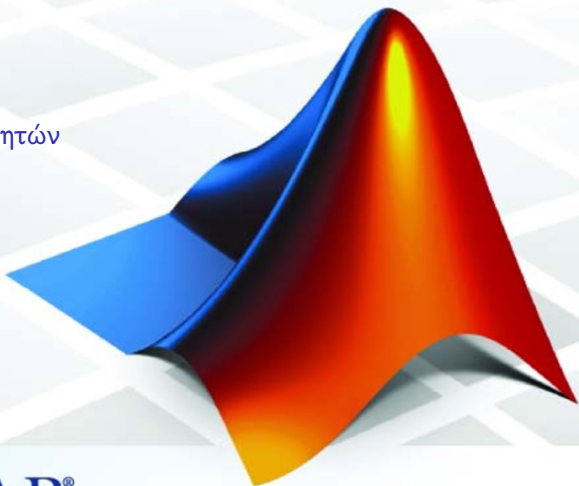


# Προγραμματισμός & Εφαρμογές Η/Υ

Λογικές συναρτήσεις  
Έλεγχοι  
Αναπαράσταση μεταβλητών

Π. Οικονόμου  
Τμήμα Πολιτικών Μηχανικών  
2023–2024



**MATLAB**<sup>®</sup>

# Σχέδιο Σημειώσεων

- Λογικές συναρτήσεις
- Έλεγχοι
- Αναπαράσταση μεταβλητών

# Λογικές προτάσεις και τελεστές

Έστω

`x=4; y=5; z=0:5;`

- **μεγαλύτερο** ('>')

```
>> x>y
ans =
    0
```

- **μικρότερο** ('<')

```
>> x<y
ans =
    1
```

- **μικρότερο ή ίσο** ('<=')

```
>> z<=x
ans =
    1    1    1    1    1    0
```

- **μεγαλύτερο ή ίσο** ('>=')

```
>> z>=y
ans =
    0    0    0    0    0    1
```

↪ Οι λογικές προτάσεις επιστρέφουν το **μηδέν** αν είναι ψευδείς και το **ένα** αν είναι αληθείς.

# Λογικές προτάσεις και τελεστές

- **ίσο** ('==')

```
>> x==y
```

```
ans =
```

```
0
```

```
>> x==z
```

```
ans =
```

```
0    0    0    0    1    0
```

- **όχι ίσο** ('~=')

```
>> x~=x
```

```
ans =
```

```
0
```

```
>> x~=z
```

```
ans =
```

```
1    1    1    1    0    1
```

# Λογικές προτάσεις και τελεστές

- και ('&&')
- ή ('||')
- άρνηση ('~')
- \* αποκλειστική διάζευξη ('xor')

```
>> x<y && 2*x>y
ans =
     1
>> x<y && y<3
ans =
     0
```

```
>> x<y || y<3
ans =
     1
>> x<y || ~(y<3)
ans =
     1
```

```
>> xor(x<y,2*x>y)
ans =
     0
>> x<y && ~(2*x>y) || ~(x<y) && 2*x>y
ans =
     0
```

## άρνηση ('~')

```
>> ~0
```

```
ans =  
    1
```

```
>> ~1
```

```
ans =  
    0
```

```
>> ~[1,2,3,4,0,1,0]
```

```
ans =  
    0    0    0    0    1    0    1
```

```
>> ~(~[1,2,3,4,0,1,0])
```

```
ans =  
    1    1    1    1    0    1    0
```

# Προτεραιότητα πράξεων

- 1 Παρενθέσεις
- 2 δυνάμεις (από αριστερά προς στα δεξιά)
- 3 άρνηση ('~')
- 4 πολλαπλασιασμοί και διαιρέσεις (από αριστερά προς στα δεξιά)
- 5 προσθέσεις και αφαιρέσεις από αριστερά προς στα δεξιά)
- 6 (':')
- 7 λογικές προτάσεις ('>', '<', '>=', '<=', '==', '=')
- 8 και ('&&')
- 9 ή ('||')

# Λογικές συναρτήσεις – Συναρτήσεις αλήθειας

- isequal
- logical
- any
- all
- isempty
- isinf
- isnan
- ...



# Παραδείγματα

```
>> z=0:5; w=[0,1,22,-3,0,1]; q=[0,1,2,3,4,5]
>> zero_mat=zeros(1,3); nonzero_mat=[2,-2,3]; one_mat=ones(1,3);

>> isequal(z,q)
ans =
     1
>> ~isequal(z,q)
ans =
     0
>> logical(w)
ans =
     0     1     1     1     0     1
>> [any(z),any(w),any(zero_mat),any(nonzero_mat),any(one_mat)]
ans =
     1     1     0     1     1
>> [all(z),all(w),all(zero_mat),all(nonzero_mat),all(one_mat)]
ans =
     0     0     0     1     1
```

Νέες εντολές: `zeros(n, k)`, `ones(n, k)`

# Παραδείγματα

```
>> a=[]; b=[Inf,-3,0,NaN];
```

```
>> isempty(a)
```

```
ans =
```

```
1
```

```
>> isinf(b)
```

```
ans =
```

```
1    0    0    0
```

```
>> isnan(b)
```

```
ans =
```

```
0    0    0    1
```

# Έλεγχοι

Η ανάπτυξη πιο σύνθετων προγραμμάτων απαιτεί συχνά τη χρήση λογικών προτάσεων και συναρτήσεων για την πραγματοποίηση διάφορων ελέγχων και την επιλογή της ομάδας εκείνων των εντολών που θα εκτελεστούν.

