

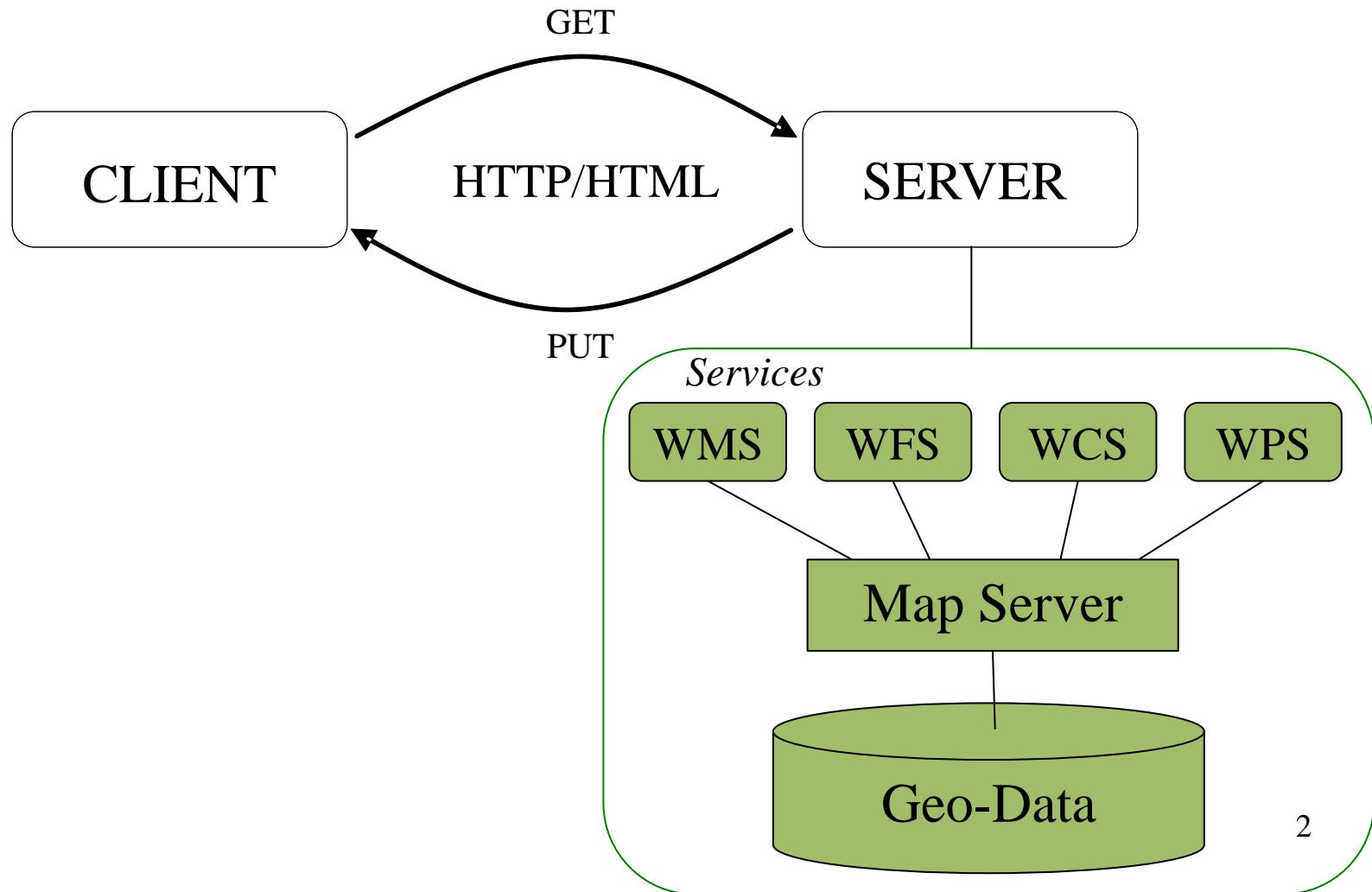


Thin Web Mapping Clients

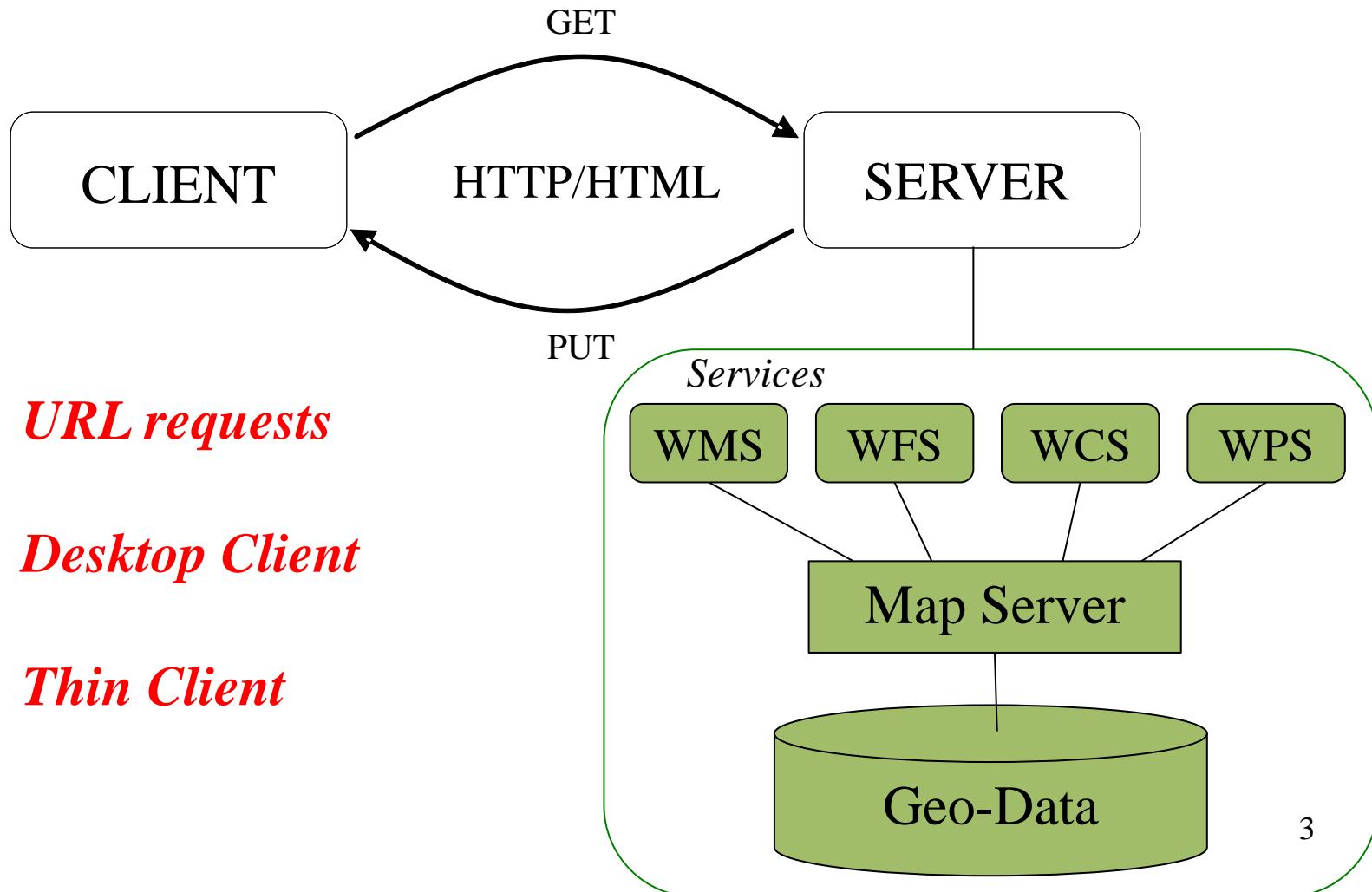
Emmanuel Stefanakis

estef@unb.ca

How to Access a Map Server ?



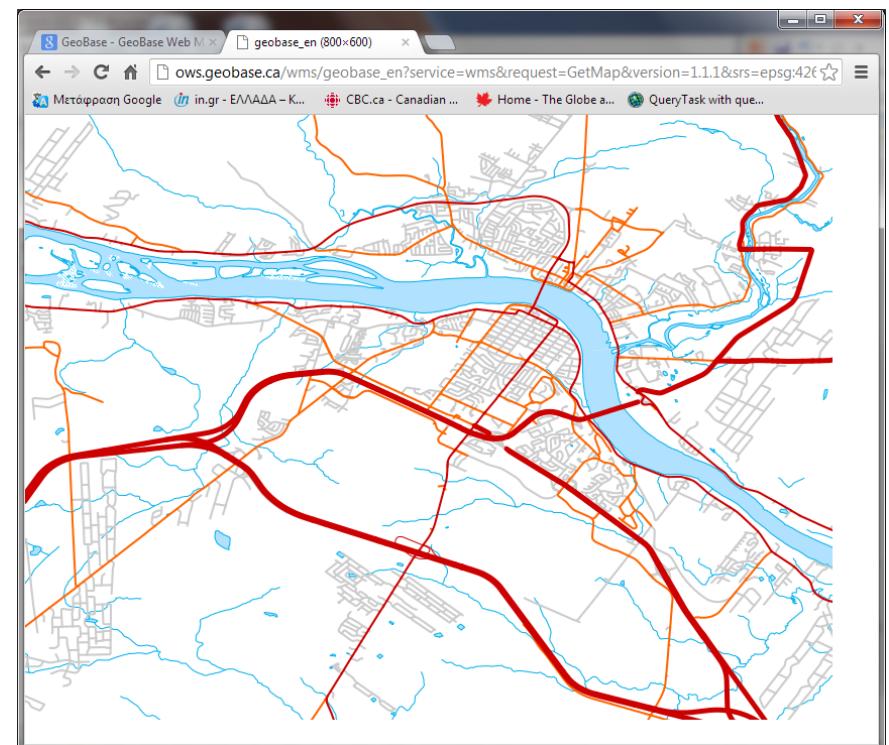
How to Access a Map Server ?



How to Access a Map Server ?

- Option 1...
 - Type requests on the web browser...

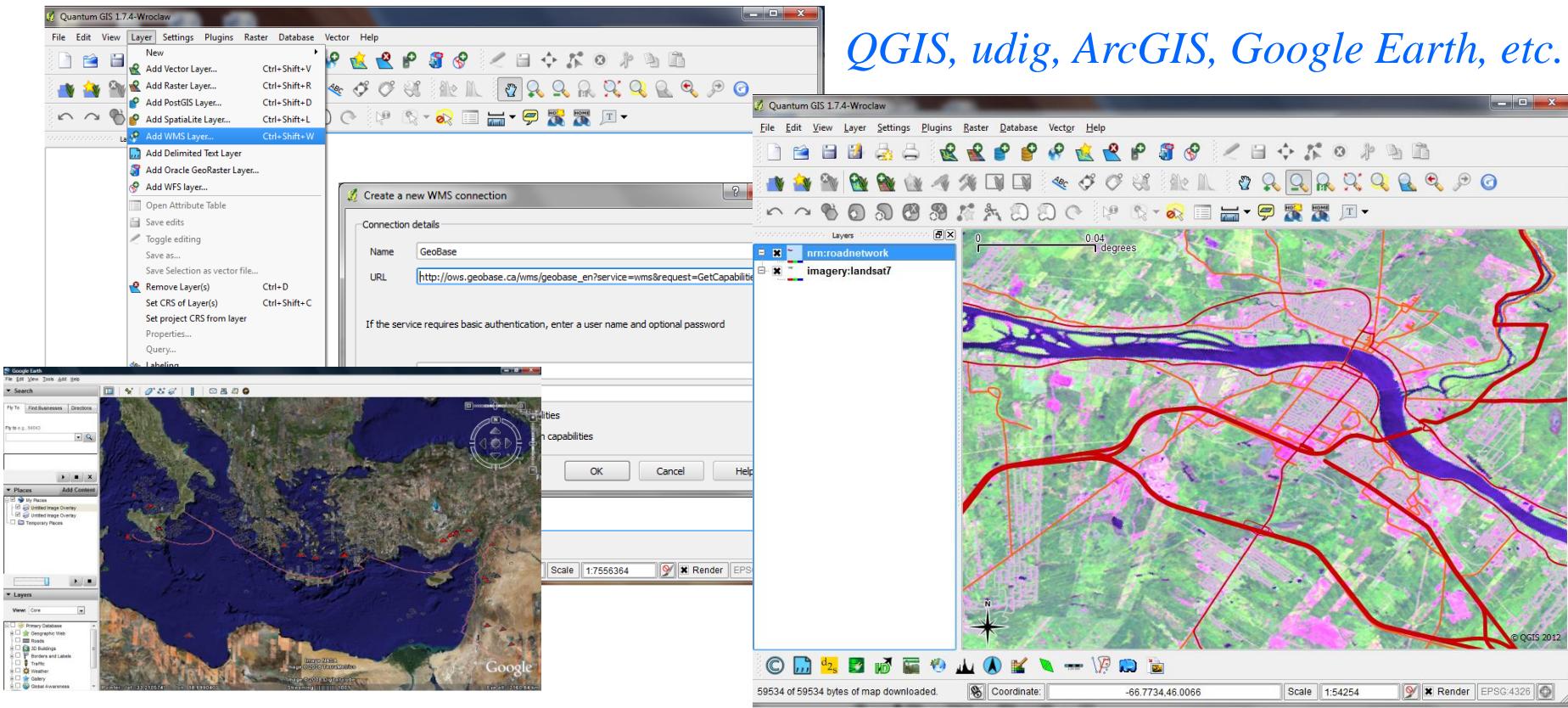
```
http://ows.geobase.ca/wms/geobase_en?  
service=wms  
&request=GetMap  
&version=1.1.1  
&srs=epsg:4269  
&bbox=-66.77,45.87,-66.57,46.01  
&width=800&height=600  
&style=&format=image/png  
&layers=nhn:hydrography,nrn:roadnetwork
```



How to Access a Map Server ?

