HDMA
@SDSU
THE CENTER FOR
HUMAN DYNAMICS
IN THE MOBILE AGE

Technical Document

HealthWebMapper 1.5

Project conducted by:
San Diego State University
5500 Campanile Drive
San Diego, CA 92182-4493

Report prepared by:
Haihong Huang
Department of Geography, San Diego State University

8/27/2018

Abstract

HealthWebMapper 1.5 is a new version of HealthWebMapper1.0 developed using Leaflet 1.3.1 with customized Leaflet Cascading Style Sheets (CSS). HealthWebMapper 1.5 keeps the original features in HealthWebMapper 1.0 but includes the following new features: 1) upgrading Leaflet from 0.7.3 to 1.3.1; 2) Layer Control function: base maps (Black&White, Terrain, OpenStreetMap, World Imagery, National Geographic Map) and over layer (Hospitals, Freeways, Moores Cancer Center, Labels). This document will emphasize on the code implementation of the above two new features.

Chapter 1 Introduction

HealthWebMapper 1.5 is built upon HealthWebMapper 1.0 keeping its original interface and functionalities. The major change has been made in HealthWebMapper 1.5 includes:

1) Upgrading from Leaflet 0.7.3 to Leaflet 1.3.1
2) Layer control: base maps and over layers

In chapter 2, actual JavaScript code with detailed explanation of the two new features will be presented.

Chapter 2 Code implementation with explanation

2.1 Upgrading from Leaflet 0.7.3 to Leaflet 1.3.1

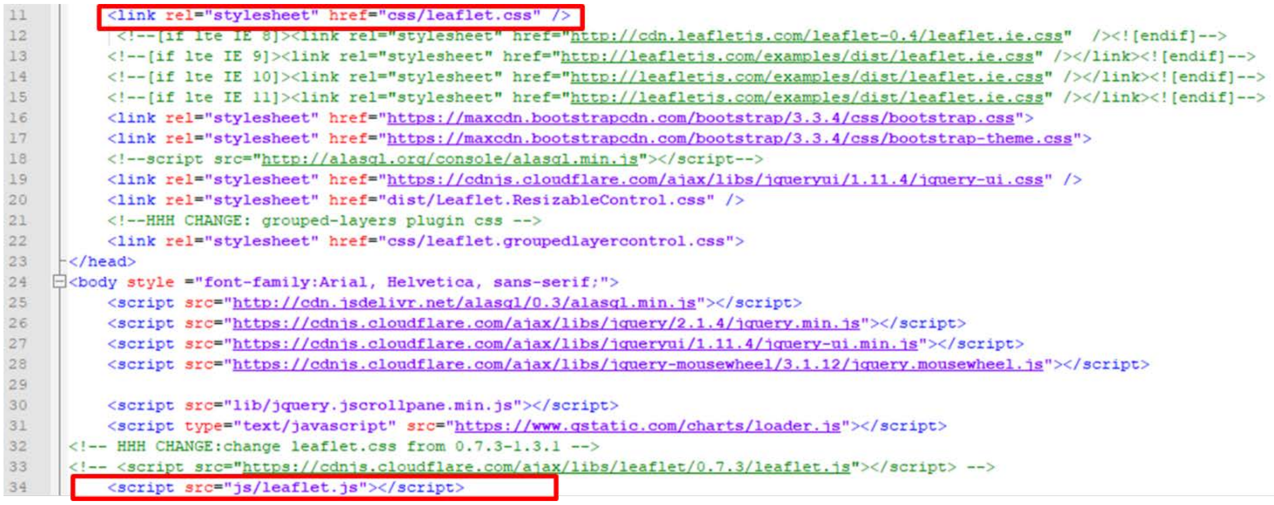
HealthWebMapper 1.0 is built upon Leaflet 0.7.3 but its built-in label function decrepated after Leaflet migrated to 1.0+ and label function has been replaced by built-in function called “tooltip”. For this reason, we decide to upgrade the whole program to Leaflet 1.3.1 (latest version on July 2018).

In order to upgrade the program, you need to follow the following steps:

Step 1: download Leaflet 1.3.1 from https://leafletjs.com/download.html. You can chose a hosted version or downloaded version, in our case, we chose downloaded version because we want to customize the source code (Leaflet.js and Leaflet.css) flexibly according to our needs.

Step 2: unzip the downloaded version of leaflet and put leaflet.js and leaflet.css in js and css folders

Step 3: correctly include these two files in your html file

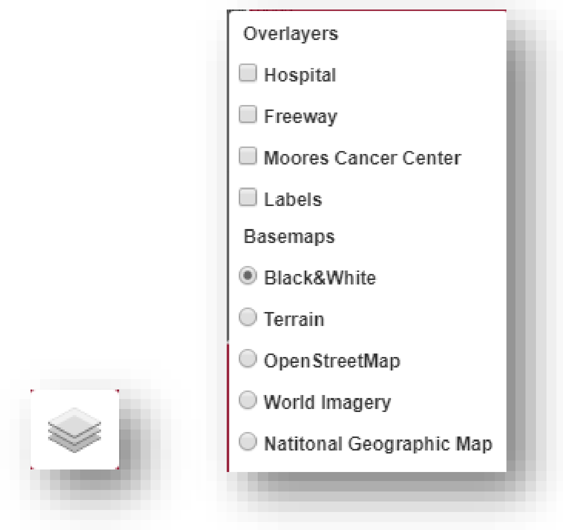


```
    <link rel="stylesheet" href="css/leaflet.css" />
     <!--[if lte IE 8]><link rel="stylesheet" href="http://cdn.leafletjs.com/leaflet-0.4/leaflet.ie.css"  /><![endif]-->
    <!--[if lte IE 9]><link rel="stylesheet" href="http://leafletjs.com/examples/dist/leaflet.ie.css" /></link><![endif]-->
    <!--[if lte IE 10]><link rel="stylesheet" href="http://leafletjs.com/examples/dist/leaflet.ie.css" /></link><![endif]-->
    <!--[if lte IE 11]><link rel="stylesheet" href="http://leafletjs.com/examples/dist/leaflet.ie.css" /></link><![endif]-->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap.css">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap-theme.css">
    <!--script src="http://alasql.org/console/alasql.min.js"></script-->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/jqueryui/1.11.4/jquery-ui.css" />
    <link rel="stylesheet" href="dist/Leaflet.ResizableControl.css" />
    <!--HHH CHANGE: grouped-layers plugin css -->
    <link rel="stylesheet" href="css/leaflet.groupedlayercontrol.css">
</head>
<body style ="font-family:Arial, Helvetica, sans-serif;">
    <script src="http://cdn.jsdelivr.net/alasql/0.3/alasql.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.4/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jqueryui/1.11.4/jquery-ui.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-mousewheel/3.1.12/jquery.mousewheel.js"></script>

    <script src="lib/jquery.jscrollpane.min.js"></script>
    <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<!-- HHH CHANGE:change leaflet.css from 0.7.3-1.3.1 -->
<!-- <script src="https://cdnjs.cloudflare.com/ajax/libs/leaflet/0.7.3/leaflet.js"></script> -->
    <script src="js/leaflet.js"></script>
```

Step 4: debug: when I ran the program after finishing Step 3, there is error message in the browser’s console log. The major problem is that some built-in functions defined in leaflet 0.7+ have changed their names from leaflet 1.0+. So the executor cannot identify the old names in Leaflet 1.0+. Check out leaflet documentation will help solve this problem.

