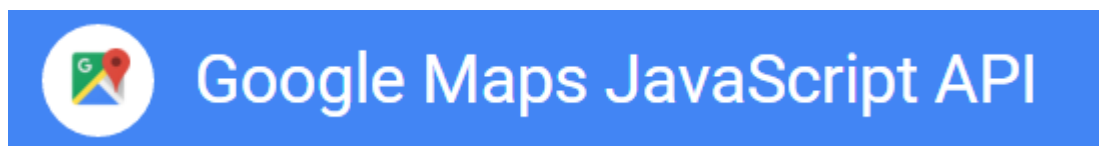


Térinformatika gyakorlat 3.Beadandó

Név: Wei Cheng

Neptun: ZJZ2AG

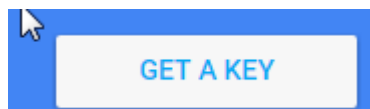
11, Google Maps JavaScript API v3. Mutassa be a Google Maps weboldalba ágyazható API-ját röviden (kezdő lépések, korlátok, alapok). Gondolja át, hogyan lehetne egy georeferált raszteres térkép egyes (a felhasználó által választott) rétegeit digitalizálni a weben. Készítsen egyszerű mintaalkalmazást. A digitalizált eredmény legyen menthető (fájlba vagy adatbázisba).



Használati útmutatás:

Első lépések

1. Google account szükséges az API használatához
 - a) Google account regisztrálás: <https://developers.google.com>
2. Google map API key lekérés a Google API weboldalon <https://developers.google.com/maps>



3. Ezt a sort kell beleszúrni a `<head></head>` közé

```
<script
  type="text/javascript"
  src="https://maps.googleapis.com/maps/api/js?key=XXXAPIKEYXXX&sensor=false&language=en" >
</script>
```

Külső Javascript kódként hivatkozunk az Google Térkép programozói felületére. A `sensor=false` paraméter jelentése, hogy egy olyan eszközön fogjuk használni a térkép szolgáltatást, aminek nincs hely meghatározó szenzora. Egy webes böngésző ilyen. Egy okostelefon esetében már lehetne igazra állítani a sensor paramétert.

4. Térkép generálás

Az API által nyújtott Map nevű JS osztály reprezentálja a térképet. Egy adott példány meghatároz egy térképet. Lehet több példány is, ha több térképre van szükség egy oldalon.

Létrehozáskor paraméterbe vár egy HTML konténert, ami majd tartalmazni fogja a térképet, ez egy DOM elem, ami tipikusan egy div szokott lenni. Valamint vár egy natív objektumot, ami a beállításokat tartalmazza.

Lépések:

- a) Az initialize() függvénnyel tudunk térképet generálni megfelelő paraméterek megadása után.
- b) Az inicializálásnál meg kell adni egy canvas azonosítót, és majd ezzel az azonosítóval tudjuk majd összekötni és megjeleníteni.
`<div id="map-canvas" class="test-map" style="height:600px;"></div>`

5. Egy Hello world kód minta:

```
<!DOCTYPE html>
<html>
  <head>
    <style type="text/css">
      html, body { height: 100%; margin: 0; padding: 0; }
      #map { height: 100%; }
    </style>
  </head>
  <body>
    <div id="map"></div>
    <script type="text/javascript">

var map;
function initMap() {
  map = new google.maps.Map(document.getElementById('map'), {
    center: {lat: -34.397, lng: 150.644},
    zoom: 8
  });
}
```

A példa programokat feltöltöttem a saját weboldalamra:

https://weicheng.web.elte.hu/terinfo/teszt1.html	(egyszerű alakzatok, 4 városnak a népsűrűsége)
https://weicheng.web.elte.hu/terinfo/teszt2.html	(szimbólumok és térkép módosítások, átszínezés stb)
https://weicheng.web.elte.hu/terinfo/teszt3.html	(felhasználói térkép)
http://weicheng.web.elte.hu/terinfo/save/	(adatok digitalizálása és mentése)

Térkép beállítások:

A google térképet egy JS Objektummal lehet beállítani. Ezt célszerű előre létrehozni. A fent említett példában látható módon. Megnézzük, hogy mik az alap beállítások és hogyan kell használni őket.

```
var mapOptions = {  
    center: new google.maps.LatLng (-25.363882,131.044922),  
    zoom: 7,  
    mapTypeId: google.maps.MapTypeId.ROADMAP  
};
```

Térkép középpontjának meghatározása: szélesség és hosszúság koordinátákkal (LatLng)

```
center: new google.maps.LatLng (-25.363882,131.044922);
```

A térkép középpontját meg kell határoznunk, ide fogja pozícionálni a nézetet, ez független a jelölőpontok koordinátáitól. Közepontot mindig meg kell határozni! Erre segítségünkre lehet az API LatLng objektuma, melynek használata értelem szerű. Első paraméter a szélesség, második a hosszúság.

Beállíthatjuk a térkép típusát. Jó lehet, ha autós térképről szeretnénk váltani domborzati térképre, vagy ezek kombinációi.

Lehetséges térkép típusok:

ROADMAP: Az alapértelmezett térkép.



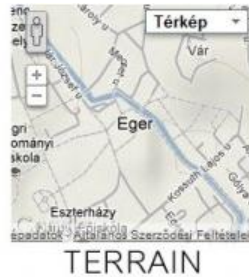
SATELLITE: Műholdas térkép.



