


## Különleges karakterek

;	Alt Gr + ,?;
=	Shift + 7
()	Shift + 8 és 9
{ }	Alt Gr + B és N
[ ]	Alt Gr + F és G
” (macskaköröm)	Shift + 2
' (apoztróf)	Shift + 1
\	Alt Gr + Q
& (ÉS)	Alt Gr + C
(VAGY)	Alt Gr + W
< > (kisebb-nagyobb)	Alt Gr + Í és Y

```
namespace ErettsegiFeladat
```

```
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```

 Start	program fordítása, futtatása
namespace	névtér, objektumok összefoglalója
class	osztály
Main	a fő függvény, ezzel indul a program
static	osztályon kívül is hívható függvény
void	„üres”, nem tér vissza értékkel
args	ezekkel a paraméterekkel indul a program



## Változók

bool	logikai (igaz/hamis)	true, false
byte	8 bites, előjel nélküli egész szám	0 - 255
sbyte	8 bites, előjeles egész szám	-128 - 127
char	16 bites Unicode karakter	'a' '\x263a'
ushort	16 bites, előjel nélküli egész szám	0 - 65,535
short	16 bites, előjeles egész szám	-32,768 - 32,767
uint	32 bites, előjel nélküli egész szám	0 - 4,294,967,295
int	32 bites, előjeles egész szám	-2,147,483,648 - 2,147,483,647
ulong	64 bites, előjel nélküli egész szám	0 - 18,446,744,073,709,551,615
long	64 bites, előjeles egész szám	-9,223,372,036,854,775,808 - 9,223,372,036,854,775,807
float	32 bites lebegőpontos szám, 7 számjegy pontosság	$-3.4 \times 10^{38} - 3.4 \times 10^{38}$
double	64 bites lebegőpontos szám, 15 számjegy pontosság	$\pm 5.0 \times 10^{-324} - \pm 1.7 \times 10^{308}$
decimal	128 bites lebegőpontos szám, 28 számjegy pontosság	$-7.9 \times 10^{28} - 7.9 \times 10^{28}$
string	szöveg	"hello" "szöveg"
object	objektum	minden



## Nyelvtan

függvény, amit mi csinálunk

```
void Fuggveny(){ }
```

void	Fuggveny	()	{ }
visszatérési érték, mit ad (semmi, alma, körte, 42)	függvény neve	paraméterek	függvény törzse, ebbe jönnek a parancsok

függvény, amit használunk, paraméter nélkül

```
WriteLine ();
```

WriteLine	()	;
függvény neve	zárójelek (shift + 8-9)	parancs vége (Alt Gr + ,?;)

függvény amit használunk, egy paraméterrel

```
WriteLine (42);
```

WriteLine	(	42	)	;
függvény neve	paraméter kezdődik	paraméter	paraméter vége	parancs vége

függvény amit használunk, több paraméterrel

```
SetWindowPosition(30, 40);
```

SetWindowPosition	(	800	,	600	)	;
függvény neve	paraméterek kezdődnek	első paraméter	paraméter elválasztása	második paraméter	paraméterek vége	lezáró pontosvessző



## Struktúra, osztály, objektum

```
class Szemely
{
    public string Nev { get; set; }
    public int Kor { get; protected set; }
    public Szemely(string nev, int kor)
    {
        this.Kor = kor;
        this.Nev = nev;
    }
}
Szemely sz = new Szemely("Sanyi", 42);
sz.Kor = 43; // mert protected!
int SanyiKora = sz.Kor; // jó!
sz.Nev = "Sándor";
```

## Adattípusok

```
string[] szovegTomb = new string[5];
int[,] szorzotabla = new int[10, 10];
List<string> szovegLista = new List<string>();
LinkedList<int> Lanclista = new LinkedList<int>();
```

## Kiírás

```
System.Console.Write("Szöveg");
System.Console.WriteLine("Szöveg");
WriteLine("param1: {0}, param2: {1}", 1, 2);
```

## Felhasználótól bekérés

Hamarosan!



## LINQ

```
List<Szemely> RendezettSzemelyek =  
Szemelyek.OrderBy(szemely => szemely.Kor).ToList();
```

```
List<Szemely> VisszafeleRendezettSzemelyek =  
    Szemelyek.OrderByDescending(szemely =>  
        szemely.Kor).ToList();
```

```
List<Szemely> IdosSzemelyek =  
    Szemelyek.Where(szemely =>  
        szemely.Kor > 70).ToList();
```

```
List<Szemely> HosszuNevuSzemelyek =  
    Szemelyek.Where(szemely =>  
        szemely.Nev.Length > 6).ToList();
```

```
List<Szemely> OeBetusNevuSzemelyek =  
    Szemelyek.Where(szemely =>  
        szemely.Nev.Contains("ö") ||  
        szemely.Nev.Contains("Ö")).ToList();
```

