

# Powerful Mapping Web Applications with Open Source Tools

## Part Two: OpenStreetMap

Mele Sax-Barnett and  
Grant Humphries

---

# An Introduction to OpenStreetMap

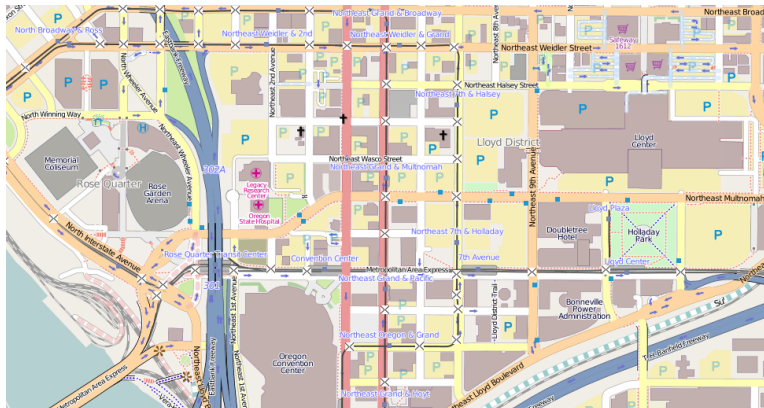
# OpenStreetWhat?

- OpenStreetMap or OSM
- **Not** “Open Street Maps”
- Founded in 2004
- Worldwide and seamless
- “Wikipedia of Maps”
  - Editable by anyone with an account
  - Data available to anyone, for free
  - ... as long as you credit the contributors
- <http://www.openstreetmap.org>



# What does it look like?

OSM.org default



Stamen Design



MapBox



MapQuest

# Who uses OpenStreetMap?

- <http://switch2osm.org>

**flickr**®  
from YAHOO!

**foursquare**

**craigslist**

**TRI MET**

 **MapBox**

(you)

**m<sup>a</sup> mapquest**

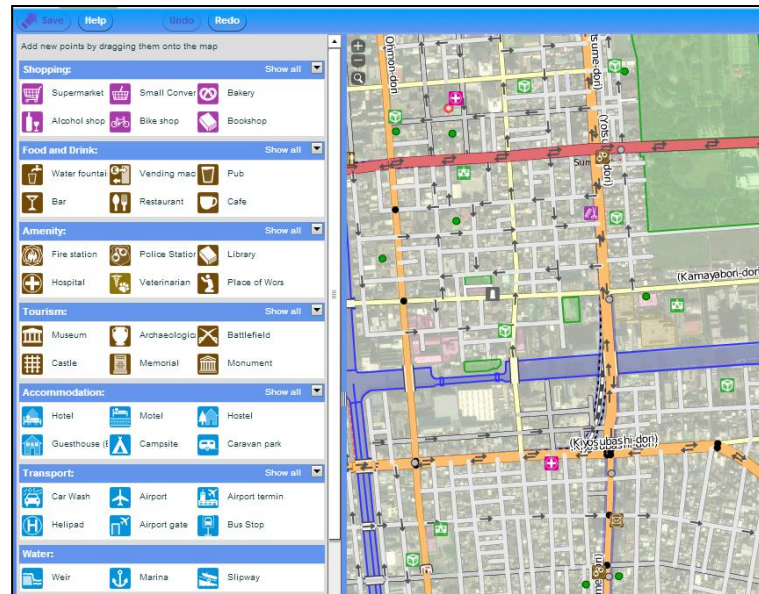
 **nestoria**



**WIKIPEDIA**  
The Free Encyclopedia

## How do you edit it?

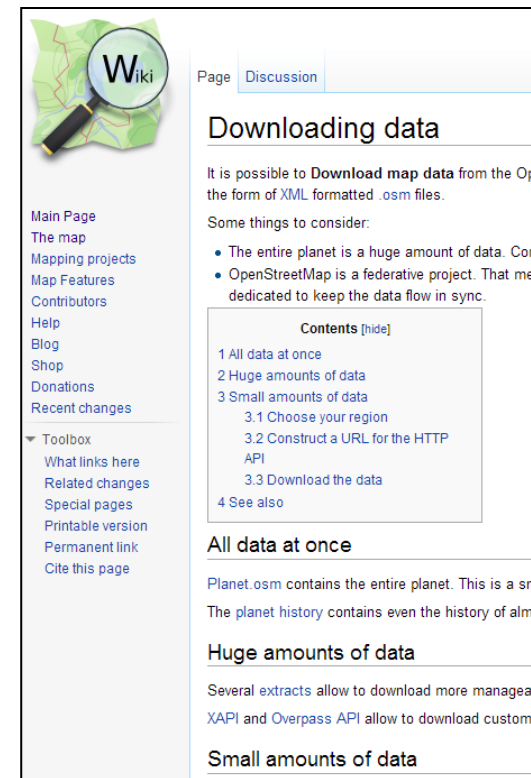
- Create an account
- Various free and open source editors available
- Potlatch and JOSM are very popular
- We will be using Potlatch 2, a flash-based, in-browser editor





# How do you get the data?

- [http://wiki.openstreetmap.org/wiki/Downloading\\_data](http://wiki.openstreetmap.org/wiki/Downloading_data)
- Download via JOSM
  - Easily download and save small areas
  - “Mirrored Download” plugin for large areas
- Large extracts hosted around the web, updated at differing schedules
- We will be using the OSM plugin for QGIS



# **Editing basics: Points of interest**



# Step 1: Sign up for an account

- Go to <http://osm.org>
- Click the “sign up” link in the top right corner
- Set up your account!



## Create a User Account

Fill in the form and we will send you a quick email to activate your account.

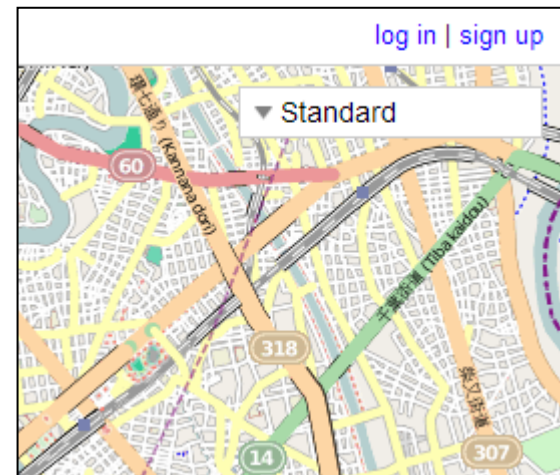
Email Address:

Confirm Email Address:

Not displayed publicly (see [privacy policy](#))

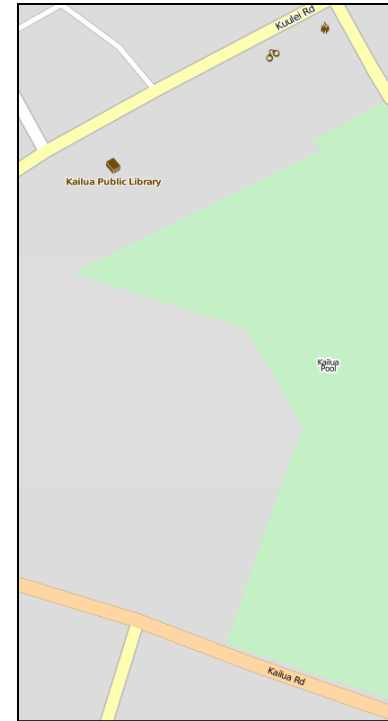
Display Name:

Your publicly displayed username. You can change this later in the preferences.



