

Microsoft Small Basic

Rövid leírás kezdőknek

I. rész



Kedves versenyzők!

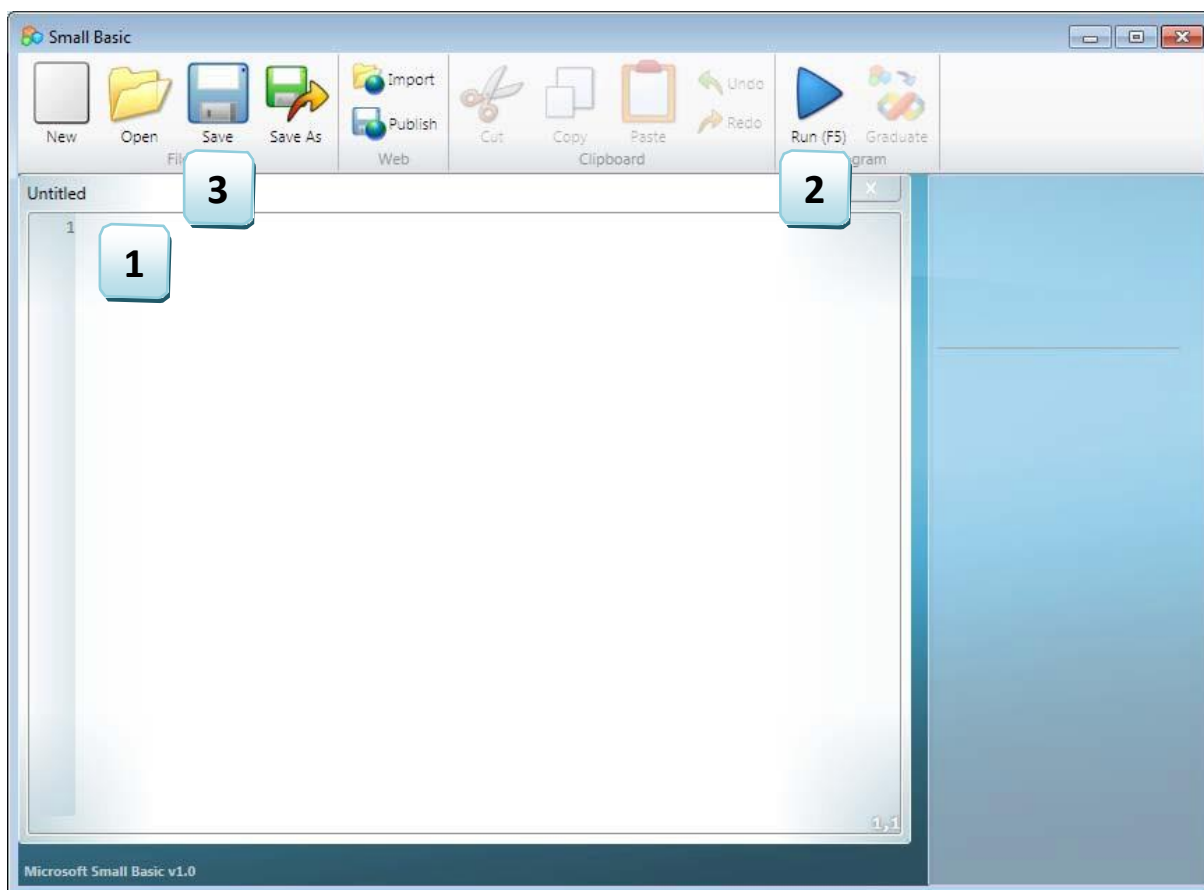
A következőkben egy új, és lehet, hogy eddig számotokra ismeretlen programozási nyelvvel fogtok megismerkedni, ez a **Small Basic**. Ezt a programozási környezetet a Microsoft cég kimondottan a programozást tanuló fiatalok számára fejlesztette ki. Az alapvető programozási tudás megszerzése mellett lehetőség van többek között a technográfika gyakorlására (ebben hasonlít kicsit a többek által ismert Comenius Logo-ra), különféle alakzatok rajzolására is. A szoftver nem magyar nyelvű, hanem angol, de ez ne ijesszen meg senkit, hiszen hamar rá lehet jönni egy kis próbálgatással arra, hogyan kell használni. Reméljük, hogy ez a rövid dokumentum is segít a tanulásban.

Munka a SmallBasic programozási környezettel

A telepítés után indítsd el a Small Basic programot a Windows Start menüjében a SmallBasic indító ikonjával!

