Δεδομένα: μεταβλητές και σταθερές

- Μια **μεταβλητή** είναι μια **θέση μνήμης** με συγκεκριμένο όνομα. Το περιεχόμενο της θέσης μνήμης είναι η **τιμή** της μεταβλητής.
- Σε μια μεταβλητή μπορούμε να εκχωρήσουμε αρχικά μια τιμή ενώ η τιμή αυτή μπορεί να αλλάξει (να μεταβληθεί) στη συνέχεια κατά την εκτέλεση του προγράμματος.
- Η προσπέλαση της τιμής μιας μεταβλητής γίνεται με άμεσο τρόπο καθώς γνωρίζουμε εκ των προτέρων τη διεύθυνση της αντίστοιχης θέσης μνήμης και ονομάζεται **άμεση προσπέλαση μνήμης**.
- Σε αντίθεση, οι **σταθερές** προκαθορίζονται και δεν αλλάζουν τιμή κατά την εκτέλεση του προγράμματος.

Ονόματα μεταβλητών

- Το όνομα μιας μεταβλητής περιλαμβάνει πεζά [a, b, …, z] ή κεφαλαία [A, B, …, Z] γράμματα του αγγλικού αλφαβήτου, ψηφία [0, 1, …, 9] ή τον χαρακτήρα υπογράμμισης '_'.
- Ο πρώτος χαρακτήρας πρέπει να είναι είτε γράμμα ή ο χαρακτήρας υπογράμμισης (δεν επιτρέπεται να ξεκινάει από ψηφίο).
- Στην C, οι πεζοί χαρακτήρες εκλαμβάνονται ως διαφορετικοί από τους κεφαλαίους κατά την ονοματοθεσία των μεταβλητών (π.χ. οι μεταβλητές **X1** και **x1** ή οι **area** και **Area** είναι διαφορετικές).
- Δεν επιτρέπεται μια μεταβλητή να έχει το ίδιο όνομα με κάποια **δεσμευμένη** λέξη της C.
- Είναι καλό να δίνονται μνημονικά ονόματα στις μεταβλητές μας.

Οι δεσμευμένες λέξεις της C

auto	break	case	char	const
continue	default	do	double	else
enum	extern	float	for	goto
if	int	long	register	return
short	signed	sizeof	static	struct
switch	typedef	union	unsigned	void
volatile	while			

Τύποι Δεδομένων

```
#include <stdio.h>
int main()  /* data types */
{
  int i,j,hours,months;  /* integer */
  short int years = 12;  /* short integer */
  short s1;              /* short integer */
  long int seconds;      /* long integer */
  long l1,l2;            /* long integer */
  unsigned int u,v;      /* unsigned integer */
  unsigned u1,v1;        /* unsigned integer */
  unsigned short us1;    /* unsigned short */
  unsigned long ul1;     /* unsigned long */
  char ch = 'A';         /* character */
  float pi = 3.14;       /* floating point */
  double x,y,z;          /* double precision */

  printf("int: %d bytes\n",sizeof (int));
  printf("short: %d bytes\n",sizeof (short));
  printf("long: %d bytes\n",sizeof (long));
  printf("float: %d bytes\n",sizeof (float));
  return 0;
}
```

Τύποι Δεδομένων

```
#include <stdio.h>
int main()  /* data types */
{
  int i,j,hours,months;  /* integer */
  short int years = 12;  /* short integer */
  short s1;              /* short integer */
  long int seconds;      /* long integer */
  long l1,l2;            /* long integer */
  unsigned int u,v;      /* unsigned integer */
  unsigned u1,v1;        /* unsigned integer */
  unsigned short us1;    /* unsigned short */
  unsigned long ul1;     /* unsigned long */
  char ch = 'A';         /* character */
  float pi = 3.14;       /* floating point */
  double x,y,z;          /* double precision */

  printf("int: %d bytes\n",sizeof (int));
  printf("short: %d bytes\n",sizeof (short));
  printf("long: %d bytes\n",sizeof (long));
  printf("float: %d bytes\n",sizeof (float));
  return 0;
}
```

Οι τύποι δεδομένων της C

Η C χρησιμοποιεί τους εξής βασικούς τύπους δεδομένων: **char**, **int**, **float**, **double**. Με χρήση όμως των **προσδιοριστών** (**qualifiers**) **short**, **long**, **signed**, **unsigned** οι τύποι της C είναι όπως στον παρακάτω πίνακα.

Τύπος Δεδομένων	Σύνηθες όνομα τύπου	Μέγεθος σε bytes	Εύρος τιμών
signed char	char	1	-128 έως 127
unsigned char	unsigned char	1	0 έως 255
short int	short	2	- 32.768 έως 32.767
unsigned short int	unsigned short	2	0 έως 65.535
int	int	4	-2.147.483.648 έως 2.147.483.647
unsigned int	unsigned int	4	0 έως 4.294.967.295
long int	long	4	-2.147.483.648 έως 2.147.483.647
unsigned long int	unsigned long	4	0 έως 4.294.967.295
float	float	4	Θετικοί: 1.17e-38 έως 3.4e38 Αρνητικοί: -3.4e38 έως -1.17e-38
double	double	8	περίπου: $-1.8*10^{308}$ έως $1.8*10^{308}$
Σε μεταγενέστερα πρότυπα υπάρχουν κι'άλλοι τύποι. Π.χ. **long long**, **long double**			