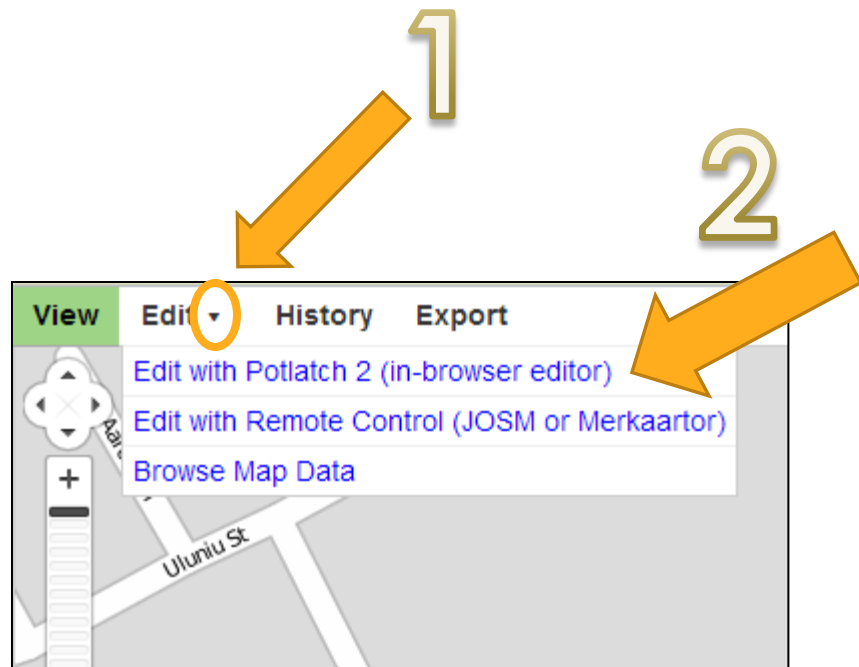
## Step 2: Where and what

- Congratulations! You are now the owner of one of ~850,000 OSM editing accounts!
- Go back to <http://osm.org>
- Find a neighborhood that you know well
- Think about what you know about in that place
- **Local knowledge**
- Notice anything missing?



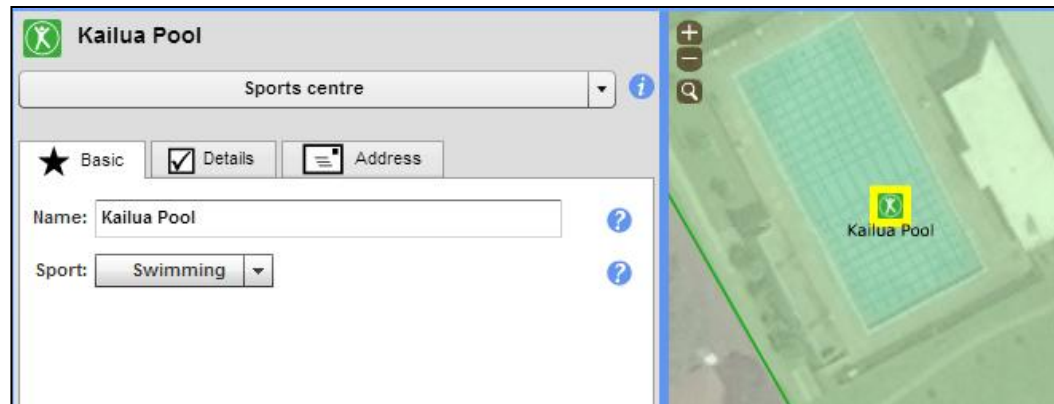
## Step 3: Open the editor

- Click on the down-arrow next to the “Edit” menu
- Select “Edit with Potlatch 2”



## Step 4: Explore

- Start familiarizing yourself with Potlatch
- Don't worry, you can't break anything ***until you click "Save"***
- Click things to see how they're classified



## Step 5: Add something

- But first, what's your source?
- **Don't copy from other maps**
- Local knowledge is best, but a dataset with the correct license and permissions can be a source
- You even need permission for aerial imagery (imagery included in OSM editors is OK)
- More info about the OpenStreetMap license (OdBL) can be found at <http://www.openstreetmap.org/copyright>

# Step 5: Add something

- Pick something in the panel
- Click the arrow to expand
- What do you know about in this area? What are you interested in mapping?
- I picked a café
- Drag it onto the map where the POI is



# Step 5: Add something

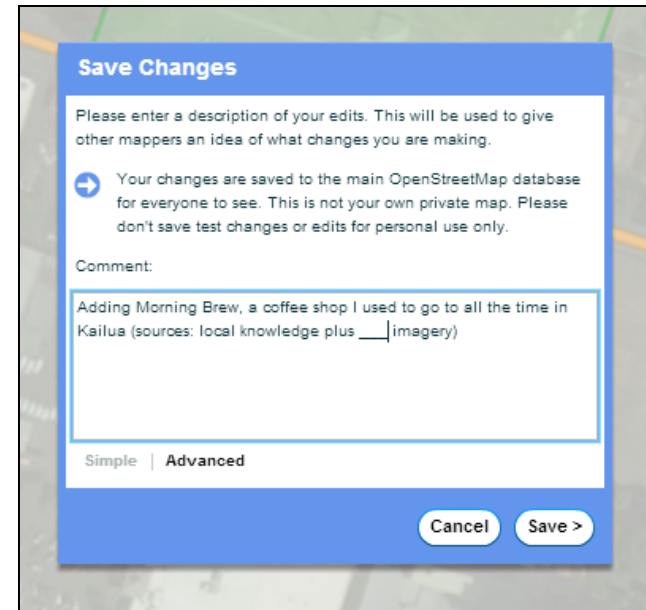
- Next, fill in the details that you know

The screenshot shows the Google Maps 'Morning Brew' location editing interface. At the top, there are buttons for 'Save', 'Help', 'Undo', and 'Redo'. Below these, the location name 'Morning Brew' is displayed next to a coffee cup icon. A dropdown menu shows 'Cafe' with an information icon. Below this, there are three tabs: 'Basic' (selected), 'Details' (checked), and 'Address'. The 'Details' tab contains several fields: 'Name' (Morning Brew), 'Cuisine' (Unset), 'Wifi' (Free), 'Official classification' (Unset), and 'Source' (Unset). Each field has a question mark icon to its right. At the bottom, there is a checkbox for 'Additional names'. To the right of the form is a map view showing the location of 'Morning Brew' on a street, with a yellow pin and a coffee cup icon.



# Step 5: Save your edits

- Save early, save often
- This creates a “changeset” that is sent to the database
- **Give an informative changeset comment** *that includes what you were working on and what your sources were*
- What imagery are you using? Do you know how to change it?



