



*Eötvös Loránd Tudományegyetem  
Informatikai Kar  
Térképtudományi és Geoinformatikai Tanszék*

# **Magyar vonatkozású népességtérképek (Internetes adatbázis építése)**

## **DIPLOMAMUNKA**

Készítette:

Jeney János György  
térképész hallgató

Témavezető:

Márton Mátyás  
egyetemi tanár

Elek István

habilitált egyetemi docens

Németországi témavezető:

Prof. Dr. phil. Heinz Musall

Budapest, 2011

# Tartalomjegyzék

1. Bevezetés .....	4
2. Magyar vonatkozású térképek Németországban .....	4
2.1 A Badeni Tartományi Könyvtár, Karlsruhe .....	5
2.2 A Württembergi Tartományi Könyvtár, Stuttgart .....	5
2.3 A Heidelbergi Egyetem Földrajzi Intézetének könyvtára .....	6
2.4 A Tübingeni Egyetem Földrajzi Intézetének könyvtára .....	7
2.5 A Berlini Állami Könyvtár .....	8
2.6 A térképek rendezése, honlapszerkesztés .....	9
3. Térképek digitalizálása .....	10
4. Többnyelvű rendszer alapjai .....	13
5. Felhasználóbarát koncepciók .....	14
5.1 A piros keret megjelenítése .....	15
5.2 Tooltípek megjelenítése .....	16
5.3 Sorok kiemelése a keresési eredményben .....	17
5.4 A térképeken lévő aktív felületek megjelölése .....	18
6. Stíluslapok .....	19
7. Dataset átadása a kliens részére .....	22
8. Az adatbázis és az svg közötti kapcsolat .....	29
9. Szöveges adat utólagos beágyazása a honlapba .....	30
9.1 A szerveren tárolt raszteres képek megjelenítése .....	32
10. Tematikus térképek készítése .....	33
10.1 Adott vármegyét ábrázoló térképek száma .....	33
10.2 Adott nemzetiséget bemutató térkép készítése .....	34
10.3 Kiválasztott térképet tartó intézet megjelölése a térképen .....	34
10.4 A térképen ábrázolt vármegyék bemutatása .....	35
10.5 Az Európa térkép feliratainak elkészítése .....	35
11. A kereső működése .....	36
11.1 A vármegyés kereső működése .....	37
11.2 A hagyományos kereső működése .....	39
12. A keresési eredmény .....	40
13. Vármegye adatlapja .....	43
14. Összefoglalás .....	45
15. Irodalomjegyzék .....	46
16. Köszönetnyilvánítás .....	47
I. Objektumok elhelyezkedése a honlapon .....	48
I.1 A popup ablak objektumai .....	49
II. Külön lapon megjelenő térképböngésző elemei .....	52

III. Az adatbázis felépítése .....	53
III.1 Az adatbázist felépítő táblák .....	53
III.2 A táblákban lévő mezők .....	54
IV. A könyvtárstruktúra felépítése.....	61
IV.1 A webszerver adatai.....	61
IV.2 A weben nem elérhető könyvtárak .....	61
IV.3 A weben elérhető könyvtárak felépítése.....	63
V. A honlapon lévő vektoros térképek .....	63
V.1 Felhasznált vektoros térképek: .....	63
V.2 Alapanyagok minősége:.....	64
VI. Kliens oldali függvények.....	64
VII. Szerver oldali függvények.....	101
VIII. Az Európa térképen használt országkódok .....	121
IX. A vármegyék kódjai .....	123
X. Intézetek ahonnan térképeket gyűjtöttem .....	125

# 1. Bevezetés

Magyarországon a mai tendenciák a térképek digitális archiválását és online térképmúzeumok elkészítését mutatják. Ennek a tendenciának egyik úttörője a Virtuális Glóbuszok Múzeuma (vgm.elte.hu). Hasonló rendszer felépítésébe kezdtem a magyar néprajzi térképekkel. Több kereső kifejllesztésével, és számos forrást felkutatva elkészítettem a Magyar Néprajzi Térképek honlapját.

Három féléves hallgatói ösztöndíjjal volt lehetőségem több németországi térképtár felkutatására. Ezekben a térképtárakban sok térképet találtam, amelyeket Magyarországon még nem ismertek. Legtöbbjük Németországban vagy más német nyelvű országban készült, de volt közöttük amerikai és angol készítésű is, illetve olyan is, amely hazánkban készült. Általában jobb állapotban voltak, mint a magyarországi térképtárakban lévő térképek, mivel nem volt akkora irántuk az érdeklődés, és így nem érte őket a gondatlan használatból adódó sérülés.

Ezeket a térképeket egy honlapon tettem elérhetővé, amelyen lehet közöttük lelőhely szerint, megadott kulcsszavakkal vagy pedig a térképen ábrázolt vármegyék szerint keresni. A honlap a <http://demogmap.elte.hu> címen érhető el.

## 2. Magyar vonatkozású térképek Németországban

A magyar néprajzi térképezés a kiegyezés és a trianoni békediktátum aláírása között fénykorát élte. Az akkor készült térképek ma már sok esetben nagyon rossz állapotban vannak. A szelvényezett műveknél gyakran hiányoznak egyes szelvények, de az is előfordul, hogy egész térképművek is hiányoznak, amelyeket a szakirodalomból egyébként ismerünk. A fény és levegő hatására a térképek kifakultak, ami különösen az egylapos térképeknél fordul elő. Nem ritkán sokáig falra is ki voltak téve egyes térképművek. Sokszor már az eredeti állapotuk visszaállítása is lehetetlenné válik. Magyarországon és a környező országokban ezeket a térképeket elég sokan kutatják, s a gondatlan használat rontotta le állagukat.

Térképész hallgatói ösztöndíjasként alkalmam nyílt több németországi térképtárat felkeresni, ahol készülő diplomamunkámhoz anyaggyűjtést végeztem. Nagy meglepetésemre szolgált, hogy Németország területén is sok térkép található ebben a témában. Bár gyakran ezek sem a legjobb körülmények között voltak tárolva, viszont az a tény, hogy kevesen érdeklődnek irántuk, segítette megőrizni az állagukat. A falra nem voltak kitéve, és legtöbb esetben a fénytől elzárva tárolták ezeket.

A térképek egy részét Magyarországon nyomtatták, és valamilyen úton Németországba

kerültek. Ezek legtöbb esetben egylapos vagy szelvényezett térképművek, de vannak magyar készítésű atlaszok is. Mások a német nyelvterülethez köthetők, de előfordulnak még Amerikából származó térképek is. Szembetűnő különbség, hogy a Magyarországon készített térképeken a magyar népesség vörös színnel van ábrázolva és a német sárgával, míg a Németországban és Ausztriában készített térképeken a német népesség van vörössel ábrázolva és a magyar sárgával. Több könyvtárat és levéltárat is felkerestem.

## **2.1 A Badeni Tartományi Könyvtár, Karlsruhe**

Karlsruhéban – mint vendéghallgató – kutatásaimat a Badeni Tartományi Könyvtárban található térképekkel kezdtem. Először a Richard Andree (1835–1912) által készített Andree's Allgemeiner Handatlas (Verlag von Volhagen und Kalssing, Bielefeld és Lipcse, 1887) című műnek különböző kiadásaiban találtam néhány térképet. Volt még ezen kívül egy fekete-fehér fotómásolat a Joseph Chavanne (1846–1902) által készített Physikalisch-Statistisch Hand-Atlas von Österreich-Ungarn (Eduard Hölzel's Geographisches Institut, Bécs, 1887) című atlaszműről. Az ilyen másolat csupán arra jó, hogy eldönthessük, hogy érdemes-e az eredetit valahol felkutatni.

A Heidelbergi Egyetemen talált nyomatok alapján Karlsruhéban megkerestem a Geographical Review című lapnak a 4. évfolyamát (1918), és abban megvolt a Bruce (?) C. Wallis (?-?)<sup>1</sup> által készített három Magyarország térkép 4-4 szelvénye. Ezek után kezdtem átnézni a földrajzi folyóiratokat, és sok korabeli lapban találtam néprajzi térképet. A Geographical Review eggyel korábbi évfolyamában találtam több fekete-fehér térképet. A The Geographical Journal című lap 47. számában (1916) akadtam rá hat, a Bánságot ábrázoló térképre, amelyek bemutatják a néprajzi térképezés hatféle ábrázolási módját. Ebben a folyóiratban található egy Horvátországot bemutató térkép is, amin az adott népesség által lakott terület a jelmagyarázatban megadott színnel van körberajzolva, de nincs kitöltve, így nem takar ki más térképi tartalmat. Ezen a térképen barna domborzatárnyékolás, a vasutak, valamint kiegészítő megírások vannak feltüntetve.

## **2.2 A Würtembergi Tartományi Könyvtár, Stuttgart**

A Stuttgartban lévő Würtembergi Tartományi Könyvtár is sok kutatásra érdemes térképet rejtett magában. A könyvtár főépületétől távol eső térképtár nagyon gazdag térképanyaggal rendelkezik.

---

<sup>1</sup> A születési és halálozási évszámot nem sikerült felderítenem.

Három atlaszban is voltak olyan térképek, melyek Magyarországot ábrázolták. Az első a Budapesten nyomtatott több nyelven is kiadott Halász Albert (?–?)<sup>2</sup> és Edvi-Illés Aladár (1858–1927) Magyarország háborús veszteségét bemutató atlasz, német nyelven (eredeti cím: Ungarn vor und nach dem Krieg in Wirtschaftstatistischen Karten). A nyelv a településnevekre nem mindig vonatkozik, de a magyarázó névrajzra, valamint a kereten kívüli megírásokra igen. Itt megtalálható egy eredeti színes példány a Karlsruheban csak fekete-fehér másolatban meglévő Joseph Chavanne által készített Physikalisch-Statistisch Hand-Atlas von Österreich-Ungarn című atlasznak. A harmadik egy német kiadású Közép-Európa atlasz, amiben a térség népességét ábrázoló térképkivágaton rajta van Magyarország. Külön sem Magyarország népességét, sem a Habsburg Birodalom népességét bemutató térkép nincs benne. Ezek az atlaszok viszont mind nagyon jó állapotban vannak.

### **2.3 A Heidelbergi Egyetem Földrajzi Intézetének könyvtára**

A Heidelbergi Egyetem Földrajzi Intézetének könyvtára is számos térképet őriz. Ezek legtöbbje nincs katalógusban nyilvántartva. Csak néhány található meg a Heidi-katalógusrendszerben (a Heidelbergi Egyetem könyvtárainak központi számítógépes katalógusa), így feltételezhető, hogy ott, az eddig találtaknál sokkal több térkép van.

Az ottani kutatásoknak első számú akadálya, hogy a térképtár csak kedden és csütörtökön van nyitva, és azokon a napokon is csak másfél órát. Szerencsére itt a térképek kölcsönözhetőek abban az esetben, ha valaki be van iratkozva a könyvtárba. Ez azt az előnyt jelenti, hogy el lehet a térképeket vinni, és kielemezni, vagy beszkenneálni. A hátránya az, hogy ezek a térképek nincsenek annyira jó állapotban, mint más könyvtárakban, mivel hazaviszik őket a diákok is. A Magyarországot ábrázoló térképműveknek itt is megvan az az előnyük, hogy kevesen viszik haza, de az állapotuk ennek ellenére sem kiváló. A szakadások nincsenek megragasztva, ami azért jó, mert így a ragasztó által okozott elszíneződés nem okoz gondot a szkennelés során, de csak két fólia között lehet őket szkennelni. A fóliában könnyen elcsúszhat a térkép, illetve a fólia elektrosztatikusan feltöltődik, így a por hozzáragadhat, ami rontja a szkennelés minőségét. Ez tiszta körülményekkel, és a fólia rendszeres tisztításával elkerülhető. Vigyázni kell, hogy a fólia ne törjön, ne karcolódjon, és a szkenneren átmenő oldala ne hordozzon port, mivel tönkretelheti a szkennert. A munka az üveg gyakori tisztítását igényli. Az előnézet alkalmazása sem ajánlott, mert elcsúszhat a térkép a fóliában, és akkor nem azt szkenneli, amit kijelöltünk. Jobb beszkenneálni az egész látható területet, és egy általános grafikai programban kivágni.

---

<sup>2</sup> A születés és elhalálozás évét nem sikerült kideríteni

Heidelbergben két meglepő, B. C. Wallis által készített, New Yorkban kiadott népességtérképet találtam, amely a történelmi Magyarországot három részben ábrázolja: Észak-Magyarország, Közép-Nyugat-Magyarország és Dél-Magyarország. Mivel Erdély mindkettő térképről hiányzik, feltételezhető, hogy volt egy Közép-Kelet-Magyarország is, csak az mindkét mű esetében hiányzik. Ahogyan a Badeni Tartományi Könyvtárról szóló beszámolómban írtam, ez a feltételezés helyes volt. A Heidelbergben lévő három szelvény mindkét műben össze van ragasztva. Ez elég sajátos alakot ad nekik, viszont ezen kívül nincs más ragasztás, így a ragasztás miatti elszíneződés sincs. A bélyegzőkből látszik, hogy a ragasztásnál az átfedéseket és a keretet levágták. Ezeknek a továbbkutatását különösen érdekesnek tartottam mivel nem gondoltam arra, hogy az Amerikai Egyesült Államokban is készült magyar vonatkozású néprajzi térkép. Ezen kívül megtalálható itt Heidelbergben a Kogutowicz-féle négynyelvű pontszórásos népességtérkép, amelynek sajnos egyes szelvényei hiányoznak, és helyenként van benne szakadás, viszont a fakulás enyhébb mértékű. Ezek közül csak a Kogutowicz által készített térkép van nyilvántartva a Hei-  
delsbrunn rendszerben.

## **2.4 A Tübingeni Egyetem Földrajzi Intézetének könyvtára**

A Tübingeni Egyetem Földrajzi Intézetének könyvtárában is sok magyar vonatkozású térkép található. Tübingent egyik világháború alatt sem bombázták, így a háború nem pusztította el a térképanyagot. Kizárólag atlaszban vannak népességtérképek, bár néhány atlaszba bele van csúsztatva egy-egy egylapos térképmű is. Ez viszont az állagát ugyanúgy óvta mintha bele lenne kötve, de a szkennelés sokkal könnyebb, mivel át lehet futtatni egy nagy szkenneren.

Több magyar készítésű atlaszt is birtokolnak. Az egyik a már korábban említett Halász Albert–Edvi-Illés Aladár által készített atlasz angol nyelvű változata. Kogutowicz Károly (1886–1948) zsebatlaszának 1922-es és 1923-as kiadása is megvan, bár az 1923-as kiadást még valószínűleg senki nem nézte meg, mivel még meg sincs vágva. Továbbá egy 1943-as német kiadású Burgenland atlasz is megtalálható itt. A könyvtárban van egy ismeretlen eredetű francia nyelvű fekete-fehér néprajzi atlasz is. Ebben se a kiadás helye, se kiadó, de a kiadás éve sincs feltüntetve. Még az adatok forrására sincs utalás, így csak valószínűsíteni lehet, hogy 1920 körül készülhetett.

A könyvtárban található egy 1914-ben nyomtatott román történelmi atlasz, amiben van három néprajzi, illetve történelmi néprajzi térkép. Összehasonlítva a magyar atlaszokkal számos ellentmondást fedezhetünk fel. Feltűnő, hogy a román térképeken 1867–1914 között jelentősen csökken Erdélyben a magyar népesség. Az adatok forrása is bizonytalannak tűnik. A térképek címei angol és román nyelven vannak megírva, míg az egyéb megírások csak románul szerepelnek.

Található egy német nyelvű Kárpát-térség atlasz is (eredeti cím: Karpatenraum). Ebbe bele van csúsztatva egy nagyméretű összehajtott térkép, amit a német birodalmi vezetés számára készítettek. Ezen csak a kisebbségek vannak feltüntetve. Minden kisebbségnek van egy megfelelő jelkulcsi eleme, ami az egész térképen ugyanaz, viszont a fehér minden országban az államalkotó népeiséget ábrázolja. Ez eleinte nagyon zavaró a térkép elemzésénél, mert első ránézésre nem lehet tudni, hogy a fehér mit jelent. Ugyanakkor ritka darab, mivel kevés térkép van, ami az 1943-as határokat és adatokat tünteti fel. Könnyen elképzelhető, hogy ezek többnyire elpusztultak a háborúban, és Tübingenbe azért maradtak meg, mert a város nem szenvedett háborús károkat. Van ott továbbá egy Budapest atlasz is, amit Dr. Illyefalvi I. Lajos (1881–1944) készített. Ennek egyik fejezete foglalkozik Budapest népeiségeivel, valamint azzal, hogy az ország többi részéről mekkora a bevándorlás Budapestre.

## **2.5 A Berlińi Állami Könyvtár**

A Berlińi Állami Könyvtár, amely egyben a porosz kulturális örökség része is, számos térképet tartalmaz e témában. A könyvtár két épületben helyezkedik el.

A Potsdammer Strassében lévő sárga épület földszintjén lehet beiratkozni. Itt az olvasótermekbe se lehet bemenni tagsági kártya nélkül, mivel azt a bejáratnál ellenőrzik. Mindent, amit beviszünk magunkkal, be kell mutatni a portán, és papírt kapunk róla, ami nélkül nem tudjuk azokat kivinni. Vaku és állvány nélkül le lehet fényképezni a térképeket, viszont ezekről jegyzéket kell készíteni a könyvtár számára. Jelenleg a térképtár az egyetlen részlege a könyvtárnak, ami ketté van választva a két épületben.

Bár hivatalosan a Potsdammer Strasse-i épületben az 1946 után készített, míg az Unter der Lindenen lévő épületben a 1946 előtt készített térképek találhatók, a gyakorlatban azonban előfordul az ellenkezője is. A térképek közül sok benne van a StabiKat rendszerben, ami a berlińi könyvtár számítógépes katalógusrendszere. A térképeket papíron kell megrendelni, tehát nem lehet előre kérni az interneten keresztül. Ez azt jelenti, hogy a papírok kitöltése majd leadása után 1 órát kell várni, míg kihozzák a kért művet, feltéve, ha egyből megtalálják. Sok térkép viszont nincs benne a StabiKat rendszerben, ezeket a mikrofilmre lefényképezett egykori cédulakatalógus alapján lehet kikeresni. Sajnos néhány esetben a jelzet helytelen. Néha gyorsan megtalálják a könyvtár dolgozói, de néha több napot is várni kell rá. A két épület közötti állományköltöztetés és a rossz katalógusjelzetek miatt sok anyagot nem sikerült még megtekinteni. Az Unter der Lindenen található épületet jelenleg átépítik, tehát ez is nehezíti a kutatást.



## **2.6 A térképek rendezése, honlapszerkesztés**

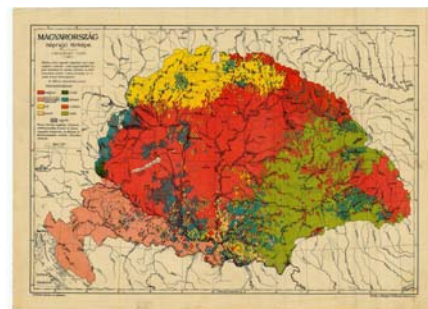
A Németországban gyűjtött térképekből online adatbázist készítettem, ami egy webes felületen elérhető. Ezen különböző keresési lehetőségek vannak, és a térképeket meg lehet tekinteni egy térképnézegetőben. Lehet keresni kulcsszavak (térkép leírása, szerző, kiadó, kiadás helye, előállítás éve, méretarány, valamint a könyvtárban/levéltárban használt katalógusjelzet) alapján. A keresési kulcsszót el lehet küldeni egy kattintással, vagy pedig leütésekkel is lehet szűkíteni az eredményt. Abban az esetben, ha nem ír be valaki keresési kulcsszót, akkor a megjelölt lelőhelyről mindent kilistáz a program. Abban az esetben pedig, ha minden lelőhely ki van választva, akkor az összes térképet kilistázza. Tovább lehet szűkíteni a keresést a lelőhely megadásával, ami kiválasztható egy listából vagy kijelölhető egy térképen.

Egy másik keresési módszer az ábrázolt terület alapján történhet. A történelmi Magyarország vármegyei jelennek meg egy térképen, és az egyes vármegyéket mint keresési feltételt lehet megadni, hogy a keresés eredményeképpen csak azok a térképek jelenjenek meg, amelyek az adott vármegyének teljes területét vagy területének egy részét ábrázolják; vagy pedig az eredményként megjelent térképek ne ábrázolják az adott vármegyét. Nem kell, hogy minden vármegye feltétel legyen. Kijelölhetjük a vármegyéket egyenként egérrel, vagy pedig az 1910-es népszámlálás alapján megadott etnikai aránnyal. Ekkor az alsó és felső határát kell megadni annak, hogy az adott etnikum aránya az össznépességhez viszonyítva meddig terjedjen a vizsgálni kívánt vármegyékben, és azt, hogy milyen feltételként jelölje meg a vármegyéket. Ezt többször is megismételhető különböző népcsoportokra. Ezek az adatok a honlapon megtekinthetők a vármegyékről szóló információkban, továbbá egy tematikus térkép is rendelkezésre áll, ami az 1910-es népszámlálás vármegyékre összesített adatait megjeleníti. Adott vármegyét ábrázoló térképek számát mutató térkép is megjeleníthető, ami egy adott levéltárra is szűkíthető. Ezen kívül a levéltárak adatait is meg lehet tekinteni. A térképek feltöltése jelenleg még folyamatban van. Eddig közel 70 térképmű 1000 szelvényének feldolgozása történt meg.

A honlap a <http://demogmap.elte.hu> webcímen érhető el magyar, angol és német változatban. Az alapértelmezett nyelv a böngésző nyelve, abban az esetben, ha ez nem az előbb felsorolt nyelvek egyike, vagy nem felismerhető a böngésző nyelve, akkor angolul jelenik meg a honlap, de a nyelv a jobb felső sarokban lévő zászlókra való kattintással bármikor megváltoztatható.

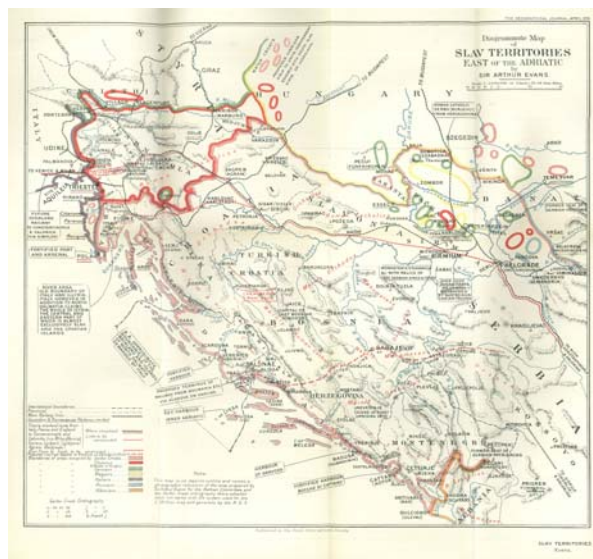
### 3. Térképek digitalizálása

A térképek digitalizálása raszter állományba történt szkennelés vagy fényképezés útján. Mivel a szkennelésnek jobb a minősége, ha lehetséges volt a szkennelést választottam. Az A4-es vagy A3-as szkennerre ráférő térképek esetében ez általában nem okozott gondot, függetlenül attól, hogy könyvben vagy külön lapon voltak. A nagy térképlapok esetében sem volt gond, ha nem voltak könyvbe kötve, mivel



*Egy dobszkenneren átfutott nagyméretű térképlap*

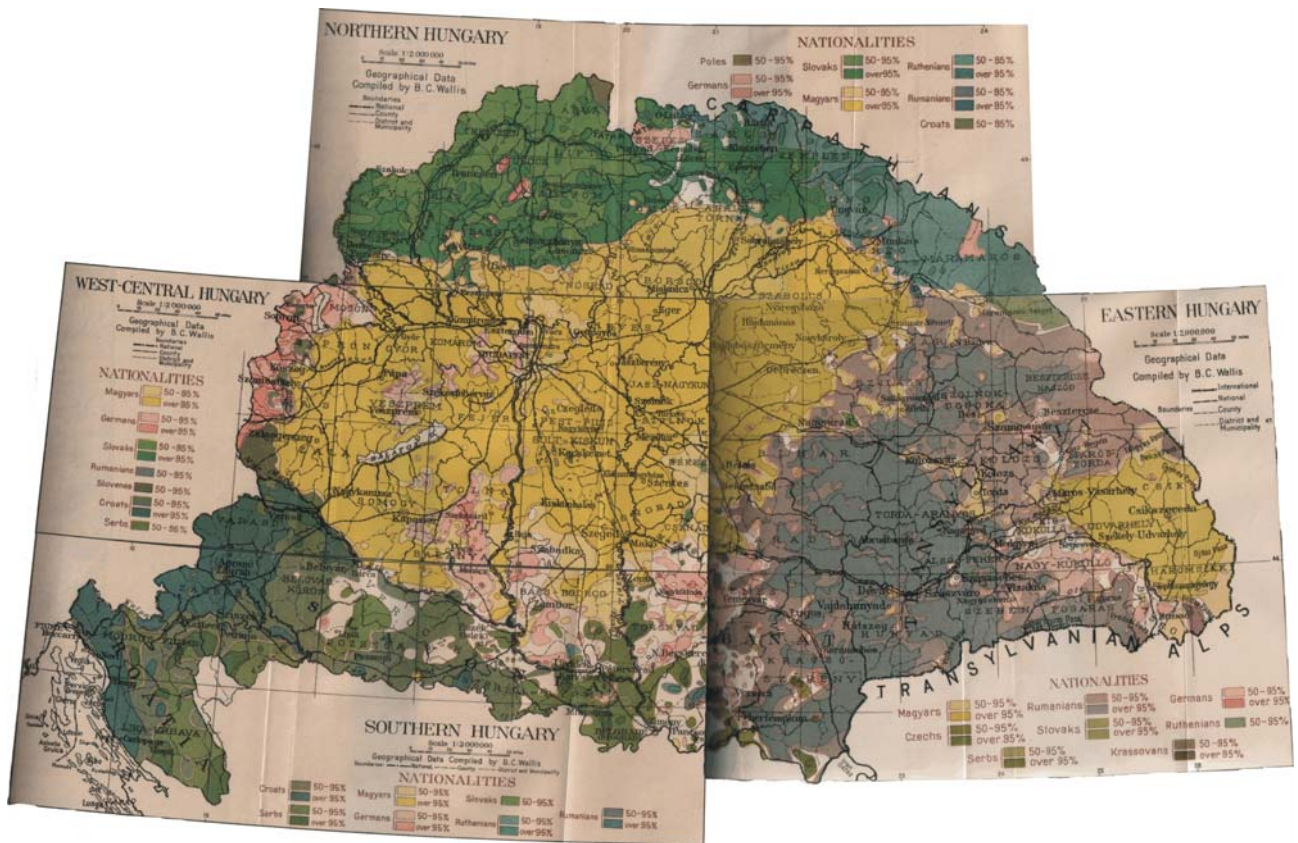
azt egy nagy szkenneren átfuttattam. Viszont mivel az keresztülhúzza a lapot, ha könyvbe van kötve, nem lehet keresztülegedni. Az ilyen térképeket egy A3-as vagy A4-es szkenneren kellett beszkennelni, és folyamatosan arrébb csúsztatni úgy, hogy minimum 1-2 cm átfedés legyen minden irányban a szelvények között. Nagyon fontos odafigyelni, hogy a fehér-egyensúlyt a szkennerek kalibrálni tudja. Azoknál a szkennereknél, ahol az üveglap előtt van egy kivágat fontos arra figyelni, hogy ezt ne takarjuk le, illetve arra, hogy a szkennerek fedele le legyen csukva, mikor szkennelünk. Fontos, hogy az összes szelvényt ugyanazon a szkenneren szkenneljük. A másik gyakran felmerülő probléma, hogy a kötésnél nem simul teljesen a szkennerre a lap. Ilyenkor a kötés körül sötétebb lesz a kép. Szerencsére ez minden esetben a kép széle. Ezt a jelenséget csak redukálni lehet azáltal, hogy a kötést lenyomom, amikor megy a szkennerek, de fontos arra odafigyelni, hogy ne mozogjon közben a könyv.



*Egy Photoshopban összeillesztett térkép. Néhány apró hiba látszik*

Ugyanakkor nem lehet nagy erővel lenyomni, mert

akkor vagy a szkennerek üvege törik össze, vagy a könyv kötése sérül. Szerencsére sok régi kötésű atlasznak olyan kötése van, ami ha kinyitom, akkor kisimul. A legtöbbször ez a jelenség az utólag egybekötött folyóiratoknál jelentkezett. A szelvényeket a lehető legnagyobb felbontásban érdemes szkennelni, hogy az összeillesztésnél a program minél jobban felismerje az átfedéseket. Ez legjobban a B. C. Wallis által készített négyszelvényes térképművön látszott, mivel annak a keleti szelvényének a nyugati széle volt a kötésnél. Ezt, mivel nem volt átfedés a négy szelvény között, kézzel illesztettem össze. A szelvényhatárokon látszik egy sötét csík. Sajnos nem lesz soha tökéletes az összeillesztés, de közel áll hozzá.

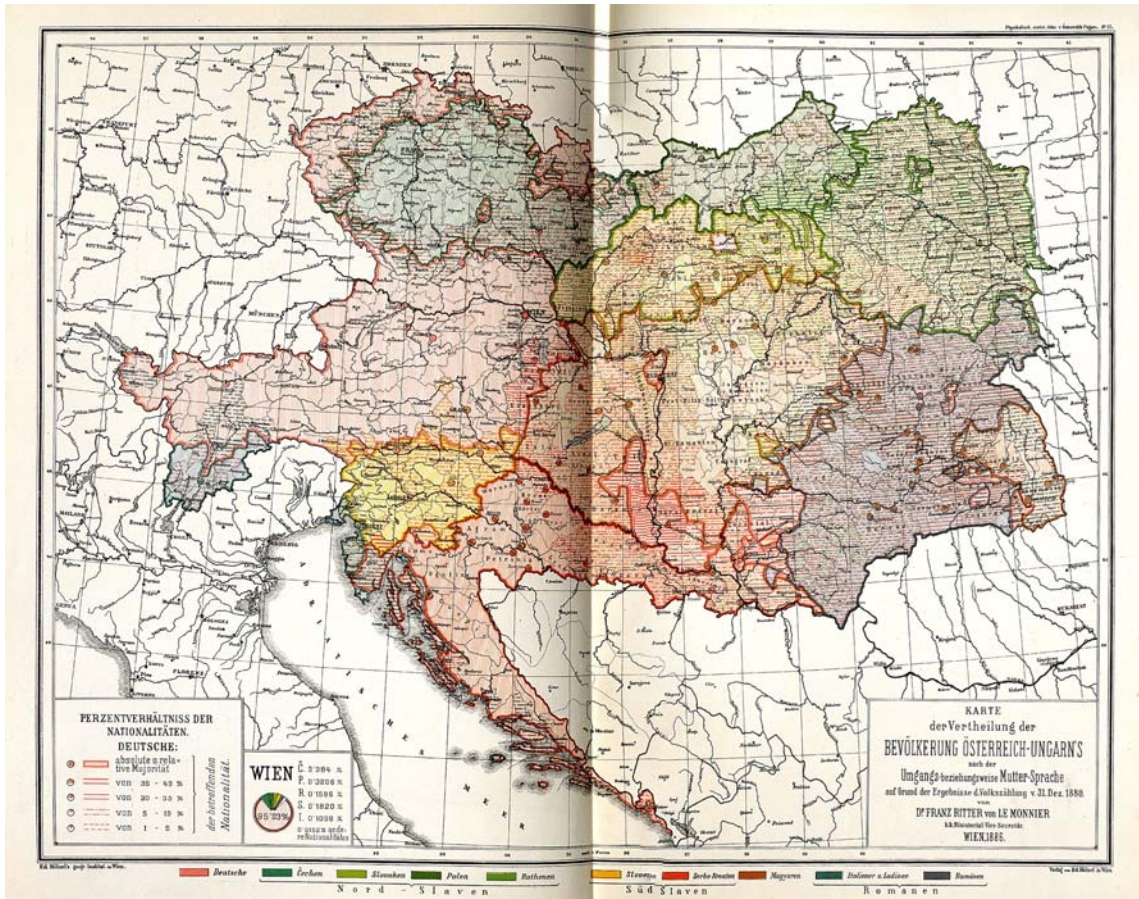


*A B.C. Wallis által készített három szelvény összemontirozott képe*

Ha az összeillesztett képnek utólag csökkentem a felbontását sok hiba láthatatlanná válik. Az összeillesztést az Adobe Photoshop nevű programmal végeztem. Az Automatizálás menüpont alatt van egy Photomerge nevű funkció. Ezzel betöltöttem az összes szelvényt, és bár sokszor akár egy órát is igénybe vett, és tiff formátumban lehetett a nyers szelvény akár 3.2Gb is, összeillesztette. Ennek a méretét utána lehetett csökkenteni a felbontás csökkentésével, és jpeg-ben való tárolással. A nyers képeket mind tiffben tároltam az adatvesztés elkerülése érdekében, de a honlapon a jpeg formátumot használtam, mivel annak a terjedelme sokkal kisebb, mint a tiffé.

Sok könyvtárban ma már van könyvszkennerek. Ebbe csak bele kell tenni a könyvet, és végigmegy egy fénycsóva fentről, és beszkeneli. Ezzel két probléma van, ami sajnos csak végszükség esetén teszi lehetővé, hogy használjuk. Az első, hogy ezeknek viszonylag alacsony a felbontásuk. A másik, hogy van benne egy program, ami a könyv kötésénél a torzulásokat megpróbálja kivenni, de sajnos ezáltal olyan torzulásokat tesz bele, amelyek korrigálása lehetetlen, és így sajnos a térképet használhatatlanná teszi. Ez csak akkor jó, ha nincs más, mivel akkor legalább van egy áttekintése a felhasználónak a térképről.





*Egy könyvszkenneren készített felvétel. A térkép keretén látni a torzulást. A felbontása is viszonylag alacsony*

Volt olyan könyvtár, ahol vagy nem engedtek szkennelni, vagy nagyon sokat kellett érte fizetni. Szerencsére ezekben engedték a fényképezést. Mivel vakut meg állványt nem lehetett használni, így ez az utolsó megoldás volt, amit választottam. Ilyenkor a megvilágítás sajnos nem a legjobb, és a színe is torzítja a térképi színeket. Mivel a felvételt csak kézből lehet készíteni, a perspektív torzulást is csak csökkenteni, teljesen kiküszöbölni nem lehet. Hosszú expozíciót használni sem lehet, mivel a kéz remegése elmosza a képet. Sokszor nehéz olyan helyet találni, ahol a megvilágítás is jó, és elég magasra lehet elemelni a gépet a térkép fölé, hogy perspektív torzulás nélkül el lehessen készíteni a felvételt. Annak érdekében, hogy a felbontás a lehető legnagyobb legyen, a gépet a maximum felbontáson kell használni, és figyelni kell arra, hogy a látómezőt a lehető legjobban kitöltse a térkép. Túl rövid gyújtótávolságú objektív használata sem jó, mert akkor maga az objektív torzítja a képet. Ha



*Egy kézben tartott digitális fényképezőgéppel készített felvétel*

leteszem a térképet egy székre, hogy jobban fölé tudjak hajolni, akkor sokszor az asztalok eltakarhatják a fényt. Erre oda kell figyelni, és arra is, hogy a saját árnyékom se takarja el.

## 4. Többszöveges rendszer alapjai

A honlap három nyelven készült el: magyarul, németül és angolul. Mivel minden szöveges elem egy adott nyelvhez kötődik, ezért a honlap csak egy üres keretrendszer, és a betöltéskor kerülnek rá a szöveges elemek. Az összes szöveges elemet az adatbázisban helyeztem el. A kívánt nyelv megállapítása a következőképpen történik: ha a felhasználó a cím mezőben megadta a *lang*= paramétert, vagy a rákattintott az egyik nyelvi változatra a menüben, akkor ennek az értéke kerül a *lang* globális változóba, és ebből lesz mindig a nyelv kiolvasva a honlapon. Ha ez nincs beállítva, akkor a böngésző nyelve lesz az alapértelmezett. Ha ez nem egyike a honlap nyelveinek, akkor az angol lesz az alapértelmezett.

