

## Az SVG világa (1. rész)

### Alapvetés, történet

Most induló sorozatunkban az SVG-t fogjuk körüljárni. Tartson velünk, aki érdekli a webfejlesztés és a grafikus programok iránt, vagy ha egy kicsit szeretne a jövőbe tekinteni... Ízelítőként a tartalomból: W3C, MMS, XML, Sodipodi, Inkscape, Mozilla, Thunderbird, Konqueror, KDE, GNOME, PHP és így tovább. Ennek a hosszú felsorolásnak a tagjaiban az a közös, hogy mindegyik valamilyen kapcsolatban áll az SVG-vel. A lista az alkalmazások nagyon széles spektrumát jelenti a mobiltelefonoktól egészen a programozási nyelvekig. A sorozat első részében a W3C, az MMS és az XML titkairól lebben fel a fátyol.

#### Pixel- és vektorgrafika

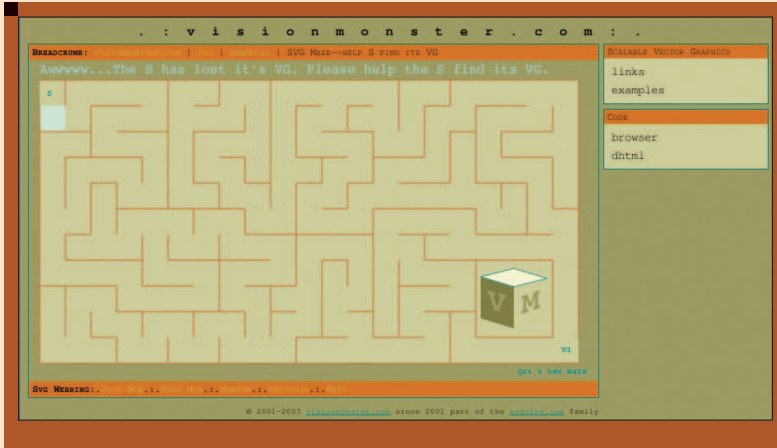
Ahhoz, hogy az SVG-vel bármilyen szinten tudjunk foglalkozni, sőt, még mielőtt megtudnánk, mi ez egyáltalán, ismernünk kell a számítógépes grafikák két nagy csoportját. A raszteres képeket mindannyian jól ismerjük, ilyen képeket készít például a digitális fényképezőgép, a lapolvasó és a pixelgrafikus programok, például a *Gimp*. A sokféle részterület közös jellemzője, hogy az elkészült kép tovább nem nagyítható minőségromlás nélkül, valamint a kép méretével és színmélységével egyenes arányban nő a lemezen elfoglalt mérete, mivel elemi képpontokra felbontott képet tárol. Ezen tulajdonsága miatt gyakran a veszteségesen tömörített alfajával találkozunk (*JPG* és *GIF*), ahol a mérete már kevésbé riasztó, de az állandó újratömörítések sokat rontanak a minőségen. A vektorgrafika ellenben alakzatokat és tulajdonságokat tárol, tehát csak a grafika összetettsége határozza meg a képméretet, a megadott információk alapján bármekkora kép kirajzolható, azaz tetszőlegesen nagyítható. Tipikus felhasználási területe a műszaki célú (például *CAD*) és a általános vektorgrafikus programok. Ha a webes felhasználást tekintjük, akkor a kettő előnyös kombinációja a legcélszerűbb. Jelen pillanatban ezt a legkiforrottab-

ban a *Macromedia Flash*-sel tudjuk elérni. Ezen formátumok is néha tömörítettek, ellenben felépítésénél fogva veszteségmentesen tömörítik a vektorgrafikát. Az *SVG* esetében is tömöríthetjük a végeredményt, például a *gzip*-pel. Igaz, hogy ezt egy *BMP* (pixelgrafikus) képpel is megtehetjük, de az egyszerű szövegfájl, az *SVG*, jobban tömöríthető, mint a pixelgrafika. Ebben a környezetben tetszés szerint keverhetjük a raszteres és vektoros elemeket a leggyorsabb és leglátványosabb megoldások érdekében. De mi is a gond a *Flash*-sel? Előnye, hogy egész animációkat, valóságos kisfilmeket lehet vele gyártani, de nem nyílt szabvány, a web fejlesztők központjában, a *World Wide Web Consortiumnál* (a továbbiakban *W3C*) úgy vélték, szükség van egy teljesen platformfüggetlen, általános célú, szabványos vektorgrafikai formátumra, mely teljesen beleillik a többi *W3C* szabvány közé.