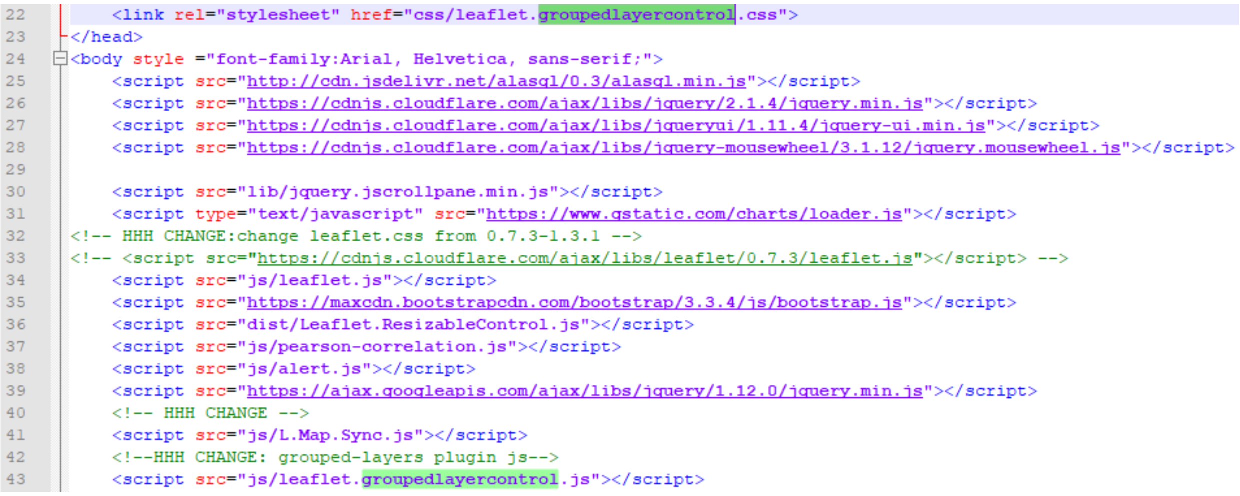
2.2 Layer control: base maps and over layers

Layer control presents choices of base maps and over layers as a default collapsed layer icon which will expand into a list of basemaps and overlayers switchers when users put mouse over the layer icon (see picture below). By default, you can choose only one base map at a time (exclusive choice) while you can add multiple over layers at the same time (checkbox). In HealthWebMapper1.5, we want to group base maps and over layers under their category (Basemaps and Overlayers). Since Leaflet built-in layer control function doesn't support categorization feature so we choose leaflet plugin "groupedlayercontrol" (https://github.com/ismyrnow/leaflet-groupedlayercontrol).



2.2.1 Import plugin

Step 1: correctly include groupedlayercontrol css and js into your html file



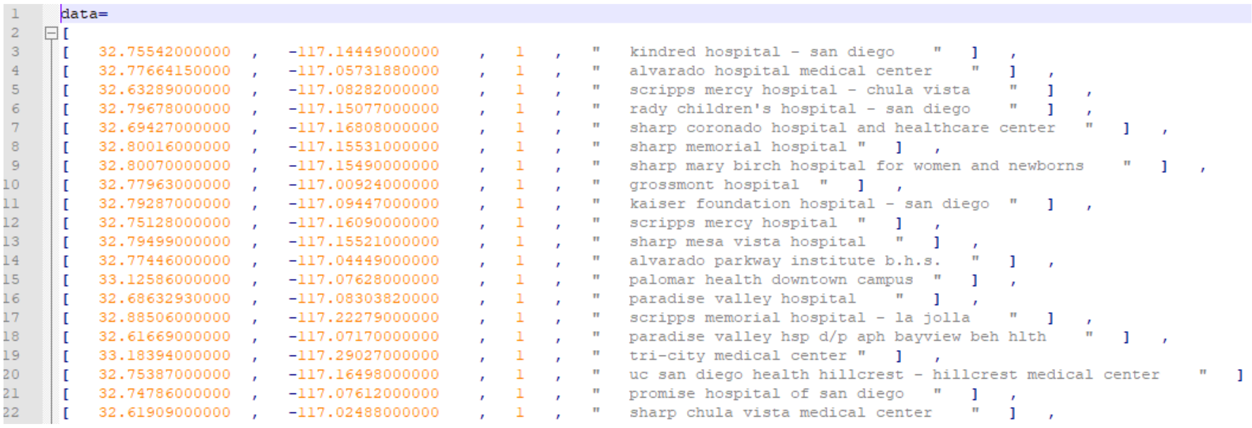
```
    <link rel="stylesheet" href="css/leaflet.groupedlayercontrol.css">
</head>
<body style ="font-family:Arial, Helvetica, sans-serif;">
    <script src="http://cdn.jsdelivr.net/alasql/0.3/alasql.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.4/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jqueryui/1.11.4/jquery-ui.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-mousewheel/3.1.12/jquery.mousewheel.js"></script>

    <script src="lib/jquery.jscrollpane.min.js"></script>
    <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<!-- HHH CHANGE:change leaflet.css from 0.7.3-1.3.1 -->
<!-- <script src="https://cdnjs.cloudflare.com/ajax/libs/leaflet/0.7.3/leaflet.js"></script> -->
    <script src="js/leaflet.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/js/bootstrap.js"></script>
    <script src="dist/Leaflet.ResizableControl.js"></script>
    <script src="js/pearson-correlation.js"></script>
    <script src="js/alert.js"></script>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
    <!-- HHH CHANGE -->
    <script src="js/L.Map.Sync.js"></script>
    <!--HHH CHANGE: grouped-layers plugin js-->
    <script src="js/leaflet.groupedlayercontrol.js"></script>
```

