MAP SCRIPTING.com
start here guide

Easy St

Table of Contents

This *Start Here Guide* is written by Adam DuVander

To download the latest version of this guide for free, please visit:

   http://mapscripting.com/start-here-guide

Map Scripting Start Here Guide

Is it time to add some "where" to your website? For most of us, this answer should be an emphatic "yes." That's because location on the Web is becoming more important. Luckily, as you'll see in this collection of tutorials, it's also very easy to include some geographic elements on your website.

The Map Scripting Start Here Guide provides a brief introduction to creating your own maps. You'll learn how to:

1. **Write code once** that will work on Google Maps, Yahoo Maps and other mapping providers.

2. **Stand out** with custom marker icons and add a message window with details about the place.

3. **Make things easier** by auto-centering and provide controls to zoom and switch map types.

Sound good? Read about all that and a few more tricks.

Create a Basic Map

Before you make a map, you need to decide whether to use Google, Yahoo or one of the many other choices, right? Well, yes and no. Wherever possible, including throughout this guide, I show mapping examples using an open source project called Mapstraction. At the same time, the examples use Google Maps. How is that possible?

Like other mapping APIs, Mapstraction is based on JavaScript. However, once you've written for Mapstraction, switching from one provider to another is a cinch. So, while you'll need to choose a provider now, the decision isn't forever when you use Mapstraction.
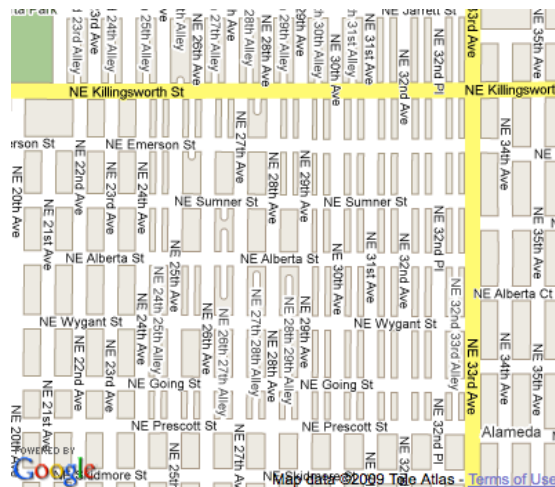
What You'll Need:

- Download Mapstraction from http://mapstraction.com

- Sign up for a Google API key from http://code.google.com/apis/maps/

Now you're ready to create a new map. Create an HTML file and add this code:

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Basic Mapstraction Map</title>
    <script src="http://maps.google.com/maps?
file=api&amp;v=2&amp;key=YOURKEY"
    type="text/javascript"></script>
    <script src="js/mxn.js?(google)"></script>
    <style>
      div#mymap {
        width: 400px;
        height: 350px;
      }
    </style>
    <script type="text/javascript">
      var mapstraction;
      function create_map() {
        mapstraction = new mxn.Mapstraction('mymap', 'google');
        mapstraction.setCenterAndZoom(
            new mxn.LatLonPoint(45.559242, -122.636467), 15);
      }
    </script>
  </head>
  <body onload="create_map()">
    <div id="mymap"></div>
  </body>
</html>
```

Does that look like the code for a normal web page? That's because it basically is. The map is added with JavaScript, hooking itself to the "mymap" div tag (which you could name anything you want, as long as the names match in the HTML and JavaScript).

5

To make this code work, you'll need to do two things:

1. Replace YOURKEY with your Google API key.

2. Make sure mxn.js (and the rest of the Mapstraction code) is in the "js" directory, one level below your HTML file.

You can set the size of your map by changing the CSS within the <style> tag. You could also move those styles into an external stylesheet, if you prefer.

I have put all the JavaScript to create the map inside a function I named (not-so-creatively) create_map. There it tells Mapstraction that is is creating a Google Map, then sets the center of the map on a geographic point, declared by a latitude/longitude pair.

The location I chose to center the map is my favorite Mexican restaurant in Portland. You may want to choose a different spot (unless you, too, are a La Bonita fan). How do you do that? What is this latitude/longitude stuff? Read on to find out a little more.

How to Find Latitude and Longitude

Some time ago geographers took the earth, its sphere hard to represent on paper, and stretched it out onto a flat map. Then, they split that map into quadrants, like an X/Y graph from algebra class, only the vertical axis is called latitude and the other longitude.


Every city, every restaurant, every place has its own coordinates which show its place on the earth. There are a number of ways to find these, so I'll show a couple methods for determining them using web tools that I have created.

Geocode in a Click

Will you recognize the places you want to convert to latitude/longitude points if you see them on a map? If so, you want the click-to-geocode tool, which can be found here:
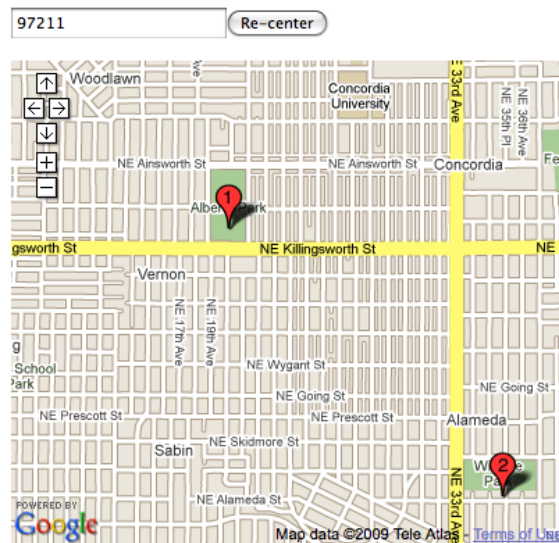
http://mapscripting.com/examples/guide/clickpoint.html

Use the zoom tools, drag the map, or type a city name / postal code to zero the map into the neighborhood where you want to find coordinates. Click on the spot you want and your latitude/longitude points will display below the map. Click a few more times to get more coordinates.

Type an Address, Get a Geo Point

Have a few addresses you want to turn into coordinates? Try out my geocoding tool here:

http://mapscripting.com/examples/guide/address.html
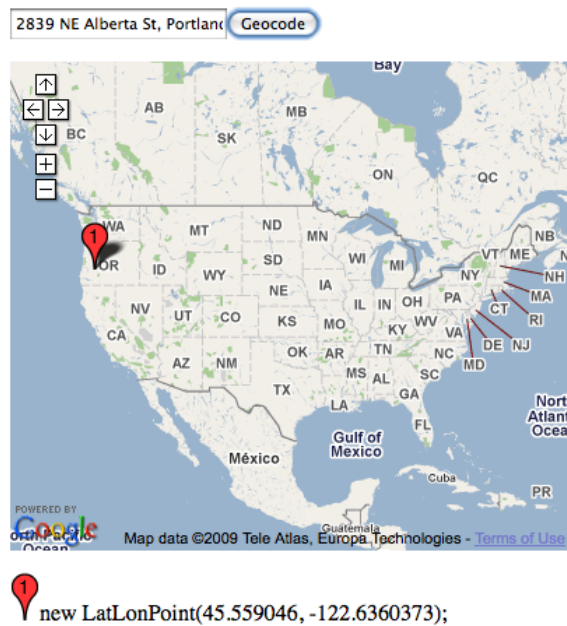
Type in a city name, postal code, or full address. Your latitude/longitude points will display below the map. Type another address to get more coordinates.

Add Markers to Your Map

Now that you can create a map and find coordinates, let's plot some locations. Again, I'm going to use that Mexican restaurant as an example. Feel free to substitute your own place.

Do you still have the first example, that basic map? Let's try adding a few lines to the create_map function:

```
function create_map() {
  mapstraction = new mxn.Mapstraction('mymap', 'google');
  mapstraction.setCenterAndZoom(
    new mxn.LatLonPoint(45.559242, -122.636467), 15);
  var mk = new mxn.Marker(
    new mxn.LatLonPoint(45.559242, -122.636467));
  mapstraction.addMarker(mk);
}
```

I added the two lines in bold. They create a marker, give it the same coordinates as the map's center, then add it to the map.
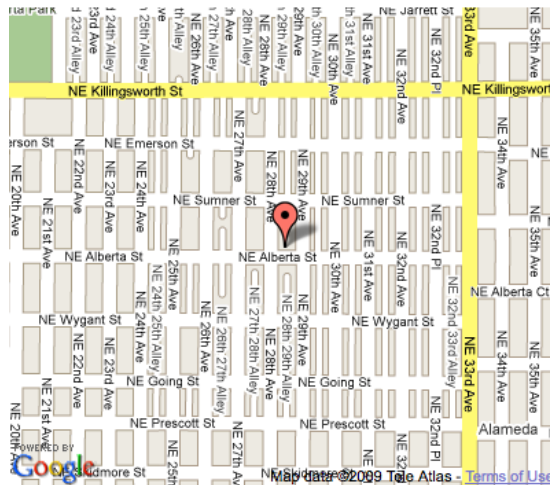
If I wanted to add another marker, I'd add two more lines, changing just the coordinates.

9

Creating markers is easy, but does it look too plain? Let's spice them up.

Give Markers a Message

I've shown how markers can be used to *mark* a location. Now I'll give that marking a little more meaning by adding a message to be shown when the user clicks the marker.

Remember the two lines we used to add a marker to the page? We'll insert one line between them now, which will set the marker's message:

```
function create_map() {
  mapstraction = new mxn.Mapstraction('mymap', 'google');
  mapstraction.setCenterAndZoom(
    new mxn.LatLonPoint(45.559242, -122.636467), 15);
  var mk = new mxn.Marker(
    new mxn.LatLonPoint(45.559242, -122.636467));
  mk.setInfoBubble("La Bonita tastes good");
  mapstraction.addMarker(mk);
}
```

Pretty simple, eh? Go ahead and click the marker.
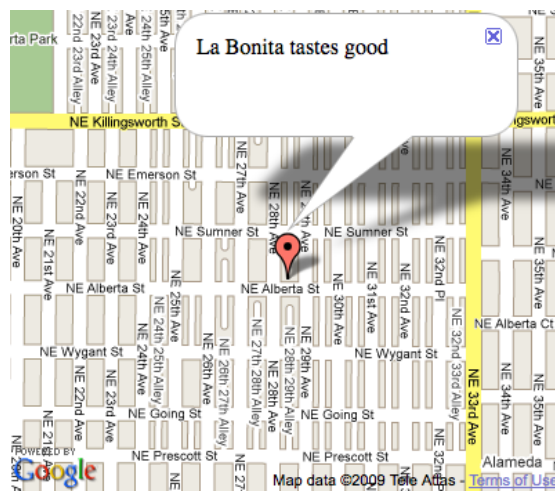
Use Mapstraction's infoBubble to store extra info about a place. You can even use HTML to add links, pictures and more. It's handy and your users will thank you for helping your map make even more sense.

Customize the Icon Graphic

If there's one thing you can do to set your map apart from all the others, it's using your own graphic in place of the default marker. Google, for example, uses a reverse teardrop, which has come to cry out "generic map." Let's change it up.

Instead of the boring marker showing my favorite Mexican restaurant, let's use a little sombrero graphic I made. The icon can be whatever size you want, but it is probably best to keep each dimension between 20 and 50 pixels. My sombrero is 40 pixels wide and 24 pixels tall.

Google also shows a shadow for each marker. Rather than create this myself, I used the Shadowmaker tool available at http://www.cycloloco.com/shadowmaker/

Both of these files can be downloaded from Map Scripting at http://mapscripting.com/custom-markers

Armed with these two graphics, here's a create_map function to add to your basic map:

```
function create_map() {
  mapstraction = new mxn.Mapstraction('mymap', 'google');
  mapstraction.setCenterAndZoom(
    new mxn.LatLonPoint(45.559242, -122.636467), 15);
  var mk = new mxn.Marker(
    new mxn.LatLonPoint(45.559242, -122.636467));
  mk.setIcon('sombrero.png', [40,24]);
  mk.setShadowIcon('sombrero-shadow.png', [53,24]);
  mapstraction.addMarker(mk);
}
```

We set the two graphics (the lines in bold above) after the marker is created, but before it's been added to the map. The code assumes the graphics are in the same directory as the map HTML file, though you could include a relative path (such as img/sombrero.png) to

11

store it with other images.

Repeating these same lines, we could add many markers with the same icon. Omit the lines and you'll still have default markers. Change the graphic names and, if the graphics exist, you can have a map with many different marker icons.

I like custom markers because they are an easy way to make your map unique. They can also help the markers make more sense. For example, show a company's logo as a marker, or add a different icon for types of restaurants.

Plus, they're a lot of fun, aren't they?

Automatically Center the Markers

Do you have more than one marker on your map yet? When you do, things start to get complicated. The center of your map is no longer in the best spot. One of your markers might not even be visible, which is a big bummer.

Mapstraction simplifies working with multiple markers by providing a function to automatically show all your markers. On the next page you'll find a create_map function that has two markers and intelligently sets the center and the zoom level.

Add these lines to your basic map, replacing your previous create_map function:

```
function create_map() {
  mapstraction = new mxn.Mapstraction('mymap', 'google');
  mapstraction.setCenterAndZoom(
    new mxn.LatLonPoint(45.559242, -122.636467), 15);
  var mk1 = new mxn.Marker(
    new mxn.LatLonPoint(45.559242, -122.636467));
  mapstraction.addMarker(mk1);
  var mk2 = new mxn.Marker(
    new mxn.LatLonPoint(45.559046, -122.646861));
  mapstraction.addMarker(mk2);
  mapstraction.autoCenterAndZoom();
}
```

If you try loading this map without the bolded line, it will only show one marker. Or, so you'll think! The second marker will be there, off to the side of the visible portion of the map.
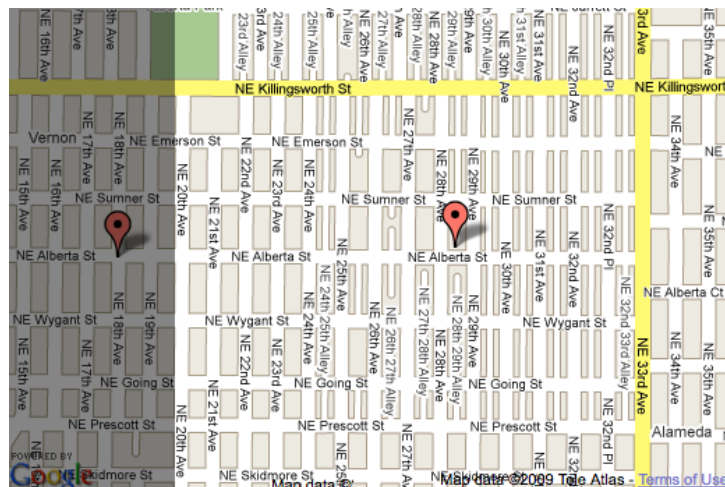
If you were to drag the map to the side, you'd see the second marker.

13

Now load the map with all the code above, including the bolded auto-center line. Your map now shows all the markers within the visible area. Mapstraction finds the center of all the markers, then sets the appropriate zoom level.
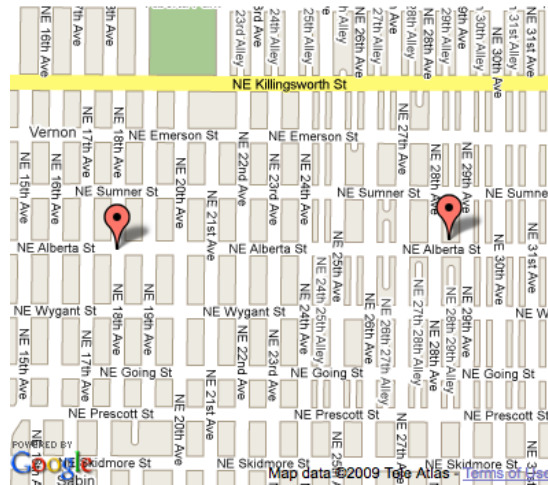
Wouldn't you rather not have to worry about markers going missing? Mapstraction makes it easy by making sure everything on the map is within view.

Give the User Control

Okay, now you have some really nice maps, right? They look just like you want them. It's time to let your user have a hand in navigating. There are a couple controls that any mapping provider will let you put on a map. In this section, I'll show how to let your user zoom and change the map type.
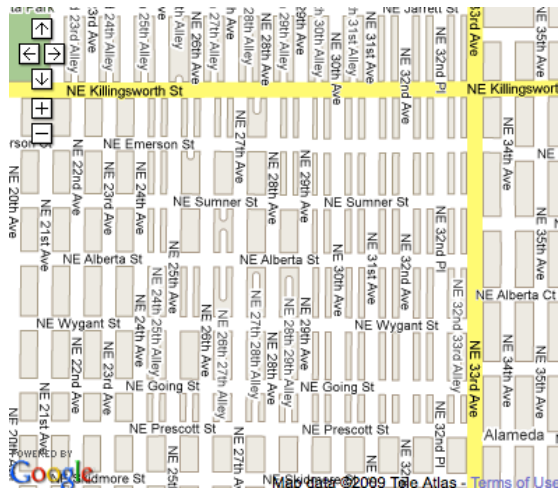
Add a Zoom Interface

Previously I've shown you how to set the zoom level automatically, so that it shows all the markers. So, why would users ever need to zoom? Perhaps they want to focus on what's near only one marker, or see more of the neighborhood to gain a little context. Mapping providers let you give zooming power to your users, so you might as well pass it on.

Here is a create_map function that contains a zoom control:

```
function create_map() {
  mapstraction = new mxn.Mapstraction('mymap', 'google');
  mapstraction.setCenterAndZoom(
    new mxn.LatLonPoint(45.559242, -122.636467), 15);
  mapstraction.addSmallControls();
}
```

The line in bold adds a small zoom interface to the map.

If you want a larger zoom interface, try:

**mapstraction.addLargeControls()**

Add a Map Type Interface

Maps are neat, but there is more you can show than simply streets and such. Many large providers also have satellite imagery, which gives a top-down view of a place. Some users will prefer the satellite, while others like the normal map. Why not give them a choice?

15

Here is a create_map function that contains a map type control:

```
function create_map() {
  mapstraction = new mxn.Mapstraction('mymap', 'google');
  mapstraction.setCenterAndZoom(
    new mxn.LatLonPoint(45.559242, -122.636467), 15);
  mapstraction.addMapTypeControls();
}
```

The line in bold adds a map type interface to the map.
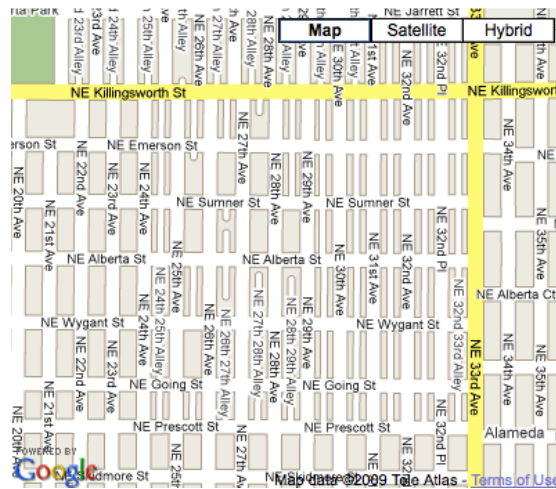
Most providers give three choices: normal, satellite and hybrid. The latter is a combination of the first two, with streets layered over satellite imagery.

What's Next?

Do you feel like a map champ yet? The basics that you've learned so far can help you add some location to your website. You'll be happy and your users might even thank you, too.

If you're itching for more mapping, here are some additional resources:

- http://mapscripting.com – My website where I have several examples and blog about mapping APIs.

- http://mapstraction.appspot.com/ - The Mapstraction sandbox has editable code showing many features of the mapping platform.

- http://mapscripting.com/book/ - My upcoming cookbook, with over 50 projects to get some "where" in your website.

Please contact me and let me know how useful this Start Here Guide has been for you and what else you would like to see in future guides.