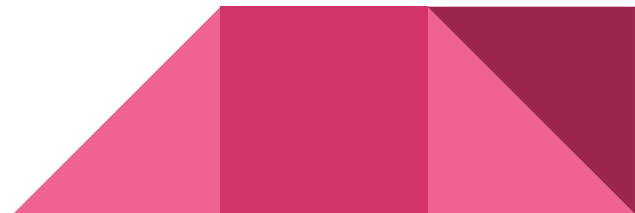


# D3.js könyvtár

Gergely Szabolcs

# Mi a D3

- JavaScript könyvtár
- D3 = Data-Driven Documents(Adatirányított Dokumentumok)
- Dinamikus, interaktív adatvizualizáció böngészőkben
- HTML<sup>[1]</sup>, SVG<sup>[2]</sup> és CSS<sup>[3]</sup> használata
- Adatok `hozzákötése` a DOM<sup>[4]</sup>-hoz, majd ezeken végzett transzformációk
- Gyors sok adatok kezelésére, moduláris



# Adatvizualizáció

- a könyvtár szerzője a következő lépéseket javasolja az adatvizualizációra:
  - beszerzés - adatok beszerzése; pl. fájlból vagy adatbázisból
  - elemzés - adatstruktúrálás értelelem alapján, kategorizálás
  - szűrés - csak a jellegzetes adatokat tartjuk meg
  - bányászat - statisztika- vagy adatbányászattal matematikai kontextusba tesszük az adatokat
  - reprezentáció - adatok reprezentációja
  - finomítás - reprezentáció finomítása, precizitása
  - interakció - adatok manipulációja



# TopoJSON

- földrajzi adatok JSON-szerű formátumba való reprezentációja
- geoJSON<sup>[5]</sup> bővítménye
- topoJSON fájlok `össze vannak olvasztva` közös vonal alapján(*ív - arc*)
- az `összeolvasztás` miatt 85-90%-os méretcsökkentés érhető el geoJSON-hoz képest
- viszont a D3 könyvtár geoJSON-al dolgozik:
  - visszaalakítja a topoJSON-t geoJSON-ba mielőtt műveleteket végez rajta



# TopoJSON

- az ív egy koordinátasorozat, ami topológia szinten van tárolva :

```
{
  "type": "Topology",
  "transform": {
    "scale": [0.036003600360036005, 0.017361589674592462],
    "translate": [-180, -89.99892578124998]
  },
  "objects": {
    "aruba": {
      "type": "Polygon",
      "arcs": [[0]],
      "id": 533
    }
  },
  "arcs": [
    [[3058, 5901], [0, -2], [-2, 1], [-1, 3], [-2, 3], [0, 3], [1, 1], [1, -3],
    [2, -5], [1, -1]]
  ]
}
```

# D3 és topoJSON

- topoJSON formátumú adatokat használva szép térképeket lehet készíteni
- SVG manipuláló függvények könnyítik a térképkészítést/módosítást
- CSS segítségével az SVG térképek könnyen stilizálhatóak
- lehetőséget ad nagyításra és mozgatásra is
- ezeket viszonylag egyszerű javascript módszerekkel lehet kezelni
- összes funkciói részletesen megtalálhatók a [wiki](#)-oldalán
- egy kezdetleges térképminta látható az 1. ábrán



# D3 és topoJSON

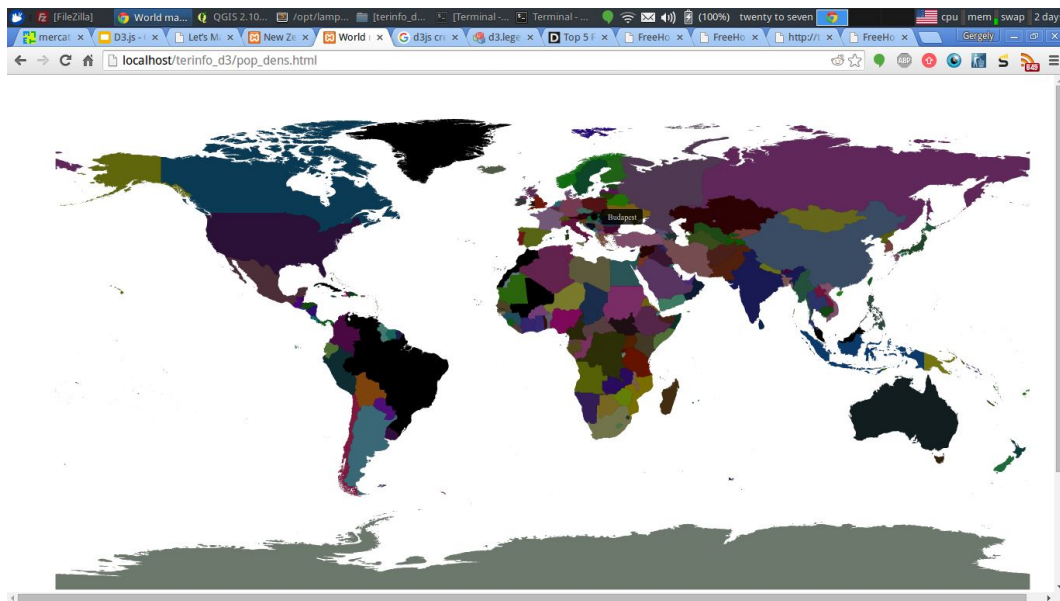


1. ábra

Új-Zéland térképe régiókra osztva, nagyobb településeket is feltüntetve. Településre mozgatva az egeret megjelenik a település neve, régió esetében pedig kivilágosodik a régió