A katalógus tartalma is három nyelven lett elkészítve. Minden bejegyzés háromszor van elkészítve, mindegyik egy külön nyelven. A honlapon mindig csak az a bejegyzés jelenik meg, ami a nyelvnek megfelel. A nyelv mezőben meg van adva, hogy melyik nyelven van az adott bejegyzés. Hasonlóan az intézeteket felsoroló táblában, minden bejegyzés három nyelven van, és a *nyelv* mező alapján határozza meg a program, hogy milyen nyelven van az adott bejegyzés. A keresési eredmény és a térkép adatai ablakba a nyelvet ezen mező szerint történő szűrés alapján dönti el, és így írja ki az eredményt. A források az a szerkesztő adatait tartalmazó táblákban hasonlóképpen a fentiekhez van egy *nyelv* mező, és minden bejegyzés háromszor jelenik meg, mind más nyelven. A szűrésnél a *lang* változó értékét összeveti a program a nyelv mező értékével (többszöri átdalalással) és így szűri meg az eredmény táblát. A tooltip tartalmát is ugyanígy van definiálva. Minden ikonhoz három bejegyzés tartozik más-más nyelven, és a *lang* változó értéke alapján szűri meg a tartalmat a program, így a megfelelő nyelvű tooltip jelenik meg. Az Európa térképen az országok nevei az *orszagok* táblában vannak. Ennek a táblának van egy kód mezője, ami tartalmazza az országok kódját, majd három nyelvi mezője, ami tartalmazza az országok neveit a különböző nyelveken. Az Európa térkép feliratait ezen tábla alapján készülnek el (lásd: 10. Tematikus térképek készítése című fejezetet). A honlapon található minden egyéb szöveges elem a *nyelvek* táblában található. Ennek a táblának a *nyelv* mezője tartalmazza a bejegyzés nyelvét, a *cim* mezője a honlap címét három nyelven, és a többi mező *elx* ahol *x* egy egész szám 1-től felfelé, az összes szöveges elemet, ami a honlapon bárhol megjelenik, és nincs felsorolva az előbb említett táblákban.

A feliratok elhelyezésének több módja van. Ha egy adott objektum tartalmát egy php szkript generálja, akkor az adatbázisból közvetlenül kiírja az összes feliratot. Ekkor az objektum tartalma

feliratokkal együtt érkeznek a kliensre. Ezeknek a szkripteknek mindig meg kell adni a *lang* változó értékét.

Egy másik megoldás a *felirat* függvény használata. A függvény bemeneti paramétere a mező neve, amiből a feliratot nyerni akarjuk. Ezt és a *lang* globális változó értékét elküldi a program a *felirat.php* szkriptnek, ami visszaküldi a várt eredményt. Ez a függvény visszatérítési értéke lesz. Ez a módszer akkor használatos, ha egy objektum tartalma kizárólag egy felirat (pl. *h1* objektum) vagy pedig az objektum tartalma a kliens oldalon készül el.

A harmadik megoldás nagyon hasonlít az előzőhöz, viszont kizárólag a tooltipknél használatos. A *tooltip* függvénynek megadott *icon* paramétert, és a *lang* változó értékét átadja a *tooltip.php* szkriptnek, és a visszakapott eredményt a *tooltip* objektumba helyezi.

A negyedik módszer az Európa térkép feliratainak elkészítésére használatos. A beágyazott térkép objektumait keresi ki a program az országkódok alapján, és azokba helyezi feliratokat az *appendFirstChild* eljárással. Ez bővebben le van írva a tematikus térképekről szóló fejezetben.

## 5. Felhasználóbarát koncepciók

A kliens oldali eseménykezelők egy része az oldal felhasználóbaráttá tételét célozza meg. Az oldalon az objektumok, valamint a tartalom elhelyezése sok esetben gondot okozott. Figyelembe kellett venni, hogy ha a felhasználó átállítja a nyelvet, ugyanabba a keretbe kell beleférjen más nyelven is az oldal. Továbbá az oldal minél több képernyőn kell megjelenjen, és minél szélesebb felhasználói rétegben használható kell legyen.

Ebből következik, hogy a helytel spórolni kellett. Mivel a menüben a szöveges elemek több helyet foglaltak volna, mint a piktogramok, így a menüben minden elem egy piktogram. Próbáltam minél egyértelműbb piktogramokat készíteni. Azért, hogy egyértelmű legyen az, hogy egy elem aktív, azaz ha rákattint a felhasználó, akkor történik valami, ha az aktív elemek fölé húzza a felhasználó az egeret, akkor megjelenik egy piros keret. A piktogramok felett ilyenkor megjelenik egy tooltip is.

A keresési eredményben ha az egeret a felhasználó egy sor fölé húzza, akkor kijelöli a sort lila színekkel. Bárhova a sorra történő kattintással ki lehet választani az elemet. Ekkor is átugrik a „radio” gomb. A szelvények kiválasztásánál nem jelenik meg piros keret, viszont ott van a szelvények mellett a „radio” gomb, és ha a képre kattint a felhasználó, akkor is kiválasztja. A nagyobb előnézet körül megjelenik a piros keret, ha fölé viszi a felhasználó az egeret, de tooltip nem jelenik meg. A térképek aktív felületei felett az eger a meglévő kitöltés egy sötétebb

árnyalatára cseréli a kitöltést amíg az egér az elem felett van. Azokon a térképeken, ahol nem aktív minden elem, az aktív elemeknek más a kitöltésük.

## 5.1 A piros keret megjelenítése

Annak érdekében, hogy a piros keret megjelenhessen az objektum körül, az objektumot a megfelelő vastagságú átlátszó kerettel kell definiálni. A keret vastagságát a *border* attribútummal definiáltam. Az átlátszóság definiálására az objektum attribútumaként definiált stíluslapot használtam. Ez utóbbit fontos a stíluslapon definiálni, mivel a piros keretet létrehozó függvények a stíluslapot módosítják. Egy példa eseménykezelők nélkül az objektum definiálására:

```
<IMG src="kep.png" border="2" style="border-color:transparent;"/>
```

Bár a fenti példában egy képet mutattam be, és a honlapon eddig mindenhol képpel lett használva a függvény, a későbbi esetleges továbbfejlesztéshez lehet használni bármilyen objektummal, amihez egy keret definiálható. A függvény mindössze a *border-color* attribútumot módosítja. A függvény részletes leírásához lásd: a *mouseoverIcon* és a *mouseoutIcon* függvényeket a VI. Kliens oldali függvények mellékletben. A *mouseoverIcon* függvényt az *onmouseover* eseménykezelő kell hogy futtassa, míg az *mouseoutIcon* az *onmouseout* eseménykezelő. Mindkettőnek az attribútuma azonos, az egyetlen különbség a két függvény között, hogy az *mouseoverIcon* függvény a meglévő keret színét pirosra (FF0000) cseréli, míg az *mouseoutIcon* átlátszóra állítja. A függvény egyetlen paramétere az objektum, aminek a keretének a színét meg kell változtatni. Ha eseménykezelőből futtatjuk a függvényt, akkor a *this* nevű objektum mindig arra az objektumra hivatkozik, aminek az eseménykezelője futtatja az adott sort. Tehát egy képet, ami bár nem aktív, de az egér ha felette van megjelenik a piros keret, a következőképpen definiálhatunk:

```
<IMG src="kep.png" border="2" style="border-color:transparent;"  
onmouseover="mouseoverIcon(this);" onmouseout="mouseoutIcon(this);"/>
```

Így kapunk egy ikont, ami körül van egy üres keret, így tehát a leendő piros keret helye fenn van tartva. Ha fölé visszük az egeret, akkor a piros keret megjelenik körülötte, ha meg eltávolítjuk, akkor eltűnik. Tooltip nem jelenik meg.

## 5.2 Tooltipek megjelenítése

A fenti példában jeleztük a felhasználó felé azt a tényt, hogy az adott objektum aktív, de azt nem, hogy mi történik, ha rákattint. Ha szeretnénk egy tooltipet is az objektumhoz, akkor előbb definiálnunk kell, hogy a tooltipben mi fog megjelenni.

Előbb az adatbázisban kell definiáljuk a *tooltip* táblában. A tábla három elemet tartalmaz. A *nyelv* mező adja meg, hogy milyen nyelven van az adott bejegyzés. Minden tooltiphez kell tartozzon egy magyar, német és angol bejegyzés. Ha a bejegyzés magyar, akkor a *nyelv* mező tartalma HU; ha német, akkor DE; ha pedig angol, akkor EN. Az *icon* mező egy megnevezést tartalmaz, amit az ikon eseménykezelőjében kell majd megadni, hogy a rendszer tudja, melyik tooltip kell megjelenjen. Ebben a mezőben lehetőleg ne legyenek mellékjeles vagy egyéb különleges karakterek, mivel ekkor nem garantált minden esetben a működés. A *tart* mezőben az előbbi két mezőben leírtaknak megfelelően (tehát adott ikonhoz adott nyelven) megjelenő tooltip tartalma kell kerüljön. A mellékjeles karakterek entitykkel kell hogy beírva legyenek.

Ezek után visszatérünk az objektum eseménykezelőihez.

```
<IMG src="kep.png" border="2" style="border-color:transparent;"  
onmouseover="mouseOverIcon(this); tooltip('vm',event);" onmouseout="mouseOutIcon(this);  
tooltipb();"/>
```

A fenti példában a *vm* ikonhoz tartozó tooltip fog megjelenni, tehát az, aminek az *icon* mezőjében a *vm* kifejezés jelenik meg. A nyelv attól függ, hogy mi a honlap nyelve, tehát a *lang* globális változóból lesz kiolvasva. Ezt az objektum első paramétere határozza meg. A második paraméter a jelen eseményt definiálja, amiből az egérmutató koordinátáit olvassa ki.

A *tooltip* függvény a *tooltip* objektum bal felső sarkát az egérmutató mellé helyezi és minden objektum fölé helyezi. Ezek után küld a szerverre egy kérést, a *tooltip.php* eljárásnak, amiben megadja a *lang* globális változó értékét és a 'vm' stringet mint paramétert. A *tooltip.php* ezen két paraméter alapján kikeresi az adatbázisból a megfelelő tooltip tartalmat, és visszaküldi a kliensnek. A kliens ezt behelyezi a *tooltip* objektumba. Ha az egeret eltávolítja a felhasználó az objektum fölül, a *tooltipb* függvény fut le paraméterek nélkül. Az minden objektum mögé helyezi a *tooltip* objektumot, és a tartalmát törli. Ezzel láthatatlanná válik, és egy újabb tooltip eseményre áll készen.

Ezek után még az *onclick* eseményt kell definiálni az objektumba, hogy megadjuk, hogy mi történik, hogy ha az ikonra rákattint a felhasználó.



### 5.3 Sorok kiemelése a keresési eredményben

A keresési eredményben a sorok kiemelése is eseménykezelőkkel történik. A háttérszínünket változtatja meg a függvény, ami alapbeállításban átlátszó. A táblázatban minden sort egy `<TR>` és `</TR>` tag között definiálunk. A `<TR>` tag eseménykezelőiben adjuk meg a `mouseoverRow` és a `mouseoutRow` függvényeket. Az ikonokhoz hasonlóan a `mouseoverRow` és a `mouseoutRow` között mindössze annyi a különbség, hogy a `mouseoverRow` lilára (31016F) állítja a háttérszínt, míg a `mouseoutRow` átlátszóra. Mindkét függvénynek az egyetlen attribútuma az objektum, aminek a tulajdonságait meg kell változtatni, tehát a `this` objektum. Ezen két eseménykezelő a következőképpen néz ki a `<TR>` tag-ben:

```
<TR onmouseover="mouseoverRow(this);" onmouseout="mouseoutRow(this);">.....</TR>
```

A fenti példában nincs definiálva az összes szükséges eseménykezelő.

Mikor a sor fölé viszi a felhasználó az egeret, a sor háttérszíne lila lesz, majd mikor eltávolítja ismét átlátszó lesz, így látszik a mögötte lévő réteg színe.

Az `onclick` eseménykezelő is fontos szerepet játszik a felhasználóbarát környezetben, mivel ennek köszönhetően lehet bárhova a sorra kattintani, és nem kell közvetlen a „radio” gombra. Mivel a táblázat egy szerveren lévő php szkripttel lesz generálva, minden sornak tudunk más-más konstans értékeket adni. Így tehát az `onclick` eseménykezelőben a paraméterek bár a kliens oldalon konstansként jelennek meg, a szerver oldalon sok esetben változókból lettek generálva. Egy sor generálása következőképpen néz ki:

```
<TR onmouseover="mouseoverRow(this);" onmouseout="mouseoutRow(this);"
onclick="clickRow('id','sorszam','kepszam','kepnev','intezet','tk')><TD><INPUT name="tk"
id="tk*" onclick="clickRow(.,.....);"/></TD> .....</TR>
```

A fenti példában a `clickRow` paraméterei ugyanazok, mint a `<TR>` tag-ben. A `<TR>` többi elem nincs feltüntetve, ezt a kereső működése című fejezetben fejtem ki. Az `id` attribútum „tk\*”, ahol \* egy szám, 1-től bárméddig, ami azt jelzi, hogy hányadik sor a táblázatban. Ez megegyezik a `id` értékével a `clickRow` eljárás paramétereiben. A `name` attribútum fontos, hogy minden elemnél megegyezzen, mivel így lehet elérni, hogy ne lehessen két elemet bejelölni, és ha rákattint a felhasználó egy másik sorra, akkor vegye le a kijelölést a meglévő sorról. A `sorszam`, `kepszam`, `intezet` és `tk` az azonos nevű mezők megfelelő értékei a `katalogus_bov` táblából. Ezt bővebben kifejtem a 11. A kereső működése című fejezetben.

## 5.4 A térképeken lévő aktív felületek megjelölése

A honlap betöltésekor lefut az *init* függvény, ami többek közt lefuttatja a *terkersofi* függvényt. A függvény részletes leírása a VI. Kliens oldali függvények című mellékletben található. Az *euaktiv* táblában az Európa térképen lévő aktív felületek vannak definiálva. Az *ország* mezőben az ország kódja csupa nagybetűvel, a *polygon* mezőben ugyanez csupa kisbetűvel van. Ez utóbbi megegyezik az ország poligonjának azonosítójával. A *terkepnev* mezőben van annak a térképnek a neve, ami feljön, ha rákattint a felhasználó az ország poligonjára. A *terkeresofi* függvény lekéri a szerverről a táblázat tartalmát, és az összes poligon kitöltését – aminek megjelenik ebben a táblázatban az azonosítója – megcseréli lilára (7E45B3). Ezen túl hozzárendel eseménykezelőket. Az *onmouseover* eseménykezelő a *mouseOver* függvényt futtatja le, benne a poligon azonosítójával. A függvény az azonosító alapján kikeresi a poligont, és megcseréli a kitöltését egy sötétebb lilára (4B007F). Az *onmouseout* eseménykezelő a *mouseOut* függvényt futtatja ugyazokkal a paraméterekkel, mint a *mouseOver* függvény. Ez utóbbi visszacseréli az eredeti világoslilára (7E45B3) a poligon kitöltését. A további eseménykezelők és egyéb funkciói ennek a függvénynek a VI. Kliens oldali függvények című mellékletben és a 11. A kereső működése című fejezetben található. Ugyanígy végigfut a *germany* tábla alapján, és ugyanígy hozzárendeli az eseménykezelőket. A Németország térképéhez a *mouseOverg* és a *mouseOutg* függvényeket rendeli, amelyek ugyanazt végzik a Németország térképen, mint a *mouseOver* és *mouseOut* függvények az Európa térképen. A *terkereso* tábla alapján végigmegy a Baden-Württemberg térképen, és ugyanígy bejelöli az aktív felületeket. Az eseménykezelőket hasonlóan rendeli hozzá, a *mouseOverMap* és a *mouseOutMap* függvényeket használva. Az eseménykezelőknél fontos, hogy mivel a függvények a html oldal kliens oldali függvényei, de az svg állomány az oldalba beágyazott állomány, így minden függvényt a *parent* objektumon keresztül érünk el. Ez azt jelenti, hogy úgy kell definiálni, hogy *parent.fv* tehát pl. *parent.mouseOverMap*.

Mivel a vármegyes keresőben minden poligon aktív, nem különbözteti meg a kitöltést a program. Ekkor a *vm* tábla összes elemét kiolvassa, és végigfut az összes vármegyén, és hozzárendeli az eseménykezelőket a fentiekhez hasonló módon. A vármegyék kitöltését a *mouseOverVM* és a *mouseOutVM* függvények változtatják meg. A vármegye kódot kell megadni, mint paramétert, ami egyben a vármegye poligon azonosítója is. A *vm* globális tömbből kinézi a program, hogy mi a vármegye színe, és az alapján változtatja meg a megadott szín egy sötétebb árnyalatára. A vármegyes kereső eseménykezelői a 11.1 Vármegyes kereső működése című fejezetben bővebben ki vannak fejtve.

## 6. Stíluslapok

A honlap stílusát a stíluslapokon lehet definiálni. Ennek három fajtája van. Lehet külső stíluslap, ami egy .css kiterjesztésű szöveges állomány, amibe bele van írva a stíluslap tartalma. Ezt a link tag-gel lehet az oldalhoz kapcsolni. Minden esetben a fejlécben kell megadni:

```
<LINK rel="stylesheet" href="stylesheet.css" type="text/css"/>
```

Ha a tartalmát közvetlen a fejlécbe szeretnénk beírni, akkor a style tag-et kell használni:

```
<STYLE type="text/css">
```

.....

.....

```
</STYLE>
```

A style tag közötti adatok és a .css állomány formátuma azonos. Ha egy adott elem adott tulajdonságát – tegyük fel – kétszer definiálja, akkor mindig az utolsó definíció lesz érvényes, függetlenül attól, hogy az egy link vagy style tag-gal van definiálva. Én a style tag-et nem használtam, viszont sokszor kísérletezésre jó volt.

A harmadik megoldás attribútumként történő megadása. Ekkor az elem style attribútumában definiálom a különböző tulajdonságokat. Ez minden esetben felülírja az eddig definiált tulajdonságokat.

A stíluslappal lehet pl. definiálni, hogy az oldalon milyen színűek legyenek a hivatkozások, milyen színű legyen az oldal háttér, vagy mekkorák legyenek a betűk, milyen legyen a színük stb. Lehet definiálni viszont objektumok helyeit is, hogy milyen kerete legyen az adott objektumnak, a keret színét stb. A tulajdonságokat lehet a javascript függvényekkel is módosítani, tehát eseménykezelők is tudják módosítani.

Egy .css állomány formátuma a következő:

```
body {  
    background-color:#030C3D;  
    color:#FFFFFF;  
}  
h1 {  
    color:#FFFFFF;  
}  
a:link {  
    color:#FFFF00;
```

```

}
a:visited {
    color:#FF9C00;
}
a:hover {
    color:#FF0000;
}
a:active {
    color:#FF0000;
}

```

A fenti példában az összes adott típusú objektumra vonatkozik. Tehát pl. az összes link sárga lesz az oldalon, kivéve ha felül lesz írva egy másik stíluslappal vagy stíluslap-elemmel. A háttér színe sötétkék lesz, a *h1* tartalma pedig fehér. Létre lehet hozni a felhasználó által definiált osztályokat is. Ekkor ponttal kezdődik a sor.

Tehát a következő lehetne egy osztály:

```

.zold {
    color:#00FF00;
}

```

Ha egy objektumhoz hozzá akarom rendelni, akkor a class attribútumban kell megadni. Pl. így:

```
<P class="zold">.....</P>
```

Lehet egy osztályt, vagy annak részét csak adott fajta objektumhoz rendelni:

```

p.zold {
    color:#00FF00;
}

```

Ebben az esetben csak akkor lesz érvényes az osztály, ha azt a `<P>` tag-re alkalmaznom. Ha egy bizonyos objektumra akarom az osztályt, akkor a `#azonosító` formátumban kell megadni.

```

#zold {
    color:#00FF00;
}

```

Így csak a *zold* azonosítóval rendelkező objektumom lesz zöld.

A stíluslapon definiált stílusokat a javascript *style* eljárással is lehet változtatni. Ezt a következőképpen kell:

```
docuemnt.getElementById(„zold”).style.color = „#00BB00”;
```

Így az objektum színe sötétebb zöld lesz.

A *setProperty* eljárással is meg lehet ugyanezt csinálni. Ezt akkor érdemes használni, ha nem működik a fenti megoldás, vagy nem tudjuk, hogy milyen eljárásal kell megváltoztatni, de tudjuk a tulajdonság nevét. Akkor a következőképpen kell eljárni:

```
docuemnt.getElementById(„zold”).style.setProperty('color', '#00BB00', );
```

Az első helyen a tulajdonság neve áll, a másodikon az új értéke, a harmadikon pedig egy üres string. Ezt beágyazott objektumokra is lehet használni. Abban az esetben a *contentDocument* objektumon keresztül kell rá hivatkozni. Ez a következőképpen történik:

```
document.getElementById(„map”).contentDocument.getElementById(„item1”).style.setProperty('fill', '#334455', );
```

A fenti sor megváltoztatja a *map* nevű objektumban lévő vektoros térképen az *item1* poligon kitöltését. Ezt a módszert használtam tematikus térképek elkészítéséhez.

Lehet a stílussal adott objektum pozicionálóját vagy a rétegszerkezetben elfoglalt helyét is definiálni. Ezzel lehet pl. lebegő objektumokat definiálni. Ehhez a *position: absolute* tulajdonságot megadjuk neki a stíluslapon, majd a javascriptel változtatjuk a *top* és *left* tulajdonságokat. Ezt használtam a tooltip objektumhoz, hogy minden réteg fölé helyeztem, és a *top* és *left* tulajdonságait az egérmutatóhoz viszonyítottam.

A stílusok változtatásával értem el pl. a piros keret megjelenését az aktív objektumok körül. Minden aktív objektum körül van egy átlátszó keret (*border-color: transparent;*). Ha az egeret fölé viszem, megváltoztatja ezt a tulajdonságot #FF0000-re a következő paranccsal:

```
obj.style.setProperty('border-color', '#FF0000', );
```

Ha elhagyja az egér az objektumot, akkor pedig visszaalakítja:

```
obj.style.setProperty('border-color', 'transparent', );
```

A keresési eredményben hasonlóan működik a sor kijelölése is, csak ott a *background-color* tulajdonságot változtatja meg az eseménykezelő, és nem a *border-color* tulajdonságot.

Egy további alkalmazása a stíluslapoknak a *display* attribútum tulajdonságai. Ha *inline*, akkor megjelenik az objektum, ha pedig *none* akkor el van rejtve. Amikor a honlap betölt, akkor betölti az összes objektum tartalmát amit csak tud, és amikor a felhasználó egy olyan funkciót aktivál, amihez az a tartalom kell, akkor csak ezt a tulajdonságot változtatja meg a program, és így megjelenik az objektum. Egy másik esetleg útban lévő objektum ugyanezen tulajdonságának a változtatásával elrejtethető. Így a funkciók sokkal gyorsabban jelennek meg. Ezt használom mind a *res* mind a popup ablakban lévő objektumok elrejtéséhez és megjelenítéséhez. Sőt magát a popup ablakot is így rejtem el. További előnye ennek a módszernek, hogy a rejtett objektum tartalma nem törlődik, csak átmenetileg láthatatlanná válik, így újra elő lehet hozni ugyanolyan állapotban, mint

mielőtt el lett rejtve. Így lehet pl. az utolsó keresési eredményt ismét felhozni, ha azt már valamiért nem lehet a képernyőn látni.

## 7. Dataset átadása a kliens részére

A dataset egy adatbázis-lekérdezés következtében létrejött tábla. Ennek elemeit lehet közvetlen felhasználni, vagy újabb lekérdezéseket végrehajtani rajta. A kliensnek, a rendszer felépítéséből következően nincs közvetlen hozzáférése az adatbázishoz. Viszont bizonyos esetekben nem elég egy szöveges eredményt leküldeni, hanem szükség van az eredménytáblára a kliensen. A webes szabvány által is támogatott xml erre alkalmasnak bizonyult. A webböngészők legtöbbje ma már támogatja.

A webböngészőben a szerverről visszakapott információ lehet szöveges vagy xml formátumban. Bár igaz, hogy az xml is egy szöveges dokumentum, de ha a böngészőnek megadjuk, hogy xml a várt adat, akkor a böngésző képes az attribútumok változóba történő kiolvasására. Továbbá lehet az állomány egyes soraira hivatkozni, ill. meg lehet határozni a sorok számát. Egy xml állományban lehet akár több dataset is, mivel lehet a tag-ek neve szerint is szűrni. Tehát ebből következik, hogy minden dataset-hez tartozik egy sor tag, ahol minden tag egy bejegyzés a datasetbe, és a tag-ekhez tartoznak attribútumok, ahol minden attribútum egy mező a dataset táblában.

A php nyelv egy olyan felületet biztosít, ami lefuttat egy programot a szerveren, és az eredményt, legyen az bármi, elküldi a böngészőnek. Tehát ha a böngészővel rámutatok egy php állományra, akkor a szerveren lefut a program, majd az eredmény megjelenik a böngészőben. Az eredmény viszont lehet szinte bármi. Hogy mi, azt a php állomány fejlécében kell megadni a következőképpen:

```
header("Content-type: text/xml");
```

A fenti példában megadjuk, hogy a php által generált állomány egy xml állomány lesz. Így a visszakapott tartalomnak viszont meg kell felelni az xml követelményeinek, ugyanakkor a böngészőben megadhatjuk, hogy a kért állomány xml formátumban van. Bár az xml formátum nagyon hasonlít a html-re, két apróságra mégis nagyon oda kell figyelni:

Ha egy tag nincs lezárva, akkor a böngésző egy hibaüzenetet ír ki, és a honlap struktúrája onnantól általában felborul, mert ezt magára a honlapra írja.

A másik, ami hasonló következményekkel járhat, hogy entityket csak akkor lehet használni, ha azokat a stíluslapban definiáltuk. Tehát vagy definiáljuk őket, vagy nem használunk benne entityket. Én az utóbbit választottam. Annak érdekében, hogy a karakterkódolással ne legyen gond,

minden olyan stringhez, amiben lehettek mellékjeles karakterek escape karaktereket rendeltem. Ezeket bár el kellett távolítani a végén, de ezt a böngészőben meg lehetett tenni. Tehát egy alapvető, a fenti célokra használatos xml állomány a következőképpen kell hogy kinézzen:

```
<?xml version="1.0"?>
<DATA>
<DATASET mezo1="szoveg" mezo2="szoveg" mezo3="szam"/>
<DATASET mezo1="szoveg" mezo2="szoveg" mezo3="szam"/>
<DATASET mezo1="szoveg" mezo2="szoveg" mezo3="szam"/>
<DATASET mezo1="szoveg" mezo2="szoveg" mezo3="szam"/>
</DATA>
```

A fenti példában is látható, hogy mellékjeles karaktereket nem használtam. A mező neveknél ez nem gond, mivel sehol se fognak megjelenni, de ha – tegyük fel – a tartalmuk olyan, ami valahol közvetlen a honlapon megjelenik, akkor ilyen módon az ékezeteket leahagyni nem lehet. Erre jók az escape karakterek.

A másik dolog, ami látszik a fenti példában, az a tag-ek lezárása. Azok a tag-ek, amelyeknek nincs külön lezáró tag-jük, azoknak a végén ott van a / karakter. Ez azt jelenti, hogy azok le vannak zárva. Ezek közé viszont utólag sem lehet semmit beilleszteni, vagyis tartalmat adni nekik nem lehet. Bár igaz, ebben az esetben az nem is célunk. Az első sor meg egy olyan sor, ami minden xml állomány első sorában ott kell legyen. Továbbá fontos, hogy érzékeny kis és nagy betűkre. Bár igaz, hogy mindegy, hogy a tagnevek csupa kisbetűk, vagy csupa nagybetűk, de fontos, hogy mindig ugyanolyanok legyenek. Tehát egy ilyen állományt kell visszaadjon nekünk a php.

A phpnek mi kétféleképpen tudunk küldeni attribútumokat. Post és get módszerekkel. Mivel a diplomamunkámban a get módszert használtam, ezért ezt fogom most kifejteni.

A get változóba a böngésző címsorába is be tudunk írni értékeket. Ez a következőképpen történik:  
*oldal.php?att1=ertek1&att2=ertek2&att3=ertek3*

Itt is, az előzőhöz hasonlóan fontos, hogy ha mellékjeles karaktereket akarunk használni, akkor használjunk escape karaktereket, mivel az átviteli közegben sokszor elromolhat a karakterkódolás. Mind az attribútumok neveit, mind az értéküket elküldi a szervernek a program.

Ezek a phpben egy tömbbe kerülnek, ami az attribútumok neveivel lesz indexelve. Az attribútumok neveiben nem ajánlott az ékezetek használata. A tömb neve *\$\_GET*

Tehát a következőképpen tudjuk az értéket kiolvasni:

```
$att1=$_GET['att1'];
```

```
$att2=$_GET['att2'];
```

Fontos, hogy ez is érzékeny a kis- és nagybetűkre, és a *\$\_GET* mindig csupa nagybetűvel kell legyen.

A php, mivel a szerveren fut, kap közvetlen hozzáférést az adatbázishoz. Sql lekérdezéseket hajthat végre. Ezek a lekérdezések a php szempontjából stringek, amiket elküld az adatbázis szervernek.

Tehát mondjuk azt, hogy az adatbázisból, pontosabban a *katalogus\_bov* táblából minket az érdekel, hogy mi a térkép leírása, és ki készítette a térképet, és méghozzá a magyar nyelvű bejegyzést akarjuk, és tudjuk mondjuk a térkép sorszámát, akkor a következő lekérdezést hajtjuk végre:

```
select `leiras`,`szerzo` from `katalogus_bov` where `sorszam`=42 and `nyelv`='HU';
```

Fontos, hogy a mező neveket, valamint a tábla nevet visszafelé aposztrófok közé tettem. Ez fontos, mivel ezzel garantálja a rendszer, hogy semmiképpen nem fogja az ott lévő szöveget egy, az adatbázis szerver számára parancsnak értelmezni. A 42 nincs idézőjelben, mivel az integer típus, viszont a HU az igen, mivel az string. Mivel maga a lekérdezés a php számára egy hosszú string, így ezt el is tudjuk készíteni a php-nek olyan függvényeivel, amelyek a szöveg kezelésére lettek kitalálva. Mivel jelenleg egyik megadott mező sem tartalmazhat ékezeteket, így azokat a kliensen sem kódoltuk be escape karakterekkel, így azokat el sem kell távolítani. Tehát elkészítjük hozzá az sql lekérdezést:

```
$sqlstring = "select `leiras`,`szerzo` from `katalogus_bov` where `sorszam`=42 and `nyelv`='HU';"
```

Viszont itt nem a kliens által megadott attribútumokat használtuk, hanem két konstans értéket.

Ahhoz, hogy a kliens le tudja adni az értékeket, definiálni kell a bemeneti paraméterek neveit. Az egyszerűség kedvéért legyen a két paraméter *sorszam* és *nyelv*, minden ékezet nélkül. Tehát akkor a php állományra a következő képen kell hivatkozni:

```
oldal.php?sorszam=42&nyelv=HU
```

Most a *\$\_GET* tömbnek lesz egy *sorszam* és egy *nyelv* nevű eleme. Tehát, ha a fenti példában használt lekérdezést ennek megfelelően szeretnénk módosítani, akkor a következőképpen írjuk meg:

```
$sqlstring = "select `leiras`,`szerzo` from `katalogus_bov` where `sorszam`={$_GET['sorszam']} and `nyelv`='{$_GET['nyelv']}';"
```

A kapcsos zárójelek között a php képes tömböket értelmezni abban az esetben, ha az egész string dupla idézőjel között van. A string a tömbben nem meglévő elemei, mint pl. az idéző jel a nyelv elem értéke körül ugyanúgy ott vannak. Ezt kell elküldeni az adatbázis szervernek, de előbb ahhoz csatlakozni kell. Ez a következőképpen zajlik:

```
mysql_connect('hoszt','felhasznalo','jelszo')
```

Ha az adatbázis szerver ugyanazon a gépen van, mint a webszerver, akkor a hoszt 127.0.0.1. Ne feledjük, hogy a php a szerveren fut, tehát minden hálózati hivatkozás a szerverről történik, tehát



a címeket annak megfelelően kell megadni. A felhasználó névnél érdemes egy olyan felhasználót megadni, aminek csak a szükséges jogai vannak engedélyezve az adatbázis szerveren. Ebben a példában csak a select jog szükséges. A jelszó, mivel az ebben az állományban nincs semmilyen módon kódolva, lehetőleg ne egyezzen más teljes jogú felhasználó jelszavával, és semmiképpen ne az admin jelszóval. A felhasználó nevekben és jelszavakban lehetőleg kerülni kell az ékezetek használatát. Ezek után ki kell választani azt is, hogy a szerveren melyik adatbázist akarjuk használni. Ez a következőképpen történik:

```
mysql_select_db("adatbazisnev");
```

A felhasználónevekhez és -jelszavakhoz hasonlóan az adatbázis nevében is lehetőleg kerülni kell az ékezetek használatát. Tegyük fel, hogy az adatbázisszerver ugyanazon a gépen van, mint a webszerver, és a felhasználónév, amivel be tudunk lépni, *juzer*; a jelszó pedig *almafa*, az adatbázis neve pedig *demogmap*, akkor a csatlakozás a következő két sorral történik:

```
mysql_connect('127.0.0.1','juzer','almafa')
```

```
mysql_select_db('demogmap');
```

Ezek után elküldhetjük a korábban elkészített lekérdezésünket a szervernek:

```
$query = mysql_query($sqlstring);
```

Az eredmény a *\$query* változóba kerül. Bár jelen esetben csak egy sort kapunk vissza a szervertől, de kaphatunk akár többet is.

A következő sorral lehet egy ciklust létrehozni, ami addig fut, amíg vannak sorok az adatbázisban. A ciklus soronként megy végig az eredménytáblán.

```
while($assoc_query=mysql_fetch_assoc($query))
```

A ciklusmagban létrejön egy *\$assoc\_query* nevű tömb, aminek az elemei a dataset mező nevei szerint lesznek indexelve. Így a ciklus futásakor egyszerűen egy print paranccsal létrehozzuk a már korábban tárgyalt xml állomány sorait. Tehát a ciklusmag mindössze egy sorból fog állni:

```
print "<DATASET leiras=\"{$assoc_query['leiras']}\" szerzo=\"{$assoc_query['szerzo']}\"/>";
```

A \ karakter a print parancs számára escape karakter. Ez azért kell, hogy ott ne zárja le a szöveget, hanem küldje ki az idézőjelet az eredménybe. Így viszont entitykkel megy ki, mivel az adatbázisban úgy van tárolva. Ezt a `html_entity_decode` függvénnyel tudjuk megszüntetni. Ezek után az escape karaktereket az `urlencode` függvénnyel tudjuk beletenni. Akkor a sor a következőképpen módosul:

```
print ("<DATASET leiras=\"\".urlencode(html_entity_decode($assoc_query['leiras'])).\"\" szerzo=\"\".urlencode(html_entity_decode($assoc_query['szerzo'])).\"\"/>");
```

Mivel itt függvények is vannak benne, így a kapcsos zárójeleket nem tudjuk használni, viszont az egész belső függvényt íves zárójelek közé kell helyezni. A php-ben a szövegtoldó a pont. Viszont így egy hiányos xml állományt kapunk, ami nem jó. Mivel az első sorba a `<?xml`

`version="1.0"?>` sort akarjuk, így a ciklus megkezdése előtt a következő sort írjuk be a php állományba:

```
print "<?xml version='1.0'?">";
```

Ezt a sort nem lehet a `<?php` sor előtt, vagy azon kívül megadni, mivel a `<?` -et is úgy értelmezi a program, mintha php kód következne, és hibajelentést kapunk.

Mivel még mindig hiányzik egy sor a dataset előtt, még egy sort be kell írjunk a ciklus megkezdése előtt. Bár e nélkül is lefut, így talán átláthatóbbá válik az xml állomány.

```
print "<DATA>";
```

Mivel ez megnyit egy tag-et, amit le kell zárni, így a ciklus lefutása után kell egy sor, ami elküldi a kliensnek a lezáró tag-et is:

```
print "</DATA>";
```

Ezek után még az adatbázis kapcsolatot kell lezárni. Ez a következő paranccsal történik:

```
mysql_close();
```

Tehát a kész program a következőképpen nézne ki:

```
<?php
header("Content-type: text/xml");
$sqlstring = "select `leiras`, `szerzo` from `katalogus_bov` where
`sorszam`={$_GET['sorszam']} and `nyelv`='{$_GET['nyelv']}';";
mysql_connect('127.0.0.1','juzer','almafa');
mysql_select_db('demogmap');
$query = mysql_query($sqlstring);
print "<?xml version='1.0'?">";
print "<DATA>";
while($assoc_query=mysql_fetch_assoc($query))
    print ("<DATASET
        leiras='\"'.urlencode(html_entity_decode($assoc_query['leiras']))."\"
        szerzo='\"'.urlencode(html_entity_decode($assoc_query['szerzo']))."\""/>");
print "</DATA>";
mysql_close();
?>
```

A fenti példa, futáskor, a következő eredmény adja:

```
<?xml version="1.0"?>
<DATA>
<DATASET leiras="térkép leírása escape karakterekkel" szerzo="térkép szerzőjének a neve
escape karakterekkel"/>
```

</DATA>

Ezt hibátlanul fogja értelmezni a böngésző.

