

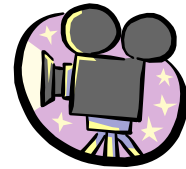


# Μεταβλητή Τύποι δεδομένων

## Εισαγωγή στον Προγραμματισμό

(CEID\_NY131)

## Variable Data Types



Kleanthis Thramboulidis  
Prof. of Software and System Engineering  
University of Patras  
<https://sites.google.com/site/thramboulidiskleanthis/>

## Βασικές Κατασκευές Γλώσσας Προγραμματισμού

- Αναπαράσταση διεργασιών
  - συνάρτηση
- Για αποθήκευση και ανάκληση πληροφορίας
  - **Μεταβλητές**
  - **Σταθερές**
  - **Τύποι δεδομένων**
- Για επεξεργασία πληροφορίας
  - τελεστές
- Έλεγχο ροής προγράμματος
  - Προτάσεις επανάληψης, διακλάδωσης, ..



## Μεταβλητή Ορισμός 1/3

- **Συμβολικό όνομα**
  - που δίνει λύση στο πρόβλημα της αναφοράς σε πληροφορία της κύριας μνήμης του υπολογιστή.
- Το **όνομα της μεταβλητής** μας επιτρέπει να αναφερόμαστε
  - **στη τιμή**, χωρίς να είναι απαραίτητο να γνωρίζουμε την ακριβή διεύθυνση της μνήμης που αυτή είναι αποθηκευμένη,
  - **στην θέση της μνήμης** για να τροποποιήσουμε την τρέχουσα-τιμή της.

```
count = count + 1
```



## Μεταβλητή Ορισμός 2/3

- είναι (σύμφωνα με τον Horowitz) μια τετράδα που αποτελείται από
  - ένα όνομα
  - ένα σύνολο ιδιοτήτων
  - μία αναφορά και
  - μία τιμή.



## Μεταβλητή Ορισμός 3/3

- **ΟΙ ιδιότητες**
  - συνήθως εκφράζουν το είδος των τιμών που μπορεί να πάρει
  - ορίζονται
    - στη διάρκεια της μεταγλώττισης και δεν μεταβάλλονται (π.χ. C) ή
    - στο χρόνο εκτέλεσης και μεταβάλλονται
- **η αναφορά**
  - προσδιορίζει μια περιοχή αποθήκευσης που είναι μικρότερη ή μεγαλύτερη από μία θέση μνήμης



## Δήλωση μεταβλητής στη C 1/2

- **προσδιορίζει** (στη διάρκεια μεταγλώττισης) τις ιδιότητες της μεταβλητής
  - τύπο δεδομένων
  - απαγόρευση αλλαγής τιμής στο χρόνο εκτέλεσης (**const**)
- **παραχωρεί** ένα συμβολικό όνομα
  - σε θέση μνήμης που θα δεσμευτεί στη διάρκεια εκτέλεσης
- **ενημερώνει** τον compiler για
  - πόσα bytes πρέπει να δεσμευθούν στη διάρκεια εκτέλεσης για την αποθήκευση της τιμής της
  - πως αυτά τα bytes πρέπει να ερμηνευθούν

## Δήλωση μεταβλητής στη C 2/2

- κάθε μεταβλητή πρέπει να δηλωθεί πριν χρησιμοποιηθεί
- Παραδείγματα δηλώσεων
  - `int num1;` // δηλώνει ακέραια μεταβλητή
  - `double result;` // δηλώνει μεταβλητή κινητής υποδιαστολής διπλής ακρίβειας
  - `const double pi = 3.1416;` // όπως η `result` παραπάνω αλλά σταθερά
- δήλωση με ανάθεση αρχικής τιμής
  - `int num1 = 12;`
  - `double temp = 24.35;`

© Κλεάνθης Θραμπουλίδης      Μεταβλητή – Τύποι δεδομένων      7

## Κανόνες δήλωσης μεταβλητών




- μέχρι 31 χαρακτήρες (ANSI C)
  - lower case, upper case, digits, underscore
  - πρώτος χαρακτήρας letter or underscore
- δήλωσε κάθε μεταβλητή πριν τη χρήση της
  - αποφεύγονται δύσκολα στην ανεύρεση λάθη
- βάλε όλες τις μεταβλητές σε μια θέση
- χρησιμοποίησε εκφραστικά ονόματα (readability)
- Προτίμησε τις τοπικές (**local**) αντί για τις γενικές μεταβλητές (**global**), εκτός και αν υπάρχει λόγος για αυτό.
 