Σημαντικά εργαλεία για την πραγματοποίηση πολύπλοκων ελέγχων είναι οι εντολές

- if
- switch

## Έλεγχοι

if

## Συντακτικό

```
if συνθήκη A
    εντολές A

elseif συνθήκη B % προαιρετικά
    εντολές B

elseif συνθήκη Γ % προαιρετικά
    εντολές Γ

...
else % προαιρετικά
    εντολές

end
```

## Έλεγχοι

## if – Παράδειγμα

```
a = 2;  
b = 3;  
if a<b  
    j = -1;  
end
```

```
a = 4;  
b = 3;  
if a<b  
    j = -1;  
elseif a>b  
    j = 2;  
end
```

```
a = 4;  
b = 4;  
if a<b  
    j = -1;  
elseif a>b  
    j = 2;  
else  
    j = 3  
end
```

## Έλεγχοι

## if – Παράδειγμα

```
AM=input('Dwse AM foithth: ');
grade_average=0; final_grade=0; pass=0;

grade1=input('vathmos stin 1 proodo: ');
grade2=input('vathmos stin 2 proodo: ');
grade_exams=input('vathmos stis exetaseis: ');

if grade1>0 && grade2>0
    final_grade=(grade1+grade2)/2;
    if final_grade<5
        final_grade=grade_exams;
    end
elseif xor(grade1>0,grade2>0)
    if max(grade1,grade2)>grade_exams
        final_grade=0.3*max(grade1,grade2)+0.7*grade_exams;
    else
        final_grade=grade_exams;
    end
else
    final_grade=grade_exams;
end

if final_grade>=5
    pass = 1;
end

[AM, final_grade, pass]
```

# Έλεγχοι

## if – Παράδειγμα

```
mynumber = input('type a real number: ');  
  
if mynumber<0  
    a=-1; disp('a is set equal to negative one');  
elseif mynumber>0  
    a=1;   disp('a is set equal to positive one');  
else  
    a=0;   disp('a is set equal to zero');  
end
```

# Παράδειγμα

## if – Παράδειγμα

```
x = input('Dwse enan arithmo: ');
minVal = 2;
maxVal = 6;

if (x >= minVal) && (x <= maxVal)
    disp('Value within specified range.')
elseif (x > maxVal)
    disp('Value exceeds maximum value.')
else
    disp('Value is below minimum value.')
end
```



# Έλεγχοι

switch - case

## ΣΥΝΤΑΚΤΙΚΟ

```
switch switch_expr
case case_expr,
    statement, ..., statement
case {case_expr1, case_expr2, case_expr3,...}
    statement, ..., statement
...
otherwise,
    statement, ..., statement
end
```

# Έλεγχοι

## switch - case – Παράδειγμα

```
mynumber = input('select one number between -1, 0 and 1: ');

switch mynumber
    case -1
        a=-1; disp('a is set equal to negative one');
    case 0
        a=0;   disp('a is set equal to zero');
    case 1
        a=1;   disp('a is set equal to positive one');
    otherwise
        disp('please select one number between -1, 0 and 1');
end
```

# Μορφοποίηση

## Αναπαράσταση αριθμών

`format long`

`format short`

`format bank`

### Ύσκιση

- Ο αριθμός  $e$  είναι ένας άρρητος αριθμός που ισούται προσεγγιστικά με το 2.718. Ποια είναι η τιμή του  $e$  που τυπώνει το MATLAB; Ελέγξτε αν η απάντηση του MATLAB για την τιμή του  $e$  ισούται πράγματι με την τιμή του  $e$  που χρησιμοποιείται εσωτερικά για την τέλεση διαφόρων πράξεων. Προσπαθήστε να δικαιολογήσετε το αποτέλεσμα του MATLAB.
- Εκτελέστε την εντολή `format long` και τυπώστε την τιμή του  $e$ .
- Ελέγξτε αν η τωρινή απάντηση του MATLAB για την τιμή του  $e$  ισούται με το  $e$ . Ποιο είναι το συμπέρασμά σας για την πραγματική τιμή του  $e$ ;
- Εκτελέστε την εντολή `format short`. Πώς τυπώνει τώρα το MATLAB την τιμή του  $e$ ;

---

## Αναπαράσταση αποτελεσμάτων

`format compact`

`format loose`

# Παράδειγμα - άσκηση - eps, realmax, realmin

## άσκηση

Θέστε  $x = 1$ ,  $y1 = 1 + eps$  και  $y2 = 1 + 0.5 * eps$ . Παρατηρήστε πώς τυπώνει το MATLAB τις μεταβλητές  $y1$  και  $y2$ . Ελέγξτε αν οι μεταβλητές  $y1$  και  $y2$  είναι ίσες με  $x$ . Εκτελέστε τις εντολές `eps(1)` και `eps(0)`. Παρατηρήστε ότι η διακριτική ικανότητα του MATLAB δεν είναι ίδια  $\forall x \in \mathbb{R}$ . Εκτελέστε τις εντολές:

```
>>realmax
>>eps(realmax)
>>realmax+10^291
>>realmax+10^292
```

Παρατηρήστε ποιους αριθμούς το MATLAB τους αναπαριστά ως άπειρο. Εκτελέστε τις εντολές

```
>>realmin
```

Οποιοσδήποτε θετικός αριθμός μικρότερος από το `realmin` θεωρείται ίσος με το μηδέν. Παρόλα αυτά, πολλά προγράμματα, όπως και το MATLAB, επιτρέπουν τη χρήση denormal or subnormal floating-point αριθμών (όπως αποκαλούνται) στο διάστημα (`eps(0) * realmin, realmin`).

```
>>0.1*realmin
>>eps(0)*realmin
```

Περισσότερα για αυτές τις σταθερές μπορείτε να βρείτε στο  
Cleve Moler, Floating points  
[on-line pdf file](#)