

Makrók írása Excelben

Ha nem vezérlőhöz rendelünk makrót és munkafüzetünk minden lapján egyformán szeretnénk tudni futtatni, akkor egy **modul** elemhez kell kapcsolni. A VBA ablakban *Insert* menüből *Modul*. Rögzítésnél már láttuk a makró kezdő – és zárósora: **Sub** makrónév () ... **End Sub**
A VBA NEM TESZ KÜLÖNBSÉGET KIS BETŰ ÉS NAGY BETŰ KÖZÖTT! Egy sorba több utasítás is írható egymástól kettősponttal elválasztva.

VBA Hivatkozások az Excel objektumaira (Munkafüzet, Munkalap, tartomány, cella)

Az Excelbeli elnevezések	Programkódbeli megfelelő
1. A B3 cella	Cells(3,2)
2. A C4:G6 tartomány	Range(Cells(4,3),Cells(6,7)),Range("c4:g6")
3. A B oszlop	Columns(2)
4. A H, I, J oszlopokból álló tartomány	Range(Columns(8),Columns(10))
5. A 2. sor	Rows(2)
6. A 13.-tól a 16. sorig tartó tartomány	Range(Rows(13),Rows(16))
7. A munkalap összes cellája	Cells
8. Meghatározott munkalap cellái	Sheets(2).Cells
9. Az éppen aktuális cella (ha tartomány van kijelölve, aktuális cella akkor is csak egy van, aminek a tartalma a szerkesztőlécen látszik)	ActiveCell
10. Az éppen kijelölt objektum (cella, tartomány, rajz, diagramm)	Selection

Excel objektumok leggyakrabban használt tulajdonságai

(hivatkozás a pont segítségével: <objektum neve>.<tulajdonság>)

– **Value**: a cella értéke, ha szöveg, akkor idézőjelek közé kell tenni. Ez az alapértelmezett tulajdonság, ezért ha nem írunk tulajdonságot, akkor ezt tudjuk megadni. A következő két sor eredménye ugyan az: az A1-es cellába az Alma felirat kerül:

`Cells(1,1).Value = "Alma" vagy: Cells(1,1) = "Alma"`

– **Interior.Color**: a cella háttérszíne szövegesen megadva. Az angol színnevezéseket lehet használni, csak elé kell írni egybe vele, hogy vb (*csak az alapszínek: fekete- black, fehér- white, sárga- yellow, piros-red, kék-blue és zöld-green*): `Cells(1,1).Interior.Color = vbRed`

– **Interior.ColorIndex**: a cella háttérszíne számmal megadva. A VB 0-tól 56-ig számozza a színeket. Az A1 cella háttérszínét pirosra lehet állítani így is (*a 3-as a piros szín kódjával*):

`Cells(1,1).Interior.ColorIndex = 3`

– **Font**: a cella betűtípusa. Újabb ponttal elválasztva lehet megadni stílust, típust, színt a fent ismertetett módokon. A következő táblázatban példákat találunk az A1 cella betűformázásaira:

Betűtípus Arial-ra állítása	<code>Cells(1,1).Font.Name = "Arial"</code>
Betűszín pirosra állítása	<code>Cells(1,1).Font.Color = vbRed</code>
vagy	<code>Cells(1,1).Font.ColorIndex = 3</code>
Betűméret 12-esre állítása	<code>Cells(1,1).Font.Size = 12</code>
Félkövér betűre állítás	<code>Cells(1,1).Font.Bold = True</code>
Dőlt betűre állítás	<code>Cells(1,1).Font.Italic = True</code>

- **Row, Column**: a cella sor és oszlopszáma. Csak lekérdezhető, megváltoztatni nem lehet!

- **Formula** (elhagyható) idézőjelek között megadhatjuk a képletet, amit a cellába írunk. - - ----

FormulaR1C1 (elhagyható): a képletben a sor és oszlopszámok megadásával hivatkozhatunk a cellákra. A sor számát **R**, az oszlop számát pedig **C** betű után kell írni. Pl. a B1-es cella az R1C2. Ez a hivatkozás abszolút! Relatív hivatkozásnál szögletes zárójelben a sorszám helyett azt kell számmal megadnunk, hogy hány cellát kell elmozdulnunk. Pl. az eggyel fel és kettővel jobbra lévő cellára hivatkozhatunk az R[-1]C[2] kóddal.

