

## 14 WEB ÉS MOBIL ALAPÚ TÉRKÉPEZÉS

Michael P. Peterson, USA

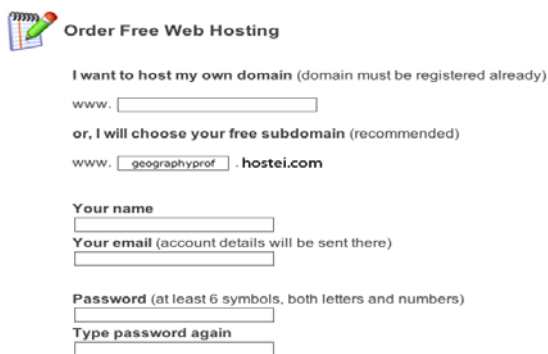
Fordította: Ungvári Zsuzsanna

A felmérés, és a térképkészítés során felhasznált eszközök mindig erősen befolyásolták a térképezés folyamatát. 2005 óta, mikor megjelentek az első alkalmazás-programozási felületek (Application Programming Interface, röviden API), új online- és felhő alapú eszközöket is alkalmazhatunk. Ezeket bárki, aki rendelkezik Internet-kapcsolattal, szabadon elérheti. Ebben a fejezetben megvizsgáljuk a legújabb webes alkalmazás-fejlesztőket, és bemutatjuk, hogyan készíthetők és oszthatók meg az ingyenes programokkal készült térképek.

### 14.1 Bevezetés

A térképek jelentős szerepet játszanak a kommunikációban: a térvonatkozású információkat hatékonyan továbbítják. A térképek segítségével megérthetjük a közvetlen környezetünket, vagyis az észlelt teret. A térképek nemcsak a világról alkotott gondolatainkat befolyásolják, hanem a tájékozódásunkat is. Szorosan összekapcsolnak minket a környezetünkkel. Tulajdonképpen mindenki térképész: az agyunkban mentális térképet tárolunk. Néha szükség van arra, hogy ezt a mentális térképet lerajzoljuk, hogy segítsünk másoknak megtalálni a keresett helyet.

A térképek készítése és a térképkészítés alapjául szolgáló információk elemzése értékes szaktudás lehet sokféle tevékenység során. A térképezés eszközei közül sok az úgynevezett felhőbe került át. A felhő egy távoli számítógép – tárhely – amelyet az interneten keresztül érhetünk el. Ezen tárolhatjuk adatainkat, térképeinket. További előnye, hogy az online térképek bárki számára könnyen elérhetők.



Order Free Web Hosting

I want to host my own domain (domain must be registered already)

www.

or, I will choose your free subdomain (recommended)

www.  .hostei.com

Your name

Your email (account details will be sent there)

Password (at least 6 symbols, both letters and numbers)

Type password again

14.1a. ábra. A <http://000webhost.com> bejelentkezési oldalát mutatja be. Az aldomain szolgáltatás ingyenesen igényelhető (pl.

<http://geographyprof.hostei.com>), ezzel szemben a domain név igényléséért már fizetni kell (pl.

<http://www.peterson.com>). (© 2014 First Class Web Hosting).

A webtérképek tárolásához egy távoli számítógép – szerver – szükséges. Ma már minden számítógép szerverre alakítható, de a legegyszerűbb és legbiztonságosabb megoldás a már meglévő, jól működő tárhelyek használata. A felhő alapú szolgáltatások, mint például az Amazon Web Service és a Microsoft Azure képesek egyszerre akár több millió felhasználót kiszolgálni. Ezek a szolgáltatások felelősek a szabad tárhelyért és a szabad szerver-kapacitásért egyaránt.

Egy másik lehetséges megoldás, hogy a szerveren csak a weboldal adatait tároljuk. Ilyen például 000webhost.com és a podserver.info is. Ezek a szolgáltatásokat nemcsak egyszerű használni, hanem szabad aldomain hozzáférést is biztosítanak, akár 1500 MB-ig. A következő alfejezetben felvázoljuk, hogyan lehet weboldalakat készíteni ezekkel az ingyenes, webes megosztó szolgáltatásokkal.

Account Information	
Domain	geographyprof.hostei.com
Username	a8040697
Password	*****
Disk Usage	0.2 / 1500.0 MB
Bandwidth	100000 MB (100GB)
Home Root	/home/a8040697
Server Name	server33.000webhost.com
IP Address	64.120.177.162
Apache ver.	2.2.13 (Unix)
PHP version	5.2.*
MySQL ver.	5.0.81-community
Activated On	2011-03-10 16:23
Status	Active

14.1b. ábra. Információk a felhasználói fiókról.

### 14.2 Szerverek a felhőben

#### 14.2.1 Hogyan szerezzünk tárhelyet?

A 14.1-es ábra a 000webhost.com szolgáltató bejelentkezési oldalát (14.1a), és egy regisztrált felhasználó adatait (14.1b) mutatja be. A hostei.com oldalán kétféle domaint igényelhetünk. Ha például a <http://geographyprof.com> új domain nevet akarjuk regisztrálni, azért a szolgáltatásért külön fizetnünk kell. Ingyenes szolgáltatás igénybevételéhez fontos, hogy ne regisztráljunk saját domain nevet – ezt a lépést illusztrálja a 14.1a ábra felső sora – hanem csak egy szabad aldomain nevet (subdomain) kell választanunk (pl. <http://geographyprof.hostei.com>). Ilyenkor a linkben látszani fog a tárhely-szolgáltató neve is.