Εκχώρηση τιμών σε μεταβλητές

- Η **εκχώρηση τιμής** σε μια μεταβλητή αρχικοποιεί την μεταβλητή. Μπορεί να γίνει είτε κατά τη δήλωση της μεταβλητής είτε και αργότερα. Παράδειγμα:

```
...
int  a, b, c = 5;
float r = 2.0;

a = 0;
b = 1;
...
```

Εκχώρηση κατά τη δήλωση

Εκχώρηση μετά τη δήλωση

Υπερχείλιση (overflow)

```
/* overflow example */

#include <stdio.h>
int main()
{
  short int   i = 32767;
  unsigned char  c = 255;

  printf("i+1 = %d, c+2 = %d\n", i+1, c+2);
  return 0;
}
```

Έξοδος προγράμματος:

```
i+1 = -32768, c+2 = 1
```

Ακρίβεια τύπου float (precision)

```
/* float precision example */
#include <stdio.h>
int main()
{
  float  pi = 3.14;  /* double is better */

  if (pi == 3.14)
    printf("H perimetros einai %f",2*pi);
  else
    printf("Lathos timh gia to pi.");
  return 0;
}
```

Έξοδος προγράμματος:

```
Lathos timh gia to pi.
```

Προσαρμογή τύπου (type casting)

Γενική μορφή προσωρινής προσαρμογής ενός τύπου δεδομένων σε κάποιον άλλο:

```
(τύπος δεδομένων) παράσταση
```

Παραδείγματα:

```
float  x, y, z, r=5.27, pi=3.14;
x = 2 * pi * r;          /* x = 33.0956 */
y = 2 * (int)pi * r;     /* y = 31.62 */
z = 2 * (int)(pi * r);   /* z = 32.0 */
x = 5/4;            /* x = 1.0 */
y = (float) 5/4;    /* y = 1.25 ⇔ 5.0/4 */
y = 5/(float) 4;    /* y = 1.25 ⇔ 5/4.0 */
z = (float)(5/4);   /* z = 1.0 */
```

Η `printf()`

- Χρησιμοποιείται για την εμφάνιση αριθμητικών δεδομένων και χαρακτήρων στην οθόνη.
- Το πρώτο όρισμα της `printf()` είναι μια ακολουθία χαρακτήρων (συμβολοσειρά) η οποία καθορίζει τον τρόπο εμφάνισης των δεδομένων στην οθόνη.
- Τα επόμενα ορίσματα (αν υπάρχουν) είναι τα δεδομένα (λίστα μεταβλητών) που θέλουμε να εμφανίσουμε.
- Η θέση εμφάνισης των δεδομένων καθορίζεται από τα λεγόμενα **προσδιοριστικά μετατροπής** (**modifiers**) δηλαδή, τον χαρακτήρα '%' ακολουθούμενο από κάποιο συγκεκριμένο χαρακτήρα που προσδιορίζει τον τύπο των δεδομένων προς εμφάνιση.

Προσδιοριστικά μετατροπής της `printf()`

`%d` ή `%i`	εμφάνιση ακεραίου
`%f`	εμφάνιση πραγματικού αριθμού
`%u`	εμφάνιση θετικού (μη προσημασμένου) ακεραίου
`%c`	εμφάνιση χαρακτήρα
`%s`	εμφάνιση συμβολοσειράς
`%e` ή `%E`	εμφάνιση πραγματικού σε εκθετική μορφή
`%g`	χρήση της πιο σύντομης μορφής από `%f` ή `%e`
`%p`	εμφάνιση διεύθυνσης μνήμης μιας μεταβλητής
`%x` ή `%X`	εμφάνιση ακεραίου σε δεκαεξαδική μορφή
`%o`	εμφάνιση ακεραίου σε οκταδική μορφή
`%%`	εμφανίζει τον χαρακτήρα '%'

```
#include <stdio.h>
#define PI 3.14159        /* Ορισμός σταθεράς */

int main()
{
  float r,perim,area;
  char  name[40];         /* character string*/

  printf("Geia sou! Poio einai to onoma sou?\n");
  scanf("%s",name);       /* Εισαγωγή ονόματος */

  printf("%s, dosmou mia aktina kuklou \n",name);
  scanf("%f",&r); /* Εισαγωγή πραγματικού αριθμού */

  perim = 2.0*3.14*r;
  area = PI*r*r;

  printf("Perimetros kuklou: %f\n",perim);
  printf("Epifaneia kuklou: %f\n",area);

  return 0;
}
```

Η `scanf()`

- Χρησιμοποιείται για την είσοδο δεδομένων από το πληκτρολόγιο.
- Η **`scanf()`** χρησιμοποιεί μια ακολουθία ειδικών χαρακτήρων μετατροπής (%d, %f, κ.λπ.) για το διάβασμα των δεδομένων και στη συνέχεια εκχωρεί τα δεδομένα σε μια ή περισσότερες μεταβλητές.
- Για την εκχώρηση των δεδομένων στις μεταβλητές, η **`scanf()`** χρειάζεται (στα επόμενα ορίσματα) τις διευθύνσεις των μεταβλητών στη μνήμη του συστήματος.
- π.χ. **`scanf("%f%d",&r, &a);`**
  Διάβασμα ενός πραγματικού και ενός ακεραίου από το πληκτρολόγιο και αποθήκευσή τους στη διεύθυνση της μεταβλητής r και στη διεύθυνση της μεταβλητής a αντίστοιχα.