

ΔΟΜΗΜΕΝΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Εντολές Επανάληψης: Η εντολή `while`

Η εντολή `while`

```
while λογική_συνθήκη  
    εντολή1;  
    εντολή2;  
    ...  
    εντολήN;  
end
```

- Η εκτέλεση των *εντολών* επαναλαμβάνεται όσο η *λογική συνθήκη* είναι αληθής.
- Όταν η *λογική συνθήκη* γίνει ψευδής, τότε η επαναληπτική διαδικασία σταματάει και εκτελούνται οι εντολές (αν υπάρχουν) που βρίσκονται αμέσως μετά το `end`.

Παράδειγμα: Καταμέτρηση

- Θέλουμε να τυπώσουμε όλους τους ακέραιους αριθμούς από το 1 έως το 10.
- Λύση:

Με for :

```
for i=1:1:10
    disp(i);
end
```

Με while :

```
i = 1;
while i <= 10
    disp(i);
    i = i + 1;
end
```

Σχέση for και while

```
for var = AT:B:TT
    εντολές
end
```



```
var = AT;
while var <= TT
    εντολές_S
    var = var + B;
end
```

- Χρησιμοποιούμε την for σε περιπτώσεις καταμέτρησης, όπου το πλήθος των επαναλήψεων είναι γνωστό εξαρχής.
- Χρησιμοποιούμε την while σε περιπτώσεις όπου η επανάληψη γίνεται υπό συνθήκη και το πλήθος των επαναλήψεων δεν είναι γνωστό εξαρχής.

Η εντολή `while`

```
while λογική_συνθήκη  
    εντολές;  
end
```

- Η τιμή της συνθήκης [*αληθής* ή *ψευδής*] πρέπει:
 - να μεταβάλλεται με κάποια από τις εντολές που βρίσκονται μέσα στο block της `while`, διαφορετικά η τιμή της λογικής συνθήκης δεν θα αλλάξει.
- Συνήθως η εντολή `while` χρησιμοποιείται σε περιπτώσεις όπου:
 - το πλήθος των επαναλήψεων δεν είναι γνωστό.
 - το πλήθος των επαναλήψεων εξαρτάται από την ικανοποίηση ή όχι μιας συνθήκης.

Η εντολή `while` – Βήματα εκτέλεσης

```
while λογική_συνθήκη  
    εντολές;  
end
```

1. Ελέγχουμε τη `λογική_συνθήκη` :
2. Αν είναι αληθής, τότε εκτελούμε *όλες* τις εντολές της `while` *με τη σειρά* και μετά επιστρέφουμε πάλι στο βήμα 1.
3. Αν είναι ψευδής, τότε εκτελούμε τις εντολές μετά το `end` (αν υπάρχουν).

Παράδειγμα κατανόησης 1

- Τι θα τυπώσει το παρακάτω πρόγραμμα?

```
x = 0;

while x <= 18
    x = x + 3;
end

disp(x);
```

Απάντηση: Θα τυπώσει 21. Η τελευταία επανάληψη θα γίνει όταν το x πάρει την τιμή 18, κατά την οποία η συνθήκη θα είναι αληθής. Στη συνέχεια το x θα πάρει την τιμή 21, και η λογική συνθήκη της `while` θα πάψει να ισχύει, οπότε ο βρόχος τερματίζεται. Μετά το τέλος της επανάληψης η μεταβλητή x θα έχει την τιμή 21, η οποία και θα τυπωθεί με την `disp()`.

Παράδειγμα κατανόησης 2

- Τι θα τυπώσει το παρακάτω πρόγραμμα?

```
x = 6;

while x > 4
    x = x - 1;
    disp(x);
end
```

Απάντηση: Θα τυπώσει 5 και 4. Οι περισσότερες `while` κάνουν κάτι και στη συνέχεια ενημερώνουν τη μεταβλητή επανάληψης, αλλά εδώ γίνεται το ανάποδο: πρώτα μειώνεται η μεταβλητή `x` και μετά τυπώνεται η τιμή της. Την πρώτη φορά, καθώς το 6 είναι μεγαλύτερο από το 4, εκτελείται μια επανάληψη, η οποία εκχωρεί το 5 στο `x` και μετά το τυπώνει. Στη συνέχεια έχουμε άλλη μια επανάληψη (το 5 είναι μεγαλύτερο από το 4), η οποία εκχωρεί την τιμή 4 στο `x` και το τυπώνει. Μετά η συνθήκη δεν ισχύει και ο βρόχος τερματίζεται.

Οι εντολές `break` και `continue`

- Η εντολή `break` μέσα σε `while` προκαλεί τερματισμό της εκτέλεσης της επανάληψης, παρόμοια με τη `for`, και η εκτέλεση του προγράμματος μεταφέρεται στην εντολή που υπάρχει αμέσως το `end` της επανάληψης.
- Παρόμοια η εντολή `continue` προκαλεί διακοπή των εντολών της επανάληψη που εκτελείται τώρα και η επανάληψη συνεχίζεται με την επόμενη επανάληψη που υπαγορεύεται από την επαναληπτική διαδικασία.

