωρίς την κατάληξη στο παράθυρο εντολών (command window), για να τρέξουμε το m-file.
- Για να ορίσουμε ένα m-file ως συνάρτηση με όνομα namefun, ορίσματα in1,in2,in3 κτλ και αποτελέσματα out1,out2, στην πρώτη γραμμή του αρχείου γράφουμε

```
function [out1,out2] = namefun(in1,in2,in3)
```

ή

```
function out = namefun(in1,in2,in3)
```

αν η συνάρτηση επιστρέφει μόνο έναν αριθμό out.

ΠΡΟΣΟΧΗ: Το όνομα του m-file πρέπει να είναι το ίδιο με το όνομα που χρησιμοποιείται στην πρώτη γραμμή, δηλαδή namefun.m

- Ό,τι υπάρχει δεξιά από το σύμβολο % θεωρείται σχόλιο. Οι γραμμές που είναι εξολοκλήρου σχόλιο και βρίσκονται στην αρχή του m-file ή αμέσως μετά τον ορισμό της συνάρτησης φαίνονται με την εντολή help.

Παράδειγμα συνάρτησης

```
%This function finds the sum and the product of two numbers
```

```
function [sum1,prod1] = praxeis(a1,a2)
```

```
sum1 = sum([a1,a2]);
```

```
prod1 = prod([a1,a2]);
```

Τα παραπάνω τα σώζουμε σε αρχείο με όνομα praxeis.m και μετά γράφουμε στη γραμμή εντολών [s,p] = praxeis(2,3) το οποίο καταχωρεί στη μεταβλητή s το άθροισμα δηλαδή 5 και στη μεταβλητή p το γινόμενο δηλαδή 6.

Βασικές εντολές

`x = input('Enter a value for x')`

Μήνυμα για εξωτερική εισαγωγή μεταβλητής (δεν προτείνεται)

`eps`

Παράμετρος που δίνει το μέγιστο απόλυτο σφάλμα κάθε πράξης στη Matlab στο συγκεκριμένο υπολογιστή (π.χ. $2.22044604925031e-016$)

`format long`

Θέτει ως προεπιλεγμένο τρόπο εμφάνισης των αριθμών τα 15 δεκαδικά ψηφία.

`format short`

Όμοια, εμφανίζει 4 δεκαδικά ψηφία.

Ανάγνωση και εγγραφή σε αρχείο

`save('xyfile','x','y')` Δημιουργεί το αρχείο `xyfile.mat` και αποθηκεύει τα `x` και `y`.

`load xyfile`

Φορτώνει στο workspace τις μεταβλητές που είναι αποθηκευμένες (`x` και `y`) στο `xyfile.mat`.

`save xyfile x y -ascii`

Δημιουργεί το αρχείο `xyfile.txt` και αποθηκεύει τα `x` και `y` σε μορφή κειμένου.

`D = load('xyfile.txt')`

Δημιουργεί από τα `x` και `y`, που είναι αποθηκευμένα στο `xyfile.txt` τον πίνακα `D`. Τα `x` και `y` όμως πρέπει να συμφωνούν σε διαστάσεις (ίδιο αριθμό στηλών). Γενικά δε συνίσταται το φόρτωμα από αρχείο τύπου `-ascii`.

`fnum = fopen('results.dat','wt')`

Ανοίγει το αρχείο `results.dat` για εγγραφή (δημιουργεί ένα νέο αρχείο είτε αυτό υπάρχει είτε όχι, σβήνοντας προηγούμενα δεδομένα) και καταχωρεί μια τιμή (θετική για επιτυχημένη δημιουργία, -1 για αποτυχημένη) στο `fnum`, που είναι και ο αριθμός αναγνώρισης του αρχείου.

- Η άδεια μπορεί να είναι `'wt'`, `'rt'`, όπου ανοίγει ένα υπάρχον αρχείο για ανάγνωση ή `'at'`, όπου δημιουργεί το αρχείο, αν αυτό δεν υπάρχει ή προσθέτει στο τέλος του αρχείου τα νέα δεδομένα, αν υπάρχει ήδη.

