



GIS Career Awareness Learning Module A

Version 4.1 June 25, 2009

Google Map API Tutorial

- **Estimated Time:** 1.5 hours
- **Learning Goal:** This module will introduce GIS web-based applications using the Google Maps API.
- **Platforms:** PC, Mac, or UNIX.
- **Software Tools Required:** Web browser with high-speed Internet access and a text editing program (such as Notepad or WordPad).
- **Allied to California Content standards:** GIS Learning module A can be allied to the following California Content standards (<http://www.cde.ca.gov/be/st/ss/>):

English-Language Arts Content Standards (December, 1997)

- 1.1 (Writing) Establish a controlling impression or coherent thesis that conveys a clear and distinctive perspective on the subject and maintain a consistent tone and focus throughout the piece of writing.

Technology Standards (Agriculture and Natural Resources Industry Sector May, 2005)

- 4.2 Understand the use of technological resources to gain access to, manipulate, and produce information, products, and services.
- 4.4 Understand geographic information systems (G.I.S.).

Note: This Web-based GIS learning module was developed with support from a National Science Foundation – Advanced Technological Education Program Research Grant (NSF-ATE DUE # 0801893): “National Geospatial Technology Center.”

(<http://geoinfo.sdsu.edu/hightech>)

**ANSWER ALL QUESTIONS ON THE SEPARATE ANSWER
SHEET**

Learning Module a Table of Contents

Part 1 – Embedding Google Maps	4
1.1 Get Standard Google Map Display Link	4
1.2 Embedding Map Link in a Google Sites Page	6
Part 1 – Questions.....	10
Part 2 – Displaying Custom Information with Google Maps	11
2.1 Overlaying a KMZ on top of a Google Maps base map.....	11
2.2 Adding Custom Features to Google Maps.....	12
Part 2 – Questions	14
Part 3 – Embedding and Customizing Google Maps Using the Google Maps API ...	15
Part 3 – Questions.....	24
Part 4, Optional – Creating a KMZ File Using ESRI ArcMap 9.3	25
4.1 Cartographic design in ArcMap	25
4.2 Exporting KMZ file	27
4.3 Previewing KMZ file in Google Earth	28
Part 4 - Questions	30
Part 5, Optional – JavaScript Programming	31
5.1 JavaScript Basics	31
5.2 Variables	31
5.3 Functions.....	33
5.4 JavaScript in Google Maps	34
Part 5 – Questions.....	36
Student Survey Learning Module A.....	37

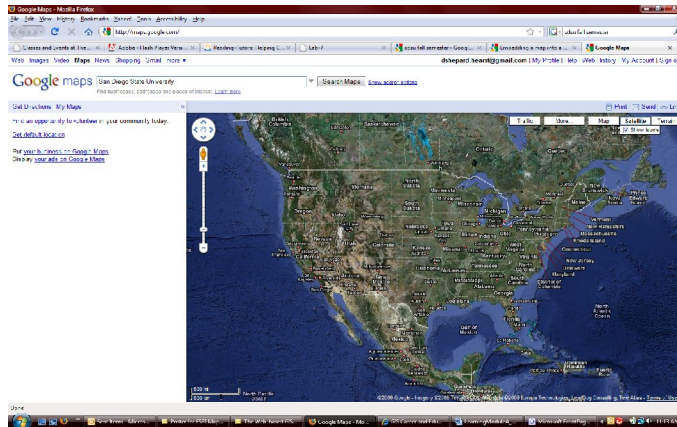
Part 1 – Embedding Google Maps

To finish this part, you may need to have access to the Internet, Web Browser, and a Google Sites account (if you don't have a Google Sites account we will explain how you can create one for free in section 1.2)

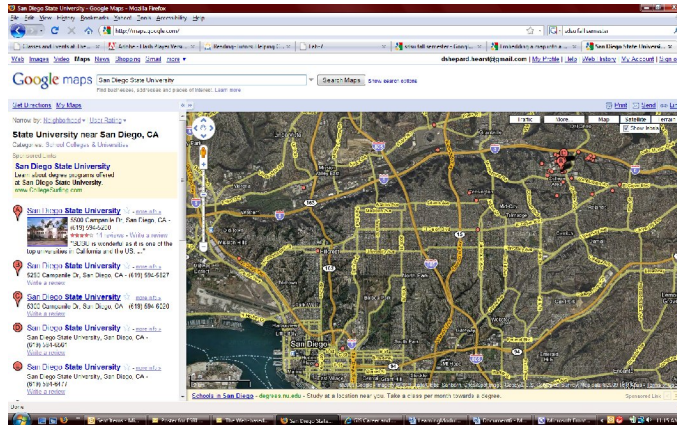
1.1 Get Standard Google Map Display Link

This section explains how to generate a link for embedding a Google Map.

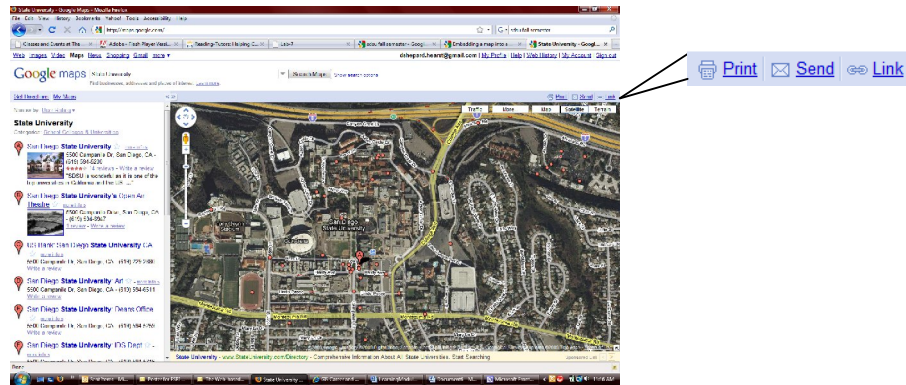
1. Go to Google Maps, maps.google.com and search for a point of interest. In our example we will be using San Diego State University as our example. The search may return more than one result, this is fine.



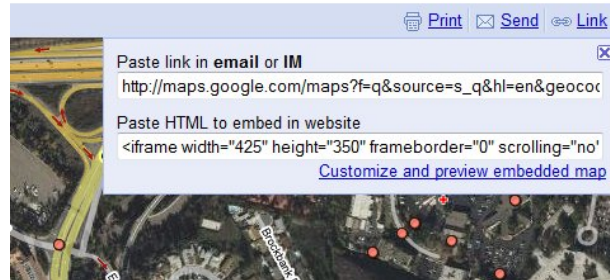
2. The results of your search will be displayed on the map.