The screenshot shows the 'Save Changes' dialog box in OpenStreetMap. It has a blue header with the title 'Save Changes'. Below the header, there is a text area for a comment. The text area contains the following text: 'Adding Morning Brew, a coffee shop I used to go to all the time in Kailua (sources: local knowledge plus \_\_\_|imagery)'. Below the text area, there are two tabs: 'Simple' and 'Advanced'. At the bottom right, there are two buttons: 'Cancel' and 'Save >'. The background of the dialog box is a blurred map.

**Save Changes**

Please enter a description of your edits. This will be used to give other mappers an idea of what changes you are making.

→ Your changes are saved to the main OpenStreetMap database for everyone to see. This is not your own private map. Please don't save test changes or edits for personal use only.

Comment:

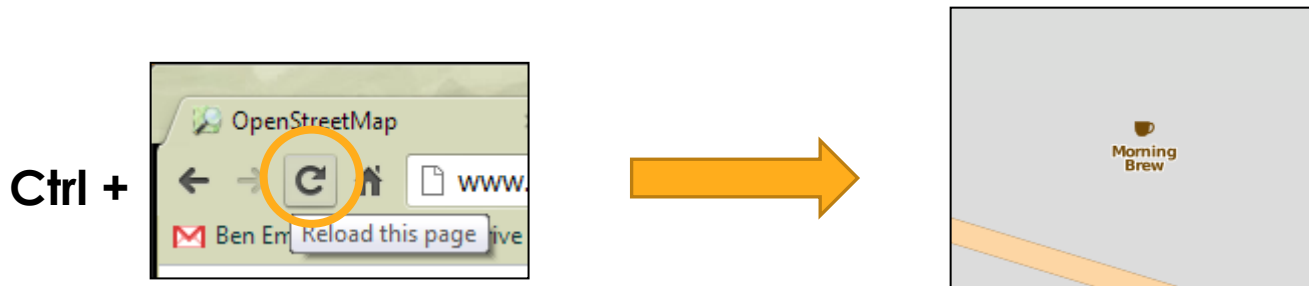
Adding Morning Brew, a coffee shop I used to go to all the time in Kailua (sources: local knowledge plus \_\_\_|imagery)

Simple | **Advanced**

Cancel Save >

## Step 6: View your edits

- Click the “View” button in the top left to go back to the map
- *Hold down Ctrl while clicking refresh to clear your browser’s cache of map tiles (shift-refresh with Firefox)*



## Example workflow: Fixing something

- Find a POI (point of interest) you know about that's already in the map
- Click the “Advanced” button at the bottom to see all the tags on the POI

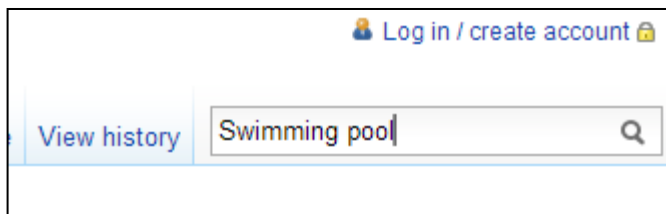


Node: 1165895778

Key	Value	
leisure	sports_centre	✕
name	Kailua Pool	✕
sport	swimming	✕

## Example workflow: Fixing something

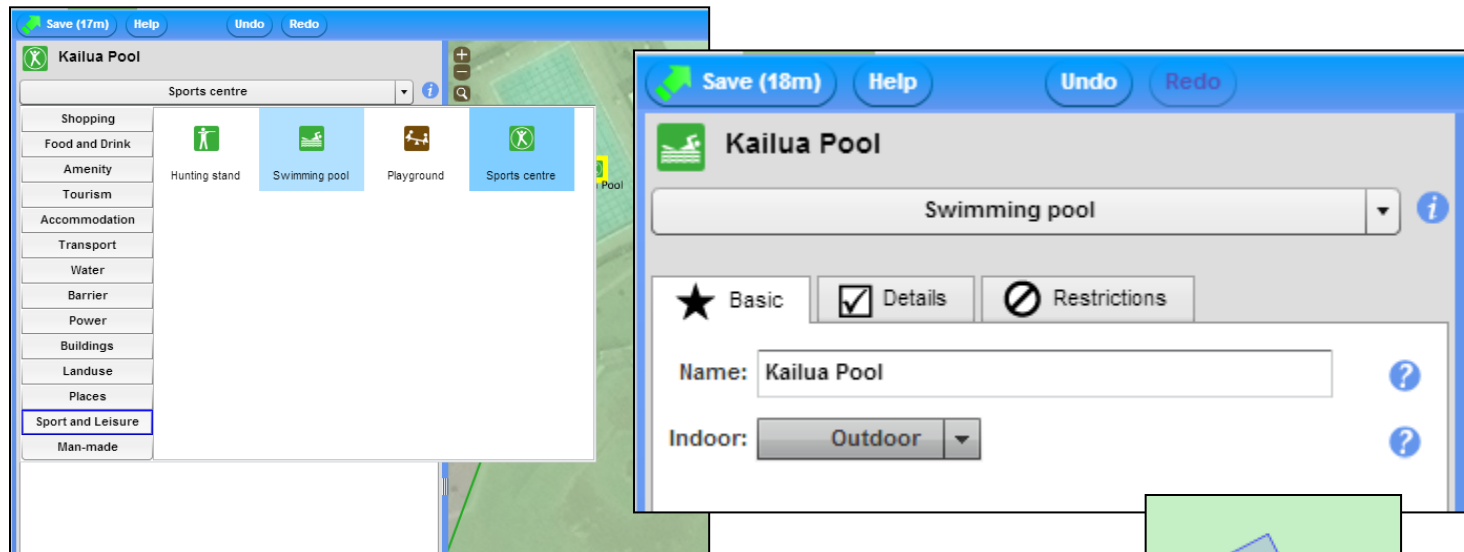
- Look that type of thing up on <http://wiki.openstreetmap.org>
- Is it tagged correctly?
- This one should be fixed



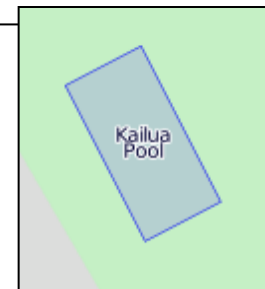
leisure=swimming\_pool,  
*not* leisure=sports\_centre



# Example workflow: Fixing something



... and after changing it to a polygon...