`fprintf(fnum, 'Example %s:\n %7.4f %11.3e\n \t...\n', 'One', -100*pi, pi)`

Εκτυπώνει στο αρχείο με αριθμό αναγνώρισης `fnum` (ή αν έχει παραληφθεί, στο command window) τις μεταβλητές `'One'` (string), `-100*pi` και `pi` με format που δίνεται από το δεύτερο όρισμα. Το `%s` εμφανίζει την πρώτη μεταβλητή κατά σειρά σαν συμβολοσειρά. Το `%7.4f` σημαίνει ότι ο (δεύτερος) αριθμός θα εμφανίζεται με 4 δεκαδικά ψηφία και μαζί με τα ψηφία του ακέραιου μέρους, την τελεία και το πρόσημο θα καταλαμβάνει τουλάχιστον 7 θέσεις. Όμοια για το `%11.3e` με τη διαφορά ότι εμφανίζει τον αριθμό σε μορφή $a \cdot 10^b$. Το `\t` προσθέτει το χαρακτήρα tab και το `\n` αλλάζει γραμμή. Κάθε άλλος χαρακτήρας εμφανίζεται κανονικά. Οι δύο αριθμοί στοιχίζονται δεξιά

στο χώρο που τους αναλογεί, επειδή ο πρώτος αριθμός μετά το % είναι θετικός. Αν ήταν αρνητικός, θα στοιχίζονταν αριστερά. Το αποτέλεσμα στο παράδειγμα είναι:

```
Example One:  
-314.1593  3.142e+000  
...
```

`a = fgetl(fnum)` Διαβάζει μια γραμμή από το αρχείο με αριθμό αναγνώρισης `fnum` και το καταχωρεί στη μεταβλητή `a` ως συμβολοσειρά. Για το προηγούμενο δίνει: `a = Example One:`

`b = fscanf(fnum, '%f %f', 3)` Διαβάζει 2 αριθμούς από το αρχείο με αριθμό αναγνώρισης `fnum` και τους αποθηκεύει στο διάνυσμα - στήλη `b`. Το τρίτο όρισμα 3 σημαίνει ότι θα διαβάσει το πολύ 3 αριθμούς (εδώ δεν επηρεάζει). Αν ο αριθμός είναι μεγαλύτερος από το πλήθος των μεταβλητών που καλείται να διαβάσει μέσω του 2^{ου} ορίσματος, τότε επαναλαμβάνεται π.χ. ισοδύναμα θα μπορούσαμε να δώσουμε `b = fscanf(fnum, '%f', 2)`. Η εμφάνιση του αποτελέσματος επηρεάζεται από το προεπιλεγμένο format:
`b = -314.1593
 3.1420`

Τόσο για την `fgetl` όσο και για την `fscanf`, η ανάγνωση αρχίζει από το σημείο που σταμάτησε τελευταία φορά η ανάγνωση και τελειώνει όταν έχει τελειώσει η γραμμή για την `fgetl` ή όταν έχει διαβάσει τις μεταβλητές για την `fscanf`.

`fclose(fnum)` Τερματίζει την επικοινωνία του προγράμματος με το αρχείο που έχει αριθμό αναγνώρισης `fnum` και σώζει τα αποτελέσματα.

`type results.dat` Εμφανίζει τα περιεχόμενα του αρχείου `results.dat` που βρίσκεται στον ενεργό κατάλογο στο `command window`.

Σύγκριση τιμών μεταβλητών

`x=3<4, y=4>=5` Επιστρέφει `x=1` και `y=0`. Το 1 αντιπροσωπεύει το αληθές και το 0 το ψευδές. Τα 0 και 1 είναι τύπου `logical` και όχι `double`. Επίσης χρησιμοποιούμε `==` (ισότητα), `~` (αντίθετο), `&` (και), `|` (ή). π.χ. το `a = 3==3 & (4~4 | ~1)` δίνει `a=0`.