Új, ingyenes aldomainnév regisztrálásához élő e-mail cím szükséges. A felhasználói fiók (Account Information) oldal adatainál látható az a webcím, amelyet a szolgáltató kiutalt számunkra pl. <http://geographyprof.hostei.com> (vagy <http://64.120.177.162>). Minden domain-hez 1500 MB tárhelyet biztosít. Az felhasználói fiók adatainál az

Apache webservert elérhetőségéről, és más online eszközökről – beleértve a PHP-t és a MySQL-t is – információkat nyerhetünk.

A legtöbb webes szolgáltatáshoz egy grafikus felület is tartozik, amelyen a tárhely beállításait könnyen kezelhetjük. Ezt vezérlő oldalnak hívjuk (14.2 ábra). A szolgáltatás általában képes e-mailek fogadására és küldésére, fájlok szerkesztésére, folyamatok ütemezésére, és a fiók kezelésére. A rendelkezésünkre álló eszközök mindegyike egy-egy open source (ingyenes, nyílt forráskódú) projekt eredményeként született meg. Ezeket a szoftvereket programozók egy-egy kisebb csoportja hozta létre, és fejleszti. A vezérlő oldalon a File Manager-rel (fájlkezelő) tudunk fájlokat feltölteni a szerverre. A MySQL-t és a phpMyAdmin felületet adatbázisok építésére használhatjuk. A legtöbb weboldal-szolgáltatás vezérlő oldala igen hasonló az említetthez.

A 14.3. ábra a File Manager ablakot mutatja be; itt tölthetünk fel új fájlokat, illetve szervezhetjük őket mappákba. Lehetőségünk van áthelyezni, törölni és átnevezni is őket. A listanézetben a fájl nevét, típusát, méretét láthatjuk, valamint hozzáférhetünk a fájlbiztonság (Perms) beállításaihoz is (tulajdonos, csoport, engedélyek). A „Mod Time” jelzi, mikor változott meg utoljára a fájl tartalma. A fájl közvetlenül szerkeszthetővé válik az ablakban, ha a fájlnevet után az „Edit” gombra kattintunk.



14.2. ábra. A képen az említett szolgáltatás kezelőpanelje látható, amelyről elérhetjük a különféle eszközöket. A leghasznosabb talán ezek közül a File Manager, amellyel a távoli szerverre tölthetjük fel saját állományainkat.




14.3. ábra. A File Manager menü. Itt lehet feltölteni, illetve szerkeszteni a fájlokat. Minden fájl, amely a webre kikerül, a public\_html mappában kell, hogy kerüjön.

A „public\_html” mappa az a könyvtár, ahol a webre kikerülő anyagainkat tároljuk. Ha azt akarjuk, hogy egy HTML fájl elérhető legyen az interneten keresztül, ebbe a mappába kell kerülnie. Általában ez a mappa tartalmazza az index.htm (index.html, vagy index.php) fájlokat; ezt az oldalt érjük el először, ha egy weboldalra hivatkozunk például így:

<http://geographyprof.hostei.com/CloudMapping/>. A böngésző a megadott domain név alatt létező „public\_html” mappában keres egy CloudMapping-nek nevezett almappát, ezen belül pedig automatikusan az „index” nevű fájlt. A böngésző címsorában csak ezt látjuk: [http://geographyprof.hostei.com/Online\\_Mapping/](http://geographyprof.hostei.com/Online_Mapping/) vagy [http://geographyprof.hostei.com/Online\\_Mapping/index.htm](http://geographyprof.hostei.com/Online_Mapping/index.htm)

Alapesetben az index.htm fájl tartalmazza a mappában lévő egyéb fájlok elérési útját. Az index.htm fájlban a lenti példában viszonylag egyszerű a felépítése – a címsort beágyazott linkek követik. Ebben a fájlban megadtunk egy képet a weboldal tulajdonosáról, és hallgatók honlapjának linkjeit is kitettük, ezt a 14.4 ábra illusztrálja.

**Exercises for Mapping in the Cloud**



Michael Peterson's page

**Student Pages**  
[V. Alapo](#) | [K. Edwards](#) | [G. Elliott](#) | [J. Krajewski](#) | [H. Maezawa](#) | [B. Reeves](#) | [J. Rumpfelt](#) | [J. Schiermeyer](#) | [Y. Shi](#) | [K. Smith](#)

- Ch 4: [Map Gallery](#)
- Ch 6: [Online Street Map](#)
- Ch 8: [JavaScript](#)
- Ch 10: [Map Mashups](#)
- Ch 12: [Point Mashups](#)
- Ch 14: [Line and Polygon Mashups](#)
- Ch 16: [Layer Mashups](#)
- Ch 18: [php and MySQL Mashups](#)
- Ch 20: [Local Mapping](#)
- Ch 22: [Animated Maps](#)

```

<html>
<head></head>
<body bgcolor="#CCCCFF">
<h2> Exercises for <i>Mapping in the
Cloud</i> </h2>
<img src=peterson.jpg height=150><b>
Michael Peterson's page
</b><p>
<p>
<b> Student Pages </b><br>
<a href=http://victoriaA.site88.net> V. Alapo </a> | |
<a href=http://mapsarefuntoo.web44.net> K. Edwards </a> | |
<br> <hr>
<ul>
<li>
Ch 4: <a href=http://maps.unomaha.edu/onlinemapping/
code04.zip>
Map Gallery </a><br>
</li>
Ch 6: <a href=code06.zip> Online Street
Map</a><br>
</li>
Ch 8: <a href=http://maps.unomaha.edu/onlinemapping/
code08.zip>
JavaScript </a><br>
</li>
</ul>
<hr>
</body>
</html>

```

14.4. ábra. Egy index.htm fájl szerkezete, és a böngészőben látható kép (az előző oldalon). Ezen a honlapon egy képet, és linkeket találunk az érdeklődők.