Most már csak a böngészőnek kell megadni, hogy küldje el a kérést a szerverre, és utána értelmezze a választ.

Az alábbi példa feltételezi, hogy valahol a kliens oldali alkalmazásban az *xhttp* objektum el lett készítve. Erről a kliens oldali függvények mellékletben lehet olvasni az *init* függvény alatt. Tehát létre kell hozni a kapcsolatot a webszerver felé. Mivel magát a honlapot is azon a szerveren tároljuk, így a szerver hivatkozását nem kell külön megadni. Tehát a következő sorral történik a kapcsolat megnyitása:

```
xhttp.open("GET", "oldal.php?sorszam="+sorszam+"&nyelv="+lang,false);
```

A fenti sor első eleme megadja, hogy a *get* módszerrel hajtjuk végre a lekérdezést. A második elem annak az állománynak a neve, amitől kérjük az adatot. Mivel a *get* módszer a paramétereket is egyből az állománynév mellé adja meg, így azok is itt vannak megadva. A fenti példa feltételezi, hogy a keresett sorszám a *sorszam* nevű változóban van eltárolva, a nyelv pedig a *lang* nevű változóban. A következő érték hamis, mivel nem készítünk külön fejléct az átvitelre. Ez a *post* módszer esetén lenne szükséges.

Utána elküldjük a következő sort:

```
xhttp.send("");
```

A fenti sor a Firefox esetében jó, az IE nem akar null stringet benne. A null string azért van, mert itt már nem adunk meg paramétereket, mivel megadtuk őket az *open* sorban. Itt akkor adnánk meg bármit is, ha a *post* módszert használnánk.

Ezután a választ XML-be kérjük:

```
xmldoc = responseXML;
```

Mivel ennek az elemeihez közvetlen hozzáférést szeretnénk, ezért egyből kiadjuk a következő parancsot is:

```
rootNode = xmldoc.documentElement;
```

Ennek lehet több sora is, így kell hogy tudjuk a sorok számát. A *dataset* tag neve *dataset*, tehát így pontosabban azt kell tudjuk, hogy hány darab *dataset* nevű tag van. Ez a következő sorral történik:

```
datanum = rootNode.getElementsByTagName("dataset").length;
```

Végigfuttatunk egy ciklust az összes elemen:

```
for(i=0;i<datanum;i++) {  
    dataset = rootNode.getElementsByTagName("dataset").item(i);  
    leiras = unescape(dataset.getAttribute("leiras"));  
    szerzo = unescape(dataset.getAttribute("szerzo"));  
    .....  
    .....  
}
```

Fontos, hogy az *unescape* függvény ott legyen, mivel escape karaktereket tettünk bele még a szerveren. Ezeket el kell távolítani. A ciklus végigfut minden soron. Azért kell a sorok száma, hogy az utolsó sor után ne fusson tovább, különben hibaüzenetet kapunk. Magát az tag-et az xml állományból betölti a *dataset* változóba, majd a *getAttribute* eljárással az egyes attribútumok értékét tölti be a megadott változóba. A ciklusmagból még hiányoznak azok az eljárások, amelyek valamit csinálnak az adatokkal. Ezekhez fontos tudni, hogy minden alkalommal, amikor lefut a ciklus, az előző futás eredményeit kitörli az ott definiált változókból. Tehát a kliens oldali program egészben így nézne ki:

```
xhttp.open("GET", "oldal.php?sorszam="+sorszam+"&nyelv="+lang,false);  
xhttp.send("");  
xmldoc = responseXML;  
rootNode = xmldoc.documentElement;  
datanum = rootNode.getElementsByTagName("dataset").length;  
for(i=0;i<datanum;i++) {  
    dataset = rootNode.getElementsByTagName("dataset").item(i);  
    leiras = unescape(dataset.getAttribute("leiras"));  
    szerzo = unescape(dataset.getAttribute("szerzo"));  
    .....  
    .....  
}
```

## 8. Az adatbázis és az svg közötti kapcsolat

Az előző fejezetben kifejtettem, hogy a datasetet hogyan tudjuk a kliensre leküldeni. Ennek egyik – és ezen a honlapon leggyakrabban használt – alkalmazása a honlapon lévő vektoros állományok adatbázissal való kapcsolatának létesítése. Ez a folyamat a kliensen történik az ott már betöltött svg állományon.

Az svg formátum egy módosított változata az xml formátumnak. Nagyon hasonlít az xml-re és az xhtml-re. Minden objektum egy tag, amibe lehetnek beágyazva további tag-ek. A tag-eknek vannak attribútumai és tulajdonságai. A tulajdonságok alatt azt kell érteni, amiket a stíluslapon is definiálhatunk, és ezzel az egyes objektumok megjelenését határozzuk meg. Ezeket a Javascript segítségével meg tudjuk változtatni.

Az első dolog, amit meg kell tennünk az, hogy elhelyezzük a megfelelő helyen, a honlapon. Ez a következőképpen történik:

```
<OBJECT data="állománynév.svg" type="image/svg+xml" id="azonosító" height="magasság" width="szélesség"></OBJECT>
```

Ahol az

*állománynév.svg* az svg állomány neve kiterjesztéssel együtt,

*azonosító* az objektum azonosítója. Ez magát a beágyazott objektumot azonosítja.

*magasság* az objektum magassága,

*szélesség* az objektum szélessége.

A magasság és a szélesség megadása nem kötelező. Ha csak egyiket adom meg, akkor a másikat úgy igazítja hozzá a program, hogy a kép ne torzuljon. Mivel térképeket helyezünk be, ez fontos, így hacsaknem biztosak vagyunk az arányokban nem érdemes mindkettőt megadni. A lezáró tag-et mindenképpen külön tag-be kell helyezni, és közvetlen utána; mivel az <OBJECT> tag-nek van lezáró tag-je, és ahogy vége van az svg adatnak le kell zárni, különben az utána beírt html kódot is megpróbálja mint az svg állomány részét értelmezni, és ez problémát okozhat.

Ezek után az állomány fejlécében létrehozunk egy függvényt, ami ennek az állománynak az attribútumait megváltoztatja. Az előző fejezetből a ciklust bővítjük tovább. Viszont előtte elő kell készíteni az állományhoz való hozzáférést. Ez a következőképpen történik.

```
map = document.getElementById("azonosító").documentElement;
```

Ezek után a *map* objektumon keresztül hozzáférésünk lesz az svg állomány objektumaihoz. Majd a ciklust bővítjük, amint az előző fejezetben tárgyaltam. Fontos, hogy azok, amik ott le vannak írva, a ciklus előtt is fontosak hozzá, mivel az ott szerzett dataset-et használjuk fel. Viszont itt más attribútumokat kérünk le, mivel köztük kell legyen a poligonok azonosítója is:

```

for(i=0;i<datanum;i++) {
    dataset = rootNode.getElementsByTagName("dataset").item(i);
    vmkod = unescape(dataset.getAttribute("vmkod"));
    vmtart = unescape(dataset.getAttribute("vmtart"));
    .....
    .....
}

```

Ezzel a ciklussal megkapjuk az xml állományban lévő *vmkod* és *vmtart* attribútumok értékét. Mind a kliensen, mind a szerveren ennek megfelelően kell módosítani a szkriptet.

A *vmkod* attribútumban a vármegye kódja található, ami megegyezik a vármegye poligonjának azonosítójával. Ez alapján ki tudjuk keresni a poligont. A *vmtart* meghatározza, hogy a vármegyét a keresett térkép ábrázolja-e vagy sem. Ez alapján döntjük el, milyen kitöltést kap a poligon.

## 9. Szöveges adat utólagos beágyazása a honlapba

Az ajax technológia alkalmas a szerverről kapott adatok alapján egy adott html oldal részének megváltoztatására anélkül, hogy az oldalt újra kellene tölteni. A fenti leírásban már láthattuk, hogy hogyan kell az ajaxsal dataseteket visszatéríteni a webböngésző számára. Lehetséges viszont teljes objektumok tartalmát kicserélni, vagy a meglévő tartalomhoz hozzátoldani. Az ajax a szerverre küld egy kérést, és megadja a kezdeti paramétereket hozzá. A szerveren lefut a szerver oldali szkript, majd a visszakapott eredményt behelyezi az objektumba. Ehhez előbb létre kell hozunk egy *xhttp* objektumot. A Firefox webböngészőbe ez a következő paranccsal történik:

```
var xhttp = new XMLHttpRequest();
```

Ezek után megnyitjuk a kapcsolatot a szerver irányába. Ezt a következő paranccsal tehetjük meg:

```
xhttp.open(„GET”, „szript.php?param=ertek”, false);
```

Majd elküldjük:

```
xhttp.send(„”);
```

Ezek után ha nem xml az állomány, mint a fenti példában, akkor a *responseText* eljárással kapjuk vissza nyers szövegben:

```
visszater = xhttp.responseText;
```

Ahhoz, hogy valahova el tudjuk helyezni, kell egy üres objektum. Ezt a html oldal *<BODY>* tag-jén belül a *<DIV>* tag-gel hozzuk létre. Meg kell adnunk egy azonosítót, hogy tudjunk rá hivatkozni:

```
<DIV id="objektum"></DIV>
```

Fontos, hogy lezárjuk az üres objektumot, hogy mikor belehelyezi a tartalmat, akkor tudja a böngésző, hogy hol van az objektum eleje és a vége. Ha bele akarjuk helyezni az eredményt, akkor a következő sorra lesz még szükségünk a javascript függvényünkben:

```
document.getElementById(„objektum”).innerHTML = visszater;
```

Tehát a teljes függvény következőképpen néz ki:

```
var xhttp = new XMLHttpRequest();  
xhttp.open(„GET”, „szript.php?param=ertek”, false);  
xhttp.send(„”);  
visszater = xhttp.responseText;  
document.getElementById(„objektum”).innerHTML = visszater;
```

Ha szeretnénk görgetősávokat az objektumon, akkor az *overflow:auto*; paramétert kell megadni a stíluslapon. Ekkor szükség esetén megjelennek a görgetősávok. Ha azt szeretnénk, hogy semmiképpen ne jelenjenek meg a görgetősávok, akkor az *overflow:none*; paramétert kell megadni a stíluslapon. Annak érdekében, hogy a karakterkódolással ne legyen a gond, az ajaxszal lehívni szánt php állományok fejlécébe a *Content-type: text/plain* van megadva, abban az esetben is, ha annak tartalma valójában text/html. Így érhető el, hogy az entityket ne alakítsa át, majd amikor be lesz szűrve az objektum belsejébe, átalakítja a megfelelő karakterkódolásba, így mindenképpen jól jelennek meg a karakterek.

Bár ez a megoldás kinézetre hasonlít a frame-ekre, viszont sok előnye van a frame-ekkel szemben. Az első, hogy be van ágyazva a html dokumentumba, tehát a meglévő html dokumentum tulajdonságait veszi át. Ha ezek mégsem jók, akkor a stíluslapon meg tudjuk ezeket változtatni. A html oldal globális változói és függvényei mind közvetlen elérhetők lesznek. Az objektumok helye, mérete is változtatható, ami pl. az ablak méretének megváltoztatása esetén következhet be, vagy az egérmutató adott objektum fölé húzása esetén. Ezzel lehet lebegő objektumokat létrehozni, amikkel pl. tooltipeket lehet készíteni. Egy adott objektum több célra is használható, mivel az így berakott tartalom bármikor kitörölhető. Az *innerHTML* eljárás értéke egy string, tehát bármilyen string eljárást lehet rajta végezni. Akár közvetlen a javascriptból is megadható neki egy konstans vagy változó érték. Ezt sokszor akkor használjuk, ha pl. az objektumban egy képet akarunk megjeleníteni. Akkor a következőképpen tudjuk megadni:

```
objektum.innerHTML = '<IMG src="kep.png"/>';
```

Ekkor az objektum meglévő tartalma törlődik, és helyette megjelenik a *kep.png*.

## 9.1 A szerveren tárolt rászteres képek megjelenítése

A szerveren a rászteres térképek és fotók nem egy, a weben nyilvános mappában vannak tárolva. Annak érdekében, hogy a weblapon is láthatók legyenek a kívánt képek, egy php eljárás hívja le ezeket a szerverről. A php-nek van hozzáférése a szerveren lévő nem nyilvános adatokhoz is. Annak érdekében, hogy a kliens tudja, hogy az adat egy kép, a következő sorral kell kezdeni a php szkriptet:

```
header(„Content-type: image/jpeg”);
```

vagy adott esetben:

```
header(„Content-type: image/gif”);
```

Így a böngésző tudja, hogy egy képet fog kapni. A *readfile* eljárás egy megadott állományt olvas ki egy meghatározott helyről a szerveren, és kiküldi a kimenetre. Az eljárás a következőképpen néz ki:

```
readfile(„/home/kepek/kep.jpg”);
```

Fontos, hogy az állományra való hivatkozást úgy kell megadni, ahogy a szerveren helyileg hivatkozunk rá. Ha az állománynevet egy paraméterben akarjuk megadni, akkor a használhatjuk a *get* paramétert:

```
readfile(„/home/kepek/{$_GET['img']}”);
```

A fenti példában a php szkript *img* attribútumnak megfelelő állományt fogjuk megkapni. A teljes szkript a következőképpen néz ki:

```
header(„Content-type: image/jpeg”);
```

```
readfile(„/home/kepek/{$_GET['img']}”);
```

Ha a *kep.jpg*-t szeretnénk vele megtekinteni, és a szkriptünk neve *kep.php* akkor a következőt kellene beírni a webböngészőbe:

```
cím.tld/kep.php?img=kep.jpg
```

Ha a *kep.jpg* a szerveren a */home/kepek/* mappába van, akkor megjelenik a webböngészőn a kép. Ezt ugyanígy behelyezhetjük az *IMG* tag-be egy html állományba:

```
<IMG src=„kep.php?img=kep.jpg”/>
```



## 10. Tematikus térképek készítése

A program alkalmas néhány egyszerű tematikus térkép készítésére. A fent leírtakat használjuk fel hozzá, tehát az egyes poligonok attribútumait változtatjuk meg egy az adatbázisból generált xml állomány segítségével. A program négy fajta tematikus térképet tud készíteni: azon térképek száma, amelyek adott vármegyét ábrázolják, nem ábrázolják vagy részben ábrázolják; vármegyék össznépsége és egy adott nemzetiség aránya; hogy egy adott térkép mely vármegyét ábrázolja és melyeket nem; és végül hogy a térképet tartó intézet hol helyezkedik el. Tehát az első kettő szerkesztése általános adatok alapján történik, ami bármikor elkészíthető, a második kettő viszont egy adott térképhez kapcsolódik, tehát csak akkor készíthető el, ha ki van választva egy térkép az adatbázisból.

### 10.1 Adott vármegyét ábrázoló térképek száma

Ha ezt a funkciót választjuk ki, akkor feljön a képernyőre egy történelmi térkép, ami Magyarország vármegyéit ábrázolja az 1914-es állapot szerint, és alatta egy menü, amiben ki lehet választani azt az intézetet, amelynek gyűjteményében lévő térképeket vizsgálunk a tematikus térkép elkészítéséhez. Az adatbázis *statszin* táblájában van nyolc darab szín, ami a nyolc kategóriának a színek kódjait tartalmazza. A *kat* oszlop tartalmazza a kategóriák számát nullától hétig. A *t* oszlop a színek kódokat ahhoz a térképhez, ami azt mutatja, hogy hány térkép ábrázolja a vármegye teljes területét. Az *r* oszlop a színek kódokat ahhoz a térképhez, ami azt mutatja, hogy hány térkép ábrázolja a vármegye területének egy részét. Végezetül az *e* oszlop azokat a színeket, amelyek annál a térképnél kellene, ami azt mutatja be, hogy hány térkép egyáltalán nem ábrázolja a vármegye területét.

Ha a *statkat.php int* paramétere be van állítva, akkor az eljárás az összes térképet megszámlolja, amely adott intézetben van, és felosztja aszerint, hogy hány térkép ábrázolja a vármegye teljes területét, hány térkép ábrázolja csak egy részét, és hány térkép nem ábrázolja a vármegyét. Mindhárom kategóriánál megvizsgálja, hogy melyik vármegyére esik a legalacsonyabb érték az adott kategóriában, és melyikre a legtöbb. A két érték közötti különbséget felosztja hét részre, és a kategóriákat egytől hétig hozzárendeli az egyenlő részekhez. Attól függően, hogy melyik vármegye melyik kategóriába esik, hozzárendeli a kategória színét a megfelelő csoportból (*t*, *r* vagy *e*). Ha az adott vármegye értéke nulla a nulladik kategóriába kerül.

Az xml állomány kimenetében lesz egy *dataset* tag minden vármegyére. Az *id* attribútum a

vármegye kódja. A *tcolor* azon térképen használatos színkód, amelyen láthatjuk, hogy hány térkép ábrázolja egészében a vármegyét. Az *rcolor* azon térképen lévő színkód, ami a részben ábrázolt vármegyét mutatja be. Végül az *ecolor* azon térképen használatos színkód, ami azt mutatja be, hogy hány térkép nem ábrázolja az adott vármegyét. Ha nincs megadva az *int* paraméter, akkor az összes, az adatbázisban lévő térképet számításba veszi, és úgy készíti el a tematikus térképet.

## **10.2 Adott nemzetiséget bemutató térkép készítése**

Ha a felhasználó erre a funkcióra kattint, akkor a popup ablakban megjelenik egy történelmi térkép, ami Magyarország vármegyéit mutatja be az 1914-es állapot szerint. Alatta egy menü jelenik meg, amiben ki lehet választani a nemzetiségek közül az ábrázolni kívánt nemzetiséget. Ha a *nepszam.php* szkriptet a *nat* paraméterrel futtatjuk le, pontosabban átadjuk neki a lenti menü értékét, akkor az eredményben visszaadja az összes vármegyére vonatkozólag azt, hogy mi a vármegye kódja, az össznépessége, és azt is, hogy mi az adott nemzetiség lélekszáma az adott vármegyében. Ebből a kliensen kiszámolja az arányt, majd a kliensen előre definiált jelkulcs alapján megváltoztatja a vármegyék kitöltését. Ennek a pontos leírása a fejezet elején található. Minden alkalommal, amikor a menü kiválasztott értéke megváltozik, a térképen újrakészíti a jelkulcsot azon nemzetiségnek megfelelően, amely ki lett választva. A jelkulcs a kliens oldali függvények mellékletben tekinthető meg.

## **10.3 Kiválasztott térképet tartó intézet megjelölése a térképen**

A keresési eredmény eseménykezelőjében benne van az, hogy melyik térkép ábrázolja az intézet helyét. Ezt a *katalogusok* tábla *tk* mezőjéből veszi arra az intézetre vonatkozóan, ami a *lelohely* mezőben van megjelölve a *katalogus\_bov* tábla adott térképre vonatkozó bejegyzésében. Ez alapján kiválasztja a megfelelő térképet, és megjeleníti az adatok ablakban. Az *archives.php* *lel* attribútumában megadja azon intézet kódját, ahol a térképet őrzik. A visszakapott xml állomány *polygon* attribútumában benne lesz a térképen annak a poligonnak az azonosítója, ahol az intézet található. Ennek a poligonnak a kitöltését vörösre (990000) cseréli, így megjelöli azt.

## 10.4 A térképen ábrázolt vármegyék bemutatása

Ekkor egy, a történelmi Magyarország vármegyéit bemutató térképet használunk alaptérképnek. A *vmsz.php* szkript *tkszam* paraméterébe megadjuk a térkép sorszámát, aminek a kiterjedését szeretnénk bemutatni. A szerver oldali szkript visszaadja vármegyénként, hogy a térkép az adott vármegyét egészben vagy részben ábrázolja, vagy hogy az adott vármegyét nem ábrázolja. A *vmkod* attribútumból kiolvassuk a vármegye kódját, a *tart* attribútumból pedig egy értéket ami „t”, ha a vármegye teljes területét ábrázolja a térkép; „r” ha a területének csak részét ábrázolja; és „e” ha a vármegyét nem ábrázolja. A jelkulcs a programban van meghatározva. Ha a *tart* attribútum értéke „t” akkor kékre (1C2FD8) cseréli a vármegye kitöltését, és „r” akkor zöldre (147C34), ha pedig „e” akkor vörösre (BB0000).



A jobb oldalon látható térképhez elkészített vármegye áttekintő

## 10.5 Az Európa térkép feliratainak elkészítése

Az Európa térképen a névrajz nyelve attól függ, hogy a honlap nyelve mire van beállítva. Az *országok* táblában van egy *jel* mező, amiben az adott ország kódja van, valamint három mező: *HU*, *DE* és *EN*, amelyekben az ország neve az adott nyelven UFT-8 kódolással. Az Európa térképen egy általános grafikai programban (fontos, hogy tudjon svg állományokat kezelni) elkészítettem a térkép névrajzát. Minden névrajzi elemnek *tországkód* azonosítót adtam. Elhelyeztem az összes elemet a megfelelő helyen, majd elmentettem az állományt svg formátumban, és bezártam a programot.

Megnyitottam az imént elkészített állományt egy szövegszerkesztőben, és megkerestem a megfelelő azonosítóval rendelkező objektumokat, majd kitöröltem a szöveges tartalmukat. Ha ekkor megnyitottam az svg állományt, ugyanúgy nézett ki, mint a névrajz elkészítése előtt, viszont a névrajzi elemek ott voltak, csak mivel üresek voltak, nem látszottak. A honlapon ugyanúgy van

elhelyezve a térkép, nem a többi, tehát az <OBJECT> tag-ben (lásd feljebb). A kliens oldali szkript lekér egy xml állományt az *orszagoknev.php* szkripttől. A *lang* paraméter értékének a *lang* globális változó értékét adja meg. Visszakapja egy xml állományban az országok kódjait és neveit UTF-8 kódolással. A *kod* attribútumba az ország kódját helyezi, míg a *text* attribútumba az ország nevét UTF-8-as kódolással. Minden ország egy bejegyzés az eredménybe. Ezek után kikeresi az elemeket a *tországkód* szerint, majd a *text* attribútum tartalma alapján elkészít egy *TextNode*-ot a *createTextNode* eljárással. Ezt az *appendChild* eljárással behelyezi az üres szöveges elembe. Végigmegy a ciklus minden országon, és megjelennek az országnevek a térképen. A következő ciklust írtam hozzá. A *map3* az Európa térkép, és a *rootNode* az xml állomány gyökere.

```
for (i=0;i<dataNum;i++) {  
    dataset = rootNode.getElementsByTagName("dataset").item(i);  
    var kod = dataset.getAttribute("kod");  
    var text = dataset.getAttribute("text");  
    item4 = map3.getElementById("t"+kod);  
    var txtval = document.createTextNode(text);  
    item4.appendChild(txtval);  
}
```

## 11. A kereső működése

A keresőben két keresési mód van. Az egyik kulcsszó alapján keres, a másik az ábrázolt vármegyék alapján.

A honlap betöltésekor a program betölti a kereső, a bővített kereső és a vármegyés kereső, valamint a lelőhely térképes változatainak felületeit a *search*, *advsearch* és *terkeresofi* függvények segítségével (lásd VI. Kliens oldali függvények című mellékletben). A vármegyés keresőt és a bővített keresőt elrejtí. A *terkeresofi* függvény létrehozza a vektoros térképeken az aktív felületeket azáltal, hogy eseménykezelőket rendel hozzájuk. Az eseménykezelőkben, mivel a végrehajtandó függvények az *index.html* fejlécébe vannak beágyazva, azért az eseménykezelőknél a *parent* objektumon keresztül kell rájuk hivatkozni. Ezen túl bizonyos esetekben az aktív felületeket megjelöli egy más kitöltéssel. Ezek a folyamatok a VI. Kliens oldali függvények című mellékletben a *terkeresofi* függvény leírása alatt található.

## 11.1 A vármegyes kereső működése

Az egerrel ha rákattint a felhasználó a vármegyére, akkor az eseménykezelő a *onClickVM* függvényt hívja be. A függvény részletes leírását lásd a VI. Kliens oldali függvények című mellékeltben az *onClickVM* függvény leírásánál. Egy globális tömb van definiálva, amelyben a kliens tárolja az egyes vármegyék állapotát. A tömb neve *vm*, az indexek nevei pedig a vármegye kódok (lásd: IX. A vármegyék kódjai című mellékletben). Alaphelyzetben az összes elem értéke „m”. Ha rákattint a felhasználó a poligonra, akkor a poligon kódját a program leküldi az *onClickVM* függvénynek. Ennek segítségével megvizsgálja a poligon beállítását, és megváltoztatja. Ha a poligonhoz tartozó elem értéke a tömbben „m”, akkor „t”-re cseréli; ha „t”, akkor „r”-re; ha „r”, akkor „e”-re; ha pedig „e”, akkor „m”-re. Ezen túl a poligon kitöltését is megcseréli, hogy megfeleljen a *vm* tömb megfelelő eleme értékének. Ha a *vm* tömb adott vármegyéhez tartozó kódja „m”, akkor a vármegye világosbarna; ha „t”, akkor kék; ha „r”, akkor zöld; ha „e”, akkor pedig vörös. Ha a felhasználó az egeret fölé viszi, akkor a vármegye meglévő színének egy sötétebb változata lesz a kitöltés, míg az eger el nem hagyja az adott poligont, ezzel mutatva, hogy egy aktív felület. Azt, hogy milyen színnel kell kitölteni, a *vm* tömb megfelelő elemének értékéből tudja a rendszer. A poligonokhoz rendelt eseménykezelők biztosítják ezt az aktív felületet, és az általuk használt függvények a *vm* tömböt használják elsődleges forrásként. Többek között ezért is fontos, hogy egy globális tömb legyen.

A tömb tartalmát nem csak a poligonokra való kattintással tudjuk megváltoztatni. A térkép alatt van egy gomb, ami kijelöli az összes vármegyét. Ha erre a gombra rákattintunk, akkor a *vmall* függvényt lefuttatja a program. Ez a függvény a *vm.php* szerver oldali program segítségével az összes vármegye kódot kilistázza az adatbázisból. Ezek után végigfut az összes vármegyén, és az összes vármegye poligon kitöltését kékre cseréli, továbbá az összes vármegyéhez tartozó elemet a *vm* tömbben „t”-re cseréli. Hasonlóképpen a mellette lévő gomb is. Ez a gomb a *vmres* függvényt futtatja, ami hasonlóképpen az előzőhöz, megkapja az összes vármegye kódot az adatbázisból, majd az összes vármegye poligon kitöltését világosbarnára cseréli, és a *vm* tömb összes elemének értékét „m”-re.

A harmadik megoldás a vármegyék kijelölésére a népesség alapján történő keresés. Az adatbázisban a *nepesseg* nevű táblában (lásd: III.1 Az adatbázist felépítő táblák című melléklet) benne vannak a vármegyék 1910-es népszámlálás vármegyékre lebontott népszámlálási adatai (a szabad királyi városok is hozzá vannak adva a vármegyék adataihoz). Az oldal jobb oldalán lévő mezőkbe be lehet írni az alsó és felső határát egy adott nemzetiségnek az ösznépesseghhez képest. A program kiköti, hogy a „Min. arány” mezőben legyen a kisebb érték és a „Max. arány” mezőben a nagyobb. Nem fordítja meg a relációs jelet, ha fordítva van, viszont ez esetben egyszerűen nem

reagál.

A menüben továbbá megjelenik az 1910-es népszámlálásba felvett nemzetiségek neve, valamint egy másik menü, amiben megjelenik, hogy milyen feltételként lehet bejelölni az érintett vármegyéket. Amikor minden helyesen ki van töltve, akkor a program a mezők felett lévő kis áttekintő térképen bejelöli a vármegyéket, amelyek érintettek a megadott feltételek mellett. Ez úgy jön be, hogy minden alkalommal, amikor a mező megváltozott, a program megvizsgálja, hogy az összes feltétel megvan-e a vármegyék bejelöléséhez, és ha igen, akkor az áttekintőn bejelöli. Ha bármi változik, akkor is újra megvizsgálja ezt a tényt, és ha már valamiért nincs meg az összes feltétel, akkor törli a kijelölést az áttekintőről. Ha továbbra is megvan a feltétel, akkor is újra megvizsgálja a beadott adatokat, és azok szerint újragenerálja az áttekintőn a tematikát. A kitöltés színe attól függ, hogy a feltételben mi van megadva. Ha egészben ábrázolja van megadva, akkor kék; ha részben ábrázolja van megadva, akkor zöld; ha nem ábrázolja van megadva, akkor vörös; ha pedig nem feltétel van megadva, akkor barna lesz az áttekintőn az érintett vármegyék kitöltése.

Ha minden feltétel megvan, és rákattintunk a végrehajt gombra, akkor a program ismét megvizsgálja a már végleges feltételeket, és azon vármegyék kódját lekéri, amelyek érintettek a feltételek szerint, így a nem érintett vármegyékhez kötött feltételt nem változtatja meg. Miután megvan az érintett vármegye kódja, akkor a nagy térképen megcseréli a kitöltést ugyanarra, mint ami a megadott feltételek szerint kell legyen, vagyis arra, amin az áttekintőn látunk. A *vm* tömb érintett elemeinek értékét is megcseréli: „t”-re, ha a feltételben az egész vármegyét ábrázolja van megadva; „r”-re, ha az van megadva, hogy elég, ha a vármegyének csak egy részét ábrázolja; „e”-re cseréli, ha az a feltétel, hogy a keresési eredményben lévő térképek ne ábrázolják az érintett vármegyéket; és „m”-re ha az érintett vármegyék nem szerepelnek feltételként. Ezzel ugyanúgy felépíti a kereső tömböt, mint manuálisan az egérrel. A tömböt addig lehet építeni, amíg akarjuk, viszont ha nem kapunk eredményt a keresésnél, az azért van, mert egy kizárást okoztunk.

Ha rákattintunk a keresés gombra, akkor beindul a keresés. Lefuttatja a *vmkeres* függvényt (lásd: VI. Kliens oldali függvények című mellékletben). Ez elküldi a szerverre a *vm* tömb összes elemét. Ott az adatbázisban végigfut a *vmtart* táblán, aminek minden mezője egy vármegyére vonatkozik, és minden bejegyzése egy térképműre: a *sorszam* mezőben megtaláljuk a térkép sorszámát, majd a többi mezőben, aminek neve megegyezik a benne érintett vármegye kódjával, egy „t”, „r” vagy „e” értéket, a fentieknek megfelelően. A szervernek átadott tömb elemeinek tartalma alapján a program végrehajt egy keresést a *vmtart* táblán. Kiválasztja az összes olyan elemét, amiben az összes olyan oszlop szerepel, amihez tartozó a *vm* tömbből kapott értékek megegyeznek, kivéve azokat az oszlopokat, amelyeknek „m” az értékük a tömbben. Ha egy elem értéke „r”, akkor abban az esetben is megfelelőnek tartja az adott elemet, ha az annak megfelelő oszlop értéke „t”. Így kapunk egy eredményt, amiben van egy sor sorszám. A sorszámokat

összevetjük a *katalogus\_bov* táblában lévő sorszám mezővel, és az összes megegyező mezőt kiválasztjuk, továbbá megszűrjük még nyelv szerint is, ami a *lang* globális változónak a szervernek átadott értéke alapján történik. Összevetjük a *nyelv* mező értékét a *katalogus\_bov* táblában a *lang* változóból átadott értékkel, és csak azokat az elemeket választjuk ki, amelyekben megegyezik a nyelv a honlap éppen beállított nyelvével. Ezek után a szerver megformázza, majd leküldi a kliensnek a keresési eredményt, amit az megjelenít.

## 11.2 A hagyományos kereső működése

A hagyományos kereső a keresési mezőkben megadott értéket összeveti a keresési mező melletti menüben kiválasztott mezővel az adatbázisban. Abban az esetben is pozitív eredményt hoz ki, ha a mezőben lévő érték az adatbázisban lévő mezőben valahol megtalálható. Ha a mező üres, akkor nem szab feltételt, tehát az összes keresési eredményt kilistázza. Ha megnyitjuk az aktív keresőt, megjelenik néhány újabb mező. Ezekben a mezőkben lévő adatok az előző sor végén lévő menüben



**Feliart** **keres\_inp** **Feliart** **mezo** **Feliart** **lelohely** **gomb**

*Az egyszerű kereső elemei*

megadott és/vagy kapcsolattal kapcsolódnak az előző sorhoz. Ezen túl a lelőhely mezőben megjelölt lelőhely egy és kapcsolattal kapcsolódik az egészhez. Ezt azzal éri el, hogy az sql keresési parancsot a mezők tartalma alapján építi fel.

Ebből a következőképpen épül fel a kereső parancs. A kézzel megírtak esetében a php a megfelelő mezők értékeit helyettesíti be.

```
select * from katalogus_bov where mezo like '%keres_inp%' kapcs1 mezo1 like '%keres_inp1%'  
kapcs2 mezo2 like '%keres_inp2%' kapcs3 mezo3 like '%keres_inp3%' kapcs4 mezo4 like  
'%keres_inp4%' kapcs5 mezo5 like '%keres_inp5%' kapcs6 mezo6 like '%keres_inp6%' kapcs7  
mezo7 like '%keres_inp7%' and lelohely='lelohely';
```

Ezzel eléri a program, hogy ne kelljen a teljes stringet beadni, vagyis ne kelljen hogy egyezzen: mind az eleje mind a vége hiányozhat. A lelőhely viszont minden esetben meg kell egyezzen. Ha az aktív kereső van kapcsolva akkor az *aktivkeres* globális változó értéke „true” lesz, és akkor minden leütésnél végrehajtja a keresést. Ha „false” akkor csak az enter billentyű leütésével vagy a gombra való kattintással hajtja végre a keresést. Mivel az aktív keresőt csak elrejt, ha kikapcsolja a felhasználó, és nem semmisíti meg az objektum tartalmát, a beleírt keresési feltételek akkor is érvényesek lesznek, ha el van rejtve.

## 12. A keresési eredmény

A kereső, a keresési módtól függetlenül létrehozza a keresési eredményt a *searchres* objektumban. Ennek az objektumnak szükség esetén vannak görgetősávjai. Az eredmény egy html táblázat, aminek a fejlécébe az egyes oszlopok megnevezése áll. Alatta a tartalmuk, a *katalogus\_bov* tábla alapján.

Minden sor egy aktív felület. Az *onmouseover* eseményre megváltoztatja a háttérszínét, az *onmouseout* eseményre elveszíti. Az *onclick* eseményre reagál a *clickRow* függvénnyel. A *clickRow* függvénynek a következő paramétereket küldi:

```
clickRow(i,sorszám,képszám,'állománynév','lelőhely','tk');
```

ahol

*i* egy sorszám, annak a sorszáma, hogy hányadik sor a keresési eredményen belül az adott sor. A php a sorokat egy ciklusban hozza létre, így ez az érték a ciklusváltozó. Mivel ez csak a szerver oldalon változó, a kliensen konstansként fog megjelenni:

*sorszám* – a térkép sorszáma a *katalogus\_bov* táblából;

*képszám* – a térkép szelvényeinek a száma a *katalogus\_bov* táblából;

*állománynév* – a raszteres állomány neve, amiben a térkép van szelvényszám és kiterjesztés nélkül;

*lelőhely* – a térképet őrző intézet kódja a *katalogus\_bov* táblából. A *katalogusok* táblából ki lehet keresni a nevét;

*tk* – a térkép, amelyen be van jelölve az az intézet, ahol az eredeti térkép van.

Fontos, hogy – mivel a kliensen konstansként jelentkeznek –, a string típusú változókat idézőjelbe kell tenni.

Az első oszlopban egy rádió gomb van. A neve mindegyiknek ugyanaz. Ezzel éri el a rendszer, hogy ne lehessen egyszerre kettő bejelölve. Az azonosítója mindegyiknek más. Mindegyik azonosító *t*-vel kezdődik, majd utána következik a keresési eredményben lévő sorszáma az elemnek. Így az azonosító *ti*, ami a *clickRow* függvények adott *i* paraméter alapján kikereshető és aktívvá tehető. Ez jelzi, hogy éppen melyik térkép van kiválasztva. A *képszám* paraméterből tudja megállapítani a program, hogy egy- vagy többszelvényes műről van szó. Ha egy szelvény van, akkor kikerül a nagy előnézetbe, és megjelenik az adatok gomb; ha több, akkor jobbra lent megjelennek az egyes szelvények kis előnézetei. Ezek neveit az *állománynév* paraméter alapján keresi ki a rendszer. Mivel a szelvények az intézetkód szerinti mappákban vannak elhelyezve, a mappa nevét a *lelőhely* paraméterből tudja beazonosítani. Ha egy térkép nagy előnézetére rákattint a felhasználó, akkor feljön egy popup ablak egy térképböngészővel. Ennek az ablaknak a tartalmát a *mapper.php* generálja le. A térkép böngészése a *shifzoom* program segítségével történik.