2.2.2 Data preparation and import over layers and base maps

Step 2: process raw data into JSON format for over layers

1) hospital.js

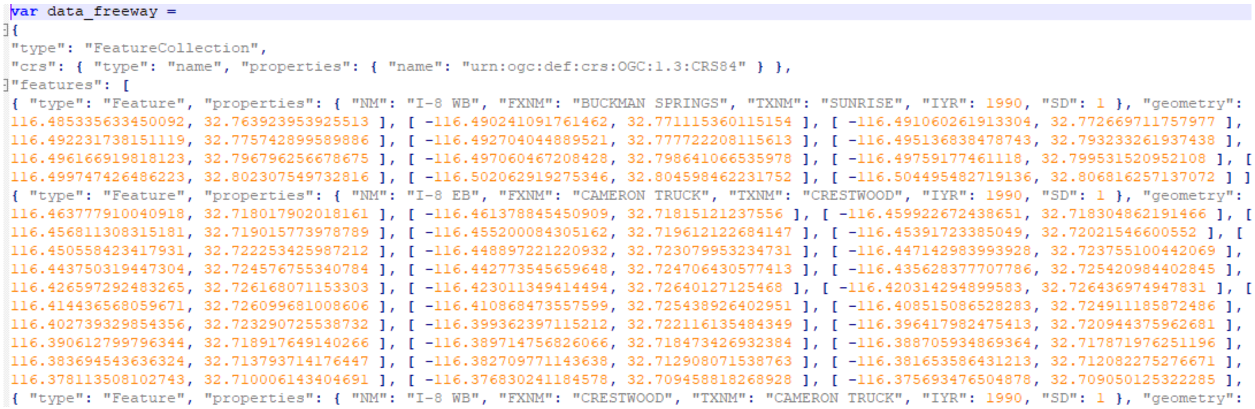
- download hospital shapefile from SANDAG GIS data warehouse (https://www.sandag.org/index.asp?subclassid=100&fuseaction=home.subclasshome)
- use ArcMap to extract geometry and names and make it looks like the following format in csv file
- covert csv to json in http://www.convertcsv.com/csv-to-json.htm
- add “var data = ” to the js file and save it as hospital.js



```
data=
[
[   32.75542000000   ,   -117.14449000000    ,   1   ,   "   kindred hospital - san diego    "   ]   ,
[   32.77664150000   ,   -117.05731880000    ,   1   ,   "   alvarado hospital medical center    "   ]   ,
[   32.63289000000   ,   -117.08282000000    ,   1   ,   "   scripps mercy hospital - chula vista    "   ]   ,
[   32.79678000000   ,   -117.15077000000    ,   1   ,   "   rady children's hospital - san diego    "   ]   ,
[   32.69427000000   ,   -117.16808000000    ,   1   ,   "   sharp coronado hospital and healthcare center    "   ]   ,
[   32.80016000000   ,   -117.15531000000    ,   1   ,   "   sharp memorial hospital "   ]   ,
[   32.80070000000   ,   -117.15490000000    ,   1   ,   "   sharp mary birch hospital for women and newborns    "   ]   ,
[   32.77963000000   ,   -117.00924000000    ,   1   ,   "   grossmont hospital  "   ]   ,
[   32.79287000000   ,   -117.09447000000    ,   1   ,   "   kaiser foundation hospital - san diego  "   ]   ,
[   32.75128000000   ,   -117.16090000000    ,   1   ,   "   scripps mercy hospital  "   ]   ,
[   32.79499000000   ,   -117.15521000000    ,   1   ,   "   sharp mesa vista hospital   "   ]   ,
[   32.77446000000   ,   -117.04449000000    ,   1   ,   "   alvarado parkway institute b.h.s.   "   ]   ,
[   33.12586000000   ,   -117.07628000000    ,   1   ,   "   palomar health downtown campus  "   ]   ,
[   32.68632930000   ,   -117.08303820000    ,   1   ,   "   paradise valley hospital    "   ]   ,
[   32.88506000000   ,   -117.22279000000    ,   1   ,   "   scripps memorial hospital - la jolla    "   ]   ,
[   32.61669000000   ,   -117.07170000000    ,   1   ,   "   paradise valley hsp d/p aph bayview beh hlth    "   ]   ,
[   33.18394000000   ,   -117.29027000000    ,   1   ,   "   tri-city medical center "   ]   ,
[   32.75387000000   ,   -117.16498000000    ,   1   ,   "   uc san diego health hillcrest - hillcrest medical center    "   ]
[   32.74786000000   ,   -117.07612000000    ,   1   ,   "   promise hospital of san diego   "   ]   ,
[   32.61909000000   ,   -117.02488000000    ,   1   ,   "   sharp chula vista medical center    "   ]   ,
```

2) freeway.js

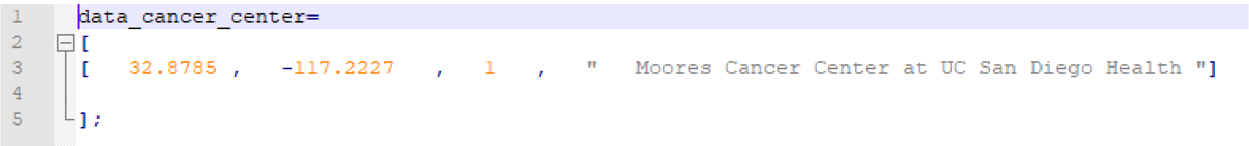
- download freeway shapefile from SANDAG GIS data warehouse (https://www.sandag.org/index.asp?subclassid=100&fuseaction=home.subclasshome)
- use Mapshaper(http://mapshaper.org/) to convert shapefile into GeoJSON format
- add “var data_freeway = ” to the freeway.geojson and save it as freeway.js



```
var data_freeway =
{
"type": "FeatureCollection",
"crs": { "type": "name", "properties": { "name": "urn:ogc:def:crs:OGC:1.3:CRS84" } },
"features": [
{ "type": "Feature", "properties": { "NM": "I-8 WB", "FXNM": "BUCKMAN SPRINGS", "TXNM": "SUNRISE", "IYR": 1990, "SD": 1 }, "geometry":
116.485335633450092, 32.763923953925513 ], [ -116.490241091761462, 32.771115360115154 ], [ -116.491060261913304, 32.772669711757977 ],
116.492231738151119, 32.775742899589886 ], [ -116.492704044889521, 32.777722208115613 ], [ -116.495136838478743, 32.793233261937438 ],
116.496166919818123, 32.796796256678675 ], [ -116.497060467208428, 32.798641066535978 ], [ -116.49759177461118, 32.799531520952108 ], [
116.499747426486223, 32.802307549732816 ], [ -116.502062919275346, 32.804598462231752 ], [ -116.504495482719136, 32.806816257137072 ] ]
{ "type": "Feature", "properties": { "NM": "I-8 EB", "FXNM": "CAMERON TRUCK", "TXNM": "CRESTWOOD", "IYR": 1990, "SD": 1 }, "geometry":
116.463777910040918, 32.718017902018161 ], [ -116.461378845450909, 32.71815121237556 ], [ -116.459922672438651, 32.718304862191466 ], [
116.456811308315181, 32.719015773978789 ], [ -116.455200084305162, 32.719612122684147 ], [ -116.45391723385049, 32.72021546600552 ], [
116.450558423417931, 32.722253425987212 ], [ -116.448897221220932, 32.723079953234731 ], [ -116.447142983993928, 32.723755100442069 ],
116.443750319447304, 32.724576755340784 ], [ -116.442773545659648, 32.724706430577413 ], [ -116.435628377707786, 32.725420984402845 ],
116.426597292483265, 32.726168071153303 ], [ -116.423011349414494, 32.72640127125468 ], [ -116.420314294899583, 32.726436974947831 ], [
116.414436568059671, 32.726099681008606 ], [ -116.410868473557599, 32.725438926402951 ], [ -116.408515086528283, 32.724911185872486 ],
116.402739329854356, 32.723290725538732 ], [ -116.399362397115212, 32.722116135484349 ], [ -116.396417982475413, 32.720944375962681 ],
116.390612799796344, 32.718917649140266 ], [ -116.389714756826066, 32.718473426932384 ], [ -116.388705934869364, 32.717871976251196 ],
116.383694543636324, 32.713793714176447 ], [ -116.382709771143638, 32.712908071538763 ], [ -116.381653586431213, 32.712082275276671 ],
116.378113508102743, 32.710006143404691 ], [ -116.376830241184578, 32.709458818268928 ], [ -116.375693476504878, 32.709050125322285 ],
{ "type": "Feature", "properties": { "NM": "I-8 WB", "FXNM": "CRESTWOOD", "TXNM": "CAMERON TRUCK", "IYR": 1990, "SD": 1 }, "geometry":
```