### 14.2.2 A HTML nyelv

A HTML a web alap építőköve, a nyelv, amely lehetővé teszi számunkra információk közlését weboldalakon keresztül. A HTML nyelv a szkript nyelvekkel, mint a JavaScript vagy a PHP, együtt alkalmazható. Ahhoz, hogy térképeket osszunk meg az Interneten, elengedhetetlen egy alapvető HTML ismeret.

A HTML nyelv vezérlő szerkezetekből áll, ezeket angolul tag-eknek hívjuk. Ezekkel a tag-ekkel lehet megadni az oldal kinézetét (Willard 2009). A „<” és a „>” relációs jelek közé kell elhelyezni a megadott szöveges kódokat, ezek adják meg, hogyan jelenjen meg a honlap a böngészőben. Vannak olyan HTML tag-ek, amelyekkel például linkeket lehet megadni dokumentumokhoz, vagy grafikus állományokat megjeleníteni. HTML állományok megnyitásához és szerkesztéséhez egy hagyományos szövegszerkesztő is használható, pl. Windows alatt a Jegyzet-tömb, vagy Macintosh rendszer alatt pedig a TextEdit a megfelelő beállításokkal (a kódszerkesztést igen megkönnyíti a Notepad++ ingyenesen letölthető kódszerkesztő – a fordító). Ellentétben a fejlett szövegszerkesztő programokkal pl. MS Word, ezekben nem formázzuk meg a szöveget. Ha létrehoztuk a fájlt, az megnyitható bármely böngészőben, például Explorer-ben, Firefox-ban, vagy Chrome-ban.

Minden HTML fájl a <html> tag-gel kezdődik, és a </html> tag-gel végződik. Ebben a „/” jel az adott kód-részlet végét mutatja. Például a <h1> parancs mögé a weboldal címét szoktuk elhelyezni, amelyet a </h1> parancs zár.

Az <img> tag-ben képeket és térképeket is megjeleníthetünk. Ebben a tag-ben használható fájlformátumok a következők: GIF, JPG (vagy JPEG), vagy PNG, ezek a böngészők által felismert szabványos fájlformátumok. Az <img> tag-ben az „src” tulajdonság adja meg a beillesztendő fájl elérési útját. Fontos, hogy ne csak a fájl neve szerepeljen az elérési útban, hanem a fájl kiterjesztése is. A következő példában ez a GIF. Az <img> parancsban a kép tulajdonságai megváltoztathatók, mint a kép mérete, vagy az elhelyezkedés az oldalon. Nem minden HTML tag-nek van záróeleme, ilyen az <img> valamint a <hr> (vízszintes elválasztó vonal) is.

 vagy  


Az <embed> tag-gel grafikus fájlokat jeleníthetünk meg, amelyek nem GIF, JPEG vagy IMG formátumúak. Az elfogadott formátumok lehetnek például Adobe PDF, Flash, SVG, vagy QuickTime fájlok.

A beágyazásuk hasonló az előzőhöz:

<embed src="http://maps.unomaha.edu/Cloud\_Mapping/Chapter4/MapExample4.pdf" – width="500" height="389">

Az <a> tag arra szolgál, hogy a linkeket tegyük ki az oldalunkra.

<a href=http://maps.unomaha.edu/Cloud\_Mapping/Chapter4/MapExample4.pdf> Click for PDF file</a>

Ha a linket egy képfájl köré tesszük, akkor a képre kattintva nyílik meg a kívánt weboldal.

### 14.2.3 A JavaScript

Önmagában a HTML egy egyszerű lapleíró nyelv. Ha a HTML-t JavaScript-tel kombináljuk, végrehajtható programkódokat szúrhatunk be az oldalba, és ezáltal dinamikussá tehetjük azt (W3Schools.com, 2011). Mára a programozás nemcsak az informatikusok kiváltsága, hanem hála az ingyenesen elérhető online tananyagoknak, bárki könnyen elsajátíthatja ezt a különleges gondolkodásmódot. A kódolást bárki megtanulhatja, úgymint az írást-olvasást. A legnépszerűbb ingyenes, online tananyagok közé tartozik a CodeAcademy, és a KhanAcademy (ezek angol nyelvűek). A következő példával röviden ismertetjük, hogyan lehet a JavaScriptet használni.

CODE	Result
<html>	x=4
<body>	
<script type="text/javascript">	
var x = 2 * 2	
document.write("x = ", x)	
</script>	
</body>	
</html>	

14.5. ábra. Egy olyan JavaScript függvény, amelyet egy HTML dokumentum <body> elemébe ágyaztak.

