

PC World

2000. június

Szolgáltatások az interneten 1.

1. Űrlapok

Készítette: Móricz Attila

www.moricznet.hu

Előszó

A honfoglalással foglalkozó füzetünkkel sikerült felbolygatunk az állóvizet, és nagyon sokan nekiálltak életük első honlapjának elkészítéséhez. Sokan jelezték, hogy elkészült a honlapjuk, azt sikeresen ki is tették az internetre, de az ismertetett szolgáltatások használatakor gondjaik adódtak.

Valóban a leírások elsődleges célja az volt, hogy felkeltsük az önök figyelmét az internet iránt, hogy megmutassuk, viszonylag könnyen kikerülhetünk a webre, és mindez semmibe sem kerül (*az internetkapcsolat árát leszámítva*). Azonban a kivitelezés nem olyan nagyon egyszerű, ráadásul sokaknak az angol nyelvű oldalak is gondot okoznak.

Ezért a most induló sorozatunkban sokkal részletesebben bemutatjuk az egyes feladatok végrehajtását, bár a munka részletes kidolgozása így is önökre vár. Viszont a weblapjaik egyes részleteihez felhasználhatják az itt szerzett ismereteket, így gazdagítva oldalaik tartalmát. Érdemes lesz visszalapozni a *Visual Basic* programnyelvről szóló cikkekhez is, ahol már szintén kapcsolódunk a HTML-nyelvhez.

Először az űrlapok tartalmával foglalkozunk, amelyekhez kis példaprogramokat is találunk. A HTML-kód Jegyzetombbe való beírásával és HTML-kiterjesztéssel történő elmentésével az űrlapok a legtöbb esetben már a helyi fájlrendszerünkben is kipróbálhatók (*élő internetkapcsolat mellett*), de másoknál a fájl előbb FTP-vel fel kell másolnunk a webhelyszolgáltatónkhoz.

Részletesen ismertetjük az *Extra* webhelyen regisztrált felhasználók részére biztosított űrlapkészítés fortélyait, előnyeit és hátrányait. Szinte minden esetben a weblapunk tartalmaz egy-egy VBScript programot, amely biztosítja a weblapunk dinamikus változásait, az egyes szolgáltatások jobb használatát.

Utána egy *eGroups* levelezőlista létrehozását nézzük végig lépésenként, amely a temérdek szolgáltatásával méltán érdemli ki figyelmünket. Természetesen ez is ingyenes, és szerintem csak előnye van. Ezért mindenkinek javaslom, hogy már most kezdje el tervezgetni, hogy a weblapján melyik levelezőlistáját fogja jobban hirdetni, a bélyeggyűjtőket vagy a WAP-címgyűjtőket....:-)

Folytatni fogjuk mindegyik témát, mert mindegyik szolgáltatás olyan gazdag tartalommal szolgál, hogy nem lehet azokat egyszerre feldolgozni.

Móricz Attila

www.moricznet.hu

Tartalomjegyzék

Az űrlapok tartalma.....	4
Az űrlapok felhasználási módja.....	4
Űrlapelemek hiperhivatkozások helyett.....	4
Információgyűjtés űrlapokon.....	5
Információmorzsák e-mailben vagy SMS-ben.....	5
A szükséges programok.....	6
Egysoros szövegmező	7
Gombok fajtái	9
Űrlapküldés (Extra).....	13
A kitöltendő mezők	13
Válaszlevél a feladónak.....	16
Válaszoldalak a feladónak.....	17
A hivatkozott oldalak elhelyezkedése.....	17
Hibás e-mail-címnél megjelenő válaszoldal	18
Tovább lépés a válaszoldalról jó adatok esetén	19
Levelezőlista felállítása (eGroups).....	20
A lista megtervezése.....	20
Hírlevél véleményekkel.....	20
Figyelmeztetők a közös programokra	20
A szükséges dokumentumok elkészítése	21
Dokumentum a feliratkozáshoz	21
A fájlok feltöltése	23
A listakezelés tesztelése	24
Feliratkozás a listára.....	24
A listatulajdonos levélforgalma.....	24
A lista lemondása	24
Tagok felvétele a listára	25
Csoportos felvétel.....	25
Feliratkozás weblapról.....	25
A levelezőlista hirdetése a saját weblapjainkon.....	26
Olvasók	28

Az űrlapok tartalma

Mielőtt nekiállnánk űrlapokat készíteni, gyűjtsük össze, milyen feladatokra használhatjuk az űrlapelemeket. A legalapvetőbb felhasználási mód az, amikor a felhasználónak kérdéseket teszünk fel, és a válaszait begyűjtjük valamilyen célból, de használhatjuk az űrlapelemeket navigáláshoz is.

Ez utóbbi az egyszerűbb, mert csak egy eseménykezelő programot kell írunk, amely egy listaelem vagy egy gomb kiválasztása esetén arra utasítja a webböngészőt, hogy az általunk kiválasztott oldalt töltsse be.

Ha már a felhasználói válaszokat szeretnénk begyűjteni, de nem akarunk megtanulni még CGI-nyelven programozni, akkor kereshetünk egy (ingyenes) szolgáltatást, amely elvégzi helyettünk a levélküldést. Ezekhez nekünk csak a címet kell tudnunk, és a kötött nevű mezőknek megfelelő értéket kell adnunk.

Az űrlapok felhasználási módja

Ha sikerül hasznos felhasználási lehetőségeket találunk az űrlapelemekhez, akkor viszonylag könnyen el tudunk indulni a felfedezés útján, mert a helyi fájlrendszerünkben elkezdhetjük a kísérletezést egyszerű weblapokon. Nem kell hozzá sem webkiszolgáló, sem hálózat. Már egy átlagos képességű weblapszerkesztő program is megfelel a célnak. Ha semmi sincs, akkor egy jó sűgőfájl birtokában egy karakteres szövegszerkesztő is megteszi. A sűgő természetesen a HTML nyelv ismertetését tartalmazza.

Űrlapelemek hiperhivatkozások helyett

A honlapunk elkészítésekor először általában a legalapvetőbb weblapelemeket használjuk, de egy idő után azokkal már nehézkessé válik a munka. Ha ugyanis sok weblapunk van, akkor azok átrendezése, felépítésük megváltoztatása olyan sok hiperhivatkozás változását igényli, amely már túlzott lenne.

Persze erre is van megoldás. A FrontPage webszerkesztő program figyel a hiperhivatkozások változásaira, azokat az összes lapunkon szinkronban tartja. Ez persze nem mindenkinél található meg, így kénytelen a felhasználó figyelni a változásokra, ami sok nem létező fájlra mutató linket eredményezhet.

Különösen keretes (frame) weblapoknál hasznos, ha az oldalon nem helyezzünk el egyetlen hiperhivatkozást sem, mert így a szörföző felhasználó nem tudja azt külön ablakban megnyitni. Így csak egy ablakban tud böngészni, és sok kellemetlenségtől kímélhetjük meg magunkat és a felhasználót.

Ehhez a hiperhivatkozások helyett legördülő listákat, gombokat kell elhelyeznünk a weblapon, és valamelyik eseménykezelőnél megadott saját programunkkal kell azokat feldolgoznunk. A programunk lehet egy külön fájlban is, így újabb oldalak készítésekor nem kell minden oldalunkat módosítani, csak a programot kell kibővíteni. A változás azonnal minden oldalon megjelenik. Ebben az esetben a teljes feladatmegoldás a mi kezünkben van.

Információgyűjtés űrlapokon

Ha már az e-mail-címünkre szeretnénk postáztatni a felhasználó által megadott adatokat, akkor ahhoz szükségünk van egy szerveroldali programra. Bizonyos webhelyszolgáltatóknál lehetőség van CGI-programok elhelyezésére is, így az űrlapok, hiperhivatkozások feldolgozása történhet egy általunk megírt programmal is. Így viszont a CGI-vel is meg kell ismerkednünk.

Sok helyen biztosítanak a szolgáltatók vagy más felhasználók előre elkészített programokat. Ebben az esetben nekünk nincs más dolgunk, mint a megfelelő nevű mezőket kitölteni, és az ő programjuk az űrlapot postázza nekünk.

Erre több ingyenes szolgáltatást is megnézünk a további fejezetekben. Használatuk egyszerű, különösebb szakértelmet nem igényel, így kezdők is bátran próbálkozhatnak vele.

Információmorzsa e-mailben vagy SMS-ben

Gyűjthetünk adatokat csepegtetve is. Ha az űrlapelemeknél szerzett információkat a felhasználó gépén létrehozható *cookie*-fájlokban tároljuk, akkor elég minden oldalon a tartalom között csak egy-egy adatot bekérni, beíratni a felhasználóval, és idővel sok információ összegyűlhet. Ezeket aztán felhasználhatjuk arra, hogy a weblapjainkat leginkább a felhasználói igényekhez igazítsuk, ami a nézettséget növeli.

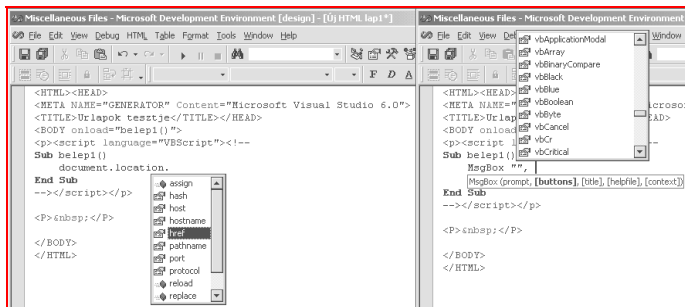
Akinek még olyan gondjai vannak, hogy az oldalát egyáltalán látogatják-e, annak lehet hasznos, ha a nyitóoldalon egy belépő gombot helyez el. Ha a felhasználó erre például *Belépés* gombra kattint, akkor azzal elküld egy e-mailt a weblap tulajdonosának. Ez egy olyan e-mail-cím is lehet, amelynek tartalmát egy kiszolgáló továbbítja SMS-ben a felhasználó mobiltelefonjára, aki azonnal értesül a jó hírről, valaki meglátogatta az oldalát.

