

## 2. Adatok bevitele, kivitele konzol felületen

### Példa: Társalgás

A következő programrész egyszerű társalgást folytat a felhasználóval:

```
Console.WriteLine("Szervusz!");
Console.Write("Írd ide az utóneved:");
string utonev = Console.ReadLine();
Console.WriteLine("Szép neved van " + utonev + "!");
Console.Write("És mi a vezetékneved? ");
string veznev = Console.ReadLine();
Console.WriteLine("Teljes neved tehát {0} {1}.", veznev, utonev);
Console.Write("Add meg egy budapesti barátod irányítóját:");
string irszam = Console.ReadLine();
Console.WriteLine("Na hát a barátod a {0}. kerületben lakik!",
    irszam.Substring(1,2)); Console.Write

Console.ReadKey();
```

A programban a `Console`. kezdetű utasítások szolgálnak a karakteres felületen az adatok be és kivitelére.

A `Write` és a `WriteLine` a zárójelben lévő, macskakörmök közötti részt kiírják. A `WriteLine` a kiírás után új sort kezd.

A társalgást záró `Console.ReadKey()` egy billentyű lenyomását várja, utána bezáródik a konzol. A `string utonev = Console.ReadLine();` sorban a billentyűzetről a beolvas egy *szöveget* és azt az `utonev` nevű *változóban* eltárolja.

### A változó egy memóriaszelet ahol adatot tárolunk, névvel azonosítjuk.

A változót első használata előtt deklarálni kell, azaz meg kell adni típusát és nevét, pl.: `string utonev`

Az `utonev` változó `string` típusú, ez azt jelenti, hogy a változó szöveget (és nem pl. számot) tárol.

A változókat kiírásakor „bele is láncolhatjuk a szövegbe:

```
Console.WriteLine("Szép neved van " + utonev + "!");
```

de egyszerűbb megoldás a `{0}`, `{1}`... azonosítókkal helyettesíteni, és a változókat a számoknak megfelelő sorrendben vesszővel elválasztva a szöveget záró macskaköröm után felsorolni:

```
Console.WriteLine("Teljes neved tehát {0} {1}.", veznev, utonev);
```

### A szöveges típus (sztring)

A szöveges típus a szöveget karakterenként kódolva tárolja. A C# UNICODE-ot használ, így az ékezetes betűk és egyéb karakterek is helyesen jelennek meg.

A sztringekkel végezhető műveletek:

*Konkatenáció* (összefűzés): A két sztringet egymás után fűzi.

Például: `"fa" + "fej" → "faj"`. A művelet nem kommutatív, mivel `"fej" + "fa" → "fejfa"`

*Alsztring képzés:* A sztringből egy részéből új sztringet készít.

Például: "Gyermekkönyvtár".Substring(4,3)= "mek"

(Balról a negyedikkel kezdve három betűt választ ki,  
a karakterek számozása nullától indul.)

*Sztringek szétvágása:* A Split metódus a paraméterként megadott *karakternél* szétvágja a sztringet, és a szögletes zárójelben megadott sorszámú darabot adja vissza:

Például: utonev = teljesnev.Split(' ')[1];

A karaktert aposztrófok közé kell írni.

(Megjegyzés: A karakter ténylegesen egy szám, a karakter kódja.)

*Sztring hossza:* Például: irszam.Length = 4;

*Speciális karakterek a sztringek kiírásakor:*

" " → egy darab szóköz

"\n" → új sor

"\t" → tabulátor