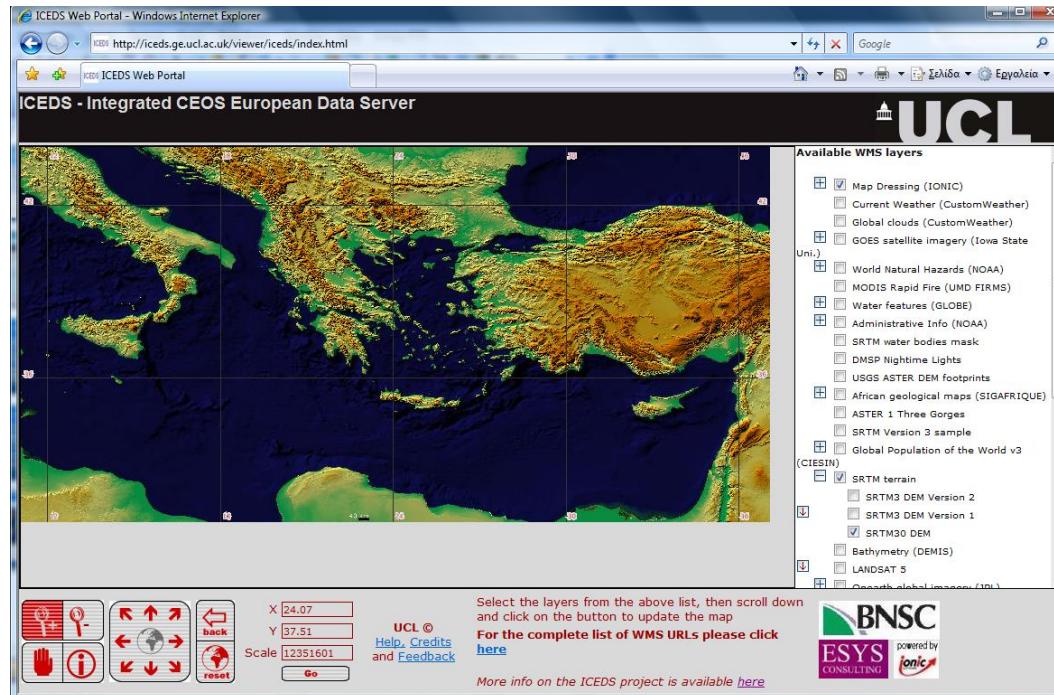
- Option 2...
 - Use a desktop (thick) client...

QGIS, udig, ArcGIS, Google Earth, etc.



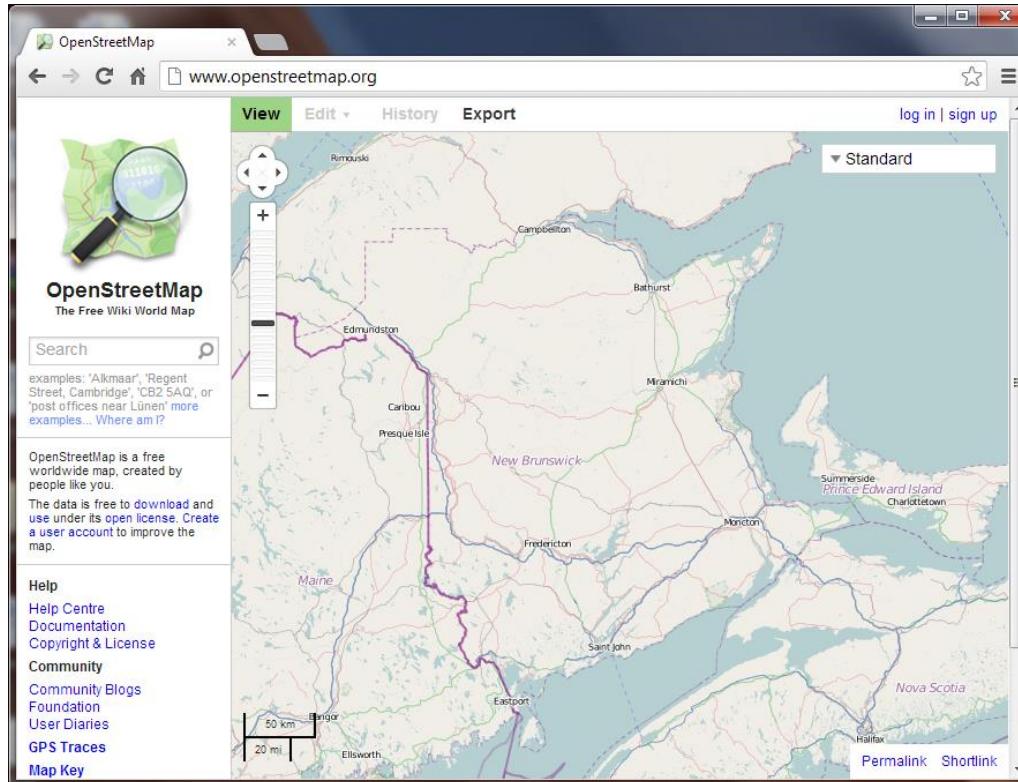
How to Access a Map Server ?

- Option 3...
 - Use a **thin** client...
 - an application that runs within the web browser



How to Access a Map Server ?

- Option 3... <http://www.openstreetmap.org/>
 - Use a **thin** client... built in OpenLayers



Thin Client

- Advantages...
 - all computers have a web browser installed
 - not any installation process for the end user to do
 - most users are comfortable working in a web environment
 - recent developments in AJAX (Asynchronous JavaScript and XML) systems offer many of the advantages of a desktop client

Thin Client

- How to develop it?
 - programming from scratch
 - e.g., html + javascripts
 - using a pre-built framework or library
 - e.g., LeafLet, OpenLayers, GeoMoose, MapBuilder, etc.

Thin Client

- Use a pre-built framework/library...
 - Steps to produce an application...
 - define scope and requirements of client
 - choose software framework/library
 - install software / library (optional)
 - construct client application
 - test client application

OpenLayers

<http://openlayers.org>

A screenshot of a web browser displaying the OpenLayers website at openlayers.org. The page features a large map of Europe with blue oceans and brown landmasses. Below the map, a banner text reads: *A high-performance, feature-packed library for all your mapping needs.* On the left, there's a "LATEST" section with an RSS icon and the text: "OpenLayers v6.5.0 is here! Check out the [docs](#) and the [examples](#) to get started. The full distribution can be downloaded from the [release page](#)." At the bottom left, there's an "OVERVIEW" section with the following text: "OpenLayers makes it easy to put a dynamic map in any web page. It can display map tiles, vector data and markers loaded from any source. OpenLayers has been developed to further the use of geographic information of all kinds. It is completely free, Open Source JavaScript, released under the [2-clause BSD License](#) (also known as the FreeBSD)."

LeafLet

<https://leafletjs.com/>

The screenshot shows the homepage of the Leaflet website. At the top, there's a dark header bar with the Leaflet logo (a green leaf icon) and the text "Leaflet - a JavaScript library for in". Below the header is a browser-style navigation bar with back, forward, and search icons, and the URL "leafletjs.com". The main content area features the Leaflet logo in a large, stylized, grey font, with a green leaf icon integrated into the letter "t". Below the logo, the text "an open-source JavaScript library for mobile-friendly interactive maps" is displayed. To the right of this text are three social media sharing icons: Twitter (blue), GitHub (black), and a speech bubble (green). At the bottom of the main content area, there's a horizontal menu with links: "Overview", "Tutorials", "Docs", "Download", "Plugins", and "Blog". A light gray sidebar on the left contains the text "Sep 4, 2020 — Leaflet 1.7.1 has been released!". The main content area also contains descriptive text about Leaflet's features and design philosophy.

Sep 4, 2020 — Leaflet 1.7.1 has been released!

Leaflet is the leading open-source JavaScript library for mobile-friendly interactive maps. Weighing just about 39 KB of JS, it has all the mapping [features](#) most developers ever need.

Leaflet is designed with *simplicity*, *performance* and *usability* in mind. It works efficiently across all major desktop and mobile platforms, can be extended with lots of [plugins](#), has a beautiful, easy to use and [well-documented API](#) and a simple, readable [source code](#) that is a joy to [contribute](#) to.

LaefLet

- What is it ?
 - a pure JavaScript library
 - for displaying map data in most modern web browsers
 - with no server-side dependencies
 - it implements a JavaScript API
 - for building rich web-based geographic applications
 - similar to the Google Maps and MSN Virtual Earth APIs
 - it is Open and Free Software
 - OSGeo Project

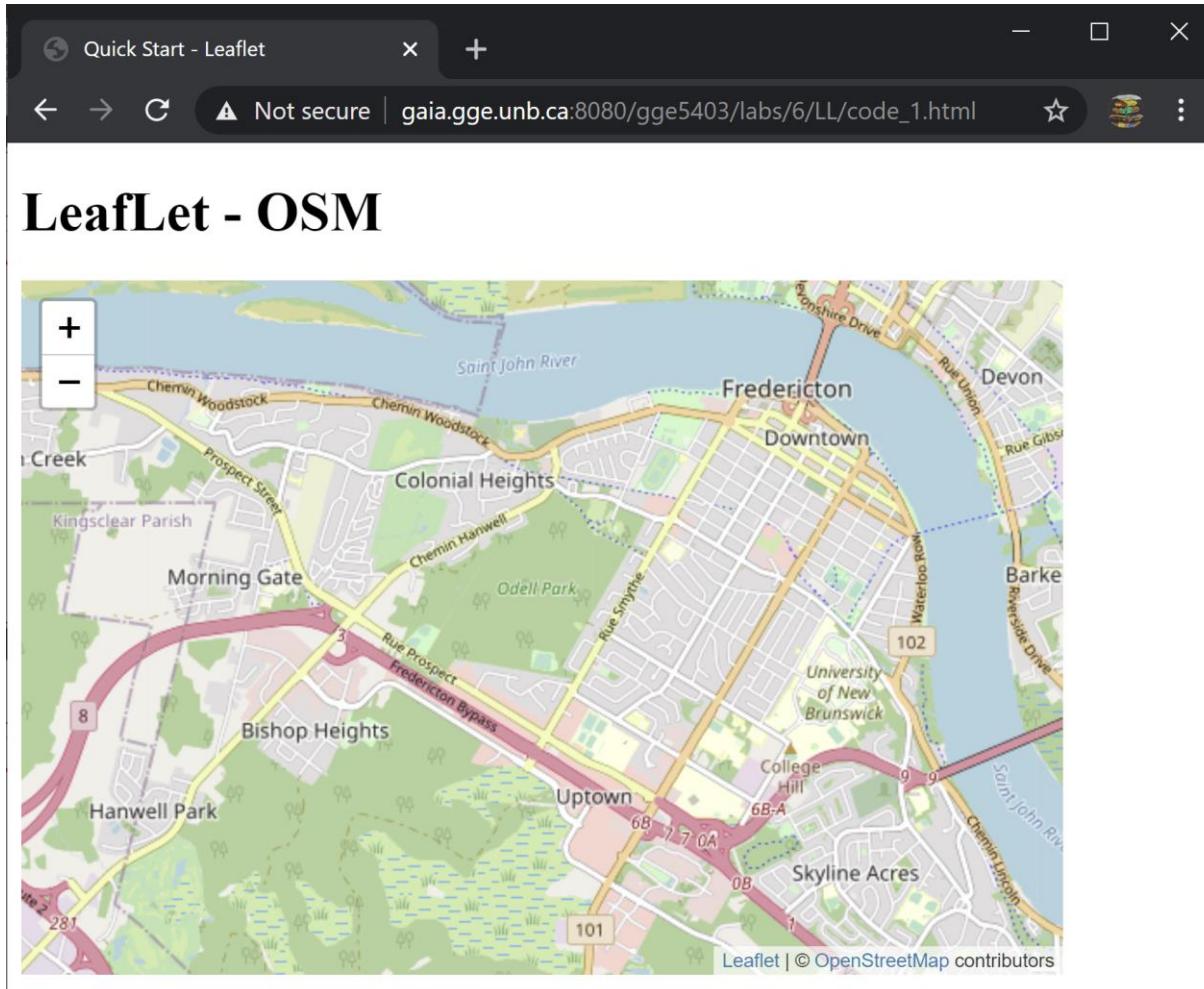
LeafLet Examples

- Simple LeafLet JS examples: (VPN needed)
 - <http://gaia.gge.unb.ca:8080/gge5403/labs/6/LL/>