Az SVG története és fejlődése

Ez lett az *SVG*, a *Scalable Vector Graphics*, mely immáron hat éves múltat tekint vissza, hiszen 1999-ben jelent meg az első piszkozata a formátum leírásának. Nem kellett sok idő és az új specifikáció 2001-ben *W3C*



ajánlássá lépett elő **SVG 1.0** néven. Azóta frissebb változatok és új hajtások is kinőttek a **W3C** talajából, a tervek egészen szerteágazóak: a nyomtatásra szánt dokumentumoktól (**SVG Print**) a mobilalkalmazáson át (**SVG Mobile**) egészen a jobb szkriptelhetőségig (a **Document Object Model**lel való jobb együttműködés) mindenre van már javaslat. Ezeket a <http://www.w3.org/Graphics/SVG/> weboldalon magunk is szemügyre vehetjük.

A webes tartalom-előállítás (**Authoring Tool Guidelines**) és a fogyatékos látogatók számára készülő (**Accessibility Techniques**) elképzelések már igazán a jövőbe mutatnak, olyannyira, hogy még piszkoztat sem jelent meg ezekről. Ami viszont a közeli jövőt illeti, az egyik friss újdonság az **SVG-s MMS**. A betűszó feloldása: **Multimedia Message Service**. Eddig pixelgrafikus képeket, videókat küldözgethettünk, most viszont akár SVG fájlokat is. Mi ebben a nagyszerű? Az **MMS**-ért adatforgalom alapján kell általában fizetni, ezért nem mindegy, hogy mekkora az elküldött üzenet. A **Nokia** 2005 májusában adta ki az útmutatót (<http://svg.org/story/2005/7/5/181035/3748>). Mi sem természetesebb, hogy egyre-másra kerülnek az üzletek polcaira az **SVG**-t ismerő mobiltelefonok, mint a **Nokia 7710** vagy a **Sony Ericsson s600i**. Bár ezeknél a készülékeknél jogos a kérdés, hogy minek nevezzelek, mert gyakorlatilag mobiltelefonnal beoltott kézisámítógépek.

## A szabvány

Jelen pillanatban a legtöbb alkalmazás az **1.0-ás SVG**-t sem tudta teljes mértékben megvalósítani, az alkalmazás-

szállítók a szabványkiagyálók után kullognak ebben a tekintetben. A felhasználó számára olyan sok különbség nincs is az **1.0-ás** és az **1.2-es** formátum között, ellentétben például a szoftverek számozásával, ahol a végfelhasználó által látványos fejlesztések történnek. A fájlformátumok kidolgozása nem ilyen, az apró módosítások rejtve maradnak, minden az adott alkalmazáson múlik.

Térjünk rá most már magára a szabványra. Sok más formátumhoz hasonlóan az **SVG** is az **XML 1.0**-ra (<http://www.w3.org/TR/2004/REC-xml-20040204/>) épül, amely mutat ugyan csekély hasonlóságot a webet jelen pillanatban is uraló **HTML** formátummal, de sokkal alkalmasabb programok általi kezelésre, mivel szigorú struktúrát követel meg. A **HTML** leírónyelvet ismerőknek például komoly változás, hogy minden címkénél(tag) kötelező a záróelem. Bővebben a ([http://www.w3c.hu/forditasok/XML\\_10\\_pontban.html](http://www.w3c.hu/forditasok/XML_10_pontban.html)) **W3C** magyar oldalain olvashatunk arról, hogy az **XML** miben tér el a **HTML**-től. Az **SVG**-t a **Flash**-hez hasonlóan kezdték animációkra és interaktív működésre is felkészítették és éppúgy lehet benne raszteres kép is. Egy **SVG** dokumentum ehhez hasonlóan épül fel:

```
<?xml version="1.0"
standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
```

```
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="2cm" height="2cm"
version="1.1"
```

```
xmlns="http://www.w3.org/2000/
svg">
<desc>Egy negyzet
</desc>
<rect x="0.5cm" y="0.5cm"
width="1cm" height="1cm"/>
</svg>
```

Amint a **<desc>** tagek között megadtuk, ez nem tesz mást, mint megjelenít egy négyzetet. Aki az **XHTML** szabványt ismeri, annak ismerősek a szerkezeti elemek, hiszen a technikailag fejlett weboldalak ma már a **HTML 4** helyett az **XHTML**-t részesítik előnyben. Habár a példát bármilyen szövegszerkesztővel elő lehet állítani, az **SVG** összetett nyelv, csak **SVG**-szerkesztővel érdemes nekivágni a komolyabb munkának. Aki mégis szeretné **XML** szinten beleásni magát az **SVG**-be, annak az egyszerű szövegszerkesztő helyett a **KXML Editor** (<http://kxmleditor.sourceforge.net/>) nyújt segítséget. Ezzel a szoftverrel bármilyen **XML** formátumú fájl megjeleníthetünk többféle struktúrában és szerkeszthetjük is.