A függvények a JavaScript alapvető építőkövei. Egy függvényben utasítások sorozatát hívjuk meg azért, hogy elvégezzünk egy feladatot. A függvényeket általában a HTML dokumentum fejlécében adjuk meg. Ebben az esetben a függvényeket azelőtt definiáljuk, mielőtt bármilyen tartalom megjelent volna az oldalon. A 14.6-os ábra egy egyszerű függvényt mutat be a HTML dokumentum fejlécében. Ezt a függvényt csak később, a HTML dokumentum **<body>** elemében hívjuk meg. A **square** nevű függvénynek (négyzetre emelés) egy argumentuma van, ez a **number**. A **number** egy változó (ebben az esetben egy szám), amelyet a függvényben felhasználunk, ennek az értéke határozza meg a függvény visszatérési értékét. A függvény mindössze egy műveletet tartalmaz, ez a négyzetre emelés. Az eredmény az argumentumként megadott szám négyzete. Végül a HTML dokumentumban kiírjuk az eredményt **document.write()** paranccsal.

JavaScript kódokat a saját gépünkön, vagy akár másik a számítógépen, szervereken, külső fájlokban is származhatunk, ezek meghívását a 14.7 ábra illusztrálja.

Code	Result
<code>&lt;head&gt;</code>	The function returned 25.
	All done.
<code>&lt;script LANGUAGE="JavaScript"&gt;</code>	
<code>function square(number) {</code>	
<code>return number * number</code>	
<code>}</code>	
<code>&lt;/script&gt;</code>	
<code>&lt;/head&gt;</code>	
<code>&lt;body&gt;</code>	
<code>&lt;script&gt;</code>	
<code>document.write("The square of 5 is ",</code>	
<code>square(5), ".")</code>	
<code>&lt;/script&gt;</code>	
<code>&lt;P&gt;All done.&lt;/P&gt;</code>	
<code>&lt;/body&gt;</code>	

14.6. ábra. A függvény visszatérési értéke a paraméterként adott változó négyzete.

Code	Result
<code>function square(number) {</code>	A separate file
<code>return number * number</code>	called "common.js"
<code>}</code>	
<code>&lt;head&gt;</code>	The function
	returned 25.
	All done.
<code>&lt;title&gt;Referencing a file of functions&lt;/title&gt;</code>	
<code>&lt;script src="common.js"&gt;</code>	
<code>&lt;/script&gt;</code>	
<code>&lt;/head&gt;</code>	
<code>&lt;body&gt;</code>	
<code>&lt;script&gt;</code>	
<code>document.write("The square of 5 is ",</code>	
<code>square(5), ".")</code>	
<code>&lt;/script&gt;</code>	
<code>&lt;p&gt;All done.&lt;/p&gt;</code>	
<code>&lt;/body&gt;</code>	

14.7. ábra. A kódrészlet a JavaScript kód külső fájlban történő elhelyezését mutatja. Erre a `common.js` fájlra így hivatkozunk `<script src="common.js">`.

Így működnek az API-k is. Egy központi szerveren a JavaScript kód tartalmazza az összes, működéshez szükséges függvényt.

A JavaScript kódot vagy a HTML fájl fejlécébe ágyazzuk be, vagy egy külső fájlban tároljuk. Ekkor a `<script>` tag `src` tulajdonságban meg kell adni a külső fájl forrását is. A 14.7 ábra bemutatja, hogy hívjuk meg a `common.js`-nek nevezett fájlt a fejlécben. Külső fájlban elhelyezett JavaScript program nem tartalmazhat HTML kódokat.

Code
<code>&lt;head&gt;</code>
<code>&lt;title&gt;Google Maps JavaScript API</code>
<code>Example&lt;/title&gt;</code>
<code>&lt;script type="text/javascript"</code>
<code>src="http://maps.google.com/maps/api/js?sen</code>
<code>sor=false"&gt;</code>
<code>&lt;/script&gt;</code>
<code>&lt;/head&gt;</code>

14.8. ábra. A Google Maps API beágyazását mutatja a HTML dokumentumba. Itt a `sensor` változó értéke hamis. Ha igazra állítjuk, akkor a mobil készüléken meghatározhatjuk az aktuális pozíciókat.

A 14.8 ábra a Google Maps API hivatkozását mutatja. A Google Maps API-ban írt kódok a felhasználó gépen futnak le, anélkül, hogy szükséges lenne feltölteni őket egy központi szerverre. Bár megjegyzem, a honlapot csak akkor fogják mások is látni, ha a weboldalunk egy szerveren elérhető.

### 14.3 A Google Maps API

A Google Maps API tulajdonképpen egy JavaScript függvénykönyvtár, és tartalmazza az összes, a térkép megjelenéséhez és működéséhez szükséges függvényt, úgy mint a méretarány, a pozíció, a vektoros elemek, pl. pontok, vonalláncok (polyline-ok) és felületek (polygon-ok) és a kapcsolódó címkék leírása. A Google Maps API használata lényegében ingyenes, a hozzáférésért nem kell fizetnünk. A térkép használata viszont korlátozott: 90 egy-mást követő nap alatt maximum 25 000 térképet tölthetünk le. Egy térkép betöltése egy térkép megjelenítését jelenti a Google Maps API-ban. Egy átlagos Google Maps API felhasználó nehezen lépi túl ezt a korlátot, akkor is, ha az oldalt sokan látogatják. A napi látogatószám akár követhető is. Ha túlléptük a keretet, a Google regisztrációra kötelez minket, és 0,5 USD megfizetésére 1000 térkép letöltéseként. Például, ha százazerszer töltik le az



oldalt kilencven nap alatt, akkor 37,5 USD-t kell fizetnünk a következő hónapban (75000÷1000·0,5).