Az ikon: 

Az alábbi ablakot látod majd. A számokkal a munkához szükséges fontos elemeket jelöltük meg.



1. Ez egy szövegszerkesztő felülete, ide kell írni sorról sorra a program utasításait.
2. Erre a gombra (Run) kattintva tudod elindítani az általad megírt programot, ilyenkor sorban végrehajtja a számítógép a program utasításait.
3. Ide kattintva (Save) tudod elmenteni a programodat, hogy legközelebb is megnézhesd, kipróbálhasd vagy megváltoztathasd.

Nézzük az alapvető utasításokat:

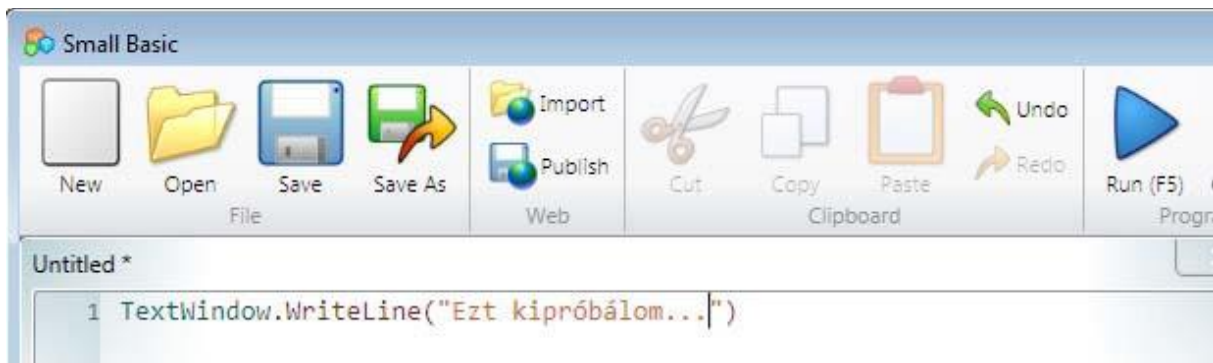
Emlékeztetünk arra, hogy a tanuláshoz kitartás is kell. Elsőre lehet, hogy túl sok lesz az információ, de igyekszünk sok példát adni, és ha végig követed ezt az írást, a példaprogramokat pedig kipróbálsz a saját gépeden, akkor biztos a siker!

Képernyőre írás

Gépelj be a következő szöveget:

```
TextWindow.WriteLine("Ezt kipróbálom...")
```

úgy, ahogy a következő képen látod. Ez már egy egyszerű program, ami kiírja a zárójelben idézőjelek közé írt szöveget. Természetesen más szöveget is kiíráthatsz. Például próbáld ki a saját neveddel!



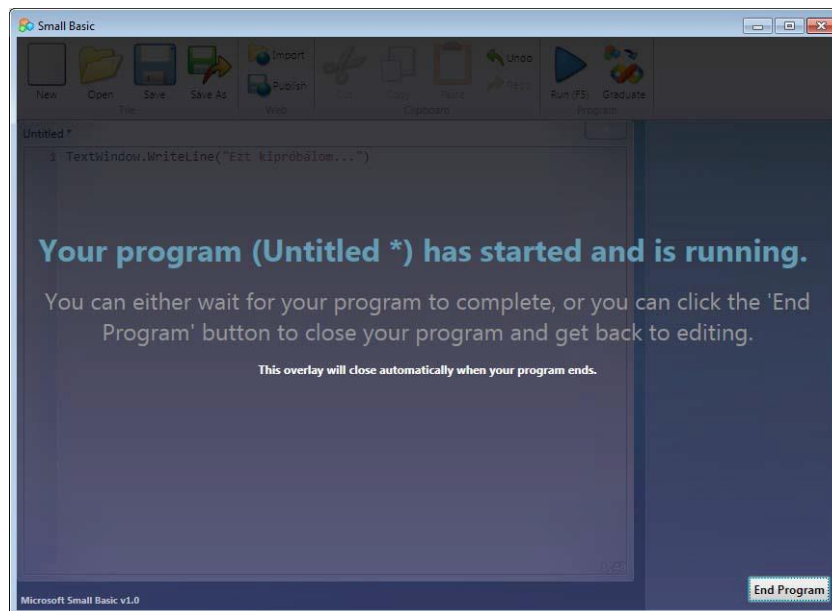
Biztos észrevetted, hogy amikor beírtad azt, hogy **TextWindow** és lenyomtad a pontot, feljárni egy menü, hogy milyen utasításokból választhatunk, ezek között találjuk a **WriteLine**-t is. A menüben navigálhatunk a kurzormozgató Le-Fel nyilakkal, az egér görgető gombjával, illetve ahogy gépelünk, úgy is odagördül ahhoz a lehetőséghez, ami legjobban hasonlít ahhoz, amit begépelünk.