# D3 és topoJSON

- kicsit komolyabb dolog a térképnagyítás ami a 2. és 3. ábrákon látható

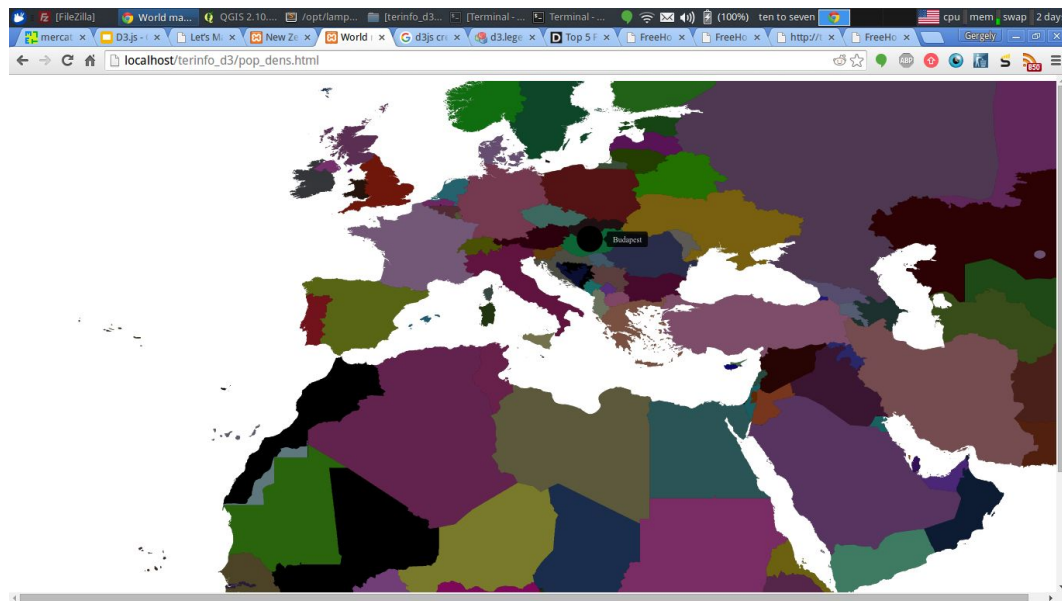


## 2. ábra

Világtérkép különböző színű országokkal, Budapest-el megvilágítva



# D3 és topoJSON



## 3. ábra

Világtérkép különböző színű országokkal, Budapest-el megvilágítva, kinagyítva

# D3 és topoJSON

- hátrány hogy WEB-es felületű megjelenítésről beszélünk
- ez azt jelenti hogy nagyon részletes térképek készítése bonyolult, optimalizáló eljárásokat igényel(pl. részletes magassági térképek), mivel ezek nagy mennyiségű adatok(~> 1GB topoJSON fájl)
- ilyen nagy adatok megjelenítését nem biztosítja a D3 js
- a könyvtár adatbeolvasó algoritmus módosításával viszont ez is megvalósítható(bonyolult)
- webszerver szükséges a működéséhez



# Térkép D3.js-el

- Világ- és Új-Zéland térkép D3 könyvtárral
- shapefile-ből szerzett adatok
- adatok átalakítása geoJSON majd topoJSON formátumba ogr2ogr programmal
- topoJSON felhasználása a WEB-es térkép megteremtéséhez
- webszerverre való feltöltés



# Index

- [1] : HTML - weboldalt leíró nyelv
  - kulcsszavakat(*TAG*) használ különböző elemtípusok leírására (pl. `<a>` = link, `<p>` = paragrafus)
  - elemek hierarhikus elhelyezkedése
  - a böngészők képesek ezen elemeket vizuálisan megjeleníteni
- [2] : SVG - XML formátumú fájl képek, ábrák megjelenítésére
  - HTML-hez hasonlóan hierarhikus *TAG*-eket tartalmaz
  - *TAG*-ek lehetnek elemi mértani alakzatok pl. kör
  - CSS segítségével is lehet stilizálni

# Index

- [3] : CSS - stílus leírása
  - HTML elemekre alkalmazható stílusok
  - elemek *ID* vagy *class* attribútumokkal azonosíthatók
  - egy elemre több stílus alkalmazható, prioritás vagy sorrend alapján
- [4] : DOM - HTML elemekből generált gráf
  - interaktív objektumokból áll
  - elemeket lehet követni, eseményeket kezelni rajtuk



# Index

- [5] : geoJSON - földrajzi adatok reprezentációja
  - egy objektum lehet egy geometriai alakzat, jellemzője vagy jellemzői
  - lehetséges geometriatípusok : *Pont, Vonal, Poligon, MultiPont, MultiVonal, MultiPoligon, Geometriai gyűjtemény*
  - egy jellemző egy geometriai objektumot és tulajdonságait tartalmazza :

```
{  
  "type": "Feature",  
  "geometry": {  
    "type": "Point",  
    "coordinates": [125.6, 10.1]  
  },  
  "properties": {  
    "name": "Dinagat Islands"  
  }  
}
```

# Irodalom

- [https://en.wikipedia.org/wiki/D3.js#Technical\\_principles](https://en.wikipedia.org/wiki/D3.js#Technical_principles)
- <https://github.com/mbostock/d3>
- <https://www.dashingd3js.com/>
- <https://square.github.io/intro-to-d3/>
- <https://github.com/mbostock/d3/wiki/API-Reference>
- <http://bost.ocks.org/mike/map/> - részletes térképkészítés D3.js-el

