

# Struktúrák

## Mi az a struktúra?

A struktúra (structure) egy felhasználó által definiált adattípus, amely különböző típusú változókat (adatokat) képes egy név alatt összefogni. Ez hasonló az osztályhoz más nyelvekben, csak egyszerűbb, mert nincs benne metódus.

Célja: összetett adatok (pl. hallgató, autó, pont, dátum stb.) logikai egységbe foglalása.

## Struktúra definiálása

A struktúra típust a struct kulcsszóval hozzuk létre:

```
struct Szemely {
    char nev[50];
    int életkor;
    float magassag;
};
```

Ezután létrehozhatunk belőle változókat:

```
struct Szemely ember1, ember2;
```

VAGY egyből definiálhatjuk és példányosíthatjuk:

```
struct Szemely {
    char nev[50];
    int életkor;
    float magassag;
} ember1, ember2;
```

## Típusnév (typedef)

A typedef segítségével egyszerűbb nevet adhatunk a struktúrának:

```
typedef struct {
    char nev[50];
    int életkor;
    float magassag;
} Szemely;
```

```
Szemely ember1, ember2;  
//Így már nem kell mindenhol struct kulcsszót írni.
```

## Struktúratagok elérése

A tagokat a pont (.) operátorral érjük el:

```
strcpy(ember1.nev, "Kovacs Bela");  
ember1.eletkor = 25;  
ember1.magassag = 1.82;
```

Ha mutatóval dolgozunk, akkor a nyíl (→) operátort használjuk:

```
Szemely *ptr = &ember1;  
printf("%s %d %.2f", ptr->nev, ptr->eletkor, ptr->magassag);
```

## Struktúrák átadása függvényeknek

Érték szerint:

```
void kiir(Szemely s) {  
    printf("%s (%d ev)\n", s.nev, s.eletkor);  
}
```

Mutatóval (hatékonyabb):

```
void kiir(const Szemely *s) {  
    printf("%s (%d ev)\n", s->nev, s->eletkor);  
}
```

## Beágyazott struktúrák

Egy struktúra tartalmazhat másik struktúrát is:

```
typedef struct {  
    int ev, honap, nap;  
} Datum;  
  
typedef struct {  
    char nev[50];  
    Datum szuletes;
```

```
} Hallgato;
```

## Gyakorlás

1. Készíts egy struktúrát, ami egy pontot (x, y) tárol, és írja ki a koordinátáit!

```
#include <stdio.h>

struct Pont {
    int x;
    int y;
};

int main() {
    struct Pont p = {3, 7};
    printf("A pont koordinátái: (%d, %d)\n", p.x, p.y);
    return 0;
}
```

2. Kérj be 3 személy nevét és életkorát, majd írd ki őket!

```
#include <stdio.h>

typedef struct {
    char nev[50];
    int eletkor;
} Szemely;

int main() {
    Szemely emberek[3];
    for (int i = 0; i < 3; i++) {
        printf("%d. személy neve: ", i+1);
        scanf("%[^\n]", emberek[i].nev);
        printf("%d. személy életkora: ", i+1);
        scanf("%d", &emberek[i].eletkor);
    }

    printf("\n--- Adatok ---\n");
    for (int i = 0; i < 3; i++) {
        printf("%s (%d év)\n", emberek[i].nev, emberek[i].eletkor);
    }
}
```

```
    return 0;
}
```

3. Írj függvényt, ami egy személy adatait kiírja!

```
#include <stdio.h>

typedef struct {
    char nev[50];
    int kor;
} Szemely;

void kiir(const Szemely *s) {
    printf("%s - %d ev\n", s->nev, s->kor);
}

int main() {
    Szemely ember = {"Kiss Anna", 22};
    kiir(&ember);
    return 0;
}
```

4. Készíts struktúrát, ahol egy hallgató neve és születési dátuma van eltárolva!

```
#include <stdio.h>

typedef struct {
    int ev, honap, nap;
} Datum;

typedef struct {
    char nev[50];
    Datum szuletes;
} Hallgato;

int main() {
    Hallgato h = {"Nagy Peter", {2002, 4, 15}};
    printf("%s szuletett: %d.%02d.%02d\n",
           h.nev, h.szuletes.ev, h.szuletes.honap, h.szuletes.nap);
    return 0;
}
```

5. Adj meg 3 személyt, és írd ki, ki a legidősebb!

```
#include <stdio.h>
#include <string.h>

typedef struct {
    char nev[50];
    int kor;
} Szemely;

int main() {
    Szemely emberek[3] = {
        {"Kovacs Bela", 34},
        {"Kiss Anna", 28},
        {"Toth Eva", 41}
    };

    int maxIndex = 0;
    for (int i = 1; i < 3; i++) {
        if (emberek[i].kor > emberek[maxIndex].kor)
            maxIndex = i;
    }

    printf("A legidosebb: %s (%d ev)\n",
           emberek[maxIndex].nev, emberek[maxIndex].kor);
    return 0;
}
```

```
#include <stdio.h>

typedef struct{
    char nev[30];
    int kor;
    double magassag;
}EMBER;

int main()
{
    EMBER e;
    int osszeg = 0;
    double atlag = 0;
    double max = 0;
    for(int i = 0; i < 3; i++){
        printf("Adja meg az ember nevét: ");
        fgets(e.nev, sizeof(e.nev), stdin);
        printf("Adja meg a korát: ");
        scanf("%d", &e.kor);
        printf("Magassága: ");
        scanf("%lf", &e.magassag);
        osszeg += e.kor;
    }
}
```

```
        if(max < e.magassag){
            max = e.magassag;
        }
    }
    atlag = (double)osszeg / 3;
    printf("\n\nA korok átlaga: %lf", atlag);
    printf("\nLegnagyobb magassag: %lf", max);
}
```

```
#include <stdio.h>
#include <string.h>

struct DIAK{
    int id;
    char nev[50];
    double atlag;
};

int main()
{
    struct DIAK d1;
    d1.id = 1;
    strcpy(d1.nev, "Asd Feri");
    d1.atlag = 4.8;
    //d1 = {0,"sdg asg", 5.0};
    struct DIAK diakok[2] = {
        {2, "Jkl Éva", 4.4},
        {3, "Qwerty Béla", 3.6}
    };
    printf("ID: %d\n", d1.id);
    printf("Név: %s\n", d1.nev);
    printf("Átlag: %lf\n\n", d1.atlag);
    for(int i = 0; i < 2; i++){
        printf("ID: %d\n", diakok[i].id);
        printf("Név: %s\n", diakok[i].nev);
        printf("Átlag: %lf\n", diakok[i].atlag);
    }
}
```

//Összetett példa

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:strukturak?rev=1761724872>

Last update: **2025/10/29 08:01**