Egyéb, speciális korlátozások a Google Maps API-ben.

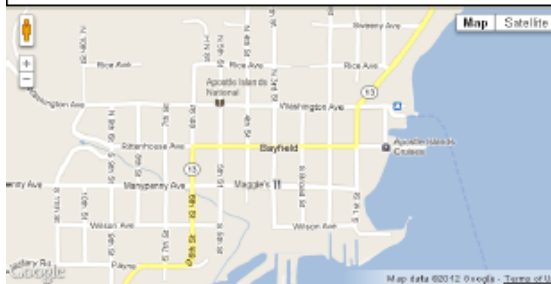
- Irányok: szöveges útleírás esetében max. 2500 kérés naponta.
- Távolság mátrix: eredményül a két pont távolságát, és a megtételéhez szükséges időt adja. Lekérdezésenként max. 100 elem, és max. 2500 lekérdezés naponta.
- Magasságok: egy pont magassága. Max. 2500 kérés naponta, kérésenként max. 512 pont magassága.
- Geokódolás: a címeket szélességi és hosszúsági koordinátapárrá konvertálja. Max. 2500 darab naponta.
- Helyek: üzleti létesítmények, és egyéb érdekes pontok. Ehhez API kulcs szükséges, és max. 1000 kérés küldhető naponta a szervernek.

A Google Maps API kulcs egy numerikus kód, amellyel a felhasználó regisztrálja a honlapját a Google rendszerében. Csak akkor szükséges, ha a felhasználó túllépi a korlátozásokat, vagy igénybe veszi a „Helyek” szolgáltatást.

A 14.9-es ábra egy JavaScript kódot mutat, amellyel egyszerű Google Maps API térkép meghívható, és megadható egy választott nézet. A nagyítás szintje 0 és 21 között változtatható, ebben a példában 15. A térkép közép-pontját szélességi és hosszúsági koordinátapárral állítottuk be. A kiválasztott térképi nézet a ROADMAP (utca-térkép). Ezeket a kezdeti beállításokat az `initialize` függvényben kell megadni, és az oldal betöltésével együtt (body onload esemény) hívódik meg.

Egyszerű változtatás hajtható végre a kódban, ha új, általunk választott szélességgel és hosszúsággal akarjuk helyettesíteni a meglévőt. Egy földrajzi pont koordinátáinak meghatározása többféle módon, illetve szoftverben is történhet:

```
<html>
<head>
<script type="text/javascript"
src="http://maps.google.com/maps/api/js?sensor=false">
</script>
<script type="text/javascript">
function initialize() {
var latlng = new google.maps.LatLng(46.810928, -
90.817981);
var myOptions = {
zoom: 15,
center: latlng,
mapTypeId: google.maps.MapTypeId.ROADMAP
};
var map=new
google.maps.Map(document.getElementById("map_canvas"),
myOptions);
}
</script>
<title></title>
</head>
<body onload="initialize()">
<div id="map_canvas" style="width:600px; height:300px">
</div>
</body>
</html>
```



14.9. ábra. A Google Maps alaptérképének kiválasztása, és a térképi nézet középpontjának, illetve a hozzá tartozó nagyítási szintnek a beállítása.

- A Google Maps-ben jobb egérgomb kattintással (Mac rendszeren CTRL+ kattintás jobb egérgombbal) ki kell választani a „Mi van itt?” (What’s here?) opciót, és a bal felső sorban feltűnik a koordináta.
- Jobb kattintás az egérgombbal a MapQuest-ben, és előreugró ablakban megjelennek a koordináták.

- A Bing Maps-en a szélességi és hosszúsági koordináták jobb egérgomb kattintással előjönnek.
- Ahhoz, hogy megjelenítsük Google Earth-ben fokban a koordinátákat, az Eszközök/Opciók-ban kell a Szélesség/hosszúság megjelenítését decimális fokra állítani.
- Végül, többféle oldalon lehet szélességeket és hosszúságokat keresni. Erre példa a Google Maps-et felhasználó <http://findlatituteandlongitude.com>

A Google Maps-ben többféle térkép közül lehet választani, amely az alapbeállítások része:

- **MapTypeId.ROADMAP** esetén az utcatérképet mutatja
- **MapTypeId.SATELLITE** a Google Earth műholdképei látszódnak
- **MapTypeId.HYBRID** a műholdkép és az utcatérkép keveréke
- **MapTypeId.TERRAIN** a domborzati térképet láthatjuk

A kezdeti nagyítási szint (zoom) szintén változtatható. 0 érték esetén a teljes Föld kis méretarányú képét láthatjuk. Ahogy a nagyítás szintje változik, a térkép méretaránya is növekszik. Általában 20. nagyítási szint részletessége elegendő, de a világ némelyik részén, több nagyítási szint is előfordul.