Μεταβλητή που ορίζεται έξω από συνάρτηση συνάρτηση	Μεταβλητή που ορίζεται μέσα στο σώμα συνάρτησης
--	---

© Κλεάνθης Θραμπουλίδης      Μεταβλητή – Τύποι δεδομένων      8

## Τύποι Δεδομένων

- Ένας τύπος δεδομένων **προσδιορίζει**
  - ένα σύνολο από αντικείμενα με κοινά χαρακτηριστικά τα οποία συνήθως εκφράζονται από ένα σύνολο πράξεων πάνω στα αντικείμενα.



© Κλεάνθης Θραμπουλίδης      Μεταβλητή – Τύποι δεδομένων      9

## Βασικοί τύποι δεδομένων

- **Αριθμοί**
  - βασίζονται στην υλοποίηση της αριθμητικής της μηχανής
- **Boolean**
  - με τιμές true, false
- **Χαρακτήρας**
- **Αλφαριθμητικό (String)**

© Κλεάνθης Θραμπουλίδης      Μεταβλητή – Τύποι δεδομένων      10



## Τύποι Δεδομένων

- Οι σύγχρονες προτακτικές γλώσσες
  - **περιλαμβάνουν** ορισμένους **ενσωματωμένους τύπους**,
    - **int, char, float, double ...**
  - **προσφέρουν** ένα **μηχανισμό για τον ορισμό νέων τύπων**
    - `struct student { ... };`
    - `struct rectangle { ... };`
- Η δυνατότητα ορισμού νέων τύπων αυξάνει:
  - όχι μόνο την **αναγνωσιμότητα** του προγράμματος, αλλά και
  - την **αξιοπιστία** του.



## Τύποι δεδομένων στη C

- **βασικοί**
  - **char** (χαρακτήρας)
  - **int** (ακέραιος)
  - **float** (κινητής υποδιαστολής απλής ακρίβειας)
  - **double** (κινητής υποδιαστολής διπλής ακρίβειας)
- **πίνακας** (array) `[]`
- **δομή struct** - **ένωση** union
- **δείκτης** `*`
- **απαριθμητικός τύπος** enum





## Εκτύπωση ακεραίων 1/2

```
int ten = 10;  
printf("%d minus %d is %d\n", ten, 2, ten-2);
```

**10 minus 2 is 8**

### ■ Δεκαδικοί, Οκταδικοί, Δεκαεξαδικοί

- Προσδιοριστές %d, %o, %x

```
int num = 100;  
printf("dec=%d; , octal=%o; , hex=%x\n", num,  
num, num);
```

**dec=100; octal =144; hex=64**



## Εκτύπωση ακεραίων 2/2

### ■ long, short, unsigned

- %u
- %ld, %lu     l -> long
- %lx
- %lo
- %hd, %hu     h -> short
- %ho
- %hx

**χρησιμοποιώντας λάθος σύμβολο έχουμε απρόβλεπτα αποτελέσματα**



## Χαρακτήρας - char 1/3

- κάθε χαρακτήρας έχει ένα μοναδικό αριθμητικό κωδικό
- **ASCII code**  
American Standard Code for Information Interchange
- **EBCDIC code** (IBM large computers)  
Extended Binary-Coded Decimal Interchange Code
- `char ch = 'A'; char ch = 65;`

Γράψτε ένα πρόγραμμα για την εμφάνιση των χαρακτήρων a-z και των αντίστοιχων ascii κωδικών τους

```
char ch;
printf("Enter a character:")
scanf("%c", &ch);
printf("Its numeric code value is : %d\n",ch)
```

## Χαρακτήρας - char 2/3

```
char ch = 'A';           /* ο.k */

ch = "A";                /* λάθος */

ch = '\65';             /* Not portable */

ch = 65;                 /* Not portable */

ch = A;                 /* η A θεωρείται μεταβλητή */
```

### ASCII codes

A -> 65  
Z -> 90  
a -> 97  
z -> 122



## Χαρακτήρας - char 3/3

- Οι χαρακτήρες αντιμετωπίζονται ως μικροί αριθμοί

```
int j;  
j = 'A' + 'B';  
printf("%c\n", j);
```



