

Εντολές εισόδου:

- `input`: Με την εντολή `input` μπορούμε να αναθέσουμε μία τιμή σε μία μεταβλητή κατά τη χρήση του προγράμματος. Στην `input` μπορεί να προστεθεί κείμενο, το οποίο συνήθως είναι μία περιγραφή του τι ζητείται, προκειμένου το πρόγραμμα να δουλέψει.

```
x=input('Give me a number: ')
```

Η εντολή αυτή θα εμφανίσει:

```
Give me a number:
```

Εμείς, αφού εμφανιστεί το κείμενο, θα μπορούμε να γράψουμε έναν αριθμό:

```
Give me a number: 21
```

Πατώντας το `enter`, στο `x` ανατίθεται η τιμή 21.

Βασικοί λογικοί τελεστές:

Οι διάφοροι βρόγχοι επανάληψης ή ελέγχου έχουν τη βασική δομή:

```
Εάν [μία λογική συνθήκη] είναι αληθής:
```

```
    [Κάνε αυτές τις εντολές]
```

```
Τέλος εντολών.
```

ή:

```
Όσο [μία λογική συνθήκη] είναι αληθής:
```

```
    [Κάνε αυτές τις εντολές]
```

```
Τέλος εντολών.
```

Είναι λοιπόν αναγκαίο να μπορούμε να εκφράσουμε στη γλώσσα του υπολογιστή λογικές συνθήκες. Στη συνέχεια θα αναφερθούμε σε λογικούς τελεστές με τους οποίους μπορούμε να δομούμε λογικές συνθήκες.

- **Είναι ίσο:** Χρησιμοποιείται το `==`

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

```
2==1
ans = 0
```

(Τις γραμμές 1 και 3 γράφουμε εμείς, ενώ τις 2 και 4 εμφανίζει το Matlab).

Προσέξτε εδώ ότι **δεν** χρησιμοποιείται το =, καθώς έχει δεσμευτεί για την εκχώρηση τιμής. Χρησιμοποιείται το ==. Τον συγκεκριμένο τελεστή μπορούμε να τον φανταζόμαστε ως ερώτηση προς τον υπολογιστή: Ρωτάμε αν δύο αντικείμενα είναι ίδια, και ο υπολογιστής απαντάει 1 εάν είναι ίδια και 0 εάν δεν είναι. Το 1 μπορεί να αντικατασταθεί από το **true** και το 0 από το **false**.

- **Αρνηση:** Χρησιμοποιείται το ~ (ή !)

```
~(1==1)
ans = 0
~(2==1)
ans = 1
```

(Τις γραμμές 1 και 3 γράφουμε εμείς, ενώ τις 2 και 4 εμφανίζει το Matlab).

Στην ουσία η ~ εναλλάσσει τις λογικές τιμές μίας λογικής συνθήκης. Όταν $P==\text{true}$ (δηλαδή $P==1$) έχουμε $\sim P==\text{false}$ (δηλαδή $\sim P==0$), ενώ όταν $P==\text{false}$ (δηλαδή $P==0$) έχουμε $\sim P==\text{true}$ (δηλαδή $\sim P==1$).

- **Δεν είναι ίσο:** Χρησιμοποιείται το ~= (ή !=)

```
1~=1
ans = 0
2~=1
ans = 1
```

(Τις γραμμές 1 και 3 γράφουμε εμείς, ενώ τις 2 και 4 εμφανίζει το Matlab).

Παρατηρήστε ότι $\sim(1==1) == (1\sim=1)$ και $\sim(2==1) == (2\sim=1)$.

- **Ή το ένα ή το άλλο ή και τα δύο:** Χρησιμοποιείται το ||

```
(1==2) || (12==4)
ans = 0
(1==1) || (12==4)
ans = 1
(1==2) || (12==12)
ans = 1
```

```
(1==1) || (12==12)
ans = 1
```

(Τις γραμμές 1, 3, 5 και 7 γράφουμε εμείς, ενώ τις 2, 4, 6 και 8 εμφανίζει το Matlab).