Amikor pedig ez már idegesítően sokszor történik meg, akkor kikapcsolhatjuk, és átállíthatjuk valamilyen más, különleges információkat tartalmazó oldal figyelésére.

A szükséges programok

Akinél a Windows 98 vagy az IE 4 van telepítve, ott a FrontPage Express program megtalálható. A további változatok sajnos nem tartalmazzák ezt az alkalmazást, így ott valamilyen külön weblapszerkesztő program szükséges.

Ez a program bármilyen lehet, amelyiknél az űrlapelemeket be lehet szűrni, és amely biztosítja, hogy a különféle weblapelemeknél további paramétereket adjunk meg. Nem kell, hogy a program ezeket (például a stílusparancsokat) értelmezni tudja. Elég az is, ha a webböngésző programunkban jelenik meg helyesen a weblap. Természetesen, ha olyan programunk van, amelyikben a forrásfájl is látjuk és a megjelenési nézetben is szerkeszthetünk, akkor az még jobb. Az viszont nagy segítség, ha a programjaink beírásakor az objektumok tulajdonságait kis helyi menükben tudjuk tallózni, mint amire jó példa számos Microsoft fejlesztőkörnyezet. A *Microsoft Development Environment 6.0* program az Office 2000 telepítésekor kerülhet fel a gépünkre.



1. ábra. A beírásakor megjelenő helyi menü vagy sűgő nagy segítség

Az ilyen beírás előnye, hogy nem lehet eltéveszteni a hivatkozásokat, mert nem tudunk a listából olyan tulajdonságot kiválasztani, amire később hibaüzenetet kapnánk. Ugyanehhez a programhoz tartoznak azok a sűgők is, amelyek a HTML, a VBScript 5, a JScript 5 és az ASP teljes leírását tartalmazzák.

A továbbiakban értelemszerűen járunk el. A forrásprogramot egy arra alkalmas nézetben írjuk be, míg az elemek tulajdonságait a tervező nézetben kell keresnünk. A weblapok forrása a Jegyzetömbben is módosítható.

Egysoros szövegmező

Hozzunk létre egy új weblapot, vagy írjuk be az ahhoz szükséges minimális HTML-elemeket a szövegszerkesztőnkbe, majd szúrjunk be egymás mellé két szövegmezőt és egy gombot. A programok általában szaggatott vonallal jelzik, hogy egy űrlapot (FORM) is beszúrtak. Erre nincs szükségünk, így azt ki is törölhetjük. Utána egészítsük ki úgy a fájlt, hogy az tartalmazza az alábbi listában olvasható összes paramétert, és a VBScript programunkat.

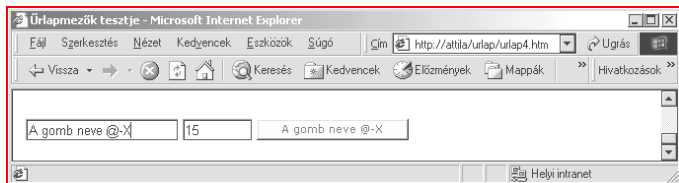
```
<html>
<body onload="GnT1.focus()">
  <script language="VBScript"><!--
    Sub Gomb1_onClick()
      Gomb1.value = "[" + GnT1.value + "]"
    End Sub

    Sub GnT1_onKeyUp()
      szaml.value = Len(GnT1.value)
      Gomb1.value = GnT1.value
    End Sub
  --></script>
<input type="text" size="20" name="GnT1" value="A
  gomb neve">
<input type="text" size="6" name="szaml" value="0"
  align="right">
<input type="button" name="Gomb1" value="(Klikk
  ide!)" style="font: 8pt 'Verdana'; background:
  yellow; color: red">
</body>
</html>
```

Ezen a weblapon az első szövegmezőben lesz egy kezdőszöveg, és a weblap megjelenésekor már a kurzor is ott fog villogni (*lásd body onload=...*). Ha beírunk további karaktereket ebbe a szövegmezőbe, akkor az megjelenik a gomb felirataként is. Közben a beírt szöveg hosszát is folyamatosan kiírjuk a második szövegmezőbe. A gomb megnyomásával is a gomb feliratán változtatunk.

Mindezeket a **GnT1** szövegmezőnél meghatározott **onKeyUp()** esemény kezelésével érjük el, amely egy billentyű felengedésekor következik be. Tehát

amikor beírunk vagy törölünk egy karaktert, akkor a nevezett szubrutinra adódik át a vezérlés. Ott beírjuk a **szaml1** nevű szövegmezőbe az előzőben lévő karakterlánc hosszát, majd a szöveget a gomb feliratának is beállítjuk



2. ábra. Eseménykezelés néhány űrlapelemnél

Érdemes összevetni a gomb megjelenését is a forráslistával. A **Style=** paraméterben pontosvesszővel elválasztva felsorolhatunk számos stíuselemet, amelyekkel tetszőlegesen megformázhatjuk bármelyik HTML-elemet. Itt a gomb piros színű 8 pontos Verdana betűtípussal jelenik meg sárga háttérrel.

Érdemes megnézni egy leírásban, hogy milyen további stíuselemek vannak, és azokat alkalmazni más elemeknél is, különösen a kérdőíveknél. Így az űrlapjaink nem lesznek egyhangúak, mert olyan megjelenést, stílust kölcsönzünk nekik, amely szokatlan, így figyelemfelkeltő is egyben.

Az itt bemutatott példán begyakorolhatjuk azt, hogyan kell a szövegmezőhöz különböző eseményeket rendelni, és hogyan módosíthatjuk annak megjelenését. Az eseménykezelő programrészben az ellenőrzésünk alatt tarthatjuk a beírt szöveget, számot, akár minden egyes karakter leütése után is.

Mind az INPUT nevű HTML-elemnél, mind a többinél sokkal több paraméter megadható, mint amennyit itt megemlítettünk. Itt csak az a cél, hogy önök egy jól működő, kipróbálható példát lássanak, azt megértsék, és annak működése esetén sikerélményük legyen. A többi paraméter megadása is hasonlóan történik, így a közölt példák mintának is tekinthetők. A stíuselemek is minden HTML-elemnél alkalmazhatók. Egyszerre több eseménykezelést is alkalmazhatunk, csak ne bonyolítsuk túl, mert akkor magunk sem fogjuk átlátni az egészet, abból pedig csak keveredés lesz. Több eseménykezelés esetén érdemes a programokat külön VBS- vagy JS-kiterjesztésű fájlokba menteni.

Gombok fajtái

Az űrlapok, kérdőívek legfontosabb elemének a gombokat tartjuk, mert úgy érezzük, hogy azokra kattintva indítjuk el a folyamatot. Ez régebben így is volt, de mára már megváltozott a helyzet.

Ha egy kicsit belegondolunk, akkor rájöhetünk, hogy ez csak kérdőívek esetén van így, mert a weblapok közötti navigálásoknál akár egy legördülő listából történő választással is azonnal elindíthatjuk egy újabb weblap betöltését. Ehhez elég az egyik elemen bekövetkező eseményhez egy szubrutint rendelni, és a **window.top.location.href = "http://" + Gomb1.value** paranccsal már el is indíthatjuk az új oldal betöltését. Ebben a példában a „+ **GnT1.value**” jelentheti az előző HTML-fájlban látott szövegmező tartalmát is.

A következő kis példában már rá sem kell kattintani a gombra, elég csak az egérmutatót fölé mozgatni (*onMouseOver*), és már ugrik is. Ha viszont hiányzik a kezelőprogram, akkor hiába kattintunk a gombra, semmi nem fog történni! Nem a gomb indítja el a letöltést, hanem a megfelelő parancs! Ugyanígy az Enter lenyomása sem indít el semmit, mert ez nem egy űrlap, itt nincs alapértelmezett gomb vagy parancs. Viszont írhatunk arra is programot, hogy az Enter (#13) lenyomása egy gombra való kattintásnak feleljen meg.

```
<html>
<body onload="GnT1.focus()">
<script language="VBScript"><!--
    Sub Gomb1_onMouseOver()
        window.top.location.href = "http://" + Gomb1.value
    End Sub

    Sub GnT1_onKeyUp()
        Gomb1.value = "www." + GnT1.value + ".hu"
        If window.event.keyCode = 13 Then Gomb1_onMouseOver()
    End Sub
--></script>
<input type="text" size="20" name="GnT1" value="swi">
<input type="button" name="Gomb1" value="_">
</body>
</html>
```

Az előző kis példát egy karakteres szövegszerkesztőbe beírva, és HTML-fájlként elmentve, egy olyan űrlapot kapunk, ahol a szövegmezőben leütött Enter kiváltja a gombra kattintáshoz rendelt program végrehajtását.

Ha olyan űrlapot készítünk, amelyik FORM elemek között helyezkedik el, és egy interneten lévő program (általában CGI) fogja ezt feldolgozni, akkor a **Küldés** gombunk típusa nem **button** lesz, hanem **submit**. Ezt általában meg is kapjuk a szolgáltatótól a leírásban, így nem kell nekünk külön **Küldés** gombot készíteni, hanem elég csak annak feliratát megváltoztatni (*pl.: Mehet!*).