## More about editing

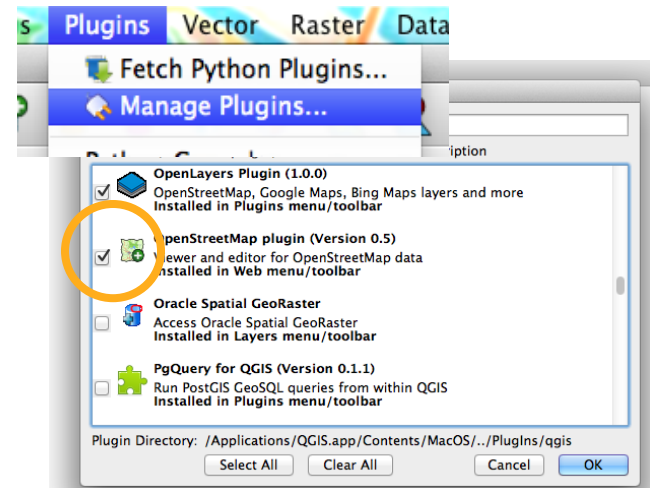
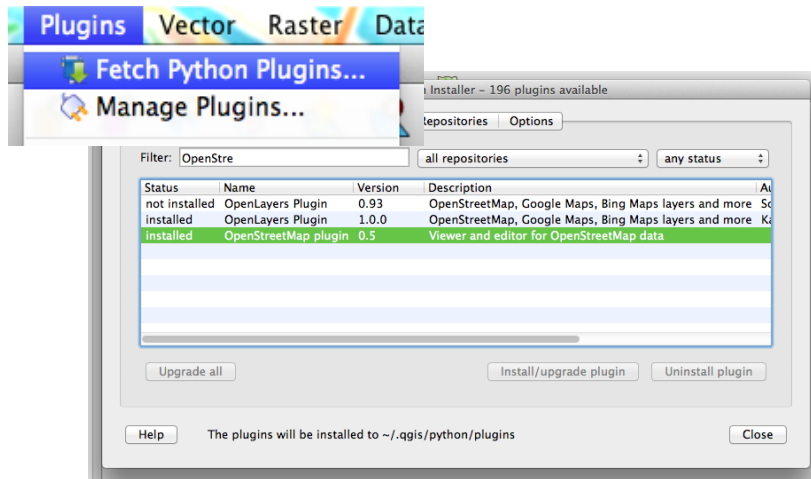
- For anything more than point of interest editing, we recommend using JOSM (Java OpenStreetMap Editor)
- Not too hard to learn, especially if you're familiar with GIS software
- Handy tools that automatically line up nodes, orthogonalize, and make circles
- Remember to visit the wiki for tagging help, or ask on the newbies listserv  
<http://lists.openstreetmap.org/listinfo/newbies>

# Getting and Preparing your OSM Data



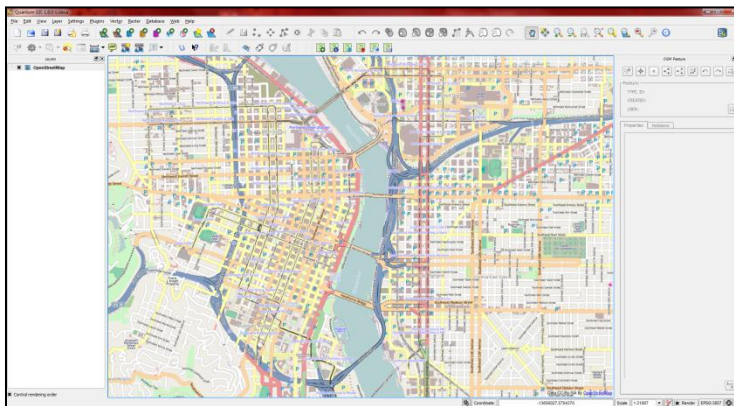
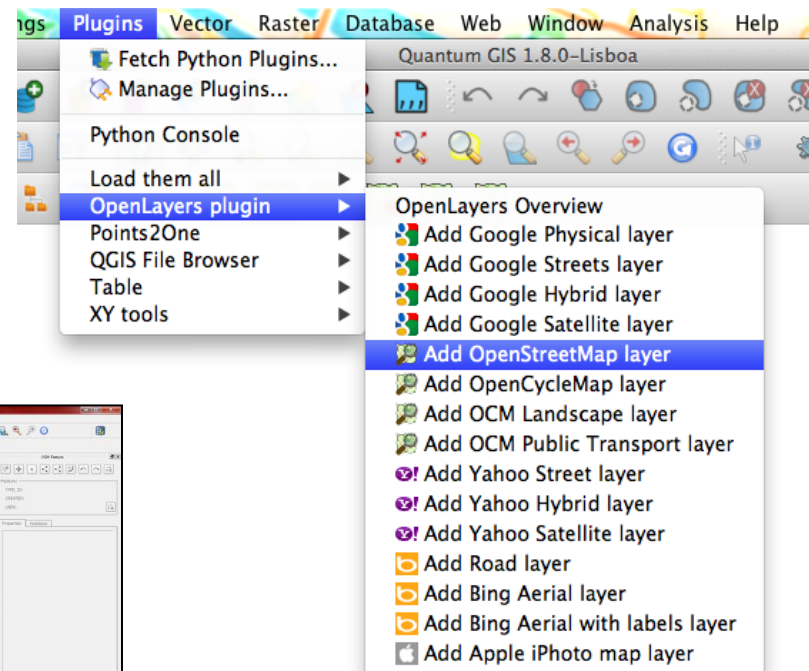
## Step 1: Get your plugin working

- We will be using QuantumGIS today to limit the software you needed to install
- Make sure you've not only downloaded the OSM plugin, but turned it on (2 steps)



## Step 2: Select an area of interest

- Go to Plugins-> OpenLayers plugin -> Add OpenStreetMap Layer
- Zoom and pan to an area of interest



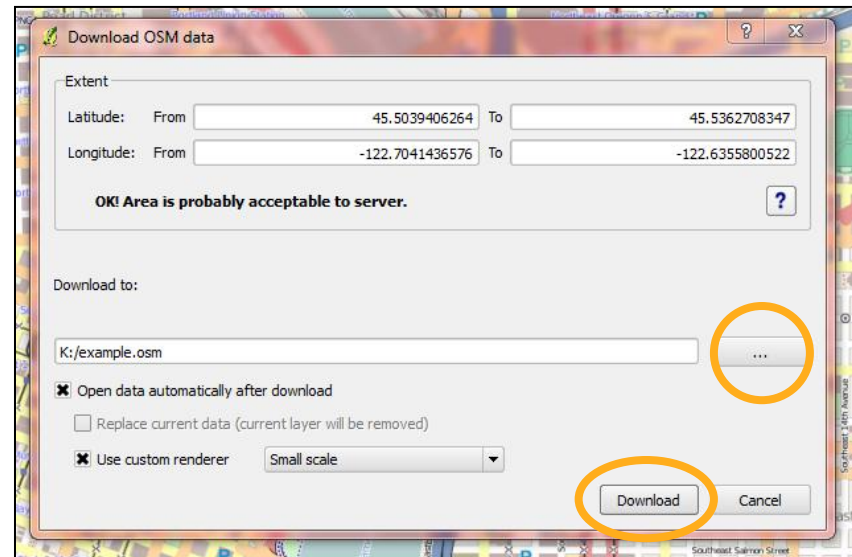
*Not too big, now!*

# Step 3: Download from OSM

- Click the “Download OSM data” button (blue arrow pointing down)
- If you can't see the button, look for and expand the plugin's toolbar