Εάν P και Q είναι δύο λογικές συνθήκες, η $P || Q$ αληθεύει εάν και μόνο αν κάποια από τις P, Q αληθεύει.

- **Και:** Χρησιμοποιείται το `&&`

```
(1==2) && (12=4)
ans = 0
(1==1) && (12=4)
ans = 0
(1==2) && (12==12)
ans = 0
(1==1) && (12==12)
ans = 1
```

(Τις γραμμές 1, 3, 5 και 7 γράφουμε εμείς, ενώ τις 2, 4, 6 και 8 εμφανίζει το Matlab).

Εάν P και Q είναι δύο λογικές συνθήκες, η $P \&\& Q$ αληθεύει μόνο αν και οι δύο P, Q αληθεύουν.

Επίσης, θα αναφέρουμε χωρίς παραδείγματα τις σχέσεις διάταξης.

- **Είναι μικρότερο:** Χρησιμοποιείται το `<`
- **Είναι μεγαλύτερο:** Χρησιμοποιείται το `>`
- **Είναι μικρότερο ή ίσο:** Χρησιμοποιείται το `<=`
- **Είναι μεγαλύτερο ή ίσο:** Χρησιμοποιείται το `>=`

Βρόγχοι `if`:

- **Βρόγχοι `if`:** Με την εντολή `if` [κάποια λογική συνθήκη] μπορούμε να εκτελέσουμε εντολές μόνο σε συγκεκριμένες περιπτώσεις, όταν η λογική συνθήκη είναι αληθής. Για παράδειγμα, το παρακάτω θα εκτελεστεί **μόνο** όταν `x>=0`:

```
if x>=0
    disp('This displays only when x>=0')
endif
```

και θα εμφανίσει, μόνο όταν `x>=0`:

```
This displays only when x>=0
```

Προσοχή: Για να δουλέψουν τα παραπάνω, θα πρέπει η τιμή του x να έχει κάπως προσδιοριστεί (από εμάς ή από το πρόγραμμα).

- **elseif:** Ενδέχεται σε δύο διαφορετικά ενδεχόμενα να θέλουμε να εκτελέσουμε δύο διαφορετικά είδη εντολών. Αυτό μπορεί να επιτευχθεί μέσω του `if`, με την επιπλέον εντολή `elseif`.

```
if x>=0
    disp('This displays only when x>=0')
elseif x==-1
    disp('This displays only when x==-1')
endif
```

Το παραπάνω πρόγραμμα θα εμφανίσει:

```
This displays only when x>=0
```

μόνο όταν $x \geq 0$. Εάν $x == -1$, θα εμφανιστεί:

```
This displays only when x==-1
```

Για τις υπόλοιπες περιπτώσεις, το πρόγραμμα δεν θα εμφανίσει **τίποτα**.

Προσοχή: Για να δουλέψουν τα παραπάνω, θα πρέπει η τιμή του x να έχει κάπως προσδιοριστεί (από εμάς ή από το πρόγραμμα).

- **else:** Με την επιπλέον εντολή `else` στο `if` μπορούμε να συμπεριλάβουμε όσες περιπτώσεις απέμειναν. Για παράδειγμα:

```
if x>0
    disp('This displays only when x>0')
elseif x<0
    disp('This displays only when x<0')
else
    disp('This displays only when x==0')
endif
```

Ένα άλλο παράδειγμα (χωρίς `elseif`):

```
if x>=0
    disp('This displays only when x>=0')
else
```

```
    disp('This displays only when x<0')  
endif
```

Προσοχή: Για να δουλέψουν τα παραπάνω, θα πρέπει η τιμή του x να έχει κάπως προσδιοριστεί (από εμάς ή από το πρόγραμμα).

Προσοχή: Από την στιγμή που το `if` εκτελεστεί, κανένα `elseif` ή `else` δεν εκτελείται. Αντίστοιχα, αν κάποιο `elseif` εκτελεστεί, κανένα επόμενο `elseif` ή `else` δεν εκτελείται.