#### 14.4 A pontok, a vonalak, a felületek és a rétegek beállításai

Minden térkép pontokból, vonalakkól és felületekből áll. A térkép rétegei ezekből az alkotóelemekből épülnek fel, ezen alapszik a földrajzi információs rendszer modellje is (Geographic Information System=GIS). Ebben a részben megismerkedünk, hogyan adhatók hozzá az egyes térképi rétegek a Google Maps térképhez.

### 14.1.1 Pontok

Az alapértelmezett helyjelző szimbólum egy feje tetejére állított esőcsepp, de számos, másik ikon is elérhető a tárházban. Lehetséges 32x32 pixel méretű, PNG formátumú képeket is szimbólumként alkalmazni. A helyjelzők lehetnek statikusak, vagy interaktívak. Az interaktív helyjelzőkre kattintva akár egy buboréklablak (infoWindow) is előhívható.

A 14.10-es példánál a **contentString** egy szöveges változó, ennek a tartalma jelenik meg az **infoWindow**-ban, vagyis a buborékban. Ezt a **google.maps.event.addListener**-ben, mint eseményt tudjuk definiálni. A buboréklablakban nemcsak szöveget, hanem képet és videót is megjeleníthetünk.

Abban az esetben, ha a pontok nagy számban fordulnak elő, és ezek adatait rendszeresen frissítik, gyakran használják az RSS webes formátumot (Really Simple Syndication). Az RSS csatorna használatának rengeteg előnye van. Az RSS csatorna közzétevője folyamatosan frissíti a tartalmat, és a felhasználók mindig a legújabb információhoz juthatnak. Ezt a szabványos formátumot többféle programban is meg lehet tekinteni.

A KML formátumot (Keyhole Markup Language) eredetileg a két- és háromdimenziós tér leírására fejlesztették ki a Google Earth-ben. Mára a nyílt szabvánnyá vált (ez az OpenGIS® KML Encoding Standard (OGC KML)), amelyet az Open Geospatial Consortium (OGC) tart karban. A formátum lehetővé teszi helyjelzők, képek, felületek, és 3D modellek definiálását. A pontokat mindig egy koordinátpárral adjuk meg. Az interneten rengeteg KML példafájl elérhető.

```
function initialize()
{
  var myLatLng = new google.maps.LatLng(41.258531, -96.012599);
  var myOptions = {
    //this defines the zoom level
    zoom: 14,
    center: myLatLng,
    mapTypeId: google.maps.MapTypeId.ROADMAP
  }
  var map = new
  google.maps.Map(document.getElementById("map_canvas"),
  myOptions);
  var contentString="Cartography & GIS Laboratory-br-University of
  Nebraska at Omaha"
  var infowindow = new google.maps.InfoWindow(content:
  contentString);
  var marker = new google.maps.Marker({
  position: myLatLng,
  map: map,
  title: "Cartography and GIS Laboratory"
  });
  google.maps.event.addListener(marker, 'click', function(){
  infowindow.open(map, marker);
  });
}
```



14.10 ábra. Egy olyan helyjelzőt mutat, amelyre ha rákattintunk, egy buboréklablakban jelenik meg további információ.

A **google.maps.KmlLayer** függvénye képes olvasni KML-ként formázott RSS csatornát is egy megadott HTTP címen. Ezzel a technikával igen gyorsan jelennek meg a térképek. Ebben az esetben viszont kompromisszumot kell kötnünk, mivel kevésbé befolyásolható a KML fájlban meghatározott szimbólumok megjelenési stílusa.

A 14.11-es ábrán egy naponta frissülő RSS csatorna alkalmazást láthatunk, amely az elmúlt hét nap földrengéseit mutatja. Minden helyjelzőre rákattintva előhívható a buboréklablak, amelyben több információ olvasható a földrengésről. A kód azt is bemutatja, hogy válasszuk ki a Föld egy bizonyos részletét.

### Earthquakes in the past week



```
var georssLayer = new
google.maps.KmlLayer('http://earthquake.usgs.gov/earthquakes/
catalogs/eqs7day-M2.5.xml');
Earthquakes in the past week

var ctaLayer = new var ctaLayer = new
google.maps.KmlLayer('http://earthquake.usgs.gov/earthquakes/
catalogs/eqs7day-M2.5.xml', {preserveViewport:true});
ctaLayer.setMap(map);
```

14.11 képen az Amerikai Egyesült Államok Geológiai Szolgálat (USGS) által karbantartott RSS csatorna implementálását látjuk KML formátumban. Minden helyjelző az elmúlt heti földrengések kipattanási helyét adja meg, illetve a helyjelzőre kattintva olvashatóvá válnak a részletes információk a rengésről. Az alsó térképen nem adtuk meg a térképi nézet középpontját és a nagyítási szintet, ezért ebben az esetben megduplázva látjuk a térképet. A felső térképen ellenben ezeket az adatokat megadtuk.

#### 14.4.2 Vonalak

A Google Maps Polyline (több egyenes szakaszból álló vonallánc) függvényével rajzolhatunk vonalláncokat a Google Maps API-ban. Különböző megjelenítési beállításokat választhatunk, például a vonal színe, átlátszósága, vastagsága. Vonál ábrázolása esetén a térkép középpontjával általában a vonal matematikai közepét választjuk.



14.12. ábra. A képen egy négy töréspontból álló vonalláncot adtunk meg.