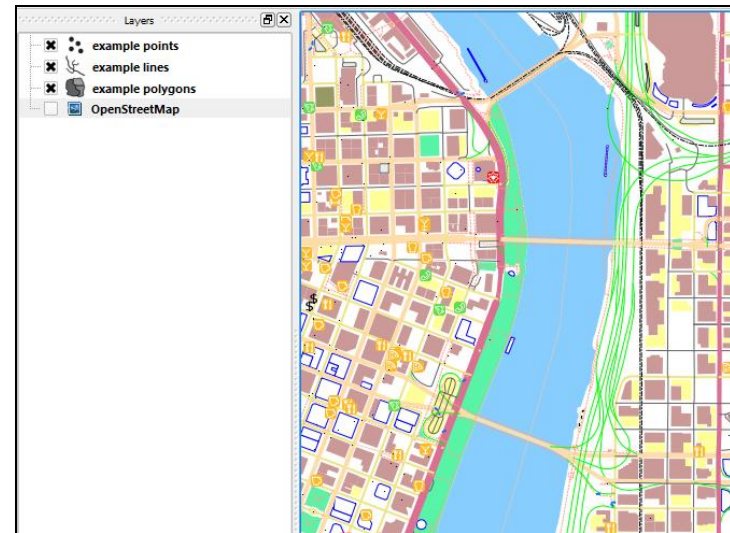
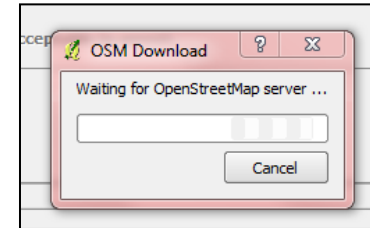


- The extent should be filled automatically
- Click the “...” button to browse for where you want to save the .osm file
- Click “Download”



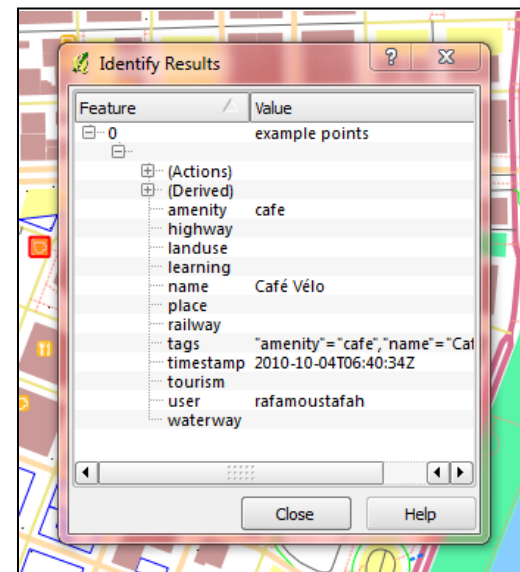
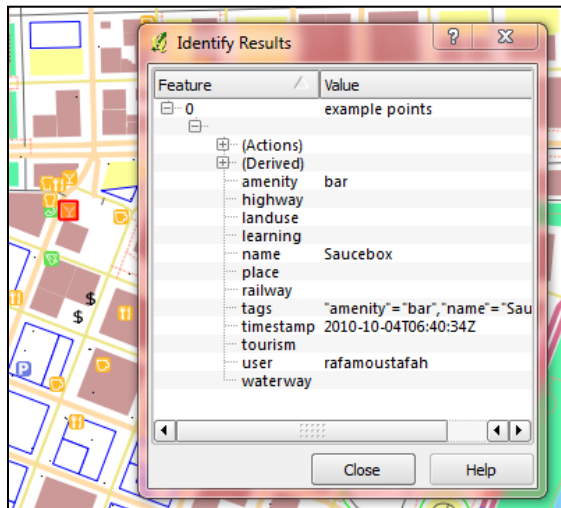
# Step 4: Wait... then view

- Again, you can do faster, larger downloads via JOSM's "Mirrored download" plugin, and then open them in QGIS for the next step (though QGIS will still take a while to process large files)
- Check it out! You may want to turn off the OpenLayers OSM Layer



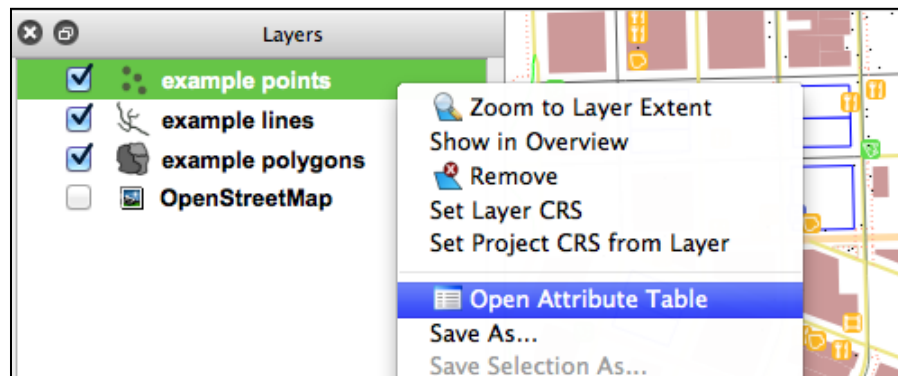
## Step 5: Explore

- Use the identify tool to check out the attributes of some points (may need to select that layer first)



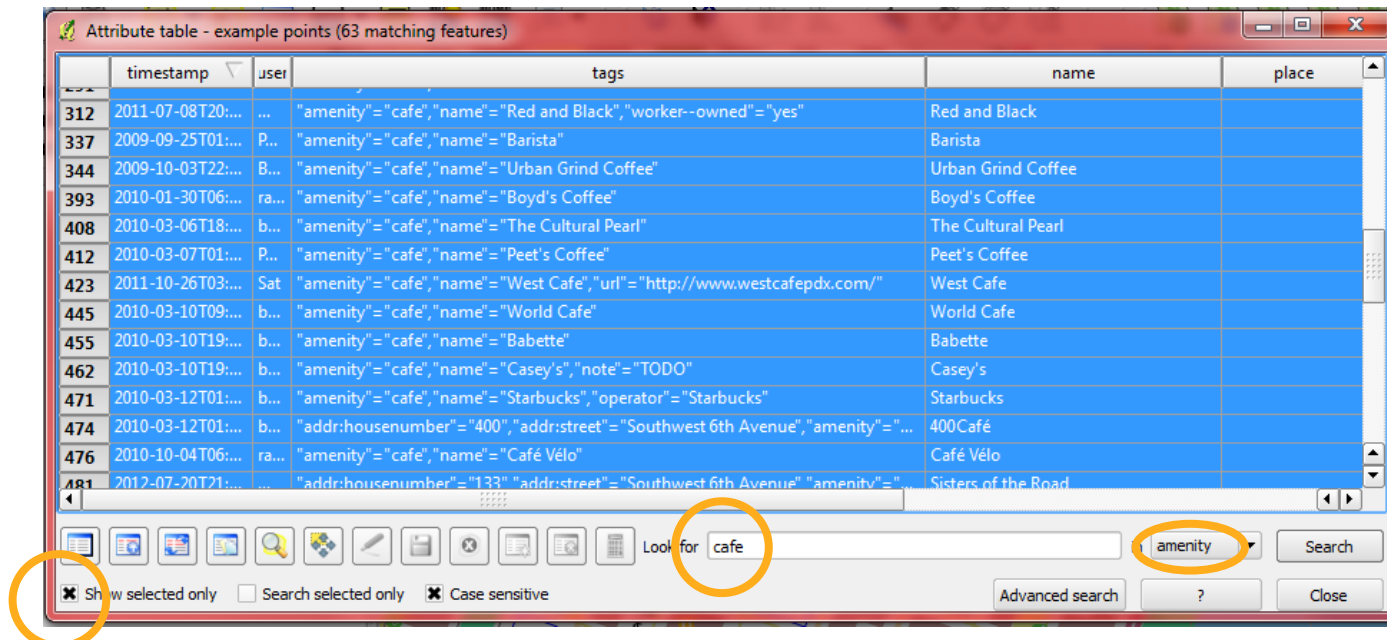
## Step 6: Choose what you want

- Right-click on the points layer and select “Open Attribute Table”



