

Geography 485L/585L - Internet Mapping

Karl Benedict - kbene@unm.edu

Spring 2016

Contents

1 Week 1 - Module 1 - Introduction and Outline	7
Overview	7
Introductions	7
Syllabus Review (link)	7
Class Topics	7
Basics	8
Outline	8
What is Internet Mapping	8
Definitions	8
Definitions	8
Definitions	8
Definitions	9
Tools	9
Computer Hardware Requirements	9
Software Requirements	9
2 Week 2 - Module 2a - Web-based Mapping Clients. HTML, CSS & Javascript	11
Overview	11
Web Development	11
Parts of a Web Page	11
Web Site Components - Structure	12
Web Site Components - Presentation	12
CSS Selectors	13
Web Site Components - Behavior	13
Reference Links	13
Simple Web Page	14
Simple Web Page with CSS	14
Simple Web Page with Javascript	14
More Complete Web Page Example	15

3 Week 3- Module 2a - Web-based Mapping Clients. Google Maps API	17
Outline	17
What is an API	17
Google Maps API Version	17
Reference Information	18
Key Components	18
Controls	18
Overlays	18
Overlays (continued)	18
Services	19
Services	19
Events	19
Examples	20
Simple - Roadmap	20
Simple - Roadmap Code	20
Simple - Satellite	22
Simple - Satellite Code	22
Simple - Hybrid	23
Simple - Hybrid Code	23
Simple - Terrain	25
Simple - Terrain Code	25
Simple - Hybrid - Zoomed	26
Simple - Hybrid - Zoomed Code	26
Simple - Zoomed - Modified Controls	28
Simple - Zoomed - Modified Controls Code	28
Markers	29
Markers Code	30
Polyline	31
Polyline Code	31
Polygon	33
Polygon Code	33
Adding an Info Window	35
Adding an Info Window Code	35

CONTENTS	5
----------	---

4 Week 4 - Module 2a - Web-based Mapping Clients. Google Maps API (pt. 2)	39
Overview	39
<i>Getting Started with Styled Maps</i> - Video	39
Map Example: Simple - Styled	39
<i>Google I/O 2011: Managing and visualizing your geospatial data with Fusion Tables</i> - Video .	42
Bringing It All Together	42

Chapter 1

Week 1 - Module 1 - Introduction and Outline

Overview

- Introductions
- Review of the Syllabus
- Topics to be Covered
- Basics/Definitions

Introductions

- Who am I?
- Who are you?
- What brought you here?

Syllabus Review ([link](#))

Class Topics

- Internet Mapping Clients: Basic HTML, Javascript, CSS; Google Maps API; OpenLayers javascript library
- Geospatial Services Oriented Architectures (SOA)
- Open Standards: Open Geospatial Consortium (OGC - [WMS](#), [WFS](#), [WCS](#), [KML](#)); Extensible Markup Language ([XML](#))
- Desktop client use of Open Standards
- Data sharing/publication using Open Standards

Basics

Outline

- What is Internet Mapping?
- Definitions
- Tools

What is Internet Mapping

Extended Desktop Mapping Use of open standards based remote data and map services in desktop applications

Geospatial Data Sharing Establishing open standards based services to share geospatial data and mapping capabilities over the Internet

Web-client Mapping The delivery of mapping and geospatial data tools through web browsers, again based upon open standards

Definitions

Internet The global computer network of computers that typically connect with each other over TCP/IP

World Wide Web The subset of applications that are run over the Internet, typically using the HTTP protocol in combination with data (HTML, XML, XHTML), presentation (CSS), and behavior (JavaScript) components

Mapping The generation of cartographic products that include map images (pictures of geospatial data) and other elements (e.g. legends, tools, scale information, north-arrow)

Definitions

Analysis The development of models (statistical and otherwise) that enable the exploration of geospatial data and testing of hypotheses using those data

Open Standards While the definition varies from one organization to the next, Open Standards are often characterized by the following:

- Developed through a public process by a national or international standards group
- May be implemented royalty-free

Definitions

Interoperability Ability of systems to share data and information with each other

COTS Commercial Off-the-Shelf Software. Applications that are “purchased” from vendors, often with license terms that restrict the use of the software to the specific platform for which it is licensed. Often comes with implicit or explicit technical support

Open Source Software licensed under terms that are consistent with the Open Source definition, which includes access to source code, and freedom to modify and redistribute

Definitions

Data Actual values associated with geographic locations. For example - numeric elevation values associated with locations within a Digital Elevation Model.

Metadata Data about a particular data product or service. Metadata provide critical documentation that supports the discovery and use of data products and data and mapping services

Tools

Computer Hardware Requirements

- At least 2 GB RAM
- At least 20 GB of available disk space
- Internet Connection (broadband [>728 Kb/sec] recommended)

Software Requirements

- Supported Operating System
- Geographic Information System (GIS)
- Text Editor
- Secure File Transfer Protocol Client
- Secure Shell (SSH) Client
- Web Browser (at least one of the following)
- A desktop Git/GitHub client for your operating system of choice

Chapter 2

Week 2 - Module 2a - Web-based Mapping Clients. HTML, CSS & Javascript

Overview

- Web Development
- Parts of a web page
- Web Site Components
 - Structure (X/HTML)
 - Presentation (CSS)
 - Behavior (Javascript)
- Simple Web Pages
- More Complete Web Page Example

Web Development

- Requirements
 - Web Server
 - File location that the web server accesses for requested content
 - Files must be readable by all users
- General Process
 - Create basic content in HTML or XHTML (structure)
 - Change appearance of content through the definitions of styles using CSS (presentation)
 - Add dynamic capabilities to content through Javascript (behavior)
 - REPEAT over and over and over and over again

Parts of a Web Page

```
1  <html>
2      <!-- The HTML block is the container for all of your page content -->
3      <head>
```

```

4      <!-- The head is where you include pointers to external resources
5      (i.e. style sheets and javascript files), blocks of Javascript code
6      , styles, etc. -->
7      <title>The page title also goes in here</title>
8  </head>
9  <body>
10     <!-- The body is where you put all of the content for the page
11     (i.e. the material that will be displayed in the web browser) -->
12     <h1>Headers</h1>
13     <div>Generic blocks of content</div>
14     <p>Paragraphs</p>
15     <table>Tables</table>
16     <img ...>Images</img>
17     <form ...>Forms</form>
18     <ul>Unordered Lists</ul>
19     <ol>Ordered Lists</ol>
20     <li>List Items</li>
21
22     <!-- Javascript can go here as well -->
23   </body>
24 </html>

```

[Link to example/Preview](#)

Web Site Components - Structure

Content is defined in terms of the structural elements available in HTML/XHTML

- Sample HTML/XHTML Tags
 - Paragraphs (i.e. blocks of text) are contained within `<p>...</p>` tags
 - Headings (i.e. section headings, sub-headings) are contained within numerically defined header tags: `<h1>...</h1>`, `<h2>...</h2>`, `<h3>...</h3>`, etc.
 - Tabular data are within `<table>...</table>` tags
 - Lists are specified within `...` or `...` tags, depending upon whether the list is ordered (numbered) or unordered (e.g. bulleted)
 - User input elements are put within `<form>...</form>` tags
 - Blocks of content (i.e. sections or divisions) are defined within `<div>...</div>` tags
- Structure is translated into the Document Object Model (DOM) for later use by CSS and Javascript

Web Site Components - Presentation

Modifications to default rendering of HTML/XHTML elements are made through styles defined in CSS

- Styles may be
 - defined in an external file that is referenced within the `<head>` block (the preferred method when doing “real” web development)
 - directly defined within the `<head>` block of a web page
 - directly embedded in the elements to which they apply (generally not a “Good Thing”)
- When not embedded within an element, a style definition consists of

- A selector
- The style definition, enclosed in “curly-brackets”, separated by “semi-colons”
- For example: `h1 {color:red; font-size:18px;}`

CSS Selectors

Selectors may be based on several criteria

- Element name: `h1, p, table, ul`, etc.
 - Element: `<h1>A top level heading</h1>`
 - Selector: `h1 {color:red; font-size:18px;}`
- Element ID: a unique name assigned to HTML/XHTML elements within the structure of the document
 - Element: `<p id="para01">Some text goes here</p>`
 - Selector: `#para01 {color:blue; font-size:12px;}`
- Class ID: a name assigned to multiple elements which may be modified through reference to their class
 - Element: `<p class="instructions">Here are some instructions</p>`
 - Another Element: `<p class="instructions">Here are some more instructions</p>`
 - Selector: `.instructions {color:red; font-size:12px; text-decoration:blink;}`
- Selectors may be combined in a variety of ways