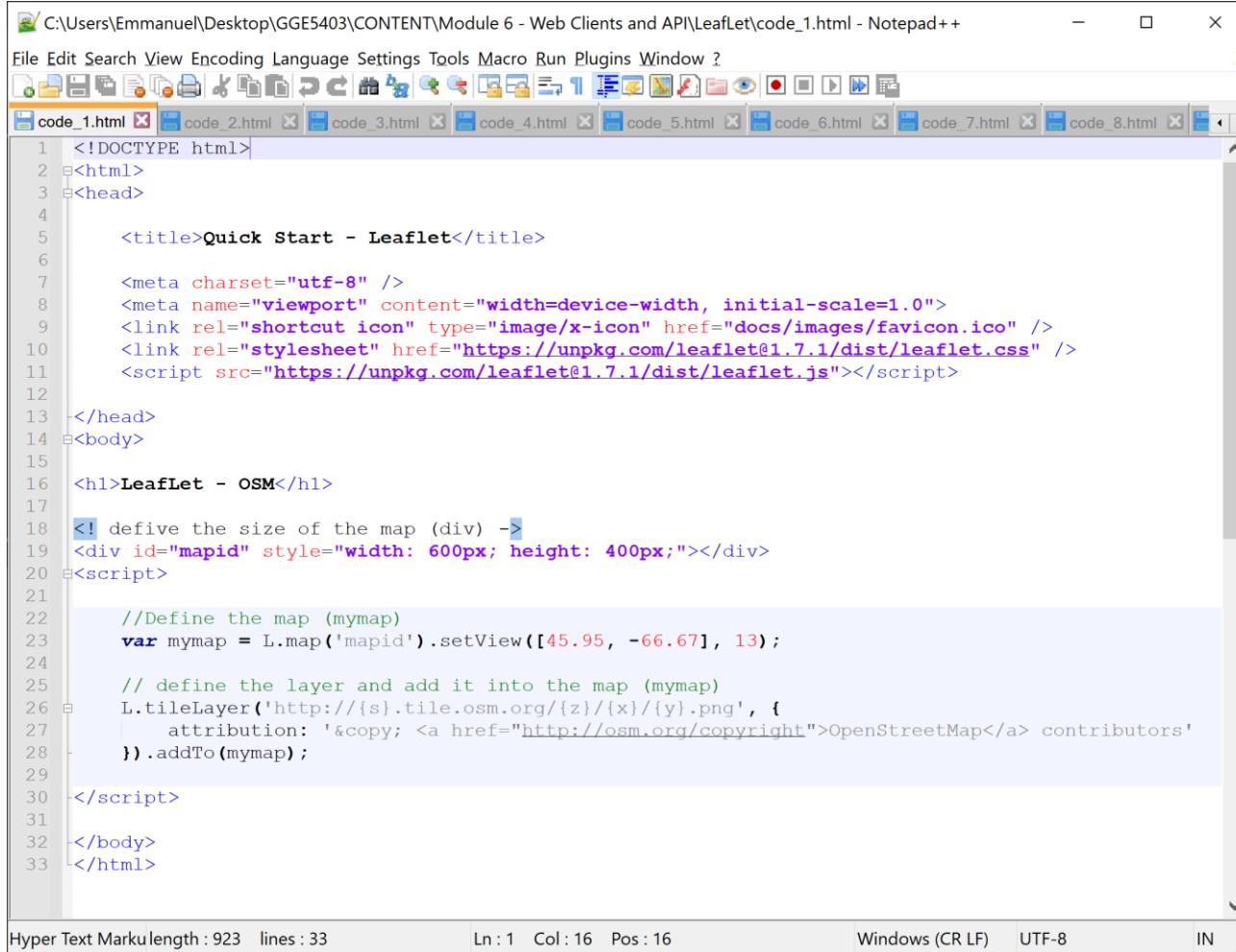
The screenshot shows a web browser window with the title "Index of /gge5403/labs/6/LL". The address bar indicates the URL is "gaia.gge.unb.ca:8080/gge5403/labs/6/LL/". The page content is titled "Index of /gge5403/labs/6/LL" and displays a table of files and directories:

	Name	Last modified	Size	Description
	Parent Directory		-	
	Leaflet.Geodesic-mas..>	2021-04-06 14:33	126K	
	Leaflet.Geodesic-mas..>	2021-04-06 14:34	-	
	code_1.html	2021-04-06 12:51	923	
	code_2.html	2021-04-06 12:51	1.0K	
	code_3.html	2021-04-06 12:51	1.5K	
	code_4.html	2021-04-06 12:51	1.2K	
	code_5.html	2021-04-06 12:57	1.2K	
	code_6.html	2021-04-06 12:58	1.6K	
	code_7.html	2021-04-06 14:15	2.2K	
	code_8.html	2021-04-06 14:28	2.5K	
	code_9.html	2021-04-06 14:30	1.3K	

LeafLet Examples



LeafLet Examples

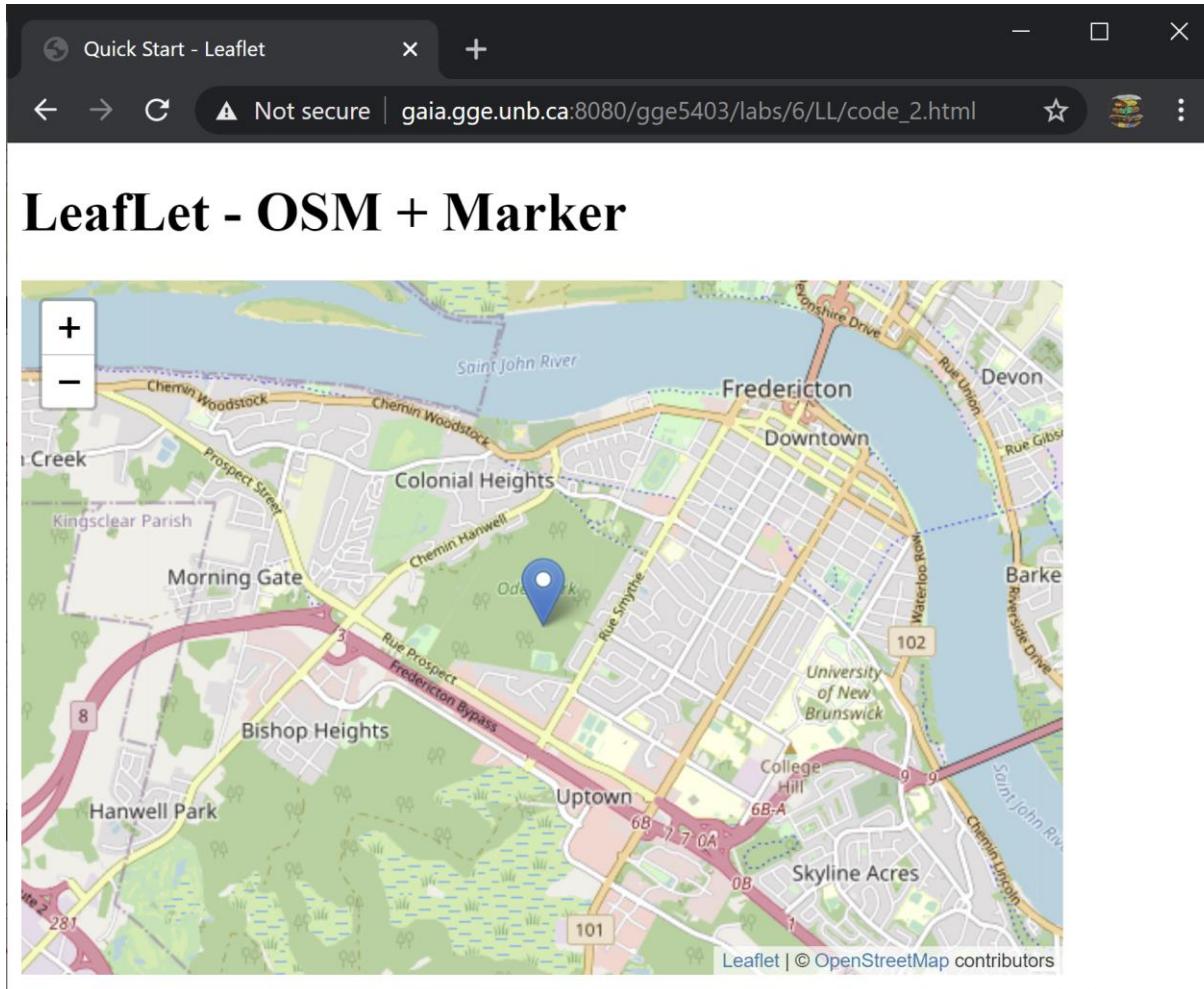


The screenshot shows a Notepad++ window displaying a file named 'code_1.html'. The window title bar reads 'C:\Users\Emmanuel\Desktop\GGE5403\CONTENT\Module 6 - Web Clients and API\Leaflet\code_1.html - Notepad++'. The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, and Help. The toolbar below the menu has various icons for file operations like Open, Save, Find, Copy, Paste, etc. The code editor pane contains the following HTML and JavaScript code:

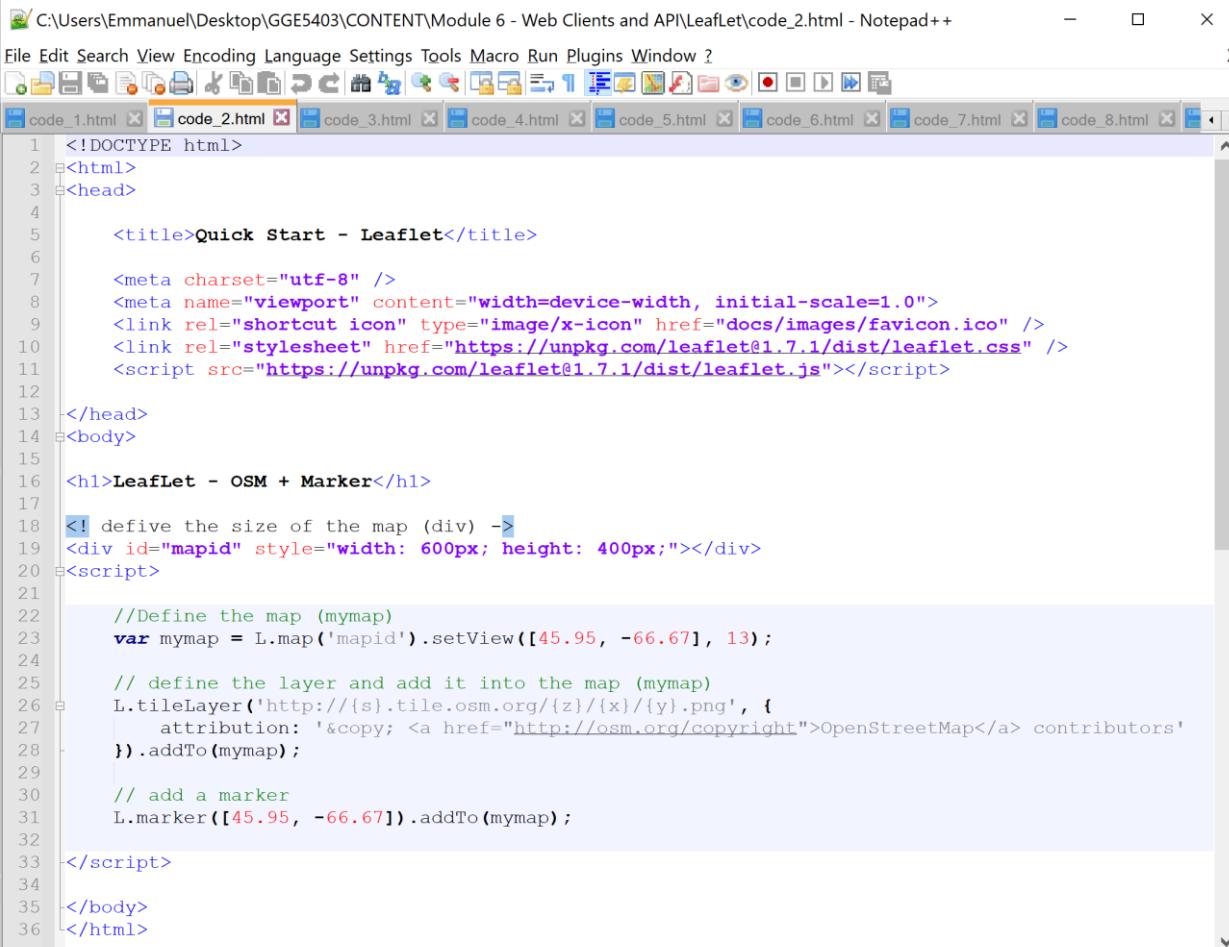
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4
5     <title>Quick Start - Leaflet</title>
6
7     <meta charset="utf-8" />
8     <meta name="viewport" content="width=device-width, initial-scale=1.0">
9     <link rel="shortcut icon" type="image/x-icon" href="docs/images/favicon.ico" />
10    <link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css" />
11    <script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"></script>
12
13 </head>
14 <body>
15
16     <h1>LeafLet - OSM</h1>
17
18     <! define the size of the map (div) ->
19     <div id="mapid" style="width: 600px; height: 400px;"></div>
20
21     <script>
22
23         //Define the map (mymap)
24         var mymap = L.map('mapid').setView([45.95, -66.67], 13);
25
26         // define the layer and add it into the map (mymap)
27         L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png', {
28             attribution: '&copy; <a href="http://osm.org/copyright">OpenStreetMap</a> contributors',
29         }).addTo(mymap);
30
31     </script>
32
33 </body>
</html>
```

The status bar at the bottom of the Notepad++ window displays: 'Hyper Text Markup length : 923 lines : 33' and 'Ln : 1 Col : 16 Pos : 16'. It also shows encoding 'Windows (CR LF)', character set 'UTF-8', and input mode 'IN'.

LeafLet Examples



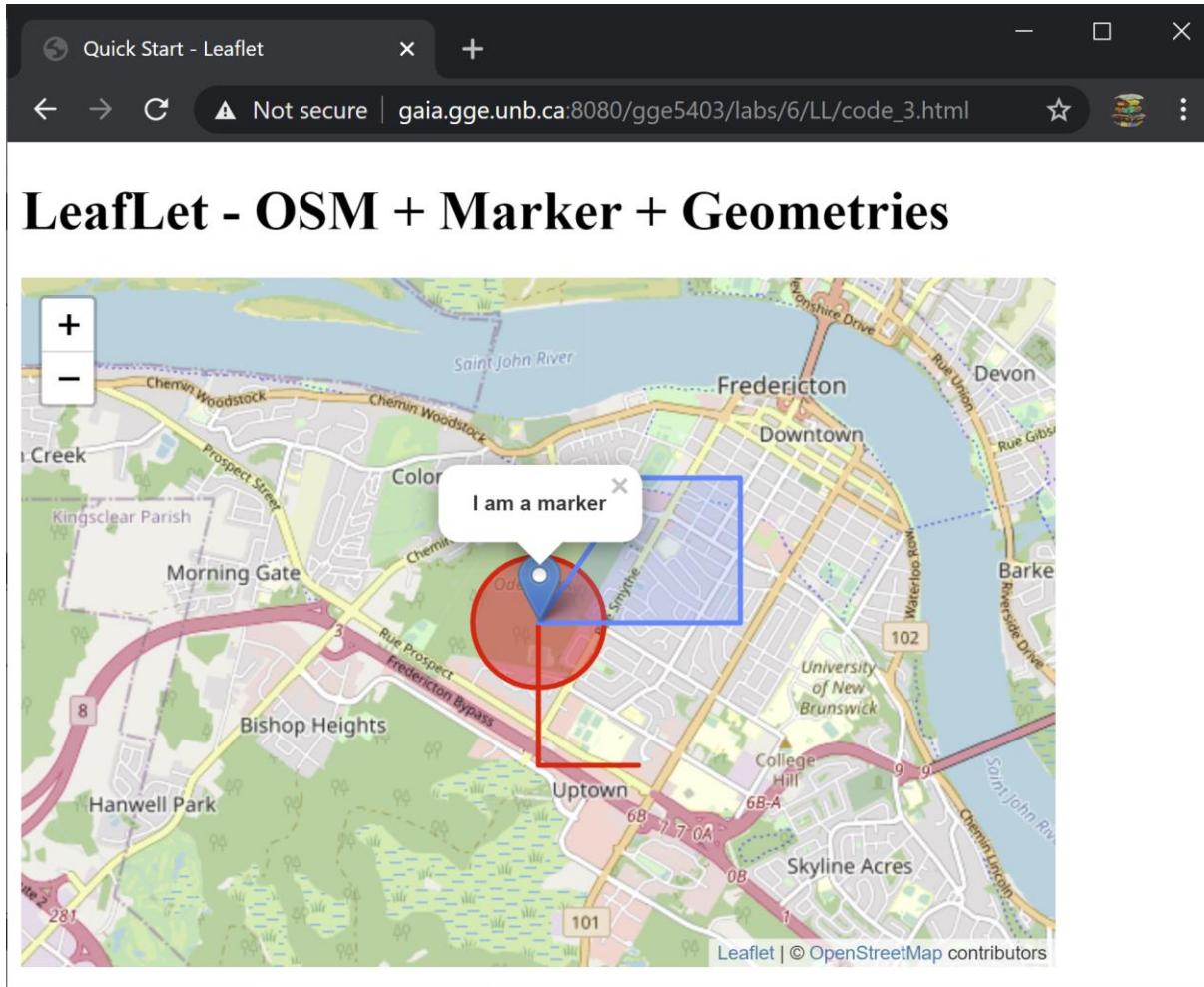
LeafLet Examples



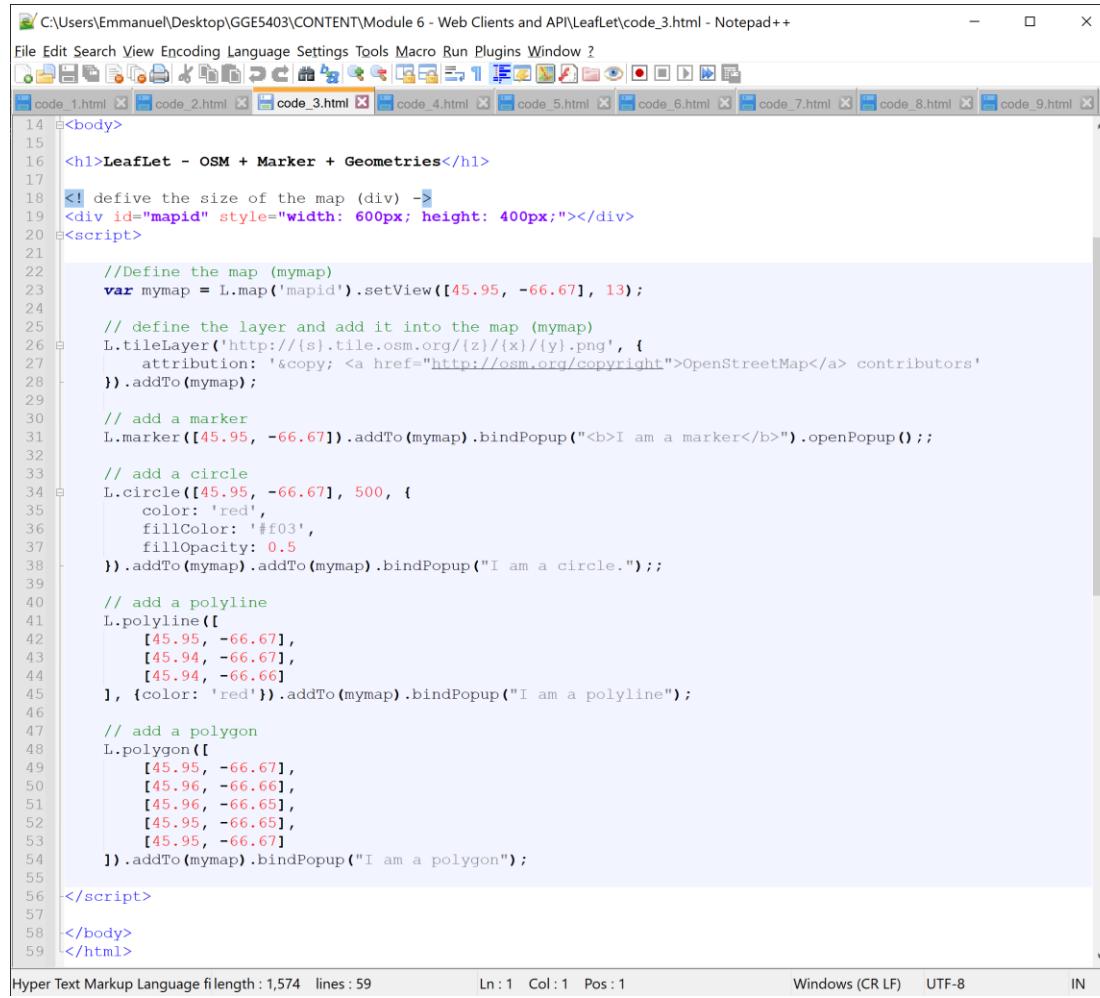
The screenshot shows a Notepad++ window displaying a HTML file named 'code_2.html'. The file contains code for a Leaflet map. The code includes DOCTYPE, head (with title 'Quick Start - Leaflet', meta charset, viewport, and links to favicon and CSS/JS files), body (with h1 'LeafLet - OSM + Marker', a map div with id 'mapid' and style 'width: 600px; height: 400px;', and a script block defining a map 'mymap' with view [45.95, -66.67] at level 13, adding a tile layer from osm.org, and a marker at the same coordinates). The Notepad++ status bar at the bottom shows: Hyper Text Markup length : 997 lines : 36, Ln : 1 Col : 1 Pos : 1, Windows (CR LF), UTF-8, IN.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4
5     <title>Quick Start - Leaflet</title>
6
7     <meta charset="utf-8" />
8     <meta name="viewport" content="width=device-width, initial-scale=1.0">
9     <link rel="shortcut icon" type="image/x-icon" href="docs/images/favicon.ico" />
10    <link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css" />
11    <script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"></script>
12
13 </head>
14 <body>
15
16     <h1>LeafLet - OSM + Marker</h1>
17
18     <!-- Define the size of the map (div) -->
19     <div id="mapid" style="width: 600px; height: 400px;"></div>
20
21     //Define the map (mymap)
22     var mymap = L.map('mapid').setView([45.95, -66.67], 13);
23
24
25     // define the layer and add it into the map (mymap)
26     L.tileLayer('http://{s}.tile.osm.org/{z}/{y}.png', {
27         attribution: '&copy; <a href="http://osm.org/copyright">OpenStreetMap</a> contributors'
28     }).addTo(mymap);
29
30     // add a marker
31     L.marker([45.95, -66.67]).addTo(mymap);
32
33 </script>
34
35 </body>
36 </html>
```