## Step 6: Choose what you want

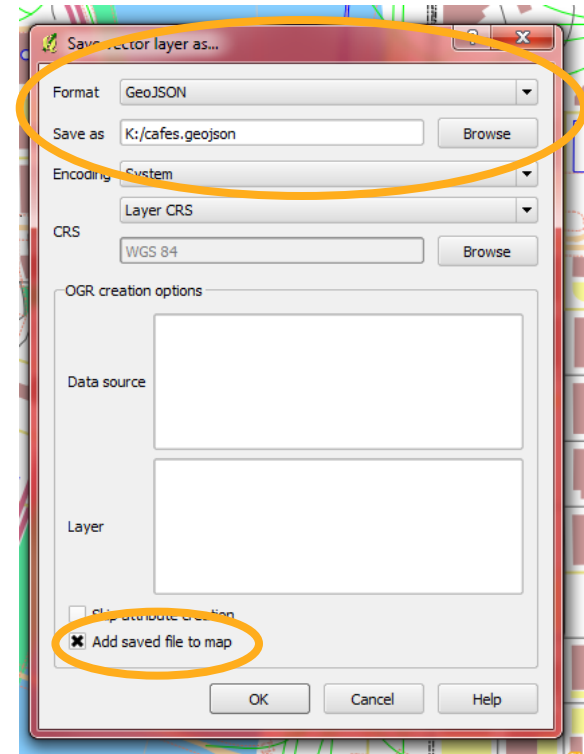
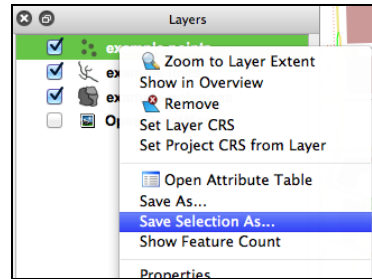
- Look for **cafe**, in **amenity**
- Click the box for “**Show selected only**”





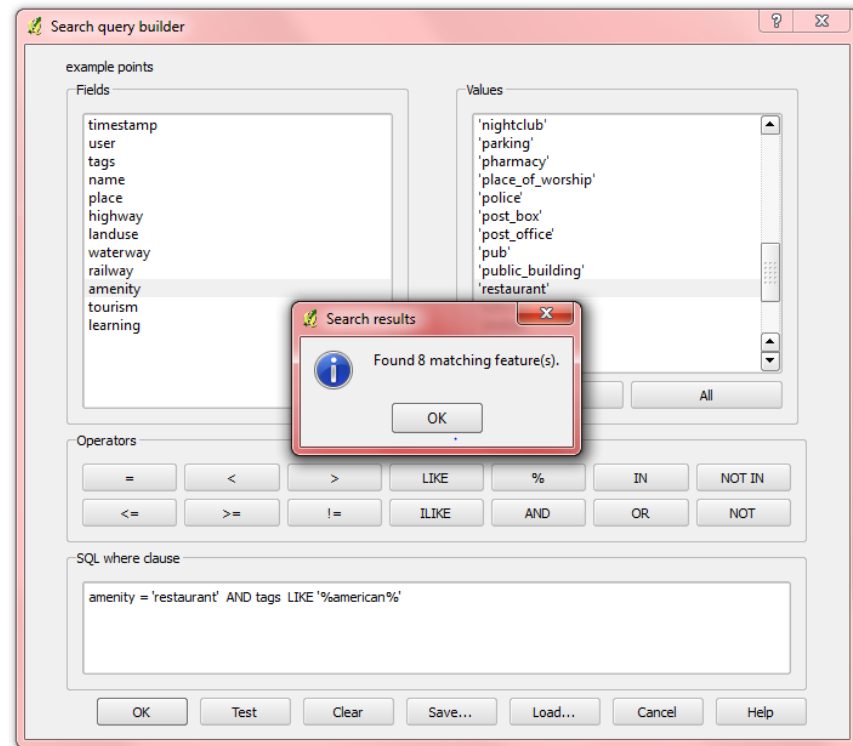
## Step 7: Save it as a GeoJSON

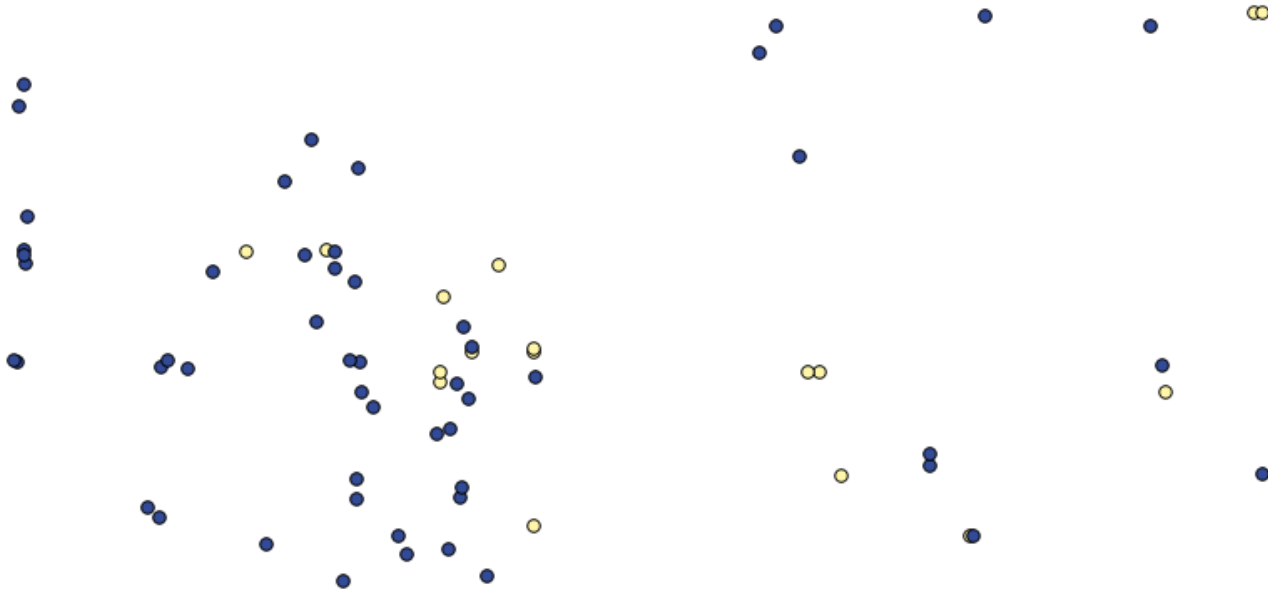
- Right-click on the layer and choose “Save Selection As...”
- Choose GeoJSON for the format
- Browse for a file location and choose a name
- Click the box for “Add saved file to map”