Web Site Components - Behavior

The most interoperable language for adding dynamic behavior to web sites is *Javascript* - supported by most browsers on most operating systems

- A full-fledged programming language
 - A non-trivial undertaking to become proficient in
 - Experience in other programming languages can contribute to learning Javascript
- Defines actions that may be taken on/by DOM elements
- Allows for modification of existing DOM elements, creation of new DOM elements after the page has finished loading from the server, retrieval of new content after page loads
 - An interactive web page that may behave like a local desktop application

Reference Links

- w3schools.com
 - [HTML 4.0 / XHTML 1.0 Tag Reference](#)
 - [Cascading Style Sheet \(CSS\) selectors and elements](#)
 - [Javascript reference](#)
- World Wide Web Consortium (W3C)
 - [HTML and CSS Background](#)
 - [HTML and CSS Tutorial Links Page](#)
 - [Validators Page](#)
- Webmonkey.com
 - [HTML Cheat Sheet](#)
 - [CSS Guide](#)

Simple Web Page

```

1  <html>
2      <head>
3          <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4          <title>This is a simple web page</title>
5      </head>
6      <body>
7          <h1>They don't get any simpler than this!</h1>
8          <p>OK, not much simpler than this.</p>
9          <p>Hello World?</p>
10     </body>
11 </html>
```

[link to example/Preview](#)

Simple Web Page with CSS

```

1  <html>
2      <head>
3          <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4          <title>This is a simple web page - with styling</title>
5          <style type="text/css">
6              h1 {color:blue; font-size:large}
7              p.para {color:#777777; font-size:small}
8              #annoying {color:red; text-decoration:line-through}
9          </style>
10     </head>
11     <body>
12         <h1>They don't get any simpler than this!</h1>
13         <p class="para">OK, not much simpler than this.</p>
14         <p id="annoying" class="para">Hello World?</p>
15     </body>
16 </html>
```

[link to example/Preview](#)

Simple Web Page with Javascript

```

1  <html>
2      <head>
3          <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4          <title>This is a simple web page with Javascript</title>
5          <script type="text/javascript">
6              function genericAlert() {
7                  alert("You just did something ...")
8                  document.getElementById("clickMe").style.color = "red"
9              }
10         </script>
11     </head>
12     <body>
13         <h1>They don't get any simpler than this!</h1>
```

```

14    <p>OK, not much simpler than this.</p>
15    <p>Hello World?</p>
16    <p id="clickMe" onclick="genericAlert();">What happens when you click me?</p>
17  </body>
18 </html>

```

[link to example/Preview](#)

More Complete Web Page Example

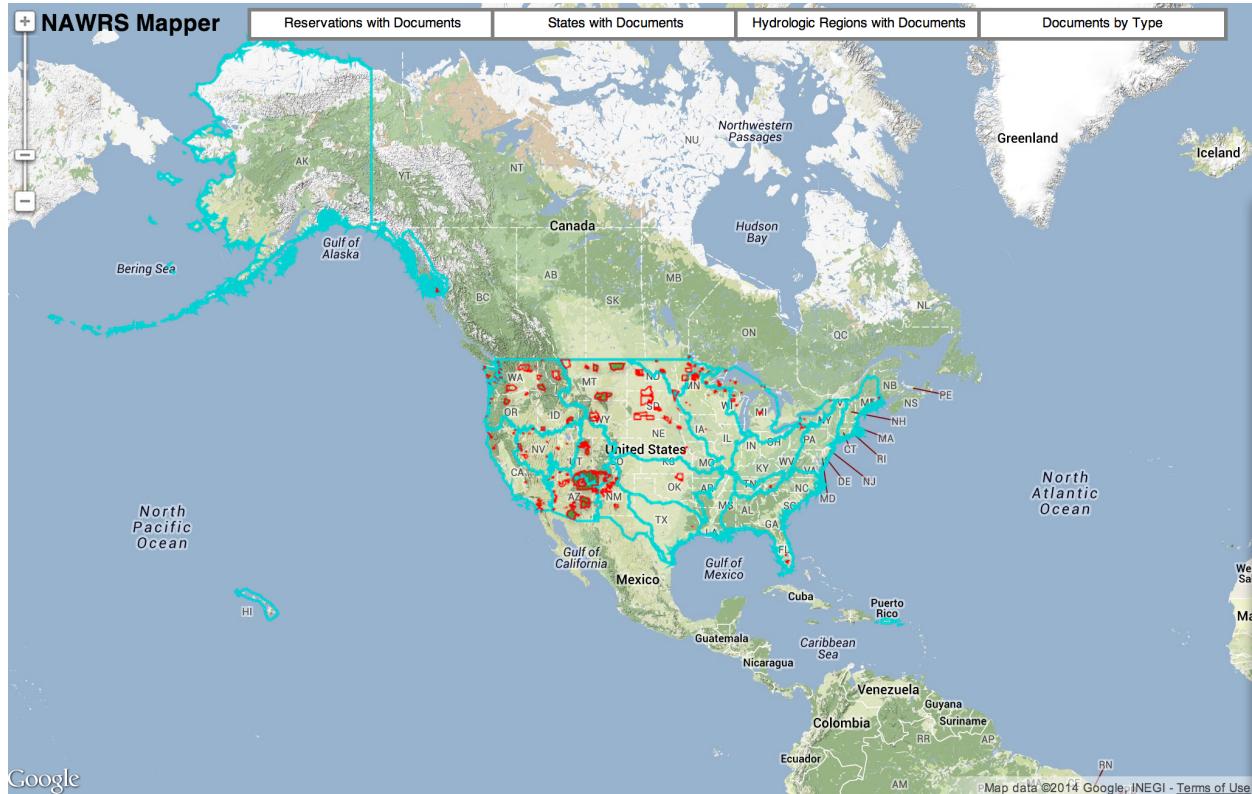


Figure 2.1: [NAWRS Mapper](#). *HTML*: 39 Lines; *CSS*: 136 Lines; *core.js*: 515 Lines + Google Maps API and JQuery Framework

Chapter 3

Week 3- Module 2a - Web-based Mapping Clients. Google Maps API

Outline

- What is an API
- The Google Maps API
 - Version
 - Reference Information
 - Key Components
 - Examples

What is an API

- API Stands for Application Programming Interface

An Application Programming Interface (API) is a particular set of rules and specifications that a software program can follow to access and make use of the services and resources provided by another particular software program that implements that API. It serves as an interface between different software programs and facilitates their interaction, similar to the way the user interface facilitates interaction between humans and computers. – From Wikipedia: <http://en.wikipedia.org/wiki/API>

- The Google Maps API provides an interface for interacting with Google's mapping services from external web applications

Google Maps API Version

- The version of the Google Maps API used in this class is v3 of the Javascript API
 - Freely usable for free applications
 - Subject to Google's Terms of Service
 - Google [API key is optional for our work](#)
- Key capabilities in v3
 - Interactive maps based on Google's mapping engine (contrast w. static maps API)
 - Optimized for desktop and mobile platforms and applications

Reference Information

Google Maps API Family <http://code.google.com/apis/maps/>

Javascript API Home Page <http://code.google.com/apis/maps/documentation/javascript/>

Javascript API v3 Tutorial Page <http://code.google.com/apis/maps/documentation/javascript/tutorial.html>

Key Components

- Map object options

Types (required) ROADMAP

SATELLITE

HYBRID

TERRAIN

Latitude and Longitude (required) specification of where the map should initially be centered

Zoom Level (required) 0=global, higher values increasingly local. Limited by map type

Controls

- Available Controls (enabled through map options) [default controls](#)
 - Zoom Control
 - Pan Control
 - Scale Control
 - MapType Control
 - Street View Control
 - Rotate (for maps that contain 45-degree imagery)
- Different control styles may be defined
- Controls may be positioned [positioning options](#)
- Custom controls may be defined and attached to fixed location in the map

Overlays

Overlay Types [documentation](#)

Marker points depicted by specified or defined icons at locations within the map

Polyline linear features defined by multiple points with a defined style for the line

Polygon closed features defined by multiple points. Supports multi-polygons, and donuts. Line and fill styles may be specified.