Az Excel beépített függvény a képletben, csak akkor ad jó eredményt, ha az **angol** függvényneveket és paraméter megadási sorrendet használjuk. Ehhez azonban elkerülhetetlen az angol függvénynevek ismerete. Vigyázat! A képletben az argumentumokat elválasztó ; (**pontosvessző**) helyett (**vessző**)-t kell használni. A tartomány kezdő és végpontja között a : (kettőspont) változatlan.

Pl. – `Cells(1, 2) = "=POWER(R1C1, 2)"` – Hatványfüggvény angolul

VB változótípusok (definiálhatók Dim <változó neve> As <típus neve>)

Típusnév	mit tárol
String	szöveg
Integer	egész szám (32768-ig)
Single, Double	valós szám
Long	hosszú egész
Date	dátum
Boolean	logikai(igaz/hamis)
Variant	általános – ez az alapértelmezett!

Programozási struktúrák

Elágazás: feltételhez kötjük, hogy melyik utasítások hajtódjanak végre. Szintakszisa:

If feltétel Then

utasítások, amelyek akkor futnak le, ha a feltétel igaz

Else

utasítások, amelyek akkor futnak le, ha a feltétel hamis

End If

Ebből az **If** és az **End If** kötelező, az **Else** ág elhagyható.

Ciklus: ugyanazt az utasítást szeretnénk többször megismételteni a programmal. A megfelelő kulcsszavak közé írt, ismétlődő utasítást **ciklusmagnak** nevezzük.

Leszámláló (iterációs) ciklus: ha előre tudjuk, hogy **pontosan hányszor** szeretnénk futtatni az utasítást. Szintakszisa:

For ciklusváltozó = kezd_érték To végérték Step lépésköz

ciklusmag

Next

A **ciklusváltozó** bármilyen számváltozó lehet (leggyakrabban az i,j, k betűk valamelyikét használjuk). Első lefutáskor a változó a **kezdőértéket** veszi fel, aztán minden újabb lefutáskor **lépésközzel** változik az értéke, amíg **meg nem haladja a végértéket**. Amikor a változó értéke a végértéknél nagyobb, akkor a ciklusmag már nem fut le. (Ha nem adunk meg lépésközt, akkor annak értéke: +1).

Feltételes ciklusok, ha nem tudjuk előre, hogy hányszor kell lefutnia az utasításoknak, hanem egy **feltételhez** akarjuk **kötni** a ciklus végét. VB-ben 4 féle feltételes ciklus létezik, ezek közül kettőt fogunk megismerni. A ciklus elejét a **Do**, végét a **Loop** kulcsszó jelzi.

Elöl tesztelés ciklus: a feltételt a ciklus elején a **Do While** kulcsszavak után adjuk meg. Ebben az esetben a **feltétel** teljesülése esetén történik az ismétlés. (Előfordulhat, hogy már az első lefutáskor sem teljesül a feltétel, ilyenkor **egyszer sem fut le** a ciklus belsejében lévő utasítássorozat.)

Hátul tesztelés ciklusnál a feltételt a ciklus végén a **Loop Until** kulcsszavak után adjuk meg. Ebben az esetben mindaddig történik az ismétlés, amíg a **feltétel hamis (azaz amíg a feltétel igaz nem lesz!)**. A hátul tesztelés ciklus mindig **lefut legalább egyszer**. A lehetséges szintaxisok:

Do While <i>feltétel</i> <i>ciklusmag</i> Loop	Do <i>ciklusmag</i> Loop Until <i>feltétel</i>
--	--

A **leszámláló ciklus átírható feltételes ciklussá**. Ez esetben létre kell hoznunk egy változót a ciklus előtt, és **kezdőértéket** kell neki adnunk, majd a ciklusmagban a változó értékének megfelelő növekedését, vagy csökkenését nekünk kell biztosítani. A feltételben azt kell megfogalmazni, hogy ha a változó meghaladta a végértéket, akkor legyen vége az ismétlésnek. Pl. első 10 szám összege:

<pre> ossz = 0 For i = 1 To 10 ossz = ossz + i Next </pre>	<pre> ossz = 0 : i = 1 Do While i <= 10 ossz = ossz + i i = i + 1 Loop </pre>	<pre> ossz = 0 : i = 1 Do ossz = ossz + i i = i + 1 Loop Until i > 10 </pre>
---	---	--