## Step 8: Do it again

- Repeat with a different type of POI
- Grant and I are here to help!
- I'm choosing bars
- You can also use the "Advanced search" button to use the Search query builder for more complex queries
- Additional tags should be in the 'tags' field

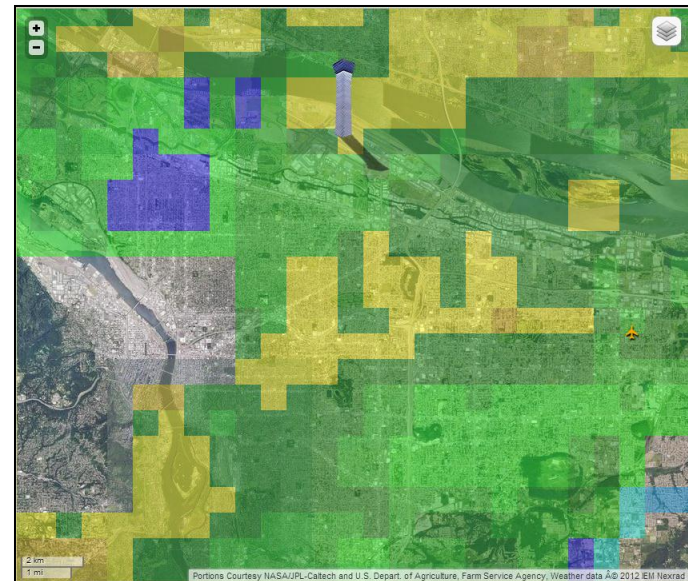




# Putting GeoJSON OSM POIs into a Leaflet Map

# Step 1: Getting started

- Let's build off of one of Wm's examples (Exercise 9)
- Oh no, the rain is coming! Where are some bars or coffee shops that I can hide out in for a while?
- How does the code need to be modified?



## Step 2: Adapt what we have

- Let's get rid of:
  - The airport (and its icon)
  - The planes
  - Google street overlay
- Do this yourself, or get some tidy files to start with at <http://pdxmele.github.com/OSM-Leaf-example>
- What we'll be doing:
  - Changing the zoom and center
  - Adding a couple of new GeoJSON layers

## Step 3: Change the center & zoom

- <http://itouchmap.com/latlong.html> and similar tools can help you determine your new map center
- Paste it in

```
var map = L.map('map_div', {  
  center: [45.521115, -122.673383],  
  zoom: 14,  
});
```

- Increase the zoom

**iTouchMap.com**  
Mobile and Desktop Maps

Maps | [Country - State](#) | [Places](#) | [Google Earth](#) | [Cities](#) | [Earthquakes](#) |

Home » [Latitude and Longitude of a Point](#)

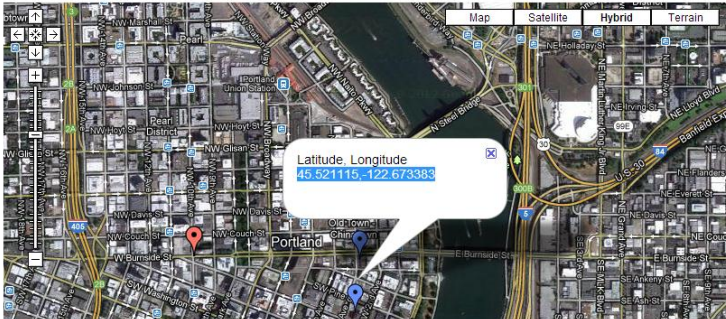
To find the latitude and longitude of a point **Click** on the map. **Drag** the marker, or enter the...

**Address:**

**Map Center:** [Get Address](#) - [Land Plat Size](#) - [Street View](#) - [Google Earth 3D](#) - [Area Photographs](#)

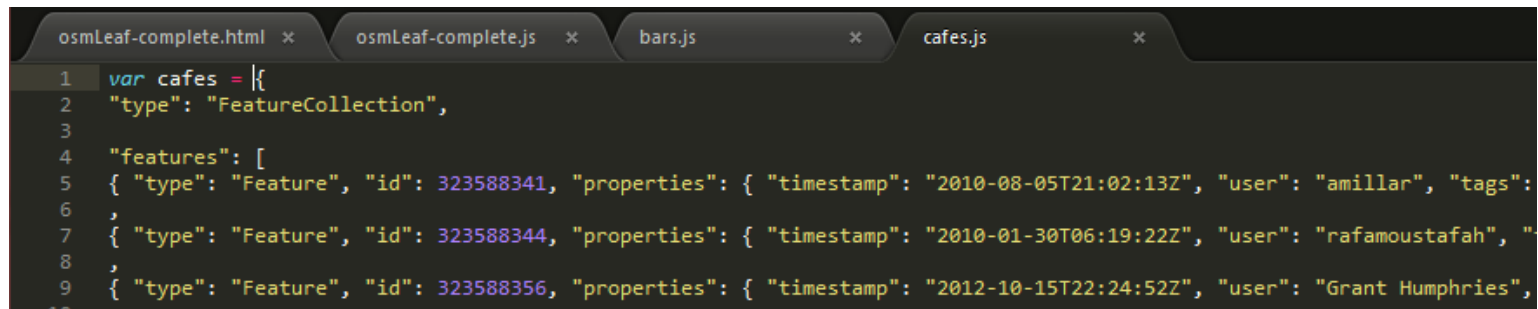
Try out the [Google Earth Plug-in](#). Google Earth gives you a 3D look of the area around the center of the map, which is usually your last click point, and includes latitude, longitude and elevation information.

**Latitude and Longitude of a Point**



## Step 4: Prepare the GeoJSON

- So what is GeoJSON anyway? Open up your files in SublimeText and have a look
- They are just specially-formatted JavaScript files
- Save them as .js files instead of .geojson files and put them in the same folder as your other files
- Add “var *name* = ” (e.g. var cafes =) to the beginning of each document

A screenshot of a code editor with four tabs: 'osmLeaf-complete.html', 'osmLeaf-complete.js', 'bars.js', and 'cafes.js'. The 'cafes.js' tab is active, showing a JavaScript object representing a GeoJSON FeatureCollection. The code is as follows:

```
1 var cafes = {  
2   "type": "FeatureCollection",  
3  
4   "features": [  
5     { "type": "Feature", "id": 323588341, "properties": { "timestamp": "2010-08-05T21:02:13Z", "user": "amillar", "tags":  
6       ,  
7     { "type": "Feature", "id": 323588344, "properties": { "timestamp": "2010-01-30T06:19:22Z", "user": "rafamoustafah", "  
8       ,  
9     { "type": "Feature", "id": 323588356, "properties": { "timestamp": "2012-10-15T22:24:52Z", "user": "Grant Humphries",  
10
```