(Ground) Overlay Maps Image-based map layers that replace or overlay Google layers - registered to the map coordinates

Overlays (continued)

Info Windows floating content windows for displaying content defined as HTML, a DOM element, or text string

Layers Grouped display content assigned to a specific layer type: Data (including GeoJSON), KmlLayer (& GeoRSS), Heatmap, FusionTablesLayer, TrafficLayer, TransitLayer BicyclingLayer

Custom Overlays definition of programmatically controlled layers

Services

- Geocoding Service
 - Forward and reverse geocoding:
 - * address to LatLon
 - * LatLon to Nearest Address
 - May be biased to current viewport, region
- Directions
 - Based upon an origin, destination, and a variety of additional options
 - Available directions and rendered route
- Distance Matrix
 - Travel distance and duration given a specific mode of travel

Services

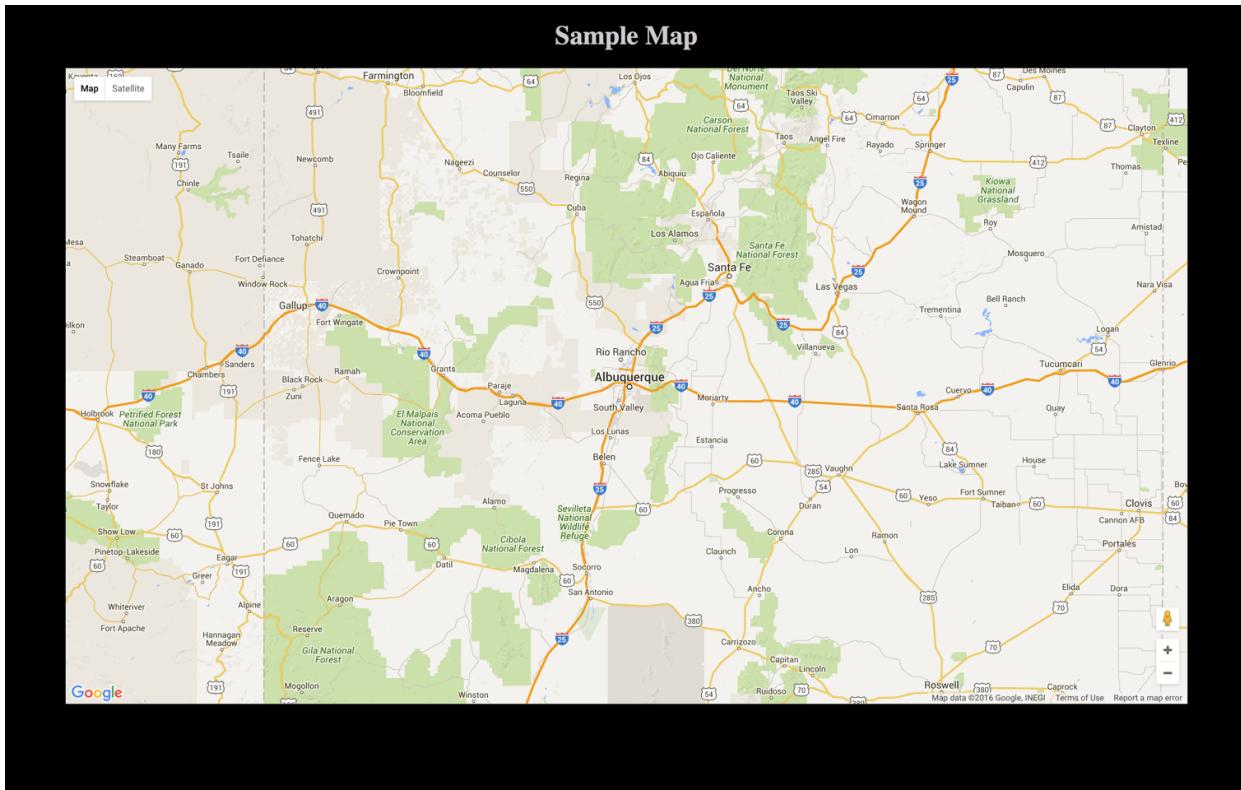
- Elevation
 - Delivery of elevation data for locations or paths
- Streetview
 - Integration of Google Streetview within a DOM element
- Maximum Zoom
 - Provides information about the maximum available zoom level

Events

- Events provide the ability to attach custom behaviors to events in the interface. For example:
 - Changing items in the interface as the user zooms in on a map
 - Displaying additional information outside the map when the user clicks a location in the map
 - Synchronizing the behavior of multiple maps as the user interacts with one map
- Requires higher-level Javascript than we will cover in this course

Examples

Simple - Roadmap



Simple - Roadmap Code

gmmaps01.html

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <link rel="stylesheet" type="text/css" href="css/mapPage.css">
5      </head>
6
7      <body>
8          <h1>Sample Map</h1>
9          <div id="map_canvas"></div>
10
11         <!-- Let's put our JavaScript down here -->
12         <!-- Load the external JavaScript file with the map definition code -->
13         <script src="js/mapPage_01.js"></script>
14
15         <!-- Load the API in asynchronous mode and execute the initialize
16             function when done -->
17         <!-- The optional 'key=<API Key>' parameter is not used here but should be
18             for a production map -->
19         <script async defer

```

```

20      src="https://maps.googleapis.com/maps/api/js?callback=initialize">
21    </script>
22  </body>
23</html>
```

mapPage.css

```

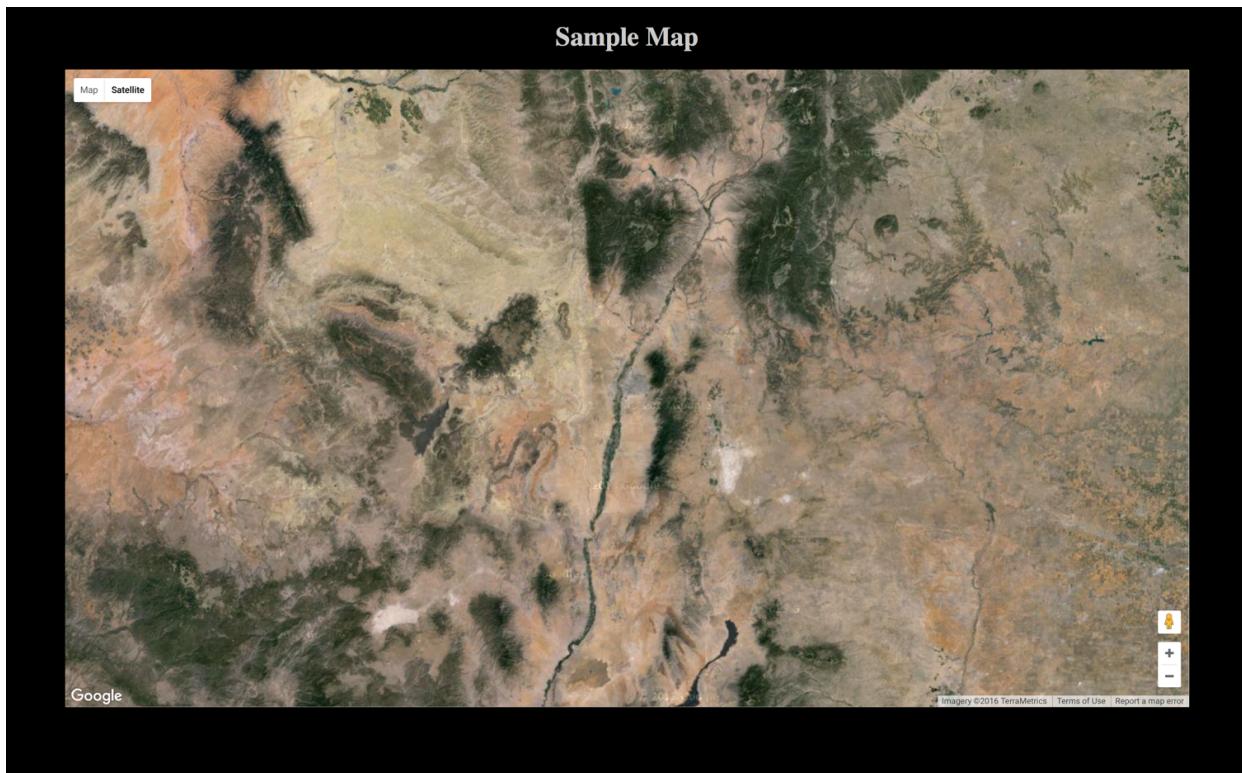
1  /* You must set the height of either the 'html' or 'body' elements for some
2   browsers to properly render the map with a height taller than 0px */
3 html {
4   height: 100%
5
6 body {
7   height: 100%;
8   margin: 0px;
9   padding: 0px;
10  background-color: black;
11  color: #CCCCCC;
12  text-align: center}
13
14 #map_canvas {
15   width:90%;
16   height:80%;
17   margin-left:auto;
18   margin-right: auto }
19
20 .infoBox {
21   color:black }
```

mapPage_01.js