Például: **urlap-sms.html**

```
<html>
<body>
<FORM ACTION="http://w3.swi.hu/webm/cgi-bin/urlap.cgi"
  METHOD="post">
<INPUT TYPE="hidden" NAME="mailto"
  VALUE="0620NnnNnNn@sms.pgsm.hu">
<INPUT TYPE="text" NAME="felado" Value="nev@dom.hu">
<INPUT TYPE="text" NAME="Uzenet1" Value="Latogatas">
<INPUT TYPE="submit" Value="Mehet!">
</FORM>
</body>
</html>
```

A fenti weblapot elkészítve kapunk egy kis űrlapot, amelyik megjelenít egy szövegmezőt a feladó e-mail-címének beírásához, és egyet az üzenetnek. A feladó címet ki is hagyhatjuk, nem kötelező megadni. Csupán a feladat jellege szabja meg, hogy erre szükség van-e.

Egyedül a **mailto** nevű mező megadása kötelező, ahol a címzett szerepel. Ha csak azt hagyjuk meg, és e-mail-címnek megadunk például egy olyan címet, amely SMS-ként megérkezik a mobiltelefonunkra, akkor ezt jelzésnek is használhatjuk, bizonyos oldalainknál. Ha meghagyjuk az **uzenet1** mezőt (*más is lehet a neve*), akkor így a látogató küldhet nekünk SMS-t is.

Itt az űrlapot az **Action** paraméterben megadott címen lévő program fogja elküldeni a rejtett (*hidden*) mezőben megadott címre. A küldést a **submit** gomb (*Mehet! feliratú*) fogja elindítani. Ez a program ellenőrzi az e-mail-címek helyességét, és válaszdalt is megjelenít.

Ha jobban megnézzük ezt a kis forráslistát, akkor elvégezhetünk rajta néhány módosítást is. Például a **mailto** nevű rejtett mezőt szövegmezővé alakíthatjuk, ha a típusát megváltoztatjuk rejtettől szövegesre (*hidden – text*). Ebből viszont az is következik, hogy létrehozhatunk további rejtett mezőket is, amelyek tartalmát előre megadhatjuk. Úgy is lehet, hogy a weblap megjelenésekor a felhasználó környezetéről lekérhető információkat egy rejtett mezőbe tesszük, és a gombnak egy egészen más feliratot adunk.

```
<html>
<body onload="kezd()">
<script language="VBScript"><!--
Sub kezd()
    F2.Uzenet2.value="(" + navigator.appVersion + ") " + _
        window.top.location.href
End Sub
--></script>

<FORM ACTION="http://w3.swi.hu/webm/cgi-bin/urlap.cgi"
    METHOD="post" Name="F2">
<INPUT TYPE="text" NAME="mailto"
    VALUE="0620NnnNnNn@sms.pgsm.hu">
<INPUT TYPE="hidden" NAME="Uzenet2" Value="X">
<INPUT TYPE="text" NAME="Uzenet1" Value="(Üzenet)">
<INPUT TYPE="submit" Value="Mehet!">
</FORM>
</body>
</html>
```

Ezen az űrlapon megjelenik a feladó e-mail-címe, bár ebben az esetben nem kellene megadni. Utána szerepel egy rejtett mező, amelynek tartalmát a weblap megjelenítésekor mi állítjuk be a **kezd()** szubrutinban. Előbb zárójelben szerepel a böngésző verziója és neve, majd a weblap címe. A felhasználó erről nem tud semmit, mert ő csak a másik mezőt látja, ahová az üzenetet írja. A **Mehet!** gombbal pedig mindet egyszerre el is küldi.

Ha elrejtjük a **mailto** mezőt, kihagyjuk a látható szövegmezőt, akkor az űrlapból csak a **Mehet!** gomb marad.

Ebben az esetben viszont adjuk neki a „**Belépés a webhelyre**” feliratot egy üres oldalon, formázzuk meg jó nagyra a gombot, nagy felirattal, és amikor rákattint a felhasználó, akkor beléphet ugyan a webhelyünkre, de ezzel egy időben el is küldte nekünk SMS-ben vagy e-mailben a belépési pont címét és a böngészője verzióját. *(Ebben az esetben nem lép be a webhelyre, mert ez az űrlapküldő szolgáltatás generál egy lapot, ahol ráadásul „lebuktat” bennünket, mert felsorolja, hogy milyen adatokat küldött el...:-))*

További elküldhető adatokat biztosítanak még a **navigator**, a **window**, a **screen** vagy a **location** objektumok. Ezek tulajdonságait az *IE Platform SDK*-ban megnézhetjük, ahol sok érdekes adatot találunk *(de semmi titkosat :-)*.

A fenti példákat nem ijesztgetésül mutattam, hanem azért, hogy lássuk, hogy egy gomb megnyomása szélsőséges esetben már információadással is járhat. Ha pedig a másik oldalról nézzük a dolgot, akkor ez a módszer kezdő weblaptulajdonosoknál egyfajta azonnali visszajelző-rendszerként is működhet, mert percre pontosan láthatja az e-mailjeiben, hogy mikor léptek be az oldalára. Haladóknál ez a különleges tartalmú weblapok látogatóinak figyelésére szolgálhat. Megtudhatjuk belőle például, hogy milyen böngészőt és milyen képernyő-felbontással használnak azok a felhasználók, akik az illető weblapot töltötték le. Így jobban az ő igényeikhez igazíthatjuk a tartalmat is anélkül, hogy egyetlen kérdést is fel kellene tennünk nekik.

A továbbiakban több űrlapküldő szolgáltatást is bemutatunk. Azonban mindegyiknél a program generál egy weboldalt, amelyen tudatja az űrlap küldőjével, hogy mit küldött el. Kivétel az Extra, ahol mi adhatjuk meg ezt az oldalt.

Ha azt akarjuk, hogy az oldal megjelenésekor a szövegmezőre kerüljön a fókusz, és a benne lévő szöveg azonnal ki legyen jelölve, akkor az alábbi módon is megadhatjuk azt a **body** elemnél.

```
<body onload="F2.Uzenet1.focus():F2.Uzenet1.select:
  kezd()">
```

A példa paraméterei az előző forráslistába illeszthetők be. Mivel van egy **kezd()** nevű szubrutinunk, ezért a felsorolásnak nincs értelme, de ha nem lenne, akkor így is, kettősponttal elválasztva felsorolhatjuk a parancsainkat.

Az első parancs az **F2-es** űrlap **Uzenet1** nevű elemére helyezi a fókuszt, ami itt egy szövegmező. Utána ugyanennek az elemnek a szövegét ki is jelöli, majd átadja a vezérlést a **kezd()** szubrutinra.

Úrlapküldés (Extra)

Az űrlapküldő szolgáltatásoknál az a közös, hogy van egy címen megadott program, amely az űrlapot feldolgozza. Ezt kell a megadnunk a FORM űrlap **Action** paraméterében. Általában kötelező még a címzett megadása, de más nem. Az Extránál az űrlapküldés több szolgáltatást is tartalmaz, ezért több rejtett mezőt is meg kell adnunk.

Ezekben a mezőkben adjuk meg valójában azt, hogy a kiszolgáló az űrlapot hogyan küldje el. Olyan ez, mint egy DIR parancs a DOS-ban. Az az általunk beírt paraméterek szerint listáz, ez pedig aszerint küldi el a kérdőív tartalmát az e-mail-címünkre.

Az űrlapot csak az Extránál regisztrált webhelyünkön helyezhetjük el, és a küldéshez is a regisztrációnál megadott e-mail-címünkre küldik el a levelet. Így az adatok helytelen kitöltése esetén azt nem kapjuk meg. Ha más szolgáltatónál van a honlapunk, akkor több dolgot tehetünk.

Regisztrálhatunk az Extránál is egy webhelyet magunknak, és az űrlapjainkat rendszeresen külön ablakban jelenítjük meg, amelyet használat után be lehet csukni. Így itt csak az űrlapot tároljuk, amelyre teljes címmel kell hivatkoznunk. A másik megoldás az, hogy átköltözünk erre a webhelyre minden weblapunkkal együtt, de ha ez nem járható, akkor keresnünk kell olyan űrlapküldő szolgáltatást, amelyik bárhol lévő űrlapot továbbít. Ilyen is van.

Érdekes körülnézni vagy e-mailben érdeklődni a szolgáltatónknál, hogy milyen lehetőségek vannak. Lehet, hogy elhelyezhetünk CGI-programot a saját webhelyünkön, azt pedig szintén kereshetünk ingyen az interneten. Ehhez kezdjük a keresést a Weblabor oldalain (<http://weblabor.externet.hu>).

A kitöltendő mezők

A <http://stop.extra.hu/extra/> oldalain találunk egy súgót, amelyből letölthetjük az *urlap.txt* fájlt. Ez részletes információkkal szolgál a kérdőív elkészítéséhez. Az alábbiakban ezeket nézzük meg rendezett és kiegészített formában.

```
<form action="/sysprg/formreply.php3" method=POST  
Name="F8">
```

Az űrlap első paramétere a küldés helyét és módját határozza meg. Ezt érdemes kiegészíteni egy **Name** paraméterrel, amelyre a programunkban fogunk hivatkozni. Ha ugyanis írunk hozzá valamilyen VBScript programot, amelyben a

mezőket leellenőrizzük, akkor valahogyan hivatkozunk kell az űrlapra, illetve annak mezőire.