Ebben a példában csupán egyetlen grafikát leíró taget próbáltunk ki, a **rect**-et, így most tekintsük át, mi mindent lehet **SVG**-ül mondani. A legfontosabb, hogy a **CSS 2**-ből megismert formázások többnyire alkalmazhatók az **SVG** elemekre. De melyek is ezek az elemek? Lehet négyszöget, kört, elipszist, sokszöget és törtvonalat rajzolni. Alapvetően ezekből a geometriai elemekből és a tetszőleges kombinációjukból állnak a bonyolultabb **SVG** rajzok is. Emellett használhatunk szöveget, különféle átmeneteket, szűrőket és átalakításokat végezhetünk, forgathatjuk a koordináarendszert például. A nyelv nagyon sokrétű, így ahhoz, hogy megállapítsuk egy fájlról, hogy van-e benne hiba, a **W3C** kiadta az **SVG Validator**ot, mellyel egyetlen paranccsal kiderül, követi-e a **W3C** specifikációját az adott fájl. Ezt a hasznos **Javaban** írt (azaz szinte minden platformon működő) programot a <http://jiggles.w3.org/svgvalidator/> címről tudjuk letölteni illetve ugyanitt on-line is kipróbálhatjuk.

## A gyakorlat

Most, hogy egészen a mélyére néztünk az **SVG**-nek, váltsunk perspektívát és nézzük meg, hogy ez az elméletben jól

csengő formátum hol tart a gyakorlatban. Ma már szinte mindegyik webböngésző felkészíthető az SVG fájlokra, azonban a *Macromedia Flash*-sel ellentétben az elterjedtsége csekély, nagyon kevés böngészőben van beépített támogatás, az SVG bővítmények (plugin) pedig nem terjedtek el. Ráadásul a natív SVG támogatás a böngészőkben gyenge minőségű, *Firefox*-szal (1.0.4-es változat) a szkriptelhető SVG nem ment, a *Konqueror* (3.3.2) ugyan támogatta, de sok esetben rövid használat után összeomlott. Érdemes mégis foglalkozni az egésszel annak ellenére, hogy jelen pillanatban ilyen csekély az SVG elfogadottsága? Feltétlenül. Az *Adobe* több platformra is fejleszti az SVG pluginjét és SVG-nézegetőjét és minden magára valamit is adó webböngésző gőzerővel dolgozik az egyre tökéletesebb natív SVG támogatáson. Olyan komoly grafikus alkalmazások és cégek sorakoztak fel az SVG mögött, mint a *KDE*, a *Gimp*, a *Mozilla Foundation*, a *Gnome*, az *Adobe Illustrator*ja és még sokan mások. Elsősorban a webes használat tör előre, azonban *Gnome* és *KDE* esetében már a rendszeren belüli használat is lehetséges.

### Látványos SVG a weben

A <http://www.carto.net/papers/svg/samples/> címen számtalan látványos SVG képet, animációt és interaktív alkalmazást találunk, melyek nagyon jól szemléltetik, hogy mire is alkalmas már jelen pillanatban az SVG. A példák többségéhez nem elegendő a böngészők beépített SVG támogatása, érdemes beszerezni az *Adobe* által kibocsátott SVG nézegetőt (<http://www.adobe.com/svg/viewer/install/main.html>), mert így minden korlátozás nélkül úgy tudjuk megnézni az összetett SVG animációkat, ahogy azt a szerző kitalálta. Az *Adobe* csupán egy *Red Hat Linux* alatt használható bétát adott ki, de szerencsére általános célú telepítőprogramja van, így bárhova feltelepíthető. Elméletileg a letöltött *.tar.gz* fájlból kitömörítés után csupán az *install.sh*-t kell elindítani, de *Debian Sid* alatt nem működött, viszont kézzel végrehajtva a fájlban lévő utasításokat, kiválóan be lehetett üzemelni és életre keltek *Firefox*-ban a halott SVG-animációk is. Azonban jó tudni, hogy az *Adobe* SVG megoldásai zárt forráskódúak.

