Adattípusok

A C nyelvben az adattípusok határozzák meg, hogy egy változó milyen típusú adatot tárolhat, és mennyi memóriát foglal.

Alap adattípusok

- int – egész számok tárolására (pl. -5, 0, 42)
- float – lebegőpontos szám, kisebb pontossággal (pl. 3.14, -0.5)
- double – lebegőpontos szám nagyobb pontossággal (pl. 3.1415926535)
- char – egyetlen karakter tárolására (pl. 'a', 'Z', '1')

Megjegyzés: a char valójában egy kis egész számot (ASCII kódot) tár

Változók és konstansok

Változó: olyan névvel ellátott memóriahely, amelyben az érték program közben változhat.

```
int x = 5;     // egész típusú változó
float pi = 3.14; // lebegőpontos változó (tört, 32bit)
double tort = 4.6765 // lebegőpontos változó (tört, 64bit)
char betu = 'A'; // karakter változó
char szoveg[] = {'E', 'P', 'E', 'R'}; // karakter tömb (vektor), szöveg
```

Konstans: olyan érték, amely a program futása közben nem változtatható.

```
const double PI = 3.14159;
```

Be- és kimenet

A standard be- és kimenet a printf() (kiírás) és a scanf() (beolvasás).

Kiíratás (printf()):

```
int szam = 10;
printf("A szam erteke: %d\n", szam);
```

Formátumjelek:

- %d → egész szám (int)
- %f → lebegőpontos szám (float, double), double esetén inkább %lf
- %c → karakter (char)
- %s → szöveg (karaktertömb, string)

Beolvasás (scanf())

```
int x;
scanf("%d", &x);  // az & jel a változó címét adja át
```

Formátumjelek (ugyanaz, mint fentebb):

- %d → egész szám (int)
- %f → lebegőpontos szám (float, double), double esetén inkább %lf
- %c → karakter (char)
- %s → szöveg (karaktertömb, string)

Operátorok

Aritmetikai operátorok

- + összeadás
- - kivonás
- * szorzás
- / osztás
- % maradékos osztás (csak egész számokra)

Relációs (összehasonlító) operátorok

- == egyenlő
- != nem egyenlő
- > nagyobb
- < kisebb
- >= nagyobb vagy egyenlő
- ⇐ kisebb vagy egyenlő

Logikai operátorok

- && logikai ÉS
- || logikai VAGY
- ! logikai NEM

Értékadó operátorok

- = értékadás (pl. a = 5;)
- +=, -=, *=, /=, %= rövidített formák

Megjegyzés: a += 3; ugyanaz, mint: a = a + 3;

Példa program

```
#include <stdio.h>

int main() {
    int a, b;
    printf("Adj meg ket egesz szamot: ");
    scanf("%d %d", &a, &b);
```

```
    printf("Osszeg: %d\n", a + b);
    printf("Kulonbseg: %d\n", a - b);
    printf("Szorzat: %d\n", a * b);
    printf("Hanyados: %d\n", a / b);
    printf("Maradek: %d\n", a % b);

    if (a > b) {
        printf("Az elso nagyobb.\n");
    } else {
        printf("A masodik nagyobb vagy egyenlo.\n");
    }

    return 0;
}
```

Gyakorlás

Írj programot, ami kiírja a neved és az életkorod!

```
#include <stdio.h>

int main() {
    char nev[50];
    int kor;

    printf("Add meg a neved: ");
    scanf("%s", nev);

    printf("Add meg az eletkorod: ");
    scanf("%d", &kor);

    printf("Szia %s, te %d eves vagy!\n", nev, kor);
    return 0;
}
```

Kérj be két egész számot, és számítsd ki az összegüket, különbségüket, szorzatukat, osztásukat és maradékukat!

```
#include <stdio.h>

int main() {
    int a, b;

    printf("Adj meg ket szamot: ");
    scanf("%d %d", &a, &b);
```

```
    printf("Osszeg: %d\n", a + b);
    printf("Kulonbseg: %d\n", a - b);
    printf("Szorzat: %d\n", a * b);
    printf("Hanyados: %d\n", a / b);
    printf("Maradek: %d\n", a % b);

    return 0;
}
```

Kérj be három lebegőpontos számot, és számítsd ki az átlagukat!

```
#include <stdio.h>

int main() {
    float a, b, c, atlag;

    printf("Adj meg harom szamot: ");
    scanf("%f %f %f", &a, &b, &c);

    atlag = (a + b + c) / 3;
    printf("Az atlag: %.2f\n", atlag);

    return 0;
}
```

From:
https://edu.iit.uni-miskolc.hu/ - **Institute of Information Science - University of Miskolc**

Permanent link:
https://edu.iit.uni-miskolc.hu/tanszek:oktatas:szamitastechnika:adattipusok_valtozok_be-_es_kimenet_operatorok

Last update: **2025/09/24 07:53**