A további űrlapmezők mindegyike rejtett (*hidden*), mert ezek az űrlap elküldésének módját határozzák meg. Ebbe a kérdőívet kitöltő személynek nincs beleszólása, ezért nem is kell, hogy lássa. Azonban a **hidden** szavakat átírhatjuk **text** szóra, és akkor szövegmezőként jelennek meg a rejtett mezőink. Ugyanígy készíthetünk elküldendő mezőt is rejtettként. Arra viszont vigyázunk, hogy ennél a szolgáltatónál a **Name** paraméterben megadott neveket nem szabad megváltoztatni, mert azokkal adjuk át a paramétereket a szolgáltatónak. Tartalmukat viszont nekünk (*vagy a felhasználónak*) kell megadni.

```
<input type="hidden" name="extrauser" value="moricz">  
<input type="hidden" name="subject" value="[E-Űrlap]">
```

Az **extrauser** paraméterbe írjuk be az azonosítónkat, és az ehhez tartozó regisztrációban megadott e-mail-címre megy el az űrlap. Ebben az esetben a **moricz** névhez a **www.extra.hu/moricz/** címen található weblapok tartoznak. A használandó e-mail-címet pedig a regisztrációnál adhatjuk meg, így az űrlap oda megy el. Ha ez a paraméter hiányzik, akkor nem is kapjuk meg a kérdőív adatait. Más e-mail-cím itt nem adható meg.

A **subject** nevű paraméterben állíthatjuk be a nekünk érkező levél tárgyát. Ezt felhasználva az Outlook Express programunk Postabontó szabályait könnyebben alkalmazhatjuk.

```
<input type="hidden" name="maxmezo" value="32">  
<input type="hidden" name="mind" value="no">
```

A **maxmezo** paraméterben adhatjuk meg, hogy maximum hány mezőt küldjön el nekünk. Ez maximum 32 lehet. Akkor van erre szükség, ha minden mező küldését kérjük, függetlenül attól, hogy azt a felhasználók kitöltötték-e vagy sem. Erre a **mind** paraméterben utalhatunk (*yes/no*).

Ha **yes** értékkel küldjük el, akkor az üres mezőket is elküldi, ellenkező esetben csak a kitöltötteket. Általában ez utóbbit használjuk, de ha az e-mailt program dolgozza fel, akkor lehet fontos, hogy az érték nélküli mezőneveket is tartalmazza a levél.

<p>E-mail címed:

```
<input type="text" name="felado" value="a@a.hu"><br>
```

Az előbbi láthatatlan mezők után az első látható mezőnk, amelyben a felhasználótól bekérjük az e-mail-címét. Ezt kötelező neki helyesen megadni, mert különben nem megy el a kérdőíve e-mailben. Nem is lenne értelme, hiszen nem tudnánk, hogy kinek küldjük a választ.

Ha viszont mi nem adjuk meg ezt a mezőt, akkor azt azért tesszük, mert nem is vagyunk kíváncsiak rá, mert csak név nélküli véleményeket gyűjtünk. Ebben az esetben a rendszer azt nem is hiányolja.

Ha a felhasználó azt rosszul tölti ki, akkor kapja meg tőlünk az *emailoldal* paraméterben megadott weblapot (lásd a következő fejezetben). Annak hiányában a rendszer jelenít meg egyet a hibaüzenettel.

```
<p>mező1:
```

```
<input type="text" name="mező1" value="X"></p>
```

Ezután a felhasználói adatokat bekérő mezőink sorakoznak, amelyek *mező1–mező32*-ig számozhatók. Bármilyen típusú mező lehet, mindegyiknek van értéke, amely postázható. Más mezőneveket nem adhatunk meg, mert azt nem postázza a rendszer, csak a *mező1–mező32* nevűeket!

```
<INPUT TYPE="submit" value="Küldés (EXTRA)"></p>
```

```
<INPUT TYPE="reset" value="Törlés">
```

```
</FORM>
```

Végül a legfontosabb gomb, a *Küldés* következik. Ennek nincs neve, viszont a címkéje módosítható. A *Törlés* gombnak igazán nincs nagyon értelme, de fel szokták tüntetni, és bizonyos helyeken szükség lehet rá a kérdőív adatainak törléséhez.

A kérdőív kitöltése és az e-mail-cím (*ha van*) helyes megadása esetén az űrlapot az Extra programja postázza a regisztrációnál megadott e-mail-címünkre. Ezután megjelenít egy válaszoldalt. Ezt mi is meghatározhatjuk egy paraméterben (lásd később). Annak hiányában a rendszer jelenít meg egyet.

Az előzőekben megadott mezőkkel, adatokkal már elkészíthetjük az első kérdőívünket, amelyen a felhasználók üzenhetnek nekünk.

Válaszlevél a feladónak

Több helyet is felkutattam, de az a szolgáltatás mindenhol hiányzott, ami az űrlapot kitöltő személynek válaszlevelet is küld. Az Extránál erre is lehetőségünk van.

```
<input type="hidden" name="reply" value="yes">
<input type="hidden" name="repname" value="Fa Neve">
<input type="hidden" name="repsubject" value="X">
<input type="hidden" name="repbody" value="X">
```

Ez a négy paraméter határozza meg, hogy az űrlapot kitöltő személy kap-e válaszlevelet, és ha igen, akkor az hogyan nézzen ki. Ezt a **reply**-nél döntjük el a **yes/no** válaszunkkal. Küldés esetén a **repname**-nél adjuk meg, hogy milyen név jelenjen meg az e-mail-címünk helyett a levélben. *(Ez még nem működik.)* A **repsubject** paraméterben a válaszlevél tárgyát, míg a **repbody**-ban a levél szövegét adjuk meg.

Könnyen belátható, hogy a tárgyat még megadhatjuk, de a levél szövegét itt beírni nem igazán érdemes. Ezért ikszeltem én itt ki. Ez nem azt jelenti, hogy a levél így megy el! Készítenünk kell egy programot, amely vagy előre, vagy a kitöltést követően, vagy a közben „megírja” a válaszlevelet.

Ha a kérdőív megjelenésekor már megírhatjuk a levelet, akkor az alábbi módon kell azt indítanunk.

```
<body onload="kezd()">
<script language="VBScript"><!--
Sub kezd()
    F8.repsubject.value=" [Urlap] " + window.location.href
    txt = "Kedves Szörföző!"
    txt = txt + vbCrLf + "A tárgysorban olvasható címen
    kitöltött űrlapját megkaptam, hamarosan válaszolok."
    txt = txt + vbCrLf + "...stb."
    F8.repbody.value = txt
End Sub
--></script>
```

Az előző kis program a weblap betöltésekor kerül végrehajtásra. Először beállítja a válaszlevél tárgyát. Ez a megadott karakterláncból és a weblap teljes cí-

méből áll. Mást is megadhatunk, ez csak egy példa arra, hogy aktuális információval is feltölthetjük.

A válasz szövege hosszabb lesz, ezért azt előbb összeállítjuk. Egy karakterlánc-változóba gyűjtjük be a sorokat, azokat mindig sortöréssel (*vbCrLf*) kezdjük, és egyszerű összeadás művelettel bővítjük. Amikor összeállt, akkor a megfelelő mező értékeként hozzárendeljük.

A már ismert eseménykezelésekkel bármelyik mező kitöltése után vagy már közben is „írhatjuk”, bővíthetjük a válaszlevélnek szánt mező tartalmát. Így mire a **Küldés** gombot megnyomja a felhasználó, addigra már a válaszlevele is elkészülhet. Ez lehet egy nagyon impozáns levél, hiperhivatkozásokkal, konkrét adatokkal, dátummal és időponttal.

Ha az űrlapunk egy megrendelést gyűjt be, akkor a következőképpen járjunk el. A felhasználó az űrlap elküldése után megkapja a levelet, ami tartalmazhat egy egyedi azonosítót, jelszót is. Ugyanezek az adatok megrendelésében is szerepelhetnek, amit mi kapunk meg. Ha a felhasználó a levélnek letöltése után, azt változatlan formában elküldi nekünk, azzal igazolja, hogy valóban ő töltötte ki a levelet, és nem valaki más élt vissza az ő e-mail-címével. Így hitelesíthetjük a megrendelést, ami egy kezdő vállalkozásnál nagy segítség lehet.

Válaszoldalak a feladónak

Az űrlap elküldésekor, megjelenik egy válaszoldal. Ezt nem ússzuk egyik űrlapküldő szolgáltatásnál sem. Itt viszont mi is megadhatjuk egy weblap nevét, így teljes egészében a kezünkben tarthatjuk a megjelenítendő weblapokat.

```
<input type="hidden" name="emailoldal" value="err.htm">  
<input type="hidden" name="valaszoldal" value="kosz.htm">
```

Így e két paraméter megadásához készítenünk kell (*érdemes*) egy-egy weblapot, amelyeket a rendszer válaszul megjelenít. Ha rossz e-mailcímet adott meg a kitöltő személy, akkor kapja meg az **emailoldal** paraméterben megadott weblapot. Ha ilyen hiba nincs, és a kérdőív elküldhető, akkor kapja válaszul a **valaszoldal** paraméterben megadott weblapot.