3. You can pan around the map and zoom in to a level that you like.



4. Click 'Link' and copy both the 'Paste HTML to embed in website' field and the 'Paste Link in email or IM' field.



5. Open your text editor (Notepad, WordPad, Word, etc.) and paste the links into the new document... we'll come back to this text later. Below is the link code for our SDSU example.

```
<iframe width="300" height="300" frameborder="0"
scrolling="no" marginheight="0" marginwidth="0"
src="http://maps.google.com/maps?f=q&source=s_q&
amp;hl=en&geocode=&q=San+Diego+State+Univers
ity& sll=32.774502,-117.069819& sspn=0.010627,
0.022745& ie=UTF8& t=h& radius=0.66& rq=
1& ll=32.775296,-117.073617& spn=0.010825,0.01
2875& z=15& output=embed"></iframe>
<br/>
<small><a
href="http://maps.google.com/maps?f=q&source=emb
ed&hl=en&geocode=&q=State+University&
sll=32.774502,-117.069819& sspn=0.010627,0.02274
5& ie=UTF8& t=h& radius=0.66& rq=1& l
l=32.775296,-117.073617& spn=0.010825,0.012875& am
p; z=15" style="color:#0000FF;text-align:left">View
Larger Map</a></small>
```

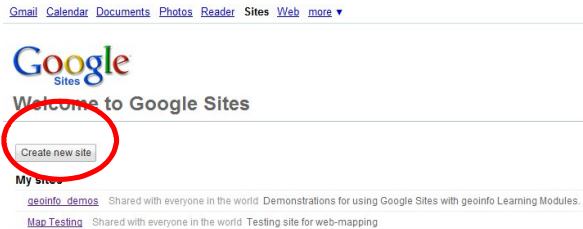
1.2 Embedding Map Link in a Google Sites Page

In this section we will create a Google Sites Page to use for embedding the map from section 1.1. If you are using a web development program (such as Dreamweaver, FrontPage, and SharePoint) you can past the HTML link provided in step 5 of section 1.1 directly into your HTML code.

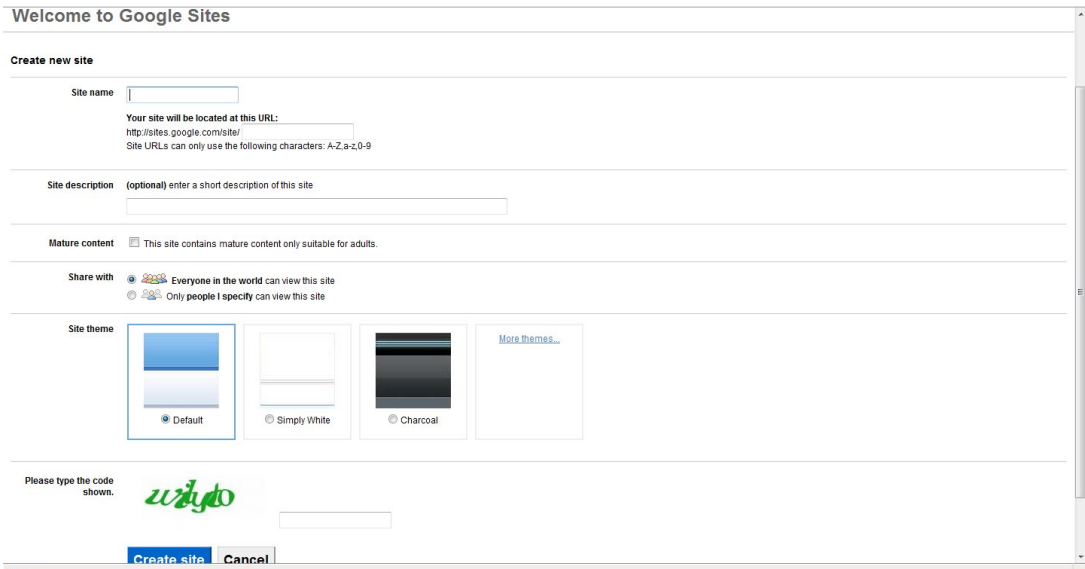
1. To create a new Google Sites account, go to <http://sites.google.com/>. Login to your Google account; if you don't already have a Google account you can follow the link provided on this page to create an account.

*** Skip to step 4 if you already have a Google Sites account that you want to use.

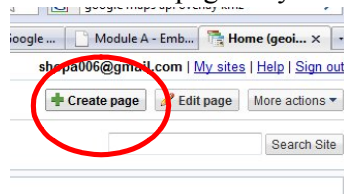
2. Create a new site by clicking 'Create New Site'. You can also access existing sites here.



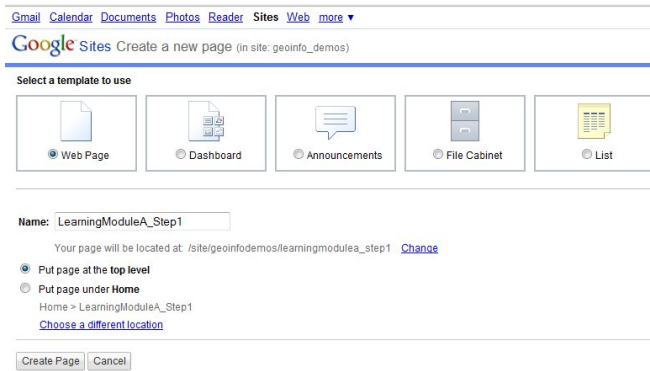
3. Complete the form for the new site and click 'Create Site'.

A screenshot of the 'Create new site' form in Google Sites. The form is titled 'Welcome to Google Sites' and 'Create new site'. It contains several sections: 'Site name' with an input field; 'Your site will be located at this URL:' with a pre-filled URL 'http://sites.google.com/site/' and a note that site URLs can only use characters A-Z, a-z, 0-9; 'Site description (optional) enter a short description of this site' with an input field; 'Mature content' with a checkbox and the text 'This site contains mature content only suitable for adults.'; 'Share with' with two radio button options: 'Everyone in the world can view this site' (selected) and 'Only people I specify can view this site'; 'Site theme' with three theme thumbnails: 'Default' (selected), 'Simply White', and 'Charcoal', plus a 'More themes...' link. At the bottom, there is a CAPTCHA section with the text 'Please type the code shown.' and a distorted image of the word 'wzjydo'. A 'Create site' button is circled in red.

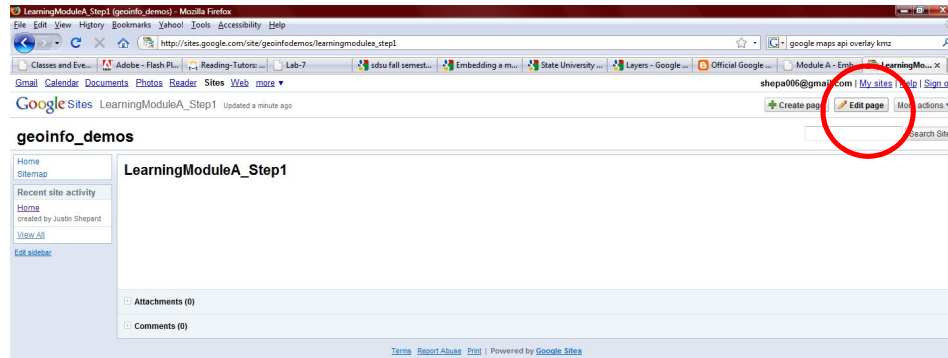
4. Create a new page on your site by clicking 'Create Page'.