```

1 function initialize() {
2   var classroom = new google.maps.LatLng(35.084280,-106.624073)
3   var mapOptions = {
4     zoom: 8,
5     center: classroom,
6     mapTypeId: google.maps.MapTypeId.ROADMAP
7   };
8   var map = new google.maps.Map(
9     document.getElementById("map_canvas"),
10    mapOptions);
11 }
```

Simple - Satellite



Simple - Satellite Code

gmaps02.html

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <link rel="stylesheet" type="text/css" href="css/mapPage.css">
5      </head>
6
7      <body>
8          <h1>Sample Map</h1>
9          <div id="map_canvas"></div>
10
11     <!-- Let's put our JavaScript down here -->
12     <!-- Load the external JavaScript file with the map definition code -->
13     <script src="js/mapPage_02.js"></script>
14
15     <!-- Load the API in asynchronous mode and execute the initialize
16         function when done -->
17     <!-- The optional 'key-<API Key>' parameter is not used here but should be
18         for a production map -->
19     <script async defer
20         src="https://maps.googleapis.com/maps/api/js?callback=initialize">
21     </script>
```

```
22   </body>
23 </html>
```

mapPage_02.js

```
1 function initialize() {
2     var classroom = new google.maps.LatLng(35.084280,-106.624073)
3     var mapOptions = {
4         zoom: 8,
5         center: classroom,
6         mapTypeId: google.maps.MapTypeId.SATELLITE
7     };
8     var map = new google.maps.Map(
9         document.getElementById("map_canvas"),
10        mapOptions);
11 }
12 }
```

Simple - Hybrid



Simple - Hybrid Code

gmaps03.html

```
1 <!DOCTYPE html>
2 <html>
```

```

3   <head>
4     <link rel="stylesheet" type="text/css" href="css/mapPage.css">
5   </head>
6
7   <body>
8     <h1>Sample Map</h1>
9     <div id="map_canvas"></div>
10
11    <!-- Let's put our JavaScript down here ----->
12    <!-- Load the external JavaScript file with the map definition code -->
13    <script src="js/mapPage_03.js"></script>
14
15    <!-- Load the API in asynchronous mode and execute the initialize
16        function when done -->
17    <!-- The optional 'key-<API Key>' parameter is not used here but should be
18        for a production map -->
19    <script async defer
20      src="https://maps.googleapis.com/maps/api/js?callback=initialize">
21    </script>
22  </body>
23 </html>

```

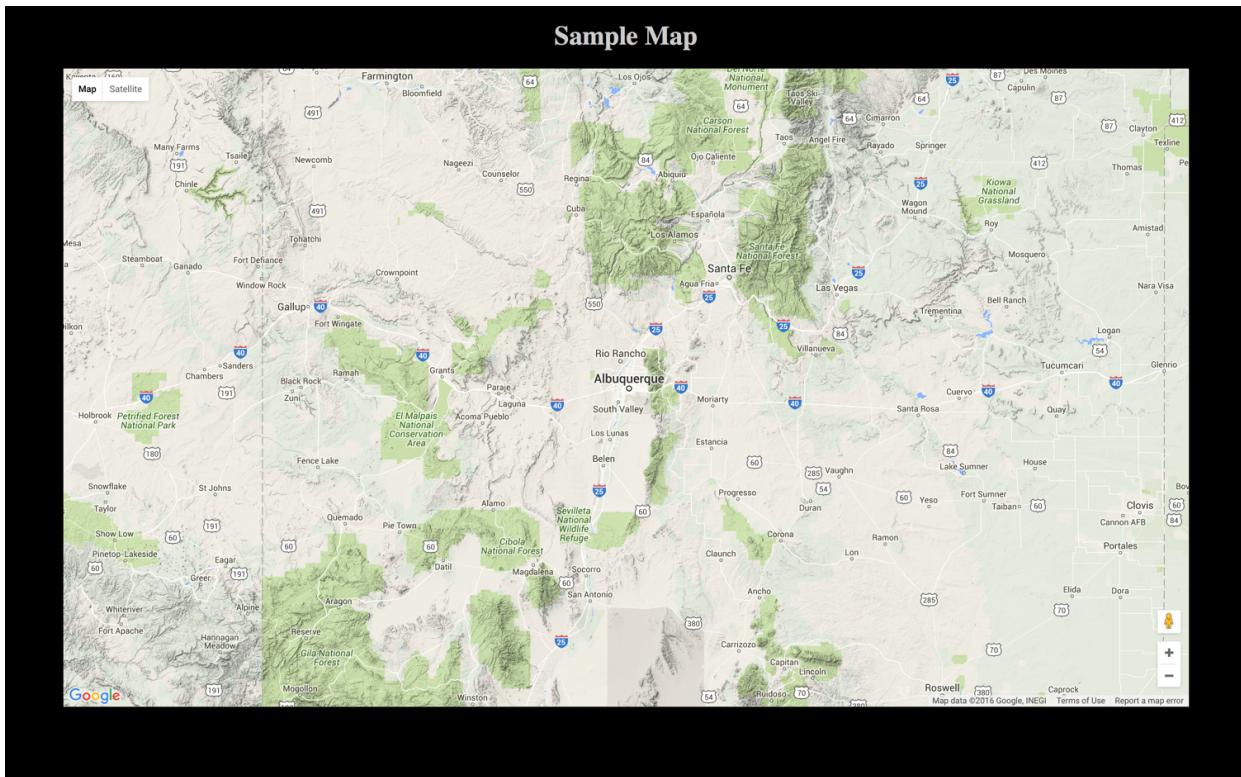
mapPage_03.js

```

1  function initialize() {
2    var classroom = new google.maps.LatLng(35.084280,-106.624073)
3    var mapOptions = {
4      zoom: 8,
5      center: classroom,
6      mapTypeId: google.maps.MapTypeId.HYBRID
7    };
8    var map = new google.maps.Map(
9      document.getElementById("map_canvas"),
10     mapOptions);
11 }

```

Simple - Terrain



Simple - Terrain Code

gmmaps04.html

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <link rel="stylesheet" type="text/css" href="css/mapPage.css">
5      </head>
6
7      <body>
8          <h1>Sample Map</h1>
9          <div id="map_canvas"></div>
10
11     <!-- Let's put our JavaScript down here -->
12     <!-- Load the external JavaScript file with the map definition code -->
13     <script src="js/mapPage_04.js"></script>
14
15     <!-- Load the API in asynchronous mode and execute the initialize
16         function when done -->
17     <!-- The optional 'key-<API Key>' parameter is not used here but should be
18         for a production map -->
19     <script async defer
20         src="https://maps.googleapis.com/maps/api/js?callback=initialize">
21     </script>

```

```
22   </body>
23 </html>
```

mapPage_04.js

```
1  function initialize() {
2      var classroom = new google.maps.LatLng(35.084280,-106.624073)
3      var mapOptions = {
4          zoom: 8,
5          center: classroom,
6          mapTypeId: google.maps.MapTypeId.TERRAIN
7      };
8      var map = new google.maps.Map(
9          document.getElementById("map_canvas"),
10         mapOptions);
11 }
12 }
```

Simple - Hybrid - Zoomed



Simple - Hybrid - Zoomed Code

gmaps05.html

```
1  <!DOCTYPE html>
2 <html>
```

```

3   <head>
4     <link rel="stylesheet" type="text/css" href="css/mapPage.css">
5   </head>
6
7   <body>
8     <h1>Sample Map</h1>
9     <div id="map_canvas"></div>
10
11    <!-- Let's put our JavaScript down here --->
12    <!-- Load the external JavaScript file with the map definition code -->
13    <script src="js/mapPage_05.js"></script>
14
15    <!-- Load the API in asynchronous mode and execute the initialize
16        function when done -->
17    <!-- The optional 'key-<API Key>' parameter is not used here but should be
18        for a production map -->
19    <script async defer
20      src="https://maps.googleapis.com/maps/api/js?callback=initialize">
21    </script>
22  </body>
23 </html>

```

mapPage_05.js