A hivatkozott oldalak elhelyezkedése

Érdekes, de nem írhatunk be teljes webcímet, de keretes (*frame*) oldalét sem, hacsak nem kifejezetten erre a célra készítettük. Ugyanis a küldés pillanatában az aktuális könyvtár a **www.extra.hu**, így ha csak fájlnev szinten hivatkozunk

például az *index.html* oldalunkra, akkor annak az előbbi címen kellene lenni. Ha ebben a fájlban *nevünk/elso.htm* stílusban, tehát útvonallal megadva hivatkozzunk a keretek tartalmára a *frameset*-ben, akkor a teljes címünk beírásakor fogja feleslegesen keresni a megadott könyvtárat. Akkor ugyanis a mi könyvtárunk az aktuális könyvtár.

A lényeg az, hogy csak fájlnevet adjunk meg, mert azt a mi könyvtárunkból veszi, de ebben a fájlban hivatkozott többi oldalnak már útvonallal kellene hivatkoznia.

A másik lényeges szempont, hogy a küldés után a webböngésző címezőjében a *http://www.eztra.hu/sysprg/formreply.php3* cím olvasható. Mivel ez nem mutat jól, ezért érdemes az űrlapot olyan új ablakban megnyitni, ahol nincs címező. Utána pedig olyan oldalt jelenítsünk meg, amelyben van egy program (pl.: *VBScript*) vagy *META*-tag, ami azonnal megnyitja a következő oldalt, célszerűen a nyitóoldalunkat. Itt már hivatkozhatunk a teljes címünkre.

Hibás e-mail-címnél megjelenő válaszdoldal

Ha a felhasználó hibás e-mail-címet adott meg, akkor az *emailoldal* paraméterben megadott weblapot kapja. Ezen az oldalon tájékoztathatjuk a felhasználót, hogy a megadott e-mail-címet hibásan írta be.

Mivel más mondandónk nem nagyon van, ezért érdemes pár másodperces megjelenítés után automatikusan kiadni egy visszaléptető parancsot, így a felhasználó vissza is kapja az előző űrlapot a beírt adataival együtt.

```
<HTML>
<BODY onload="kezd()">
<SCRIPT LANGUAGE=vbscript><!--
Sub kezd()
    st=window.setTimeout("window.history.back()", 2000)
End Sub
--></SCRIPT>
<H1>Hibás címet adott meg. Most visszalépünk...!</H1>
</BODY></HTML>
```

A weblap kiíró részét természetesen díszesebbre is készíthetjük, hosszabb szöveggel, itt most nem azon van a hangsúly. A weblap megjelenítésekor elindul az *onload* eseménykezelőnél megadott *kezd()* program, a *setTimeout()* metódus használatával beállítjuk, hogy 2 másodperc (2000 msec) múlva hajtson végre egy visszalépés parancsot.

Ez azzal egyenértékű, mintha a felhasználó önként megnyomná a webböngészőjében a *Vissza (Back)* gombot. Az idézőjelek között megadhatjuk akár egy saját szubrutin nevét is, amelyre más feladtnál lehet szükségünk. Így miután az űrlapot kitöltő személy értesült a hibájáról, már vissza is kerül az űrlapra, amelyen kijavíthatja az e-mail-címét.

Tovább lépés a válaszdoldalról jó adatok esetén

Az adatok kitöltését követően megjelenő weboldal nem feltétlenül érdekes a felhasználó számára, ezért el kell érünk, hogy azt minél gyorsabban elhagyja. Ehhez felhasználhatjuk az előző módszert.

Készítünk egy oldalt, amelyen megjelenik valami felirat az előzőhöz hasonlóan, majd a *setTimeout()* metódus használatával megadjuk a végrehajtandó parancsot. Ebben az esetben azt adjuk meg, hogy az ablakba töltse be az általunk megadott címen lévő oldalt.

A parancsot beírhatnánk egyszerűen is, de akkor nagyon hosszú lenne a sor, és az nehezen áttekinthető programokhoz vezet. Ezért most soronként építjük fel a *parancs*ot. Előbb megadjuk a címet, majd összeállítjuk a parancsot, végül ezt írjuk be a *setTimeout()* metódusnál. Mennyivel szebb így!

```
<HTML>
<BODY onload="kezd()">
<SCRIPT LANGUAGE=vbscript><!--
Sub kezd()
    cime = "http://www.extra.hu/moricz/": ap = ""
    parancs = "window.top.location=" + ap + cime + ap
    MsgBox parancs
    st = window.setTimeout(parancs, 2000)
End Sub
--></SCRIPT>
<H1>Köszönöm. Most tovább lépünk...!</H1>
</BODY></HTML>
```

Levelezőlista felállítása (YahooGroups)

Már a februári Tippek füzetben volt szó saját levelezőlista felállításáról. Az internet gyors fejlődésének lehettek tanúi azok, akik akkor még a *onelist.com*-nál regisztráltak listát, mára pedig *egroups.com*-nál folytatják a megújult felületen a beállításukat. A webhely ugyanis időközben átlakult, megújult, és újabb hasznos szolgáltatásokat nyújt. *(Azóta pedig YahooGroups a neve !☺)*

Most részletesebben is megnézzük, hogyan kel egy ilyen levelezőlistát elkészíteni, működtetni. Leginkább a második szón van a hangsúly, mert a lista létrehozása 5-10 perc alatt elvégezhető a regisztrációval, utána pedig csak leveleznünk kell. Erre a webhelyre akár vissza se kell menünk soha, a levelezőlista akkor is jól működhet akár éveken keresztül.

A lista megtervezése

Levelezőlistát általában azért hoznak létre az emberek, hogy a valamilyen témában hozzá hasonló gondolkodásúakkal levelezni tudjon. Nem kell, hogy feltétlenül ismerjék egymást a listatagok. Sőt, akik csak feliratkoznak egy levelezőlistára, de soha nem szólnak hozzá, csak olvassák a leveleket, azok a többiek számára mindvégig rejtve maradnak.

Hírlevél véleményekkel

A levelezőlistát létrehozó személy *(általában a lista vezetője, esetleg moderátora)* azért is készíthet ilyen listát, hogy hírlevelet küldjön az üzleti partnereknek, amelyre azok akár reagálhatnak is. Így a lista „alaphangját” a heti vagy havi hírlevél adja meg, majd utána ennek kapcsán mindenki elmondhatja a véleményét, megvitatják a leírtakat.

Figyelmeztetők a közös programokra

A levelezőlista weblapjain bejegyezhetünk közös programokra emlékeztetőket is. Az időpont bekövetkezésekor a rendszer automatikusan küld egy levelet minden listatagnak. A listatagok is elkészíthetik ezeket a bejegyzéseket, így a saját üzleti eseményeikre is felhívhatják a többiek figyelmét.

Ilyen levelezőlistát létrehozhatnak közös üzleti érdekeltségbe tartozó személyek is, akik így informálják egymást az eseményeikről. Egy ilyen lista formáját csak az események adják, így egyáltalán nem zavaró.

A szükséges dokumentumok elkészítése

Ha egy kicsit is komfortosabbá akarjuk tenni a listánkat, akkor érdemes néhány dokumentumot elkészíteni. Kettő mindenképpen fontos, a feliratkozáskor és a lemondáskor küldendő fájl. Ezekben tájékoztatjuk az illető személyt a lehetőségekről, a házirendről, az elvárt viselkedésről és a kialakult szokásokról. Lemondásnál egy emlékeztető levelet érdemes küldeni, ami a visszatérés lehetőségét biztosítja.

Dokumentum a feliratkozáshoz

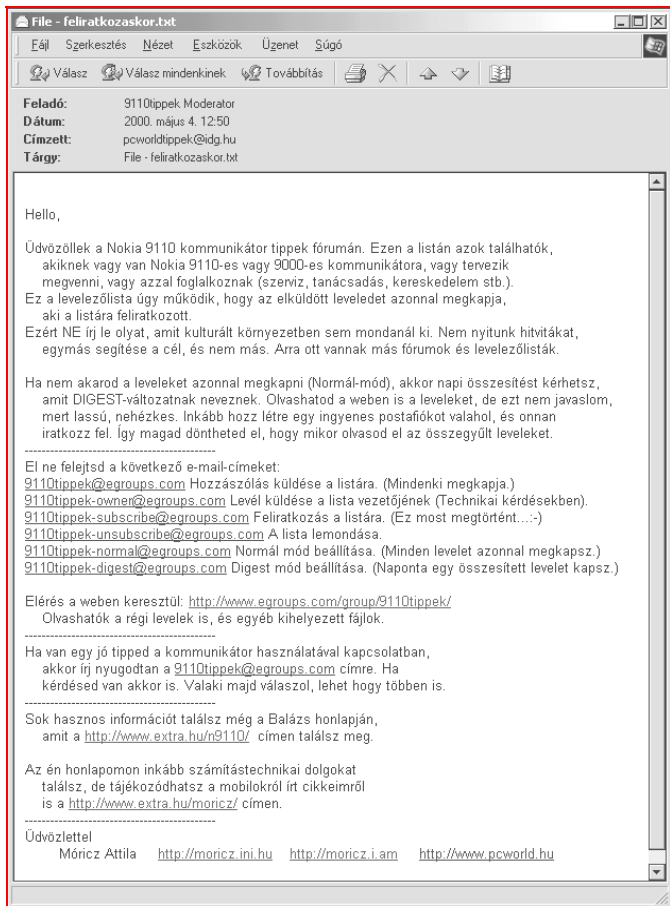
Ez a dokumentum feltétlenül tartalmazza azokat az e-mail-címeket, amelyek a lemondáshoz, hozzászóláshoz és egyéb műveletekhez kellenek. Érdemes elmagyarázni egy kicsit ebben a levélben a lista legalapvetőbb működését is, mert nem biztos, hogy mindenki képzett számítástechnikus.