### SVG, WEB és HTML?

Ha a weboldalainkon SVG-t akarunk használni, akkor ugyan használhatjuk a *HTML 4*-es leírónyelvet is, azonban jó ha tudjuk, hogy ahogyan grafikában XML alapú formátumot használunk, ezt megtehetjük az egész weboldalon is. Az SVG-t támogató böngészők ugyanis követik a friss szabványokat, így aztán nem veszítünk látogatót, ha magát az oldalt sem elavult módon építjük fel. Mivel ez a cikksorozat alapjában véve az SVG-ről szól, részletesen nem foglalkozunk az XHTML-el, de magyar nyelven (<http://htmlinfo.polyhistor.hu/xhtml1se/cover.html>) is utánajárhatunk, hogyan kell XHTML oldalakat írni. Kicsit ugyan mellékszálnak tűnhetnek a fentiek, de amennyiben az SVG-t a webfejlesztés szemüvegén keresztül nézzük, kikerülhetetlen az a kérdés, hogy magát az oldalt, ahova a grafika kerül, hogyan készítjük el.

### Átjárás az alkalmazások között

Ha általánossá válik a vektorgrafikus programok körében az SVG támogatás, egy közös nevezővé válik ezen alkalmazások között, hiszen egy nyílt formátum, melyet bárki megvalósíthat szoftverében, nem gátolja jogi problémák. A másik eshetőség, hogy minden nagy szoftvergyártó csavar egy kicsit a formátumon és egymást csak helyel-közzel értő részekre esik szét az SVG-t használók tábora. Jelen pillanatban a linuxos grafikai programokat kipróbálva szerencsére az első változat látszik megvalósulni, bonyolultabb grafikákat betöltve, mentve a sok különféle SVG alkalmazás között semmi probléma nem bukkant föl. Ráadásul már most is lehet például *AutoCad*-ból SVG-be menteni, ami tovább erősíti azt az elképzelést, hogy a csereformátum szerepét is betölti az új szabvány. Idevágó költői kérdésünk: lehet ennél még kényelmesebb is? Azon túl, hogy a vektorgrafikus programok egymás közt remekül eltársalognak SVG-ül, a vektorgrafikát egy csapásra pixelgrafikussá alakíthatjuk, ha a helyzet úgy kívánja. A *GIMP* ugyan elsősorban raszteres képekkel dolgozik, de az SVG-t be tudja olvasni és onnantól kezdve a *GIMP* által támogatott összes formátumban elmenthetjük a képet.

Remélhetőleg a közeljövőben, amikor nem csak a szabad szoftver közössége fogja felismerni az SVG jelentőségét (azért már ma is vannak kivételek), akkor is megmarad ez az átjárhatóság. Mivel az SVG-hez számtalan egyéb webes formátum kapcsolódik (*XHTML*, *CSS 2*, stb), ha elkezdődik például az *Internet Explorer*-ben az SVG támogatása, az új webes technológiákkal is lépést fog tartani a most kissé elmaradt (nem felhasználószámában, hanem szabványkövetésben) böngésző.

A következő részben már SVG-t támogató alkalmazásokkal fogunk dolgozni! Akár *Linux*-ot, akár valamilyen *BSD*-t vagy esetleg *Windows*-t használunk is, kényelmes SVG-szerkesztőkkel fogjuk gazdagítani webes fegyvertárunkat! Addig is figyeljük mobilszolgáltatónk kínálatát, mikor tűnik fel az első SVG-megfelelő mobil *Magyarországon!*



**Novák Áron**

(aaron@szentimre.hu)  
BME-VIK-es gólya,  
műkedvelő rendszer-  
gazda. Jelenleg leg-  
inkább a NetBeans-szel  
és mindenféle hordozható eszközzel  
foglalkozik, legalábbis mindazokkal  
amelyeket meg lehet szólaltatni  
Linux alatt.

### KAPCSOLÓDÓ CÍMEK

SVG példaoldalak:

- [www.carto.net/papers/svg/samples/](http://www.carto.net/papers/svg/samples/)
- [www.w3schools.com/svg/svg\\_examples.asp](http://www.w3schools.com/svg/svg_examples.asp)
- [www.croczilla.com/svg/samples/](http://www.croczilla.com/svg/samples/)
- [www.w3.org/Graphics/SVG/Test/](http://www.w3.org/Graphics/SVG/Test/) – a hivatalos tesztoldal
- [www.taedium.com/svg/](http://www.taedium.com/svg/)
- [www.visionmonster.com/svg/](http://www.visionmonster.com/svg/) – egy egész oldal SVG-ben

SVG leírások:

- [www.w3.org/TR/SVG/](http://www.w3.org/TR/SVG/) – a hivatalos leírás
- [www.w3schools.com/svg/default.asp](http://www.w3schools.com/svg/default.asp)
- [svg.lap.hu/](http://svg.lap.hu/)