```

1  function initialize() {
2    var classroom = new google.maps.LatLng(35.084280,-106.624073)
3    var mapOptions = {
4      zoom: 18,
5      center: classroom,
6      mapTypeId: google.maps.MapTypeId.TERRAIN
7    };
8    var map = new google.maps.Map(
9      document.getElementById("map_canvas"),
10     mapOptions);
11 }

```

Simple - Zoomed - Modified Controls



Simple - Zoomed - Modified Controls Code

gmaps06.html

```

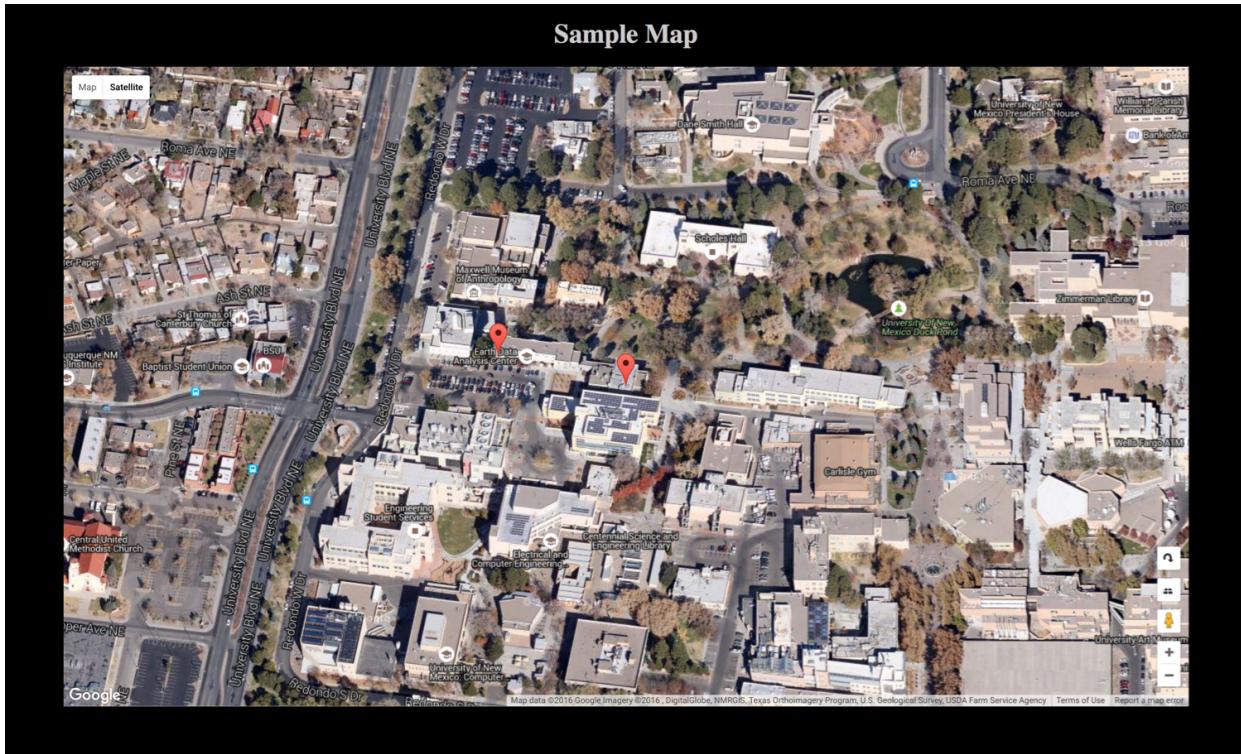
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <link rel="stylesheet" type="text/css" href="css/mapPage.css">
5      </head>
6
7      <body>
8          <h1>Sample Map</h1>
9          <div id="map_canvas"></div>
10
11     <!-- Let's put our JavaScript down here -->
12     <!-- Load the external JavaScript file with the map definition code -->
13     <script src="js/mapPage_06.js"></script>
14
15     <!-- Load the API in asynchronous mode and execute the initialize
16         function when done -->
17     <!-- The optional 'key-<API Key>' parameter is not used here but should be
18         for a production map -->
19     <script async defer
20         src="https://maps.googleapis.com/maps/api/js?callback=initialize">
21     </script>
```

```
22   </body>
23 </html>
```

mapPage_06.js

```
1  function initialize() {
2  var classroom = new google.maps.LatLng(35.084280,-106.624073)
3  var myOptions = {
4      zoom: 18,
5      center: classroom,
6      mapTypeId: google.maps.MapTypeId.HYBRID,
7      zoomControl: true,
8      zoomControlOptions: {style: google.maps.ZoomControlStyle.SMALL},
9      mapTypeControl: true,
10     mapTypeControlOptions: {
11         style: google.maps.MapTypeControlStyle.DROPDOWN_MENU},
12     streetViewControl: false
13 };
14 var map = new google.maps.Map(
15     document.getElementById("map_canvas"),
16     myOptions);
17 }
18
```

Markers



Markers Code

gmaps07.html

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <link rel="stylesheet" type="text/css" href="css/mapPage.css">
5      </head>
6
7      <body>
8          <h1>Sample Map</h1>
9          <div id="map_canvas"></div>
10
11     <!-- Let's put our JavaScript down here ----->
12     <!-- Load the external JavaScript file with the map definition code -->
13     <script src="js/mapPage_07.js"></script>
14
15     <!-- Load the API in asynchronous mode and execute the initialize
16         function when done -->
17     <!-- The optional 'key-<API Key>' parameter is not used here but should be
18         for a production map -->
19     <script async defer
20         src="https://maps.googleapis.com/maps/api/js?callback=initialize">
21     </script>
22     </body>
23 </html>
```

mapPage_07.js

```

1  function initialize() {
2      var classroom = new google.maps.LatLng(35.084280,-106.624073)
3      var office = new google.maps.LatLng(35.084506,-106.624899)
4      var myOptions = {
5          zoom: 18,
6          center: classroom,
7          mapTypeId: google.maps.MapTypeId.HYBRID
8      };
9      var map = new google.maps.Map(
10         document.getElementById("map_canvas"),
11         myOptions);
12
13      var classroomMarker = new google.maps.Marker({
14          position: classroom,
15          title:"Geography 485L/585L Classroom, Bandelier East, Room 106"
16      );
17      classroomMarker.setMap(map);
18
19      var officeMarker = new google.maps.Marker({
20          position: office,
21          title:"Office, Bandelier West, Room 107"
22      );
23      officeMarker.setMap(map);
```

```
24    }
25
```

Polyline



Polyline Code

gmaps08.html

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <link rel="stylesheet" type="text/css" href="css/mapPage.css">
5      </head>
6
7      <body>
8          <h1>Sample Map</h1>
9          <div id="map_canvas"></div>
10
11         <!-- Let's put our JavaScript down here -->
12         <!-- Load the external JavaScript file with the map definition code -->
13         <script src="js/mapPage_08.js"></script>
14
15         <!-- Load the API in asynchronous mode and execute the initialize
16             function when done -->
17         <!-- The optional 'key-<API Key>' parameter is not used here but should be
18             for a production map -->
```