Fontos, hogy ebben a levélben tájékoztassuk az olvasót a legalapvetőbb illemszabályokról is, mert nagyon megkeserítheti az egészet egy-egy nevetlen levelezőpartner. Lásd például a *Mobil@Hix*-listát, ahol április táján már egészen másról szólt a levelezés egy nevetlen listatag miatt.

A lista témájában sok esetben konkrét webhelyekhez is kapcsolódhat, vagy azok propagálására jön esetleg létre. Így ez a levél tájékoztathatja az olvasót a témához kapcsolódó legfontosabb webcímeiről is.

Pár sort írhatunk a hozzászólás, behallgatás módjáról is. A listatagoknak érdemes egy ideig csak figyelni a leveleket, és ha kérdésük van, akkor azt ne a feliratkozás után tegyék meg azonnal, mert lehet, hogy egy hete választott rá valaki. Az előző levelek a weben is megtekinthetők, így érdemes előbb ott tájékozódni. Ehhez viszont elengedhetetlen, hogy a listára menő levelek tárgysorába a felhasználók valóban azt írják be, amiről a levelük szól. *(Ez azokra is vonatkozik, akik a pcworldtippek@idg.hu címre írnak levelet, mert még mindig sok a „kérdés” és a „Móricz Attilának” tárgysorú levél...:-))*

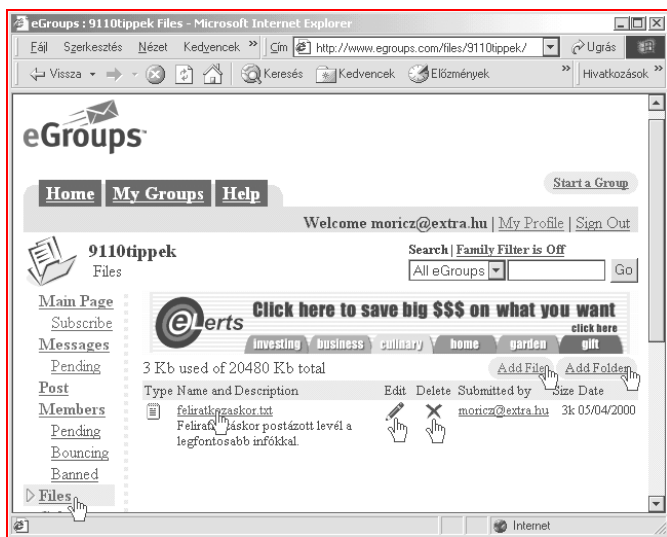
A fentiekből is látható, hogy ha egy kicsit is komolyan akarjuk működtetni a listánkat, akkor sok mindenre oda kell figyelnünk. Tapasztalatokat más levelezőlistákon szerezhethetünk, így kialakíthatjuk a sajátunk arculatát. Ha nem akarunk sok bosszúságot, és hasznossá akarjuk tenni a listánkat, akkor minden apró részletre, problémára figyelünk és reagálnunk kell. Érdemes a gyakran feltehető kérdéseket egy FAQ (GYIK) listába gyűjteni, amelyet időről-időre az interneten közzétehetünk, így elkerülhetjük a további felesleges kérdéseket.



3. ábra. Egy feliratkozaskor postázott levél fontos információkkal (élő lista)

A fájlok feltöltése

Az előbbi ábrán látható levél forrásául szolgáló szöveget a listavezető töltheti fel a **Files** linkre kattintás után megjelenő oldalon. Itt az **Add File** linkre kattintsunk, és a **Tallózás** gombbal keressük meg a helyi fájlrendszerünkben a feltöltendő fájlt. Ugyanitt bejelölhetjük, hogy a fájlt milyen esemény bekövetkezésekor postázza (*feliratkozás, lemondás, 2 hetente, havonta, kitiltáskor*). Utána az alábbi ábrán látható tartalmat kapjuk vissza. Itt láthatjuk, hogy létrehozhatunk mappákat is az **Add Folder** linkre kattintva, de a szövegfájlunkat meg is nézhetjük, szerkeszthetjük a hozzá megjelenő információkat, de törölhetjük is.



4. ábra. A Files csoportban gyűjthetünk más fájlokat is 20 MB-ig

Ugyanitt kell feltöltenünk a lemondás és más említett események bekövetkezésekor postázandó leveleket is. Amint az ábrán is látható, akár 20 MB-nyi fájl is elhelyezhetünk itt. Ennél a listánál például máshonnan is letölthető, de kipróbált, működő ingyenes vagy Shareware programokat érdemes feltölteni.

A listakezelés tesztelése

Feliratkozás a listára

Miután elkészítettük a listát, feltöltöttük a tájékoztató szövegfájlokat, küldjünk egy feliratkozó levelet egy másik e-mail-címünkről. Jó esetben másodpercek múlva már le is tölthetjük a postafiókunk tartalmát, mert megérkezik egy tájékoztató angol nyelvű levél a lista kiszolgálójától, hogy a mi címünkről feliratkozási kérés érkezett.

Ha nem állnánk magas szinten az angol nyelv művelésében, akkor sincs baj, mert nagy betűkkel ki van írva, PLEASE REPLY, szóval küldjük vissza változtatás nélkül. Ezzel igazoltuk, hogy nem valaki más szórakozik a mi címünkkel. A küldést követően másodperceken belül kapunk egy levelet a kiszolgálótól angolul, hogy a listához hozzáadtak minket. Utána jön automatikusan egy rövid levél a moderátortól, amit szintén a lista létrehozásakor kellett megadni, amelyben már magyarul tájékoztathatjuk a feliratkozót, hogy a lista tagja lett. Ha pedig feltöltöttük az előbbieken részletezett feliratkozáskor küldendő levelet, akkor azt is megkapja a feliratkozó.

Ezzel tudjuk ellenőrizni, hogy hogyan zajlanak a folyamatok, mi, hogyan jelenik meg, így segíteni tudunk azoknak, akiknek az angol nyelvű levelek gondot okoznak.

A listatulajdonos levélforgalma

Ha készítettünk egy listát, és az előzőekben leírt módon teszteljük, akkor egyszerre csak a tesztelt e-mail-címre érkező leveleket töltjük le, mert különben belekavarodhatunk. Ugyanis mi vagyunk a saját listánk tulajdonosai is, így minden listával kapcsolatos eseményről értesülni fogunk.

Így az új személy feliratkozásáról, lemondásáról, a fájlok feltöltéséről és minden változásról kapunk egy e-mailt, amivel a gyarapodást az ellenőrzésünk alatt tudjuk tartani.

A lista lemondása

Az előbbieket után természetesen lemondhatjuk erről az e-mail-címünkről a listát (*unsubscribe*), amellyel azt is ellenőrizni tudjuk, hogy a listatagok valóban el tudják hagyni a listát, ha azt akarják. Itt is vissza kell küldenünk egy levelet a jóváhagyáshoz. *(Ennek feldolgozása viszont sokkal tovább tart!)*

Tagok felvétele a listára

A levelezőlistára nemcsak e-mailben lehet feliratkozni. Ha van egy felhasználói csoport, akiknek egyik pillanatról a másikra létre kell hozni egy listát, akkor az ő e-mail-címeiket előbb össze kell gyűjteni egy szövegfájlba. Utána a lista-kezelő weboldalon kattintsunk a **Members** hivatkozásra, és megjelenik a tagok listája.

Csoportos felvétel

Az előbbi példa szerint ez a <http://groups.yahoo.com/group/9110tippek> (☞) címen található meg. Ezen az oldalon találunk egy **Add/Invite** gombot. Erre kattintva kapunk egy másik weblapot, ahol egy többsoros szövegmezőbe írhatjuk be vagy vágólapon keresztül másolhatjuk be a tagok e-mail-címeit (*maximum 100 főt*). Soronként csak egy cím legyen.

Ugyanitt azt is beállíthatjuk, hogy a listatagoknak kell-e jóváhagyást küldeni vagy nem. Ilyen esetben nem kell kérnünk, mert tudatos a tagok felvétele, ők is tudnak róla, így az felesleges és zavaró lenne. Ráadásul, ha valaki nem küldi vissza két napon belül a levelet jóváhagyásul, akkor az nem kerül fel a listára. Egy másik szövegmezőben írhatunk pár sort, amelyben a felhasználókat tájékoztatjuk, hogy hozzáadtuk őket a listához.

Utána ismét a tagok listájának oldalára kerülünk vissza. Ezen az oldalon egyszerre vagy külön-külön is beállíthatjuk a levelek postázásának módját. Ez lehet **email message** – normál levelezés, **daily digest** – napi összesítő, **web only / no mail** – csak a weben olvasva, levélküldés nincs, **unsubscribe** – lemondás. Ugyanitt a **Download list** hivatkozásra kattintva a listát megnyithatjuk szövegfájlként, azt elmenthetjük a saját gépünkre, ha szerkeszteni akarjuk.

Feliratkozás weblapról

A beállítási weblapon találunk bal oldalon a sok hiperhivatkozás között egy **Promote** nevűt. Ezen az oldalon HTML-űrlapokat találunk, amely egy képes gombból és egy szövegmezőből áll. Ezt kell elhelyeznünk a saját személyes weblapjainkon, hogy a szörföző felhasználók csatlakozhassanak a listánkhoz.

A felhasználónak nincs más dolga, mint beírni az e-mail-címét a mezőbe, és a kis képre kattintani. Ezzel a <http://groups.yahoo.com/group/9110tippek> (☞) címen lévő oldalra jut el (*a példánál maradva*), ahol néhány beállítást elvégezve csatlakozhat a listához.