Ha a menüből választod ki az utasítás nevét, amit használsz, azt a tabulátor vagy az Enter billentyűvel teheted meg.



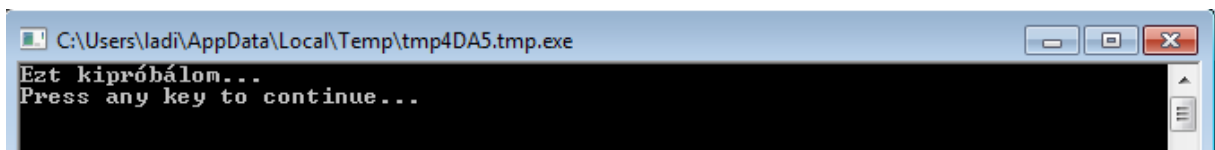
Már a **TextWindow** begépelése közben is egy hasonló menü jelenik meg, itt is ez előbb leírt módon választhatunk, hogy mivel kezdjük az utasítást.

Most futassuk le a programot! Kattints a **Run** menüpontra, vagy nyomd meg az F5 funkcióbillentyűt!



A fenti képernyő röviden arra utal, hogy a program elindult, és amíg nem fejeződik be a program futása, addig ez a képernyő eltakarja azt a felületet, ahol programozni tudunk.

A program eredménye az alábbi képen látható. Ez egy ablaknak a része, amit a programunk minden futáskor létrehoz. Az első sorban látjuk az utasításunk eredményét: egy szöveg kiírását. A 2. és egyben utolsó sor: *Press any key to continue...* annyit jelent, hogy nyomj le bármilyen gombot a folytatáshoz, ami itt a program befejezését jelenti, és visszatérést az előző felületre.



Ha valami hiba van

Ha a **Sorry, we found some errors...** feliratot látjuk, azt jelenti, hogy valamit elrontottunk, vagyis hibásan gépeltünk be, például lefelejtettünk egy zárójelet vagy idézőjelet, de az is lehet, hogy túl sok van belőlük, esetleg kifelejtettünk egy betűt valamelyik utasításból, stb.

Rövid magyarázat a `TextWindow.WriteLine("Ezt kipróbálok...")` utasításhoz. Az utasítások több részből állnak, az első rész, a **TextWindow** adja meg, hogy a szöveges ablakot szeretnénk használni. A pont utáni rész, adja meg, hogy milyen utasítást szeretnénk ezen az ablakon végrehajtani. A **WriteLine** képernyőre írást jelent. A kiírás után új sor elejére teszi a kurzort.

✍ Próbálj több szöveget is kiírni, egyiket a **TextWindow.WriteLine**, a másikat a hozzá teljesen hasonló **TextWindow.Write** utasítással, és próbáld megfejtetni, mi a különbség a két utasítás között!



Mentsük el a programot, nehogy elveszzen valamilyen üzemzavar következtében! Kattints a **Save** gombra, add meg a mentés helyét ugyanúgy, mintha például a Jegyzettömbbel mentenél egy szöveget! Az elkészült fájlok kiterjesztése **SB** lesz.

Munka a számokkal

Nézzük a 4 alapl művelet jelét:

összeadás: +
kivonás: -
szorzás: *
osztás: /

Természetesen használhatunk zárójeleket is. Például írassuk ki a programmal a 7 és a 9 számok átlagát!

```
TextWindow.WriteLine((7+9)/2)
```

Változók használata

A programokban a használt adatokat eltárolhatjuk a memóriában a későbbi felhasználáshoz. Az ilyen tárolókat **változóknak** nevezik, mert a tartalmuk a program működése alatt bármikor megváltoztatható. Amikor szükség van egy tárolóra, egy újabb adat eltárolásához, akkor egyszerűen kezdj el használni egy újabb változót!

A változó nevét Te választod meg. A név az angol ábécé betűiből és számjegyekből állhat, de nem kezdődhet számjeggyel, illetve nem lehet benne szóköz sem.