Hasonlóan más webes, térképi alkalmazásokhoz, a Google Maps is a szférikus Mercator vetületet (szögtartó hengervetület) használja. A valóságban a gömb alapfelületen két pont közötti legrövidebb szakasz az ortodróma, ez egy görbeként képeződik le a térképre (14.13 ábra vörös vonal). Mivel az ortodróma gömb alapfelületen egy gömbi főkör, ezért a Földet két félgömbre osztja. A Google lehetővé teszi ennek a görbének a megjelenítését is, a **geodesic** változó igaz értékre állításával. Egyébként a repülőgépek is az ortodróma mentén repülnek. Ha a térképen, vagyis a síkban jelöljük ki két pont között a legrövidebb távolságot, ez egy egyenesként jelenik meg, de a valóságban – mivel gömb alapfelületen mozgunk – távolabb esik az egyenes vonal mentén egymástól ugyanaz a két pont (fekete).

```
var flightPlanCoordinates = [
  new google.maps.LatLng(37.772223, -
    122.214897),
  new google.maps.LatLng(21.291982, -
    157.821856),
  new google.maps.LatLng(-18.142599, 178.431),
  new google.maps.LatLng(-27.46758, 159.027892)
];
var flightPath = new google.maps.Polyline({
  path: flightPlanCoordinates,
  strokeColor: "#FF0000",
  strokeOpacity: 1.0,
  strokeWeight: 3
});
```



14.13. ábra. A geodesic változó két értéke: ha igaz, a két pontot egy gömbi főkörrel kötjük össze, ennek a térképi vetülete pirossal látható. Ha hamis, a két pont között egyenest rajzol a térképen. A szögtartó Mercator vetület tulajdonságai miatt a gömbi főkör hosszabbnak tűnik. (© 2014 Google)

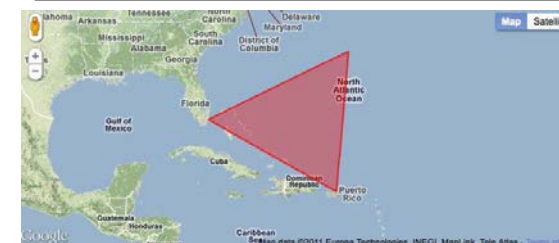
#### 14.4.3 Felületek

A polygonok, vagy másképpen felületek, valójában zárt vonalak. A körvonal koordinátapárok sorozatából áll, ahol az első és utolsó pont megegyezik. A vonalakhoz képest két új tulajdonságot adhat meg a felhasználó (**google.maps.Polygon**), ez a kitöltés és annak átlátszósága.

A 14.14-es képen a Bermuda-háromszög látható az Atlanti-óceánban. A háromszög megadásához négy pontot kellett megadni, amelyek egy **triangleCoord**-nak nevezett tömbben tárolódnak. A **google.maps.Polygon**-nal a koordinátákon kívül az előbb ismertetett tulajdonságok megadhatók (**strokeColor**=körvonal színe,

**strokeOpacity**=körvonal átlátszósága, **strokeWeight**=körvonal vastagsága, **fillColor**=kitöltés színe, **fillOpacity**=kitöltés átlátszósága).

```
<script type="text/javascript">
function initialize() {
  var myLatLng=new
  google.maps.LatLng(24.886436490787712,-
    70.2685546875);
  var myOptions = {
    zoom: 5,
    center: myLatLng,
    mapTypeId: google.maps.MapTypeId.TERRAIN
  };
  var map = new
  google.maps.Map(document.getElementById("map_canvas"),
    myOptions);
  var triangleCoords = [
    new google.maps.LatLng(25.774252, -80.190262),
    new google.maps.LatLng(18.466465, -66.118292),
    new google.maps.LatLng(32.321384, -64.75737),
    new google.maps.LatLng(25.774252, -80.190262)
  ];
  var bermudaTriangle = new
  google.maps.Polygon({
    paths: triangleCoords,
    strokeColor: "#FF0000",
    strokeOpacity: 0.8,
    strokeWeight: 2,
    fillColor: "#FF0000",
    fillOpacity: 0.35
  });
  bermudaTriangle.setMap(map);
}
</script>
```



14.14. ábra. A Google Maps térképére rajzolt zárt polygon. Lehetséges különféle megjelenésű polygonok definiálása. (© 2014 Google)



#### 14.4.4 A rétegek

Az előzőekben áttekintettük, hogyan lehet vektoros rétegeket, pl. pontokat, vonalláncoakat, és felületeket definiálni. Most raszteres fedvények (overlay), úgymint légifotók, úrfelvételek, vagy szkennelt térképek megadását nézzük meg. A módszer előnye, hogy könnyen a Google Maps térképre feszíthető a saját térképünk fedvényként, ehhez nem szükséges adatkonverzió, csak a sarokpontok koordinátapárjainak beírása. Ha túl nagy méretű a raszteres kép, ekkor önálló elemekként is megjeleníthetők, kisebb szeletekbe darabolva; ezeket tile-oknak, magyarul csempéknek nevezzük.

