

# Έλεγχος Ροής Εκτέλεσης

## Τελεστές Σύγκρισης

|          |                                 |
|----------|---------------------------------|
| $x == y$ | x είναι ίσος με y               |
| $x != y$ | x δεν είναι ίσος με y           |
| $x > y$  | x είναι μεγαλύτερος από y       |
| $x < y$  | x είναι μικρότερος από y        |
| $x >= y$ | x είναι μεγαλύτερος ή ίσος με y |
| $x <= y$ | x είναι μικρότερος ή ίσος με y  |

# Έλεγχος Ροής Εκτέλεσης

## Προτεραιότητα Τελεστών

Οι εκφράσεις μέσα σε παρενθέσεις αποτιμώνται πρώτες. Η προτεραιότητα των τελεστών, από τη μεγαλύτερη στη μικρότερη, είναι η εξής:

Αριθμητικοί τελεστές (με τη γνωστή προτεραιότητα) ,

<=, >=, !=, ==

not

and

or (28.1)



# Έλεγχος Ροής Εκτέλεσης

## Εντολή if

- Χρησιμοποιείται για έλεγχο της ροής εκτέλεσης ενός προγράμματος
- Ελέγχεται μία συνθήκη και ανάλογα με το αποτέλεσμα (Αληθής ή Ψευδής) εκτελείται ή δεν εκτελείται μία ή κάποια άλλη ομάδα (μπλοκ) εντολών. (29.1)

```
if συνθήκη:  
    μπλοκ εντολών όταν η συνθήκη είναι αληθής  
else:  
    μπλοκ εντολών όταν η συνθήκη είναι ψευδής
```

# Έλεγχος Ροής Εκτέλεσης

## Εντολή if

- Μπορούμε να έχουμε και μόνο έναν κλάδο εκτέλεσης (**χωρίς else**) και να εκτελείται ένα μπλοκ εντολών στην περίπτωση που η συνθήκη είναι αληθής. Με άλλα λόγια ο κλάδος **else** είναι προαιρετικός. (30.1)
- **Αλυσιδωτές συνθήκες (elif)**: με πολλούς κλάδους εκτέλεσης (όσους θέλουμε). Σε κάθε εκτέλεση ένας **ΜΟΝΟ** κλάδος εκτελείται. Κάθε συνθήκη ελέγχεται με τη σειρά, αν η πρώτη είναι ψευδής, πάμε στη δεύτερη, κ.ο.κ. Αν κάποια συνθήκη είναι αληθής, τότε ο αντίστοιχος κλάδος (μπλοκ εντολών) εκτελείται και η εντολή if ολοκληρώνεται, **ακόμη** και αν ακολουθούν και άλλες συνθήκες που αληθεύουν. (30.2)
- **Εμφωλευμένες συνθήκες**: μία συνθήκη μέσα σε άλλη συνθήκη. (30.3)



# Έλεγχος Ροής Εκτέλεσης

## Εντολή for:

- Συγκεκριμένες ομάδες (μπλοκ) εντολών επαναλαμβάνονται (εκτελούνται) πολλές φορές
- Μία τέτοια ομάδα εντολών που επαναλαμβάνεται ονομάζεται **βρόγχος (loop)**
- Η επικεφαλίδα ενός βρόγχου **for** ξεκινάει πάντα με τη δεσμευμένη λέξη **for**, έπειτα ακολουθεί μία μεταβλητή, η λέξη κλειδί **in**, συνήθως η συνάρτηση **range** και τελειώνει με άνω κάτω τελεία. (31.1)
- **range(αρχή, τέλος, βήμα)** (31.2, 31.3, 31.4)

# Έλεγχος Ροής Εκτέλεσης

## Άσκηση

- Γράψτε κώδικα ο οποίος να περιέχει έναν βρόγχο for για εκτύπωση των αριθμών από το 1 έως και το 10.
- Γράψτε κώδικα ο οποίος να περιέχει έναν βρόγχο for για αντίστροφη εκτύπωση των προηγούμενων αριθμών (από το 10 έως και το 1).
- Γράψτε κώδικα ο οποίος να διαβάζει την ηλικία σας και στη συνέχεια να τυπώνει τους άρτιους αριθμούς μέχρι να φτάσει την ηλικία.



# Έλεγχος Ροής Εκτέλεσης

## Εντολή `while`:

- Η οποία επαναλαμβανόμενα εκτελεί μία ομάδα (μπλοκ) εντολών, όσο μια συνθήκη (προϋπόθεση) παραμένει αληθής.
- Χρησιμοποιείται για τον λεγόμενο **Αμυντικό Προγραμματισμό**
- Ένας βρόγχος **while** ξεκινάει με μία επικεφαλίδα που περιλαμβάνει τη δεσμευμένη λέξη **while**, μία συνθήκη (οποιαδήποτε Boolean έκφραση) και τελειώνει με μία άνω κάτω τελεία.
- Το «σώμα» του βρόγχου θα πρέπει να αλλάζει τις τιμές μιας ή περισσότερων μεταβλητών, τις οποίες ελέγχει η συνθήκη στην επικεφαλίδα, ώστε η συνθήκη να γίνεται κάποια στιγμή ψευδής και ο βρόγχος να τερματίζεται. (33.1)

# Έλεγχος Ροής Εκτέλεσης

## Εντολή `while`:

- Ο βρόγχος που επαναλαμβάνεται επ' άπειρον, ονομάζεται **ατέρμων βρόγχος (infinite loop)**
- Η μεταβλητή ή οι μεταβλητές που αναφέρονται στη συνθήκη θα πρέπει να έχουν πάρει νωρίτερα στον κώδικά μας αρχικές τιμές με τις λεγόμενες **εντολές αρχικοποίησης**.
- Η εντολή **`break`** χρησιμοποιείται για άμεση έξοδο από οποιοδήποτε σημείο ενός βρόγχου (το αποφεύγουμε) (34.1)