HYBRID: Az előző két típus kombinációja. Műholdas térkép utca és város név információkkal



TERRAIN: Domborzati térkép.



Digitalizálás:

Alakzatok

Minta program: <http://weicheng.web.elte.hu/terinfo/teszt1.html>

Pontok:

Pontot sok féle képen tudjuk megadni, az alábbi 2 a leggyakoribb módszer

1. Pontos Latitude/Longitude megadásával (ez a leggyakoribb, és legjobb módszer).

Példa: `var London = new google.maps.LatLng(51.5072220, -0.1275000);`

2. Address megadásával is lehet (nem mindig pontos)

Példa: `var address= "Szőnyi út, Budapest, 1142 ";`

```
geocoder.geocode( { 'address': address}, function(results, status) {  
    if (status == google.maps.GeocoderStatus.OK) {  
        map.setCenter(results[0].geometry.location);  
    }  
})
```

Vonalak (Polyline):

Először meg kell adni a vonal és út tulajdonságait.

```
var polyOptions =  
    strokeColor: '#FF0000',  
    strokeOpacity: 1.0,  
    strokeWeight: 3,  
    map: map,  
};
```

Path-nél megadjuk a kordináta tömböt, és majd az alábbi sorral definiáljuk a polyline-t

```
teszt = new google.maps.Polyline(polyOptions);
```

majd a `poly.setPath(path)` függvénnyel tudjuk összekötni a vonalat.

Törlésnél pedig a `teszt.setMap(null)` függvényt kell használni

Poligonok:

Itt a definiálásnál a `google.maps.Polygon`-t kell használni

```
var teszt = new google.maps.Polygon({
```

```

    paths: tesztpath,
    strokeColor: '#FF0000',
    strokeOpacity: 0.8,
    strokeWeight: 2,
    fillColor: '#FF0000',
    fillOpacity: 0.35
  });
  Törlés: teszt.setMap(null)

```

A előbb felsorolt három leggyakrabban használt alakzaton kívül, van még sok egyéb más alakzatok is, mint pl: a rectangle(négyzet), circle(kör) stb.

Részletesebb információkat az api dokumentáció weboldalán lehet megtekinteni:
<https://developers.google.com/maps/documentation/javascript/shapes>

Szimbólumok

Minta program: <http://weicheng.web.elte.hu/terinfo/teszt2.html>

Marker-rel lehet szimbólumokat rajzolni a térképre

Az alábbi kóddal definiáljuk a markereket

```

marker = new google.maps.Marker({
    map: map,
    icon: pinImage,
    shadow: pinShadow,
    position: results[0].geometry.location,
    animation: google.maps.Animation.DROP,
});

```

Itt meg kell adni a térkép nevét, ikon-ját, pozícióját, stb.

Külön eseményvezérlést is tudunk adni a markerhez

```
google.maps.event.addListener(marker, 'click', toggleBounce);
```

Ikon:

a szimbólumok színét pedig az alábbi kóddal tudjuk módosítani

```
var pinColor = "bfd730";
```

A szimbólum lehet kép vagy svg, de vannak előre definiált google marker képek.

Példa:

```

var pinImage = new
google.maps.MarkerImage("http://chart.apis.google.com/chart?chst=d_map_pin_letter&chld=%
E2%80%A2|" + pinColor,

```

Saját típusú térkép:

Minta program: <http://weicheng.web.elte.hu/terinfo/teszt3.html>

Felhasználó által generált térképnél, az úgynevezett StyledMapType()-t tudjuk variálni. featureType-nél meg kell adni hogy mit szeretnénk(utak, folyók stb) és majd az elemTypenál kell megadni a típusát, majd a stylers-nél tudjuk különböző tulajdonságokat hozzáadni.

Példa:

```
var bpRoadMapType = new google.maps.StyledMapType([
  {
    featureType: 'road.highway',
    elementType: 'geometry',
    stylers: [
      {hue: '#000fff'},
      {saturation: 60},
      {lightness: -20}
    ]
  }
])
```

Itt az főutakat színeztünk át kékre,

```
    featureType: 'water',
    elementType: 'geometry',
    stylers: [
      {hue: '#000000'},
      {saturation: 40},
      {lightness: 40}
    ]
  },
```

Itt pedig a folyókat színeztünk át.

Adatok digitalizálása:

Minta program: <http://weicheng.web.elte.hu/terinfo/save/>

Elég macerás az adatok digitalizálása, mert a javascript cliens oldalú, emiatt az adatokat elég nehéz elmenteni, ezért csak egyszerű pontokat próbáltam eltárolni.

A program készítéshez PHP és MySQL -t használtam, ajax-szal átadtam a js adatokat PHP-nek, majd egy post-al elküldöm, és utána tárolom őket az adatbázisban.

Létre lehet hozni pontokat, és elmenti őket az adatbázisban.

Rákattintva törölni is lehet őket.

Előnyök és hátrányok

Előny:

- Semmit nem kell letölteni (csomagok, könyvtárak).
- Javascriptnek köszönhetően, nagyon egyszerű a használata.

Hátrány:

- Sajnos 25000 térkép betöltés-ig ingyenes az API, utána fizetni kell érte.
- Internet kapcsolat szükséges a használathoz