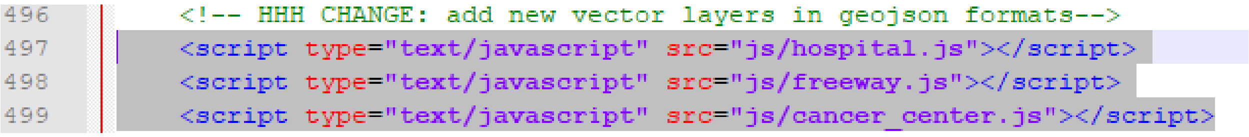
3) Moores Cancer Center

- Since there is only one cancer center in San Diego County, I used the search engine to find out the coordination of Moores Cancer Center
- Repeat the same procedure for hospital.js and get cancer_center.js



```
data_cancer_center=
[
[   32.8785 ,    -117.2227    ,    1    ,    "    Moores Cancer Center at UC San Diego Health "]

];
```

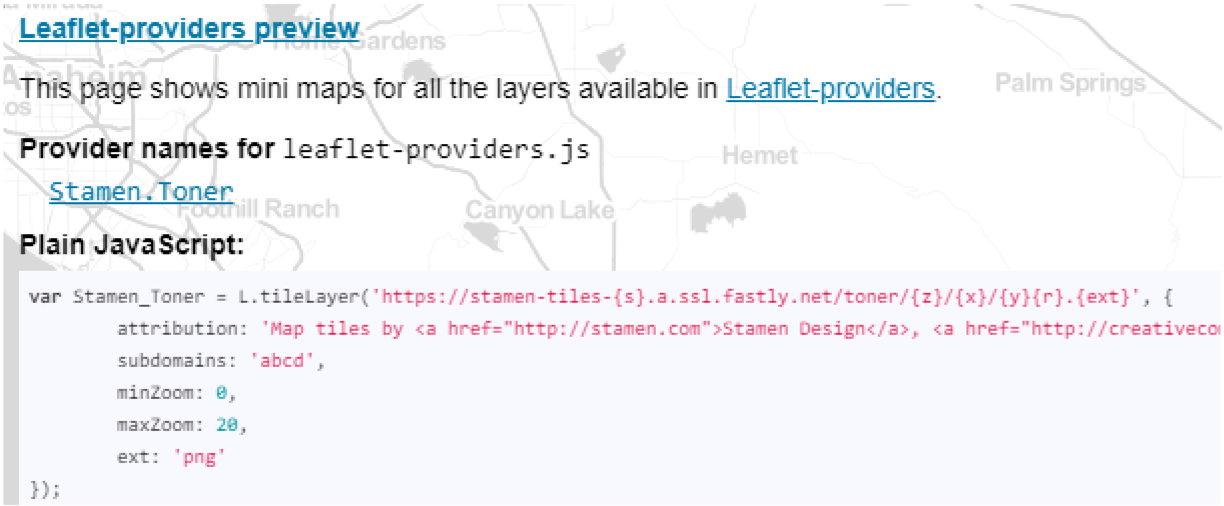
Step 3: include the three js files into html file



```
    <!-- HHH CHANGE: add new vector layers in geojson formats-->
    <script type="text/javascript" src="js/hospital.js"></script>
    <script type="text/javascript" src="js/freeway.js"></script>
    <script type="text/javascript" src="js/cancer_center.js"></script>
```

Step 4: select you desired base map from website (https://leaflet-extras.github.io/leaflet-providers/preview/), in our case, we choose the following six base maps

1) Stamen.Toner (Black&White)



Leaflet-providers preview

This page shows mini maps for all the layers available in Leaflet-providers.

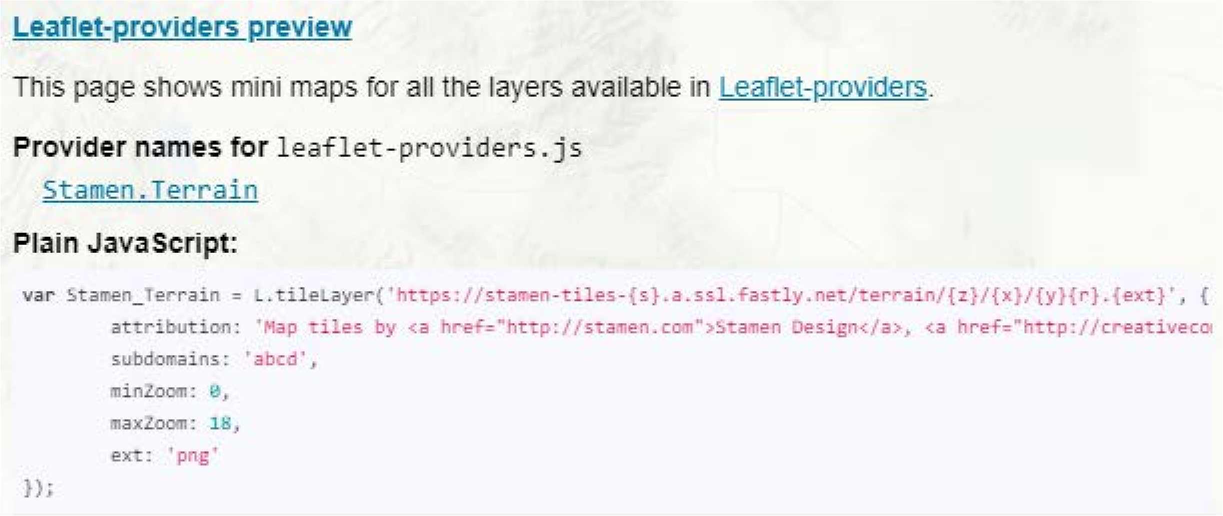
Provider names for `leaflet-providers.js`

`Stamen.Toner`

Plain JavaScript:

```
var Stamen_Toner = L.tileLayer('https://stamen-tiles-{s}.a.ssl.fastly.net/toner/{z}/{x}/{y}{r}.{ext}', {
	attribution: 'Map tiles by <a href="http://stamen.com">Stamen Design</a>, <a href="http://creativeco
	subdomains: 'abcd',
	minZoom: 0,
	maxZoom: 20,
	ext: 'png'
});
```

2) Terrain



Leaflet-providers preview

This page shows mini maps for all the layers available in Leaflet-providers.