Ez először a kép mérete alapján kiszámítja a az oldalak arányait. Majd ezek után a megadott maximum szélességhez kiszámítja a megfelelő arányokkal a magasságot. Ha ez nagyobb mint a maximum magasság, akkor a maximum magasság alapján számolja ki az arálynak megfelelő szélességet. Az így kapott paraméterek alapján létrehoz egy `<DIV>` objektumot, és azon belül egy `<IMG>` objektumot, aminek ugyanakkorák a magasság és szélesség paraméterei. Fontos, hogy ennek az arányai megegyezzenek a kép arányaival, különben torzítani fogja a térképet. Ebbe a képbe egy nagy felbontású képet kell helyezni, mivel a program maximum nagyítása a kép maximális felbontása. A képet a `shiftzoom` osztályba helyezzük. az `onload` eseménykezelő a következő parancsot kapja:

```
shiftzoom.add(this,{overview:true, zoom:100, buttons:false});
```

Ezek az adatok adják meg, hogy erre az objektumra aktiválja a `shiftzoom` programot, az áttekintő legyen bekapcsolva, indításkor a nagyítás legyen 100%, és ne legyenek rajta gombok. Bár a paraméterek közül utólag bármelyiket meg lehet cserélni, ebben az esetben csak az `overview` paramétert cseréli a program. A zoom paraméter az egér görgetőjének forgatásával változik. A `shiftzoom` program fejlesztői dokumentációja elérhető a <http://www.netzgesta.de/shiftzoom/> cím alatt. Így tehát a kép definiálása a következő sorral történik:

```
<IMG id="mapzoom" src="nkep.php?filename=tknev&int=lelohely" class="shiftzoom"
onload="shiftzoom.add(this,{overview:true, zoom:100,buttons:false});" width="w" height="h"
border="0"/>
```

A fenti paranccsal megjelenik a térkép a böngészőben. Ahhoz, hogy ez rendesen működjön az `index.html` fejlécében ott kell legyen a következő sor:

```
<SCRIPT type="text/javascript" language="javascript" src="shiftzoom.js"></SCRIPT>
```

Továbbá a `shiftzoom.js` abban a könyvtárba kell legyen, ahol a honlap van a webserveren. A `shiftzoom` a fenti címről letölthető.

A kép feletti pipa a kattintásra felhívja a `prev` függvényt, és átadja az állapotát. Előbb a függvény a `shiftzoom` függvényt rúgja a térképre. Utána attól függően, hogy a pipa milyen állapotban van, attól függ az, hogy a függvény mit csinál. Ha a pipa ki van pipálva, akkor az `overview` paramétert `true`-ra állítja, ha nincs akkor `false`-ra. Ezt a következő parancsokkal teszi:

```
shiftzoom.add(document.getElementById('mapzoom'));
if(typeof $=='undefined') {function $(v) {return(document.getElementById(v));}}
if(ck==true) shiftzoom.set($('mapzoom'),'overview',true); else
shiftzoom.set($('mapzoom'),'overview',false);
```

Az első két sor definiálja, hogy melyik objektummal dolgozik a `shiftzoom` program. Utána egy elágazás van, ami eldönti, hogy a pipa ki van pipálva vagy sem. A `shiftzoom.set` a program attribútumain állít. Ha ki van pipálva, akkor az `overview` attribútumot igazra (`true`) állítja, ha nincs

akkor hamisra (*false*).

Ezt az eljárást az adatok ablakban és a vármegyés nézegetőben is ugyanígy alkalmaztam. A nagy térképböngészőben van ezen kívül még egy funkció, amivel ki lehet emelni a térképböngészőt egy új lapra. Mivel a *mapper.php* szkriptnek meg van adva a *winw* és *winh* paraméter is, megjelenik egy piktogram a kép felett a pipától balra, amire ha rákattint a felhasználó, megjelenik a *mapperkint.html* egy új lapon. A *get* módszerrel átadja a formban a következő paramétereket az oldalnak: *maxh*, *maxw* (*winh* és *winw* értéke), *int*, *lang*, *sorszam* és *kepszam*.

Ekkor megjelenik egy új lapon egy oldal, aminek a bal oldalán három ikon található. Fent egy lap, ami fölé ha viszem az egeret, akkor a *detres* a megfelelő *sorszam*, *int* és *lang* értékeinek átadásával bekerül a popup objektumba, ami láthatóvá válik. Ha az egér elhagyja az ikont, akkor az objektum ismét láthatatlan lesz. Alatta egy üres kép található, amibe az oldal betöltésekor az *int* paraméter alapján betölti a program az intézet képét. Ha a felhasználó e fölé viszi az egeret, akkor a kép megjelenik nagyban a popup objektumban, ami a fentihez hasonlóan látható lesz, és kitakarja a térképet. Ha az egérmutató elhagyja a képet, a popup objektum ismét eltűnik. Alatta egy 1914-es történelmi Magyarország közigazgatási térkép van. Ennek a tematikája az ábrázolt vármegyék alapján van elkészítve (lásd: 10. Tematikus térképek készítése című fejezetet). Ha e fölé viszi a felhasználó az egeret, akkor a *map\_1914* objektum jelenik meg, és kitakarja a térképböngészőt. Ebben egy ugyanolyan térkép van, mint az ikonban, csak nagyobb méretben. A tematika az oldal betöltésekor kerül rá.

Az oldal jobb oldalán egy üres objektum található (*mappercont*). Ide betöltéskor a *mapper.php* tartalma kerül, és megjelenik a fentiekhez hasonlóan ugyanazon paraméterek átadásával a térképböngésző.

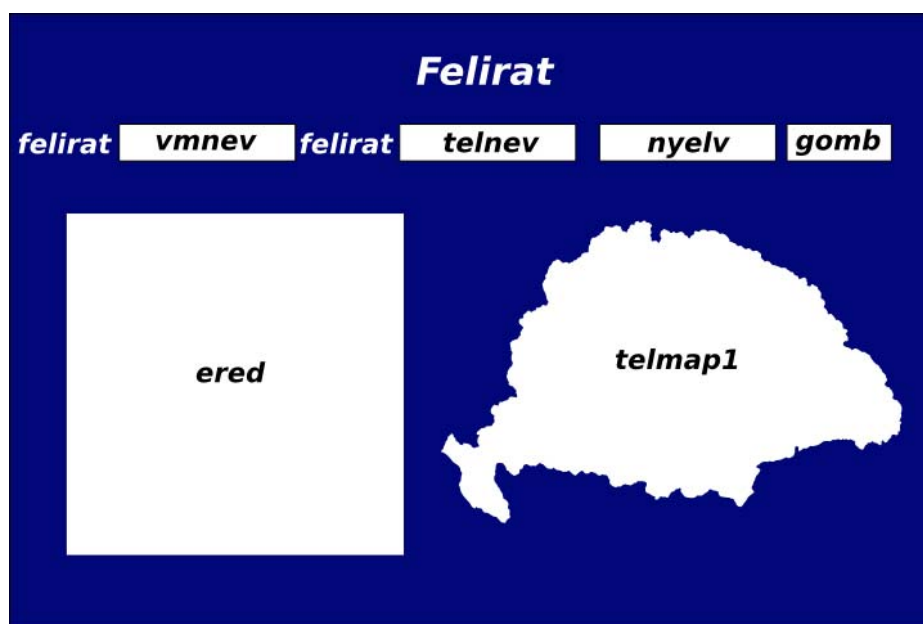
Ha a felhasználó az adatok gombra kattint, akkor megjelenik a popup ablak, és benne az adatok objektum. Ennek bal oldalán fent van egy üres objektum, amibe az ablak megnyitásakor a *detres.php* eredménye kerül. A jobb oldalon van egy üres objektum, amibe az intézet képe kerül. Ugyanitt van két rejtett objektum, amibe a Magyarország, illetve a Baden-Württemberg közigazgatási térképei kerülnek. Ezek közül a térképek közül azon, amelyen az intézet található az intézet helyének poligonját vörössel (990000) tölti ki. Szintén rejtve van egy 1914-es Magyarország közigazgatási térkép, amelyen az ablak nyitásakor megjelenik a vizsgált térképen ábrázolt, és részben ábrázolt vármegyék tematikája. Végül van egy rejtett objektum, amibe a *mapper.php* tartalma kerül.

Az ablak alján a bal oldalon van négy darab objektum. Az első az intézet képe kicsiben. Ha e fölé viszi a felhasználó az egeret, akkor a program megjeleníti jobbra fent az intézet képét, a többi objektumot elrejt. Mellette van egy Magyarország és egy Baden-Württemberg térkép, amin az előbbihez hasonlóan – azon a térképen, amelyen található –, be van jelölve az intézet helye. A másik

térkép rejtve van. Ha fölé viszi a felhasználó az egeret, akkor megjelenik nagyban fent, míg a többi ott lévő objektumot elrejtí. Mellette van egy kicsi 1914-es történelmi Magyarország közigazgatási térképe, amely fölé ha felkerül az egérmutató, akkor megjelenik a jobb oldalon az azonos térképet tartalmazó objektum, és elrejtí a többit. Ha az utolsó ikon fölé kerül az egérmutató – ami a vizsgált térkép kicsi előnézete –, akkor a térképböngészőt tartalmazó objektum jelenik meg a jobb oldalon, és a többi lesz elrejtve.

### 13. Vármegye adatlapja

Ha a felhasználó rákattint a vármegyék adataira, feljön a popup ablak, és a *telszocont* objektum láthatóvá válik. A *telszo.php* eredménye kerül bele, miután megadtuk a *lang* paraméterben a *lang* globális változó értékét. A következő ábra mutatja a létrehozott objektumokat:



A *telmap1* minden objektumához hozzá van rendelve három eseménykezelő. Mindegyik átadja a függvénynek a vármegye kódját.

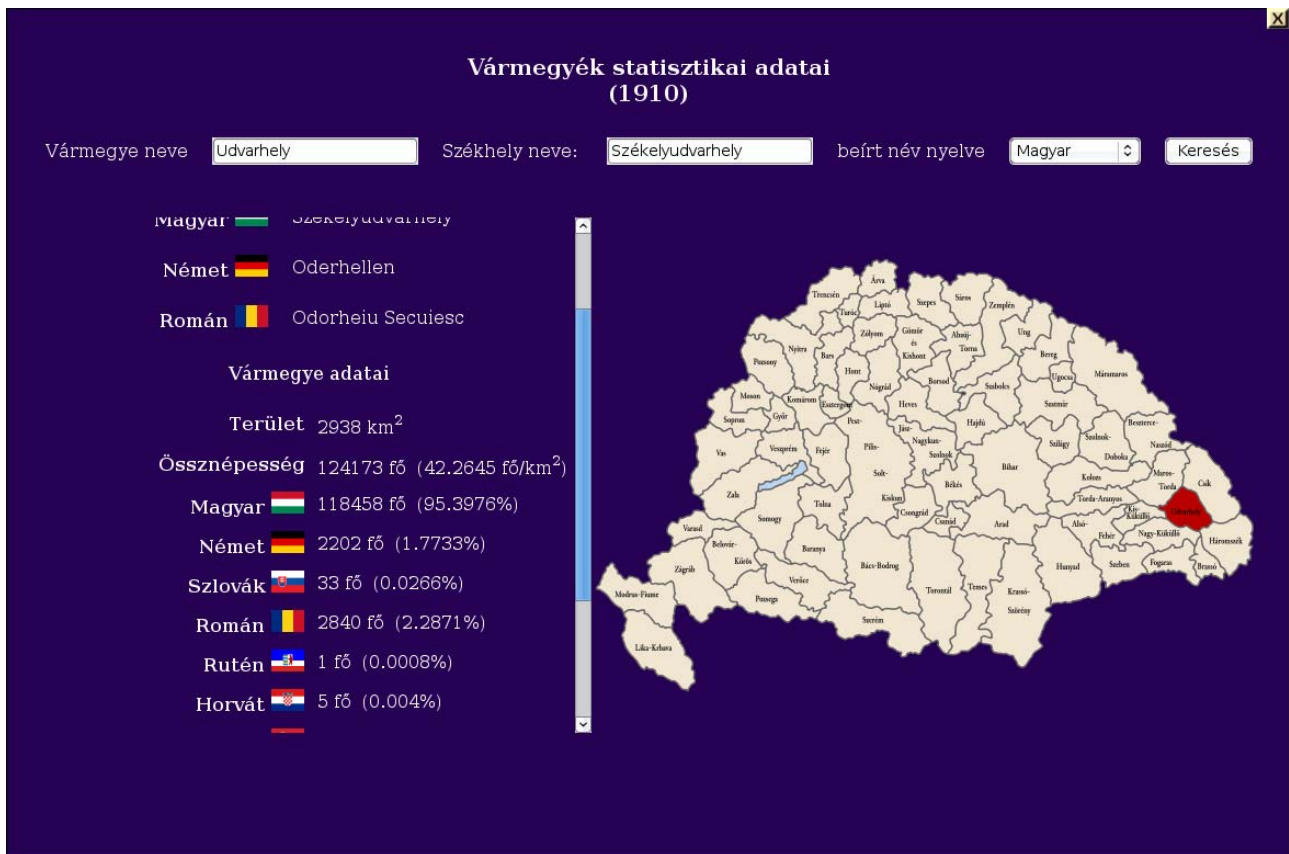
Az *onmouseover* eseménykezelő a *mouseOverVMT* függvényt futtatja le a vármegye kódjával mint paraméterrel, ami megvizsgálja, hogy a vármegye kódja megegyezik-e a kiválasztott vármegye kódjával (ez utóbbi a *selectedvm* globális változóban van). Ha egyezik, akkor egy sötétebb vörösre cseréli a vármegye kitöltését (8A0000), ha pedig nem, akkor barnára cseréli (FEBC5). Az *onmouseout* esemény pedig a *mouseOutVMT* függvényt futtatja úgyszintén a vármegye kódjával. Ez is az előzőhöz hasonlóan megvizsgálja, hogy a vármegye kódja megegyezik-e a *seletedvm* értékével. Ha megegyezik, akkor világos vörösre cseréli a vármegye

kitöltését (BB0000), ha viszont nem egyezik, akkor pedig világosbarnára (F0E6D2).

Az *onclick* eseménykezelő a *mouseClickVMT* függvényt futtatja le úgyszintén a vármegye kódjával. A függvény a vármegye poligonjának a kitöltését megcseréli vörösre (8A0000). Abban az esetben, ha a *selectedvm* globális változónak van érték adva, a benne lévő vármegye kódjával egyező vármegye kitöltését világosbarnára (F0E6D2) cseréli, így eléri, hogy ne legyen két vármegye kiválasztva a térképen. Az *szhely.php* szkriptnek elküldi a *vmkod* paraméterbe a vármegye kódját, és visszakapja a székhelyének a nevét. A *telnev* objektumba behelyezi a vármegye székhely magyar nevét. A *nyelv* objektumba kiválasztja a magyar nyelvet (*opthu*). A *selectedvm* globális változó értéke az eseményhez kapcsolódó vármegye kódja lesz. Utána lefuttatja a *keres* függvényt paraméterek nélkül.

Ha a felhasználó a *vmnev* mező tartalmát megváltoztatja, akkor a *selectedvm* globális változóban lévő vármegyekódnak megfelelő vármegye kitöltését világosbarnára cseréli, és a *telnev* objektum tartalmát törli. Ugyanígy, ha a *telnev* tartalmát változtatja meg, a kiválasztott vármegye kitöltését világosbarnára cseréli, és a *vmnev* tartalmát törli. Mindkét esetben a *selectedvm* tartalmát törli. Ha a felhasználó rákattint a gombra, és a *vmnev* tartalma van kitöltve, a *vmkeres* függvényt futtatja le. A függvény a *vm.php* szkriptnek átadja a *vmnev* paraméterbe a *vmnev* objektum tartalmát. Ha ez valóban egy vármegye neve, a szkript visszaküld egy xml állományt, aminek a *vmszekhely* attribútumába escape karakterekkel benne lesz a vármegye székhelye. Miután eltávolította az escape karaktereket az eredményt a *telnev* objektumba helyezi, és a *nyelv* objektumba kiválasztja a magyar nyelvet (*opthu*). A *selecetedvm* változó értéke a keresett vármegye kódja lesz, majd lefuttatja a *keres* függvényt. Ha a *telnev* mező van kitöltve, akkor a *nyelv* mezőnek értékével azonos nevű mezőt a *telszotar* táblából egyezteteti a *telnev* mező értékével, és a *jel* mezőből kiolvassa a vármegye kódját. Az eddig kiválasztott vármegye poligonjának a kitöltését világosbarnára cseréli, az újonnan kiválasztott vármegyének a kitöltését pedig vörösre. A *telnevq.php* eredményét az *ered* objektumba helyezi. Ekkor létrejön egy rejtett objektum *vmkods* azonosítóval, ami annak a vármegyének a kódját tartalmazza, aminek az adatait éppen az *ered* objektumba helyezte a program. Ez alapján lehet utólag kijelölni a vármegye poligonját.

A *telnevq.php* a beérkező adatok alapján kiírja a vármegye nevét, a székhely nevét az összes nyelven amin benne van a *telszotar* táblában. Alatta kiírja a vármegye területét, népességét, népsűrűségét és nemzetiségekre bontott népességét, mind lélekszámban, mind százalékos arányban. Az alján megjelenik egy kis változata a vármegye térképének, amire ha rákattint a felhasználó elrejtja a program a *telmap1* objektumot, és helyette az *oldmap* objektum jelenik meg, amelyben megjelenik a vármegye térképe egy térképböngészőben. Az alatta lévő gombra történő kattintással elrejtethetjük az *oldmap* objektumot, és ismét megjelenik a *telmap* objektum.



*Udvarhely vármegye információs lapja*

## 14. Összefoglalás

Munkám célja minél több magyar vonatkozású néprajzi térkép digitális felvételének összegyűjtése és adatbázisba rendezése, valamint az ennek megtekintéséhez szükséges felhasználói kezelői felület létrehozása volt.

Számos térképet sikerült összegyűjteni. Kutatásaim során jól átláttam a németországi levéltárakban lévő térképanyagot, amelyből kitűnt, hogy rengeteg térkép van ezekben a levéltárakban, és érdemes ezeket tovább kutatni.

Az eredményt egy honlapon tettem közzé, amelyhez több keresési módszert is kifejlesztettem. Jelenleg a honlap (2011. május) a következő címen érhető el:

<http://demogmap.elte.hu>

Érdeemesnek tartom e témában a továbbkutatást és a honlap továbbfejlesztését.

## 15. Irodalomjegyzék

### **A Magyar Szent Korona Országainak 1910. évi népszámlálása**

Első rész. A népesség főbb adatai községek és népesebb puszták, telepek szerint

Magyar statisztikai közlemények, Új sorozat, 42. kötet

*Magyar Királyi Központi Statisztikai Hivatal, Budapest, 1912*

### **Gönczy Pál-féle vármegyetérkép sorozat**

Rajzolta Kogutowicz Manó

*Posner Károly Lajos térképészeti intézet, Budapest, 1880-as évek*

### **Hajdú-Moharos József: Magyar Településtár**

*Kárpát-Pannon Kiadó, Budapest, 2000*

### **Internetes források:**

#### **SVG példák az interneten:**

<http://www.carto.net/papers/svg/samples/>

#### **PHP dokumentáció**

<http://www.php.net/manual/en/>

#### **W3Schools, internetes webfejlesztő tanfolyam**

<http://www.w3schools.com/>

#### **A Shiftzoom program és annak fejlesztői dokumentációja**

<http://www.netzgesta.de/shiftzoom/>

#### **MySQL 5.0 fejlesztői dokumentációja**

<http://dev.mysql.com/doc/refman/5.0/en/>

#### **Magyarország közigazgatási térképe**

[http://upload.wikimedia.org/wikipedia/commons/a/ac/Hungary\\_location\\_map.svg](http://upload.wikimedia.org/wikipedia/commons/a/ac/Hungary_location_map.svg)

#### **Magyarország történelmi 1914-es közigazgatási térképe**

[http://upload.wikimedia.org/wikipedia/commons/9/95/Kingdom\\_of\\_Hungary\\_counties.svg](http://upload.wikimedia.org/wikipedia/commons/9/95/Kingdom_of_Hungary_counties.svg)

## **Európa térképe**

[http://commons.wikimedia.org/wiki/File:Blank\\_map\\_europe.svg](http://commons.wikimedia.org/wiki/File:Blank_map_europe.svg)

## **Németország közigazgatási térképe**

[http://upload.wikimedia.org/wikipedia/commons/2/2c/Karte\\_Bundesrepublik\\_Deutschland.svg](http://upload.wikimedia.org/wikipedia/commons/2/2c/Karte_Bundesrepublik_Deutschland.svg)

## **Baden-Württemberg közigazgatási térképe**

[http://upload.wikimedia.org/wikipedia/commons/a/ad/Landkreise\\_Baden-Wuerttemberg.svg](http://upload.wikimedia.org/wikipedia/commons/a/ad/Landkreise_Baden-Wuerttemberg.svg)

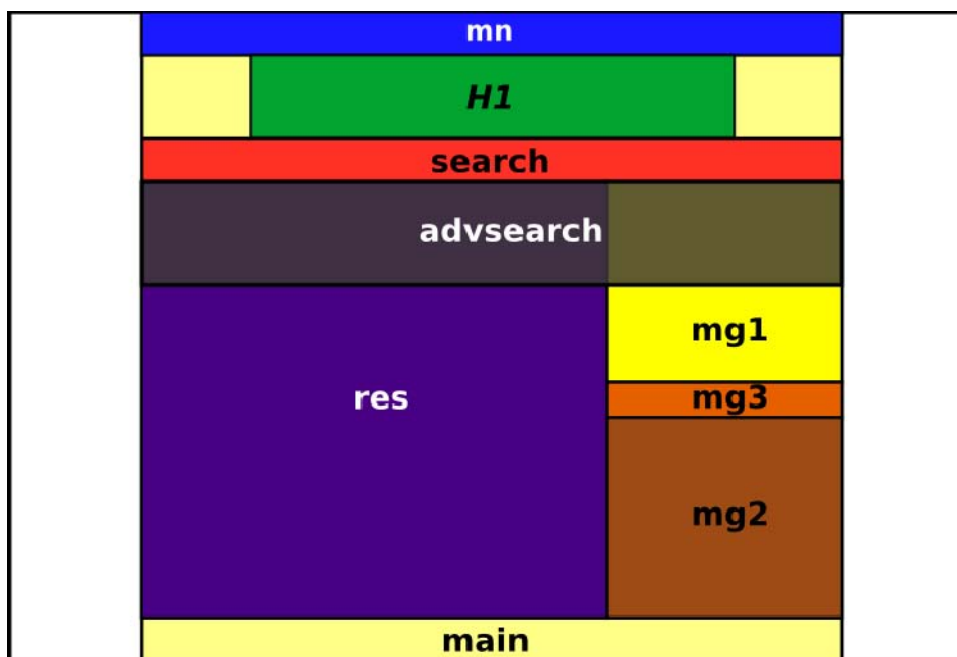
## **Honlapon felhasznált országzászlók**

<http://hu.wikipedia.org>, az adott országra vonatkozó lapjáról

## **16. Köszönetnyilvánítás**

Ezúton mondok köszönetet mindazoknak, akik segítettek ezen diplomamunka létrejöttében. Szeretném megköszönni a Hadtörténeti Intézet Térképtára, a Babeş–Bolyai Egyetem Földrajz Kara, a Tübingeni Egyetem Földrajzi Intézete, a Heidelbergi Egyetem Földrajzi Intézete dolgozóinak a segítséget a térképekhez való hozzájárásban. Külön szeretném megköszönni a Karlsruhei Főiskolának a lehetőséget, hogy három félévig ott kutathattam, ezen belül Heinz Musall professzor urnak, hogy vállalta a külső témavezetést. Szeretném még megköszönni a Térképtudományi és Geoinformatikai Tanszék minden dolgozójának a segítséget a munka során, és végül de nem utolsósorban a két témavezetőmnek Márton Mátyásnak és Elek Istvánnak a téma vezetését.

# I. Objektumok elhelyezkedése a honlapon



A fent látható ábrán lévő objektumok mind a *main* objektumba vannak beágyazva. A *main* a *body*-ba van beágyazva, és annak közepén található.

*mn*: A menüsor ikonjai vannak benne.

*H1*: Üres. A honlap címe kerül bele betöltéskor.

*search*: Üres, a kereső elemei kerülnek bele betöltéskor.

*advsearch*: Üres, a bővített kereső elemei kerülnek bele betöltéskor. Láthatatlan, a menüsorban lévő parancsikonnal jeleníthető meg. Ha megjelenik, kitakarja az alatta lévő objektumokat.

*res*: A *res* objektumba beágyazott objektumok a *res* objektumot teljesen kitöltik, és csak kivételes esetben látszik több mint kettő közülük egyszerre. Némelyik betöltésekor aktiválja a program a görgetősávokat, így ha nagyobb az objektum a *res* objektumnál, akkor a görgetősávokkal lehet mozgatni a látható részeket. A következő objektumok vannak beágyazva a *res* objektumba:

- *searchres*: Egy üres objektum, amibe a keresőtől kapott eredmények kerülnek elhelyezésre.
- *maptitle*: Egy üres objektum, amibe a megjelenő térkép címe kerül. Ez az objektum megjelenik, ha térképek is megjelennek.
- *bw*: Egy táblázatot tartalmaz, aminek egy sora és két oszlopa van. Az első oszlopban két ikon van, a *gericon.png* és a *fgomb\_n.png*. A második oszlopban a *bw.svg* van, aminek azonosítója *map1*.
- Ha a *bw* objektum látható, akkor a *maptitle* is látható.
- *cont*: Ebben az objektumban mindössze egy elem van: a *europe.svg*, aminek azonosítója *eu*. Ha a *cont* objektum látható, akkor a *maptitle* is az.



- *germany*: Egy táblázat található benne, aminek egy sora és két oszlopa van. Az első oszlopban az *euicon.png* és a *fgomb\_n.png* ikonok találhatók. A második oszlopban a *germany.svg* található, aminek azonosítója *de*. Ezzel az objektummal együtt a *maptitle* is mindig látható.
- *hungary*: Egy táblázat található benne, aminek egy sora és két oszlopa van. Az első oszlopban az *euicon.png* és a *fgomb\_n.png* ikonok találhatók. A második oszlopban a *hungary.svg* található, aminek azonosítója *hu*. Ezzel az objektummal együtt a *maptitle* is mindig látható.
- *vmkereso*: Fent a *hungary\_1914.svg* található, aminek azonosítója *mo1914*. Alatta egy táblázat található, aminek két sora és három oszlopa van. Az első sor minden oszlopában egy ikon található, míg a második sor minden oszlopában egy gomb. Az első sor ikonjai balról jobbra a következők: *keknm.png*, *nmk.png* és *nagyitonmk.png*. A *vmkereso* objektummal együtt nem látható a *maptitle*.

*mg1*: Üres objektum, amibe az előnézeti képek kerülhetnek.

*mg2*: Üres objektum, amibe vagy a szelvényezett térképművek egyes szelvényei kerülnek, vagy bizonyos kiegészítő információk vagy formok kerülhetnek.

*mg3*: Üres objektum, amibe szükség esetén az „Adatok” gomb megjelenik.

A *body* objektumban még a *main* objektumon kívül két objektum található.

*tooltip*: Egy üres objektum, amibe szükség esetén belekerül egy tooltip szövege. Alaphelyzetbe a *main* objektum mögött helyezkedik el, ha viszont egy tooltip megjelenik akkor a *main* objektum felé helyezkedik.

*loading*: Egy kép van benne, *animload.gif*. Betöltéskor kitakarja a *main* objektumot. Miután az *init* függvény betöltéskor sikeresen lefutott, a *main* objektum mögé kerül, és csak a honlap újratöltésekor lesz ismét látható.

## **1.1 A popup ablak objektumai**

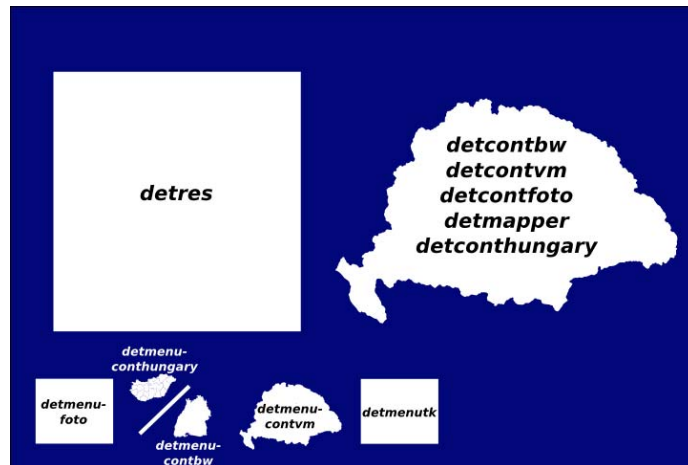
- A popup ablakban mindig mindössze egy objektum látható. Minden objektum a popup ablak egy adott funkcióját látja el. A *mapper*, *mapperclos*e és *mapviewer* mindig láthatók, ha a popup ablak meg van nyitva.
- A *mapper* kitakarja a honlap egyéb tartalmát, hogy az esetleg látszó aktív elemek se legyenek aktívak amíg a popup ablak nyitva van. A *mapperclos*e a bezárógombot helyezi az ablak jobb felső sarkába. Ha erre kattint a felhasználó, a program elrejtja a popup ablakot, és minden benne lévő objektumot. A *mapviewer* objektum tartalmazza a többi, a popup

ablakban megjelenő objektumot. Ezeket alább felsorolom:

- *forras*: Üres objektum, amibe a betöltéskor a honlapon felhasznált adatok forrásai kerülnek.
- *szdata*: Üres objektum, amibe a betöltéskor a honlap szerkesztőjének neve és a honlap szerkesztésében közreműködő személyek nevei kerülnek.
- *intinfo*: Üres objektum, amibe a betöltéskor az *intinfo.php* eredménye kerül.
- *adatok*: Az adatok ablak objektuma. A következő objektumokat tartalmazza:
  - *detres*: Üres objektum, amibe a *detres.php* eredménye kerül.
  - *detconhungary*: A *dethungary* objektumot tartalmazza, ami egy közigazgatási térkép Magyarországról (*hungary.svg*).
  - *detcontbw*: A *detbw* objektumot tartalmazza, ami Baden-Württemberg közigazgatási térképe (*bw.svg*).
  - *detconfoto*: Egy üres objektum, amibe az intézet fotója kerül.
  - *detcontvm*: A következő két objektumot tartalmazza:
    - *detcontvmcim*: Egy üres objektum, amibe a vármegyetérkép címe kerül.
    - *detvm*: Magyarország történelmi közigazgatási térképe az 1914-es állapot szerint (*hungary\_1914\_noleg.svg*).
  - *detmapper*: Üres objektum, amibe a *mapper.php* eredménye kerül.
  - *detmenu*: A következő négy objektumot tartalmazza:
    - *detmenufoto*: Az intézet képe. Ha fölé viszi a felhasználó az egeret, akkor láthatóvá válik a *detconfoto* objektum, és elrejtí a *detcontvm*, *detcontbw*, *detconhungary* és a *detmapper* objektumokat.
    - *detmenuconhungary*: A *detmenuhunagry* objektumot tartalmazza, ami Magyarország közigazgatási térképe. Ha fölé viszi a felhasználó az egeret, akkor megjeleníti a *detconhungary* objektumot, és elrejtí a *detcontvm*, *detcontbw*, *detconfoto* és *detmapper* objektumokat.
    - *detmenucontbw*: A *detmenubw* objektumot tartalmazza, ami Baden-Württemberg közigazgatási térképe. Ha fölé viszi a felhasználó az egeret, akkor megjeleníti a *detcontbw* objektumot, és elrejtí a *detcontvm*, *detconhungary*, *detconfoto* és *detmapper* objektumokat.
    - *detmenucontvm*: A *detmenuvm* objektumot tartalmazza, ami Magyarország történelmi közigazgatási térképe, mely az 1914-es állapotot mutatja be. Ha fölé viszi a felhasználó az egeret, akkor megjeleníti a *detcontvm* objektumot, és elrejtí a *detcontbw*, *detconhungary*, *detconfoto* és *detmapper* objektumokat.
    - *detmenutk*: Egy üres objektum, amibe a térkép kis előnézete kerül. Ha fölé viszi a felhasználó az egeret, akkor megjeleníti a *detmapper* objektumot, és elrejtí a

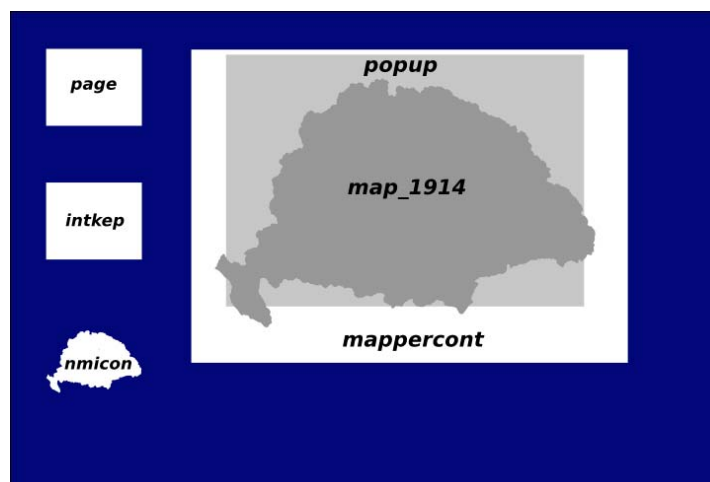
*detcontbw*, *detconhungary*, *detcontfoto* és *detcontvm* objektumokat.

- *telszocont*: Üres objektum, amibe a vármegye információ kereső kerül (*telszo.php* eredménye).
- *nepterkep*: Üres objektum, amibe a vármegyék népességét ábrázoló térkép kerül (*nepterkep.php* eredménye).
- *bigmap*: Üres objektum, amibe a térképböngésző kerül (*mapper.php* eredménye).
- *statterkep*: A térképet ábrázoló vármegyék számát bemutató térképet és a hozzá tartozó elemeket tartalmazza. Ezek a következők:
  - *statcim*: Üres objektum, amibe a térkép címe kerül.
  - *statmapcontt*: A *statmap1* objektumot tartalmazza, ami egy történelmi közigazgatási térkép Magyarországról, mely az 1914-es állapotot mutatja be (*hungary\_1914\_noleg.svg*).
  - *statmapcontr*: A *statmap2* objektumot tartalmazza, ami egy történelmi közigazgatási térkép Magyarországról, mely az 1914-es állapotot mutatja be (*hungary\_1914\_noleg.svg*) (rejtve).
  - *Statmapconte*: A *statmap3* objektumot tartalmazza, ami egy történelmi közigazgatási térkép Magyarországról, mely az 1914-es állapotot mutatja be (*hungary\_1914\_noleg.svg*) (rejtve).
  - Alatta egy táblázat található, aminek egy sora és három oszlopa van. Az oszlopok tartalma jobbról balra:
    - *statmap4*: Magyarország történelmi közigazgatási térképe, mely az 1914-es állapotot mutatja be. Ha az egeret fölé viszi a felhasználó, akkor megjeleníti a *statmap1*-et, és elrejt a *statmap2*-t és a *statmap3*-at.
    - *statmap5*: Magyarország történelmi közigazgatási térképe, mely az 1914-es állapotot mutatja be. Ha az egeret fölé viszi a felhasználó, akkor megjeleníti a *statmap2*-t, és elrejt a *statmap1*-et és a *statmap3*-at.
    - *statmap6*: Magyarország történelmi közigazgatási térképe ami az 1914-es állapotot mutatja be. Ha az egeret felé viszi a felhasználó akkor megjeleníti a *statmap3*-at, és elrejt a *statmap1*-et és a *statmap2*-t.
  - *statintsel*: Egy üres objektum, amibe *statmenu.php* eredménye kerül.



Az adatok objektumban elhelyezkedő objektumok

## II. Külön lapon megjelenő térképböngésző elemei



mapperkint.html oldalon elhelyezkedő objektumok

*page*: *icon/page.png* kép van benne. Ha fölé húzza a felhasználó az egeret, akkor az *initpopup('det',event)*; függvényt futtatja. Ekkor megjelenik a *popup* objektumban a térkép adatlapja, amit a *detres.php* szkript generál. Ez kitakarja a térképböngészőt. Ha a felhasználó eltávolítja az objektum fölül az egérmutatót, akkor a *popupdest('det')*; függvényt futtatja le, ami elrejt a popup ablakot, így ismét látszik a térképböngésző.