Εντολές υπό συνθήκες

```
if x>0
    disp('Positive')
elseif x==0
    disp('Zero')
elseif x<0
    disp('Negative')
else
    disp('Error')
end
```

Εμφανίζει το ανάλογο μήνυμα, για θετικό, μηδενικό, αρνητικό και εσφαλμένο (μιγαδικό ή που δεν μπορεί να υπολογιστεί, π.χ. NaN – Not a Number) αντίστοιχα.

Το disp εμφανίζει μήνυμα στο command window.

```
switch sign(x)
case 1
    disp('Positive')
case 0
    disp('Zero')
case -1
    disp('Negative')
otherwise
    disp('Error')
end
```

Ισοδύναμος κώδικας με τον προηγούμενο. Λιγότερο ευέλικτος, αφού οι τιμές που εξετάζονται πρέπει να είναι ακέραιοι, λογικά 0 και 1 ή συμβολοσειρές.

Το sign επιστρέφει 1,0,-1 για θετικό, μηδενικό και αρνητικό όρισμα αντίστοιχα.

Βρόχοι επανάληψης

```
for k=1:10
    fprintf('%3.0f',k)
end
```

Δίνει k = 1 2 3 4 5 6 7 8 9 10

```
k=1;
while k<=10
    fprintf('%3.0f',k)
    k=k+1;
end
```

Δίνει ότι και πριν

Διάφορες εντολές

<code>break</code>	Άμεσος τερματισμός του βρόχου επανάληψης
<code>return</code>	Άμεσος τερματισμός της συνάρτησης
<code>y = [3 6 9 12 15];</code>	
<code>x = [3 4 1 5 2];</code>	
<code>z = y(x)</code>	Επιστρέφει <code>z = 9 12 3 15 6</code> . Το <code>x</code> λειτουργεί ως δείκτης.
<code>zb = z>10</code>	Επιστρέφει <code>zb = 0 1 0 1 0</code> . Το <code>zb</code> είναι logical array.
<code>z(zb)</code>	Επιστρέφει <code>12 15</code> . Το <code>zb</code> λειτουργεί ως δείκτης (λογικός). Αν δημιουργήσουμε εμείς ένα <code>zb = [0 1 0 1 0]</code> και δώσουμε μετά <code>z(zb)</code> , θα βγάλει σφάλμα, γιατί τα 0 και 1 δεν είναι λογικά αλλά ακέραιοι συμπεριλαμβανομένου του 0.
<code>zi = find(z>10)</code>	Επιστρέφει <code>2 4</code> , δηλαδή τους δείκτες (θέσεις) των στοιχείων που ικανοποιούν τη συνθήκη.
<code>z(zi)</code>	Επιστρέφει <code>12 15</code> . Δίνει ότι και το <code>z(zb)</code> . Εδώ δεν υπάρχει σφάλμα, γιατί οι ακέραιοι 2 και 4 είναι στα όρια 1 έως <code>length(z)</code> εδώ 5.
<code>z(z>10)</code>	Όμοια με πριν
<code>all(zb)</code>	Επιστρέφει 0 (false). Επιστρέφει 1, όταν όλα τα στοιχεία είναι διάφορα του 0.
<code>any(zb)</code>	Επιστρέφει 1 (true). Επιστρέφει 1, όταν κάποιο στοιχείο είναι διάφορο του 0.
<code>nargin, nargout</code>	Επιστρέφουν τον αριθμό των εισαγόμενων και εξαγόμενων μεταβλητών σε μία συνάρτηση.
<code>global v</code>	Κάνει την παράμετρο <code>v</code> κοινή για όλες συναρτήσεις έχουν την εντολή αυτή (συμπεριλαμβανομένου και του workspace).
<code>feval('cos',0)</code>	Επιστρέφει 1, ό,τι δηλαδή θα έδινε και το <code>cos(0)</code> .
<code>trig_fun = inline('cos(x)*sin(x)')</code>	Ορίζει τη συνάρτηση <code>trig_fun</code> που φαίνεται από το workspace και από κάθε συνάρτηση, ανεξάρτητα από πού την έχουμε ορίσει.