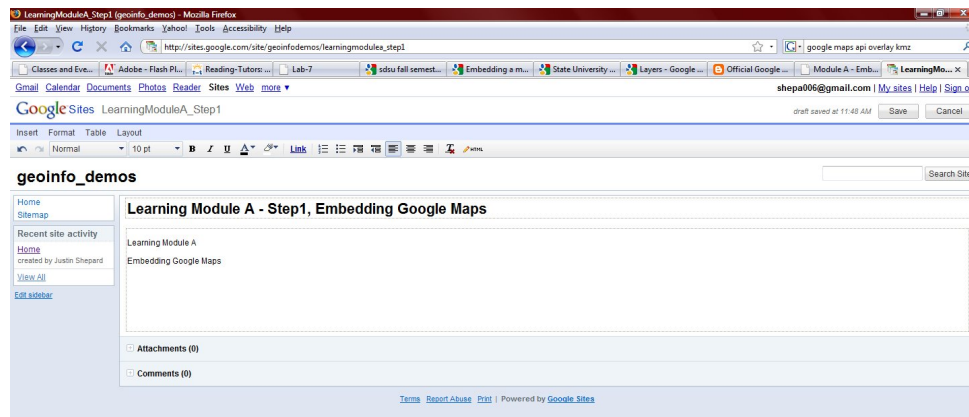
5. Select 'Web Page' for the style and click 'Create Page'. You can leave the 'Put page at the top level' option selected.



6. You will be returned to a list of pages on your site, with your new page selected. Click 'Edit Page' to edit the selected page.



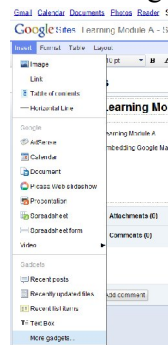
7. The page text can be typed directly into the page layout shown in the editor.



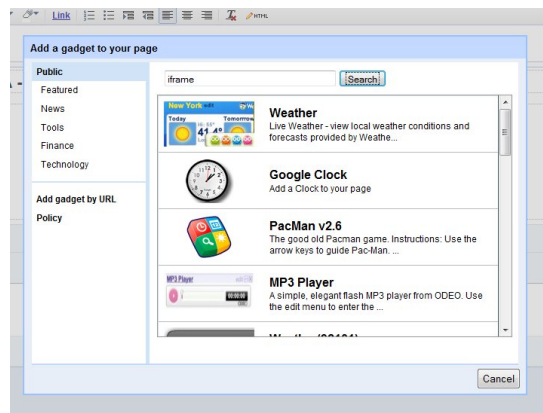
8. Insert Link to Google Map

The link copied from the Google Maps page can **NOT** be pasted directly into the Google Sites page; first we need to add an iFrame gadget to our site page.

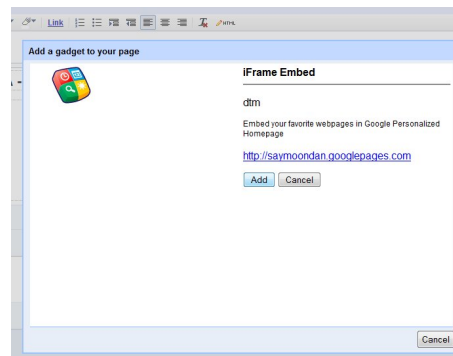
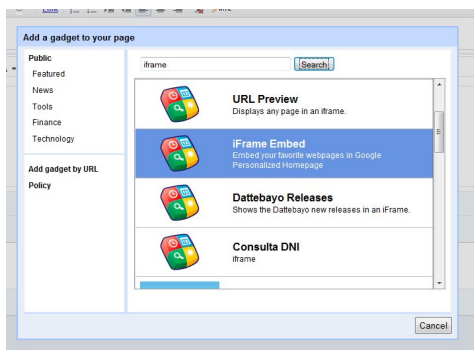
Put the cursor where you want the map to be displayed and then you will add the iFrame gadget.



Click 'Insert' → 'More Gadgets' and search for iFrame



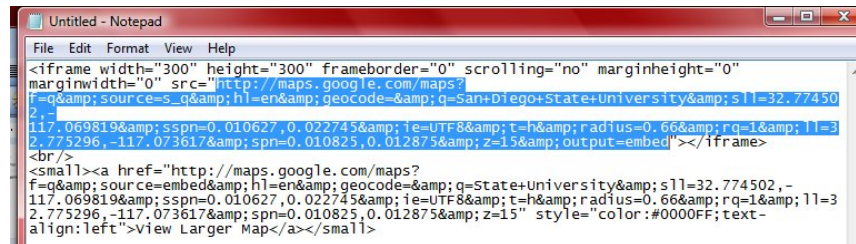
Select iFrame Embed from the results → Click Add



9. The next step is to fill out the gadget parameters form.

- You can select a Title and Title Link that make sense for your map.
- The URL is the address for the webpage that you want to display in the iFrame (the URL needed for embedding a Google Map link can be found within the Google Map Link code).

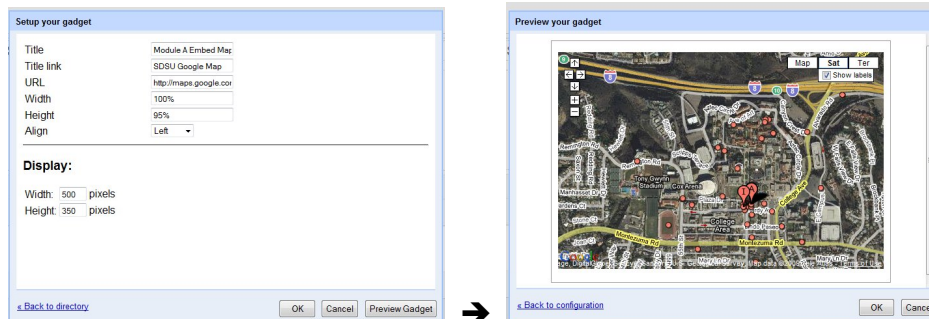
For embedding a Google Map Link copy the text in quotes following src, from the HTML link:



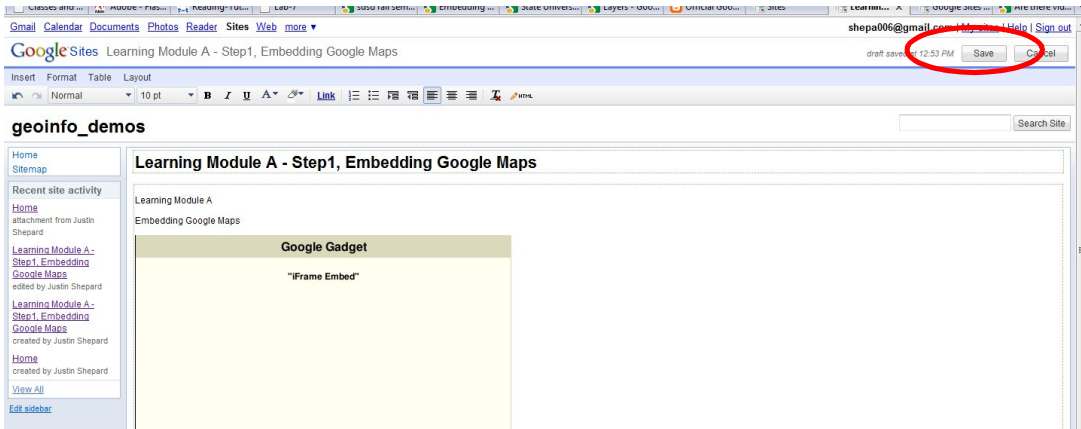
```
<iframe width="300" height="300" frameborder="0" scrolling="no" marginheight="0"
marginwidth="0" src="http://maps.google.com/maps?
f=q&source=s_q&hl=en&geocode=&q=San+Diego+State+University&sl=32.77450
2,-
117.069819&ssp=0.010627,0.022745&ie=UTF8&t=h&radius=0.66&rq=1&ll=3
2.775296,-117.073617&spn=0.010825,0.012875&z=15&output=embed"></iframe>
<br/>
<small><a href="http://maps.google.com/maps?
f=q&source=embed&hl=en&geocode=&q=State+University&sl=32.774502,-
117.069819&ssp=0.010627,0.022745&ie=UTF8&t=h&radius=0.66&rq=1&ll=3
2.775296,-117.073617&spn=0.010825,0.012875&z=15" style="color:#0000FF;text-
align:left">view Larger Map</a></small>
```

Paste this text into the gadget URL.