*intkep*: Egy üres, 100 px magas kép, amibe a honlap betöltésekor belekerül az intézet képe. Ha fölé viszi a felhasználó az egeret lefuttatja az *intpopup('kep',event)* függvényt, ami megjeleníti nagyban az intézet képét, és kitakarja vele a térképböngészőt. Ha a felhasználó eltávolítja az objektum fölül az egérmutatót, akkor a *popupdest('kep')*; függvényt futtatja le, ami elrejt a popup ablakot, így ismét látszik a térképböngésző.

*nmicon*: Magyarország történelmi közigazgatási térképe, amire a honlap betöltésekor elkészül az a tematika, amely bemutatja, hogy a térképböngészőben lévő térkép mely vármegyét ábrázolja, és

melyeket nem. Ugyanez rákerül a *map\_1914* objektumban lévő azonos térképre, csak nagyobb is. Ha a felhasználó fölé viszi az egeret, akkor lefut a *initpopup('tk')* függvény, amely megjeleníti a *map\_1914* objektumot. Ekkor megjelenik a térkép nagyban és kitakarja a térképböngészőt. Ha a felhasználó eltávolítja az egeret, akkor lefut a *popupdest('tk');* függvény, ami elrejtí a *map\_1914* objektumot, és ismét a térképböngészőt lehet látni.

*popup*: Egy üres objektum, amibe bekerül a felugró ablak tartalma.

*map\_1914*: Tartalmazza az *nm* objektumot, ami egy 1914-es állapotot bemutató történelmi közigazgatási térkép Magyarországról. Az oldal betöltésekor elkészül rá az a tematika, amely bemutatja, hogy a térképböngészőben lévő térkép mely vármegyét ábrázolja.

*mappercont*: Egy üres objektum, amibe a térképböngésző kerül.

### III. Az adatbázis felépítése

**Adatbázis neve:** demogmap

**Szerver:** MySQL 5.0

**Protokoll verzió:** 10

**Operációs rendszer:** Linux

**Karakter kódolás:** utf8

**Táblák száma:**17

#### III.1 Az adatbázist felépítő táblák

**auth:** A jelszóval védett részekre tartalmazza a felhasználókkal kapcsolatos információkat.

**euaktív:** Az Európa térképhez tartalmazza az aktív poligonokra vonatkozó információkat.

**germany:** A Németország térképhez tartalmazza az aktív poligonokra vonatkozó információkat.

**katalogusok:** Az egyes levéltárakra vonatkozó információkat tartalmazza három nyelven.

**katalogus\_bov:** Az összes néprajzi térképre vonatkozó információt tartalmazza három nyelven, ami a honlapon megtekinthető.

**mularch:** Az egyes levéltárak helyére vonatkozó információkat tartalmaz.

**nepesseg:** Az 1910-es népszámlálás vármegyékre vonatkozó adatait tartalmazza.

**nyelvek:** Az összes, a honlapon lévő, más táblába nem tartozó megírást tartalmazza három nyelven.

**országok:** Az Európa térképhez tartozó névrajzi elemeket tartalmazza.

**telszotar:** A vármegyeszékhelyek neveit tartalmazza különböző nyelveken, Hajdú-Moharos József Településtára alapján.

**terkeres:** A levéltárakat közvetlenül elérhetővé tevő térképek aktív poligonjaira tartalmaz információkat.

**tooltip:** A tooltipek tartalmát tartalmazza három nyelven.

**vm:** A vármegyék neveit tartalmazza.

**vmtart:** Az összes néprajzi térképre vonatkozó leírás, mely leírja, hogy mely vármegyét ábrázolja a térkép.

**forras:** A honlapon használt adatok forrásai.

**szadatok:** A szerző adatai és a honlap készítésében közreműködők adatai.

**statszin:** A statisztikai térképek jelkulcsa.

### **III.2 A táblákban lévő mezők**

#### **auth:**

**username:** A felhasználónév.

típus: text

**password:** A jelszó MD5-ös kódolással.

típus: text

**tiltva:** Ha 1, akkor a felhasználó le van tiltva. Ha 0, akkor bejelentkezhet.

típus: bool

**felt:** Későbbi fejlesztéshez. Ha a felhasználó tölthet fel térképet, akkor 1, amúgy 0.

típus: bool

#### **euaktív:**

**ország:** Aaz ország kétbetűs kódja, amelyre vonatkozik a bejegyzés - csupa nagybetűvel.

típus: text

**polygon:** A poligon neve, amit aktívvá kell tenni, vagyis az országot ábrázoló poligon. Az Európa térképen legtöbb esetben az ország kétbetűs kódja - csupa kisbetűvel.

típus: text

**terkepnev:** Az ország térképének állományneve a kiterjesztés nélkül.

típus: text

## germany:

**bland:** Az aktív szövetségi állam nevére utaló rövidítés.

típus: text

**polygon:** Az aktív szövetségi állam poligonjának a neve a Németország térképen.

típus: text

**tk:** Az aktív poligonra való kattintással felhívásra kerülő térkép állományneve a kiterjesztés nélkül.

## katalogusok:

**nyelv:** A bejegyzésre vonatkozó nyelv. Három értéke lehet: HU - magyar, DE - német, EN – angol.

típus: text

**katalogus:** Az intézetre vonatkozó rövidítés. Ez jelenik meg a többi táblában, ha az intézetre van utalás.

típus: text

**Int\_nev:** Az intézet hivatalos neve. A mellékjeles és egyéb különleges karakterek entitykkel vannak benne.

típus: text

**Int\_cím:** Az intézet postai címe: a mellékjeles és egyéb különleges karakterek entitykkel vannak benne.

típus: text

**Int\_honlap:** Az intézet honlapjának internetcíme.

típus: text

**Int\_rov:** Az intézet neve röviden.

típus: text

**foto:** 1, ha van fotó az intézetről; 0, ha nincs.

típus: bool

**tk:** A térkép neve, amely az adott intézethez kapcsolódik.

típus: text

## katalogus\_bov:

**sorszam:** A térképnek egy, az adatbázison belüli egyedi azonosítója. Nem lehet elsődleges kulcs, mert minden sorszám a táblában háromszor jelenik meg, mivel minden térképre három nyelven van bejegyezve, és minden nyelvre egy bejegyzés van.

típus: int

**nyelv:** A bejegyzés nyelvére vonatkozó adat. Három értékre lehet: HU: magyar, DE: német, EN: angol

típus: text

**Katalogus:** Az intézeten belül használt katalogus, amiben a térképre vonatkozó bejegyzés van. Nem kötelező kitölteni.

típus: text

**Jelzet:** Az intézeten belül használt jelzet.

típus text

**Lelohely:** az intézet azonosítója, ahol a térkép eredeti példánya van. Megegyezik a *katalogusok* tábla *katalogus* mezőjének megfelelő értékével.

típus: text

**filenev:** Az állomány neve a kiterjesztés nélkül. A többlapos térképműveknél a lapszámmra vonatkozó részt sem tartalmazza.

típus: text

**kepszam:** A jelzetben lévő képek száma.

típus: text

**ext:** A az állomány kiterjesztése.

típus: text

**prevezt:** Az előnézet állomány kiterjesztése.

típus: text

**leiras:** A térkép rövid leírása.

típus: text

**szerzo:** A térkép készítőjének a neve. Ha nem ismert, akkor üresen lehet hagyni.

típus: text

**kiado:** A térkép kiadójának neve. Ha nem ismert, akkor üresen lehet hagyni.

típus: text

**meretarany:** A térkép névleges méretaránya. Ha több lapot tartalmaz a mű, és azoknak eltér a méretaránya, vagy ha nem ismert a méretarány, akkor üres.

típus: text

**kiadas\_ eve:** A térkép kiadásának éve. Ha ismeretlen, akkor a mező üres.

típus: text

**meret:** Az eredeti térkép mérete centiméterben. Megadás formátuma: *szélességxmagasság*cm.

típus: text



**kiadas\_helye:** A térkép kiadásának a helye (településnév). Ha ismeretlen, akkor a mező üres.

típus: text

**terkep\_nyelve:** A térkép névrajzának nyelve. Ha többnyelvű a térkép, akkor minden nyelvet be kell írni.

típus: text

**atlasz:** Ha az eredeti térkép atlaszban van, akkor 1, amúgy 0.

típus: bool

#### **mularch:**

**hely:** Az intézet helye. Megegyezik a *terkereso* tábla *hely* mezőjével.

típus: text

**lelohely:** Az intézetre utaló kód. Megegyezik a *katalogusok* tábla *katalogus* mezőjével.

típus: text

**foto:** 1, ha van fotó az intézetről; 0, ha nincs.

típus: bool

#### **nepesség:**

**vm:** A vármegye kódja. Megegyezik a *vm* tábla *vmkod* mezőjével.

típus: text

**terulet:** A vármegye területe négyzetkilométerben.

típus: int

**nepesség:** A vármegye össznépességének a száma.

típus: int

**magyar:** A vármegyében élő magyarok száma.

típus: int

**nemet:** A vármegyében élő németek száma.

típus: int

**tot:** A vármegyében élő szlovákok száma.

típus: int

**olah:** A vármegyében élő románok száma.

típus: int

**ruten:** A vármegyében élő ruténok száma.

típus: int

**horvat:** A vármegyében élő horvátok száma.

típus: int

**szerb:** A vármegyében élő szerbek száma.

típus: int

**egyebnep:** A vármegyében élő egyéb, a többi mezőben meg nem jelenő népcsoporthoz tartozók száma.

típus: int

**romaikat:** A vármegyében élő római katolikus felekezethez tartozók száma.

típus: int

**gorogkat:** A vármegyében élő görög katolikus felekezethez tartozók száma.

típus: int

**reformatus:** A vármegyében élő kálvinista református felekezethez tartozók száma.

típus: int

**evangelikus:** A vármegyében élő ágostai evangélikus felekezethez tartozók száma.

típus: int

**ortodox:** A vármegyében élő ortodox felekezethez tartozók száma.

típus: int

**unitartius:** A vármegyében élő unitárius felekezethez tartozók száma.

típus: int

**izrealita:** A vármegyében élő izraelita felekezethez tartozók száma.

típus: int

**egyebvallás:** A vármegyében élő, a fent fel nem sorolt felekezethez tartozók, valamint a felekezeten kívüliek száma.

típus: int

#### **nyelvek:**

**nyelv:** A bejegyzés nyelvére való utalás. Három értékre lehet: HU - magyar, DE - német, EN - angol.

típus: text

**cím:** A honlap címe.

típus: text

**többi mező:** A honlapon lévő egyes bejegyzések.

típus: text

#### **országok:**

**kod:** Az ország kétbetűs azonosítója.

típus: text

**HU:** Az ország neve magyarul.

típus: text

**DE:** Az ország neve németül.

típus: text

**EN:** Az ország neve angolul.

típus: text

**telszotar:**

**jel:** A vármegye kódja, amelynek a székhelyére vonatkozik a bejegyzés. Megegyezik a *vm* tábla *vmkod* mezőjével. A mező kitöltése kötelező.

típus: text

**HU:** A vármegye székhelyének neve magyarul. A mező kitöltése kötelező.

típus: text

**DE:** A vármegye székhelyének neve németül.

típus: text

**SK:** A vármegye székhelyének neve szlovákul.

típus: text

**RO:** A vármegye székhelyének neve románul.

típus: text

**HR:** A vármegye székhelyének neve horvátul, illetve szerbül (latin betűs átírásban).

típus: text

**UA:** A vármegye székhelyének neve ukránul (latin betűs átírásban).

típus: text

**PL:** A vármegye székhelyének neve lengyelül.

típus: text

**RU:** A vármegye székhelyének neve oroszul (latin betűs átírásban).

típus: text

**CZ:** A vármegye székhelyének neve csehül.

típus: text

**terkeres:**

**archive:** A levéltár megnevezése az adott településen. Ha egy levéltár van, megegyezik a *katalogusok* tábla *katalogus* mezőjének értékével, ha több, akkor 'mul' az értéke.

típus: text

**polygon:** Az akív poligon neve.

típus: text

**gr:** Ha az aktív poligont kifedi egy másik, esetleg áttetsző poligon (pl. kontúr), akkor annak a poligonnak a neve, amúgy üres.

típus: text

**terk:** A térkép megnevezése, amely az adott intézethez kapcsolódik.

típus: text

**hely:** A hely megnevezése, amit az aktív poligon ábrázol.

típus: text

#### **tooltip:**

**nyelv:** A bejegyzés nyelvére vonatkozó adat. Három értékre lehet: HU - magyar, DE - német, EN – angol.

típus: text

**icon:** Az ikon neve.

típus: text

**tart:** A tooltip szövege.

típus: text

#### **szadatok:**

**nyelv:** A bejegyzés nyelvére vonatkozó adat. Három értékre lehet: HU - magyar, DE - német, EN – angol.

**megnev:** A szerep, amit a megnevezett személy a honlap készítésében betöltött.

**adat:** A személyek nevei.

#### **forras:**

**nyelv:** A bejegyzés nyelvére vonatkozó adat. Három értékre lehet: HU - magyar, DE - német, EN – angol.

**megnevezes:** Adat, aminek a forrását meg akarjuk adni.

**adat:** A megnevezett adat forrása.

#### **statszin:**

**kat:** A kategória száma.

**t:** A jelkulcsi elem azon a térképen, amely bemutatja, hogy hány térképen van ábrázolva az adott vármegye.

**r:** A jelkulcsi elem azon a térképen, amely bemutatja, hogy hány térképen a vármegyének csak egy része van ábrázolva.

**e:** A jelkulcsi elem azon a térképen, amely bemutatja, hogy hány térképen nincs ábrázolva az adott vármegye

**vm:**

**vmkod:** A vármegye kódja.

típus: text

**vmnev:** A vármegye neve.

típus: text

**vmtart:**

**sorszam:** A bejegyzéshez tartozó térkép sorszáma. Megegyezik a *katalogus\_bov* tábla *sorszam* mezőjével

típus: text

**Többi mező:** A mező neve megegyezik a vármegyék *vm* tábla *vmkod* mezőben megadott kódjával. A tartalma lehet „t”, ha az adott térkép a vármegye teljes területét ábrázolja; „r”, ha a térkép a vármegye területének egy részét ábrázolja; és „e”, ha a térkép nem ábrázolja a vármegyét. A horvátországi megyékre is vizsgálva van, viszont Fiuméra nincs. Budapest székesfőváros Pest-Pilis-Solt-Kiskun vármegyére van vonatkoztatva. A többi törvényhatóságú város is a várost körülvevő vármegyére van vonatkoztatva, nincsenek különvéve.

**Egyéb megjegyzések az adatbázishoz:**

Az összes mellékjeles, illetve különleges karakter entitýkkel van tárolva.

## IV. A könyvtárstruktúra felépítése

### IV.1 A webservert adatai

**Operációs rendszer:** Linux

**Webservert:** Apache

**Szertvert oldali szkriptek:** php

### IV.2 A weben nem elérhető könyvtárak

A [lazarus.elte.hu](http://lazarus.elte.hu)-n a webről közvetlen nem elérhető, de a honlap által használt könyvtárak a `/home/janer` könyvtárban megtalálhatók.

A webről elérhető könyvtárak a [lazarus.elte.hu-n /var/www/demogmap](http://lazarus.elte.hu-n/var/www/demogmap) könyvtárban található. Ehhez vezet egy szimbolikus link a [/home/janer/demogmap](http://home/janer/demogmap) a könnyebb elérés érdekében. A könyvtár elérhető a [lazarus.elte.hu/demogmap](http://lazarus.elte.hu/demogmap) címről, valamint a [demogmap.elte.hu](http://demogmap.elte.hu) virtuális hosztról.

## **A /home/janer könyvtárstruktúrája:**

### **imageshr:**

Beszkenelt térképek teljes felbontásban, jpeg formátumban.

### **imagesbig:**

Beszkenelt térképek maximum 3800px szélességben, jpeg formátumban.

### **images:**

Beszkenelt térképek maximum 300px szélességben, illetve 250px magasságban, jpeg formátumban.

### **gif:**

Beszkenelt térképek maximum 100px szélességben, illetve 100px magasságban, gif formátumban

### **varmegyetc:**

A Gönczy-féle vármegyetérkép-sorozat nagy felbontásban. Az állománynevek a vármegyék kódjával megegyeznek az adatbázis *vm* táblájában a *vmkod* tartalmával.

### **kvm:**

A Gönczy-féle vármegyetérkép-sorozat kis felbontásban. Az állománynevek a vármegyék kódjával megegyeznek az adatbázis *vm* táblájában a *vmkod* tartalmával.

### **foto:**

Az intézetekről készített fénykép, max 600px szélességben, illetve álló kép esetén maximum 600px magasságban. Az állománynevek megegyeznek a katalogusok tábla katalogus mezejének értékével.

## **imageshr, imagesbig, images, gif könyvtárak belső struktúrája:**

A térképek az intézetek szerinti mappákban vannak elhelyezve. A mappák nevei megegyeznek a *katalogusok* tábla *katalogus* mezejének értékével.

Ezen belül az állománynevek az egylapos térképművek esetén megegyeznek a *katalogus\_bov* táblában lévő *filenev* mező értékével + a kiterjesztés.

A többlapos térképművek esetén az állománynév a *katalogus\_bov* táblában lévő *filenev*

mező értéke + a szelvény száma (nem lehet nagyobb, mint a *kepszam* mező értéke) + a kiterjesztés.

### **IV.3 A weben elérhető könyvtárak felépítése**

#### **demogmap:**

Szimbolikus link a /var/www/demogmap könyvtárra.

#### **/var/www/demogmap/ könyvtár felépítése:**

**background:** A nyitólapon megjelenő háttérképet tartalmazza, ht1.jpg névvel.

**help:** A súgó elemei.

**icon:** A honlapon lévő ikonok.

**iconvm:** A vármegye térképek ikon méretben.

**maps:** A honlapon lévő vektoros térképek svg formátumban.

**statjm:** A statisztikai térképekhez tartozó jelmagyarázat.

## **V. A honlapon lévő vektoros térképek**

### **V.1 Felhasznált vektoros térképek:**

**bw.svg:** Baden-Württemberg térképe.

**europe.svg:** Európa térkép.

**germany.svg:** Németország térképe.

**hungary\_1914.svg:** Magyarország történelmi térképe az 1914-es határokkal, valamint vármegyehatárokkal. A jelmagyarázat a vármegyés keresőhöz kapcsolódik.

**hungary\_1914\_noleg.svg:** Magyarország történelmi térképe az 1914-es határokkal, valamint vármegyehatárokkal, jelmagyarázat nélkül.

**hungary.svg:** Magyarország térképe a mai megyékkal és Budapesttel külön.

## V.2 Alapanyagok minősége:

A névrajz mindegyiken kifejezetten rossz volt, így azt át kellett írni. A weben történő felhasználás előtt, azért, hogy a betűtípussal ne legyen gond, az összes névrajzi elemet görbévé kell alakítani. A Baden-Württemberg térképnél a poligonok neveit meghagytam, és csak átmásoltam az adatbázisba. Mivel ezen a térképen a kontúr külön poligon volt, így ezek neveit kellett betenni a gr mezőbe a terkereso táblába. Ezek lettek érzékenyek az egérmozdulatra, de a kitöltést az alatta lévő poligonnak adtam.

A Németország térképen nem volt névrajz, így azt meg kellett írni egy Németország igazgatási térkép alapján. (Forrás a 2006-os német nemzeti atlasz.)

A poligonok nevei a szövetségi államok rövidítését kapják (pl. Baden-Württemberg: BW).

A Magyarország térképen a névrajzot úgyszintén meg kellett írni. A poligonok nevei a megyékre utalnak.

Az Európa térképen sem volt névrajz. A poligonok nevei az ENSZ által is elfogadott, az országhoz tartozó kétbetűs állampolgársági kód, csupa kisbetűvel. Ez így volt az állományban, és így hagytam. A névrajzot elkészítettem magyarul, majd kitöröltem az svg forráskódjában a névrajzi elem tartalmát, otthagya magát az elemet, és a DOM segítségével a javascript visszatette az országok táblából a megfelelő nyelvű névrajzot. Az üres névrajzi elemek azonosítója: *t+országkód*

Habár a történelmi Magyarország térképen volt névrajz, mégis újra kellett írni, mivel semmiben nem felelt meg a kartográfiai követelményeknek. A vármegyék poligonjainak nevét meg kellett cserélni, hogy megegyezzen a *vm* tábla *vmkod* mezőjében lévő értékkel.

A poligonok kitöltését az összes térképen megcseréltem, hogy megegyezzen, bár a javascript eljárás sok esetben úgyis megcseréli. A jelmagyarázat a vármegyés keresőben használt színekkel van elkészítve. Van egy változat, amin nincs jelmagyarázat, ezt leszámítva a két állomány azonos.

## VI. Kliens oldali függvények

Az általam írt kliens oldali függvények két állományban vannak:

*ajaxmain.js* – Ezt az állományt az *index.html* használja, és elérhető minden olyan állományból, amit az *index.html* állományba vagy az abba beillesztett állományba illesztünk be.

*ajaxnpage.js* – Ezt az állományt a *mapperkint.html* használja, és elérhető minden olyan állományból, amit a *mapperkint.html* állományba illesztettünk bele.



## Az *ajaxmain.js* fontosabb függvényei:

### **preoverlay()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: Előkészíti az üres popup ablakot. A jobb felső sarkában megjelenik a bezárás gomb, egyébként üres. A beágyazott objektumok mind rejtve maradnak. A *mapper*, *mapperclose* és a *mapviewer* objektumok jelennek meg.

### **mouseOverIcon(obj):**

Bementeti paraméterek:

*obj*: a kezelendő objektum. Eseménykezelőben a *this* kulcsszóval adható meg, tehát az az objektum, aminek az eseménykezelője hivatkozott erre a függvényre. Az *onmouseover* eseménykezelővel használjuk.

Visszaadási érték: nincs.

Kezelt objektumok: Az objektum, amely fölé az egeret húzzuk kap egy piros keretet. Feltétele, hogy legyen egy áttetsző kerete, mivel a függvény nem definiál keretet, csak a keret színét cseréli meg.

### **mouseOutIcon(obj)**

Bementeti paraméterek:

*obj*: a kezelendő objektum. Eseménykezelőben a *this* kulcsszóval adható meg, tehát az az objektum, aminek az eseménykezelője hivatkozott erre a függvényre. Az *onmouseout* eseménykezelővel használjuk.

Visszaadási érték: nincs.

Kezelt objektumok: Az objektum, amelyet elhagyunk az egérrel elveszíti a piros keretét. Csak a színét változtatja meg áttetszőre.

### **tooltip(tt,event)**

Bemeneti paraméterek:

*tt*: Egy string érték, ami megadja, hogy melyik tooltip jelenjen meg. Megegyezik a *tooltip* tábla *icon* oszlopában lévő értékkel.

*event*: Mindig az event kulcsszót kell megadni, amely átadja neki a mouseover eseményt.

Ebből olvassa majd ki a függvény az egér koordinátáit.

Visszaadási érték: nincs.

Kezelt objektumok: a *tooltip* objektumot minden objektum fölé helyezi, a bal felső sarkát az egérmutató közelében helyezi el, az objektum tartalma a *tooltip* tábla tart oszlopának a bemeneti paraméter szerint megadott értékének megfelelő sor.

### **tooltipb():**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: A *tooltip* objektumot minden objektum alá helyezi, és kitörli a tartalmát.

### **advsearch():**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: az *advsearch* objektumot megjeleníti.

## **szadat()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: Lefuttatja a *preoverlay()* függvényt. Az *szdata* objektumba helyezi a *szadat.php* visszaadási értékét. Az *szadat.php* bemeneti paramétereinek a *lang* paraméternek megadja a *lang* globális változó értékét. Megjeleníti az *szdata* objektumot.

## **intinfo():**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: Lefuttatja a *preoverlay()* függvényt. Az *intinfo* objektumba beleteszi az *intinfo.php* eredményét, aminek a *lang* paraméterében a *lang* globális változó értékét adja meg. Megjeleníti az *intinfo* objektumot.

## **forras()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: Lefuttatja a *preoverlay()* függvényt. Megjeleníti a *forras* objektumot.

## **telszotar()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: Lefuttatja a *preoverlay()* függvényt. Megjeleníti a *telszocont* objektumot.

## **elk()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: Megjeleníti a *searchres* objektumot, ezzel megjelenik az előző keresési eredmény. A *res* objektumon szükség esetén létrehozza a görgetősávokat. A *cont*, *germany*, *hungary*, *bw*, *vmkereso*, és *maptitle* objektumokat elrejti, így a *res* objektumba csak a *searchres* objektum lesz látható. Az *mg1*, *mg2*, és *mg3* objektumok tartalmát törli, így az oldal jobb oldala üres lesz.

## **nepterkep()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: Lefuttatja a *preoverlay()* függvényt. Megjeleníti a *nepterkep* objektumot.

## **statmapshow()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: Lefuttatja a *preoverlay()* függvényt. Megjeleníti a *statterkep* objektumot.

## **vmkeresov()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: Elrejti a *cont*, *searchres*, *germany*, *hungary*, *bw* és *maptitle* objektumokat.

Megjeleníti a *vmkereso* objektumokat. Így a *res* objektumba csak a *vmkereso* objektum lesz látható.

A *res* objektumon a görgetősávokat elrejtí. Az *mg1* objektum tartalmát kicseréli a következőre:

```
„<OBJECT width=“100%” data=“maps/hungary_1914_noleg.svg” type=“image/svg+xml”  
id=“nagymouk”></OBJECT>”. Ezzel megjelenik az mg1 objektumban egy történelmi
```

közigazgatási térkép Magyarország 1914-es közigazgatásáról.

A *mg2* objektumba beleteszi a *nepkeres.php* eredményét, a *lang* paraméternek a *lang* globális változó értékét adja meg.

Az *mg3* objektum tartalmát törli.

### **aktivkeres()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: Megvizsgálja az *aktivkereso* globális változó értékét. Ha igaz, akkor a *search* objektum háttér színét a 030C3D kódolásra cseréli (sötétkék); ha hamis, akkor a 700000 kódolásra (bordó).

Globális változók értékének változása: Ha az *aktivkereso* változó hamis, akkor igaz lesz; ha igaz akkor hamis lesz.

### **terkeresov()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: Megjeleníti a *cont* és *maptitle* objektumokat. Elrejtí a *searchres*, *germany*, *hungary*, *bw* és *vmkereso* objektumokat. Az *mg1*, *mg2* és *mg3* objektumok tartalmát törli. A *res* objektumon elrejtí a görgetősávokat.

Globális változók értékének változása: Az *otk* globális változó értékét „cont”-ra állítja.

## **init()**

Bemeneti paraméterek: nincs.

Visszaadási érték: Igaz, ha sikeresen lefutott; hamis, ha hibát észlelt. Hiba esetén feljön egy hibaüzenet.

Globális változók kezelése: Az *xhttp* globális változóban létrehoz egy *XMLHttpRequest* objektumot, amelyet a többi függvény felhasznál a szerverre történő kérések elküldésére. Fontos, hogy minden szerverre történő kérelem elküldése előtt ezt a függvényt le kell futtatni.

Létrehozott eseménykezelők: Ha a böngészőablak mérete megváltozik, lefuttatja a *windowSize()* függvényt.

A következő függvényeket futtatja le:

*windowSize()*;

*browser\_detect()*;

*inittelszo()*;

*initnepterkep()*;

*terkereso()*;

*search()*;

*cim()*;

*terkeresofi()*;

*telszosetup()*;

*updatemap()*;

*stattkinit()*;

*preforras()*;

Kezelt hibák: Ha nem tudja létrehozni az *xhttp* objektumot, akkor hibakezelés lép érvénybe.

## **windowSize()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Globális változók kezelése: A *winW* globális változóba beleteszi a böngészőablak szélességét. A *winH* globális változóba beleteszi a böngészőablak magasságát.

Kezelt objektumok: Ha a *winW* globális változó nagyobb, mint 1170, akkor a *mapper*, *mapperclose* és *mapviewer* objektumok szélességét 1170-re definiálja. Ugyanezen objektumok bal oldalának koordinátáit kiszámolja: kivonja az objektum szélességének a negyedét az ablak szélességének a negyedéből. Ha a *winW* kisebb, mint 1170, akkor a *winW*-től 50-nel kisebb lesz az objektumok szélessége, és a bal szélüknek koordinátája 10 lesz.

Ha a *winH* változó nagyobb, mint 774, akkor a *mapper* és a *mapviewer* objektumok magassága 774 lesz. A felső szélének koordinátái úgy lesznek kiszámolva, hogy kivonja az objektumok magasságának a negyedét az ablak magasságának a negyedéből. Ha a *winH* kisebb, mint 774, akkor az objektumok magassága az ablak magasságtól 40-nel lesz kevesebb, és a felső szélük 10 lesz. A *mapperclose* objektum magasságát, bár nem állítja, de a felső szélét a többivel megegyezőnek állítja be.

### **browser\_detect()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Globális változók kezelése: A *browser* globális változó értékét „FF”-nek definiálja, ha Firefoxsal nyitottuk meg az oldalt, és „IE”-nek a Microsoft Internet Explorer. a *browser\_version* változóba beteszi a böngésző verziószámát. Ha az oldalon a *lang* paraméter definiálva van (tehát a böngészőbe ?lang=....) akkor annak értékét helyezi be a *lang* globális változóba. Ha nincs definiálva, akkor megvizsgálja a böngésző nyelvét. Ha magyar, akkor a *lang* értéke „HU” lesz; ha német, akkor „DE”; ha meg angol, akkor „EN”. Ha nem tudja megállapítani, vagy nem az előbb felsorolt három nyelv egyike, akkor „EN” lesz. Így, ha nem egy, a honlap által ismert nyelven van feltelepítve a böngésző, és a felhasználó nem adott meg nyelvet, akkor a lap angolul jelenik meg.

## **inittelszo()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: A *telszocont* objektum tartalmát definiálja a *telszo.php* eredménye szerint. A *telszo.php*-nak a *lang* paraméternek a *lang* globális változó értékét adja meg. A *telszocont* objektum rejtve marad.

## **initnepterkep()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: A *nepterkep* objektum tartalmát definiálja a *nepterkep.php* eredménye szerint. A *nepterkep.php*-nak a *lang* paraméternek a *lang* globális változó értékét adja meg. A *nepterkep* objektum rejtve marad.

## **terkereso()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: A *res* objektumban a szöveget középre helyezi. A *maptitle* objektum tartalmát megteszi a *felirat* függvény visszatérítési értékének, aminek a bemeneti paraméterének 'el28' -at ad meg. A *maptitle* objektumot megjeleníti.

## **search()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.



Kezelt objektumok: A *search* objektum tartalmát definiálja a *search.php* eredménye szerint. A *search.php*-nak a *lang* paraméternek a *lang* globális változó értékét adja meg. Az *advsearch* objektum tartalmát definiálja az *advsearch.php* eredménye szerint. Az *advsearch.php*-nak a *lang* paraméternek a *lang* globális változó értékét adja meg. Az *advsearch* objektum rejtve marad, viszont a *search* objektum megjelenik.

### **cim()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: A lap címének definiálja a *cim.php* visszaadási értékét, miután az escape karaktereket eltávolította. Ugyanerre cseréli a *h1* tartalmát is. A *cim.php*-nek a *lang* paraméternek a *lang* globális változó értékét adja meg.

### **terkeresofi()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: A *map1* objektum (*bw.svg*) aktív poligonjait definiálja. Az aktív poligonok listáját az *archives.php* visszaadási értéke adja meg (az eredmény egy XML állomány). Ezt a *terk=bw* paraméter megadásával kéri le a szerverről. Az xml állományba visszkapott poligon értékének megfelelő nevű, a *map1* objektumon belül lévő objektumot kitölti lilával (7E45B3). A *gr*-ben visszkapott értéknek megfelelő nevű objektumhoz három darab eseménykezelőt rendel. Mivel az *svg* állomány be van ágyazva a *html* oldalba, viszont az eseménykezelők által keresett függvények a *html* oldalban vannak, így a *parent.fvnev* formátumban kell hivatkozni rájuk.

A három eseménykezelő a következő:

*onmouseover*: `parent.mouseOverMap('poligon','terk')`, ahol a *poligon* az xml állomány *polygon* attribútumának értéke, a *terk* az xml állomány *terk* attribútumának értéke.

*onmouseout*: `parent.mouseOutMap('polygon','terk')`, ahol a *poligon* az xml állomány *polygon* attribútumának értéke, a *terk* az xml állomány *terk* attribútumának értéke.

*onclick*: `parent.onClickMap('archive','polygon','terk','hely')`, ahol az *archive* az xml állomány *archive* attribútumának értéke, a *poligon* az xml állomány *polygon* attribútumának értéke, a *terk* az xml állomány *terk* attribútumának értéke, a *hely* az xml állomány *hely* attribútumának értéke.

Hasonlóan a *hu* objektumhoz (*hungary.svg*) hozzárendeli az *archive.php* eredmény alapján az aktív felületeket. Ez esetben a *terk=mo* paramétert kell megadni. Ebben az esetben is az aktív poligont kitölti. A szerverről visszakapott xml állományban az adott sor *polygon* attribútumának értékével megegyező objektumot kitölti lilával (7E45B3). Ebben az esetben viszont az eseménykezelőket ugyanehhez az objektumhoz rendeli (lásd: V.2 Alapanyagok minősége című mellékletet):

*onmouseover*: `parent.mouseOverMap('poligon','terk')`, ahol a *poligon* az xml állomány *polygon* attribútumának értéke, a *terk* az xml állomány *terk* attribútumának értéke.

*onmouseout*: `parent.mouseOutMap('polygon','terk')`, ahol a *poligon* az xml állomány *polygon* attribútumának értéke, a *terk* az xml állomány *terk* attribútumának értéke.

*onclick*: `parent.onClickMap('archive','polygon','terk','hely')`, ahol az *archive* az xml állomány *archive* attribútumának értéke, a *poligon* az xml állomány *polygon* attribútumának értéke, a *terk* az xml állomány *terk* attribútumának értéke, a *hely* az xml állomány *hely* attribútumának értéke.

Ezek után a program az *eu* (*europe.svg*) objektumon is elkészíti az aktív felületeket és a névrajzot.

Az *orszagok.php* eredményét használja fel. Kezdeti paramétereket nem ad meg. Az eredmény itt is egy xml állomány.

Az előbb a *polygon* nevű attribútum értékével megegyező nevű objektumnak az *eu* objektumon belül megcseréli lilára (7E45B3) a kitöltését. Utána hozzárendel három darab eseménykezelőt. Ugyanúgy mint az eddigieknél, a *parent* objektumot kell használni, mivel ez is egy beágyazott objektum:

*onmouseover*: `parent.mouseOver('polygon')`, ahol a *polygon* megegyezik az xml állomány *polygon* attribútumának értékével.

*onmouseout*: `parent.mouseOut('poligon')`, ahol a *polygon* megegyezik az xml állomány *polygon* attribútumának értékével.

*onclick*: `parent.onClick('terkepnev')`, ahol a *terkepnev* megegyezik az xml állomány *terkepnev* attribútumának értékével.

Ezek után a program elkészíti a feliratokat. Ezeket egy xml állományban kapja meg az *orszagnev.php*-től. Kezdeti paraméternek a *lang*-ot adja meg, aminek az értéke a *lang* változó kezdeti értéke.

*t+* a kód attribútum értékével megegyező objektumhoz hozzáad egy újabb objektumot, amelynek a tartama a *text* attribútum értéke.

Majd a *de* objektumon hozza létre az aktív felületeket. A felületek létrehozása a *ger.php* eredményként kapott xml állomány alapján történik. A *polygon* attribútum értékével megegyező nevű objektum kitöltését lilára (7E45B3) cseréli. Ezek után ugyanehhez az objektumhoz ad három darab eseménykezelőt.

*onmouseover*: `parent.mouseOverg('polygon')`, ahol a *polygon* megegyezik az xml állomány *polygon* attribútumának értékével.

*onmouseout*: `parent.mouseOutg('polygon')`, ahol a *polygon* megegyezik az xml állomány *polygon* attribútumának értékével.

*onclick*: `parent.onClick('tk')`, ahol a *tk* megegyezik az xml állomány *tk* attribútumának értékével.

Ezek után a program létrehozza a *mo1914* (*hungary\_1914.svg*) objektum aktív felületeit. Az xml állományt a *vm.php*-től kapja vissza. Kezdeti paramétereket nem ad meg. A *vm* globális tömb, az xml állomány *vmkod* attribútumának értékével megegyező nevű indexének értékét *m*-re állítja. A *mo1914*-ben a *vmkod* attribútum értékével megegyező nevű objektumhoz három darab eseménykezelőt rendel:

*onmouseover*: `mouseOverVM('vmkod')` ahol a *vmkod* megegyezik az xml állományban a *vmkod* attribútum értékével.

