

Εισαγωγή στο MATLAB

Τι είναι ο Προγραμματισμός;

- Διαδικασία επίλυσης προβλημάτων μέσω ακολουθίας εντολών.
- Ο υπολογιστής εκτελεί βήμα-βήμα οδηγίες που έχουμε καθορίσει.
- Βασίζεται στη **λογική**, τη **δομή** και τη **συστηματική σκέψη**.

Τι είναι το MATLAB;

- Εργαλείο αριθμητικών υπολογισμών και προγραμματισμού.
- Αναπτύχθηκε για **μαθηματική ανάλυση, προσομοίωση, επεξεργασία δεδομένων και γραφικές παραστάσεις.**
- Χρησιμοποιείται ευρέως σε **μηχανικούς, φυσικούς, οικονομολόγους και επιστήμονες δεδομένων.**

Το MATLAB ως Υπολογιστής Χειρός

Εκτελεί απλές αριθμητικές πράξεις:

```
>> 3 + 5
```

```
>> sqrt(16)
```

```
>> sin(pi/4)
```

Διανύσματα και Πίνακες

Όλα τα δεδομένα στο MATLAB είναι πίνακες

Διανύσματα = 1D πίνακες, Πίνακες = 2D.

- $A = [1 \ 2 \ 3; 4 \ 5 \ 6];$
 - Ορίζει έναν πίνακα 2x3. Στην πρώτη γραμμή έχει τα στοιχεία 1, 2 και 3 και στην δεύτερη τα στοιχεία 4, 5 και 6
- $v = [10 \ 20 \ 30];$
 - Ορίζει ένα διάνυσμα v μήκους 3 με τα στοιχεία 10, 20 και 30

Αριθμητικοί τελεστές

Τελεστής	Περιγραφή / Λειτουργία	Παράδειγμα	Αποτέλεσμα
+	Πρόσθεση	$3 + 2$	5
-	Αφαίρεση	$7 - 4$	3
*	Πολλαπλασιασμός	$5 * 2$	10
/	Διαίρεση (δεξιά)	$8 / 2$	4
\	Διαίρεση (αριστερά)	$8 \setminus 2$	0.25
^	Ύψωση σε δύναμη	$2 \wedge 3$	8
.*	Πολλαπλασιασμός στοιχείο προς στοιχείο	$[1 \ 2 \ 3] .* [4 \ 5 \ 6]$	$[4 \ 10 \ 18]$
./	Διαίρεση στοιχείο προς στοιχείο	$[6 \ 9] ./ [3 \ 3]$	$[2 \ 3]$
.^	Ύψωση στοιχείο προς στοιχείο	$[2 \ 3] .\wedge 2$	$[4 \ 9]$

Τελεστές σύγκρισης/σχέσης

Τελεστής	Περιγραφή / Λειτουργία	Παράδειγμα	Αποτέλεσμα
==	Ίσο	5 == 5	1 (true)
~=	Όχι ίσο	3 ~= 4	1 (true)
<	Μικρότερο	2 < 5	1 (true)
>	Μεγαλύτερο	7 > 10	0 (false)
<=	Μικρότερο ή ίσο	4 <= 4	1 (true)
>=	Μεγαλύτερο ή ίσο	6 >= 2	1 (true)

Λογικοί τελεστές

Τελεστής	Περιγραφή / Λειτουργία	Παράδειγμα	Αποτέλεσμα
&	Λογικό ΚΑΙ (AND)	$(a > 0) \ \& \ (b < 5)$	1 αν και οι δύο αληθείς
,	,	Λογικό Ή (OR)	$\ (x == 2)$
~	Λογική Άρνηση (NOT)	$\sim (x > 0)$	1 αν ψευδής

Διάφοροι τελεστές

Τελεστής	Περιγραφή / Λειτουργία	Παράδειγμα	Αποτέλεσμα
:	Δημιουργία ακολουθίας	1:2:9	[1 3 5 7 9]
[]	Δημιουργία πίνακα	[1 2 3; 4 5 6]	2×3 πίνακας
,	Διαχωρισμός στηλών	[1,2,3]	[1 2 3]
;	Διαχωρισμός γραμμών	[1 2; 3 4]	2×2 πίνακας
'	Αντιστροφή/Μεταφορά πίνακα	[1 2 3]'	Στήλη πίνακας
=	Ανάθεση τιμής	x = 10	Μεταβλητή x=10
%	Σχόλιο	% αυτό είναι σχόλιο	—