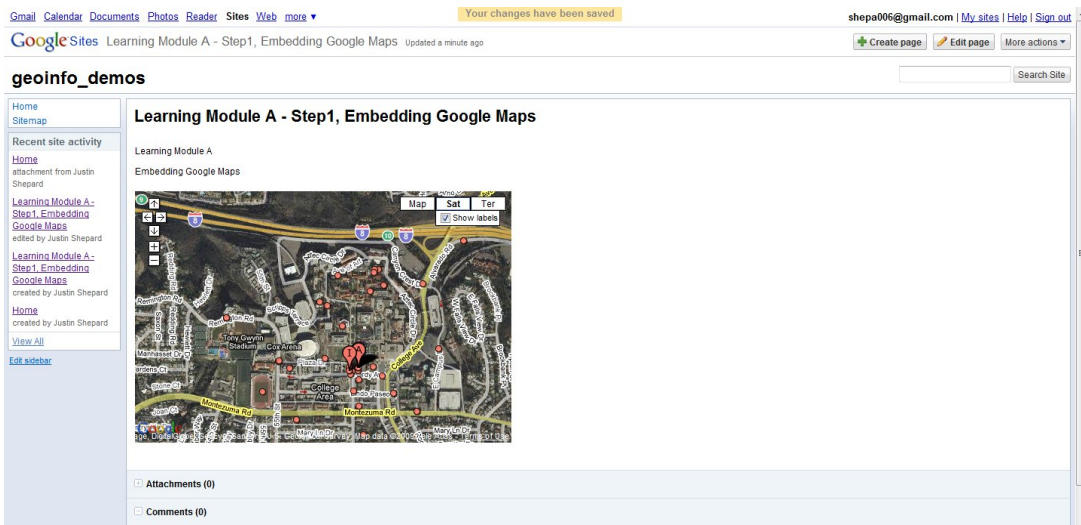
- **Try different values for Width, Height, Align, Display Width, and Display Height**
- Click Preview Gadget
- If you are happy with the look of the display click OK, otherwise click ‘Back to Configuration’ and adjust the settings.



10. Save the changes



11. Updated page will be displayed



Part 1 – Questions

Q1.1 What point of interest or address did you use for your map?

Q1.2 What is the Latitude and the Longitude for your point of interest? Hint: in the Google Map link it is in the code sll= ____, ____.

Q1.3 What is the URL for your Google Sites page?

Part 2 – Displaying Custom Information with Google Maps

Maps

This section will introduce displaying custom features using Google Maps, Google Sites, and KMZ files. A KMZ file can be created using programs such as Google Earth or ESRI ArcMap.

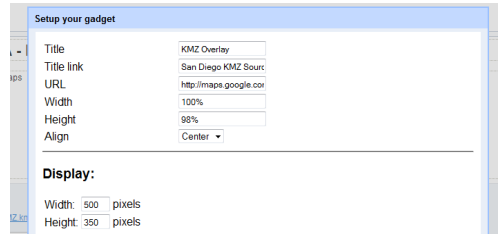
2.1 Overlaying a KMZ on top of a Google Maps base map.

1. Login to your Google Pages account if you aren't already. Create a new page in your Google Sites account and name it ModuleA_P2_FirstInitialLastName (ex. ModuleA_P2_JShepard).
2. Edit the new page so that it matches the example below (replacing Your Name and Date Created with the appropriate text).

If you have a KMZ file to use you can add it to your page under the 'Attachments' section. Otherwise you can use the KMZ provided at http://sites.google.com/site/geoinfodemos/Home/learningModuleA_SanDiegoKMZ.kmz

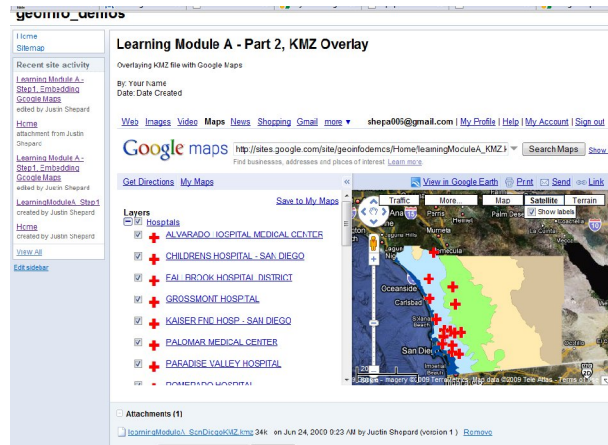


3. Add a Google Maps iFrame, see section 1.2 for help. For the iFrame URL enter http://maps.google.com/maps?f=q&hl=en&q=KMZ_URL (replace 'KMZ_URL' with the URL for your KMZ file). The following example uses the KMZ_URL of http://sites.google.com/site/geoinfodemos/Home/learningModuleA_KMZ.kmz which means the URL to enter into the iFrame gadget is http://maps.google.com/maps?f=q&hl=en&q=http://sites.google.com/site/geoinfodemos/Home/learningModuleA_KMZ.kmz



4. Preview the gadget and make any adjustments that you see fit. Then click OK and Save your changes.

Example of updated KMZ Overlay page.

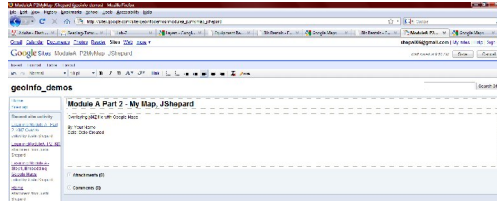


2.2 Adding Custom Features to Google Maps

1. Login to your Google Sites account if you aren't already. Create a new page in your Google Sites account and name it ModuleA_P2MyMap_FirstInitialLastName (ex. ModuleA_P2MyMap_JShepard).

Edit page text to match the example below (replacing Your Name and Date Created with the appropriate text).

Save Changes.



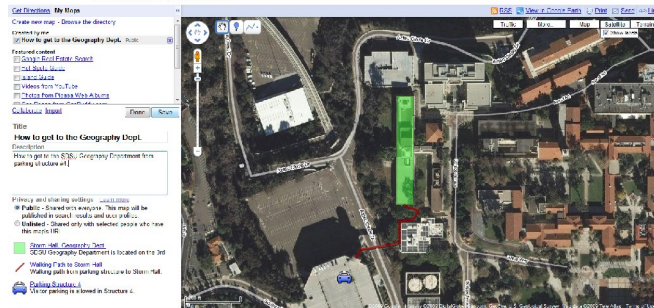
2. Keep your current browser window open and open a new browser window.
3. Watch the following video(s):
<http://maps.google.com/help/maps/mymaps/create.html> for instructions on how to create a My Map and add features.