LeafLet Examples



LeafLet Examples

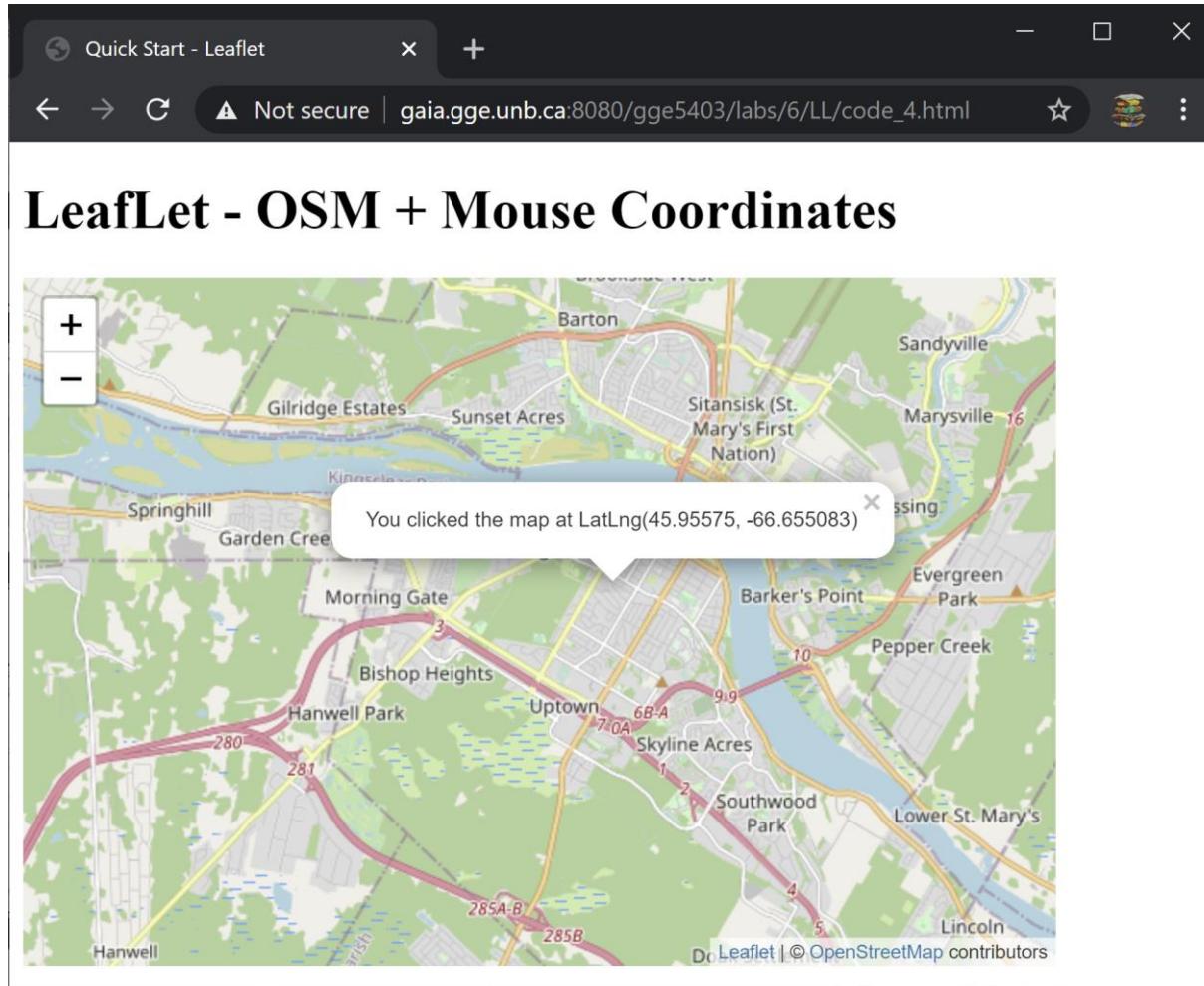


The screenshot shows a Notepad++ window displaying a JavaScript file named 'code_3.html'. The file contains code for creating a map using the LeafLet library. The code includes defining a map div, adding a tile layer from OpenStreetMap, adding a marker at coordinates [45.95, -66.67], adding a red circle at the same coordinates, adding a red polyline connecting three points, and adding a red polygon connecting six points. Each feature has a bindPopup containing descriptive text.

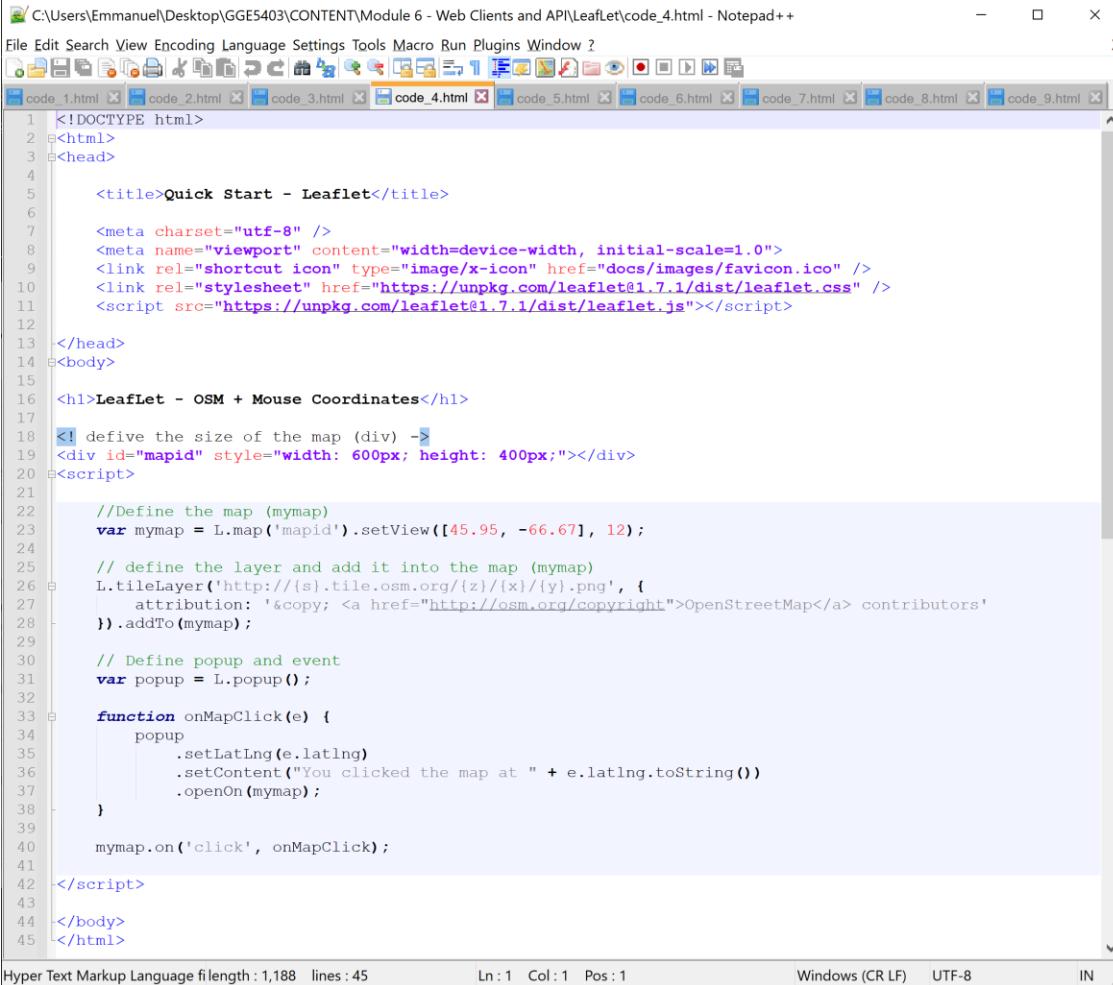
```
14 <body>
15
16 <h1>LeafLet - OSM + Marker + Geometries</h1>
17
18 // define the size of the map (div)
19 <div id="mapid" style="width: 600px; height: 400px;"></div>
20 <script>
21
22     //Define the map (mymap)
23     var mymap = L.map('mapid').setView([45.95, -66.67], 13);
24
25     // define the layer and add it into the map (mymap)
26     L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png', {
27         attribution: '&copy; <a href="http://osm.org/copyright">OpenStreetMap</a> contributors',
28     }).addTo(mymap);
29
30     // add a marker
31     L.marker([45.95, -66.67]).addTo(mymap).bindPopup("<b>I am a marker</b>").openPopup();
32
33     // add a circle
34     L.circle([45.95, -66.67], 500, {
35         color: 'red',
36         fillColor: '#f03',
37         fillOpacity: 0.5
38     }).addTo(mymap).addTo(mymap).bindPopup("I am a circle.");
39
40     // add a polyline
41     L.polyline([
42         [45.95, -66.67],
43         [45.94, -66.67],
44         [45.94, -66.66]
45     ], {color: 'red'}).addTo(mymap).bindPopup("I am a polyline");
46
47     // add a polygon
48     L.polygon([
49         [45.95, -66.67],
50         [45.96, -66.66],
51         [45.96, -66.65],
52         [45.95, -66.65],
53         [45.95, -66.67]
54     ]).addTo(mymap).bindPopup("I am a polygon");
55
56 </script>
57
58 </body>
59 </html>
```