Értékkadás

Nézzük, hogyan lehet az előbbi két szám átlagát kiszámíttatni változók használatával! A program a következő:

```
a=7  
b=9  
TextWindow.WriteLine((a+b)/2)
```

A „=” jel itt nem egyenlőséget jelöl, hanem azt jelenti, hogy a jobb oldalán lévő adatot bele kell tenni a bal oldalán lévő változóba. Ezt a műveletet nevezik **értékkadásnak**. Ha felolvassuk például az `a=7` utasítást, akkor azt mondjuk, hogy „a legyen egyenlő 7-tel”.

Adatok beolvasása a billentyűzetről

Az igazi az lenne, ha a programunknak működése közben adhatnánk meg azt, hogy mely számok átlagát számítsa ki. Ezt könnyen megtehetjük a `TextWindow.Read()` utasítással. Amikor a `TextWindow.Read()` végrehajtására kerül sor, akkor a program addig várakozik, amíg be nem írunk egy adatot a billentyűzeten (Enterrel zárva). Ezután a beírt adatot beleteszi az általunk megadott változóba. Ezt a műveletet **beolvasásnak** nevezzük. Az alábbi program beolvas két számot és kiszámítja az átlagukat.

```
a=TextWindow.Read()  
b=TextWindow.Read()  
TextWindow.WriteLine((a+b)/2)
```

Ha az eredmény elé magyarázó szöveget is ki akarunk írni, akkor a kiírandó dolgokat + jellel kell összefűzni.

Fontos: a változókat soha ne tegyük idézőjelek közé! Az előbbi eredményt most így írjuk ki:

```
TextWindow.WriteLine(„Átlag: ”+(a+b)/2)
```

Próbáld ki a fenti programot, ha elindult, akkor gépeld be neki előbb az egyik, majd a másik számot! Ne feledd: minden adat beírását Enterrel kell befejezni!

Nézzünk egy másik példát! Írjunk programot, ami bekér 2 számot, majd egy új változót alkalmazva kiszámítja és kiírja a számok összegét és hányadosát!

```
TextWindow.Write(„Kérem az egyik számot! ")
a=TextWindow.Read()
TextWindow.Write(„Kérem a másik számot! ")
b=TextWindow.Read()
c=a+b
TextWindow.WriteLine("Összeg: "+c)
TextWindow.WriteLine("Hányados: "+a/b)
```

Megjegyzés

Látjuk, hogy a + jel összeadást is és összefűzést is jelent. Ezért ha összeget akarsz kiíratni, akkor tedd zárójelbe.
Például:

```
TextWindow.WriteLine("Összeg: "+(a+b) )
```

Szövegek elhelyezése a változókbán

A változókbán a számokon kívül szöveges adatok is eltárolhatók. Természetesen az ilyen adatokkal nem végezhetők matematikai műveletek, de értéküket ugyanúgy kiíratjuk a képernyőre, mint a számokat tároló változókét.

Az alábbi példában az `udv` nevű változó értékadással, míg a `nev` nevű változó beolvasással kap értéket! Figyeld meg, hogy az értékadó utasításban is idézőjelek közé kell tenni a szövegeket.

```
udv="Hello "
nev=TextWindow.Read()
TextWindow.WriteLine(udv + name)
```

Kiegészítés matematikai műveletekhez

Ha bizonyos matematikai műveleteket szeretnél használni, akkor azokat a `math` kulcsszóval kell kezdeni és utána egy pontot téve jöhet a művelet neve és zárójelbe pedig a szükséges paraméterek. Javaslom próbáld ki őket, mielőtt konkrét feladatban felhasználod fel!

Véletlen számok generálása

`math.GetRandomNumber(N)`: Véletlen számot generál 1 és a megadott szám között, így:
`1<=math.GetRandomNumber(N)<=N`

Például így használhatod: `x=math.GetRandomNumber(10)`

Osztás során keletkező maradék meghatározása

`Math.Remainder(x,y)` Megadja az `x/y` hányados maradékát.

Például így használhatod `TextWindow.WriteLine(Math.Remainder(7,2))`

Számok kerekítése

`Math.Floor(number)`: lefelé egészre kerekít, pl.: 32.533 -ból 32 lesz

pl.: `TextWindow.WriteLine(Math.Floor(32.533))`

`Math.Ceiling(number)` felfelé egészre kerekít, pl.: 32.233 ból 33 lesz

pl.: `TextWindow.WriteLine(Math.Ceiling(32.533))`

`Math.Round(szám)`, a matematika szabályai szerint kerekít, pl.: 32.233 -ból 32.0 , 32.566-ból 33 lesz

```
p1.: TextWindow.WriteLine(Math.Round(32.533))
```

Melyik nagyobb? Melyik kisebb?