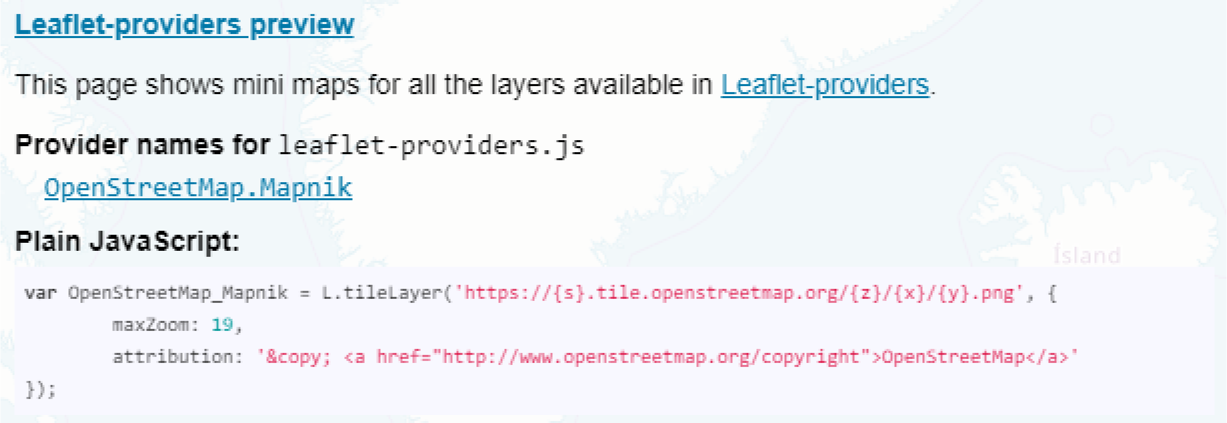
Provider names for `leaflet-providers.js`

`Stamen.Terrain`

Plain JavaScript:

```
var Stamen_Terrain = L.tileLayer('https://stamen-tiles-{s}.a.ssl.fastly.net/terrain/{z}/{x}/{y}{r}.{ext}', {
	attribution: 'Map tiles by <a href="http://stamen.com">Stamen Design</a>, <a href="http://creativeco
	subdomains: 'abcd',
	minZoom: 0,
	maxZoom: 18,
	ext: 'png'
});
```

3) OpenStreetMap.Mapnik(OpenStreetMap)



Leaflet-providers preview

This page shows mini maps for all the layers available in Leaflet-providers.

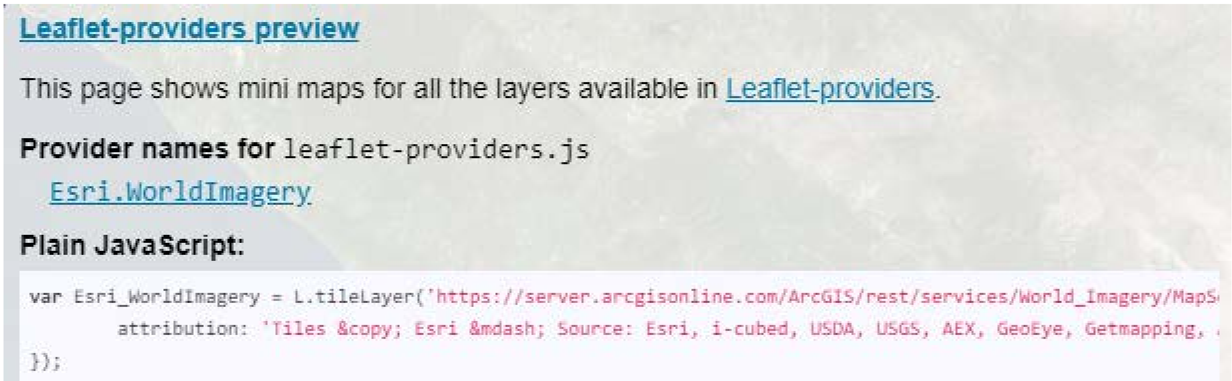
Provider names for `leaflet-providers.js`

OpenStreetMap.Mapnik

Plain JavaScript:

```
var OpenStreetMap_Mapnik = L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
	maxZoom: 19,
	attribution: '&copy; <a href="http://www.openstreetmap.org/copyright">OpenStreetMap</a>'
});
```

4) Esri.World.Imagery



Leaflet-providers preview

This page shows mini maps for all the layers available in Leaflet-providers.

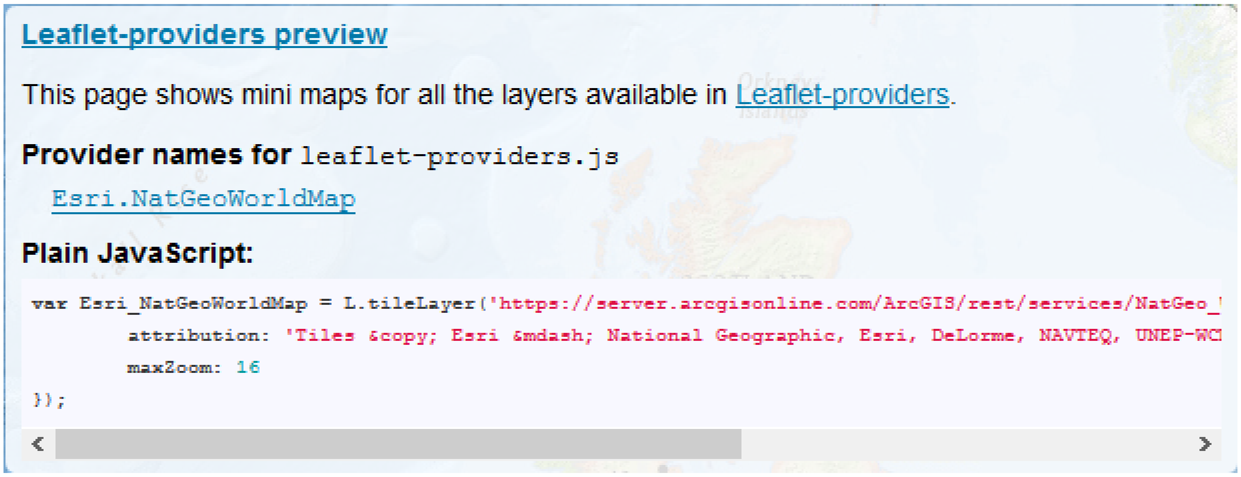
Provider names for `leaflet-providers.js`

Esri.WorldImagery

Plain JavaScript:

```
var Esri_WorldImagery = L.tileLayer('https://server.arcgisonline.com/ArcGIS/rest/services/World_Imagery/MapS
	attribution: 'Tiles &copy; Esri &mdash; Source: Esri, i-cubed, USDA, USGS, AEX, GeoEye, Getmapping,
});
```

5) National Geographic Map



Leaflet-providers preview

This page shows mini maps for all the layers available in Leaflet-providers.