Ειδικές Περιπτώσεις

- Περίπτωση 1:
 - *Η συνθήκη είναι ψευδής από την αρχή: Οι εντολές της while δεν θα εκτελεστούν καθόλου.*
 - Παράδειγμα:

```
k = 20;
```

```
while k < 10  
    disp(k);  
    k = k - 1;  
end
```

Ειδικές Περιπτώσεις

- Περίπτωση 2:
 - *Η συνθήκη είναι πάντα αληθής*: οι εντολές της `while` εκτελούνται συνέχεια (ατέρμων βρόχος).
 - Παράδειγμα:

```
k = 5;
```

```
while k < 10  
    disp(k);  
    k = k - 1;  
end
```

Ειδικές Περιπτώσεις

- Περίπτωση 3:

- *Η συνθήκη είναι πάντα αληθής και δεν αλλάζει ποτέ:* οι εντολές της `while` εκτελούνται συνέχεια (ατέρμων βρόγχος).

- Παράδειγμα:

```
g = 1;
```

```
while 1
```

```
    disp(g);
```

```
    g = g + 1;
```

```
end
```

Παράδειγμα: Επανάληψη υπό συνθήκη

- Θέλουμε να φτιάξουμε ένα πρόγραμμα το οποίο θα διαβάζει τιμές από το χρήστη και θα τις προσθέτει, μέχρι να εισάγει κάποια τιμή που σηματοδοτεί το τέλος των δεδομένων (πχ., 0 ή -1).
-

- Μπορούμε να χρησιμοποιήσουμε `for` ?
 - Όχι, καθώς δεν μπορούμε να γνωρίζουμε εκ των προτέρων το πλήθος τιμών που θα εισάγει ο χρήστης.

Παράδειγμα: Επανάληψη υπό συνθήκη

- Θέλουμε να φτιάξουμε ένα πρόγραμμα το οποίο θα διαβάζει τιμές από το χρήστη και θα τις προσθέτει, μέχρι να εισάγει κάποια τιμή που σηματοδοτεί το τέλος των δεδομένων (πχ., 0 ή -1).

- **Λύση 1** (με χρήση λογικής συνθήκης στη `while`):

```
s = 0;
value = input('Δώσε τον πρώτο αριθμό: ');

while value ~= 0
    s = s + value;
    value = input('Δώσε τον επόμενο αριθμό: ');
end

disp(s);
```

Όσο η μεταβλητή `value` είναι διαφορετική από το 0, η επανάληψη θα συνεχίζεται

Παράδειγμα: Επανάληψη υπό συνθήκη

- Θέλουμε να φτιάξουμε ένα πρόγραμμα το οποίο θα διαβάζει τιμές από το χρήστη και θα τις προσθέτει, μέχρι να εισάγει κάποια τιμή που σηματοδοτεί το τέλος των δεδομένων (πχ., 0 ή -1).

- **Λύση 2** (με χρήση εντολής `break`):

```
s = 0;
```

```
while 1  
    value = input('Δώσε αριθμό:');  
    if value == 0  
        break;  
    end  
    s = s + value;  
end
```

Η συνθήκη είναι πάντα αληθής

Όταν τη τιμή της μεταβλητής `value` γίνει 0, τότε θα εκτελεστεί η εντολή `break`, η οποία θα τερματίσει άμεσα την εντολή `while`

```
disp(s);
```

Άσκηση 1: Άθροισμα Αριθμών

- Να γραφεί πρόγραμμα στο οποίο ο χρήστης θα εισάγει αριθμούς μέχρι το άθροισμά τους να ξεπεράσει το 100.
-

- **Λύση 1** (με χρήση λογικής συνθήκης στη `while`):

```
s=0;
```

```
while s <= 100
```

```
    x = input('Enter num: ');
```

```
    s = s + x;
```

```
end
```

```
disp(s);
```

Άσκηση 1: Άθροισμα Αριθμών

- Να γραφεί πρόγραμμα στο οποίο ο χρήστης θα εισάγει αριθμούς μέχρι το άθροισμά τους να ξεπεράσει το 100.
-

- **Λύση 2** (με χρήση εντολής `break`):

```
s=0;

while 1
    x = input('Δώσε αριθμό: ');
    s = s + x;

    if s > 100
        break;
    end
end

disp(s);
```

Άσκηση 2: Προσομοίωση Επιτοκίου

- Ένα αποταμιευμένο κεφάλαιο τοκίζεται με 2% το χρόνο.
- Να γραφεί πρόγραμμα το οποίο θα ζητάει ένα αρχικό κεφάλαιο και θα υπολογίζει πόσα χρόνια χρειάζονται ώστε να διπλασιαστεί.
- Λύση:

```
money = input('Δώσε κεφάλαιο: ');  
moneyTotal = 2 * money;
```

```
years = 0;  
while money < moneyTotal;  
    money = money + (money * 0.02);  
    years = years + 1;  
end
```

```
fprintf('Χρόνια που χρειάζονται: %d\n', years);
```