```

19      <script async defer
20          src="https://maps.googleapis.com/maps/api/js?callback=initialize">
21      </script>
22  </body>
23</html>

```

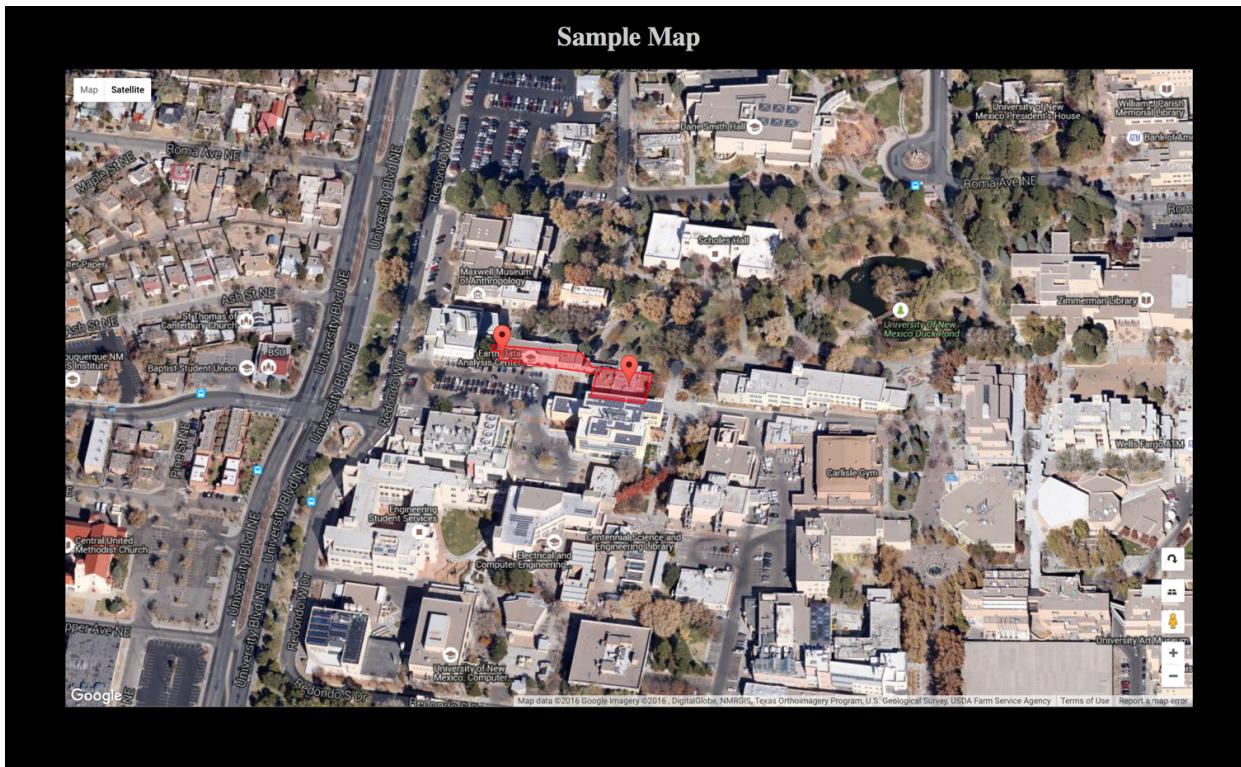
mapPage_08.js

```

1  var classroom = new google.maps.LatLng(35.084280,-106.624073)
2  var office = new google.maps.LatLng(35.084506,-106.624899)
3  var myOptions = {
4      zoom: 18,
5      center: classroom,
6      mapTypeId: google.maps.MapTypeId.HYBRID
7  };
8  var map = new google.maps.Map(
9      document.getElementById("map_canvas"),
10     myOptions);
11
12 var classroomMarker = new google.maps.Marker({
13     position: classroom,
14     title:"Geography 485L/585L Classroom, Bandelier East, Room 106"
15 });
16 classroomMarker.setMap(map);
17
18 var officeMarker = new google.maps.Marker({
19     position: office,
20     title:"Office, Bandelier West, Room 107"
21 });
22 officeMarker.setMap(map);
23
24 var officeVisitCoordinates = [
25     office,
26     new google.maps.LatLng(35.084445,-106.624327),
27     new google.maps.LatLng(35.084309,-106.624308),
28     classroom
29 ];
30 var officePath = new google.maps.Polyline({
31     path: officeVisitCoordinates,
32     strokeColor: "#FF0000",
33     strokeOpacity: 1.0,
34     strokeWeight: 2
35 });
36 officePath.setMap(map)
37 }
38

```

Polygon



Polygon Code

gmaps09.html

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <link rel="stylesheet" type="text/css" href="css/mapPage.css">
5      </head>
6
7      <body>
8          <h1>Sample Map</h1>
9          <div id="map_canvas"></div>
10
11     <!-- Let's put our JavaScript down here -->
12     <!-- Load the external JavaScript file with the map definition code -->
13     <script src="js/mapPage_09.js"></script>
14
15     <!-- Load the API in asynchronous mode and execute the initialize
16         function when done -->
17     <!-- The optional 'key-<API Key>' parameter is not used here but should be
18         for a production map -->
19     <script async defer
20         src="https://maps.googleapis.com/maps/api/js?callback=initialize">
21     </script>

```

```

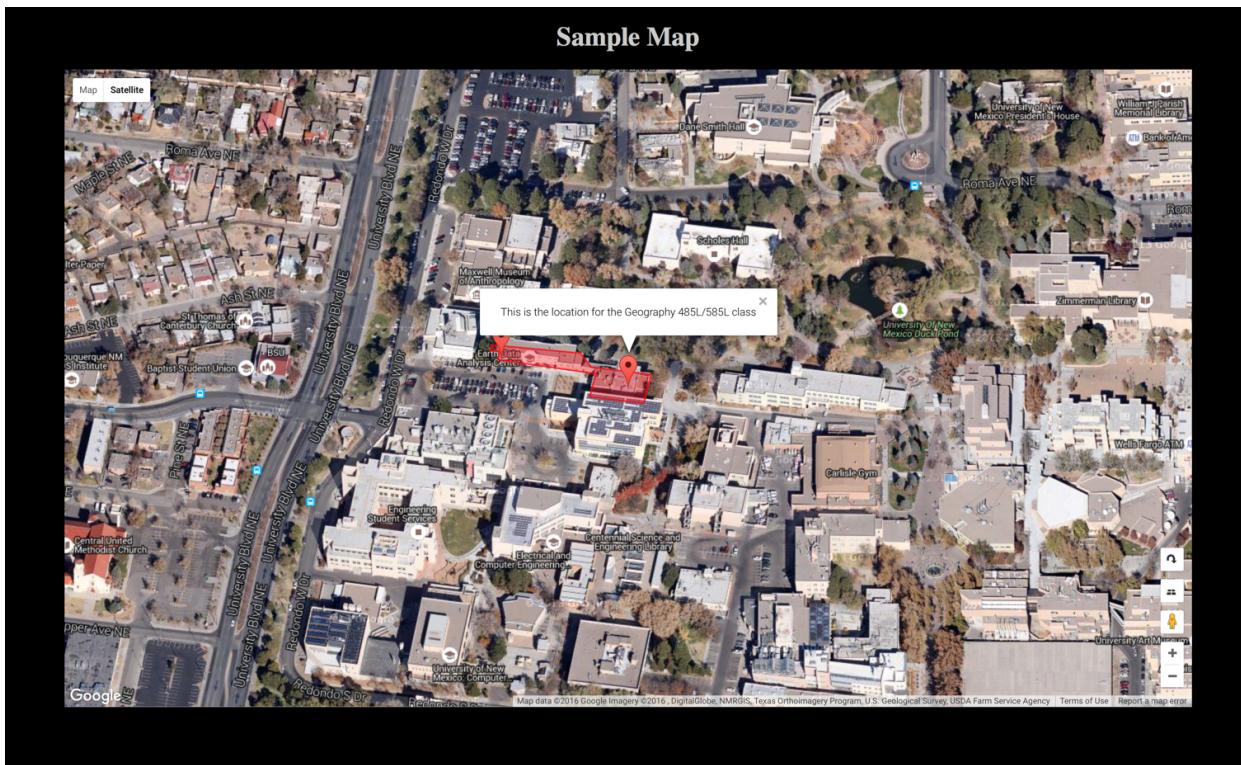
22   </body>
23 </html>
```

mapPage_09.js

```

1  function initialize() {
2      var classroom = new google.maps.LatLng(35.084280,-106.624073)
3      var office = new google.maps.LatLng(35.084506,-106.624899)
4      var myOptions = {
5          zoom: 18,
6          center: classroom,
7          mapTypeId: google.maps.MapTypeId.HYBRID
8      };
9      var map = new google.maps.Map(
10         document.getElementById("map_canvas"),
11         myOptions);
12      var classroomMarker = new google.maps.Marker({
13         position: classroom,
14         title:"Geography 485L/585L Classroom, Bandelier East, Room 106"
15     });
16      classroomMarker.setMap(map);
17      var officeMarker = new google.maps.Marker({
18         position: office,
19         title:"Office, Bandelier West, Room 107"
20     });
21      officeMarker.setMap(map);
22      var buildingCoordinates = [
23         new google.maps.LatLng(35.084498,-106.624921),
24         new google.maps.LatLng(35.084558,-106.624911),
25         new google.maps.LatLng(35.084566,-106.624970),
26         new google.maps.LatLng(35.084609,-106.624966),
27         new google.maps.LatLng(35.084544,-106.624383),
28         new google.maps.LatLng(35.084438,-106.624317),
29         new google.maps.LatLng(35.084384,-106.623922),
30         new google.maps.LatLng(35.084164,-106.623970),
31         new google.maps.LatLng(35.084214,-106.624324),
32         new google.maps.LatLng(35.084214,-106.624324),
33         new google.maps.LatLng(35.084391,-106.624284)
34     ];
35      var bldgPoly = new google.maps.Polygon({
36         paths: buildingCoordinates,
37         strokeColor: "#FF0000",
38         strokeOpacity: 0.8,
39         strokeWeight: 2,
40         fillColor: "#FF0000",
41         fillOpacity: 0.35
42     );
43     bldgPoly.setMap(map)
44 }
```