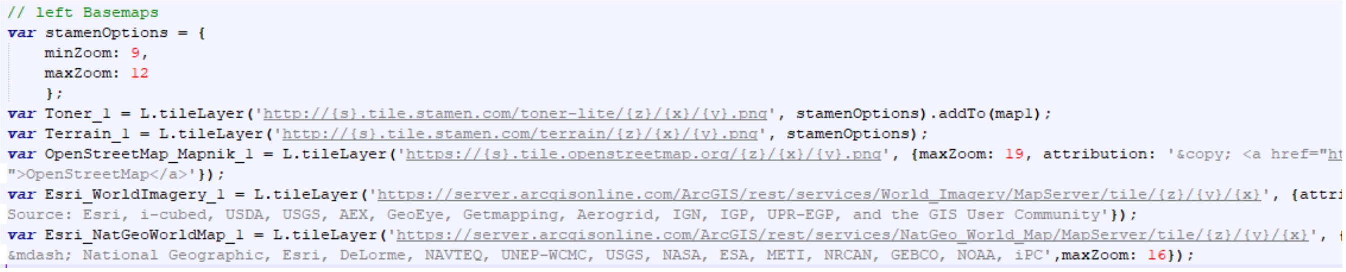
Provider names for `leaflet-providers.js`

Esri.NatGeoWorldMap

Plain JavaScript:

```
var Esri_NatGeoWorldMap = L.tileLayer('https://server.arcgisonline.com/ArcGIS/rest/services/NatGeo_
	attribution: 'Tiles &copy; Esri &mdash; National Geographic, Esri, DeLorme, NAVTEQ, UNEP-WC
	maxZoom: 16
});
```

Step 5: Take left map as example, copy and paste the Plain JavaScript to your html to create base maps tile layers



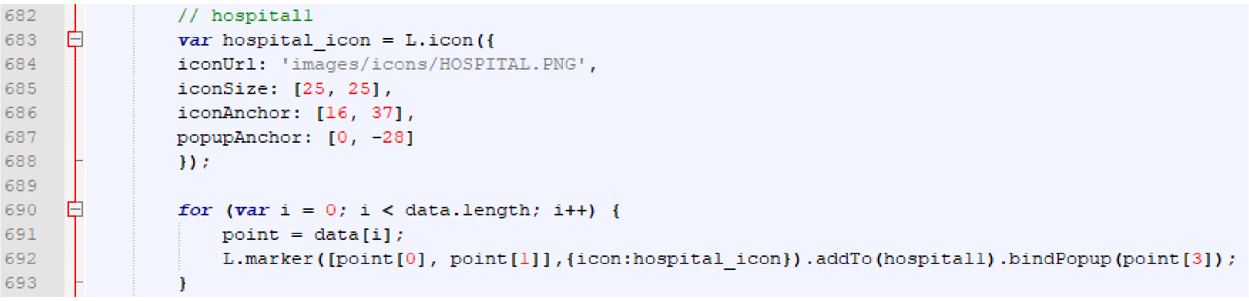
```
// left Basemaps
var stamenOptions = {
    minZoom: 9,
    maxZoom: 12
    };
var Toner_1 = L.tileLayer('http://{s}.tile.stamen.com/toner-lite/{z}/{x}/{y}.png', stamenOptions).addTo(map1);
var Terrain_1 = L.tileLayer('http://{s}.tile.stamen.com/terrain/{z}/{x}/{y}.png', stamenOptions);
var OpenStreetMap_Mapnik_1 = L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {maxZoom: 19, attribution: '&copy; <a href="ht
">OpenStreetMap</a>'});
var Esri_WorldImagery_1 = L.tileLayer('https://server.arcgisonline.com/ArcGIS/rest/services/World_Imagery/MapServer/tile/{z}/{y}/{x}', {attri
Source: Esri, i-cubed, USDA, USGS, AEX, GeoEye, Getmapping, Aerogrid, IGN, IGP, UPR-EGP, and the GIS User Community'});
var Esri_NatGeoWorldMap_1 = L.tileLayer('https://server.arcgisonline.com/ArcGIS/rest/services/NatGeo_World_Map/MapServer/tile/{z}/{y}/{x}', {
&mdash; National Geographic, Esri, DeLorme, NAVTEQ, UNEP-WCMC, USGS, NASA, ESA, METI, NRCAN, GEBCO, NOAA, iPC',maxZoom: 16});
```

2.2.3 Create over layers

Step 5: create variables Layer Group for hospital, cancer center and freeway

Step 6: hospitals layer:

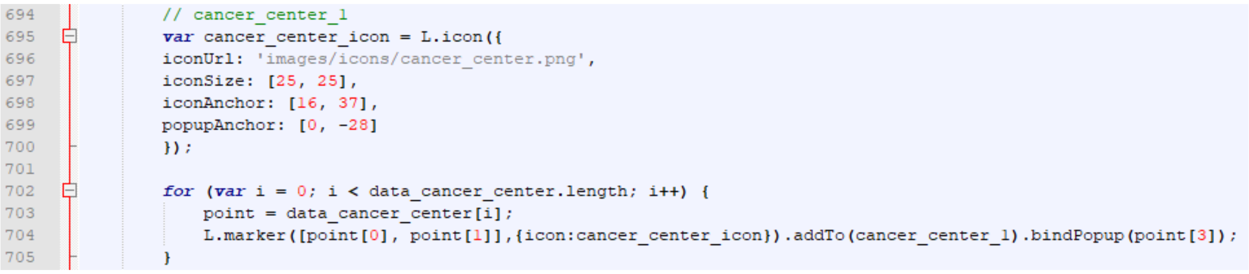
- Use L.icon to link your icon image to hospital layer
- Loop through all the points in hospitals.js to make them markers with name popup and customized icon.



```
        // hospital1
        var hospital_icon = L.icon({
        iconUrl: 'images/icons/HOSPITAL.PNG',
        iconSize: [25, 25],
        iconAnchor: [16, 37],
        popupAnchor: [0, -28]
        });

        for (var i = 0; i < data.length; i++) {
            point = data[i];
            L.marker([point[0], point[1]],{icon:hospital_icon}).addTo(hospital1).bindPopup(point[3]);
        }
```

