

# Google Maps

A Google kiváló webszolgáltatásokat tervelt ki a lelkes fejlesztők számára.

■ Az elmúlt néhány hónapban szemügyre vettünk néhány *webszolgáltatást*. Ez gyűjtőnév mindazokra a módszerekre, amelyekkel az interneten jelen lévő vállalatok nyilvánosságra hozzák adataikat, hogy azokat egyszerűen fel lehessen használni különböző alkalmazásokban. Így például az *Amazon* rendelkezésre bocsátja termékatalógusát, amelynek segítségével online áruházakat vagy árazóprogramokat készíthetünk, az *eBay* lehetővé teszi a keresést árucikkei között (közben persze kínálja is őket), és a *Google* is elérhetővé teszi a keresések eredményeit: meg lehet nézni, lehet őket manipulálni. Minden cég támaszt bizonyos korlátokat arra nézve, hogy mire lehet felhasználni az általuk kínált adatokat, de az egész folyamat a nyíltság és elérhetőség irányába mutat. Ez a nyíltság néha a webszolgáltatások megszokott formáitól meglehetősen eltérő csomagolásban jelenik meg. Vagyis bizonyos vállalatok speciális *library*-k, programkönyvtárak mögé rejtve bocsátják rendelkezésre adataikat, így az ezeket használó alkalmazások nem láthatják előhívásuk pontos részleteit. Erre az egyik legjobb példa – amit ebben a részben részletesen megvizsgálunk – a *Google Maps*, melyet az egyik legjobb és legelőremutatóbb webalkalmazásnak tartok. Nem véletlen, hogy a *Google Maps* az első

között használta az *Ajaxot*; azt a technológiát, amely leírja, hogy hogyan használható a *JavaScript* és az *XML* kombinációja arra, hogy adatokat gyűjtsünk távoli kiszolgálókról, melyekkel weboldalunkat dinamikus módon, újratöltés nélkül frissíthetjük. Ebben a részben elmagyarázom, milyen egyszerű térképeket létrehozni a *Google Maps* programozási felülete (*API*) segítségével. Meggyártunk néhány alapvető térképet, sőt el is helyezzük rajtuk apró jelzőtáblákat a bennünket érdeklő helyekre. Ez építőközként szolgálhat majd saját adatturmixainkhoz (*mashups*) – hogy az egyre népszerűbb szakkifejezést használjam arra, amikor a *Google Maps* használatával jelenítünk meg (például külső adatbázisból kinyert) információkat.

## Az alapok

A *Google Maps* – a legtöbb webalkalmazáshoz hasonlóan – elosztja a munkát az ügyfél (a webböngésző) és a webszerver között. Ám a munka hagyományos felosztása korábban meglehetősen egyenlőtlen volt: szinte a teljes számítási feladat a szerveren terhelte, a böngésző dolga pedig csak a megjelenítés volt. Az *Ajax* változtat ezen a modellen azáltal, hogy olyan *JavaScript* könyvtárakat használ, amelyekkel új és érdekes módon lehet adatokat kezelni, megjeleníteni.

Talán a *Google* egyszer kiad egy saját programozási felületet (*API*), amely lehetővé teszi saját *Ajax* alkalmazások létrehozását is térképes információkkal, a jelenlegi kiadáshoz azonban mindent egyetlen csomag formájában kell telepítenünk. A *Google* rendelkezésünkre bocsát egy *JavaScript* könyvtárat – pontosabban egy linket a *Google* szerverein elhelyezett *JavaScript* könyvtárra, amelyet beilleszthetünk oldalainkba, és segítségével térképeket hozhatunk létre. A térképek megjelenítéséhez tehát ezt a *JavaScript* könyvtárat kell használnunk. Az *API* felhasználóinak nyomon követéséhez és a felhasználás szabályszerűségének biztosítása érdekében a könyvtár elérése csak egy alkalmas kulcs segítségével történhet. Ilyenfajta megszorítást már korábban is láttunk például az *Amazon* webszolgáltatásai között vagy a *Google* fő webszolgáltatását illetően (a keresési eredményekre vonatkozóan); a *Google Maps*-ben használt kulcs azonban más jellegű: ez egyes személynek szól (aki rendelkezik *Google* fiókkal) és egy bizonyos *URL*-re vonatkozik. Ez azt jelenti, hogy a <http://www.example.com>-on működő kulcs nem fog működni a <http://www.example.net> oldalon. A *Google Maps API* használatakor az első lépés annak eldöntése, hogy mely *URL* alatt szeretnénk saját térképeket használni. Elhatároztam, hogy