Hyper Text Markup Language length: 1,574 lines : 59 Ln : 1 Col : 1 Pos : 1 Windows (CR LF) UTF-8 IN

LeafLet Examples



LeafLet Examples



The screenshot shows a Notepad++ window with the file "code_4.html" open. The code is a basic example of Leaflet.js usage. It includes HTML structure with a title, head, body, and a map div. It also includes a script section defining a map, adding an OSM tile layer, and setting up a click event to display a popup.

```
<!DOCTYPE html>
<html>
<head>
<title>Quick Start - Leaflet</title>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="shortcut icon" type="image/x-icon" href="docs/images/favicon.ico" />
<link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css" />
<script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"></script>
</head>
<body>
<h1>LeafLet - OSM + Mouse Coordinates</h1>
<! define the size of the map (div) ->
<div id="mapid" style="width: 600px; height: 400px;"></div>
<script>
    //Define the map (mymap)
    var mymap = L.map('mapid').setView([45.95, -66.67], 12);

    // define the layer and add it into the map (mymap)
    L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png', {
        attribution: '&copy; <a href="http://osm.org/copyright">OpenStreetMap</a> contributors'
    }).addTo(mymap);

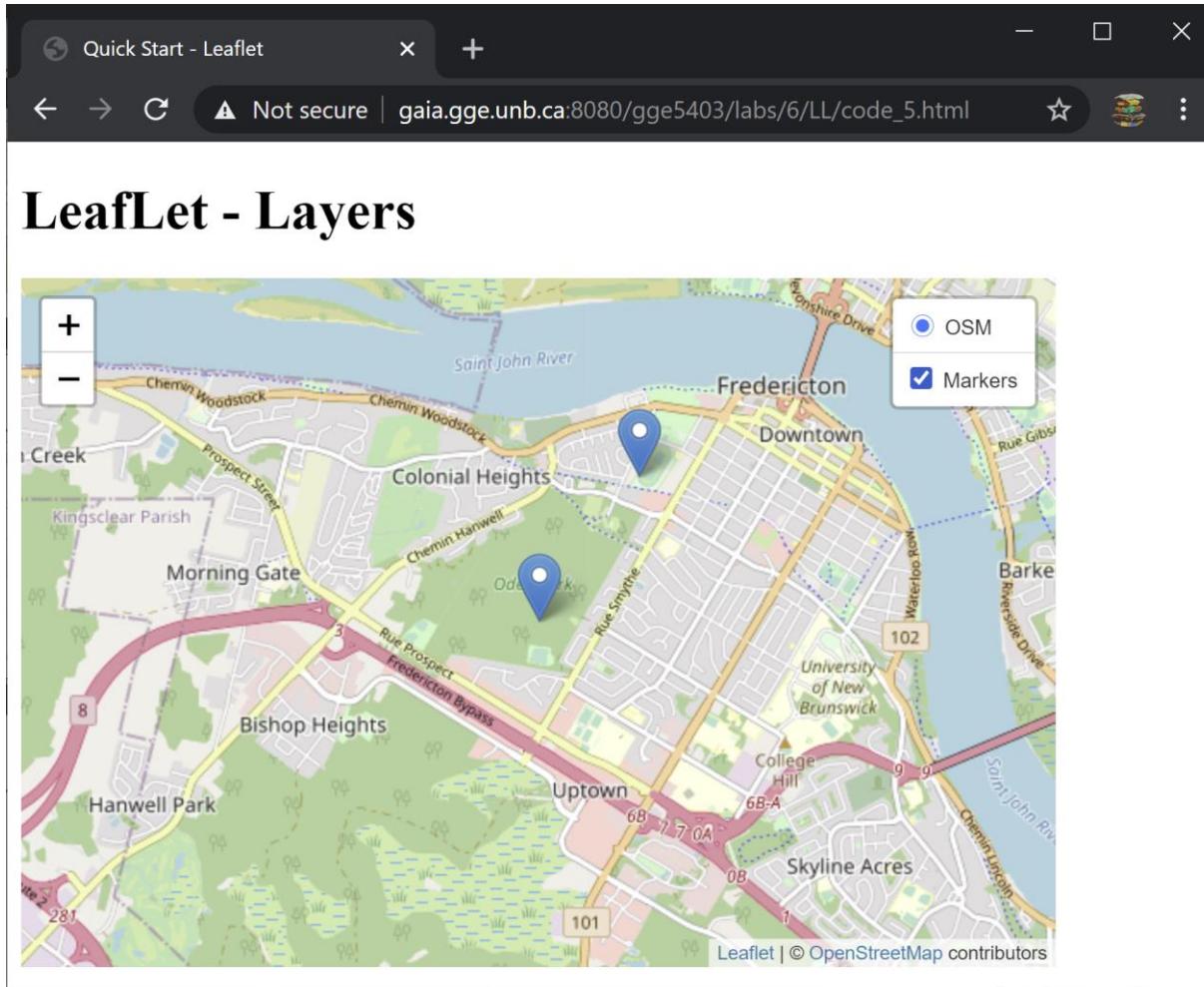
    // Define popup and event
    var popup = L.popup();

    function onMapClick(e) {
        popup
            .setLatLng(e.latlng)
            .setContent("You clicked the map at " + e.latlng.toString())
            .openOn(mymap);
    }

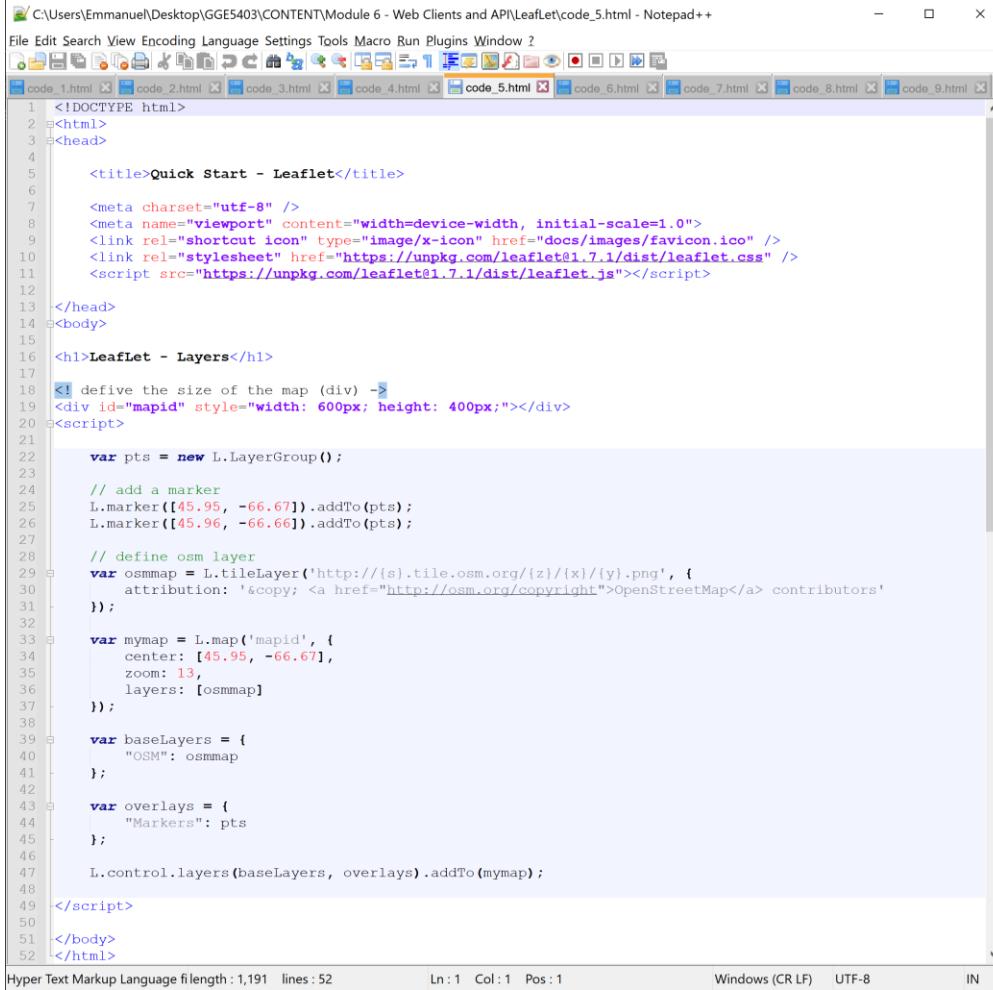
    mymap.on('click', onMapClick);
</script>
</body>
</html>
```

Hyper Text Markup Language filength : 1,188 lines : 45 Ln : 1 Col : 1 Pos : 1 Windows (CR LF) UTF-8 IN

LeafLet Examples



LeafLet Examples

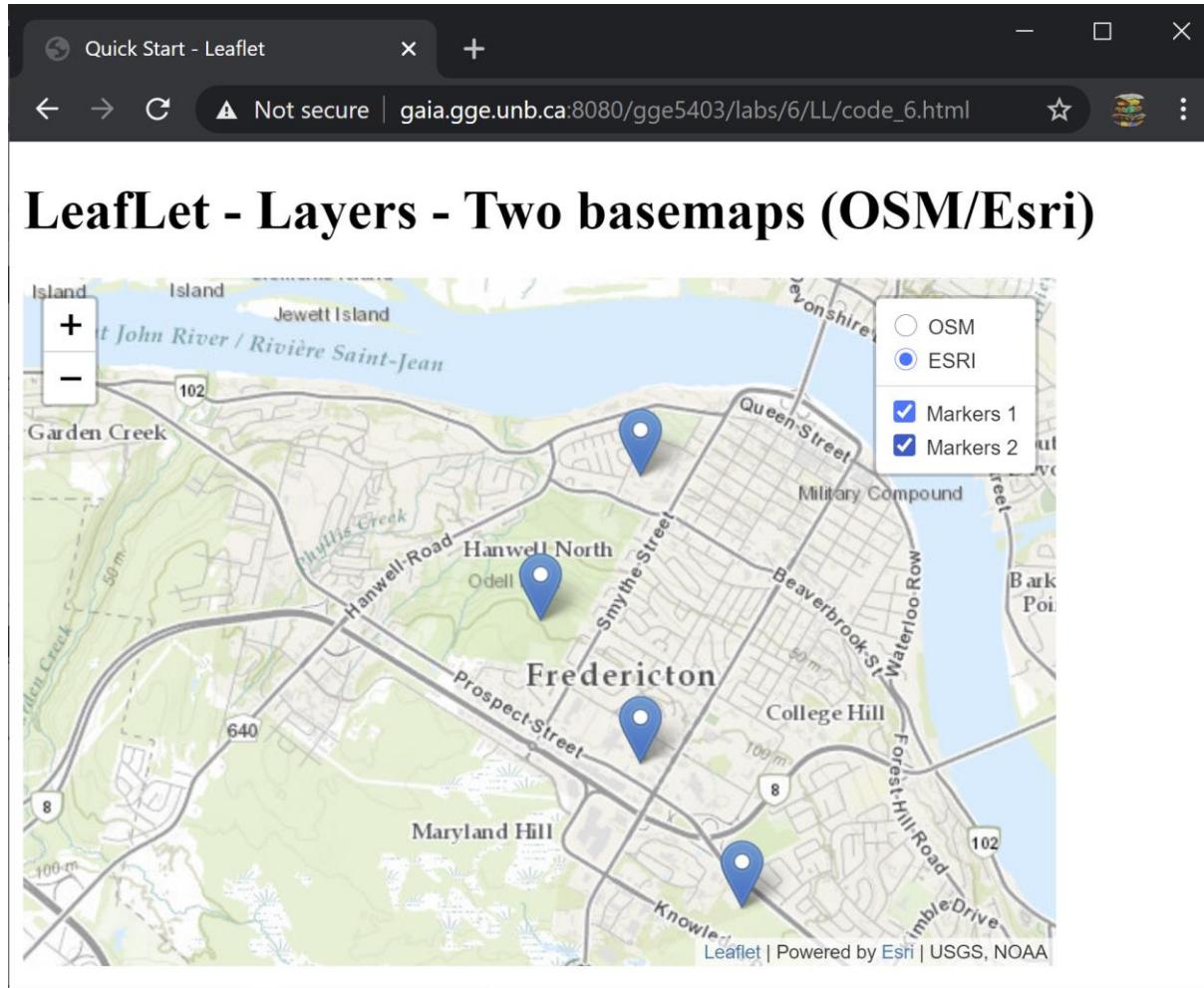


A screenshot of the Notepad++ text editor window. The title bar reads "C:\Users\Emmanuel\Desktop\GGE5403\CONTENT\Module 6 - Web Clients and API\Leaflet\code_5.html - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, and Help. The toolbar has various icons for file operations. Below the toolbar is a tab bar with multiple tabs labeled "code_1.html" through "code_9.html", with "code_5.html" currently selected. The main code editor area contains the following JavaScript code:

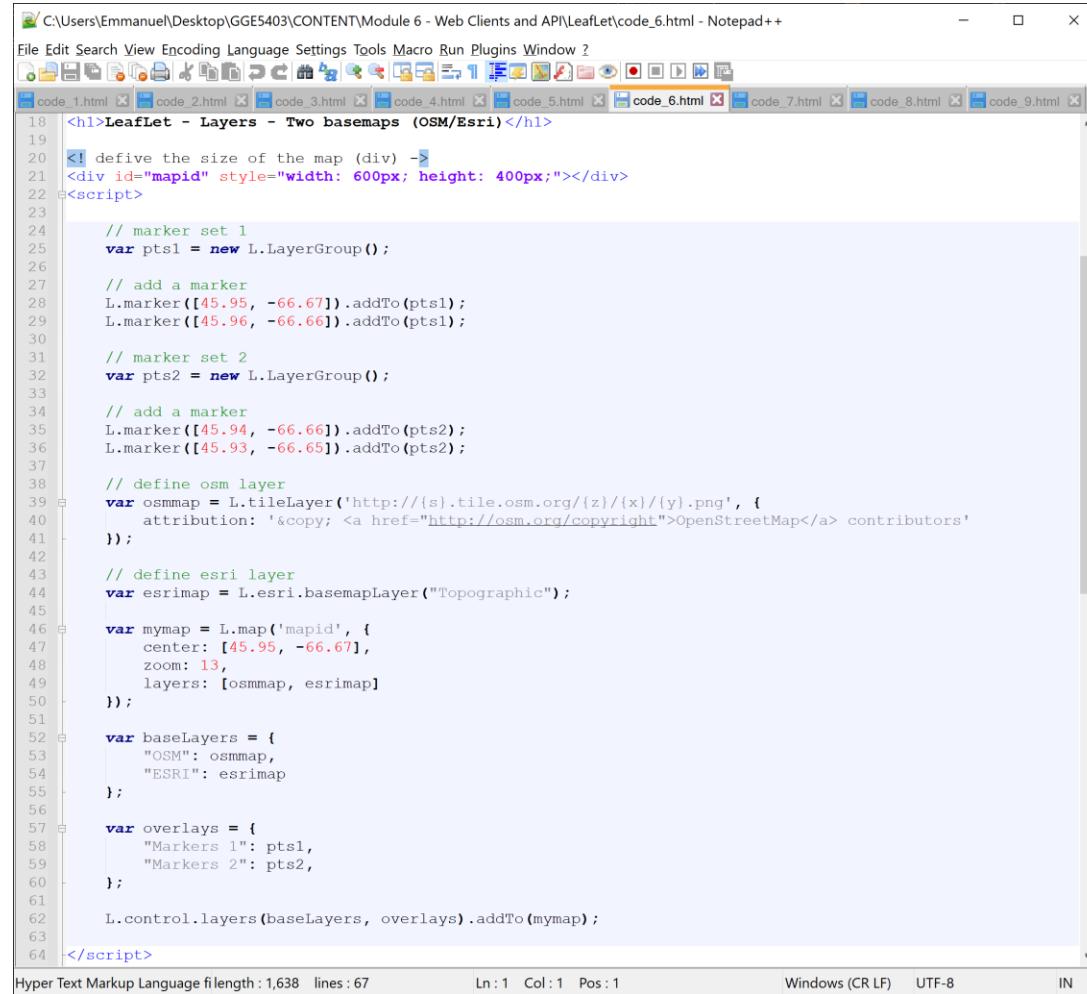
```
<!DOCTYPE html>
<html>
<head>
<title>Quick Start - Leaflet</title>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<link rel="shortcut icon" type="image/x-icon" href="docs/images/favicon.ico" />
<link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css" />
<script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"></script>
</head>
<body>
<h1>LeafLet - Layers</h1>
<!-- define the size of the map (div) -->
<div id="mapid" style="width: 600px; height: 400px;"></div>
<script>
var pts = new L.LayerGroup();
// add a marker
L.marker([45.95, -66.67]).addTo(pts);
L.marker([45.96, -66.66]).addTo(pts);
// define osm layer
var osmmap = L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png', {
    attribution: '&copy; <a href="http://osm.org/copyright">OpenStreetMap</a> contributors'
});
var mymap = L.map('mapid', {
    center: [45.95, -66.67],
    zoom: 13,
    layers: [osmmap]
});
var baseLayers = {
    "OSM": osmmap
};
var overlays = {
    "Markers": pts
};
L.control.layers(baseLayers, overlays).addTo(mymap);
</script>
</body>
</html>
```

At the bottom of the editor, status bars show "Hyper Text Markup Language filenleng : 1,191 lines : 52", "Ln : 1 Col : 1 Pos : 1", "Windows (CR LF)", "UTF-8", and "IN".

LeafLet Examples



LeafLet Examples

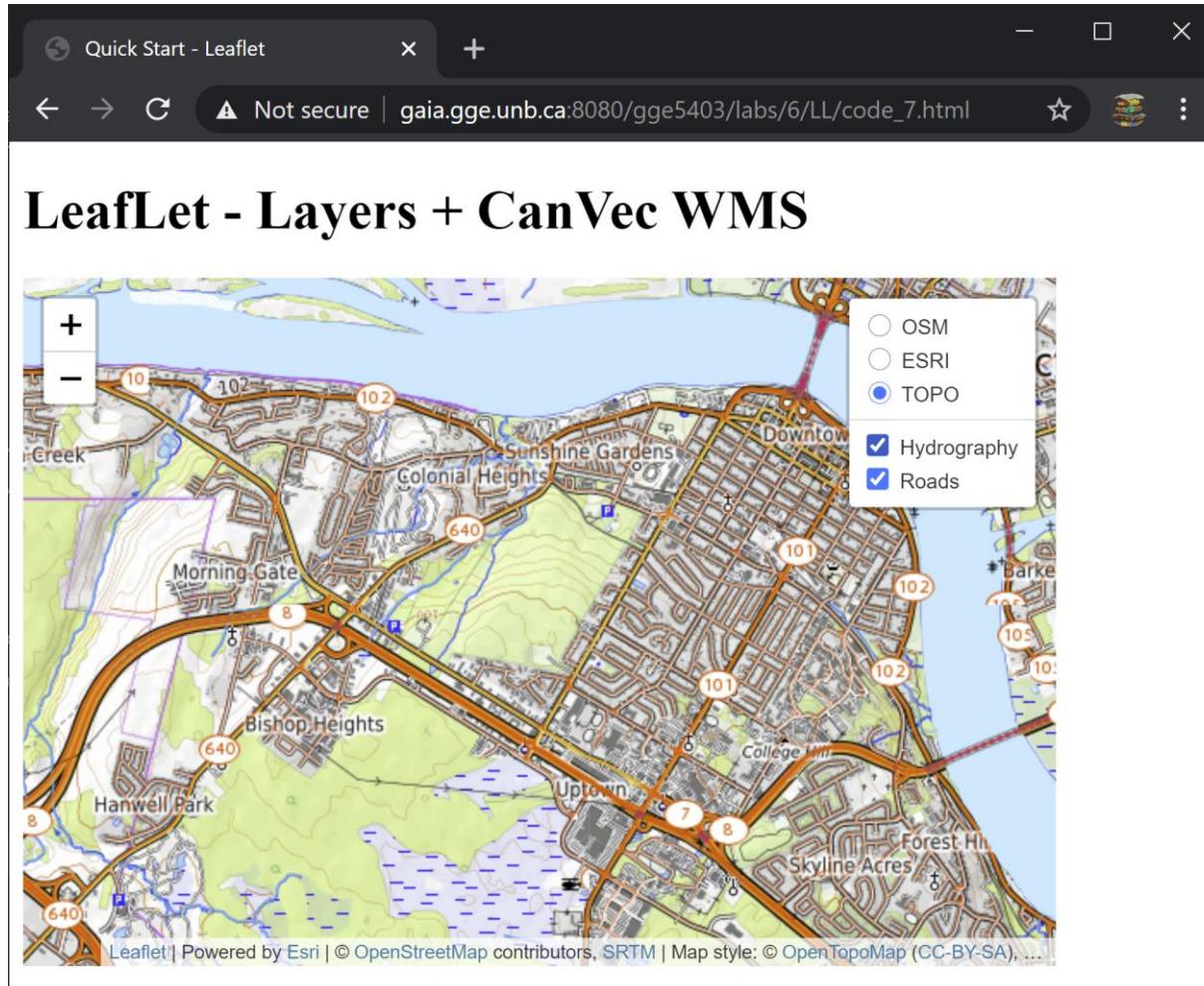


The screenshot shows a Notepad++ window displaying a JavaScript file named 'code_6.html'. The code implements Leaflet to display two different base maps (OSM and Esri) and two sets of markers. The OSM layer is defined with attribution to OpenStreetMap contributors. The Esri layer is a topographic map. The code uses LayerGroups to manage marker sets and overlays to switch between base maps.

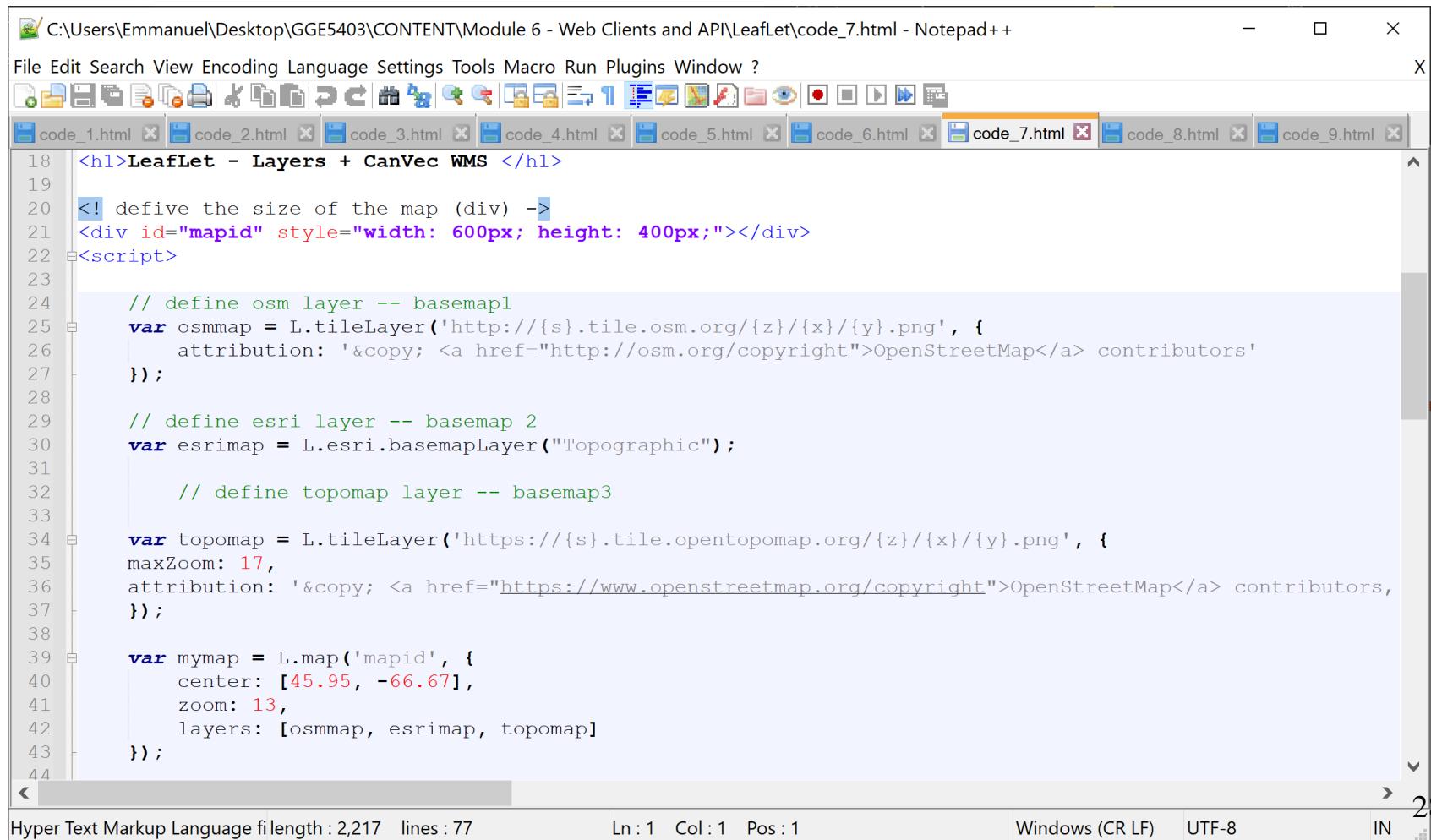
```
18 <h1>LeafLet - Layers - Two basemaps (OSM/Esri)</h1>
19
20 // define the size of the map (div)
21 <div id="mapid" style="width: 600px; height: 400px;"></div>
22 <script>
23
24 // marker set 1
25 var pts1 = new L.LayerGroup();
26
27 // add a marker
28 L.marker([45.95, -66.67]).addTo(pts1);
29 L.marker([45.96, -66.66]).addTo(pts1);
30
31 // marker set 2
32 var pts2 = new L.LayerGroup();
33
34 // add a marker
35 L.marker([45.94, -66.66]).addTo(pts2);
36 L.marker([45.93, -66.65]).addTo(pts2);
37
38 // define osm layer
39 var osmmap = L.tileLayer('http://(s).tile.osm.org/{z}/{x}/{y}.png', {
40   attribution: '&copy; <a href="http://osm.org/copyright">OpenStreetMap</a> contributors'
41 });
42
43 // define esri layer
44 var esrimap = L.esri.basemapLayer("Topographic");
45
46 var mymap = L.map('mapid', {
47   center: [45.95, -66.67],
48   zoom: 13,
49   layers: [osmmap, esrimap]
50 });
51
52 var baseLayers = {
53   "OSM": osmmap,
54   "ESRI": esrimap
55 };
56
57 var overlays = {
58   "Markers 1": pts1,
59   "Markers 2": pts2,
60 };
61
62 L.control.layers(baseLayers, overlays).addTo(mymap);
63
64 </script>
```