Step 7: Moores Cancer Center layer:

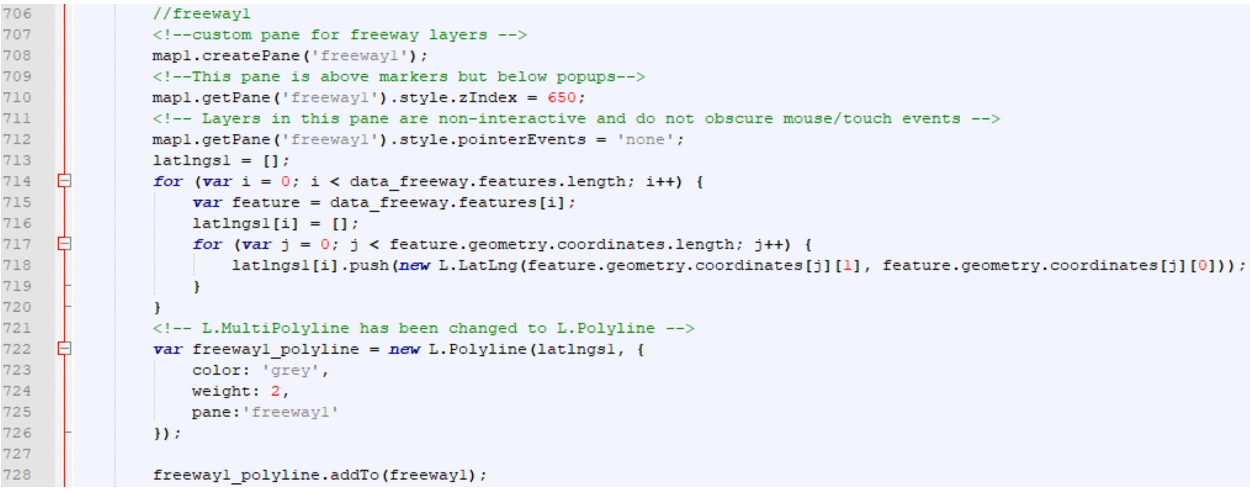
- Repeated the same procedure for hospital layer



```
        // cancer_center_1
        var cancer_center_icon = L.icon({
        iconUrl: 'images/icons/cancer_center.png',
        iconSize: [25, 25],
        iconAnchor: [16, 37],
        popupAnchor: [0, -28]
        });

        for (var i = 0; i < data_cancer_center.length; i++) {
            point = data_cancer_center[i];
            L.marker([point[0], point[1]],{icon:cancer_center_icon}).addTo(cancer_center_1).bindPopup(point[3]);
        }
```

Step 8: Freeways layer:

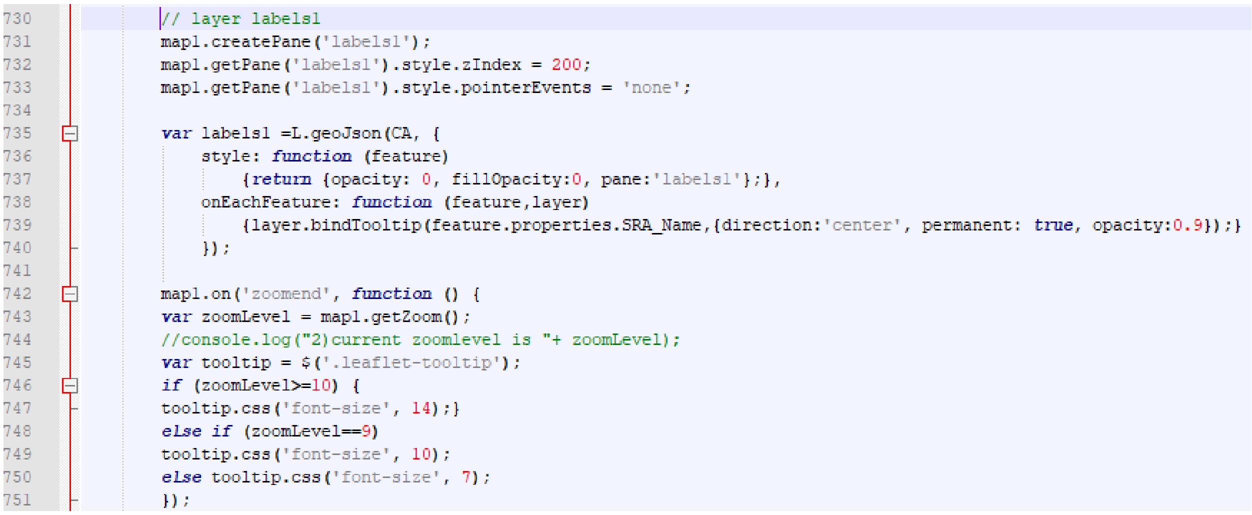
- Use Pane to set freeway way layer above markers but below popups to avoid conflict with bring-to-front function for each polygon
- Loop through freeway.js file and use L.Polyline to add the polyline layer