*onmouseout*: `mouseOutVM('vmkod')` ahol a *vmkod* megegyezik az xml állományban a *vmkod* attribútum értékével.

*onclick*: `mouseClickVM('vmkod')` ahol a *vmkod* megegyezik az xml állományban a *vmkod* attribútum értékével.

Ezek után a program a *bttm\_vmall* objektum értékét a *felirat* függvény visszatérítési értékével teszi egyenlővé, aminek 'el31'-et ad meg, mint az egyetlen bemeneti paramétert. Majd a *bttm\_vmres* objektum értékét a *felirat* függvény visszatérítési értékével teszi egyenlővé, aminek 'el32'-t ad meg, mint az egyetlen bemeneti paramétert. És végül a *bttm\_vmkeres* objektum értékét a *felirat* függvény visszatérítési értékével teszi egyenlővé, aminek 'el4'-et ad meg, mint az egyetlen bemeneti paramétert.

## **telszosetup()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: A *telmap1* objektumon létrehozza az aktív felületeket.

Ezt egy a *vm.php* eredményként kapott xml állomány alapján teszi. A *vm.php*-t bemeneti

paraméterek nélkül hívja fel.

Az xml állomány *vmkod* attribútumának értékével megegyező nevű objektumhoz három darab eseménykezelőt rendel:

*onmouseover*: `mouseoverVMT('vmkod')`, ahol a *vmkod* megegyezik az xml állományban a *vmkod* attribútum értékével.

*onmouseout*: `mouseoutVMT('vmkod')`, ahol a *vmkod* megegyezik az xml állományban a *vmkod* attribútum értékével.

*onclick*: `mouseClickVMT('vmkod')`, ahol a *vmkod* megegyezik az xml állományban a *vmkod* attribútum értékével.

## **updatemap()**

Bemeneti paraméterek:

*nat*: Azon nemzetiség neve, amelyiket a népességtérképen akarunk ábrázolni (megegyezik a *nepesseg* nevű tábla oszlopneveivel).

Visszaadási érték: nincs.

Kezelt objektumok: Előbb kiszámolja a *nat* változóban megadott nemzetiség arányát. A *nat* értékével megegyező nevű attribútum értékét elosztja a *nepesseg* nevű attribútum értékével. Ezek után a *vm* nevű attribútum értékével megegyező nevű objektum kitöltését megváltoztatja. Hogy mire, az attól függ, hogy mi a *nat* értéké, valamint hogy mi a korábban kiszámolt arány.

Ha a *nat* értéke 'magyar' és az arány:

- < 0.01, a kitöltés FFFFFFFF,
- 0.1 >= 0.01, a kitöltés FFE0E0,
- 0.2 >= 0.1, a kitöltés FFD1D1,
- 0.3 >= 0.2, a kitöltés FFC1C1,
- 0.4 >= 0.3, a kitöltés FFB8B8,
- 0.5 >= 0.4, a kitöltés FFA8A8,
- 0.6 >= 0.5, a kitöltés FF9C9C,
- 0.7 >= 0.6, a kitöltés FF8A8A,
- 0.7 >= 0.8, a kitöltés FF7373,
- 0.8 >= 0.9, a kitöltés FF5757,
- >= 0.9, a kitöltés FF3333.

Ha a *nat* értéke 'olah' és az arány:

- < 0.01, a kitöltés FFFFFFFF,
- 0.1 >= 0.01, a kitöltés B980C8,
- 0.2 >= 0.1, a kitöltés B671C8,
- 0.3 >= 0.2, a kitöltés B25CC8,
- 0.4 >= 0.3, a kitöltés 9B40B2,
- 0.5 >= 0.4, a kitöltés 962BB2,
- 0.6 >= 0.5, a kitöltés 9110B2,
- 0.7 >= 0.6, a kitöltés 7D089C,
- 0.7 >= 0.8, a kitöltés 690782,
- 0.8 >= 0.9, a kitöltés 590070,
- >= 0.9, a kitöltés 450057.

Ha a *nat* értéke 'tot' és az arány:

- < 0.01, a kitöltés FFFFFFFF,
- 0.1 >= 0.01, a kitöltés CBFFC8,
- 0.2 >= 0.1, a kitöltés BDFFBA,
- 0.3 >= 0.2, a kitöltés B1FFAD,
- 0.4 >= 0.3, a kitöltés A3FF9E,
- 0.5 >= 0.4, a kitöltés 92FF8C,
- 0.6 >= 0.5, a kitöltés 83FF7D,
- 0.7 >= 0.6, a kitöltés 72FF6B,
- 0.7 >= 0.8, a kitöltés 5CFF54,
- 0.8 >= 0.9, a kitöltés 2FD627,
- >= 0.9, a kitöltés 09BD00.

Ha a *nat* értéke 'nemet' és az arány:

- < 0.01, a kitöltés FFFFFFFF,
- 0.1 >= 0.01, a kitöltés F9D6BC,
- 0.2 >= 0.1, a kitöltés F9CAA7,
- 0.3 >= 0.2, a kitöltés F9BD90,
- 0.4 >= 0.3, a kitöltés F9B07A,
- 0.5 >= 0.4, a kitöltés F9A261,
- 0.6 >= 0.5, a kitöltés F99348,
- 0.7 >= 0.6, a kitöltés F9822A,
- 0.7 >= 0.8, a kitöltés F9730F,
- 0.8 >= 0.9, a kitöltés DE5F00,

$\geq 0.9$ , a kitöltés ED4F00.

Ha a *nat* értéke 'ruten' és az arány:

$< 0.01$ , a kitöltés FFFFFFFF,

$0.1 \geq 0.01$ , a kitöltés C8A08C,

$0.2 \geq 0.1$ , a kitöltés C89277,

$0.3 \geq 0.2$ , a kitöltés B47D61,

$0.4 \geq 0.3$ , a kitöltés A1674B,

$0.5 \geq 0.4$ , a kitöltés 91573B,

$0.6 \geq 0.5$ , a kitöltés 82492D,

$0.7 \geq 0.6$ , a kitöltés 6A371E,

$0.7 \geq 0.8$ , a kitöltés 572005,

$0.8 \geq 0.9$ , a kitöltés 3B1604,

$\geq 0.9$ , a kitöltés 361200.

Ha a *nat* értéke 'horvat' és az arány:

$< 0.01$ , a kitöltés FFFFFFFF,

$0.1 \geq 0.01$ , a kitöltés D2CAFF,

$0.2 \geq 0.1$ , a kitöltés C7BDFF,

$0.3 \geq 0.2$ , a kitöltés BEB3FF,

$0.4 \geq 0.3$ , a kitöltés B1A3FF,

$0.5 \geq 0.4$ , a kitöltés A696FF,

$0.6 \geq 0.5$ , a kitöltés 9987FF,

$0.7 \geq 0.6$ , a kitöltés 846EFF,

$0.7 \geq 0.8$ , a kitöltés 684DFF,

$0.8 \geq 0.9$ , a kitöltés 4C2BFF,

$\geq 0.9$ , a kitöltés 2700FF.

Ha a *nat* értéke 'szerb' és az arány:

$< 0.01$ , a kitöltés FFFFFFFF,

$0.1 \geq 0.01$ , a kitöltés FFFEEA,

$0.2 \geq 0.1$ , a kitöltés FFFCCF,

$0.3 \geq 0.2$ , a kitöltés FFFBBA,

$0.4 \geq 0.3$ , a kitöltés FFFAA3,

$0.5 \geq 0.4$ , a kitöltés FFF88F,

$0.6 \geq 0.5$ , a kitöltés FFF775,

$0.7 \geq 0.6$ , a kitöltés FFF559,

$0.7 \geq 0.8$ , a kitöltés FFF442,

0.8 >= 0.9, a kitöltés FFF32B ,  
>= 0.9, a kitöltés FFF000.

Ha a *nat* értéke 'egyebnep' és az arány:

< 0.01, a kitöltés FFFFFFFF,  
0.1 >= 0.01, a kitöltés FFFFFFFF,  
0.2 >= 0.1, a kitöltés D8D8D8,  
0.3 >= 0.2, a kitöltés BFBFBF,  
0.4 >= 0.3, a kitöltés B3B3B3,  
0.5 >= 0.4, a kitöltés A3A3A3,  
0.6 >= 0.5, a kitöltés 949494,  
0.7 >= 0.6, a kitöltés 808080,  
0.7 >= 0.8, a kitöltés 6E6E6E,  
0.8 >= 0.9, a kitöltés 5C5C5C,  
>= 0.9, a kitöltés 4A4A4A.

### **stattkinit()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: A *statintsel* nevű objektumba helyezi a *statmenu.php* visszaadási értékét. A *lang* kezdeti paramétert értéke a *lang* globális változó értéke. A *statcim* objektumba a *felirat* függvény visszatérítési értéket helyezi, az egyetlen bemeneti paraméter értéke 'el57'. Lefuttatja a *stattk* függvényt. Ennek az egyetlen bemeneti paraméterének a 'barmi' értéket adja.

### **stattk(sel)**

Bemeneti paraméter:

*sel*: azon intézet, amiről statisztikai térképet készítünk.

Visszaadási érték: nincs.

Kezelt objektumok: Ha a *sel* értéke 'barmi' akkor a *statkat.php* kezdeti paraméterek nélkül lesz az eljárásban felhasznált xml állomány. Ha más, akkor a *sel* értéke lesz a kezdeti paraméter a

*statkat.php*-hez (ami a megadott intézet kódja kell legyen), és akkor az így kapott eredmény az xml állomány. A *statmap1* és *statmap4*-ben az *id* attribútum értékével megegyező nevű objektumot kitölti a *tcolor* attribútum értékével. A *statmap2* és *statmap5* nevű objektumokban az *id* attribútum értékével megegyező nevű objektumot kitölti a *rcolor* attribútum értékével. A *statmap3* és *statmap6* nevű objektumokban az *id* attribútum értékével megegyező nevű objektumot kitölti az *ecolor* attribútum értékével.

### **preforras()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: A *forras.php* eredményét, ahol a *lang* kezdeti paraméter megegyezik a *lang* globális változó értékével, a *forras* objektumba helyezi.

### **mouseOverMap(polygon, terk)**

Bemeneti paraméterek:

*polygon*: A kezelendő poligon.

*terk*: A kezelendő térkép.

Visszaadási érték: nincs.

Kezelt objektumok: Ha a *selterk* globális változó megegyezik a *terk* paraméterrel, és a *selpolygon* globális változó a *polygon* kezdeti paraméterrel, akkor a *terk*-ben megadott térképen, a poligonban megadott poligon kitöltését sötétvörösre (990000) cseréli. Ha ez a két feltétel nem áll fenn, akkor sötétlilára (4B007F).

### **mouseOutMap(polygon, terk)**

Bemeneti paraméterek:

*polygon*: A kezelendő poligon.

*terk*: A kezelendő térkép.



Visszaadási érték: nincs.

Kezelt objektumok: Ha a *selterk* globális változó megegyezik a *terk* paraméterrel, és a *selpolygon* globális változó a *polygon* kezdeti paraméterrel, akkor a *terk*-ben megadott térképen, a *polygon*ban megadott poligon kitöltését világos vörösre (E60000) cseréli. Ha ez a két feltétel nem áll fenn, akkor világos lilára (7E45B3).

### **onClickMap(archive,polygon,terk,hely)**

Bemeneti paraméterek:

*archive*: Azon intézmény kódja, amely a *polygon* paraméterben megjelölt poligon által ábrázolt településen található. Ha több mint egy intézmény van a településen, akkor ennek értéke 'mul'.

*polygon*: A térképen lévő poligon azonosítója.

*terk*: A térkép azonosítója.

*hely*: A hely kódja, amit a poligon ábrázol.

Visszaadási érték: nincs

Kezelt objektumok: Ha az *archive* értéke nem 'mul' akkor az 'opt'+*archive értéke* nevű objektum *seleceted* attribútumát beállítja. Ezzel az intézményeket felsoroló menüben kiválasztásra kerül a megadott intézmény. Ha a *selpolygon* értéke be van állítva valamire, akkor a *seltk* nevű térképen a *selpolygon* nevű poligon kitöltését megcseréli világos lilára (7E45B3). Ezek után a *polygon* paraméterben megjelölt poligon kitöltését sötétvörösre cseréli (990000). Majd lefuttatja a *pictupdate* függvényt, az egyetlen bemeneti paramétere az *archive* paraméter értéke. Ha az *archive* paraméter értéke 'mul', akkor a *mulpict* függvényt futtatja le, és az egyetlen bemeneti paraméterének értéke a *hely* paraméter értéke lesz.

Globális változók kezelése: A *selpolygon* globális változó értéke a *polygon* paraméter értéke lesz. A *seltk* globális változó értéke a *terk* paraméter értéke lesz.

## **pictupdate(archive)**

Bemeneti paraméterek:

*archive*: Azon katalogus kódja, aminek a képét meg akarjuk jeleníteni.

Visszaadási érték: nincs.

Kezelt objektumok: Ha az *archive* 'barmi', az *mg1* és *mg2* tartalmát törli. Ha nem, akkor a *mg2* tartalma a következő lesz:

```
„<TABLE color="#FFFFFF" cellpadding="7"><THEAD color="#FFFFFF"><TH  
class="cim">el19</TH><TD>név</TD><TR><TH  
class="cim">el20</TH><TD>cím</TD><TR><TH class="cim">el21</TH><TD><A  
href="honlap" target="_blank">honlap</A></TD></THEAD></TABLE>”,
```

ahol

*név* az *adatok.php* visszatérítési értéke, ha az *archive* bemeneti paraméter megegyezik a függvény *archive* paraméter értékével, a *lang* bemeneti paraméter megegyezik a *lang* globális változó értékével, és a *dat* paraméter értéke 'nev';

*cím* az *adatok.php* visszatérítési értéke, ha az *archive* bemeneti paraméter megegyezik a függvény *archive* paraméter értékével, a *lang* bemeneti paraméter megegyezik a *lang* globális változó értékével, és a *dat* paraméter értéke 'cim';

*honlap* az *adatok.php* visszatérítési értéke, ha az *archive* bemeneti paraméter megegyezik a függvény *archive* paraméter értékével, a *lang* bemeneti paraméter megegyezik a *lang* globális változó értékével, és a *dat* paraméter értéke 'honlap';

*el19* a felirat visszatérítési értéke, ha a bemeneti paraméter értéke 'el19';

*el20* a felirat visszatérítési értéke, ha a bemeneti paraméter értéke 'el20';

*el21* a felirat visszatérítési értéke, ha a bemeneti paraméter értéke 'el21';

*mg1* objektum tartalma ebben az esetben: „<IMG src="kfoto.php?img=archive"/>”,

ahol *archive* az előbbihez hasonlóan a függvény *archive* paraméterének értéke.

## **mulpict(hely)**

Bemeneti paraméterek:

*hely*: A megadott település, ahol több intézmény is található.



megjeleníti Magyarország vagy Baden-Württemberg közigazgatási térképét. Utána lefuttatja a *picupdate* függvényt, és bemenetei értéknek a *kat* paraméter értékét adja meg, ezzel az intézmény képét felhozza az *mgI*-be.

Globális változók kezelése: Ha *kat* = 'barmi', akkor a *seltk* globális változó értéke 'cont' lesz. Ha ettől eltér, akkor az xml állomány *terk* attribútumának értéke lesz. Ebben az esetben a *selpolygon* értéke az xml állomány *polygon* attribútumának értékének fog megfelelni.

### **mulpictooltip(cont, event)**

Bemeneti paraméterek:

*cont*: A tooltip megjelenítendő tartalma.

*event*: Az egérrel történő esemény. Az eseménykezelőbe mindig eventet kell megadni.

Visszatérítési érték: nincs.

Kezelt objektumok: A *tooltip* objektum bal felső sarkát az egér mutató koordinátaival megegyezővé teszi. Az objektum tartalma a *cont* paraméter lesz.

### **mouseOver(pol)**

Bemeneti paraméterek:

*pol*: A kezelendő poligon azonosítója.

Visszatérítési érték: nincs

Kezelt objektumok: Az *eu* objektumon (*europa.svg*) a *pol*ban megadott poligon kitöltését megcseréli sötétlilára (4B007F).

### **mouseOut(pol)**

Bemeneti paraméterek:

*pol*: A kezelendő poligon azonosítója.

Visszatérítési érték: nincs.

Kezelt objektumok: Az *eu* objektumon (*europa.svg*) a *pol*-ban megadott poligon kitöltését megcseréli világos lilára (7E45B3)

### **onClick(tk)**

Bemeneti paraméterek:

*tk*: Annak a térképnek a neve, amelyet a függvény fel kíván hozni.

Visszatérítési érték: nincs

Kezelt objektumok: A *tk* paraméterben megadott azonosítóval rendelkező objektumot láthatóvá teszi. Az *otk* globális változóban megadott objektumot láthatatlanná teszi.

Globális változók kezelése: Az *otk* globális változó értékét a *tk* paraméter értékével teszi megegyezővé.

### **mouseOverg(pol)**

Bemeneti paraméterek:

*pol*: A kezelendő poligon azonosítója.

Visszatérítési érték: nincs.

Kezelt objektumok: A *de* nevű objektumon (*germany.svg*) a *pol* paraméterben megadott poligon kitöltését sötétlilára (4B007F) cseréli.

### **mouseOutg(pol)**

Bemeneti paraméterek:

*pol*: A kezelendő poligon azonosítója.

Visszatérítési érték: nincs.

Kezelt objektumok: A *de* nevű objektumon (*germany.svg*) a *pol* paraméterben megadott poligon

kitöltését világos lilára (7E45B3) cseréli.

### **mouseOverVMT(vmkod)**

Bemeneti paraméterek:

*vmkod*: A megadott vármegye kódja (lásd: IX A vármegyék kódjai című mellékletet).

Visszatérítési érték: nincs.

Kezelt objektumok: A *telmap1* objektumon a *vmkod* értékével megegyező azonosítóval rendelkező objektum kitöltését sötétvörösre (8A0000) cseréli, ha a *vmkod* paraméter megegyezik a *selectedvm* globális változó értékével, minden egyéb esetben barnára cseréli (FEEBC5).

### **mouseOutVMT(vmkod)**

Bemeneti paraméterek:

*vmkod*: A megadott vármegye kódja (lásd: IX A vármegyék kódjai című mellékletet).

Visszatérítési érték: nincs.

Kezelt objektumok: A *telmap1* objektumon a *vmkod* értékével megegyező azonosítóval rendelkező objektum kitöltését vörösre (BB0000) cseréli, ha a *vmkod* paraméter megegyezik a *selectedvm* globális változó értékével, minden egyéb esetben világos barnára cseréli (F0E6D2).

### **mouseClickVMT(vmkod)**

Bemeneti paraméterek:

*vmkod*: A megadott vármegye kódja (lásd: IX A vármegyék kódjai című mellékletet).

Visszatérítési érték: nincs.

Kezelt objektumok: A *telmap1* objektumon a *vmkod* értékével megegyező azonosítóval rendelkező objektum kitöltését sötétvörösre (8A0000) cseréli. Ha a *selectedvm* globális változónak van értéke, akkor annak értékével megegyező azonosítóval rendelkező poligon kitöltését világosbarnára (F0E6D2) cseréli. A *telnev* objektum értéke a *szhely.php* visszatérítési értéke lesz, az *escape*

karakterek eltávolítása után, és a *vm* kezdeti paraméter meg kell egyezzen a *vmkod* értékével. Az *opthu* nevű objektum *selected* attribútuma megkapja a „selected” értéket. Ezek után lefuttatja a *keres* függvényt.

Globális változók kezelése: A *selecetedvm* globális változó értéke a *vmkod* paraméter értéke lesz.

### **mouseOverVM(vmkod)**

Bemeneti paraméterek:

*vmkod*: A megadott vármegye kódja (lásd: IX A vármegyék kódjai című mellékletet).

Visszatérítési érték: nincs.

Kezelt objektumok: A *mo1914* (*hungary\_1914.svg*) nevű objektum a *vmkod*-dal megegyező azonosítóval rendelkező poligonjának a kitöltését megcseréli. A *vm* globális tömb *vmkod* értékével megegyező nevű indexének értékét vizsgálja. Ha az „m”, akkor barnára (FEEBC5); ha „e”, akkor sötétvörösre (8A0000); ha „r”, akkor sötétzöldre (04561D), ha pedig „t” akkor sötétkékre (07169C) cseréli a kitöltését.

### **mouseOutVM(vmkod)**

Bemeneti paraméterek:

*vmkod*: A megadott vármegye kódja (lásd: IX A vármegyék kódjai című mellékletet).

Visszatérítési érték: nincs.

Kezelt objektumok: A *mo1914* (*hungary\_1914.svg*) nevű objektum a *vmkod*-dal megegyező azonosítóval rendelkező poligonjának a kitöltését megcseréli. A *vm* globális tömb *vmkod* értékével megegyező nevű indexének értékét vizsgálja. Ha az „m”, akkor világosbarnára (F0E6D2); ha „e”, akkor vörösre (BB0000); ha „r”, akkor világoszöldre (147C34), ha pedig „t”, akkor kékre (1C2FD8) cseréli a kitöltését.

### **mouseClickVM(vmkod)**

Bemeneti paraméterek:

*vmkod*: A megadott vármegye kódja (lásd: IX A vármegyék kódjai című mellékletet).

Visszatérítési érték: nincs.

Kezelt objektumok: A *mo1914* (*hungary\_1914.svg*) nevű objektum a *vmkod*dal megegyező azonosítóval rendelkező poligonjának a kitöltését megcseréli. A *vm* globális tömb *vmkod* értékével megegyező nevű indexének értékét vizsgálja. Ha az „m”, akkor sötétkékre (07169C); ha „t”, akkor sötétzöldre (04561D); ha „r”, akkor sötétvörösre (8A0000); ha pedig „e”, akkor barnára (FEEBC5) cseréli a kitöltését.

Globális változók kezelése: A *vm* globális tömb a *vmkod* paraméterrel megegyező nevű indexének értékét cseréli meg. Ha annak kezdeti értéke „m”, akkor „t”-re cseréli; ha „t”, akkor „r”-re; ha „r”, akkor „e”-re; ha pedig „e”, akkor „m”-re.

## **keres()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: Az *ered* nevű objektumba a *telnevq.php* visszatérítési értékét teszi.

A *telnevq.php* bemenetei paraméterei:

*telnev*: a *telnev* objektum értéke escape karakterekkel kiegészítve.

*nyelv*: a *nyelv* nevű objektum kiválasztott indexének értéke.

*lang*: a *lang* globális változó értéke.

*vmkod*: a *selecetedvm* globális változó értéke.

Egy xml állományt kap a *vm.php* visszatérítési értékéből, amelynek a *vmkod* kezdeti paraméter értékének a *vmkods* objektum értékét adja meg. A *vmnev* objektum értékének az xml állomány *vmnev* attribútumának értékét adja meg, escape karakterek nélkül.

## **keresvm()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.



Kezelt objektumok: A *vm.php*-tól kap egy xml állományt. A *vm.php*-nak bemeneti paramétere a *vmnev*, ami megegyezik a *vmnev* objektum értékével escape karakterekkel kiegészítve. A *telnev* objektum értéke az xml állomány *vmszekhely* attribútumának értéke lesz az escape karakterek nélkül. A *nyelv* objektumban az *opthu* opciót fogja kiválasztani (magyar).

Globális változók kezelése: A *selectedvm* az xml állománynak *vmkod* nevű attribútumának értéke lesz.

### **vmres()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: A *mo1914* nevű objektum (*hungary\_1914.svg*) minden poligonjának világosbarna (F0E6D2) kitöltést ad. Ezt a listát a *vm.php* kezdeti paraméterek nélküli futtatásával kapja.

Globális változók kezelése: A *vm* tömb minden elemének értékét „m”-re állítja.

### **vmall()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: A *mo1914* nevű objektum (*hungary\_1914.svg*) minden poligonjának kék (1C2FD8) kitöltést ad. Ezt a listát a *vm.php* kezdeti paraméterek nélküli futtatásával kapja.

Globális változók kezelése: A *vm* tömb minden elemének értékét „t”-re állítja.

## **natkeres()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: Kiköti, hogy a *minnat* és *maxnat* objektumok értéke csak szám lehet, és ne lehessen kisebb mint 0, és ne lehessen nagyobb mint 100. Tehát ha a felhasználó nem ezek között a keretek között adja meg a beviteli értékeket, nem fut tovább a függvény. Ezen túl azt is kiköti, hogy a *minnat* értéke legyen kisebb, mint a *maxnat* értéke. Utána egy xml állományt küld a *vmkeresered.php*, amelynek bemeneti paraméterei a következők:

*maxnat*: a *maxnat* objektum értéke,

*minnat*: a *minnat* objektum értéke,

*nemzet*: a *knemzet* objektum kiválasztott indexének értéke,

*feltetel*: a *feltetel* objektum kiválasztott indexének értéke.

Ha minden mező ki van töltve, akkor az xml állomány *vm* nevű attribútumának értékével megegyező azonosítóval rendelkező poligonokat a *mo1914* azonosítójú objektumban (*hungary\_1914.svg*) megcseréli. A szín attól függ, hogy a *felt* nevű attribútumnak mi az értéke. Ha ennek értéke „t”, akkor kékre (1C2FD8) cseréli az érintett poligonok kitöltését; ha „r”, akkor zöldre (147C34); ha „e”, akkor vörösre (BB0000); ha pedig „m” akkor világosbarnára (F0E6D2).

## **updateUkAt()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: A *nagymouk* (*hungary\_1914\_noleg.svg*) objektum minden poligonját fehérrel tölti ki, majd a *vmkeresered.php*-től kér egy xml állományt, amelynek bemeneti paraméterei a következők:

*maxnat*: a *maxnat* objektum értéke,

*minnat*: a *minnat* objektum értéke,

*nemzet*: a *knemzet* objektum kiválasztott indexének értéke,

*feltetel*: a *feltetel* objektum kiválasztott indexének értéke.

Ezek után minden olyan poligonnak, amelynek azonosítója szerepel, az xml állomány *vm*

attribútumában, megcseréli a kitöltését. Ha az xml állomány felt attribútuma „t”, akkor kékre (1C2FD8) cseréli a kitöltést; ha „r”, akkor zöldre (147C34); ha „e”, akkor vörösre (BB0000); ha pedig „m”, akkor világosbarnára (F0E6D2) .

### **mouseoverStatIcon(icon)**

Bemeneti paraméterek:

*icon*: Az ikon, amihez kapcsolódó statisztikai térképet nagyban akarom megjeleníteni.

Visszaadási érték: nincs.

Kezelt objektumok: Az *icon* értékének megfelelő azonosítóval rendelkező objektumot megjeleníti. A *statelol* globális változó értékével azonos azonosítóval rendelkező objektumot elrejt.

Globális változók kezelése: A *statelol* értéke az *icon* paraméter értéke lesz.

**statintchange(sel)**, ugyanaz mint a **stattk(sel)**

### **mouseoverRow(row)**

Bemeneti paraméterek:

*row*: A sor, amit ki akarunk jelölni. Az eseménynél a *this* kulcsszóval definiáljuk.

Visszaadási érték: nincs.

Kezelt objektumok: A *row*-ban megadott objektum háttérszínét lilára cseréli (31016F).

### **mouseoutRow(row)**

Bemeneti paraméterek:

*row*: A sor, amit ki akarunk jelölni. Az eseménynél a *this* kulcsszóval definiáljuk.

Visszaadási érték: nincs.

Kezelt objektumok: A *row*-ban megadott objektum háttérszínét színtelenre változtatja.

## clickRow(id,sorszam,kepszam,kepnev,intezet,tk)

Bemeneti paraméterek:

*id*: A sor objektum azonosítója.

*sorszam*: A térkép sorszáma a *katalogus\_bov* tábla sorszám mezőjével egyezik meg.

*kepszam*: A térképműben lévő szelvények száma. A *katalogus\_bov* tábla *kepszam* mezőjével egyezik meg.

*kepnev*: A kép állományneve kiterjesztés és szelvényszám nélkül. A *katalogus\_bov* tábla *filenev* mezőjével megegyezik.

*intezet*: A térképet őrző intézet kódja. Lásd: X. Intézetek ahonnan térképeket gyűjtöttem című melléklet

*tk*: A térkép neve, amelyen a térképet tartó intézet meg van jelölve.

Visszaadási érték: nincs.

Kezelt objektumok: Ha a *kepszam* > 1, akkor az *mg1* és az *mg3* objektumok tartalmát törli.

Az *mg2* tartalma a következő lesz: „<TABLE>....*belső tartalom*....</TABLE>

ahol *belső tartalom* a következő:

– az első alkalom előtt, amikor a sort beilleszti, elé kerül egy <TR>

```
<TD><INPUT type="radio" name="kistk" id="kistki"
```

```
onclick="seltkv(i,sorszám,'kepnev_i','intezet','tk');"/></TD><TD><IMG
```

```
src="kkep.php?image=kepnev_i.gif&int=intezet"
```

```
onclick="seltkv(i,sorszám,'kepnev_i','intezet','tk'); document.getElementById('kistki').checked = 'checked';"/></TD>
```

– minden második sor után egy </TR><TR>-t illeszt be, továbbá az utolsó sor után egy </TR> -t.

Ennek az eredménye egy két oszlopos táblázat, amiben elhelyezi a térképszelvények előnézeti képeit, és mellé gombokat tesz, amelyekkel aktívvá lehet tenni őket. A képek is érzékenyek lesznek a kattintásra.

Ha rákattintunk az egyikre, akkor a *seltkv* függvényt hívja fel. A fenti sorban a dőlt betűkkel írt részek a következőt jelentik:

*i*: A szelvény száma.

*sorszám*: A térkép sorszáma, azaz a *sorszam* paraméter értéke.

*kepnev*: A térkép raszteres állományának állományneve kiterjesztés és szelvényszám nélkül.

Megegyezik a *filenev* mezővel az *katalogus\_bov* táblába.

*intezet*: A térképet őrző intézet kódja.

*tk*: A térkép, amelyen az intézet helye meg van jelölve.

Ha a *kepszam* paraméter 1, akkor a *seltkv* függvényt a következőképpen futtatja le:

```
seltkv(1,sorszám,kepnév,intézet,tk)
```

ahol a *sorszám* a térkép sorszáma, azaz a *sorszam* paraméter értéke; a *kepnév* a térkép raszteres állományának állományneve kiterjesztés és szelvényszám nélkül; az *intézet* a térképet őrző intézet kódja; a *tk* pedig a térkép, amelyen a intézet helye meg van jelölve. Ebben az esetben az *mg2* tartalmát törli.

### **seltkv(kepszam, sorszam, kepnév, intézet, tk)**

Bemeneti paraméterek:

*kepszam*: A térképműben lévő szelvények száma. A *katalogus\_bov* tábla *kepszam* mezőjével egyezik meg.

*sorszam*: A térkép sorszáma a *katalogus\_bov* tábla *sorszám* mezőjével egyezik meg.

*kepnév*: A kép állományneve kiterjesztés nélkül, de szelvény számmal. A *katalogus\_bov* tábla *filenev* mezőjével megegyezik.

*intezet*: A térképet őrző intézet kódja.

*tk*: A térkép neve, amelyen a térképet tartó intézet meg van jelölve.

Visszaadási érték: nincs.

Kezelt objektumok: Az *mg1* tartalmát a következőre cseréli:

```
„<IMG src=„kep.php?filename=kepnév.jpg&int=intézet” onmouseover=mouseOverIcon(this);”  
onmouseout=„mouseOutIcon(this);” onclick=„bigmapper('képnév','intézet','sorszám','képszám');”  
border=„2” style=„border-color:transparent;”/>”
```

Ezzel az *mg1*-ben létrehoz egy nagyobbított előnézetet, egy 2 pixeles szintelen kerettel. Ha az egeret fölé viszem, akkor a keret piros lesz, ha meg kiviszem fölüle az egeret, akkor ismét szintelenné válik. Ha rákattintok lefuttatja a *bigmapper* függvényt az alább megadott paraméterekkel.

Az *mg3* tartalmát a következőre cseréli:

```
„<INPUT type=„button” value=„felirat” onclick=„adatok(sorszám,intézet,tk,kepnév);”/>”
```

Ezzel létrehoz egy gombot, aminek a felirata a *felirat* függvény visszatérítési értéke lesz az 'e115' bemeneti paraméter szerint. A gombra való kattintás az alábbi paraméterekkel elindítja az *adatok* függvényt.

*sorszám*: A térkép sorszáma, azaz a *sorszam* paraméter értéke.

*képnév*: A térkép raszteres állományának állományneve kiterjesztés és szelvénytípus nélkül.

Megegyezik a *filenev* mezővel az *katalogus\_bov* táblában.

*intézet*: A térképet tartó intézet kódja.

*tk*: A térkép, amelyen az intézet helye meg van jelölve.

*képszám*: A kiválasztott szelvény száma.

## **vmkeres()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: Előbb a *vm.php*-től visszacapott eredmény alapján létrehozza a *vmkeres.php* leendő kezdeti paramétereit. A *vmkeres.php* lang paraméterének értéke a *lang* globális változó lesz. Ezek után minden a *vm.php*-től visszacapott *vmkod* értékre létrehoz egy paramétert a *vmkeres.php* számára, és az értékének a *vm* tömb, *vmkod* értékével megegyező indexű elem értékét adja meg. Ezekkel a kezdeti paraméterekkel lefuttatott *vmkeres.php* eredményét a *searchres* objektumba helyezi. A *cont*, *germany*, *hungary*, *bw*, *vmkereso* és *maptitle* objektumokat elrejt. Az *mg1*, *mg2* és *mg3* objektumok tartalmát törli. A *searchres* objektumot megjeleníti. A *res* objektumon szükség esetén megjeleníti a görgetősávokat.

## **keresinp ()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: A *keres.php* eredményét a *searchres* objektumba helyezi. A következő paraméterekkel futtatja le a *keres.php*-t:

*lang*: A *lang* globális változó értéke.

*keres\_inp*: A *keres\_inp* objektum értéke.

*keres\_inp1*: A *keres\_inp1* objektum értéke.

*keres\_inp2*: A *keres\_inp2* objektum értéke.

*keres\_inp3*: A *keres\_inp3* objektum értéke.

*keres\_inp4*: A *keres\_inp4* objektum értéke.

*keres\_inp5*: A *keres\_inp5* objektum értéke.  
*keres\_inp6*: A *keres\_inp6* objektum értéke.  
*keres\_inp7*: A *keres\_inp7* objektum értéke.  
*mezo*: A *mezo* objektum kiválasztott indexének értéke .  
*mezo1*: A *mezo1* objektum kiválasztott indexének értéke.  
*mezo2*: A *mezo2* objektum kiválasztott indexének értéke.  
*mezo3*: A *mezo3* objektum kiválasztott indexének értéke.  
*mezo4*: A *mezo4* objektum kiválasztott indexének értéke.  
*mezo5*: A *mezo5* objektum kiválasztott indexének értéke.  
*mezo6*: A *mezo6* objektum kiválasztott indexének értéke.  
*mezo7*: A *mezo7* objektum kiválasztott indexének értéke.  
*kapcs1*: A *kapcs1* objektum kiválasztott indexének értéke.  
*kapcs2*: A *kapcs2* objektum kiválasztott indexének értéke.  
*kapcs3*: A *kapcs3* objektum kiválasztott indexének értéke.  
*kapcs4*: A *kapcs4* objektum kiválasztott indexének értéke.  
*kapcs5*: A *kapcs5* objektum kiválasztott indexének értéke.  
*kapcs6*: A *kapcs6* objektum kiválasztott indexének értéke.  
*kapcs7*: A *kapcs7* objektum kiválasztott indexének értéke.  
*lelohely*: A *lel* objektum kiválasztott indexének értéke.

A *cont*, *germany*, *hungary*, *bw*, *vmkereso* és *maptitle* objektumokat elrejt. Az *mg1*, *mg2* és *mg3* objektumok tartalmát törli. A *searchres* objektumot megjeleníti. A *res* objektumon szükség esetén megjeleníti a görgetősávokat.

### **adatok(sorszam,intezet,tk,kepnev)**

Bemeneti paraméterek:

*sorszam*: A vizsgált térkép sorszáma.

*intezet*: A térképet tartó intézet kódja.

*kepnev*: A térkép állományneve kiterjesztés nélkül, de szelvény számmal. A *katalogus\_bov* tábla *filenev* mezőjével megegyezik.

*tk*: a térkép neve amin az intézet helye be van jelölve.

Visszaadási érték: nincs.