Hyper Text Markup Language filength : 1,638 lines : 67 Ln : 1 Col : 1 Pos : 1 Windows (CR LF) UTF-8 IN

LeafLet Examples



LeafLet Examples

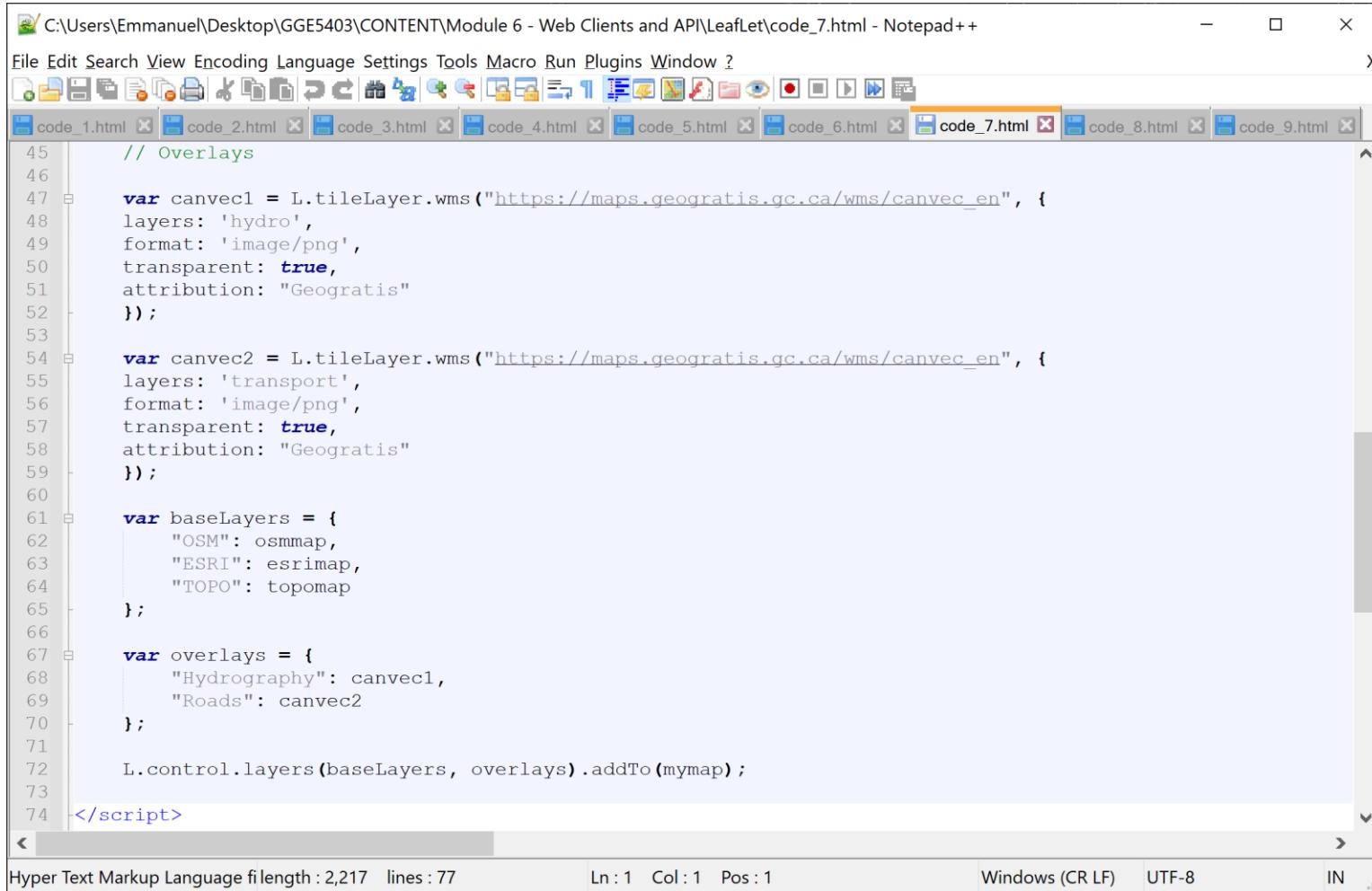


The screenshot shows a Notepad++ window with the title bar "C:\Users\Emmanuel\Desktop\GGE5403\CONTENT\Module 6 - Web Clients and API\LeafLet\code_7.html - Notepad++". The menu bar includes File, Edit, Search, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, and Help. The toolbar has various icons for file operations like Open, Save, Print, and Find. Below the toolbar is a tab bar with nine tabs labeled "code_1.html" through "code_9.html", with "code_7.html" currently selected. The main editor area contains the following JavaScript code:

```
18 <h1>LeafLet - Layers + CanVec WMS </h1>
19
20 <! define the size of the map (div) ->
21 <div id="mapid" style="width: 600px; height: 400px;"></div>
22 <script>
23
24 // define osm layer -- basemap1
25 var osmmap = L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png', {
26   attribution: '&copy; <a href="http://osm.org/copyright">OpenStreetMap</a> contributors',
27 });
28
29 // define esri layer -- basemap 2
30 var esrimap = L.esri.basemapLayer("Topographic");
31
32 // define topomap layer -- basemap3
33
34 var topomap = L.tileLayer('https://{s}.tile.opentopomap.org/{z}/{x}/{y}.png', {
35   maxZoom: 17,
36   attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors',
37 });
38
39 var mymap = L.map('mapid', {
40   center: [45.95, -66.67],
41   zoom: 13,
42   layers: [osmmap, esrimap, topomap]
43 });
44
```

The status bar at the bottom shows "Hyper Text Markup Language filength : 2,217 lines : 77" and "Windows (CR LF) UTF-8 IN". The page number "28" is located in the bottom right corner.

LeafLet Examples



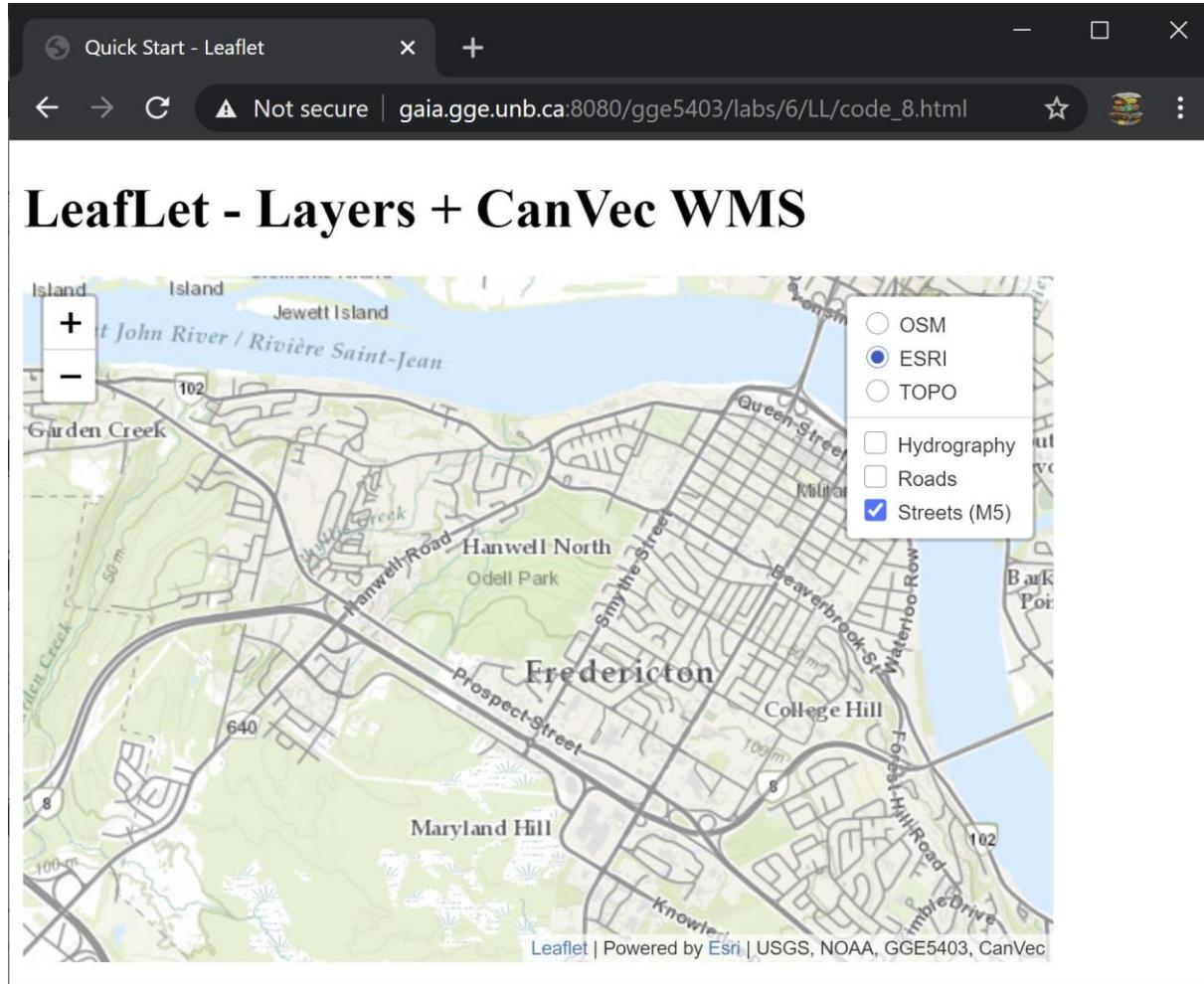
The screenshot shows a Notepad++ window displaying a JavaScript file named 'code_7.html'. The file contains code for creating a Leaflet map with overlays. The code defines two WMS layers ('canvec1' and 'canvec2') for hydrography and roads, and three base layers ('OSM', 'ESRI', and 'TOPO'). It then adds these layers to a map object ('mymap'). The code is color-coded, with variables in blue and comments in green.

```
45 // Overlays
46
47 var canvec1 = L.tileLayer.wms("https://maps.geogratis.gc.ca/wms/canvec_en", {
48   layers: 'hydro',
49   format: 'image/png',
50   transparent: true,
51   attribution: "Geogratis"
52 });
53
54 var canvec2 = L.tileLayer.wms("https://maps.geogratis.gc.ca/wms/canvec_en", {
55   layers: 'transport',
56   format: 'image/png',
57   transparent: true,
58   attribution: "Geogratis"
59 });
60
61 var baseLayers = {
62   "OSM": osmmap,
63   "ESRI": esrimap,
64   "TOPO": topomap
65 };
66
67 var overlays = {
68   "Hydrography": canvec1,
69   "Roads": canvec2
70 };
71
72 L.control.layers(baseLayers, overlays).addTo(mymap);
73
74 </script>
```

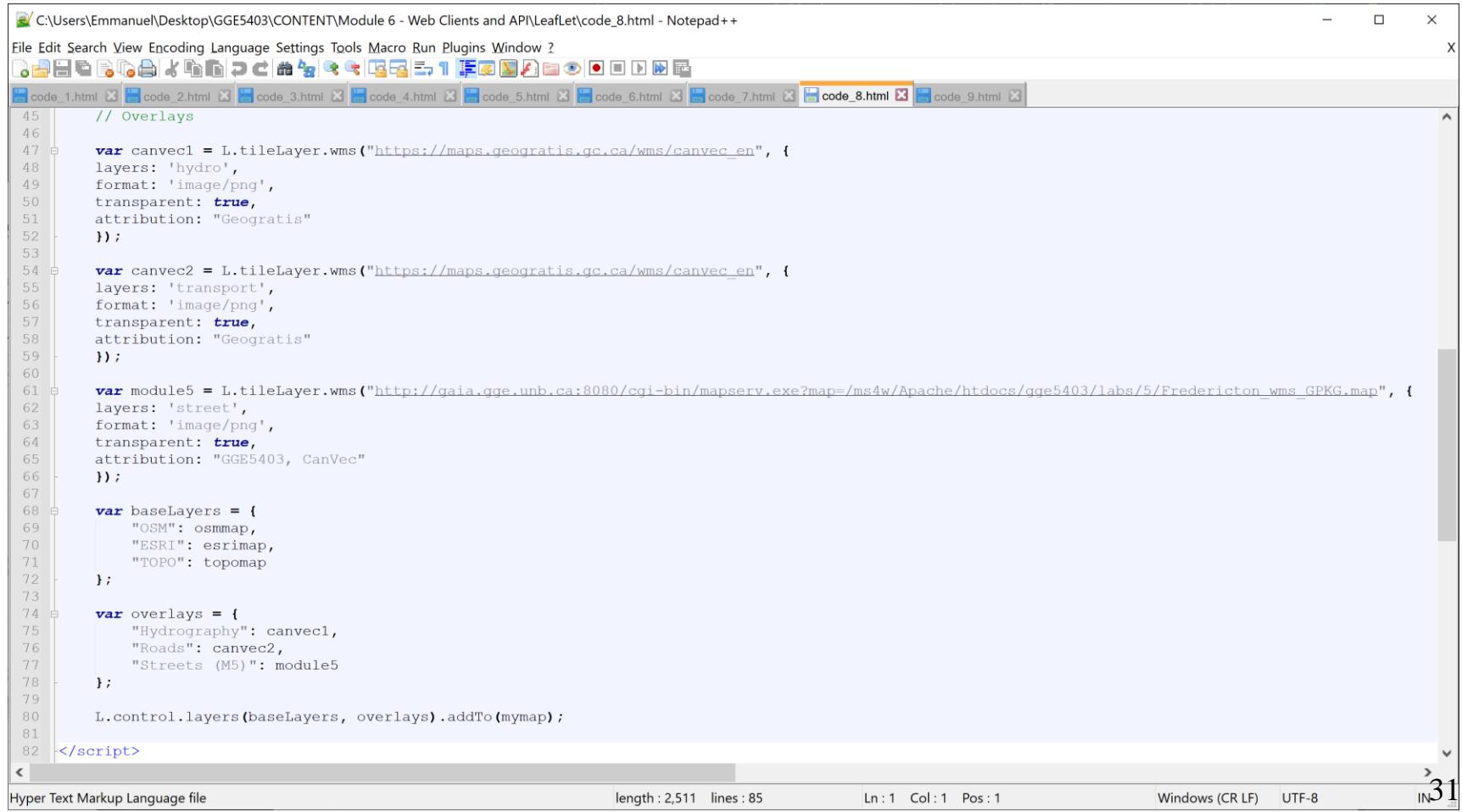
Hyper Text Markup Language filength : 2,217 lines : 77 Ln : 1 Col : 1 Pos : 1 Windows (CR LF) UTF-8 IN