Math.Max(szám1, szám2) a két szám közül a nagyobbat adja vissza

```
p1.: TextWindow.WriteLine(Math.Max(12,23))
```

Math.Min(szám1, szám2) a két szám közül a kisebbet adja vissza

```
p1.: TextWindow.WriteLine(Math.Min(12,23))
```

Rajzolás –Grafika

A Small Basicben van lehetőség a karakteres ablak mellett grafikus ablak használatára is, ahol lehetőség nyílik rajzok készítésére a programból egyszerű alakzatok felhasználásával. Ahhoz, hogy tudjunk rajzolni elő kell készíteni a "terepet". A következő utasítások lesznek szükségesek illetve hasznosak:

```
GraphicsWindow.Show()
```

Megjeleníti a grafikus ablakot.

```
GraphicsWindow.BackgroundColor = "orange"
```

Az ablak háttérszínét állítja be, itt éppen narancssárgára.

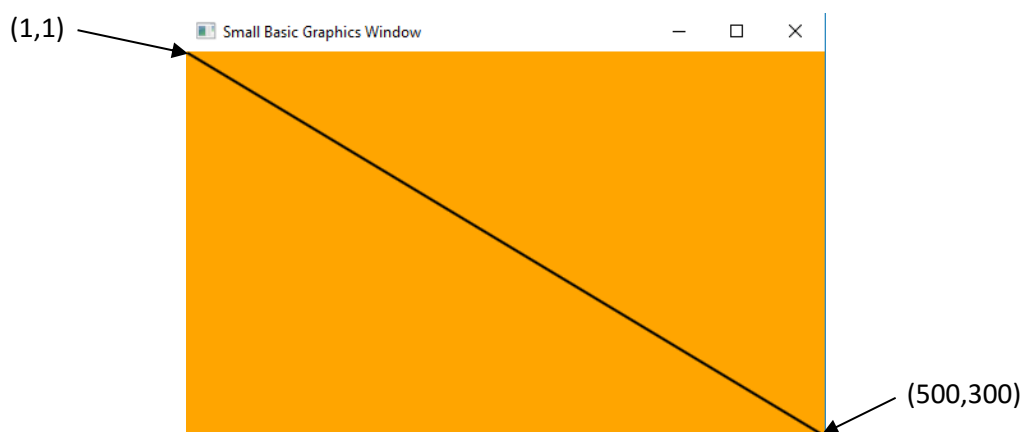
```
GraphicsWindow.Width = 500
```

Az ablak szélességét 500 pixelre állítja

```
GraphicsWindow.Height = 300
```

Az ablak magasságát 300 pixel szélesre állítja.

Az alakzatok rajzolása koordináták segítségével történik, a bal felső sarok az (1,1) a jobb alsó pedig (500,300), ha a fenti méretű ablakot rajzoljuk. A képen berajzoltuk az ablak átlóját.



Rajzó utasítások:

```
GraphicsWindow.DrawLine(x1,y1,x2,y2)
```

Vonalat rajzol, úgy, hogy összeköti az (x1,y1) pontot az (x2,y2) ponttal. Ezzel készült a fenti átló is.

```
GraphicsWindow.DrawRectangle(x, y, szélesség, magasság)
```

Téglalapot rajzol a megadott helyre, a téglalap bal felső sarka kerül az x,y koordinátára a képernyőn.

```
GraphicsWindow.FillRectangle(x, y, szélesség, magasság)
```

Az előzőhöz hasonlóan működik, csak kitölti a téglalapot.

GraphicsWindow.DrawEllipse(x, y, szélesség, magasság)

Ellipszist rajzol ahol, az ellipszis köré rajzolható téglalap bal felső sarkának a helyét adja meg az első két koordináta, a 3. és 4. pedig az ellipszis méreteit (szélesség, magasság) adják meg.

GraphicsWindow.FillEllipse(10, 10, 180, 90)

Kitöltött ellipszist rajzol, az eddigiekből következik a működése.

Színek és vonalak beállítása:

GraphicsWindow.PenColor = "Blue"

Tintaszín megadása vonalas alakzatokhoz

GraphicsWindow.PenWidth = "n"

Vonalas alakzatoknál a vonalvastagság n pixel, ahol $n \geq 1$.

GraphicsWindow.BrushColor = "Red"

Kitöltött alakzatok színét állítja be, itt éppen pirosra.

GraphicsWindow.GetRandomColor()

Ha a színeket véletlen szerűen szeretnénk beállítani, akkor használjuk, pl így:

GraphicsWindow.PenColor = GraphicsWindow.GetRandomColor()