## Step 5: Adding the GeoJSON layers to your map

- Add references to these files in your HTML file (put them before the Leaflet js and jQuery)

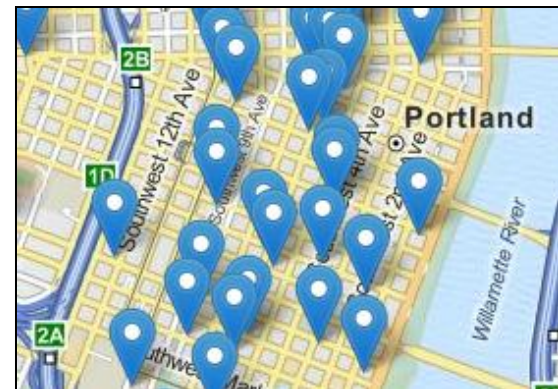
```
<script src="bars.js" type="text/javascript"></script>  
<script src="cafes.js" type="text/javascript"></script>
```

- Add this after weather:

```
var cafelayer = L.geoJson(cafes);  
var barlayer = L.geoJson(bars);
```

- Update your layer control:

```
var layercontrol = L.control.layers(  
  { street: osmtiles, satellite: sattiles },  
  { radar: weather, cafes: cafelayer, bars: barlayer }  
);
```



## Step 6: Improve it with styling

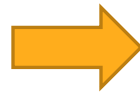
- Let's make it so we can tell our layers apart
- Start with this code, and mimic it for your second dataset

```
var cafemarkers = {  
  radius: 6,  
  fillColor: "#ff7800",  
  color: "#000",  
  weight: 1,  
  opacity: 1,  
  fillOpacity: 0.8  
};  
var cafelayer = L.geoJson(cafes, {  
  pointToLayer: function (feature, latlng) {  
    return L.circleMarker(latlng, cafemarkers);  
  }  
});
```



## Step 7: Popups

- Add this section before your GeoJSON layers

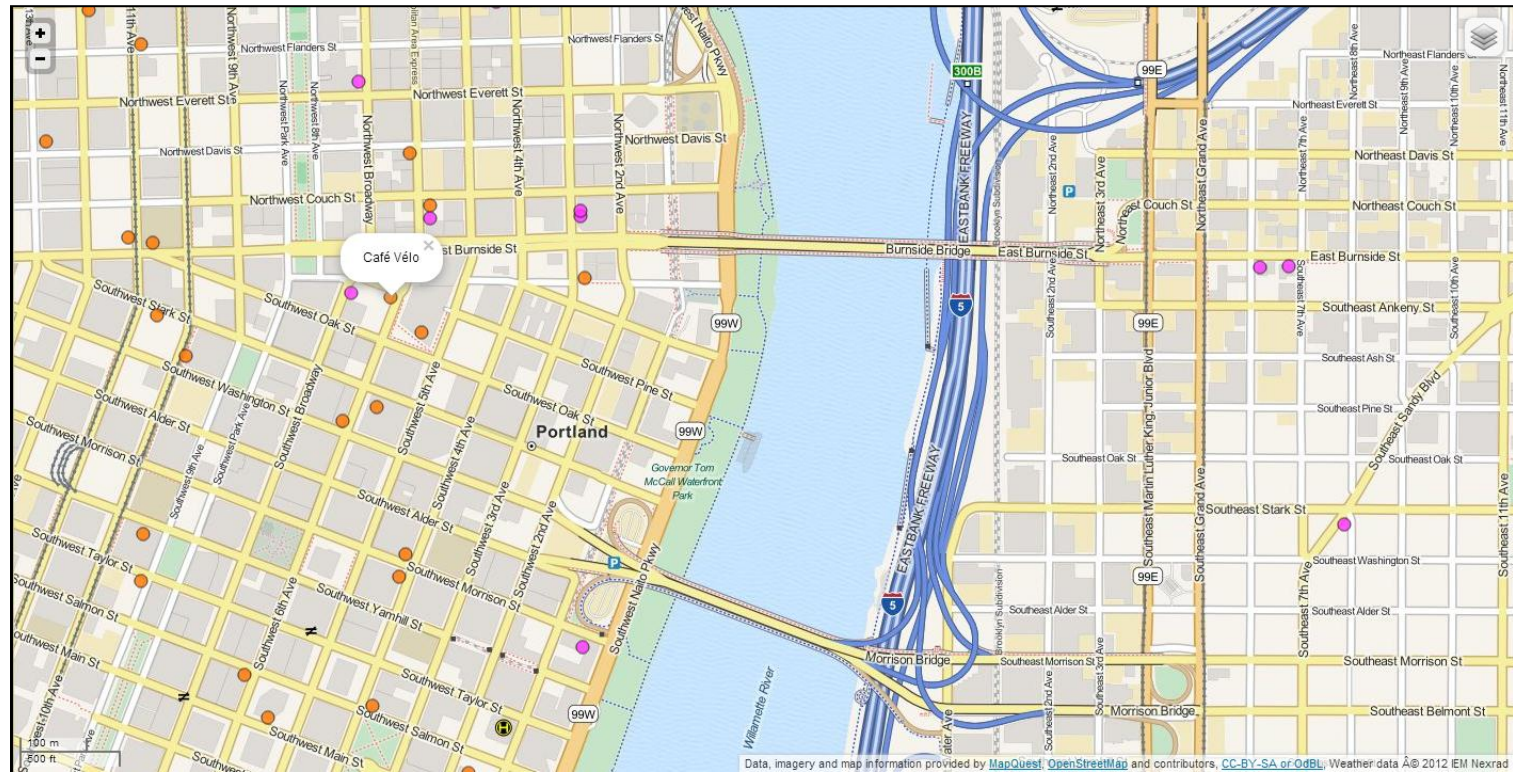


- Add this highlighted bit to each layer



```
function onEachFeature(feature, layer) {  
  // grabs the property "name" and uses it for the popup  
  if (feature.properties && feature.properties.name) {  
    layer.bindPopup(feature.properties.name);  
  }  
  
  var cafemarkers = {  
    radius: 6,  
    fillColor: "#ff7800",  
    color: "#000",  
    weight: 1,  
    opacity: 1,  
    fillOpacity: 0.8  
  };  
  
  var cafelayer = L.geoJson(cafes, {  
    pointToLayer: function (feature, latlng) {  
      return L.circleMarker(latlng, cafemarkers);  
    },  
    onEachFeature: onEachFeature  
  });
```

# Looking good!



# One last word on attribution

- The OpenStreetMap license has recently changed (it was previously CC-by-SA)
- You need to credit OSM properly somewhere in your map or on your webpage
- Since we're already using an OSM tile layer, we can just modify the attribution for that (we should really fix that anyway)

```
k">MapQuest</a> and © <a href="http://www.openstreetmap.org/copyright" target="_blank">OpenStreetMap contributors</a>
```

Congratulations!

**Great job!**

**The final code is at:**

[http://pdxmele.github.com/  
OSM-Leaf-example/complete](http://pdxmele.github.com/OSM-Leaf-example/complete)