létrehozok egy új Apache virtuális gazdagépet (*virtual host*) a rendszerem, amelyet *maps.lerner.co.il*-nek nevezek el. Ezután regisztráltam a *Google Maps API* weboldalon ([www.google.com/apis/maps](http://www.google.com/apis/maps)), jelezve, hogy az alkalmazásaim a *maps.lerner.co.il URL* alatt lesznek. Néhány másodperc múlva megjelenik egy üdvözlő oldal az API kulcsommal (amely egy igen hosszú, szöközőkkel szabdalts *ASCII* karaktersorozat) és egy egyszerű induló lappal, amellyel meg tudunk jeleníteni egy térképet. Mínthogy *HTML* alapú alkalmazást tervezzük, érdemes alaposabban megtekintenünk a forrást:

```
<!DOCTYPE html PUBLIC "-//W3C//
↳ DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/
↳ DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/
↳ 1999/xhtml1">
  <head>
    <script
src="http://maps.google.com/
↳ maps?file=api&v=1&key=
â†šABQIAAAAQK9JhAXQ9eq-
↳ G55qgu1EXSCF-BH9Y-SIKCAJU8
↳ YFS_
â†šUTREdFBSS2-11UWY0kxbuv6
↳ argoPyrx3Ytg"
type="text/javascript"></script>
  </head>
  <body>
    <div id="map" style="width:
↳ 500px; height: 400px"></div>
    <script type="text/
↳ javascript">
      var map = new GMap
↳ (document.getElementById
↳ ("map"));
      map.addControl(new GSmall
↳ MapControl());
      map.centerAndZoom(new GPoint
↳ (-122.1419, 37.4419), 4);
    </script>
  </body>
</html>
```

A *HTML* dokumentum a megfelelő *DOCTYPE* megjelenítésével kezdődik,

amelyről az derül ki, hogy szoros értelemben vett (*strict*) *XHTML*. Az *XHTML* csodálatos találmány; biztosítja, hogy a *HTML* kód a szigorú értelemben vett *XML* szabályai szerint van megalkotva. Számos *HTML* oldal nem ragaszkodik ehhez a szabványhoz, így „átmenetinek” (*transitional*) tekinthető, azaz „szabadon értelmezett *XHTML*-nek” - vagy éppen semmilyen szabványhoz sem köthetőnek. Mint-hogy a *Google Maps* próbál a lehető legtöbb böngészőnek megfelelni, szigorúan ragaszkodik az *XHTML*-hez.

A `<head>` címke után látunk egy `<script>` címkét, amely egy *JavaScript* forráskódot tölt le a *Google* webszerveréről, a [maps.google.com](http://maps.google.com)-ról. Ezáltal garantált, hogy mindig a legfrissebb *JavaScript* könyvtárverziót használják a felhasználók. A *Google* azt ígéri, hogy minden *API*-frissítéskor ad egy hónapnyi türelmi időt a fejlesztők számára, hogy a működésképtelenné vált kódrészletek átírassák.

A *HTML* dokumentum szövegtestében egy `<div>` címke következik, melynek azonosítója (*ID*-je): *térkép* (*map*). Ez az a csomópont, amit átadunk a *Google JavaScript* könyvtárának. Ehhez a `<div>` címkéhez tartozik egy stílusdefiníció is, mely szélesség- és magasságértéket tartalmaz (`style: width, height`); ezek határozzák meg a térkép méreteit. Weboldalunk tetszőleges szélesség- és magasságértékkel rendelkező térképet megjeleníthet, lehetővé téve ezzel bármilyen designhoz való alkalmazkodást. A `<div>` címke belsejében találunk rá az egész kód szívére, a három *JavaScript* parancsra.

Először létrehozunk egy *GMap* objektumot. Ahogy ez sejtethető is, ez reprezentál egy egyedi térképet a *Google Maps* világában. A térkép fent említett *ID*-jével (a „*map*” paraméterrel) adjuk meg, hogy mely csomópont tartozzon a *GMap* objektumhoz. (Ha ilyen *ID*-jű dokumentumelem nem létezik, akkor a térkép nem jelenik meg a képernyőn). Ez a fajta összekapcsolás azt is lehetővé teszi, hogy

egynél több térképet is megjelentethetünk egy-egy weboldalon több `<div>` címke megadásával, ha mind-egyiknek saját *ID*-t adunk, és mind-egyiket egy másik *GMap* példánnyal kapcsoljuk.