14.15-ös ábra egy szkennelt, JPEG formátumban elmentett térképet mutat. A délnyugati és az északkeleti sarok szélességét és a hosszúságát meghatároztuk, majd az **imageBounds** objektum segítségével megadtuk. A példában az **oldMap** objektumnak két paramétere van: a sarokpont-koordinátái és a kép elérési útja.

```
function initialize() {
    var newark = new google.maps.LatLng (40.740, -74.18);
    var imageBounds = new
    google.maps.LatLngBounds(
    new google.maps.LatLng (40.712216,-74.22655),
    new google.maps.LatLng (40.773941,-74.12544));
    var myOptions = {
    zoom: 12,
    center: newark,
    mapTypeId: google.maps.MapTypeId.ROADMAP
    }
    var map=new
    google.maps.Map(document.getElementById("map_canvas"), myOptions);
    var oldmap = new google.maps.GroundOverlay(
    "http://www.lib.utexas.edu/maps/historical/newark_nj_1922.jpg",
    imageBounds);
    oldmap.setMap(map);
}
```



14.15. ábra. Egy szkennelt, JPEG formátumú térkép, mely Newark, NJ területét ábrázolja. A kép a Texasi Egyetem Könyvtárából származik.

#### 14.5 Mobil alapú térképezés

A földrajzi helyzetünket felismerő eszközök egyre elterjedtebbé válnak. Gyakorlatilag minden mobiltelefon tudja határozni az aktuális helyzetünket néhány méteren belül. Az okostelefonok ezen kívül képesek az aktuális pozíciókat térképen is megjeleníteni. Az Apple iOS és a Google Android rendszerű táblagépek előnye az okostelefonokkal szemben, hogy sokkal nagyobb a kijelzőjük. Többféle mobilkészülék van, és többféle módja létezik a helyzet-meghatározásnak is. A W3C (World Wide Web Consortium) egy ingyenes helymeghatározó API-t hozott létre, amelyet szabványként ajánlja. Majdnem az összes böngésző támogatja, az API többféle módszert is használ a pontos hely meghatározásához a számítógépeken, vagy a mobil eszközökön.

A földrajzi helymeghatározó rendszer (Global Positioning System=GPS) használata csak az égbolttal való folyamatos, tiszta összelátás esetén működik. Városi környezetben leginkább a Wi-Fi hálózaton vagy a mobilátjátszó-állomásokon hajtunk végre háromszögelést. A helymeghatározó szoftvereket például a bostoni székhelyű Skyhook is használ. Ez a hálózat magába foglal több mint 250 millió Wi-Fi hozzáférési pontot, és mobilátjátszó toronyt. Az adatbázis fejlesztéséhez a Skyhook világszerte mérőpontokat hozott létre, hogy meghatározzák a Wi-Fi és átjátszó állomások pontos koordinátáit.

A **navigator.geolocation.getCurrentPosition(function(position))** függvény meghívásával az hálózathoz csatlakozó eszköz földrajzi koordinátáit kapjuk eredményül. Ha a jelenlegi pozíciót nem lehet GPS-szel meghatározni, akkor az API a vezeték nélküli internethálózaton alapuló háromszögeléssel dolgozik. Ebben a példában a helyjelzőre kattintva, egy infoWindow buboréklakban jelenik meg a pozíció.



```
if(navigator.geolocation) {
    browserSupportFlag = true;
    navigator.geolocation.getCurrentPosition(function(position) {
        initialLocation = new google.maps.LatLng(
            position.coords.latitude,position.coords.longitude);
        contentString = "Pt: "+position.coords.latitude+", "+position.coords.longitude;
        map.setCenter(initialLocation);
        infowindow.setContent(contentString);
        infowindow.setPosition(initialLocation);
        var marker = new google.maps.Marker({
            position: initialLocation,
            map: map,
            title:"Hello World!"
        });
        google.maps.event.addListener(marker, 'click', function() {
            infowindow.open(map,marker);
        });
    }, function() {
        handleNoGeolocation(browserSupportFlag);
    });
}
```

A 14.16 ábra mutatja azt a kódot, amelyet a felhasználó aktuális helyzetének meghatározásához a böngészőjében lefuttat.



## Összefoglalás

A térképkészítés új, fantasztikus korszakát éljük. Az 1970-es évektől az 1990-es évekig a statikus papírtérképek mellett megjelentek az első interaktív térképek. Azóta a térképek csak még inkább interaktívabbá váltak: lehetővé teszik a felhasználóknak – mind a tematikus információk, mind a háttértérképek – szerkesztését. Ez a fejezet bevezeti az olvasót az Interneten alapuló térképezés világába. Azok az eszközök, amelyeket itt bemutatunk lehetővé teszik modern térképek készítését.

## Irodalomjegyzék

Google Maps JavaScript API V3 Basics (2011). (keresőszó: Google Maps JavaScript API V3 Basics).

Neumann, A., Winter A. M. (2003). "Web-mapping with Scalable Vector Graphics (SVG): Delivering the promise of high quality and interactive web maps." In: Peterson, M. P. (ed.) *Maps and the Internet*. Elsevier, Amsterdam, pp.197–220.

Peterson MP (2008) *International Perspectives on Maps and the Internet*. Springer, Berlin.

Svennerberg, Gabriel (2010). *Beginning Google Maps API 3*. New York, NY: Apress.

W3Schools.com (2011). JavaScript Tutorial.

<http://www.w3schools.com/js/default.asp>. (keresőszó: Learning JavaScript).

Willard, Wendy (2009). *HTML: A Beginner's Guide*.

Berkeley, CA: Osborne/McGraw-Hill.