



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΧΗΜΙΚΩΝ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΙΚΟ ΚΕΝΤΡΟ

Προγραμματισμός και Χρήση Ηλεκτρονικών Υπολογιστών - Βασικά Εργαλεία Λογισμικού

Μάθημα 3ο

Λογικοί τελεστές, προτεραιότητα λογικών τελεστών
Μαθηματικές συναρτήσεις
Εισαγωγή δεδομένων (εντολή input)
Εξαγωγή δεδομένων (disp/fprintf)

Τελεστές σύγκρισης (relational operators)

Χρησιμοποιούνται για να συγκρίνουμε δυο τιμές

Το αποτέλεσμα της σύγκρισης μπορεί να είναι true (1) ή false (0)

Τελεστής	Επεξήγηση
==	Ίσο
>	Μεγαλύτερο
<	Μικρότερο
>=	Μεγαλύτερο ή ίσο
<=	Μικρότερο ή ίσο
~=	Διάφορο

```
>> 5==6
```

```
ans =
```

```
0 ← false
```

```
>> 5<6
```

```
ans =
```

```
1 ← true
```

Ελέγξτε τον τύπο (class) της μεταβλητής ans με την εντολή whos

Διαφορά μεταξύ του τελεστή `=` και του τελεστή `==`

Ο τελεστής `=` είναι τελεστής εκχώρησης (assignment)

Αλλάζει την τιμή της μεταβλητής που βρίσκεται στο αριστερό άκρο του με αυτή που βρίσκεται στο δεξιό.

```
>> x=5+2*3
x =
    11
```

Ο τελεστής `==` είναι τελεστής σύγκρισης (relational)

Συγκρίνει ως προς την ισότητα τις τιμές που βρίσκονται στα άκρα του.
Το αποτέλεσμα της σύγκρισης μπορεί να είναι true (1) ή false (0)

```
>> x=1
x =
    1

>> x==5+2*3
ans =
    0
```

Λογικοί τελεστές (logical operators)

Τελεστής	Επεξήγηση
&& ή &	AND
ή	OR
~	NOT

Τελεστής	Προτεραιότητα
~	Υψηλή
&& ή &	Μεσαία
ή	Χαμηλή

```
>> 5<6 && 7<2  
ans =  
0
```

```
>> 5<6 || 7<2  
ans =  
1
```

Boolean Algebra (AND, OR, NOT)

true (1) false (0)

AND	0	1
0	0	0
1	0	1

OR	0	1
0	0	1
1	1	1

NOT	
0	1
1	0

Προτεραιότητα τελεστών

Τελεστής	Προτεραιότητα
Λογικό NOT (~)	Υψηλή ↓ Χαμηλή
Αριθμητικός (^ * / + -)	
Σύγκρισης (== > < <= >= ~=)	
Λογικός (&&)	

```
>> 5>10-9
```

```
ans =
```

```
1
```

Χρήσιμες μαθηματικές συναρτήσεις

Μαθηματική έκφραση	Εντολή
$\sin(\alpha)$	<code>sin(a)</code>
$\cos(\alpha)$	<code>cos(a)</code>
$\tan(\alpha)$	<code>tan(a)</code>
$\sqrt{\alpha}$	<code>sqrt(a)</code>
$ \alpha $	<code>abs(a)</code>
e^α	<code>exp(a)</code>
$\ln(\alpha)$	<code>log(a)</code>
$\log_{10}(\alpha)$	<code>log10(a)</code>
$\sin^{-1}(\alpha)$	<code>asin(a)</code>
$\cos^{-1}(\alpha)$	<code>acos(a)</code>
$\tan^{-1}(\alpha)$	<code>atan(a)</code>
Στρογγυλοποίηση στον πλησιέστερο ακέραιο	<code>round(a)</code>
Στρογγυλοποίηση στον μικρότερο ακέραιο	<code>floor(a)</code>
Στρογγυλοποίηση στον μεγαλύτερο ακέραιο	<code>ceil(a)</code>

Η μεταβλητή a ονομάζεται **όρισμα** (argument) της συνάρτησης.

ΠΡΟΣΟΧΗ στην τιμή του ορίσματος

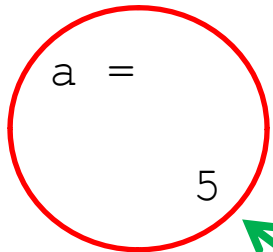
ΤΙΠ: εντολή `help`

Άσκηση: Να βρεθεί το ημίτονο των 30°

Η συνάρτηση `input`

Με τη συνάρτηση αυτή προτρέπουμε (prompt) τον χρήστη να πληκτρολογήσει μια τιμή η οποία θα εκχωρηθεί σε αντίστοιχη μεταβλητή. Για παράδειγμα:

```
>> a=input('Give me a value: ')\nGive me a value: 5
```



Είσοδος (input) από το πληκτρολόγιο

Έξοδος (output) στην οθόνη

Η συνάρτηση `disp`

Με τη συνάρτηση αυτή μπορούμε εμφανίσουμε στην οθόνη την τιμή μιας μεταβλητής:

```
>> x=10;  
>> disp(x)  
10
```

ή να εμφανίσουμε ένα μήνυμα στον χρήστη:

```
>> disp('Hello World')  
Hello World
```

Η συνάρτηση `fprintf`

Η συνάρτηση αυτή μας δίνει πλήρη έλεγχο στον τρόπο με τον οποίο θα εμφανίσουμε στην οθόνη μηνύματα ή/και μεταβλητές. Η σύνταξη της έχει την μορφή:

```
fprintf('format', x, y, . . .)
```

όπου `format` είναι η μορφή με την οποία θέλουμε να εμφανίσουμε τις μεταβλητές `x, y, ...`

<i>format</i>	Επεξήγηση
<code>%d</code>	Ακέραια μορφή
<code>%e</code>	Εκθετική μορφή με μικρό e
<code>%E</code>	Εκθετική μορφή με κεφαλαίο E
<code>%f</code>	Δεκαδική μορφή
<code>%g</code>	Σημαντικά ψηφία (Significant digits)
<code>\n</code>	Νέα γραμμή
<code>\t</code>	Tab

Παραδείγματα με `fprintf`

```
>> fprintf('The value of pi is: %e \n',pi)
The value of pi is: 3.141593e+00
```

```
>> fprintf('The value of pi is: %f \n',pi)
The value of pi is: 3.141593
```