Adding an Info Window



Adding an Info Window Code

gmaps10.html

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <link rel="stylesheet" type="text/css" href="css/mapPage.css">
5      </head>
6
7      <body>
8          <h1>Sample Map</h1>
9          <div id="map_canvas"></div>
10
11     <!-- Let's put our JavaScript down here -->
12     <!-- Load the external JavaScript file with the map definition code -->
13     <script src="js/mapPage_10.js"></script>
14
15     <!-- Load the API in asynchronous mode and execute the initialize
16         function when done -->
17     <!-- The optional 'key-<API Key>' parameter is not used here but should be
18         for a production map -->
19     <script async defer
20         src="https://maps.googleapis.com/maps/api/js?callback=initialize">
21     </script>
```

```

22   </body>
23 </html>

mapPage_10.js

1  function initialize() {
2      var classroom = new google.maps.LatLng(35.084280,-106.624073)
3      var office = new google.maps.LatLng(35.084506,-106.624899)
4      var myOptions = {
5          zoom: 18,
6          center: classroom,
7          mapTypeId: google.maps.MapTypeId.HYBRID
8      };
9      var map = new google.maps.Map(
10         document.getElementById("map_canvas"),
11         myOptions);
12      var classroomMarker = new google.maps.Marker({
13          position: classroom,
14          title:"Geography 485L/585L Classroom, Bandelier East, Room 106"
15      });
16      classroomMarker.setMap(map);
17      var officeMarker = new google.maps.Marker({
18          position: office,
19          title:"Office, Bandelier West, Room 107"
20      });
21      officeMarker.setMap(map);
22      var buildingCoordinates = [
23          new google.maps.LatLng(35.084498,-106.624921),
24          new google.maps.LatLng(35.084558,-106.624911),
25          new google.maps.LatLng(35.084566,-106.624970),
26          new google.maps.LatLng(35.084609,-106.624966),
27          new google.maps.LatLng(35.084544,-106.624383),
28          new google.maps.LatLng(35.084438,-106.624317),
29          new google.maps.LatLng(35.084384,-106.623922),
30          new google.maps.LatLng(35.084164,-106.623970),
31          new google.maps.LatLng(35.084214,-106.624324),
32          new google.maps.LatLng(35.084214,-106.624324),
33          new google.maps.LatLng(35.084391,-106.624284)
34      ];
35      var bldgPoly = new google.maps.Polygon({
36          paths: buildingCoordinates,
37          strokeColor: "#FF0000",
38          strokeOpacity: 0.8,
39          strokeWeight: 2,
40          fillColor: "#FF0000",
41          fillOpacity: 0.35
42      });
43      bldgPoly.setMap(map);
44      var classInfoContent = '<div class="infoBox">' +
45          '<p>This is the location for the Geography 485L/585L class</p>' +
46          '</div>';
47      var classInfoWindow = new google.maps.InfoWindow({
48          content: classInfoContent
49      });

```

```
50     google.maps.event.addListener(classroomMarker, 'click', function() {
51         classInfoWindow.open(map, classroomMarker);
52     });
53 }
54
```


Chapter 4

Week 4 - Module 2a - Web-based Mapping Clients. Google Maps API (pt. 2)

Overview

- Additional Google Maps API Capabilities to be Aware of
 - Styling of the base maps with custom preferences
 - Fusion Tables
- Bringing it all together in a “real” web page

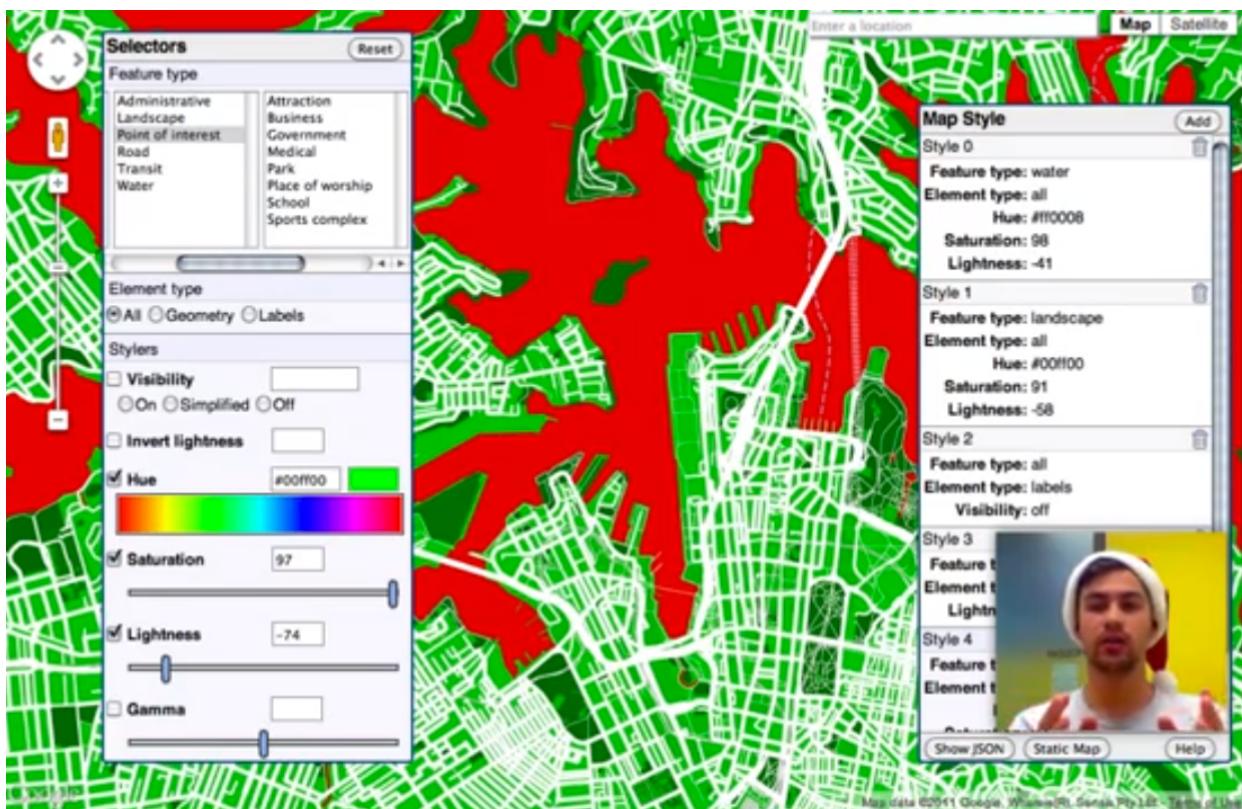
Getting Started with Styled Maps - Video

[Styled Maps Documentation](#) | [Styled Maps Wizard](#)

Map Example: Simple - Styled

gmap_styled.html

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <link rel="stylesheet" type="text/css" href="css/mapPage.css">
5   </head>
6
7   <body>
8     <h1>Sample Map - Styled (POIs Highlighted)</h1>
9     <div id="map_canvas"></div>
10
11    <!-- Let's put our JavaScript down here ----->
12    <!-- Load the external JavaScript file with the map definition code -->
13    <script src="js/mapPage_styled.js"></script>
14
```

Figure 4.1: Google Maps Styled Maps Wizard [link](#)

```

15    <!-- Load the API in asynchronous mode and execute the initialize function when done -->
16    <!-- The optional 'key-<API Key>' parameter is not used here but should be for a production map
17    <script async defer
18      src="https://maps.googleapis.com/maps/api/js?callback=initialize">
19    </script>
20  </body>
21 </html>
```

mapPage_styled.js

```

1  function initialize() {
2    var classroom = new google.maps.LatLng(35.084280,-106.624073)
3    var myOptions = {
4      zoom: 8,
5      center: classroom,
6      mapTypeId: google.maps.MapTypeId.ROADMAP,
7      styles: [
8        {
9          featureType: "water",
10         stylers: [
11           { visibility: "on" },
12           { hue: "#0008ff" }
13         ]
14       },{
15         featureType: "road.highway",
16         stylers: [
17           { hue: "#ff1a00" }
18         ]
19       },{
20         featureType: "road.arterial",
21         stylers: [
22           { hue: "#ffa200" },
23           { visibility: "simplified" }
24         ]
25       },{
26         featureType: "road.local",
27         stylers: [
28           { visibility: "off" }
29         ]
30       },{
31         featureType: "administrative",
32         stylers: [
33           { visibility: "simplified" }
34         ]
35       },{
36         featureType: "poi",
37         stylers: [
38           { visibility: "on" },
39           { hue: "#00ffff" }
40         ]
41       },{
42         featureType: "poi",
43         stylers: [
44           { visibility: "on" }
```

```

45      ]
46    }
47  ]
48  };
49
50  var map = new google.maps.Map(
51    document.getElementById("map_canvas"),
52    myOptions);
53}