*** You should still be logged in to your Google account, but if you aren't make sure to log in so that you can create a My Map

4. Create a 'New Map'; you will need to **include at least one polygon, one marker, and one line.**

Example of *creating* 'My Map' showing where Storm Hall is on the SDSU campus and how to get to it from parking structure #4.

*** Make sure to click 'Save' when you are done.



5. Embed your My Map in your My Map web page; see Part 1 for help with embedding a Google Map. Instead of using the HTML link for embedding this map we are going to use the text from the 'Paste link in **email or IM**' field.

Part 2 – Questions

Q2.1 What is the path for the KMZ file that you used? Describe what the KMZ is showing and list at least 3 features and their descriptions.

Q 2.2 List at least programs that can be used for creating a KMZ file.

Q 2.3 What are three types of features that can be created using Google My Maps?

Q2.4 Explain one example of how you could use My Maps to share information with other people?

Part 3 – Embedding and Customizing Google Maps

Using the Google Maps API

The Google Maps API lets you embed Google Maps in your own web pages with JavaScript. The API provides a number of utilities for manipulating maps (just like on the <http://maps.google.com> web page) and adding content to the map through a variety of services, allowing you to create robust maps applications on your website.

3.1 Embedding Simple Viewer and Driving Directions Maps

1. **Because Google Sites does not allow full HTML editing of the header we will be creating files to open on your desktop or upload to your personal website.**
2. Read the following pages (Map Basics) for an introduction to using the Google Maps API:
<http://code.google.com/apis/maps/documentation/v3/>
3. Use Notepad to create a new HTML document (Google-API.htm) with the following contents:

```

<html>
<head>
<meta name="viewport" content="initial-scale=1.0,
user-scalable=no" />
<script type="text/javascript"
src="http://maps.google.com/maps/api/js?sensor=false
"></script>
<script type="text/javascript">
function initialize() {
var latlng = new google.maps.LatLng(-34.397,
150.644);
var myOptions = {
zoom: 8,
center: latlng,
mapTypeId: google.maps.MapTypeId.ROADMAP
};
var map = new
google.maps.Map(document.getElementById("map_canvas"
), myOptions);
}
</script>
</head>
<body onload="initialize()">
<div id="map_canvas" style="width:100%;
height:100%"></div>
</body>
</html>

```

4. Save the file as a HTML (.htm) file, File > Save As > change Save as Type to All Files > name your document ending in .htm (ex. PageName.htm) > Save
5. Next we will be creating a map for driving directions. To do this first you need to open a new HTML document using Notepad (or another simple text editor) and paste the following code (pages 17-22) into the blank document (the code can also be copied from http://sites.google.com/site/geoinfodemos/modulea_p3api_jshepard):


```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"
>
<html
xmlns="http://www.w3.org/1999/xhtml" xmlns:v="urn:sche
mas-microsoft-com:vml">
    <head>
        <meta http-equiv="content-type" content="text/html;
charset=utf-8"/>
        <title>Learning Module A - Driving Directions
Demo</title>
        <script
src="http://maps.google.com/?file=api&v=3.x"
        type="text/javascript"></script>
        <style type="text/css">
            body {
                font-family: Verdana, Arial, sans serif;
                font-size: 11px;
                margin: 2px;
            }
            table.directions th {
background-color:#EEEEEE;
            }
            img {
                color: #000000;
            }
        </style>
        <script type="text/javascript">

var map;
var gdir;
var geocoder = null;
var addressMarker;
var trafficInfo = new GTrafficOverlay();
var toggleState = 1;
var waypoints=[], gpolys=[], routeNodes=[], myNode,
markerDragged, isDragged;
var mapControl = new GMapTypeControl();
var topRight = new
GControlPosition(G_ANCHOR_TOP_RIGHT, new GSize(10,10));
var bottomRight = new
GControlPosition(G_ANCHOR_BOTTOM_RIGHT, new
GSize(10,10));

```

```

function toggleTraffic() {
    if (toggleState == 1) {
        map.removeOverlay(trafficInfo);
        toggleState = 0;
        TrafficToggle.value = "Show traffic";
    } else {
        map.addOverlay(trafficInfo);
        toggleState = 1;
        TrafficToggle.value = "Hide traffic";
    }
    //modifyDirections("en_US", "San Francisco", "Los
Angeles");
}

function initialize() {
    //alert("Welcome... initializing!");
    if (GBrowserIsCompatible()) {
        map = new
GMap2(document.getElementById("map_canvas"));
        gdir = new GDirections(map,
document.getElementById("directions"));
        GEvent.addListener(gdir, "load",
onGDirectionsLoad);
        GEvent.addListener(gdir, "error", handleErrors);
        setDirections("en_US", "San Francisco", "San
Diego");

        map.enableDoubleClickZoom();
        map.enableScrollWheelZoom();

        map.enableContinuousZoom();

        map.addControl(new GMapTypeControl());
        map.addControl(new GLargeMapControl());
        map.addControl(new GScaleControl());
        //alert("Welcome... initializing!");
        map.addOverlay(trafficInfo);
    }
    else {
        alert("Sorry, the Google Maps API is not compatible
with this browser");
    }
}

```

```

function setDirections(locale, fromAddress, toAddress)
{
    gdir.load("from: " + fromAddress + " to: " + toAddress,
        { "locale": locale });
}

function modifyDir(locale, fromAddress, toAddress) {
    // gdir.load("from: " + fromAddress + " to: " +
toAddress,
    //          { "locale": locale });
}

function handleErrors(){
    if (gdir.getStatus().code == G_GEO_UNKNOWN_ADDRESS)
        alert("No corresponding geographic location
could be found for one of the specified addresses. This may
be due to the fact that the address is relatively new, or
it may be incorrect.\nError code: " +
gdir.getStatus().code);
    else if (gdir.getStatus().code ==
G_GEO_SERVER_ERROR)
        alert("A geocoding or directions request could not
be successfully processed, yet the exact reason for the
failure is not known.\n Error code: " +
gdir.getStatus().code);

    else if (gdir.getStatus().code ==
G_GEO_MISSING_QUERY)
        alert("The HTTP q parameter was either missing or
had no value. For geocoder requests, this means that an empty
address was specified as input. For directions requests,
this means that no query was specified in the input.\n Error
code: " + gdir.getStatus().code);

    // else if (gdir.getStatus().code ==
G_UNAVAILABLE_ADDRESS) <--- Doc bug... this is either not
defined, or Doc is wrong
    // alert("The geocode for the given address or the
route for the given directions query cannot be returned due
to legal or contractual reasons.\n Error code: " +
gdir.getStatus().code);

    else if (gdir.getStatus().code == G_GEO_BAD_KEY)
        alert("The given key is either invalid or does not