Ha létrehozunk egy *GMap* példányt, működésének vezérlésére különböző üzeneteket küldhetünk neki. Például adhatunk hozzá egy kinagyítást/kicsinyítést lehetővé tevő vezérlőt. Ebben a dokumentumban például az `addControl()` metódus révén egy kis térképvezérlőt jelenítünk meg, miközben átadunk egy új `GSmallMapControl` példányt.

Ez +/- jellel ellátott gombokkal tud nagyítani/kicsinyíteni, valamint nyilakkal is lehetővé teszi az eltolást (egérrel való húzás nélkül). A *Google* két másik vezérlőtípus is lehetővé tesz, nevezetesen a `GSmallZoomControl`-t (melynek csak +/- nagyító/kicsinyítő gombjai vannak) és a `GLargeMapControl`-t (melynek a `GSmallMapControl` vezérlőelemein kívül van egy olyan gombja is, ami egyből egy adott nagyítási szintre visz át). A vezérlőelemek mindig a térkép bal felső sarkában jelennek meg, és semmi sem gátolja meg, hogy többet is felsorakoztassunk ezen vezérlők közül. Így módon, ha nem vagyunk óvatosak, rettentően ronda térképeket (és weboldalakat) is létrehozhatunk több vezérlő felvitelével.

A térkép és a vezérlőelemek létrehozása után szeretnénk rábírnunk a térképet, hogy emeljen ki egy általunk kívánt pontot. A *Google Map* rendszerében a `GPoint` adatstruktúrával lehet megadni egy pontot, amelyet egy hosszúsági és egy szélességi koordináta képvisel. Bár a földrajzban ezek fokban-percben-másodpercben is megadhatóak, a `GPoint` – érthető okokból – csak lebegőpontos (valós) számokat fogad el. A példánkban szereplő *JavaScript* parancs a következő:

```
map.centerAndZoom(new GPoint
↳ (-122.1419, 37.4419), 4);
```

Garantáljuk weboldalad  
**100%-os rendelkezésre állását.**  
Egyetlen leállás egy hónapban, és visszafizetjük a pénzed.

[www.syrius-software.hu](http://www.syrius-software.hu)

Ez egy centerAndZoom üzenetet küld a térkép objektumnak, mégpedig azaz a felszólítással, hogy az általunk megadott GPoInt köré rajzolja ki a térképet, 4-es nagyítási szinten. A legnagyobb nagyítást az 1-es szint jelenti, a legkisebbet a 15-ös, de 16-18-as szintet is lehet kérni – az azonban csak jelképesen „csomagolva” mutatja a kért helyet. A 4-es szinten jól láthatóak az utcák; remek kiindulási pont lehet a térképeket használók számára.

Fontos tudatosítanunk, hogy a GPoInt objektumok a szélességi és hosszúsági paramétereik révén jönnek létre, és nem fordítva. Ez valószínűleg azért van így, mert a Google szoftvermérnökei x és y koordinátákban gondolkodnak, amely a legtermészetesebb a matematikusok és más természettudósok számára. Azonban a térképészetben az adatok gyakran „hosszúság, szélesség” sorrendben vannak megadva – ilyenkor figyelniük kell a helyes sorrendre, mielőtt ellenőrzés nélkül vakon átadnánk a programnak a paramétereket.

Az alapértelmezett kódrészletben létrehozott GPoInt Palo Altoban van, Kaliforniában, és a koordinátápar feltételezhetően a Google irodáira mutat. Ha más helyet szeretnénk megnézni, egyszerűen írjuk át a számokat. Az Illinois állambeli Skokie-ra például (ahol élek) az alábbi koordinátákkal lehet rámutatni:

```
map.centerAndZoom(new GPoInt
↳ (-87.740070, 42.037030), 4);
```

Az oldal újratöltése után értelemszerűen egy Skokie-ra mutató térkép jelenik meg a Palo Alto-i helyett. Végül megemlítem, hogy a Google lehetőséget nyújt négy nézetmód közötti választásra: forgalom, térkép, műhold és összefésült mód létezik (traffíc, map, satellite, hybrid). Ennek beállítása a jobb felső sarokban megjelenő gombokkal történhet, melyeket az alábbi sorral tudunk életre hívni:

```
map.addControl(new
↳ GMapTypeControl());
```

Bizonyára kitalálták, hogy ez egy addControl() üzenetet küld térkép objektumunknak, mégpedig egy új GMapTypeControl példány átadásával.