Άσκηση 3: Προσομοίωση Πειράματος

- Μία μπάλα ρίπτεται από συγκεκριμένο ύψος και αναπηδά κάθε φορά στα $2/3$ του προηγούμενου ύψους, μέχρι το ύψος να φτάσει κάτω από τα 0.1 εκατοστά.
- Να γραφεί πρόγραμμα το οποίο θα ζητάει από το χρήστη να εισάγει το αρχικό ύψος και να τυπώνει για κάθε αναπήδηση το ύψος της μπάλας.
- Λύση:

```
height = input('Υψος: ');  
  
while height > 0.1  
    height = height * (2 / 3);  
    disp(height);  
end
```

Έλεγχος τιμών εισόδου

- Πολλές φορές θέλουμε να ελέγξουμε αν η τιμή που έδωσε ο χρήστης είναι σωστή και αν δεν είναι τότε να ξαναζητήσουμε να μας δώσει
- Θα μπορούσαμε να κάνουμε τον έλεγχο με απλή `if` ?

– Παράδειγμα:

```
age = input('Δώσε την ηλικία σου: ');  
  
if age < 0  
    disp('Λάθος ηλικία');  
    age = input('Δώσε την ηλικία σου: ');  
end
```

– Τι πρόβλημα έχει ο παραπάνω κώδικας ?

Έλεγχος τιμών εισόδου

- Η `while` μας βοηθάει να ελέγχουμε επαναλαμβανόμενα αν η τιμή που έδωσε ο χρήστης είναι σωστή και σε περίπτωση λάθους να ζητάμε να ξαναδώσει σωστή τιμή (όσες φορές χρειαστεί).

- Παράδειγμα:

```
age = input('Δώσε την ηλικία σου:');
```

```
while age < 0
    disp('Λάθος ηλικία');
    age = input('Δώσε την ηλικία σου:');
end
```

- Με τη `while` το πρόγραμμά μας θα ζητάει από το χρήστη να ξαναδώσει τιμή έως ότου η τιμή περάσει τον έλεγχο.
 - Αυτό θα γίνεται όσες φορές χρειαστεί.

Άσκηση: Έλεγχος τιμής εισόδου

- Να γραφεί πρόγραμμα το οποίο θα ζητάει από τον χρήστη να εισάγει έναν ακέραιο αριθμό από 1 έως 4. Σε περίπτωση που ο χρήστης εισάγει διαφορετικό αριθμό το πρόγραμμα θα ζητάει από τον χρήστη να ξανα-εισάγει έναν σωστό αριθμό.
- Λύση:

```
ch = input('Δώσε αριθμό: ');

while ch ~= 1 & ch ~= 2 & ch ~= 3 & ch ~= 4
    ch = input('Λάθος αριθμός. Δώσε ξανά: ');
end

fprintf('Έδωσες: %d\n', choice);
```

Συχνά λάθη

- Η `while` δεν εκτελείται ποτέ λόγω λάθος συνθήκης:

```
i = 99;

while i < 10
    disp(i);
    i = i - 1;
end
```

- Η `while` δεν τερματίζεται ποτέ λόγω λάθος συνθήκης:

```
i = 0;

while i < 10
    disp(i);
    i = i * 2;
end
```

Συχνά λάθη

- Η λογική συνθήκη της `while` μένει ανεπηρέαστη από τις εντολές επανάληψης που βρίσκονται μέσα στη `while`:

```
sum = 0
i = 10;

while i > 0
    sum = sum + 1;
end

disp(sum);
```

- Ο αριθμός των επαναλήψεων δεν είναι σωστός λόγω λάθος ορίων στην συνθήκη (π.χ., χρήση `<` αντί `<=`, `>` αντί `>=` και ανάποδα):

```
i = 1;
while i < 10
    disp(i);
    i = i + 1;
end
```

Βρείτε το λάθος

- Τι μπορεί να πάει λάθος στον παρακάτω κώδικα?

```
x = input('Δώσε αριθμό: ');  
  
while x ~= 0  
    disp(x);  
    x = x - 1;  
end
```

Απάντηση: Αν ο χρήστης δώσει αρνητικό αριθμό, τότε η επανάληψη δεν θα τελειώσει ποτέ

Συμβουλές

- Χρησιμοποιούμε βρόγχους για επανάληψη μόνο όταν είναι απαραίτητο:
 - εντολές `for` για βρόγχους καταμέτρησης
 - εντολές `while` για βρόγχους υπό συνθήκη
- Στοιχίζουμε τις εντολές των βρόγχων κατάλληλα.