Kezelt objektumok: lefuttatja *detfoton()*; függvényt. Ha a *tk* paraméter értéke 'hungary', akkor

elrejt a *detmenucontbw* nevű objektumot, és megjeleníti a *detmenuconthungary* objektumot. Ha a *tk* paraméter értéke 'bw', akkor a *detmenuconthungary* objektumot rejt el, és a *detmenucontbw* objektumot jeleníti meg. Ezek után az *archive.php* segítségével megállapítja, hogy melyik poligon ábrázolja azt a helyet, ahol a megjelölt intézet van, és megcseréli annak a poligonnak a kitöltését vörösre (990000). Ugyanezt megteszi a menüben lévő ikonként. Ezek után a *vmsz.php* segítségével a megadott sorszámú térképre megszerzi azt, hogy mely vármegyét ábrázolja egészben, illetve részben, és melyeket nem ábrázolja. A *vmsz.php*-nek a *tkszam* paraméterébe beadja a *sorszam* paraméter értékét. Ezek után visszakap két attribútumot az xml állományba. A *vm* attribútum a vármegye kódját tartalmazza; míg a *tart* attribútum egy „t” betűt, ha az adott vármegyét a térkép egészben ábrázolja; egy „r” betűt, ha csak részben; és egy „e” betűt, ha nem ábrázolja. Ha ábrázolja, akkor a vármegye poligont, mind a nagy térképen, mind az ikon térképen kitölti kékkel (1C2FD8); ha csak részben ábrázolja, akkor zölddel (147C34) tölti ki; ha meg nem ábrázolja, akkor vörössel (BB0000).

A *detcontvmcim* objektum tartalma a *felirat* függvény visszatérítési értéke lesz, a kezdeti paramétere 'el43'

A *defoto* objektum tartalma a következő lesz: „<BR/><IMG src=’foto.php?img=intezet’ width=’100%’/>”. Ez megjeleníti az intézet képét a *defoto* objektumban. Az *intezet* az intézet kódját jelenti, ami a függvény *intezet* paraméterében van. A *detres* objektumba a *detres.php* visszatérési értékét teszi. A *detres.php*-nek a következő kezdeti paramétereket kell megadni:

*lang*: Megegyezik a *lang* globális változó értékével.

*int*: Megegyezik az *intezet* paraméter értékével.

*sorszam*: Megegyezik a *sorszam* paraméter értékével.

A *detmenufoto* objektumba a következőt helyezi: <IMG src=’foto.php?img=intezet’ onmouseover=’defoton();’ height=’100px’/>

Ez egy 100 pixel magasságban megjeleníti a *detmenufoto* objektumban az intézet képét.

A *detmenutk* objektumba a következőt helyezi: <IMG src=’kkep.php?int=intezet&image=kepnev.gif’ onmouseover=’detmapper();’/>

Ezzel megjeleníti a térkép kis előnézetét a *detmenutk* objektumban, és ha fölé viszem az egeret, lefuttatja a *detmapper()* függvényt.

*intezet*: A térképet őrző intézet kódja, amely megegyezik a függvény *intezet* paraméterével.

*kepnev*: A raszteres állomány neve, ahogy a *kepnev* paraméterben meg van adva.

Majd ezek után a *mapper.php* eredményét a *detmapper* objektumba helyezi. A *mapper.php*-t a következő paraméterekkel futtatja le:

*maxh*: 480,

*maxw*: 500,



*lang*: *lang* globális változó értéke,  
*int*: *intezet* paraméter értéke,  
*filename*: *kepnev* paraméter értéke+”.jpg”.

Ezek után lefuttatja a *preoverlay* függvényt, majd láthatóvá teszi az adatok objektumot.

### **bigmapper(*kepnev,intezet,sorszam,kepszam*)**

Bemeneti paraméterek:

*kepnev*: A térkép állomány neve, szelvény számmal, de kiterjesztés nélkül.

*intezet*: A térképet tartó intézet kódja.

*sorszam*: A térkép sorszáma.

*kepszam*: A térképművön belül a térkép szelvényének a száma.

Visszaadási érték: nincs.

Kezelt objektumok: A *bigmap* objektumba helyezi a *mapper.php* eredményét, a következő paraméterekkel: *maxh*=600, *maxw*=1100, *int*=*intezet*, *filename*=*képnév.jpg*, *lang*= *lang* globális változó értéke, *winW*=*winW* globális változó értéke, *winH*=*winH* globális változó értéke, *sorszam*=*sorszám*, *kepszam*=*képszám*.

Ahol

*képnév*: A térkép állomány neve, ami megegyezik a *kepnev* paraméterrel.

*intezet*: A térképet őrző intézet kódja, ami megegyezik a függvény *intezet* paraméterének értékével.

*sorszám*: A térkép sorszáma, ami megegyezik a függvény *sorszam* paraméterének értékével.

*képszám*: A térkép szelvényének a száma, ami megegyezik a függvény *kepszam* paraméterének értékével.

Ezzel megjeleníti nagyba a térképnézetet a popup ablakban.

### **defoton()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: A *detfoto* objektumot láthatóvá teszi, a *detconthungary*, *detcontbw*, *detcontvm* és *detmapper* objektumokat pedig elrejt.

### **detmapper()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: A *detmapper* objektumot láthatóvá teszi, a *detconthungary*, *detcontbw*, *detcontvm* és *detfoto* objektumokat pedig elrejt.

### **dethungarys()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: A *detconthungary* objektumot láthatóvá teszi, a *detmapper*, *detcontbw*, *detcontvm* és *detfoto* objektumokat pedig elrejt.

### **detbws()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: A *detcontbw* objektumot láthatóvá teszi, a *detmapper*, *detconthungary*, *detcontvm* és *detfoto* objektumokat pedig elrejt.

### **detvms()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: A *detcontvm* objektumot láthatóvá teszi, a *detmapper*, *detconthungary*, *detcontbw* és *detfoto* objektumokat pedig elrejt.

### **prev(ck)**

Bemeneti paraméterek:

*ck*: A checkbox státusza.

Visszaadási érték: nincs.

Kezelt objektumok: Ha a checkbox státusza „checked”, akkor megjeleníti az áttekintőt a térkép böngészőn (kivéve, ha az egész térkép látszik a megadott nagyításban). Ha nem „checked” a státusza, akkor elrejt az áttekintőt.

### **getQueryVariable(variable)**

Bemeneti paraméterek:

*variable*: A változó neve, amely meg van adva a böngésző ablak címsorában.

Visszaadási érték: A *variable* paraméterben megadott get változó értéke.

### **felirat(el)**

Bemeneti paraméterek:

*el*: A feliratot a nyelvek táblában tartalmazó mező.

Visszaadási érték: A felirat tartalma, escape karakterek nélkül, azaz *felirat.php* visszatérítési értéke, ha a bemeneti paraméterek:

*lang*: a *lang* globális változó értéke.

*el*: az *el* paraméter értéke.

## **vissza()**

Bemeneti paraméterek: nincs.

Visszaadási érték: Ha sikeres, igaz.

Kezelt objektumok: A *mapper*, *mapperclose*, *mapviewer*, *telszocont*, *nepterkep*, *statterkep*, *adatok*, *bigmap*, *forras*, *intinfo* és *szdata* objektumokat elrejt.

## **vmtk(vmtk)**

Bemeneti paraméterek:

*vmtk*: A megjeleníteni kívánt vármegyetérkép vármegye kódja.

Visszaadási érték: nincs.

Kezelt objektumok: Az *oldmap* objektumba behelyezi a *mapper.php* eredményét, ahol a paraméterei a következők: *lang*= a *lang* globális változó értéke, *maxh*=480, *maxw*=600, *vm*=*vmkod*, ahol *vmkod* a *vmtk* paraméterben megadott vármegye kód.

Ez megjelenít az *oldmap* objektumban egy térképböngészőt, amibe a Gönczy-féle vármegye térkép sorozat megfelelő vármegyéje jelenik meg. Ezek után az *oldmap* tartalmához még hozzáadja a következőt:

```
„<BR/><CENTER><INPUT type=’button’ value=’felirat’ onclick=’retelter();’/></CENTER>”
```

ahol

*felirat*: A *felirat* függvény visszatérítési értéke, ha a bemeneti paraméter 'el79'.

A *telmap* objektumot elrejt, míg az *oldmap* objektumot megjeleníti.

## **detvms()**

Bemeneti paraméterek: nincs.

Visszaadási érték: nincs.

Kezelt objektumok: Az *oldmap* objektumot elrejt, míg a *telmap* objektumot megjeleníti.

## **infupdate(kat)**

Bemeneti paraméterek:

*kat*: A megtekinteni kívánt intézet.

Visszaadási érték: nincs.

Kezelt objektumok: Az *infres* objektumba helyezi az *intinfores.php* eredményét. Az *intinfores.php* bemeneti paraméterei a következők: *lang*: a *lang* globális változó értéke, *int*: a *kat* paraméter értéke. Szükség esetén megjeleníti az *infres* objektumon a görgetősávokat.

## **vmszam()**

Bemeneti paraméterek:

*kat*: A megtekinteni kívánt intézet.

Visszaadási érték: nincs.

Kezelt objektumok: A *vm.php* szerver oldali szkripttől visszakapja az összes vármegye kódját. A *vmszam.php* szerver oldali szkriptnek elküldi a *vm* globális tömb összes értékét, amely nem „m”, mint azonos nevű paramétert. A visszakapott eredmény egy szám, ami azt mutatja, hogy a vármegyés keresőben a *vm* tömb pillanatnyi állása szerint hány térképet találna. Az *mg3* objektumban a *felirat(„el102”)* függvény, majd a *vmszam.php* visszatérítési értékét teszi egymás után.

## **VII. Szerver oldali függvények**

### **adatok.php**

Bemeneti paraméterek:

*dat*: Adat, amit akarunk az intézetről (lehetséges értékei: nev, cim, honlap).

*lang*: Nyelv megadása.

*archive*: Az intézet, amelyről adatot akarunk.

Kimenet típusa: text/plain.

Kimenet leírása: A kért adat egyszerű szöveg formátumú. Ha *dat=nev*, akkor az intézet nevét írja ki, ha *dat=cim* akkor az intézet postai címét, ha pedig *dat=honlap* akkor az intézet honlapját.

Függvény működése: Létesít egy kapcsolatot az adatbázis-kiszolgáló felé, és kikeresi a kért adatot. Az adatot elküldi a kimenetre, majd bezárja a kapcsolatot.

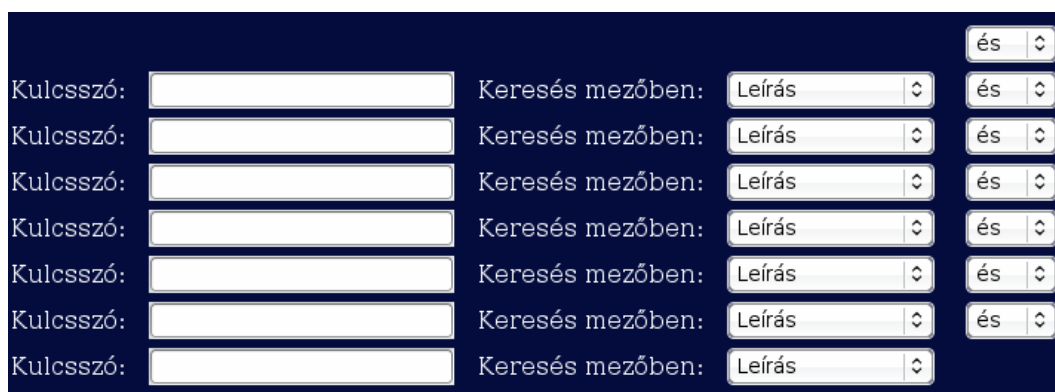
## advsearch.php

Bemeneti paraméterek:

*lang*: Nyelv megadása.

Kimenet típusa: text/html.

Kimenet leírása: Az *advsearch* objektum belső elemeit hozza létre. Egy táblázatot hoz létre, aminek az első sorában csak egy menü található, amelyben ki lehet választani, hogy a következő sor tartalma milyen kapcsolatban legyen az egyszerű kereső tartalmával. A további sorokban van kereső mező, és ugyanígy egy menü, amely megadja, hogy az adott sor milyen kapcsolatban legyen az előző sor tartalmával.



The screenshot shows a dark blue background with a grid of search fields. Each row consists of a 'Kulcsszó:' (Keyword) input field, a 'Keresés mezőben:' (Search in) dropdown menu with 'Leírás' (Description) selected, and an 'és' (and) dropdown menu. There are seven rows in total, with the top row having an additional 'és' dropdown menu on the far right.

*Az advsearch.php által létrehozott objektumok*

A szöveges mezők azonosítói mind *keres\_inpx* azonosítóval rendelkeznek, ahol *x* egy szám egytől hétig. A felső szöveges mező tehát *keres\_inp1*, alatta *keres\_inp2* stb. A keresés mezőben feliratú mezők azonosítói *mezox*, ahol ugyanígy *x* egy szám egytől hétig és a felső *mezo1*, alatta *mezo2* stb. A kapcsolatot jelző mezők azonosítói *kapcsx*, ahol *x* egy szám egytől hétig és a legfelső mező *kapcs1*, alatta *kapcs2*. Fontos, hogy ezek az elemek egy sorral feljebb vannak csúsztatva, mint a tőlük balra eső elemek.

		<b>kapcs1</b>
<b>keres_inp1</b>	<b>mezo1</b>	<b>kapcs2</b>
<b>keres_inp2</b>	<b>mezo2</b>	<b>kapcs3</b>
<b>keres_inp3</b>	<b>mezo3</b>	<b>kapcs4</b>
<b>keres_inp4</b>	<b>mezo4</b>	<b>kapcs5</b>
<b>keres_inp5</b>	<b>mezo5</b>	<b>kapcs6</b>
<b>keres_inp6</b>	<b>mezo6</b>	<b>kapcs7</b>
<b>keres_inp7</b>	<b>mezo7</b>	

*Objektumok elhelyezése és nevei az advsearch objektumon belül*

Az első oszlopban lévő objektumok szövegbevitelre alkalmas szövegdobozok. Az értékük megegyezik a bevitt szöveggel. A második oszlopba lévő objektumok menük. Lehetséges értékei megegyeznek a *katalogus\_bov* tábla mezőneveivel, amelyekbe keresni lehet. A következő értékei lehetnek: *leiras, szerzo, kiado, kiadas\_helye, kiadas\_eve, terkep\_nyelve, meretarany, jelzet*. A harmadik oszlopban lévő objektumok lehetséges értékei *OR* és *AND*. Ez vagy, illetve és kapcsolatot jelez a mezők között.

## **archives.php**

Bemeneti paraméterek:

*terk*: A térkép, amelyen lévő levéltárakat akarjuk az eredményben.

*lel*: Az a levéltár, amelynek helyét akarjuk.

Kiement típusa: text/xml.

Kimenet leírása: A bemeneti paraméterek közül csak egyet kell beállítani. Ha a „*terk*” paraméter van beállítva, akkor az adott térképen lévő levéltárakat listázza ki. Jelenleg a „*terk*” értéke lehet „*mo*” vagy „*bw*”. Ebben az esetben egy sor dataset xml tag fog megjelenni az eredményben. Az attribútumai a következők:

*id*: A tag sorszáma az eredménytáblában.

*archive*: A levéltár azonosítója a *Katalogusok* tábla szerint.

*polygon*: A poligon azonosítója a térképen, amelyen a levéltár van.

*gr*: A Baden-Württemberg térképen a második poligon, amely az intézet poligonját kifedi.

*terk*: A térkép neve, amelyen a levéltár(ak) van(nak).

*hely*: A város azonosítója, ahol a levéltár van.

A fenti adatokat a *terkereso* tábla megfelelő mezőiből olvassa ki.

Ha a *lel* paraméter van beállítva:

A *lel* paraméterben megadott lelőhelyre adja vissza a fentiek szerint ugyanazokat az adatokat. Ekkor a *mularch* táblából kikeresi, melyik városban van a levéltár; és azt összekapcsolja a *terkereso* táblával, és így adja vissza az adatokat.

## cimkeres.php

Bemeneti paraméterek:

*lang*: A kimenet nyelve.

*sorszam*: A keresett térkép sorszáma.

Kimenet típusa: text/plain.

Kimenet leírása: A *sorszam* paraméterben meghatározott sorszámú térkép a *lang* paraméterben meghatározott nyelven kikeresi a címét (leírását) a *catalogus\_bov* táblából, és escape karatkerekkel entityk nélkül visszaadja.

## cim.php

Bemeneti paraméterek:

*lang*: A kimenet nyelve.

Kimenet típusa: text/plain.

Kimenet leírása: A *lang* paraméterben megadott nyelven visszaadja a honlap címét entityk nélkül escape karakterekkel.

## detres.php

Bemeneti paraméterek:

*lang*: A kimenet nyelve.

*int*: Az intézet, ahol a térképet őrzik.

Térkép leírása:	Magyarország néprajzi térképe 4 melléktérképpel és 6 kördiagrammal
Szerző	Dr. Kogutowicz Károly
Kiadó	Magyar Királyi Állami Térképészet
Kiadás helye	Budapest
Előállítás éve	1928
Méretarány	1:1000000
Méret	93x57cm
Térkép nyelve	Magyar
Lapok száma:	1
Katalógus jelzet:	B IX c 1074/2
Térképet tartó intézet neve:	Hadtörténeli Intézet és Múzeum Hadtörténeli Könyvtár és Térképtár
Térképet tartó intézet címe	1014, Budapest, Kapisztrán tér 2-4
Térképet tartó intézet honlapja	<a href="http://www.militaria.hu/hun/riklap.php?rigazon=26">http://www.militaria.hu/hun/riklap.php?rigazon=26</a>

*detres.php* eredménye  
html-be



*sorszam*: A térkép sorszáma, amelynek az adatlapja kell.

Kimenet típusa: text/html.

Kimenet leírása: Az *int* paraméterben meghatározott intézetben lévő *sorszam* paraméterben meghatározott sorszámú térkép adatlapját a *lang* paraméterben meghatározott nyelven adja. Az eredményt a popup ablakban lévő *adatok* objektumban lévő *detres* objektumban jeleníti meg.

### **felirat.php**

Bemeneti paraméterek:

*lang*: A kimenet nyelve.

*el*: Az elem a nyelvek táblából, amelynek tartalmát szeretnénk.

Kimenet típusa: text/plain.

Kimenet leírása: A *nyelvek* táblából visszaadja az *el* paraméterben lévő a *lang* paraméterben lévő nyelvnek megfelelő értéket. A honlap szöveges elemeinek az adatbázisból történő lekéréséhez használatos.

### **forras.php**

Bemeneti paraméterek:

*lang*: A kimenet nyelve.

Kimenet típusa: text/html.

Kimenet leírása: A *lang* paraméterben megadott nyelven egy táblázatot ad vissza, amiben a honlapon felhasznált adatok forrása jelenik meg. Az eredményt a popup ablakban lévő *forras* objektumba helyezi. Az eredmény fent tartalmaz egy feliratot az alatta lévő adatokhoz, majd egy táblázatot. A táblázat tartalmát az adatbázis *forras* táblájából keresi ki. A bal oszlop a *forras* tábla megnevezés mezője, míg a jobb oszlop a *jeloles* mezője. A sorok száma megegyezik az adott nyelven lévő bejegyzések számával.

## **foto.php**

Bemeneti paraméterek:

*img*: Az intézmény kódja, amelynek a fényképét meg akarjuk jeleníteni.

Kimenet típusa: image/jpeg.

Kimenet leírása: Az *img* paraméterben megadott intézet képét jeleníti meg a kimeneten. Fontos, hogy a kimenet egy kép megjelenítésére alkalmas kimenet legyen, pl. a html nyelv *img* tag-je. A következőképpen lehet rá hivatkozni:

```
<IMG src="foto.php?img=blb"/>
```

A fenti példa a Badeni Tartományi Könyvtár képét jeleníti meg.

## **ger.php**

Bemeneti paraméterek: nincs.

Kiement típusa: text/xml.

Kimenet leírása: Egy xml datasetet hoz létre a *germany* alapján azokról a poligonokról, amelyek a Németország térképen aktívak kell legyenek. Az attribútumai a következők:

*bland*: A szövetségi állam, amelynek a poligonja aktív kell legyen.

*polygon*: A poligon azonosítója. Legtöbb esetben megegyezik a szövetségi állam kódjával.

*tk*: Az a térkép, amely meg kell jelenjen, ha rákattint a felhasználó az adott poligonra.

## **getcim.php**

Bemeneti paraméterek:

*sorszam*: A térkép sorszáma, amelynek az állománynevét akarjuk.

*lang*: A *lang* változó értéke a kliens oldalról.

*kepszam*: Ha többszelvényes műről van szó, akkor a szelvény száma, amelynek az állománynevét akarjuk. (Egylapos térképnél nem kell megadni értéket.)

Kimenet típusa: text/plain.

Kimenet leírása: A *sorszam* paraméterben megadott térkép állománynevét adja vissza. Ha több szelvénye is van a műnek, akkor a *kepszam* paraméterben megadott szelvényszámú szelvény állománynevét adja vissza.

## intinfo.php

Bemeneti paraméterek:

*lang*: A kimenet nyelve.

Kimenet típusa: text/html.

Kimenet leírása: A popup ablakba készít tartalmat az *intinfo* objektumba. Fent felírja a popup címét, majd alatta egy táblázat helyezkedik el, amelynek egy sora és két oszlopa van. A bal oldali oszlopban egy felirat van, a jobb oldaliban egy menü, amiben az intézeteket lehet kiválasztani. Alatta egy másik táblázat van, amelynek szintén egy sora és két oszlopa van. A bal oldali oszlopban az *intinfores.php* által visszaadott eredmény van, míg a jobb oldali oszlopban az intézet fényképe.

Intézet neve	Hadtörténeli Intézet és Múzeum Hadtörténeli Könyvtár és Térképtár
Intézet címe	1014, Budapest, Kapisztrán tér 2-4
Intézet honlapja	<a href="http://www.militaria.hu/hun/oldal.php?oldal=28">http://www.militaria.hu/hun/oldal.php?oldal=28</a>
Intézetből felhasznált térképművek száma	28
Intézetből felhasznált térképszelvények száma	158

*Az intinfo.php eredménye egy popup ablakban. A bal oldalon látható táblázatot az intinfores.php generálta le*

## intinfores.php

Bemeneti paraméterek:

*lang*: A kimenet nyelve.

*int*: Az intézet, amelyről az információt akarjuk kérni.

Kimenet típusa: text/html.

Kimenet leírása: Az *int* paraméterben megadott intézet nevét, címét, honlapcímét, az intézetből felhasznált térképművek és a felhasznált szelvények számát helyezi egy táblázatba (lásd a fenti ábrán).

## **kep.php**

Bemeneti paraméterek:

*int*: Az intézet, ahol a megtekinteni kívánt térkép van.

*filename*: A megtekinteni kívánt térkép teljes állományneve szelvény számmal és kiterjesztéssel.

Kimenet típusa: image/jpeg.

Kimenet leírása: Egy maximum 300x200px méretű képet küld a paraméterekben megadott térképről. A nagy előnézethez használatos.

## **keres.php**

Bemeneti paraméterek:

*lang*: A kimenet nyelve.

*kapcs1*: A *kapcs1* objektum kiválasztott indexének átadott értéke.

*kapcs2*: A *kapcs2* objektum kiválasztott indexének átadott értéke.

*kapcs3*: A *kapcs3* objektum kiválasztott indexének átadott értéke.

*kapcs4*: A *kapcs4* objektum kiválasztott indexének átadott értéke.

*kapcs5*: A *kapcs5* objektum kiválasztott indexének átadott értéke.

*kapcs6*: A *kapcs6* objektum kiválasztott indexének átadott értéke.

*kapcs7*: A *kapcs7* objektum kiválasztott indexének átadott értéke.

*lelohely*: Az intézet kódja, amelyben keresünk. Ha „barmi”, akkor minden intézetben keres.

A *lelohely* objektum kiválasztott indexének az értéke.

*keres\_inp*: Az első keresési mező tartalma. A *keresp\_inp* objektum értéke.

*keres\_inp1*: A második keresési mező tartalma. A *keresp\_inp1* objektum értéke.

*keres\_inp2*: A harmadik keresési mező tartalma. A *keresp\_inp2* objektum értéke.

*keres\_inp3*: A negyedik keresési mező tartalma. A *keresp\_inp3* objektum értéke.  
*keres\_inp4*: Az ötödik keresési mező tartalma. A *keresp\_inp4* objektum értéke.  
*keres\_inp5*: A hatodik keresési mező tartalma. A *keresp\_inp5* objektum értéke.  
*keres\_inp6*: A hetedik keresési mező tartalma. A *keresp\_inp6* objektum értéke.  
*keres\_inp7*: A nyolcadik keresési mező tartalma. A *keresp\_inp7* objektum értéke.  
*mezo*: A *mezo* objektum értéke.  
*mezo1*: A *mezo1* objektum értéke.  
*mezo2*: A *mezo2* objektum értéke.  
*mezo3*: A *mezo3* objektum értéke.  
*mezo4*: A *mezo4* objektum értéke.  
*mezo5*: A *mezo5* objektum értéke.  
*mezo6*: A *mezo6* objektum értéke.  
*mezo7*: A *mezo7* objektum értéke.

**Feliart keres\_inp Feliart mezo Feliart lelohely gomb**

*A search.php által generált objektumok elhelyezkedése*

		<b>kapcs1</b>
<b>keres_inp1</b>	<b>mezo1</b>	<b>kapcs2</b>
<b>keres_inp2</b>	<b>mezo2</b>	<b>kapcs3</b>
<b>keres_inp3</b>	<b>mezo3</b>	<b>kapcs4</b>
<b>keres_inp4</b>	<b>mezo4</b>	<b>kapcs5</b>
<b>keres_inp5</b>	<b>mezo5</b>	<b>kapcs6</b>
<b>keres_inp6</b>	<b>mezo6</b>	<b>kapcs7</b>
<b>keres_inp7</b>	<b>mezo7</b>	

*Objektumok elhelyezése és nevei az advsearch objektumon belül*

Kimenet típusa: text/html.

Kimenet leírása: Egy táblázat, aminek a fejlécében le van írva, hogy melyik oszlop mit tartalmaz.

Minden sor egy eleme a keresési eredménynek, és van hozzá rendelve három eseménykezelő:

*onmouseover*: mouseOverRow(this);

*onmouseout*: mouseOutRow(this);

*onclick*: clickRow(id,sorszám,képszám,képnév,intézet,térkép)

ahol:

*id* = A sorszám a keresési eredményben.

*sorszám* = A térkép sorszáma az adatbázisban.

*kepszám* = A térképszelvények száma.

*képnév* = Az állománynév szelvénytípus és kiterjesztés nélkül.

*intézet* = A térképet őrző intézet kódja.

*térkép* = A térkép neve, amelyen az intézet be van jelölve.

Az első oszlop egy rádió gombot tartalmaz, aminek az azonosítója a *tkid*, ahol *id* a sornak a száma a keresési eredményben. A neve minden rádió gombnak *tk*.

További oszlopai a táblázatnak balról jobbra:

- Előnézeti kép a térképről. Ha többszelvényes térképműről van szó, akkor egy erre utaló piktogram látható ebben az oszlopban.
- Térkép leírása.
- Szerző neve.
- Kiadó neve.
- Kiadás helye.
- Térkép méretaránya. Ha többszelvényes térképműnél nem egységes, akkor nincs megjelölve.
- Térképlapok mérete szélesség×magasság cm-ben.
- Kiadás éve.
- Levéltárban használt katalógusjelzet.
- Térképet őrző intézet neve.
- Térkép nyelve.

## **kfoto.php**

Bemeneti paraméterek:

*img*: Az intézet kódja, aminek a képét mutatja.

Kimenet típusa: image/jpeg.

Kimenet leírása: Egy kis képet mutat az intézetről.

## kkep.php

Bemeneti paraméterek:

*int*: Az intézet, ahol a megtekinteni kívánt térkép található.

*filename*: A megtekinteni kívánt térkép teljes állományneve szelvény számmal és kiterjesztéssel.

Kimenet típusa: image/gif.

Kimenet leírása: Egy maximum 100px szélességű képet küld a paraméterekben megadott térképről. A kis előnézethez használatos.

## mapper.php

Bemeneti paraméterek:

*maxw*: A térképböngésző maximum szélessége pixelben.

*maxh*: A térképböngésző maximum magassága pixelben.

*int*: A megjeleníteni kívánt térképet őrző intézet kódja.

*filename*: A térkép állományneve szelvény számmal és kiterjesztéssel.

*vm*: Ha vármegyetérképet akarok mutatni, az adott vármegye kódja. Ha a *vm*-nek érték van adva, akkor az *int* és a *filename* nem kaphat értéket.

*winw*: A teljes ablak szélessége. Ha van neki érték adva, akkor megjelenik egy gomb, amellyel meg lehet nyitni egy külön ablakban a térképnézegetőt.

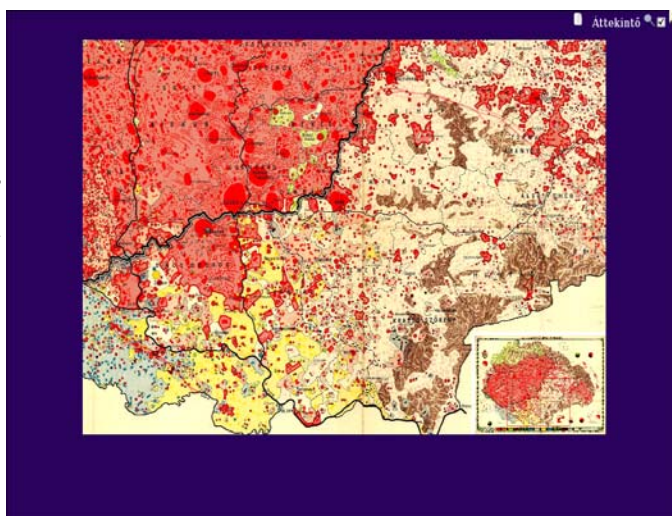
*winh*: A teljes ablak magassága.

Kimenet típusa: text/html.

Kimenet leírása: Jobbra fent létrehoz egy pipát, amivel ki-be lehet kapcsolni az előnézeti képet.

Ha a *winw*-nek van értéke, akkor ettől balra megjelenik egy kis piktogram, ami fölé ha az egérrel mutat a felhasználó, megjelenik egy tooltip; ha pedig rákattint, akkor megnyitja a térképböngészőt egy új lapon. Alatta

megjelenik a térképböngésző a *maxw* és *maxh*



A térképböngésző egy popup ablakban, áttekintővel

méretük között a lehető legnagyobb méretben. Ebben egérrel lehet húzogatni a térképet, illetve a görgetővel nagyítani. Az előnézeten lévő piros keret mozgatásával is lehet mozgatni a kivágot, valamint a nyilakkal a billentyűzeten. A Page Up és Page Down billentyűkkel úgyszintén lehet kicsinyíteni, illetve nagyítani.

## mulpict.php

Bemeneti paraméterek:

*hely*: A megadott hely kódja, ahol több mint egy levéltár van.

*lang*: A kimenet nyelve.

Kimenet típusa: text/xml.

Kimenet leírása: Egy xml dataset, ahol a *dataset* tag-nek a következő attribútumai vannak:

- *lelohely*: az intézet kódja,
- *foto*: 1, ha van kép az intézetről; 0, ha nincs,
- *tt*: az intézet rövid neve etnityk nélkül escape karakterekkel

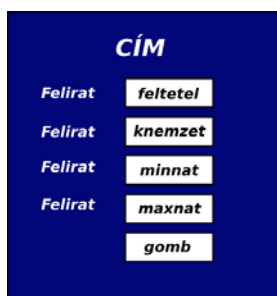
Annyi sor van, ahány intézet van az adott helyen.

## nepkeres.php

Bemeneti paraméterek:

*lang*: A kimenet nyelve.

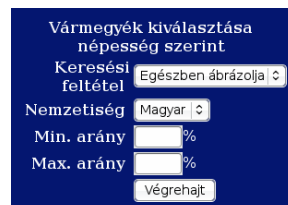
Kimenet típusa: text/html.



Objektumok elhelyezése az eredményben

Kimenet leírása: Fent egy feliratot helyez el. Alatta egy táblázatot helyez el, öt sorral és két oszloppal. A bal oldali oszlopba a jobb oldali oszlopban lévő objektumokhoz tartozó feliratokat helyezi. A jobb oldali oszlopban a következő objektumok vannak fentről lefelé:

Egy menü a keresési feltétel kiválasztására. Az objektum azonosítója: *feltétel*. Lehetséges értékei „m”, „t”, „r” és „e”. Alatta szintén egy menü található. Itt a



*nepkeres.php* eredménye



nemzetiséget lehet kiválasztani. Az azonosítója *knemzet*. Lehetséges értékei: „magyar”, „nemet”, „olah”, „tot”, „ruten”, „horvat”, „szerb”, „egyebnep”. A következő sorban egy szöveges doboz található, amibe a kiválasztott nemzetiség százalékos arányát kell beírni. Azonosítója *minnat*. Értéke egész szám lehet nullától százig. Hasonlóan az alatta lévő szöveges dobozba a kiválasztott népesség maximum arányát kell beírni. Feltétel, hogy az itt szereplő szám nagyobb legyen, mint a *minnat* objektumban. Azonosítója: *maxnat*. A legutolsó sorban egy gomb található, amire ha rákattint a felhasználó, a *natkeres* függvényt futtatja le a kliensen.

## **nepszam.php**

Bemeneti paraméterek:

*vmkod*: Annak a vármegyének a kódja, aminek a népességi adatait szeretnénk.

*nat*: Adott nemzetiség, amelynek vármegyénként akarjuk az adatait.

Kiement típusa: text/xml.

Kimenet leírása: A *vmkod* paraméterben meghatározott vármegye népszámlálási adatait adja ki, ami meg van adva. Ha nincs megadva, de a *nat* értéke egy nemzetiség, akkor az adott nemzetiség adatait adja ki. Ha értéke „mind”, akkor minden nemzetiség minden vármegyére vonatkoztatott adatait kiadja. A két paraméter közül az egyik megadása kötelező. A kimenet egy xml adathalmaz, amelyben az adott nemzetiség megnevezése lesz a hozzá tartozó attribútum, és az értéke az adott vármegyében annak lélekszáma. A *vm* attribútum a vármegye kódja. A *terulet* attribútum a vármegye területe négyzetkilométerben, a *nepesseg* attribútum pedig a vármegye teljes népessége. A nemzetiségek mind a bemenetnél, mind a kimenetnél a következő megnevezéseket kapják:

*magyar*: magyar

*nemet*: német

*olah*: román

*tot*: szlovák

*horvat*: horvát

*ruten*: rutén

*szerb*: szerb

*egyebnep*: Minden nemzetiség, amelyik nem tartozik a fentiekhez.

## nepterkep.php

Bemeneti paraméterek:

*lang*: A kimenet nyelve.

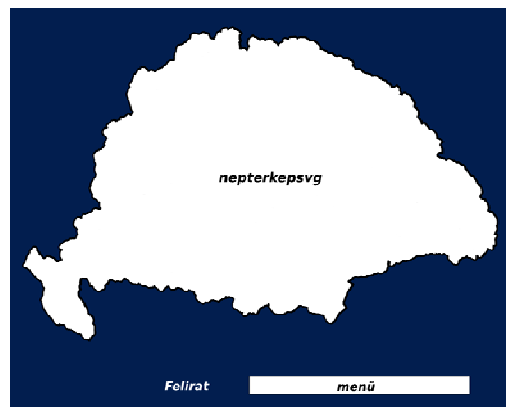
Kimenet típusa: text/html.

Kimenet leírása: Fent egy Magyarország 1914-es

történelmi közigazgatási térkép (*hungary\_1914\_noleg.svg*)

jelenik meg. Ennek azonosítója *nepterkepsvg*. Alatta egy

felirat, mellette egy menü látható. Ha változik a menü értéke, akkor a *updatemap* függvényt futtatja le a menü kiválasztott értékével. A menü lehetséges értékei „magyar”, „nemet”, „olah”, „tot”, „ruten”, „horvat”, „szerb”, „egyebnep”.



*nepterkep.php* kimenetén elhelyezkedő objektumok

## nkep.php

Bemeneti paraméterek:

*int*: Az intézet, amelyben a megtekinteni kívánt térkép van.

*filename*: A megtekinteni kívánt térkép teljes állományneve szelvénytárossal és kiterjesztéssel.

Kimenet típusa: image/jpeg.

Kimenet leírása: Nagy felbontásban kiküldi a bemeneti paraméterek szerint megadott térképet. A térképböngészőben megjelenő térképekre használatos.

## orszagnev.php

Bemeneti paraméterek:

*lang*: A kimenet nyelve.

Kimenet típusa: text/xml.

Kimenet leírása: A *lang* paraméterben megadott nyelven kiadja az európai országok neveit egy xml adathalmazban. A *dataset* tag attribútumai:

*id*: Az ország kódja.