```

```

match the domain for which it was given. \n Error code: "
+ gdir.getStatus().code);

        else if (gdir.getStatus().code ==
G_GEO_BAD_REQUEST)
            alert("A directions request could not be
successfully parsed.\n Error code: " +
gdir.getStatus().code);

            else alert("An unknown error occurred.");

    }

    function onGDirectionsLoad(){
        // Use this function to access information about the
latest load()
        // results.

        // e.g.
        // document.getElementById("getStatus").innerHTML
= gdir.getStatus().code;
        // and yada yada yada...
    }

</script>

</head>
<body onload="initialize()" onunload="GUnload()">

    <h2>Welcome to the Learning Module A Custom Google Map
using Google API.<br></h2>
    This demo is made available from Learning Module A on the
<a href="http://geoinfo.sdsu.edu/hightech"
target="_blank">GIS Career & Education Awareness</a>
website.

    The purpose of this demo is to illustrate of how various
<a
href="http://code.google.com/apis/maps/documentation/"
target="_blank">Google API</a> features can be combined to
create custom map displays and user interaction.
    <br>
    <h2>Driving Directions with Selectable Language

```

```

Feature</h2>
  <form action="#"
onsubmit="setDirections(this.locale.value,
this.from.value, this.to.value); return false">

  <table>

    <tr><th align="right">From:&nbsp;  </th>

    <td><input type="text" size="25" id="fromAddress"
name="from"
  value="San Francisco"/></td>
    <th align="right">&nbsp;  &nbsp;   To:&nbsp;  </th>
    <td align="right"><input type="text" size="25"
id="toAddress" name="to"
  value="San Diego" /></td></tr>

    <tr><th>Language:&nbsp;  </th>
    <td colspan="3"><select id="locale" name="locale">

      <option value="en" selected>English</option>

      <option value="fr">French</option>

      <option value="de">German</option>
      <option value="ja">Japanese</option>
      <option value="es">Spanish</option>
    </select>

    <input name="submit" type="submit" value="Get
Directions!" />

  </td></tr>
</table>

</form>

  <br/>
  <table class="directions" width="950">
  <tr><th>Formatted Directions</th><th>Map</th></tr>

  <tr>
  <td valign="top"><div id="directions" style="width:

```

```
350px"></div></td>
    <td valign="top"><div id="map_canvas" style="width:
600px; height: 400px"></div><br /><input type="button"
value="Toggle Traffic" id="TrafficToggle"
onClick="toggleTraffic();" /></td>

    </tr>
    </table>
</body>
</html>
```

6. Save the file as a HTML (.htm) file, File > Save As > change Save as Type to All Files > name your document ending in .htm (ex. DrivingDir.htm) > Save.

Part 3 – Questions

Q3.1 What's the major language used for Google Map API and related applications? What's the advantage of this language?

Q3.2 How many "Services" can Google Map API provided? (see the Service document:

<http://code.google.com/apis/maps/documentation/services.html>). Briefly introduce each **service** function in Google Map API.

Q3.3 List three of the language options for the driving directions demo.

Q3.4 What is the map center set to in the map viewer demo? Hint: `google.maps.LatLng`

Q3.5 Include a screenshot of your map (either the map viewer or driving directions version).

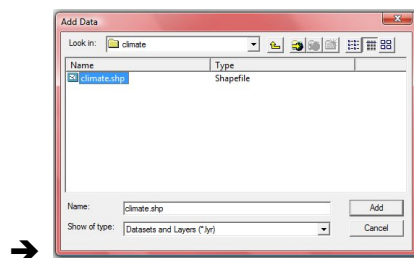
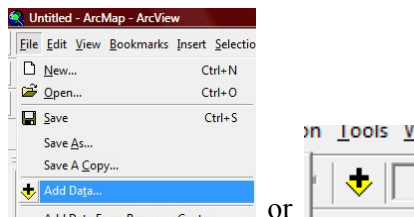
Part 4, Optional – Creating a KMZ File Using ESRI

ArcMap 9.3

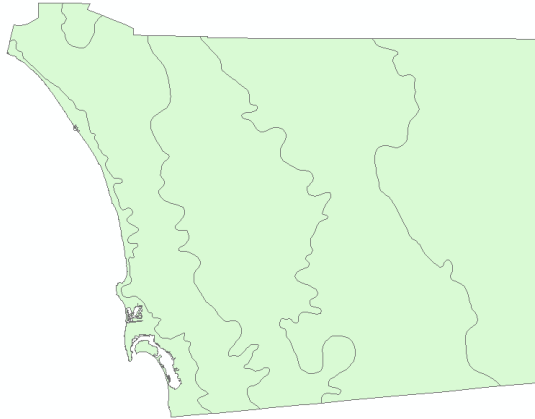
KML is a file format used to display geographic data in an Earth browser such as Google Earth, Google Maps, and Google Maps for mobile. KML uses a tag-based structure with nested elements and attributes and is based on the XML standard (http://code.google.com/apis/kml/documentation/kml_tut.html, 2009). A KMZ file is a compressed KML file that is also capable of storing 3D models and associated images.

4.1 Cartographic design in ArcMap

1. Open ArcMap
2. Add two feature classes (you can use your own datasets or the climate and hospital datasets available at <http://sites.google.com/site/geoinfodemos/part-4-creating-a-kmz-file-using-esri>).



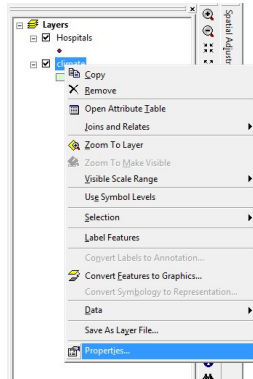
Click Add



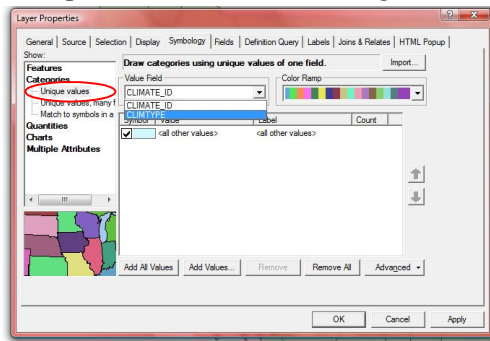
→ The feature class will be added to the display with default symbology, we'll change this in a moment.

3. Repeat step 2 to add your second feature class.
4. Change the symbology for the feature classes that you added. You can select a symbology scheme that makes sense for your data. Below is an example of changing the symbology for the climate dataset.

Right click the layer and open the properties window:

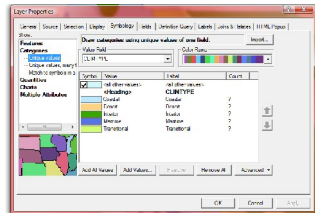


Go to the 'Symbology' tab from the 'Categories' group on the left select 'Unique Values' and then change the 'Value Field' to an appropriate attribute.



Click 'Add All Values' and change the color for each symbol to a color scheme

that makes sense for your data. You can double-click each of the rectangles to change just that color or make sure none of the categories are selected and change the 'Color Ramp' to change all of the category colors.