## Jelzőtáblák

Végül gondoljuk át, hogy hogyan helyezhetünk el térképünkön jelzőtáblákat. Ezzel egy-egy speciális helyet kiemelhetünk, rámutatva egy Google Maps ikonnal, melyet a felhasználók könnyen észrevesznek. Mi magunk is könnyedén készíthetünk JavaScript kezelőket saját jelzőtábláinkhoz – olyanokat, melyek akkor futnak le (és feltételezhetően változtatnak valamit a térkép megjelenésén), amikor valaki rákattint egy jelzőtáblára. Egy jelző előállításához létre kell hoznunk egy új GMarker példányt, melynek átadunk egy GPoInt objektumot:

```
var myMarker = new GMarker(new
↳ GPoInt(-87.740000,
↳ 42.030000));
```

Ezt a frissen létrehozott myMarker objektumot alábbi módon jeleníthetjük meg a térképen:

```
map.addOverlay(myMarker);
```

Ha ezt a két sort beillesztjük a fenti HTML kódrészlet <script> szekciójába, a frissítéskor azonnal láthatóvá válik a képernyőn egy piros jelzőtábla. Most kezdődik az igazi bűvészműtatvány. Amit eddig csináltunk, az mind JavaScriptben és HTML-ben történt, melyet a webszerver küldött el a böngészőnknek, ez pedig értelmezte és kezelte őket. Ez azt jelenti, hogy ha a HTML kódot nem statikus fájlban tároljuk, hanem dinamikus módon a webszerverrel hozzuk létre, akkor igen elegáns dolgokat művelhetünk a JavaScript révén. Létrehozhatunk például több GMarkers elemet, egyszerűen változókhöz rendelve, majd a térképhez csatolva őket. Ha Google Maps weboldalunk PHP nyelven készül, írhatunk egy rövid programot, amely beilleszti a megfelelő JavaScript kódot, például így:

```
<?php
$a = array(-87.740070,
↳ -87.730000);
$count = 0;
foreach ($a as $v) {
    echo "var myMarker$count =
↳ new GMarker(new GPoInt($v,
42.037030));\n";
    echo "map.addOverlay
```

```
↳ (myMarker$count);\n";
    $count++;
}
?>
```

Ha ezt a részletet beillesztjük oldalunk <script> szekciójába, és átnevezzük az oldalt index.php-re (az index.html helyett), akkor máris láthatunk két jelzőtáblát, azonos szélességi és némi- leg eltérő hosszúsági fokon.

Figyeljük meg, hogy a megfelelő szöveget hogyan juttatja a PHP echo parancsa a HTML dokumentumba. Azt is vegyük észre, hogy pontosvesszőket kell tennünk; egyrészt a JavaScript sorok végére (az idézőjeleken belül), másrészt a PHP utasítások végére (az idézőjeleken kívül). Ezek a kérdések mindig okoznak némi fejfájást, amikor az ember olyan programot ír, ami egy másik programot hoz létre. Végül azt is vegyük észre, hogy miként állítunk elő a ciklusban egy-egy új JavaScript változónevet (myMarker\$count), amivel elkerüljük, hogy ugyanazt a jelzőtáblát állítsuk át (és jelenítsük meg) újra és újra (de végül is csak egy példányban). Ez egy számláló (\$count) segítségével történik, amely biztosítja, hogy minden jelzőtáblához külön változónevet tartozzon.

## Összefoglalás

A Google Maps remek webalkalmazás. A fejlesztők számára ez egyben egy olyan platform is, amellyel mindenféle alkalmazásokat és szolgáltatásokat hozhatunk létre, melyekhez térképek is tartoznak. Különösen jól használható olyan HTML dokumentumok előállítására, melyeket valamely dinamikus programnyelven állítunk elő, futási időben beillesztve a kívánt JavaScript kódot és a megfelelő adatokat, amelyek egy-egy térkép megjelenítéséhez kellenek. A következő részben végigkötjük, hogy ezt miként lehet megtenni. Saját adatforrásunk segítségével létrehozunk egy adatturmixot (mashup) egy testreszabott Google térkép formájában.

Linux Journal 2006., 146. szám

A cikk forrásai:

↳ [www.linuxjournal.com/article/8939](http://www.linuxjournal.com/article/8939)

Reuven M. Lerner