LeafLet Examples

(add streets from Module 5)



LeafLet Examples

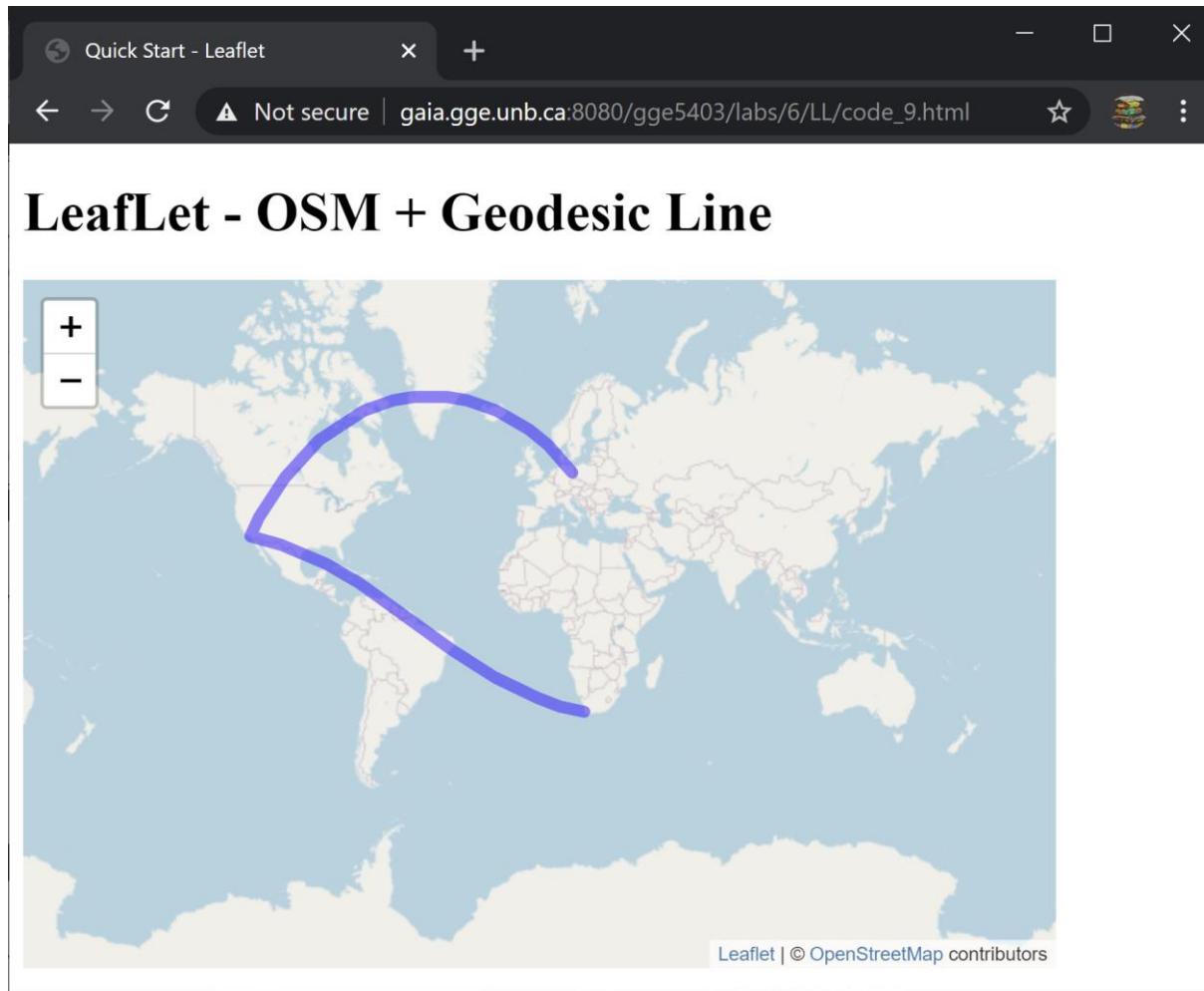


The screenshot shows a Notepad++ window displaying a JavaScript file named 'code_8.html'. The code defines several layers for a Leaflet map, including WMS layers for hydrography, roads, and streets, and base layers from OSM, ESRI, and TOPO. The code uses the L.tileLayer.wms and L.control.layers methods to create the map.

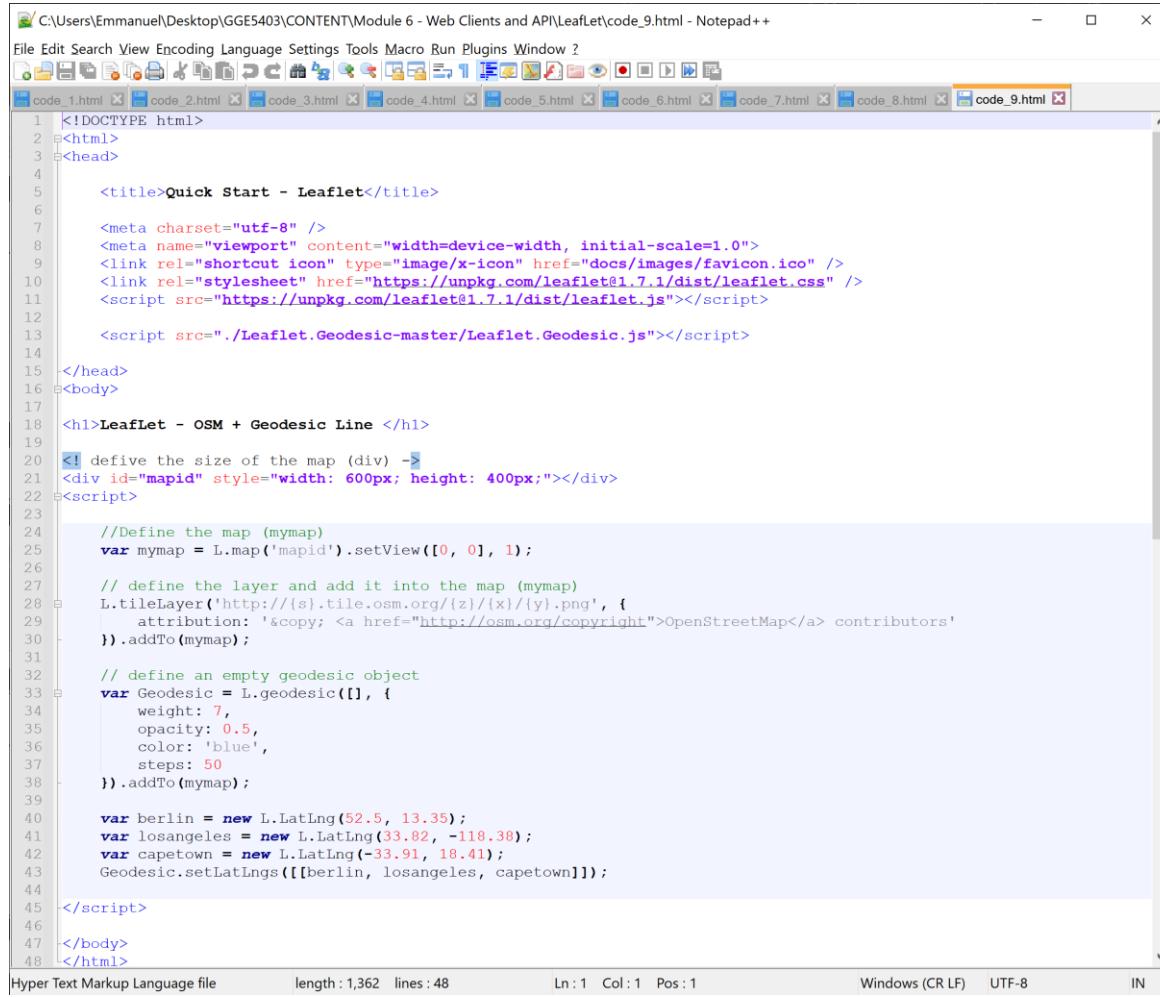
```
45 // Overlays
46
47 var canvec1 = L.tileLayer.wms("https://maps.geogratis.gc.ca/wms/canvec_en", {
48   layers: 'hydro',
49   format: 'image/png',
50   transparent: true,
51   attribution: "Geogratis"
52 });
53
54 var canvec2 = L.tileLayer.wms("https://maps.geogratis.gc.ca/wms/canvec_en", {
55   layers: 'transport',
56   format: 'image/png',
57   transparent: true,
58   attribution: "Geogratis"
59 });
60
61 var module5 = L.tileLayer.wms("http://gaia.gge.unb.ca:8080/cgi-bin/mapserv.exe?map=/ms4w/Apache/htdocs/gge5403/labs/5/Fredericton_wms_GPKG.map", {
62   layers: 'street',
63   format: 'image/png',
64   transparent: true,
65   attribution: "GGE5403, CanVec"
66 });
67
68 var baseLayers = {
69   "OSM": osmmap,
70   "ESRI": esrimap,
71   "TOPO": topomap
72 };
73
74 var overlays = {
75   "Hydrography": canvec1,
76   "Roads": canvec2,
77   "Streets (M5)": module5
78 };
79
80 L.control.layers(baseLayers, overlays).addTo(mymap);
81
82 </script>
```

Hyper Text Markup Language file length : 2,511 lines : 85 Ln : 1 Col : 1 Pos : 1 Windows (CR LF) UTF-8 IN 31

LeafLet Examples



LeafLet Examples



The screenshot shows a Notepad++ window with the title bar "C:\Users\Emmanuel\Desktop\GGE5403\CONTENT\Module 6 - Web Clients and API\LeafLet\code_9.html - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, and Help. The toolbar has various icons for file operations. The status bar at the bottom shows "Hyper Text Markup Language file", "length : 1,362 lines : 48", "Ln : 1 Col : 1 Pos : 1", "Windows (CR LF)", "UTF-8", and "IN". The code in the editor is as follows:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4
5     <title>Quick Start - Leaflet</title>
6
7     <meta charset="utf-8" />
8     <meta name="viewport" content="width=device-width, initial-scale=1.0">
9     <link rel="shortcut icon" type="image/x-icon" href="docs/images/favicon.ico" />
10    <link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css" />
11    <script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"></script>
12
13    <script src="./Leaflet.Geodesic-master/Leaflet.Geodesic.js"></script>
14
15 </head>
16 <body>
17
18 <h1>LeafLet - OSM + Geodesic Line </h1>
19
20 <!-- define the size of the map (div) -->
21 <div id="mapid" style="width: 600px; height: 400px;"></div>
22 <script>
23
24     //Define the map (mymap)
25     var mymap = L.map('mapid').setView([0, 0], 1);
26
27     // define the layer and add it into the map (mymap)
28     L.tileLayer('http://[s].tile.osm.org/{z}/{x}/{y}.png', {
29         attribution: '&copy; <a href="http://osm.org/copyright">OpenStreetMap</a> contributors'
30     }).addTo(mymap);
31
32     // define an empty geodesic object
33     var Geodesic = L.geodesic([], {
34         weight: 7,
35         opacity: 0.5,
36         color: 'blue',
37         steps: 50
38     }).addTo(mymap);
39
40     var berlin = new L.LatLng(52.5, 13.35);
41     var losangeles = new L.LatLng(33.82, -118.38);
42     var capetown = new L.LatLng(-33.91, 18.41);
43     Geodesic.setLatLngs([[berlin, losangeles, capetown]]);
44
45 </script>
46
47 </body>
48 </html>
```

LeafLet Examples

- More examples...
 - <https://leafletjs.com/examples.html>



OpenLayers Example

<http://gaia.gge.unb.ca:8080/gge5403/labs/6/OL/FrederictonOL.html>

The screenshot shows a web browser window with the title "Fredericton OpenLayer and WMS". The address bar indicates the URL is <http://gaia.gge.unb.ca:8080/gge5403/labs/6/OL/FrederictonOL.html>. The main content area displays a map of Fredericton, New Brunswick, featuring a green base layer representing land cover. Overlaid on the map are several features: red dots representing bus stops, black lines representing roads, and green areas representing parks. A legend box on the right side, titled "Base Layer", contains a radio button for "Open Street Map" which is unselected. Below the legend are three checked checkboxes: "Bus Stops", "Roads", and "Parks". The map also includes various place names such as Kingsclear Parish, Morning Gate, Bishop Heights, Hanwell Park, Uptown, Southwood, Lower St. Mary's, and Lincoln. Road numbers like 280, 281, 282, 283, 284, 285, 286, 105, and 8 are visible. A small legend in the bottom right corner identifies the colors for roads (black), parks (green), and bus stops (red). The browser interface includes standard controls for zooming and navigating.

Fredericton OpenLayer and WMS

This thin Web Client Application was developed using OpenLayers JavaScript and WMS

Base Layer
Open Street Map

Overlays
 Bus Stops
 Roads
 Parks



Thin Web Mapping Clients

Emmanuel Stefanakis

estef@unb.ca