Βασικές συναρτήσεις

Μαθηματικές Συναρτήσεις

Συνάρτηση	Περιγραφή	Παράδειγμα	Αποτέλεσμα
<code>abs(x)</code>	Απόλυτη τιμή	<code>abs(-5)</code>	5
<code>sqrt(x)</code>	Τετραγωνική ρίζα	<code>sqrt(16)</code>	4
<code>exp(x)</code>	Εκθετική συνάρτηση (e^x)	<code>exp(1)</code>	2.7183
<code>log(x)</code>	Φυσικός λογάριθμος	<code>log(10)</code>	2.3026
<code>log10(x)</code>	Λογάριθμος βάσης 10	<code>log10(100)</code>	2
<code>sin(x)</code>	Ημίτονο (x σε rad)	<code>sin(pi/2)</code>	1
<code>cos(x)</code>	Συνημίτονο	<code>cos(0)</code>	1
<code>tan(x)</code>	Εφαπτομένη	<code>tan(pi/4)</code>	1
<code>round(x)</code>	Στρογγυλοποίηση	<code>round(2.6)</code>	3
<code>floor(x)</code>	Προς τα κάτω	<code>floor(2.9)</code>	2
<code>ceil(x)</code>	Προς τα πάνω	<code>ceil(2.1)</code>	3
<code>mod(a,b)</code>	Υπόλοιπο διαίρεσης	<code>mod(10,3)</code>	1

Στατιστικές & Αναλυτικές Συναρτήσεις

Συνάρτηση	Περιγραφή	Παράδειγμα	Αποτέλεσμα
<code>sum(x)</code>	Άθροισμα στοιχείων	<code>sum([1 2 3])</code>	6
<code>mean(x)</code>	Μέσος όρος	<code>mean([2 4 6])</code>	4
<code>median(x)</code>	Διάμεσος τιμή	<code>median([1 2 9])</code>	2
<code>std(x)</code>	Τυπική απόκλιση	<code>std([1 2 3])</code>	1
<code>max(x)</code>	Μέγιστη τιμή	<code>max([2 5 1])</code>	5
<code>min(x)</code>	Ελάχιστη τιμή	<code>min([2 5 1])</code>	1
<code>var(x)</code>	Διακύμανση	<code>var([1 2 3])</code>	1
<code>sort(x)</code>	Ταξινόμηση αύξουσα	<code>sort([5 2 9])</code>	[2 5 9]
<code>diff(x)</code>	Διαφορά διαδοχικών στοιχείων	<code>diff([3 7 10])</code>	[4 3]

Λογικές & Ελέγχου

Συνάρτηση	Περιγραφή	Παράδειγμα	Αποτέλεσμα
<code>isempty(A)</code>	Ελέγχει αν ο πίνακας είναι κενός	<code>isempty([])</code>	1 (true)
<code>isnan(A)</code>	Ελέγχει για NaN (Not a Number)	<code>isnan(0/0)</code>	1 (true)
<code>isinf(A)</code>	Ελέγχει για άπειρο	<code>isinf(1/0)</code>	1 (true)
<code>all(x)</code>	True αν όλα τα στοιχεία είναι true	<code>all([1 1 1])</code>	1
<code>any(x)</code>	True αν κάποιο στοιχείο είναι true	<code>any([0 0 1])</code>	1
<code>find(x)</code>	Θέσεις μη μηδενικών στοιχείων	<code>find([0 2 0 4])</code>	[2 4]
<code>logical(x)</code>	Μετατροπή σε λογικό πίνακα	<code>logical([1 0 3])</code>	[1 0 1]

Συναρτήσεις Πινάκων

Συνάρτηση	Περιγραφή	Παράδειγμα	Αποτέλεσμα
<code>size(A)</code>	Διαστάσεις πίνακα	<code>size([1 2 3;4 5 6])</code>	<code>[2 3]</code>
<code>length(A)</code>	Μήκος διανύσματος	<code>length([10 20 30])</code>	<code>3</code>
<code>numel(A)</code>	Πλήθος στοιχείων	<code>numel([1 2;3 4])</code>	<code>4</code>
<code>sum(A)</code>	Άθροισμα γραμμών ή στηλών	<code>sum([1 2;3 4])</code>	<code>[4 6]</code>
<code>det(A)</code>	Ορίζουσα πίνακα	<code>det([1 2;3 4])</code>	<code>-2</code>
<code>inv(A)</code>	Αντίστροφος πίνακας	<code>inv([1 2;3 4])</code>	<code>[-2 1; 1.5 -0.5]</code>
<code>trace(A)</code>	Ίχνος (άθροισμα κύριας διαγωνίου)	<code>trace([2 0;0 5])</code>	<code>7</code>
<code>eig(A)</code>	Ιδιοτιμές	<code>eig([2 1;1 2])</code>	<code>[3 1]</code>
<code>A'</code>	Μεταφορά πίνακα	<code>[1 2 3]'</code>	<code>[1;2;3]</code>

