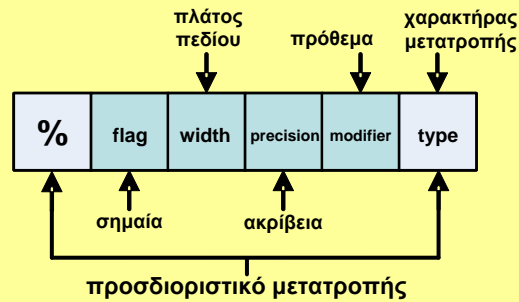


Μορφοποίηση δεδομένων: προαιρετικά πεδία προσδιοριστικών μετατροπής



Ακρίβεια (precision)

Για τους τύπους float/double καθορίζει τον αριθμό των δεκαδικών ψηφίων που θα εμφανισθούν με κατάλληλη στρογγυλοποίηση (default: 6)

Παραδείγματα:

```
printf("x = %f\n", x); /* εμφανίζει 6 δεκαδικά ψηφία */
```

```
printf("x = %.f\n", x); /* δεν εμφανίζει δεκαδικά ψηφία */
```

```
printf("x = %.3f\n", x); /* εμφανίζει 3 δεκαδικά ψηφία */
```



```
printf("x = %.*f\n", 3, x); /* εμφανίζει 3 δεκαδικά ψηφία */
```

```
printf("How many decimal digits? ");
```

```
scanf("%d", &n);
```

```
printf("x = %.*f\n", n, x); /* εμφανίζει n δεκαδικά ψηφία */
```

Ακρίβεια (precision)

Για συμβολοσειρές μπορούμε να καθορίσουμε το πλήθος των χαρακτήρων που θα εμφανισθούν με την ίδια τεχνική. Ο αριθμός μετά την τελεία καθορίζει το πλήθος των χαρακτήρων προς εμφάνιση.

Παραδείγματα:

```
#define STR "This is a string"

printf("x = %s\n", STR); /* εμφανίζει όλους τους χαρακτήρες */
printf("x = %.7s\n", STR); /* εμφανίζει 7 χαρακτήρες */

This is

printf("How many characters to print? ");
scanf("%d", &n);

printf("x = %.*s\n", n, STR); /* εμφανίζει n χαρακτήρες */
printf("x = %.s\n", STR); /* δεν εμφανίζει τίποτα */
```

Πλάτος πεδίου

Το πλάτος πεδίου καθορίζει, με έναν ακέραιο μετά το '%' και πριν την '.', το ελάχιστο πλήθος των χαρακτήρων που θα εμφανισθούν συμπεριλαμβανομένων και των ψηφίων ακριβείας ώστε να χωράει ο αριθμός.

Παραδείγματα:

```
int i = 100, j = -5;
float x = 1.15719, y = -145.25;
printf("|%d|,|%d|,|%f|,|%f|\n", i, j, x, y);
printf("|%4d|,|%4d|,|%8.2f|,|%8.2f|\n", i, j, x, y);
printf("|%2d|,|%1d|,|%5f|,|%5f|\n", i, j, x, y);
```

Εξοδος

```
|100|,|-5|,|1.157190|,|-145.250000|
```

```
| 100|,| -5|,| 1.16|,| -145.25|
```

```
|100|,|-5|,|1.157190|,|-145.250000|
```

πλάτος πεδίου
8 χαρακτήρων

Πλάτος πεδίου

Στην περίπτωση που το πλάτος μπορεί να καθορισθεί αργότερα, π.χ. με χρήση της `scanf`, υπάρχει η δυνατότητα να χρησιμοποιηθεί το `'*'` αμέσως μετά το `'%'` , και το πλάτος πεδίου να καθορισθεί στη λίστα ορισμάτων της `printf`.

Παραδείγματα:

```
int i = 100, d1, d2, a;    float x = 1.15719;

printf("Dose plath pediou kai akribeia: \n");

scanf("%d%d%d", &d1, &d2, &a); /* έστω: 4 8 2 <enter> */

/* printf("|%4d|,|%8.3|,|%8.2f|\n", i, x, x); */
```

↕

```
printf("|%*d|,|%*.3f|,|%*.*f|\n", d1, i, d2, x, d2, a, x);
```

Έξοδος

```
| 100|,| 1.157|,| 1.16|
```

Πρόθεμα

Προσδιορίζει μέγεθος τύπου: `h` για `short` και `l` ή `L` για `long`. Το πρόθεμα τοποθετείται αμέσως πριν τον καθορισμό του τύπου.

Παράδειγμα:

```
#include <stdio.h>

int main(void)
{
    short i=10;    int j=32768;    long k=1000000;
    float x=1.23;    double y=1.23;
    printf("i=%hd, j=%hd, j=%ld, k=%ld\n", i, j, j, k);
    printf("x=%.15Lf, y=%.15Lf\n", x, y);
    return 0;
}
```

Έξοδος

```
i=10, j=-32768, j=32768, k=1000000
x=1.230000019073486, y=1.230000000000000
```

↙ 0 32768 είναι εκτός των ορίων του short int (-32768 ... 32767)

Σημαίες

Οι σημαίες τροποποιούν τις εξ'ορισμού επιλογές μορφοποίησης και τοποθετούνται ακριβώς μετά το %. Παρακάτω αναφέρονται 4 από αυτές:

- Στοιχίση στα αριστερά
- 0 Το πεδίο συμπληρώνεται με μηδενικά αντί για κενά
- + Όλοι οι αριθμοί εμφανίζονται με το πρόσημό τους
- κενό Οι θετικές τιμές ξεκινούν με κενό χαρακτήρα

Παραδείγματα:

```
int i = 100;    float x = 1.23;
printf("|%-10d|,| %+6.2f|,| %-+6.2f|,| % f|",i,x,x,x);
printf(",| %06.2f|\n",x);
```

Έξοδος

```
|100          |,| +1.23|,|+1.23 |,| 1.230000|,|001.23|
```