## Αποθήκευση-Ανάκληση χαρακτήρα

Δεσμεύει χώρο για χαρακτήρα

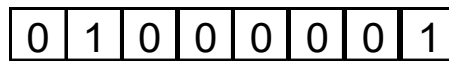
Αποδίδει τιμή στην μεταβλητή

```
int ch;
```

```
ch = 'A';
```

```
printf("%c", ch);
```

```
printf("%d", ch);
```



A

65

Τυπώνει την τιμή της μεταβλητής ως χαρακτήρα

Τυπώνει την τιμή της μεταβλητής ως ακέραιο

## Μη εκτυπούμενοι χαρακτήρες

Escape character sequences

- `\a` alert
- `\b` backspace
- `\f` form feed
- `\n` newline
- `\r` carriage return
- `\t` horizontal tab
- `\v` vertical tab
- `\\` backslash
- `\'` single quote

```
printf("Write, \"a \\ is a  
backslash. \\n");
```

Write, "a \ is a backslash."

© Κλεάνθης Θραμπουλίδης      Μεταβλητή – Τύποι δεδομένων      21

## Float - Double

- float** -> 32 bits (8 για εκθέτη και πρόσημο)  
(24 για βάση)
- double** -> συνήθως 64 bits  
μερικά συστήματα χρησιμοποιούν τα επιπλέον 32 bits για βάση αυξάνοντας την ακρίβεια

- Εκτύπωση
  - `%f` για δεκαδικό
  - `%e` για εκθετικό

```
float plank = 6.63e-34;
```

© Κλεάνθης Θραμπουλίδης      Μεταβλητή – Τύποι δεδομένων      22

# Αλφαριθμητικό ως πίνακας χαρακτήρων

Η C **δεν έχει τύπο για το αλφαριθμητικό**, το αντιμετωπίζει ως πίνακα χαρακτήρων.

```
char name[20] = {'h','e','l','l','o','\0'};
```

```
char name[] = {'h','e','l','l','o','\0'};  
char name[] = "hello";
```

Δεσμεύουν χώρο για τον αριθμό των χαρακτήρων συν ένα για τον χαρακτήρα '\0'. Στην περίπτωση μας 6 (5+1).

Δες άσκηση 2 Κεφάλαιο 11  
Διαχείριση Αλφαριθμητικού

Αναφορά στοιχείου	Μνήμη	Διεύθυνση
name[0]	h	F000
name[1]	e	F001
name[2]	l	F002
	l	F003
	o	F004
	\0	F005
		F006
....		
name[19]		F019

Το name είναι η διεύθυνση στο πρώτο στοιχείο του πίνακα

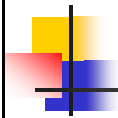
# Σταθερές (const) 1/2

## ■ Int

- 245 αποθηκεύεται ως int
- 100.000 αποθηκεύεται ως long
- για ελεγχόμενη αποθήκευση χρησιμοποιώ l ή L

## ■ Αλφαριθμητική σταθερά

- ακολουθία από κανένα ή περισσότερους χαρακτήρες κλεισμένους σε διπλά εισαγωγικά.  
“” “ ” “Hello world”



## Σταθερές 2/2

- Οι compilers by default θεωρούν τις σταθερές κινητής υποδιαστολής, ως διπλής ακρίβειας (double)
  - + μεγαλύτερη ακρίβεια,
  - μείωση ταχύτητας εκτέλεσης
- **χρησιμοποίησε :**
  - 9.11E9**F** για float
  - 4.32e4**L** για long double



## Σταθερά απαρίθμησης - Enum

- **επιτρέπει**
  - τη δήλωση συμβολικών ονομάτων για τη παράσταση ακέραιων σταθερών
- **δημιουργεί**
  - ένα **νέο τύπο** στον οποίο μπορείς να ορίσεις τις τιμές που μπορεί να πάρει μεταβλητή του τύπου αυτού.

```
enum spectrum {red,orange,green,blue};  
enum spectrum color;  
enum {bright, medium, dark} intensity;
```

```
color = blue;           /* OK */  
color = bright;        /* type conflict */  
color = 1;             /* type conflict */  
color=red + green; /*misleading usage*/
```