Είσοδος, Έξοδος & Βοήθεια

Συνάρτηση	Περιγραφή	Παράδειγμα	Αποτέλεσμα
<code>disp(x)</code>	Εμφάνιση τιμής ή κειμένου	<code>disp('Hello MATLAB')</code>	Hello MATLAB
<code>fprintf()</code>	Μορφοποιημένη εκτύπωση	<code>fprintf('x=%.2f\n', 3.1416)</code>	x=3.14
<code>input()</code>	Είσοδος από τον χρήστη	<code>x = input('Δώσε αριθμό: ')</code>	—
<code>help function</code>	Προβολή σύντομης βοήθειας	<code>help plot</code>	Εμφανίζει περιγραφή
<code>doc function</code>	Αναλυτική τεκμηρίωση	<code>doc sin</code>	Ανοίγει σελίδα βοήθειας
<code>clc</code>	Καθαρίζει την οθόνη εντολών	<code>clc</code>	—
<code>clear</code>	Διαγράφει μεταβλητές	<code>clear all</code>	—
<code>who / whos</code>	Λίστα μεταβλητών στο Workspace	<code>whos</code>	Εμφανίζει πίνακα πληροφοριών

If-elseif-else

Η δομή if-else χρησιμοποιείται όταν θέλουμε το πρόγραμμα να **παίρνει αποφάσεις** ανάλογα με δεδομένες συνθήκες.

Ελέγχει μία ή περισσότερες λογικές εκφράσεις και εκτελεί διαφορετικά τμήματα κώδικα, ανάλογα με το αν αυτές είναι αληθείς (true) ή ψευδείς (false).

If-elseif-else

- Στο παράδειγμα, το πρόγραμμα “ελέγχει” τη μεταβλητή soilMoist (ποσοστό υγρασίας εδάφους).
- Αν η τιμή είναι μικρότερη από 20%, θεωρείται ότι το έδαφος είναι **ξηρό** και το σύστημα ενεργοποιεί την άρδευση.
- Αν είναι μεταξύ 20% και 35%, το έδαφος έχει **ιδανική υγρασία**, οπότε δεν απαιτείται ενέργεια.
- Αν η τιμή είναι πάνω από 35%, τότε υπάρχει **υπερβολική υγρασία**, και το σύστημα σταματά το πότισμα για να αποφευχθεί ασφυξία ριζών ή μυκητολογικά προβλήματα.

```
soilMoist = 28;  
  
if soilMoist < 20  
    disp('Το έδαφος είναι ξηρό - απαιτείται άρδευση.')elseif soilMoist >= 20 && soilMoist <= 35  
    disp('Η υγρασία είναι επαρκής.')else  
    disp('Το έδαφος είναι πολύ υγρό - σταμάτα την άρδευση.')end
```

If-elseif-else

- Στο παράδειγμα, το πρόγραμμα “ελέγχει” τη μεταβλητή soilMoist (ποσοστό υγρασίας εδάφους).
- Αν η τιμή είναι μικρότερη από 20%, θεωρείται ότι το έδαφος είναι **ξηρό** και το σύστημα ενεργοποιεί την άρδευση.
- Αν είναι μεταξύ 20% και 35%, το έδαφος έχει **ιδανική υγρασία**, οπότε δεν απαιτείται ενέργεια.
- Αν η τιμή είναι πάνω από 35%, τότε υπάρχει **υπερβολική υγρασία**, και το σύστημα σταματά το πότισμα για να αποφευχθεί ασφυξία ριζών ή μυκητολογικά προβλήματα.

```
soilMoist = 28;

if soilMoist < 20
    disp('Το έδαφος είναι ξηρό - απαιτείται άρδευση.')
elseif soilMoist >= 20 && soilMoist <= 35
    disp('Η υγρασία είναι επαρκής.')
else
    disp('Το έδαφος είναι πολύ υγρό - σταμάτα την άρδευση.')
end
```

If-elseif-else

Με αυτόν τον τρόπο το MATLAB χρησιμοποιείται ως **εργαλείο αυτοματοποίησης και λήψης αποφάσεων** στη γεωργία ακριβείας.

1. Έλεγχος συνθηκών άρδευσης βάσει αισθητήρων.
2. Ενεργοποίηση ανεμιστήρων σε θερμοκήπιο όταν υπερβεί η θερμοκρασία ένα όριο.
3. Έλεγχος του pH νερού πριν τη χρήση σε υδροπονικά συστήματα.
4. Απόφαση εφαρμογής λίπανσης αν η αγωγιμότητα του εδάφους είναι χαμηλή.