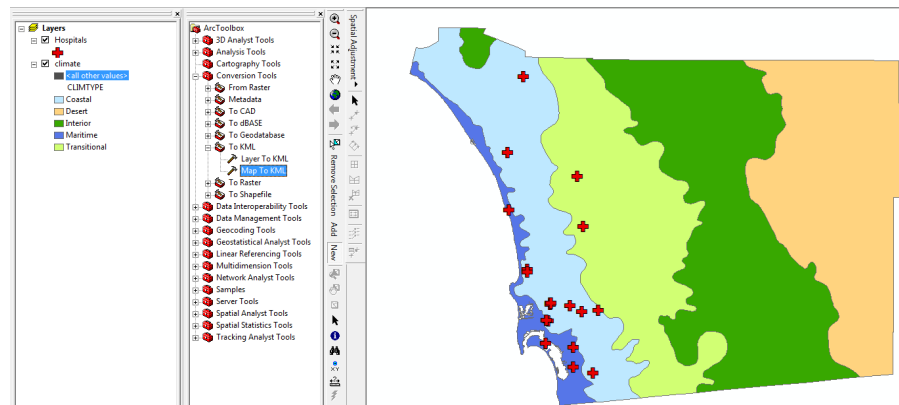
Try different settings to determine the setting that works best for your data.

5. Save your map.

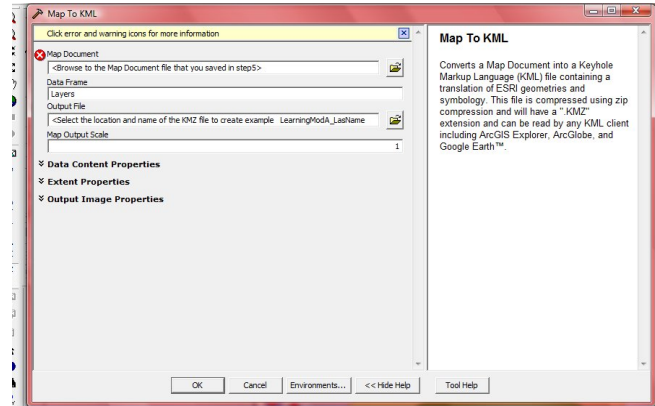
4.2 Exporting KMZ file



1. Open the ArcToolBox (the red box icon)
2. Click on the "Conversion Tools" => Select [To KML].=> [Map to KML]



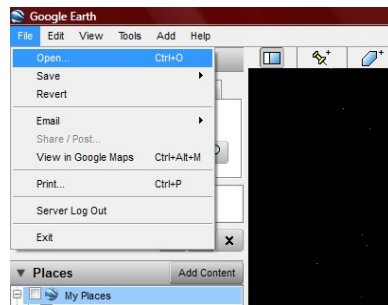
3. In the Map Document window, link to map document you created in 4.1 step 5. The 'Data Frame' field will be updated to a drop down list (unless you added data frames or changed the default data frame in 4.1 you can use the default Data Frame name 'Layers', which should be automatically filled in).



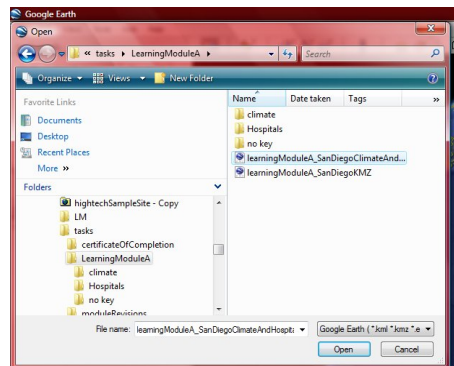
4. Click OK. The KMZ file will be created in the Output File location.

4.3 Previewing KMZ file in Google Earth

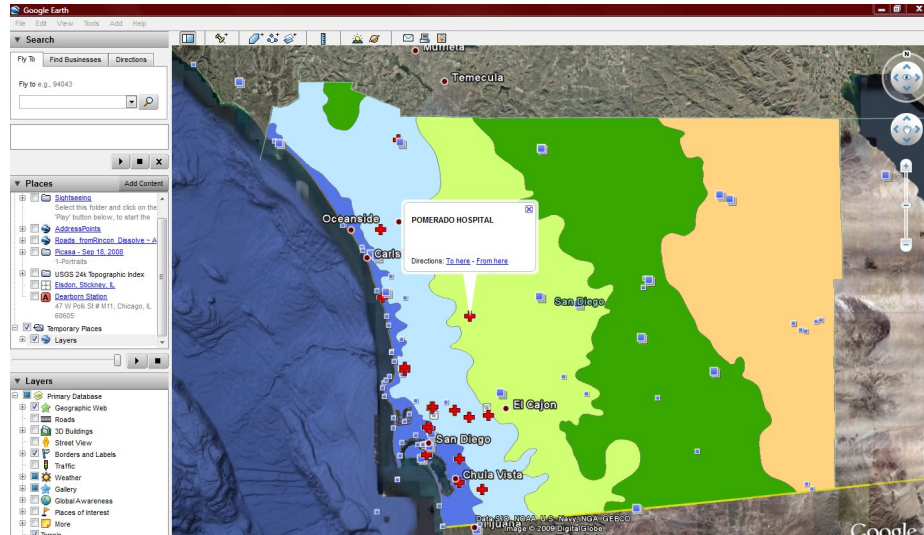
1. To preview the KMZ file you created in section 4.2 first open Google Earth, if you don't have Google Earth it can be downloaded from <http://earth.google.com/>.
2. Open the KMZ created in section 4.2. File → Open



3. Select your KMZ and click Open



- The symbology that you set in section 4.1 is used for displaying the KMZ. In addition clicking on any of the points should open an information window (in our example this includes the name of the hospital).



- The KMZ can be shared with others by emailing it or posting it on a website. For an example of posting the KMZ go to <http://maps.google.com/?q=http://geoinfo.sdsu.edu/583student/g583-01/lab-7-tsou.kmz>

Part 4 - Questions

Now that you have seen and worked with examples of digital globes, computerized maps, and web-based maps we would like you to write a 250-word paragraph (approximately one typed page) discussing your experience and one situation that you could see using this technology to help with. You can incorporate your answers to previous questions but also make sure to explain how you would apply the technology in detail. Some questions to help you structure your answer are:

1. In your paragraph, quickly describe the GIS applications that you used (ArcMap, Google Earth, and Google Maps API).
2. In your paragraph, also answer if you think that the GIS applications can help people? In what ways can they help people? In what kinds of occupations/industries would they be useful?
3. What was your favorite part, or favorite use, of working with the Google Maps API?

Part 5, Optional – JavaScript Programming

In Part 3 you had the opportunity to copy a customized map display for driving directions, which used JavaScript. Now you will have a chance to create your own map using JavaScript.

Introduce JavaScript and Google JavaScript API

5.1 JavaScript Basics

1. JavaScript is a scripting language that can be used to give your websites, and web-based maps, more functionality. Read through the JavaScript Introduction (http://www.w3schools.com/jS/js_intro.asp), ‘How To’ (http://www.w3schools.com/jS/js_howto.asp), ‘Where To’ (http://www.w3schools.com/jS/js_where.asp), and ‘Statement’ (http://www.w3schools.com/jS/js_statements.asp) sections to see what JavaScript can do.
2. Use the ‘Try it Yourself’ tool shown at the end of the ‘Statements’ section, http://www.w3schools.com/jS/tryit.asp?filename=tryjs_blocks. Change the text in the left panel from “This is another paragraph.” To “This was created by YourName on Today’s Date”.
3. Try running your new script by clicking the ‘Edit and Click Me’ button
4. Copy your new code from the left panel and paste it into Word or NotePad.