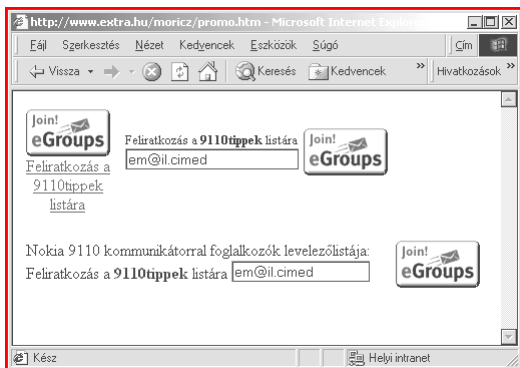
A levelezőlista hirdetése a saját weblapjainkon

Az előbb említett címen, illetve hivatkozáson elérhetjük azt a weblapot, ahol a hirdetéshez szükséges HTML-kód megtalálható. Nem bonyolult egyik sem. Az első kicsit magyarosítva az alábbi módon néz ki.

```
<center>  
<a href="http://groups.yahoo.com/group/9110tippek/join"  
target="m2">  
<br>Feliratkozás a 9110tippek listára</a>  
</center>
```

Középre igazítva jelenik meg egy kis kép és alatta a rövid felirat, amelyekre kattinthatunk, mert hiperhivatkozásként jelennek meg. A hivatkozás az előző oldalakon mutatott **eGroups** webhelyre viszi el a felhasználót, ahol a megfelelő helyen beírhatja az e-mail-címét, és ezzel feliratkozhat a listára.

Az eredeti kódban nem szerepel a **target="m2"** részlet a hiperhivatkozásban. Ez ahhoz kell, hogy külön ablakban jelenjen meg a hivatkozás. Az **m2** helyett bármit írhatunk, mert ilyen nevű keretünk (*frame*) nincs, ezért külön ablakban jelenik meg a tartalom.



5. ábra. A feliratkozásra csábítás a legegyszerűbb űrlapelemekkel

A másik HTML-kód egy űrlap, amely ráadásul egy újfajta elemet is tartalmaz. A táblázatba foglalásra az egyszerűbb elhelyezés miatt van szükség. Az első sorban jelenik meg az invitáló szöveg, míg a másodikban a szövegmező az e-mail-címnek és egy kép típusú küldés gomb. A típusa **image**, amelynek csak a forrását kell megadni. Így a felhasználó a címét már itt beírhatja, és a megjelenő weblapon már csak a személyes jelszavát kell megadnia.

```
<form method=GET
action="http://groups.yahoo.com/subscribe/9110tippek">
<table bgcolor=#FFFFCC>
<tr><td colspan=2 align=center>
  <small>Feliratkozás a <B>9110tippek</B> listára</small>
</td></tr>
<tr>
  <td><input type=text name="user" value="em@il.cimed">
  </td><td><input type=image src="kepek/join.gif"></td>
</tr></table>
</form>
```

Az előbbi elemek kombinálását láthattuk az ábrán. A pontosabb elhelyezés érdekében alkalmazzunk táblázatot, táblázatcellákat. Az elsőnél a táblacellát kell olyan keskenyre méretezni, mint amilyen a kép mérete. Erre csak akkor van szükség, ha nagyon szűkében vagyunk a helynek a weblapunkon.

A második ajánlási formánál azt használjuk ki, hogy a „Küldés” gombunk éppen olyan magas, hogy a szövegmező felett még éppen elfér egy sor szöveg. Keskenyebb betűtípussal és pontosan beállított méretű szövegmezővel jól kitölthetjük a rendelkezésünkre álló részt.

A harmadik ajánlóban az előző elemeket egymás mellé helyezzük, de így nagyon lépcsőzetesen jelenik meg, ezért felülre elhelyezhetünk még egy sort valamilyen kiegészítő információval.

Azoknál az elhelyezéseknél, ahol a táblacellák több soron vagy több oszlopon átnyúlnak, ott az illető cellákat előbb jelöljük ki, majd egyesítsük azokat. Érdekes megnézni utána a HTML-kódot is, hogy megismerjük, mikor kell a **<td colspan="3">** paramétert használni a cellánál (*td*), és mikor van szükségünk a **<td rowspan="2">** alakra. Az első egy olyan cella, amelyik felett 3 cella van az előző sorban. A **rowspan** paraméterre a második ajánlónál lehet szükségünk, ahol a jobb oldali cella két bal oldali soron átnyúlik.

Olvasók

"Hogyan tudnék olyan animált reklámcsíkot készíteni, amiben van kép, szöveg, és különböző szövegeket és képet jelenít meg különböző időközönként. Milyen program szükséges hozzá, van-e esetleg ingyenes változat, amely letölthető az internetről?"

A programot úgy hívják, hogy Microsoft GIF Animator, de van más cégnek is ilyen terméke. A lényeg az, hogy elkészít egy képet, akár a *Windows Paint* programjával is, ami tartalmazhat feliratot is, vagy csak azt, és ezeket a képeket kell a vágólapon keresztül bemásolni a programba. Ott lehet megadni azt is, hogy milyen időzítéssel tegye ki a másik képet, megőrizze-e a hátteret stb.

A program kezelése nagyon egyszerű, inkább az animáció megtervezése okozhat fejtörést. Ha kitalálta milyen feliratokat akar megjeleníteni, akkor már egy óra alatt össze tud állítani olyan animált GIF-képet, ami már megállja a helyét egy honlap tetején.

Keresse a www.lycos.ftpssearch.com címen a **gifsetup.exe** fájlt, mérete: 1'079 kB, 1996-os kiadás, de jó az. Ez ingyenes, nem időkorlátos. Van még a **gifcom.exe**, **gifcon32.exe**, időkorlátos **GIF Construction Set**, de előbb nézze meg a PC World CD-ken is.

Kapcsolódó könyv: Miklósi Viktor: Weblapszerkesztés mindenkinek.

(Kiadó: Cesare Complex & Cserkő Bt.)

„Azt szeretném megkérdezni Móricz Attilától, hogy milyen programmal készítette az internetes weblapjait? (<http://www.extra.hu/moricz/>)”

A weblapjaimat Microsoft FrontPage 97-tel készítettem, de semmi olyan nincs rajta, amit például a Windows 98-ban lévő FrontPage Express programmal ne tudtam volna elkészíteni. Talán egy kicsit kényelmesebb azt a programot használni, és könnyebb hiperhivatkozásokat és egyéb beszúrásokat készíteni vele.

A hozzá tartozó FrontPage Explorer jó olyankor, ha egyes oldalakat áthelyezünk, mert akkor minden olyan oldalt módosít, amelyben erre a weblapra történt hivatkozás. Így nem lesznek elkalandozott hiperhivatkozások, és a beszúrt, de időközben áthelyezett képek is biztosan megjelennek.

Minden dinamikus elemet az adott weblapelem formázásánál megtalálható *További...* gombra kapott párbeszédablakban írtam be (*Név/Érték párok*), ami ott ugyan nem jelenik meg, de ez nem gond.

A keretes oldalak készítéséhez van ugyan egy varázslója, de azt érdemesebb inkább egy Jegyzetömbben elkészíteni. Az oldalon történő pozicionáláshoz a táblázatok nagyon jól megfelelnek. A színezéshez szerintem nem feltétlenül kell mindenhol grafikákat használni, sokszor az alapszínek használata is lehet figyelemfelkeltő.

Érdemes tetszőleges leírónyelven készült programokat beszúrni minden oldalra legalább alulra, felülre, amely nem tartalmaz mást, csak ***document.write*** utasításokat (*JavaScript*), amelyek HTML-kódot írnak a weblapra. Így annak az egy-egy fájlnak a módosításával elérhetjük, hogy minden oldalunkon egyszerre megváltozzon a tartalom, például egy adat a kiadónál vagy a hirdetés tartalma. Ehhez a weblapon elég a Beszúrás / HTML-kód paranccsal elhelyezni az alábbi sort:

```
<script language="JavaScript" src="http://cim/neve-1.js"> </script>
```

Ez a JavaScript fájl pedig az alábbi utasításokat tartalmazza:

```
document.write("<CENTER><table><tr><td")
```

```
document.write("<A HREF='http://hle.isys.hu' TARGET='s2'>")
```

```
document.write("...stb.
```

Arra kell csak vigyázni, hogy a HTML-kódban szokásos idézőjelek helyett aposztrófot használjunk, mert az idézőjeleket a JavaScript utasítások lefoglalják. Ha nem akarjuk nagyon törni a fejünket a kódon, akkor készítsünk el egy weblaprészletet, majd a forráskódja minden sorát illesszük be a JavaScript *document.write("")* utasítás idézőjelei közé. A feladatokat akár egy Word makróval vagy egy VBScript programocska segítségével is elvégezhetjük.

Dinamikus és díszesebb weblapok készítéséhez használhatjuk a FrontPage 2000 programot, amely lényegesen nagyobb, és sokkal több szolgáltatást is nyújt, mint az elődje. Eszköztárról egyszerűen választhatunk az adott elemhez eseménykezelést, de éppen ezért lényegesen nagyobb fájlokat hoz létre, mert sok felesleges információt is tárol a weblapon.

„Találtam a D: meghajtómon egy file0001.chk nevű file-t, ami több mint 400 MB. Ezzel most mit csináljak? Nyugodtan letörölhetem?”