*kod*: Az ország kódja, egyben az Európa térképen az országok poligonjainak azonosítója, valamint a szöveges elemek azonosítja *tkod*. *text*: az ország neve utf-8 kódolásban.

## **országok.php**

Bemeneti paraméterek: nincs.

Kiement típusa: text/xml.

Kimenet leírása: listázza azon országok kódjait, amelyek poligonjai aktívak az Európa térképen. Az dataset attribútumai a következők:

*id*: Egy sorszám azonosító.

*ország*: Az ország kódja csupa nagybetűvel.

*poligon*: A poligon azonosítója, azaz az ország kódja csupa kisbetűvel.

*terkepnev*: A térkép azonosítója, ami az országot teljes területébe ábrázolja.

## **search.php**

Bemeneti paraméterek:

*lang*: A kimenet nyelve.

Kimenet típusa: text/html.

Kimenet leírása: Az egyszerűsített kereső tartalmát hozza létre, amit a program a *search* objektumba helyez.



*A search.php által generált objektumok elhelyezkedése*

## **statkat.php**

Bemeneti paraméterek:

*int*: Az intézet, amelynek a térképeiről akarunk statisztikai térképet készíteni. Ha nincs megadva, akkor az összes intézetet veszi.

Kimenet típusa: text/xml.

Kimenet leírása: Egy xml adathalmazt kapunk vissza, ami megadja, hogy az adott térképen milyen színű legyen egy bizonyos vármegye poligon kitöltése. Az *id* a vármegye poligon azonosítója. A *tcolor* attribútum a vármegye kitöltése azon a térképen, amely azt mutatja, hogy hány térkép van, ami az adott vármegyének a teljes területét ábrázolja. Az *rcolor* attribútum a vármegye kitöltése azon a térképen, amely azt mutatja, hogy hány térkép van, ami az adott vármegye területének csak egy részét ábrázolja. Az *ecolor* attribútum a vármegye kitöltése azon a térképen, amely azt mutatja, hogy hány térkép van, ami az adott vármegye területét nem ábrázolja. Minden vármegyére egy sor vonatkozik.

### **statmenu.php**

Bemeneti paraméterek:

*lang*: A kimenet nyelve.

Kimenet típusa: text/html.

Kimenet leírása: Elkészít a statisztikai térkép alá egy menüt, amely felsorolja az intézeteket, amiket vizsgálni lehet.

### **stadat.php**

Bemeneti paraméterek:

*lang*: A kimenet nyelve.

Kimenet típusa: text/html.

Kimenet leírása: Kiírja a popup ablakba a honlap készítőjének és a készítésben közreműködők nevét. Az eredmény az *szdata* objektumba kerül. Két osztályt hoz létre:

*szcim*, amely a felirat, megnevezés, és

*szbody*, amely az adott emberek neveit tartalmazza.

## szhely.php

Bemeneti paraméterek:

*vm*: A vármegye, amelyiknek a székhelyét akarjuk.

Kimenet típusa: text/plain.

Kimenet leírása: A *vm* paraméterben megadott vármegye székhelyének a magyar nevét adja vissza entityk nélkül, escape karakterekkel.

## telnevq.php

Bemeneti paraméterek:

*lang*: A kimenet nyelve.

*telnev*: A keresett vármegye székhely neve.

*nyelv*: A telnév paraméterben megadott vármegyeszékhely mely nyelven van.

*vmkod*: A keresett vármegye kódja.

Kimenet típusa: text/html.

Kimenet leírása: Több táblázat jön létre: az első a vármegye székhely különböző nyelveken használatos alakjait listázza. A második a vármegye népességének lélekszámát összesen és nemzetiségre lebontva, továbbá a nemzetiségek százalékos arányát az össznépességhez képest. Fent megjelenik egy kisméretű térkép a vármegyéről, amelyre kattintva megjelenik ugyanez nagyobbban. Ezek az adatok a vármegyék adatai popup ablak bal oldalán jelennek meg.



*telszo.php* eredménye. A bal oldalon látszik a *telnevq.php* eredménye görgetősávval

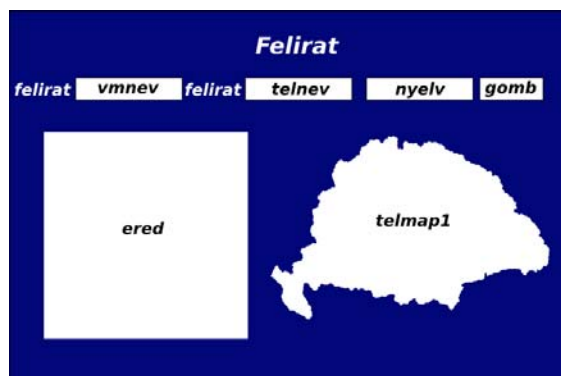
## telszo.php

Bemeneti paraméterek:

*lang*: A kimenet nyelve.

Kimenet típusa: text/html.

Kimenet leírása: A vármegyék információt ablak tartalmát hozza létre. Fent van egy felirat, alatta egy sor mező. Az első mezőbe a vármegye magyar nevét lehet beírni. Akkor kiírja az adatait. A mellette lévő mezőbe a vármegye székhely nevét a mellette lévő



*telszo.php kimenetének szemantikusan ábrázolása*

menüben kiválasztott nyelven. Keresés gomb, vagy az

„enter” billentyű lenyomásával indul a keresés. A jobb oldalon lévő térképen is ki lehet választani a kívánt vármegyét. A baloldalon létrehoz egy üres objektumot, ahová a *telnevq.php* eredménye kerül. Ez utóbbin szükség esetén megjelennek görgetősávok.

## tooltip.php

Bemeneti paraméterek:

*lang*: A kimenet nyelve.

*icon*: Az ikon megnevezése a *tooltip* tábla *icon* mezeje szerint.

Kimenet típusa: text/plain.

Kimenet leírása: Az *icon* paraméterben megadott tooltipet a *lang* paraméterben megadott nyelven adja vissza entiykkel.

## vmkep.php

Bemeneti paraméterek:

*vmkod*: A vármegye kódja, amelynek a térképét akarjuk.

Kimenet típusa: image/jpeg.

Kimenet leírása: a *vmkod* paraméterben megadott vármegyének a térképét adja vissza nagy felbontásban a térképböngésző számára.

### **vmkeresered.php**

Bemeneti paraméterek:

*minnat*: A megadott nemzetiség minimum aránya.

*maxnat*: A megadott nemzetiség maximum aránya.

*nemzet*: Azon nemzetiség, amelynek az arányait vizsgáljuk.

*feltetel*: Az a feltétel, amiként be akarjuk jelölni az érintett vármegyéket.

Az adatokat a *nepkeres.php*-ban létrehozott objektumokból adjuk át. Bővebben lásd a *nepkeres.php* leírását.

Kimenet típusa: text/xml.

Kimenet leírása: Kilistázza az érintett vármegyéket kóddal és a feltétellel, amit megadunk a bemenetnél. A *vm* attribútum az érintett vármegye kódját tartalmazza, míg a *felt* attribútum a feltételt, amit megadtunk a bementenél.

### **vmkeres.php**

Bemeneti paraméterek:

*lang*: A kimenet nyelve.

Minden vármegye kódnak megfelelő nevű paraméter, ami tartalmazza a vármegyének megadott feltételt. Bővebben lásd IX A vármegyék kódjai című mellékletet és a 11. A kereső működése című fejezetet.

Kimenet típusa: text/html.

Kimenet leírása: A vármegyes keresőben megadott feltételeknek megfelelően kiadja a keresési eredményt a *searchres* objektumba. Az eredmény felépítése megegyezik a *keres.php* eredményével.

## **vm.php**

Bemeneti paraméterek:

*vmnev*: A vármegye neve escape karakterekkel.

*vmkod*: A vármegye kódja.

Ha nincs megadva bemeneti paraméter, kilistázza az összes vármegyét. A fenti paraméterek közül csak egyet lehet megadni.

Kimenet típusa: text/xml.

Kimenet leírása: Kiadja a kért vármegye/vármegyék adatait egy xml adathalmazban. A *dataset* tag attribútumai a következők:

*id*: A sor száma.

*vmkod*: A vármegye kódja.

*vmnev*: A vármegye neve escape karakterekkel.

*vmszekhely*: A vármegye székhelyének a magyar neve escape karakterekkel.

## **vmsz.php**

Bemeneti paraméterek:

*tkszam*: A vizsgált térkép sorszám.

Kimenet típusa: text/xml.

Kimenet leírása: Kiadja, hogy mely vármegyét ábrázolja egészben, illetve részben a vizsgált térkép, valamint hogy melyeket nem ábrázolja. A *dataset* attribútumai a következők:

*id*: A vármegye kódja.

*vm*: A vármegye kódja.

*tart*: Ha „t”, akkor a vármegyét egészben ábrázolja a térkép; ha „r”, akkor csak részben; ha pedig „e”, akkor nem ábrázolja.



## vmszam.php

Bemeneti paraméterek: A vármegyekódokkal azonos paraméter. Nem kötelező megadni. Ha nincs megadva, akkor a nem megadott vármegyét mindenképpen az eredményben megszámlálja.

Kimenet típusa: text/plain.

Kimenet leírása: Megszámlálja, hogy a bemenetben megadott feltételek mellett hány térképet találna a kereső. Az eredményt egy számként elküldi a kimentre.

## VIII. Az Európa térképen használt országkódok

Országkód	Országnév	Országkód	Országnév	Országkód	Országnév
is	Izland	hr	Horvátország	no	Norvégia
al	Albánia	ie	Írország	it	Olaszország
dz	Algéria	pl	Lengyelország	ru	Oroszország
ad	Andorra	lv	Lettország	pt	Portugália
at	Ausztria	ly	Líbia	ro	Románia
be	Belgium	li	Liechtenstein	sm	San Marino
ba	Bosznia és Hercegovina	lt	Litvánia	es	Spanyolország
bg	Bulgária	lu	Luxemburg	ch	Svájc
cy	Ciprus	mk	Macedónia	se	Svédország

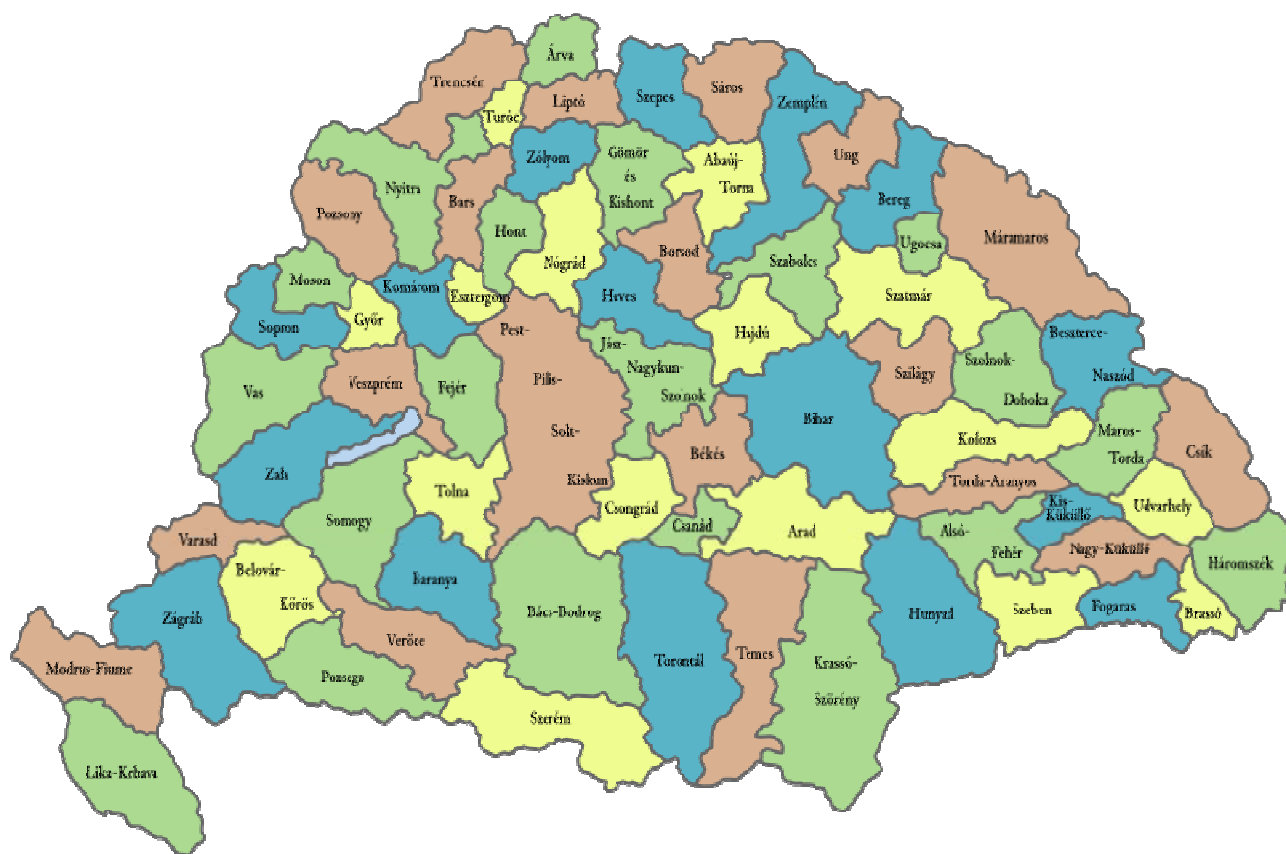


*Európa országai*

## IX. A vármegyék kódjai

Vármegye kódja	Vármegye neve	Vármegye székhelye
AT	Abaúj-Torna	Kassa
AF	Alsó-Fehér	Nagyenyed
AR	Arad	Arad
ARV	Árva	Alsókubin
BB	Bács-Bodrog	Zombor
BAR	Baranya	Pécs
BRS	Bars	Aranyosmarót
BEK	Békés	Gyula
BK	Belovár-Körös	Belovár
BG	Bereg	Beregszász
BN	Beszterce-Naszód	Beszterce
BH	Bihar	Nagyvárad
BS	Borsod	Miskolc
BR	Brassó	Brassó
CSA	Csanád	Makó
CSI	Csik	Csikszereda
CSO	Csongrád	Szentés
ESZ	Esztergom	Esztergom
FJ	Fejér	Székesfehérvár
FR	Fogarás	Fogarás
GK	Gömör és Kishont	Rimaszombat
GY	Győr	Győr
HJ	Hajdú	Debrecen
HSZ	Háromszék	Sepsiszentgyörgy
HV	Heves	Eger
HO	Hont	Ipolyság
HU	Hunyad	Déva
JNS	Jász-Nagykun-Szolnok	Szolnok
KK	Kis-Küküllő	Dicsőszentmárton
KO	Kolozs	Kolozsvár
KOM	Komárom	Komárom
KSZ	Krassó-Szörény	Lugos
LK	Lika-Krbava	Gozspics
LI	Liptó	Liptószentmiklós
MM	Máramaros	Máramarossziget
MT	Maros-Torda	Marosvásárhely
MF	Modrus-Fiume	Ogulin
MO	Moson	Magyaróvár
NK	Nagy-Küküllő	Segesvár
NO	Nógrád	Balassagyarmat
NY	Nyitra	Nyitra
PPS	Pest-Pilis-Solt-Kiskun	Budapest
PZS	Pozsega	Pozsega
PO	Pozsony	Pozsony
SA	Sáros	Eperjes
SO	Somogy	Kaposvár
SOP	Sopron	Sopron
SZA	Szabolcs	Nyíregyháza
SZM	Szatmár	Nagykároly
SZB	Szeben	Nagyszeben
SZP	Szepes	Lőcse

Vármegye kódja	Vármegye neve	Vármegye székhelye
SZR	Szerém	Vulkovár
SZL	Szilágy	Zilah
SZD	Szolnok-Doboka	Dés
TM	Temes	Temesvár
TOL	Tolna	Szekszárd
TA	Torda-Aranyos	Torda
TOR	Torontál	Nagybecskerek
TR	Trencsén	Trencsén
TUR	Turóc	Turócszentmárton
UDV	Udvarhely	Székelyudvarhely
UG	Ugocea	Nagyszőlős
UN	Ung	Ungvár
VR	Varasd	Varasd
VA	Vas	Szombathely
VC	Verőce	Eszék
VSZ	Veszprém	Veszprém
ZG	Zágráb	Zágráb
ZA	Zala	Zalaegerszeg
ZP	Zemplén	Sátoraljaújhely
ZM	Zólyom	Besztercebánya



*Magyarország vármegyéi 1914-ben*

## X. Intézetek, ahonnan térképeket gyűjtöttem

Az intézet neve:

**Hadtörténeli Intézet és Múzeum**  
**Hadtörténeli Könyvtár és Térképtár**

Az intézet címe:

1014 Budapest, Kapisztrán tér 2–4.



Az intézet honlapjának címe:

<http://www.militaria.hu/hun/rlglap.php?rlgazon=26>

Felhasznált térképművek száma: 28

Szkennelt lapok száma: 158

Az intézet kódja: hti

Az intézet neve:

**Heidelbergi Egyetem**  
**Földrajzi Intézetének Könyvtára**

Az intézet címe:

Im Neuenheimer Feld 348 Heidelberg D-69120  
Németország

Az intézet honlapjának címe:

<http://www.geog.uni-heidelberg.de/institut/bibliothek.html>



Felhasznált térképművek száma: 4

Szkennelt lapok száma: 45

Az intézet kódja: hdgi

Az intézet neve:

**Badeni Tartományi Könyvtár**

Az intézet címe:

Erbprinzenstraße 15 Karlsruhe D-76133 Németország

Az intézet honlapjának címe:

<http://www.blb-karlsruhe.de>

Felhasznált térképművek száma: 32

Szkennelt lapok száma: 49

Az intézet kódja: blb



Az intézet neve:

**Karlsruhe Institut für Technologie Könyvtára**

Az intézet címe:

Straße am Forum 2 76131 Karlsruhe

Az intézet honlapjának címe:

<http://www.bibliothek.kit.edu/cms/index.php>

Felhasznált térképművek száma: 1

Szkennelt lapok száma: 2

Az intézet kódja: kti



Az intézet neve:

**Heidelbergi Egyetem, Egyetemi Könyvtár**

Az intézet címe:

Plöck 107-109 D-69117 Heidelberg

Az intézet honlapjának címe:

<http://www.ub.uni-heidelberg.de>



Felhasznált térképművek száma: 1

Szkennelt lapok száma: 1

Az intézet kódja: hdub

Az intézet neve:

**Tübingeni Egyetem Földrajzi Intézete**

Az intézet címe:

Eberhard Karls Universität Tübingen

Geographisches Institut

Rümelinstraße 19-23 D-72070 Tübingen

Az intézet honlapjának címe:

<http://www.geo.uni-tuebingen.de/sammlungen/karten-atlanten-und-luftbildsammlung.html>



Felhasznált térképművek száma: 11

Szkennelt lapok száma: 446

Az intézet kódja: tbgi

A feldolgozott térképek jegyzéke a melléklet végén lévő táblázatban található.

<i>Térkép címe</i>	<i>Térkép szerzője</i>	<i>Kiadó</i>	<i>Méretarány</i>	<i>Kiadás éve</i>	<i>Lapméret</i>	<i>Kiadás helye</i>	<i>Jelmagyarázat nyelve</i>	<i>0: Önálló térkép 1: Atlaszlap</i>
<b>Bádeni Tartományi Könyvtár</b>								
The Geographical Review c. lap 6. évfolyamának 3. számában lévő, Magyarország északi részét ábrázoló, néprajzi, népsűrűségi és domborzati térkép	B.C. Wallis	The American Geographical Society	1:2000000	1918	-	New York	Angol	1
The Geographical Review c. lap 6. évfolyamának 2. számában lévő, Magyarország keleti részét ábrázoló, néprajzi, népsűrűségi és domborzati térkép	B.C. Wallis	The American Geographical Society	1:2000000	1918	-	New York	Angol	1
The Geographical Review c. lap 6. évfolyamának 4. számában lévő, Magyarország déli részét ábrázoló, néprajzi, népsűrűségi és domborzati térkép	B.C. Wallis	The American Geographical Society	1:2000000	1918	-	New York	Angol	1
The Geographical Review c. lap 6. évfolyamának 5. számában lévő, Magyarország nyugati részét ábrázoló, néprajzi, népsűrűségi és domborzati térkép	B.C. Wallis	The American Geographical Society	1:2000000	1918	-	New York	Angol	1
Különböző statisztikai adatokat ábrázoló fekete-fehér térképek Magyarországról a The Geographical Review c. lap 4. évfolyamából.	B.C. Wallis	The American Geographical Society	1:8250000	1917	-	New York	Angol	1
Hováország néprajzi térképe a „The Geographical Journal” c. folyóiratból	Sir Arthur Evans	Royal Geographical Society	1:2000000	1916	37x34cm	London	Angol	1
B.C.Wallis által készített 6 db térkép kivágot az Alföld déli részéről amely a népességábrázolás módszereit mutatja be. (éNy: Kecskemét, DK: Lugos)	B.C.Wallis	Royal Geographical Society	1:1250000	1916	19x13cm	London	Angol	1
Erdély domborzata, népsűrűsége és Erdői, A Petermanns Geographische Mitteilungen c. folyóirat 1857-es számából	August Petermann	Justus Perthes Gotha	-	1857		Gotha	Német	1
Az Osztrák–Magyar Monarchia népessége az írástudók aránya szerint, A Petermanns Geographische Mitteilungen c. folyóirat 1884-es számából	Hátsek Ignác	Justus Perthes Gotha	1:3700000	1884		Gotha	Német	1
A Magyar Korona országainak néprajzi térképe az 1880-as népszámlálás alapján, A Petermanns Geographische Mitteilungen c. folyóirat 1885-ös számából	Hátsek Ignác	Justus Perthes Gotha	1:4000000	1885		Gotha	Német	1
A Deutsch Erde c. folyóirat 1904-es számában található Európa népességét ábrázoló térkép 1815-ös és 1900-as állapot szerint, valamint egy másik térkép ami Erdélyben ábrázolja az ágostai Evangélikus egyház közigazgatási beosztását.	Paul Langhans	Justus Perthes Gotha	1:15000000,	1904		Gotha	Német	1
A Deutsch Erde c. folyóirat 1907-es számában található a német képviselőket az európai parlamentekbe küldő körzetek feltüntetésére (a többi nemzetiség is meg van jelölve, csak a cím nem utal rá). Nem térkép, hanem térképszerű ábrázolás	Paul Langhans	Justus Perthes Gotha	-	1907		Gotha	Német	1
A Deutsch Erde c. folyóirat 1908-as számában található a német népesség elterjedését Közép-Európában ábrázoló térkép	Paul Langhans	Justus Perthes Gotha	-	1908		Gotha	Német	1
A Deutsch Erde c. folyóirat 1908-as számában található észak-Erdélyben élő németek arányát ábrázoló térkép	Paul Langhans	Justus Perthes Gotha	-	1908		Gotha	Német	1
A Deutsch Erde c. folyóirat 1910-es számában található a német nyelv elterjedését ábrázoló térkép Sopron környékén	Paul Langhans	Justus Perthes Gotha	-	1910		Gotha	Német	1
A Deutsch Erde c. folyóirat 1910-es számában található a németeket megillető nevek Európa különböző részein	Paul Langhans	Justus Perthes Gotha	-	1910		Gotha	Német	1



<b>Térkép címe</b>	<b>Térkép szerzője</b>	<b>Kiadó</b>	<b>Méretarány</b>	<b>Kiadás éve</b>	<b>Lapméret</b>	<b>Kiadás helye</b>	<b>Jelmagyarázat nyelve</b>	<b>0: Önálló térkép 1: Atlaszlap</b>
A Deutsch Erde c. folyóirat 1910-es számában található a német nyelv elterjedését ábrázoló térkép Mosonmagyaróvár környékén	Paul Langhans	Justus Perthes Gotha	-	1910		Gotha	Német	1
A Deutsch Erde c. folyóirat 1910-es számában található a német nyelv elterjedését ábrázoló térkép Pozsony környékén	Paul Langhans	Justus Perthes Gotha	-	1910		Gotha	Német	1
A Deutsch Erde c. folyóirat 1910-es számában található a német nyelv elterjedését ábrázoló térkép Vasvár környékén	Paul Langhans	Justus Perthes Gotha	-	1910		Gotha	Német	1
A Deutsch Erde c. folyóirat 1911-es számában található Magyarország területén a németek növekedését és csökkenését ábrázoló térkép 1910. december 31-ei állapot szerint	Paul Langhans	Justus Perthes Gotha	-	1911		Gotha	Német	1
Magyarország népességtérképe a Deutsche Erde c. folyóirat 1912-es számában	Paul Langhans	Justus Perthes Gotha	-	1912		Gotha	Német	1
Európában használt írásképek a Deutsch Erde 1913-es számában	Paul Langhans	Justus Perthes Gotha	-	1913		Gotha	Német	1
Beszterce környékének néprajzi térképe	Paul Langhans	Justus Perthes Gotha	-	1920		Gotha	Német	1
Csehszlovákia gazdasági erőforrásai és népességének anyanyelvét ábrázoló térkép	Paul Langhans	Justus Perthes Gotha	-	1920		Gotha	Német	1
A német nyelv elterjedése Erdélyben	Paul Langhans	Justus Perthes Gotha	-	1920		Gotha	Német	1
Európa népessége országhatárok feltüntetésével	Slavko Smrekar	Justus Perthes Gotha	-	1920		Gotha	Német	1
Közép-Európa népessége			-	1939			Angol	1
Az ágostai Evangélikus Egyház egyházigazgatása Erdélyben	Paul Langhans	Justus Perthes Gotha	1:576000	1904		Gotha	Német	1
A németek elterjedése a Magyar Korona országaiban	Paul Langhans	Justus Perthes Gotha	1:1500000	1896		Gotha	Német	1
Az erdélyi szász terület nyelvi térképe (Nagyszeben környéke)	Paul Langhans	Justus Perthes Gotha	1:200000	1920		Gotha	Német	1
Az erdélyi szász terület nyelvi térképe (Brassó környéke)	Paul Langhans	Justus Perthes Gotha	1:200000	1920		Gotha	Német	1
A németiség földrajzi elterjedése Magyarországon múltban és jelenben	Dr. Adolf Rieth	Ausland und Heimat Verlags-Aktiengesellschaft	1:500000	1927	38x26cm 11x17cm	Stuttgart	Német	1
<b>Heidelbergi Egyetem Földrajzi Intézete Könyvtára</b>								
Magyarország néprajzi térképe település és lélekszám szerint; a Magyar Népköztársaság Külügyminisztériumának megbízásából	Bátky Zsigmond és Kogutowicz Károly	Országos Magyar Földrajzi Intézet	1:300 000	1919	-	Budapest	Magyar, Német, Angol, Francia	0
The Geographical Review című lap 6. évfolyamának 3., 4. és 5. számából származó, összeragasztott térkép, Magyarország népessége anyanyelv szerint 1918-ban	The American Geographical Society of New York, készítette B.C. Wallis	The American Geographical Society of New York	1:2 000 000	1918	-	New York	Angol	0
The Geographical Review című lap 6. évfolyamának 3., 4. és 5. számából származó, összeragasztott térkép, Magyarország népsűrűsége 1918-ban	The American Geographical Society of New York, készítette B.C. Wallis	The American Geographical Society of New York	1:2 000 000	1918	-	New York	Angol	0
Physikalisch-statistischer Hand-Atlas von Oesterreich-Ungarn	Joseph Chavanne	Hölzel	-	1887	-	Bécs	Német	1
<b>Heidelbergi Egyetem, Egyetemi Könyvtár</b>								
Csehszlovákiában élő nemzetiségek		Deutscher Verlag	1:1 500 000	1938	67 x 36 cm	Berlin	Német	0

<b>Térkép címe</b>	<b>Térkép szerzője</b>	<b>Kiadó</b>	<b>Méretarány</b>	<b>Kiadás éve</b>	<b>Lapméret</b>	<b>Kiadás helye</b>	<b>Jelmagyarázat nyelve</b>	<b>0: Önálló térkép 1: Atlaszlap</b>
A Trianoni békeszerződés térképei	-	Hadtörténelmi Intézet és Múzeum, Hadtörténeli Könyvtár és Térképtár Magyar Földrajzi Intézet	-	1920	-	Budapest	Magyar, Angol, Francia	0
Magyarország néprajzi térképe az 1910. évi népszámlálás alapján, egy nemzetiség laponként	Dr. Moricz Miklós; Dr. Kogutowicz Károly	Magyar Földrajzi Intézet	1:1000000	1919	93x57cm	Budapest	Magyar, Angol	0
Magyarország néprajzi térképe 4 melléktérképpel és 6 kördiagrammal	Dr. Kogutowicz Károly	Magyar Királyi állami Térképészet	1:1000000	1928	93x57cm	Budapest	Magyar	0
A Habsburg Birodalom néprajzi térképe	Czörnig Károly	Császári és Királyi Statisztikai Hivatal	1:864000	1855	71x52cm	Bécs	Német	0
Az Osztrák-Magyar Monarchia néprajzi és nyelvi térképe, A vliágkereskedelem és a közlekedés térképe	-	Freytag & Berndt	1:1500000	1907	85x66cm	Bécs	Német	0
Magyarország néprajzi térképe, szelvényezett, hiányos (1910-es adat nagy valószínűséggel!!)	-	-	-	-	67x60cm	-	Magyar	0
Románok és nem románok a Kárpátok és a Tisza között az 1910-es népszámlálás alapján. Néprajzi térképek az 1920-as trianoni béketárgyaláshoz	-	-	-	1920	41x35.5cm	-	Francia	0
Magyarország néprajzi térképe, Magyar Népköztársaság Külügyminisztériumának megbízásából, szelvényezett, 12 szelvény	Batky Zsigmond-Kogutowicz Károly	Magyar Földrajzi Intézet	1:300000	1919	88x68cm	Budapest	Magyar,Német,Angol,Francia	0
Magyarország népsűrűsége az 1890-es évek végén	Posner Károly Lajos és fia	Posner Károly Lajos és fia Térképészeti Műintézet	-	-	45.5x29cm	Budapest	Magyar, Német	0
Magyarország népessége hitfelekezet szerint: római katolikus, görög katolikus, ágostai evangélikus, református, zsidó	Posner Károly Lajos és fia	Posner Károly Lajos és fia Térképészeti Műintézet	-	-	13x9cm	Budapest	Magyar, Német	0
Az összes magyarul tudók az 1890-es évek végén	Posner Károly Lajos és fia	Posner Károly Lajos és fia Térképészeti Műintézet	-	-	45.5x29cm	Budapest	Magyar, Német	0
Magyarország népessége anyanyelv szerint, magyar, német, tót, szerb, vend, oláh, horvát, rutén	Posner Károly Lajos és fia	Posner Károly Lajos és fia Térképészeti Műintézet	-	-	13x9cm	Budapest	Magyar, Német	0
6 éven felüli népesség műveltségi foka, írni-olvasni tudó az össz. illetve ugyanez a női népességből, ill a férfi népességből	Posner Károly Lajos és fia	Posner Károly Lajos és fia Térképészeti Műintézet	-	-	13x9cm	Budapest	Magyar, Német	0
A nemek aránya	Posner Károly Lajos és fia	Posner Károly Lajos és fia Térképészeti Műintézet	-	-	29.5x20cm	Budapest	Magyar, Német	0
Magyarország ethnográfiai térképe az 1880-as népszámlálás alapján, Magyar nyelvterületek színesen jelölve	Réthy Ferenc	Posner Károly Lajos és fia Térképészeti Műintézet	1:1152000	1880	83x55cm	Budapest	Magyar	0
Magyarország néprajzi térképe 1900 körül, nemzetiségi többség alapján osztályozva	Cholnoky Jenő	Magyar Földrajzi Intézet	1:2350000	1900	65x52cm	Budapest	Magyar	0
Szlovákia, Galícia, Bukovina és Erdély térképe, nyelvhatárokkal	-	Freytag & Berndt	-	-	41x28cm	Bécs	Német	0
Magyarország néprajzi térképe az 1910-es adatok alapján	Döre Jenő és Lux Ernő	Klősz György és fia Térképészeti Műintézet	1:900000	1910	109x76cm	Budapest	Magyar, Angol, Francia	0
Magyarország néprajzi térképe, a nemzetiségeket különböző színű körök jelölik, 1 kör 1000 lakos	Döre Jenő és Lux Ernő	Klősz György és fia Térképészeti Műintézet	1:900000	1920	109x76cm	Budapest	Magyar, Német, Angol, Francia, Olasz	0
Magyarul tudók aránya és száma járásokként Magyarországon. hátoldal: Magyarország népessége hitfelekezetenként 1910-ben	-	Magyar Földrajzi Intézet	-	1910	54x36cm	Budapest	Magyar, Angol, Francia	0
A magyar nyelvjárások 1905-ben	Balassa József	-	-	1905	21x16cm	-	Magyar	0
Közép-Európa néprajzi térképe 1930 körül, az 1914-es és 1930-as országhatárokkal	-	-	1:2000000	1930	56.5x42cm	-	Francia	0
Közép-Európa néprajzi térképe	Dr. Wilhelm Winkler	A Bécsi Egyetem Kisebbségek Statisztikáját Kutató Intézete	1:4000000	1930	35x29cm	Bécs	Német	0
A dunai államok néprajzi térképe	Dr. Györffy István	Magyar Királyi állami Térképészet	1:2200000	-	58.533cm	Budapest	Magyar	0

<b>Térkép címe</b>	<b>Térkép szerzője</b>	<b>Kiadó</b>	<b>Méretarány</b>	<b>Kiadás éve</b>	<b>Lapméret</b>	<b>Kiadás helye</b>	<b>Jelmagyarázat nyelve</b>	<b>0: Önálló térkép 1: Atlaszlap</b>
A 6 évesnél idősebb írni-olvasni tudók és analfabéták száma anyanyelvi megoszlással egymással szembeállítva Magyarországon, törvényhatóságoként. Hátoldal: az írni-olvasni tudók száma és aránya járásoként Magyarországon a 6 éven aluliak beszámításával	-	Magyar Földrajzi Intézet	-	-	54x36cm	Budapest	Magyar, Angol, Francia	0
Magyarország néprajzi térképe 1846 körül	-	-	-	1846	25x20	-	Német	0
Magyarország nemzetiségi térképe	Dörflinger E.	-	-	1773	51x44cm	-	Magyar	0
Magyarország népessége anyanyelv szerint 1910-ben járásoként	-	Klösz György és fia Térképészeti Műintézet	-	1910	107x78cm	Budapest	Magyar, Angol, Francia	0
Karlsruhe Institut für Technologie Könyvtára								
Mitteleuropa Die Länder und Völker c. könyvben lévő közép-európa népességi és történelmi népességi térképek	Dr. J. Partsch	Gotha: Justus Perthes	-	1904	19.5x12.5cm	Gotha	Német	1
		<b>Tübingeni Egyetem Földrajzi Intézete</b>						
Zsebtasz naptárral és statisztikai adatokkal az 1922. évre	Dr. Bezdek József és Dr. Karl János	Magyar Földrajzi Intézet	-	1921	11.5x15.5cm	Budapest	Magyar	1
Kogutowicz Zsebtasza 1923	Bátky Zsigmond és Kogutowicz Károly	Magyar Néprajzi Társaság Emberföldrajzi Szakosztálya	-	1923	20.5x13.5cm	Budapest	Magyar	1
Francia nyelvű Magyarország Atlasz	-	-	-	-	22.5x30cm	-	Francia	1
Kárpátalja népességének eloszlása	Karls As. Sedlmeyer	Prágai Német Egyetem Földrajzi Intézete	1:200 000	1932	97.5x67.5cm	Prága	Német	0
Dél-Kelet Európa Népessége		Gotha: Justus Perthes	1:1500000	1943	100x82cm	Gotha	Német	0
A székesfőváros multja és jelene grafikus ábrázolásban	Dr. Illyefalvi I. Lajos	Budapest Székesfőváros Statisztikai Hivatala		1933	33.5x26cm	Budapest	Magyar, Német	1
Románia történelmi és etnográfiai atlasza				1919	31x24.5cm	Parisz	Francia, Angol	1
A Duna-Kárpát térség		Deutschen Ausland-Institut		1943	34x36cm	Stuttgart	Német	1
Petermanns geographische mitteilungen 1860 as évfolyamában lévő 7-es számú térkép	A. Petermann	Gotha: Justus Perthes		1860	22x27cm	Gotha	Német	1
Jugoszlávia kultúrföldrajzi beosztása	Elisabeth Lichtenberger és Hans Bobek			1956	45x33cm		Német	1
Magyarország gazdasága térképekben	Edvi Illés Aladár és Halász Albert	Pallas Nyomda		1921		Budapest	Magyar	1