5.2 Variables

JavaScript variables are used to hold values or expressions which can then be used later in your code. This can be very useful when you are creating your map!

1. Open the JavaScript Try It Yourself window, http://www.w3schools.com/jS/tryit.asp?filename=tryjs_variable.
2. Before we modify the code in the left panel let’s take a look at what it is telling the computer to do.

```
<html>
<body>

<script type="text/javascript">
var firstname;
firstname="Hege";
document.write(firstname);
document.write("<br />");
firstname="Tove";
document.write(firstname);
</script>

<p>The script above declares a variable,
assigns a value to it, displays the value, changes
the value,
and displays the value again.</p>

</body>
</html>
```

The main `<html>`, `<body>`, and `<p>` tags are all standard HTML tags. JavaScript is used within the `<script>` tag (that is everything between `<script>` and `</script>` will be run using JavaScript).

```
4. <script type="text/javascript">
5. var firstname;
6. firstname="Hege";
7. document.write(firstname);
8. document.write("<br />");
9. firstname="Tove";
10. document.write(firstname);
11. </script>
```

The first thing that we need to do is tell the computer what scripting language we will be using, JavaScript is the most widely used but it is not the only one. To do this we enter `type="text/javascript"`.

In line 5 we create a variable called `firstname`, this variable will be used to store names and can be reused by assigning a new name to the variable. **Don't forget to put a semicolon (;) at the end of each of your lines in JavaScript.** The next line assigns a value to our variable, in this case it assigns Hege.

Line 7 tells the browser to write the name that you have stored in the `firstname` variable. Followed by this, line 8, instructs the browser to start a new line. This is done by writing an HTML tag `
`.

Line 9 replaces Hege with Tove and then line 10 instructs the browser to write the value for the variable `firstname`.

3. Change line 6 so that the `firstname` variable uses the first name of your best friend.
4. Change line 9 so that the `firstname` variable uses the your first name.
5. Run your new code by clicking the 'Edit and Click Me' button.

5.3 Functions

So far we have told the browser what to do by writing code in sequential steps. However, what if we wanted to do things out of order or reuse a piece of code multiple times (since we don't want to create more work for ourselves). To accomplish this we need to use functions, which are sections of code that can be run at any time once the page has fully loaded.

1. Read about JavaScript functions on the Functions page, http://www.w3schools.com/js/js_functions.asp.
2. Now lets take a look at an example, http://www.w3schools.com/jS/tryit.asp?filename=tryjs_function1.

In this example there is a `<head>` (lines 2 – 9) and a `<body>` (lines 11 – 20) section, recall from the 'Where To' reading that "scripts to be executed when they are called, or when an event is triggered, go in the head section" and "scripts to be executed when the page loads go in the body section."

```
1. <html>
2. <head>
3. <script type="text/javascript">
4. function displaymessage ()
5. {
6. alert ("Hello World!");
7. }
8. </script>
9. </head>
10.
```

```
11. <body>
12. <form>
13. <input type="button" value="Click me!"
14. onclick="displaymessage()" />
15. </form>
16.
17. <p>By pressing the button above, a function will be
18. called. The function will alert a message.</p>
19.
20. </body>
21. </html>
```

3. In this example the page loads without displaying the “Hello World!” message. Instead, the page is told to wait until the user clicks the button.
4. The function called `displaymessage()` is a custom function what we created for this code. The commands between the curly braces `{}` will be executed every time this function runs. In our example it will display an alert window that says “Hello World!”.
5. To make the `displaymessage()` function run we need to have an event tell the browser to run the function. An event could be something such as hovering over a button, clicking a button, panning around a map, clicking a map, etc. For our example the event is clicking on the ‘Click me!’ button.
6. Change the `displaymessage()` function so that it displays an alert message that says “Hello World, this is YourName!”
7. Try running your new code

5.4 JavaScript in Google Maps

1. For an overview of using Google Maps API and how JavaScript is used to call functions and store variables read the Google Maps Basics introduction, <http://code.google.com/apis/maps/documentation/introduction.html>.
2. We are going to focus on the JavaScript function so go to http://code.google.com/apis/ajax/playground/?exp=maps#map_event_arguments. This page is similar to the JavaScript Try It Yourself tool that you used in 5.3, but it is specifically for Google Maps. In this example the creation of the basic map is handled for us. We will just be focusing on modifying the function that creates the map.

3. Line 1 is creating a custom function called initialize. This step is similar to the “Hello, World” of Google Maps. This function will be run when the webpage loads.
4. Line 2 checks to make sure that the browser is compatible. If the browser is compatible then the code between the curly braces will be executed. If the browser is NOT compatible then nothing will happen.
5. Line 3 and 4 create a variable called *map* and set the center of the map to 37.4419, -122.1419.
6. Line 6 tells the browser to run a function every time that the *map* is “clicked”

The function that will execute is described in lines 7 – 10. Whenever the map is clicked an infowindow will be opened and display the coordinates for the location you clicked and your current zoom level.

7. Lines 12 and 13 add navigation and display controls to your map
8. Try cutting lines 12 and 13 to see what happens if you forget to add controls to your map. Cut lines 12 and 13, then click Run.

***** If you need to return to the default code you can click on ‘Map Event Arguments’ in the Event Examples tree *****

9. Paste the map controls back into your function.
10. The setCenter command has two arguments, or settings that you can assign. These arguments are map.setCenter(startingCoordinates, zoomLevel). Change the starting coordinates to the coordinates for the city you are in. For Eureka California this becomes
GLatLng(37.4419, -122.1419) → GLatLng(40.7662, -124.1645)
11. Change the starting zoom level (any integer between 1 and 15).
map.setCenter(new GLatLng(40.7662, -124.1645), 10);
12. Click Run to see your new map.
13. Try modifying the code on your own and see what you can come up with.
14. Copy your new function and paste it into Word or NotePad

Part 5 – Questions

1. What is the code you wrote for section 5.1? You copied this to Word or Notepad in step 4.
2. What is the code you wrote for section 5.4? You copied this to Word or Notepad in step 14.
3. What is the fundamental element in any Google Maps API application?
4. Is JavaScript the same as Java?
5. Explain two things that JavaScript can do, according to the W3Schools.
6. What are two common map types from the Google Maps API?

Student Survey Learning Module A

This learning module was developed specifically for you---the students---in order to expose you to GIS and related geospatial technologies.

GIS and related geospatial technologies, similar to what you worked with in this learning module, are major components of one of the most important new industries for career growth. Businesses, organizations, and governments are increasingly seeking skilled talent in this industry.

Now, to benefit other students and the authors of this learning module, you are required to finish this Student Survey and please send it back to the San Diego State University NSF-ATE team. We would greatly appreciate your input. Please take time to answer these questions as openly and honestly as possible. **They will not affect your grade or score.** You can provide your feedback to your teacher/educator along with the learning module answers on the separate answer sheet.

Thank you for helping us improve our modules!

- S-1:** What did you like best about this learning module A? Why?
- S-2:** What did you like least about this learning module A? Why?
- S-3:** Was there anything that was really hard to understand or difficult to complete in this module A? What was it? Why?
- S-4:** Is there anything in this learning module A that we can add, exclude, or change to make it better? What is it? Why?