For loop

Η εντολή for χρησιμοποιείται για να **επαναλαμβάνει** μια ομάδα εντολών **σταθερό αριθμό φορές**.

Είναι ιδιαίτερα χρήσιμη για επεξεργασία πινάκων, ανάλυση σειρών μετρήσεων και προσομοιώσεις πειραμάτων.

For loop

Η μεταβλητή `rain` περιέχει δεδομένα ημερήσιας βροχόπτωσης (σε mm).

Ο βρόχος `for` ξεκινά από το πρώτο στοιχείο του πίνακα και προχωρά ως το τελευταίο.

Σε κάθε επανάληψη, προσθέτει την ημερήσια τιμή στο `totalRain`. Όταν ο βρόχος ολοκληρωθεί, έχουμε τη **συνολική βροχόπτωση** για το δεκαήμερο.

Η λογική αυτή μπορεί να εφαρμοστεί σε κάθε μορφή **επαναληπτικής επεξεργασίας δεδομένων** — από τη συλλογή αισθητηριακών μετρήσεων μέχρι τον υπολογισμό αποδόσεων ανά ημέρα.

```
rain = [0 2 0 5 8 0 0 10 3 0];
totalRain = 0;

for i = 1:length(rain)
    totalRain = totalRain + rain(i);
end

disp(['Συνολική βροχόπτωση: ', num2str(totalRain), ' mm'])
```

For loop

Use Cases:

- Υπολογισμός μέσης θερμοκρασίας εβδομάδας ή μήνα.
- Άθροιση ημερήσιων μετρήσεων νερού, pH, ή EC.
- Προσομοίωση ανάπτυξης φυτού (ύψος, βιομάζα ανά ημέρα).
- Επαναληπτικός έλεγχος μετρήσεων αισθητήρων πεδίου.
- Επεξεργασία δεδομένων από γεωργικούς σταθμούς ή drones.

While loop

Η εντολή `while` εκτελείται όσο μια συνθήκη παραμένει αληθής.

Χρησιμοποιείται όταν δεν γνωρίζουμε εκ των προτέρων πόσες φορές θα χρειαστεί να επαναληφθεί ο υπολογισμός

- δηλαδή όταν η επανάληψη εξαρτάται από τα ίδια τα δεδομένα.

While loop

Η μεταβλητή `temp` αυξάνεται σταδιακά μέχρι να φτάσει τους 35°C.

Το πρόγραμμα συνεχίζει να τυπώνει την τρέχουσα θερμοκρασία **όσο** η συνθήκη `temp < 35` είναι αληθής.

Μόλις φτάσει στο όριο, η συνθήκη γίνεται ψευδής και ο βρόχος σταματά.

Αυτός ο τύπος βρόχου είναι ιδανικός για **συστήματα πραγματικού χρόνου** όπου χρειάζεται συνεχής παρακολούθηση, όπως: θερμοκρασία, υγρασία, ή στάθμη νερού.

```
temp = 28;

while temp < 35
    disp(['Τρέχουσα θερμοκρασία: ', num2str(temp), '°C'])
    temp = temp + 1;
end

disp('Όριο θερμοκρασίας επιτεύχθηκε - ενεργοποίηση εξαερισμού!')
```

Συνδυασμός If και loops

Ο πίνακας `soilMoist` περιέχει μετρήσεις υγρασίας από έξι διαφορετικούς αισθητήρες τοποθετημένους σε διάφορα σημεία του αγρού.

Ο βρόχος `for` επιτρέπει στο MATLAB να **περνάει από κάθε αισθητήρα** και να ελέγχει την τιμή του.

Η δομή `if` αποφασίζει τι πρέπει να γίνει:

- Αν η τιμή είναι χαμηλή → ενεργοποιείται η άρδευση στο σημείο.
- Αν είναι υψηλή → αποφεύγεται υπερπότισμα.
- Αν είναι εντός ορίων → καμία ενέργεια.

Με αυτό τον τρόπο, το MATLAB μπορεί να λειτουργήσει ως **έξυπνος γεωργικός ελεγκτής**, που προσαρμόζει αυτόματα την άρδευση με βάση πραγματικά δεδομένα.

```
soilMoist = [15 28 22 40 19 25];

for i = 1:length(soilMoist)
    if soilMoist(i) < 20
        disp(['Σημείο ', num2str(i), ': Πότισε'])
    elseif soilMoist(i) > 35
        disp(['Σημείο ', num2str(i), ': Πολύ υγρό'])
    else
        disp(['Σημείο ', num2str(i), ': OK'])
    end
end
```