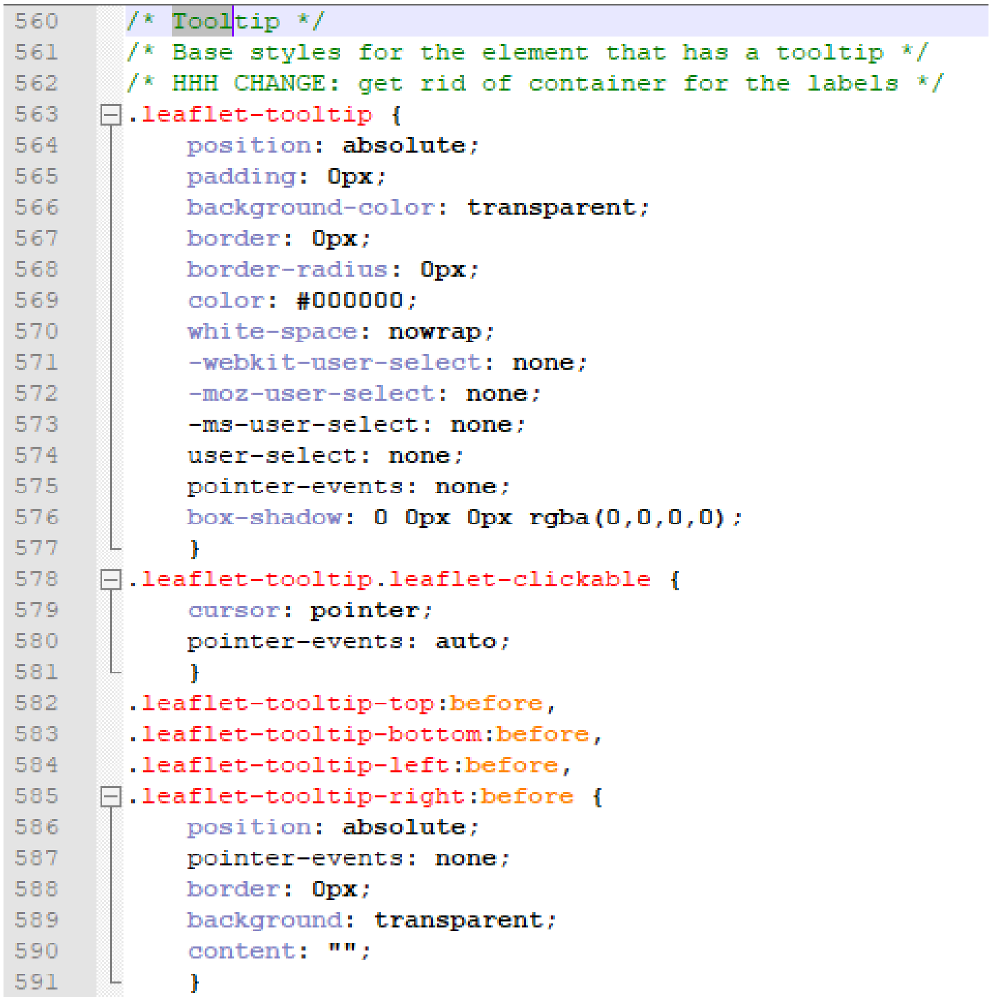
```
//freeway1
<!--custom pane for freeway layers -->
map1.createPane('freeway1');
<!--This pane is above markers but below popups-->
map1.getPane('freeway1').style.zIndex = 650;
<!-- Layers in this pane are non-interactive and do not obscure mouse/touch events -->
map1.getPane('freeway1').style.pointerEvents = 'none';
latlngs1 = [];
for (var i = 0; i < data_freeway.features.length; i++) {
    var feature = data_freeway.features[i];
    latlngs1[i] = [];
    for (var j = 0; j < feature.geometry.coordinates.length; j++) {
        latlngs1[i].push(new L.LatLng(feature.geometry.coordinates[j][1], feature.geometry.coordinates[j][0]));
    }
}
<!-- L.MultiPolyline has been changed to L.Polyline -->
var freeway1_polyline = new L.Polyline(latlngs1, {
    color: 'grey',
    weight: 2,
    pane:'freeway1'
});

freeway1_polyline.addTo(freeway1);
```

4) Labels

- Use Pane to set the zIndex of label layer to avoid conflict with other layers
- Use L.geoJson to import polygon layer but styling them as invisible since we only want a layer with label. In onEachFeature function, pull out properties SRA_Name and bind them into centers of each polygon and set it as permanent.
- Customize leaflet.css to make the default container and background for the tooltip invisible.

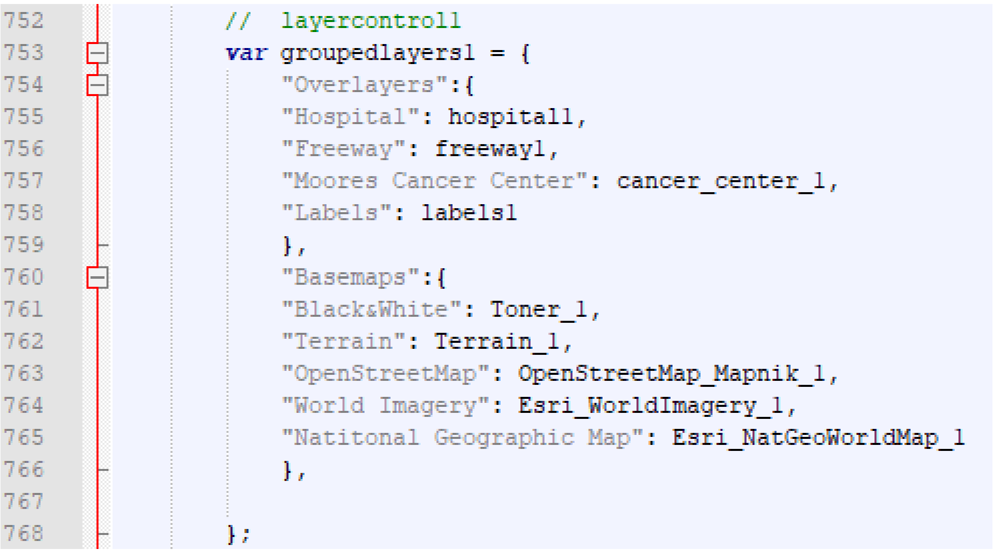


```
// layer labels1
map1.createPane('labels1');
map1.getPane('labels1').style.zIndex = 200;
map1.getPane('labels1').style.pointerEvents = 'none';

var labels1 =L.geoJson(CA, {
    style: function (feature)
        {return {opacity: 0, fillOpacity:0, pane:'labels1'};},
    onEachFeature: function (feature,layer)
        {layer.bindTooltip(feature.properties.SRA_Name,{direction:'center', permanent: true, opacity:0.9});}
    });

map1.on('zoomend', function () {
var zoomLevel = map1.getZoom();
//console.log("2)current zoomlevel is "+ zoomLevel);
var tooltip = $('.leaflet-tooltip');
if (zoomLevel>=10) {
tooltip.css('font-size', 14);}
else if (zoomLevel==9)
tooltip.css('font-size', 10);
else tooltip.css('font-size', 7);
});
```



```
/* Tooltip */
/* Base styles for the element that has a tooltip */
/* HHH CHANGE: get rid of container for the labels */
.leaflet-tooltip {
    position: absolute;
    padding: 0px;
    background-color: transparent;
    border: 0px;
    border-radius: 0px;
    color: #000000;
    white-space: nowrap;
    -webkit-user-select: none;
    -moz-user-select: none;
    -ms-user-select: none;
    user-select: none;
    pointer-events: none;
    box-shadow: 0 0px 0px rgba(0,0,0,0);
    }
.leaflet-tooltip.leaflet-clickable {
    cursor: pointer;
    pointer-events: auto;
    }
.leaflet-tooltip-top:before,
.leaflet-tooltip-bottom:before,
.leaflet-tooltip-left:before,
.leaflet-tooltip-right:before {
    position: absolute;
    pointer-events: none;
    border: 0px;
    background: transparent;
    content: "";
    }
```

2.2.3 Layer control



```
//  layercontroll
var groupedlayersl = {
    "Overlayers":{
    "Hospital": hospitall,
    "Freeway": freewayl,
    "Moores Cancer Center": cancer_center_l,
    "Labels": labelsl
    },
    "Basemaps":{
    "Black&White": Toner_l,
    "Terrain": Terrain_l,
    "OpenStreetMap": OpenStreetMap_Mapnik_l,
    "World Imagery": Esri_WorldImagery_l,
    "Natitonal Geographic Map": Esri_NatGeoWorldMap_l
    },

};
```

* repeat above process for right maps.