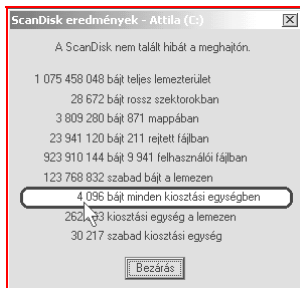
Ha lehetne, akkor a filét csak a hallal kapcsolatban emlegetnénk, mert már régóta van magyar megfelelője: fájl. Ugyanígy weblap, webhely, újabban internet, flopi, de az e-maillal még hadilábon állunk...:-)

Ha kisebb lenne az a fájl, azt mondanám nézze meg egy fájlnevezővel, hogy mi van benne, de így nincs értelme. Azonban elgondolkodtató, hogy a ScanDisk olyan hibát talált, hogy majd az egész merevlemez bepakolta ebbe a fájlba, és fel sem tűnt, hogy valami hiányzik? Ha nem, akkor törölje és gondolkodjon. Mit tett? Mert ekkora filechk.001 fájl nem jön létre minden nap az ember gépén. Előtte azért lefuttatnám a ScanDisket újra, nem-e ismét egy keresztbeláncolásos hiba van a meghajtóján. Mert ha igen, akkor esetleg ezzel a töréssel netán az egész meghajtóját letörli.

„A FAT32-vel kapcsolatban kérdezem, hogy okozhat-e valami problémát, ha Meghajtókonvertert akkor indítom el, amikor már minden adatom, programom rajtavan. Mekkora az a meghajtóméret, amelynél már elvégzi a konvertálást, mert az én 100 MB-os meghajtómra azt írja ki, hogy túl kicsi.”

Az első kérdésre természetesen nem a válasz, mert jól néznénk ki, ha gondot okozna. A konvertálás percek alatt megtörténik, viszont utána elindul a *Töredezettségmentesítő* program, amely lemezmérettől függően órákig is dolgozhat a meghajtón.

A Meghajtókonverter program használatának csak 512 MB-nál nagyobb meghajtóknál van értelme. Ugyanis az ön 100 MB-os meghajtóján a klaszterméret csak 2 kB, így még rosszabbul is járna, hiszen a FAT32-nél a legkisebb klaszterméret 4 kB. Olyan esetekben használjuk ezt, amikor már az egységek mérete 16-32 kB (*nagyobb meghajtóknál*), így a sok 1-4 kB-os fájl tárolása rendkívül gazdaságtalan. Az ön esetében a DriveSpace programot kellene használni, bár egy 100 MB-os meghajtónál lehet, hogy ez is csak rontana a helyzeten.



6. ábra. A ScanDisk eredményei között megtaláljuk a klaszterméretet is

„Időnként lefuttatom a ScanDisket. Ez azonban - nem tudom miért - nem törli a TMP-kiterjesztésű fájlokat. Így időnként átnézem a DOS-ban a gépet, és törölöm ezeket, hogy ne foglaljanak helyet. A kérdésem az, hogy létezik-e más kiterjesztés is, amire a működéshez nincs szükség? (Pl. kipróbáltam, és soha nem volt probléma, ha a "bak" kiterjesztésűeket törölöm.)”

A ScanDisk hibakeresésre szolgál, és jó esetben is csak a foglaltnak bejelölt szektorokat szabadítja fel, de nem töröl. A Lemezkarbantartó programnak lenne a tiszte törölni a felesleges fájlokat, de az sem teszi meg, csak úgy csinál, mint aki dolgozik. Ezért önnek kell kézzel törölnie a felesleges fájlokat, de ehhez nem kell a DOS-ba mennie.

Indítsa el a Keresőt (F3) akár a Start menüből is, és keressen rá a merevlemezén az alábbi fájlokra egyesével. Ha megvan mind, akkor azokat a **Ctrl+A**-val jelölje ki, és a **Shift+Del**-el törölje le. Utána keressen a következőre.

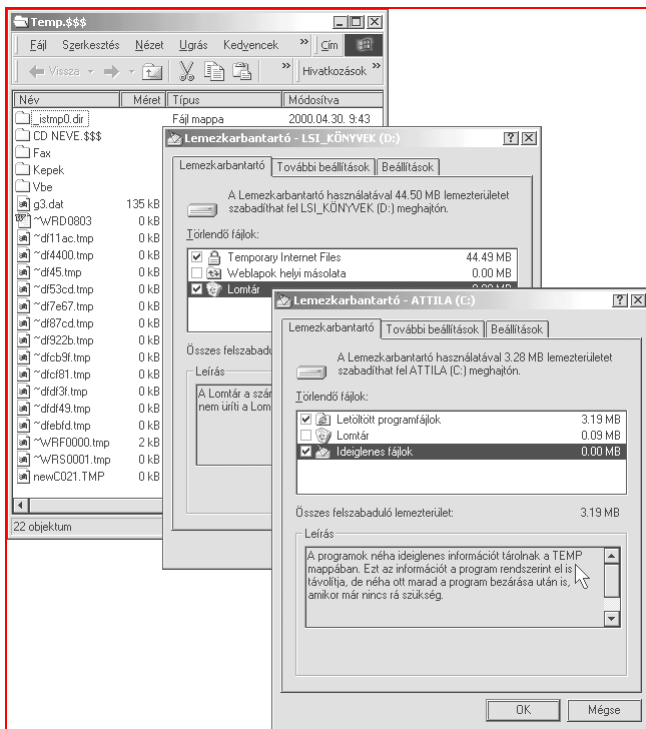
A fájlok: *.TMP és ~*.*

Ha javasolhatom, akkor az alábbiakat a *Lomtárba* tegye, tehát csak **Del**-el törölje, és előtte nézze át a *Lomtárat*, ha az ott lévők nem kellene, akkor ürítse azt, majd utána töltsön bele újabb szemetet.

*.BAK (általában biztonsági másolatfájlok)

*.WBK (Word biztonsági másolatfájlok)

Ha ugyanis az eredeti dokumentumai megsérülnek, akkor azok előző változatait tartalmazó biztonsági másolatfájlok (*.bak, *.wbk) alapján hozzájuthat egy nem olyan régi változathoz, és nem kell újra elkészíteni az egész munkát. Ezért tegye ezeket inkább a *Lomtárba*.



7. ábra. Lemezkarbantartás programmal és kézi módszerekkel

„Van-e olyan szerver valahol a nagyvilágban, ahonnan SMS-t lehet küldeni úgy, hogy nem kellene belépni a weblapra. Arra gondolok, hogy e-mail érkezésekor a levél fogadó szervernek meg lehetne adni, hogy továbbítson egy üzenetet az SMS-küldő géphez, így értesítést lehetne kapni a megjött e-mailről. Tudom, hogy van ilyen előfizetés a mobil szolgáltatóknál, csak arra gondoltam, hogy kevesebb pénzért (ingyen?) is meg lehet-e ezt oldani.”

Van több is, azonban vigyázzon. Ahogyan az ingyenes e-mailnél, weblapnál elfogadjuk a hirdetéseket, esetleg lemondjuk emiatt a szolgáltatást, és átmegyünk máshová, úgy ezt a mobiljával majd nem fogja tudni megtenni. Több hely is van, ahol regisztráció után kap egy e-mail-címet, és az ön által hozzárendelt mobilszámra kap egy értesítést SMS-ben, ahogyan ez a fizetős hazai szolgáltatóknál történik.

Azonban ezek az értesítések vagy megkésve érkeznek meg, vagy sehogyan. De mondok egy jobb történetet, ami már tényleg az X-akták kategóriájába tartozik, mert értelmes magyarázatot nem lehet rá adni.

Van egy orosz szolgáltatás, amelynek címét itt leírom, de felejsék is el azonnal. A *tel.szam@sms.gate.ru* címre küldött e-mailről elvileg értesítést kap a „tel.szam” telefonszámú mobilon. Egyszerű, olyan mint a Pannoné. A különbség az, hogy az 300 Ft/hó és működik, ez pedig ingyenes és nem megy.

Történt, hogy próbaképpen én magam is kipróbáltam, mert tudom, hogy az előrefizetett kártyásoknál (*P. Praktikum, V. VitaMax*) ilyen szolgáltatás nem igényelhető, így sokaknak szüksége lehet rá. Átirányítottam az egyik e-mail-címemre érkező leveleket erre a címre (*másolatként*), és gyorsan küldtem is rá néhány e-mailt próbaképpen.

Az értesítést megkaptam (*mint feladó*), hogy az SMS-eket a rendszer küldte, de azokat nem kaptam meg később sem. Viszont felhívott valaki azon a számon, amelyen vártam az SMS-t, és közölte, hogy ő kapott SMS-t, amelyet valószínűleg nekem kellett volna megkapnom.

Ezután bemutatkoztunk. Ekkor jött a meglepetés! Ugyanis ismertük egymást egy levelezőlistáról, de még nem hívtuk egymást soha. A számunk jelentősen különbözik egymástól, és elírás sem történt. Egyéb úton semmilyen kapcsolat nem volt közöttünk, de ő is kipróbálta már egyszer ezt a szolgáltatást, de neki sem működött. Nem tudom melyikünk lepődött meg jobban ezen a fura találkozáson, de erre egyikünk sem számított.

Arra azonban talán soha nem kapunk magyarázatot, hogy miért pont hozzá ment az én eltévedt SMS-em, miért nem egy teljesen ismeretlen emberhez? De ha valakinek van tippje, nyugodtan ossza meg velem. Ezért aztán az önök helyében óvatosan próbálgatnám az ilyen szolgáltatásokat, mert a mobilszámunkat nem tudjuk olyan egyszerűen lecserélni, mint az e-mail-címünket vagy a webcímünket.