```

Google I/O 2011: Managing and visualizing your geospatial data with Fusion Tables - Video

Some particularly relevant sections: [Introduction \(0:00 - 10:30\)](#) | [Google Maps API Integration \(21:40 - 34:42\)](#)
 | [Summary and Links \(52:00 52:40\)](#)

[Fusion Tables Documentation/Help](#)



Figure 4.2: Google Fusion Tables Introduction Video [link](#)

Bringing It All Together

gmmaps_events.html

```

1  <!DOCTYPE html>
2  <html>

```

```

3   <head>
4     <link rel="stylesheet" type="text/css" href="css/event_mapPage.css">
5     <title>Karl's Event Diary</title>
6   </head>
7
8   <body>
9     <h1>My diary of endurance events that I've participated in since joining Team in Training
10    </h1>
11
12    <p>In 2008 Cynthia and I joined the Leukemia and Lymphoma Society's
13      (<a href="http://www.lls.org/">LLS</a>) Team in Training
14      (<a href="http://www.teamintraining.org/">TNT</a>,
15      <a href="http://youtu.be/GMSKG8L6K78">info video</a>) program as
16      participants to train for the Animas Valley/Steamworks Half Marathon
17      and raise money for blood cancer research and patient services. In spite of
18      our not having any direct connection to blood cancer (at that time),
19      we found the goals of LLS admirable, the combined training and fund-raising
20      program of TNT a great idea, and have made many new friends over the many
21      seasons that we've been involved with TNT.</p>
22
23    <p>From 2008 through early 2015 we continued to volunteer with TNT, as
24      participants, mentors, and since 2010 I was a coach (check out my
25      <a href="http://youtu.be/GMSKG8L6K78#t=2m13s">half-second</a> of fame in the
26      info video) for TNT with an emphasis on training walkers for full- or
27      half-marathons. This page provides a summary of the events that I've participated
28      in in some capacity since we became involved with TNT. </p>
29
30    <div id="event-map" name="event-map"></div>
31
32    <h2>
33      <span class="date">11/13/2015</span>
34      Avengers Half Marathon
35      <span class="time">3:17:55</span>
36      (<a href="#event-map" onclick="recenter(map, eventPlaces[5].point, 12)">map</a>)
37    </h2>
38
39    <h2>
40      <span class="date">1/11/2015</span>
41      Disney World Marathon (Goofy - Day 2)
42      <span class="time">6:21:01</span>
43      (<a href="#event-map" onclick="recenter(map, eventPlaces[4].point, 10)">map</a>)
44    </h2>
45    <p class="eventDescription">blah, blah, blah ...</p>
46
47    <h2>
48      <span class="date">1/10/2015</span>
49      Disney World Half Marathon (Goofy - Day 1)
50      <span class="time">2:45:55</span>
51      (<a href="#event-map" onclick="recenter(map, eventPlaces[4].point, 10)">map</a>)
52    </h2>
53    <h2>
54      <span class="date">10/19/2014</span>
55      Duke City Half Marathon
56      <span class="time">2:45:17</span>

```

```

57      (<a href="#event-map" onclick="recenter(map, eventPlaces[0].point, 11)">map</a>
58    </h2>
59    <p class="eventDescription">blah, blah, blah ...</p>
60    <h2>
61      <span class="date">2/23/2014</span>
62      Princess Half Marathon
63      <span class="time">3:07:11</span>
64      (<a href="#event-map" onclick="recenter(map, eventPlaces[4].point, 10)">map</a>
65    </h2>
66    <h2>
67      <span class="date">2/22/2014</span>
68      Princess Enchanted 10k
69      <span class="time">1:42:43</span>
70      (<a href="#event-map" onclick="recenter(map, eventPlaces[4].point, 10)">map</a>
71    </h2>
72    <h2>
73      <span class="date">9/1/2013</span>
74      Disneyland Half Marathon
75      <span class="time">2:56:57</span>
76      (<a href="#event-map" onclick="recenter(map, eventPlaces[5].point, 12)">map</a>
77    </h2>
78    <p class="eventDescription">blah, blah, blah ...</p>
79
80    <h2>
81      <span class="date">1/13/2013</span>
82      Disney World Marathon (Goofy - Day 2)
83      <span class="time">6:46:57</span>
84      (<a href="#event-map" onclick="recenter(map, eventPlaces[4].point, 10)">map</a>
85    </h2>
86    <p class="eventDescription">blah, blah, blah ...</p>
87
88    <h2>
89      <span class="date">1/12/2013</span>
90      Disney World Half Marathon (Goofy - Day 1)
91      <span class="time">3:22:48</span>
92      (<a href="#event-map" onclick="recenter(map, eventPlaces[4].point, 10)">map</a>
93    </h2>
94    <p class="eventDescription">blah, blah, blah ...</p>
95
96    <h2>
97      <span class="date">9/29/2012</span>
98      Hot Chocolate 15k
99      <span class="time">1:56:46</span>
100     (<a href="#event-map" onclick="recenter(map, eventPlaces[6].point, 10)">map</a>
101   </h2>
102   <p class="eventDescription">blah, blah, blah ...</p>
103
104  <h2>
105    <span class="date">6/9/2012</span>
106    Animas Valley/Steamworks Half Marathon
107    <span class="time">no time: coached</span>
108    (<a href="#event-map" onclick="recenter(map, eventPlaces[1].point, 10)">map</a>
109  </h2>
110  <p class="eventDescription">blah, blah, blah ...</p>

```

```

111
112 <h2>
113   <span class="date">1/9/2012</span>
114   Disney World Marathon (Goofy - Day 2)
115   <span class="time">6:56:28</span>
116   (<a href="#event-map" onclick="recenter(map, eventPlaces[4].point, 10)">map</a>)
117 </h2>
118 <p class="eventDescription">blah, blah, blah ...</p>
119
120 <h2>
121   <span class="date">1/8/2011</span>
122   Disney World Half Marathon (Goofy - Day 1)
123   <span class="time">3:29:00</span>
124   (<a href="#event-map" onclick="recenter(map, eventPlaces[4].point, 10)">map</a>)
125 </h2>
126 <p class="eventDescription">blah, blah, blah ...</p>
127
128 <h2>
129   <span class="date">6/19/2010</span>
130   Animas Valley/Steamworks Half Marathon
131   <span class="time">no time: coached</span>
132   (<a href="#event-map" onclick="recenter(map, eventPlaces[1].point, 10)">map</a>)
133 </h2>
134 <p class="eventDescription">blah, blah, blah ...</p>
135
136 <h2>
137   <span class="date">6/6/2010</span>
138   San Diego Rock 'n' Roll Marathon
139   <span class="time">no time: coached</span>
140   (<a href="#event-map" onclick="recenter(map, eventPlaces[2].point, 11)">map</a>)
141 </h2>
142 <p class="eventDescription">blah, blah, blah ...</p>
143
144 <h2>
145   <span class="date">10/18/09</span>
146   Nike Women's Marathon
147   <span class="time">7:13:05</span>
148   (<a href="#event-map" onclick="recenter(map, eventPlaces[3].point, 12)">map</a>)
149 </h2>
150 <p class="eventDescription">blah, blah, blah ...</p>
151
152 <h2>
153   <span class="date">9/6/2009</span>
154   Disneyland Half Marathon
155   <span class="time">3:43:05</span>
156   (<a href="#event-map" onclick="recenter(map, eventPlaces[5].point, 12)">map</a>)
157 </h2>
158 <p class="eventDescription">blah, blah, blah ...</p>
159
160 <h2>
161   <span class="date">1/11/2009</span>
162   Disney World Marathon
163   <span class="time">6:57:42</span>
164   (<a href="#event-map" onclick="recenter(map, eventPlaces[4].point, 10)">map</a>)

```

```

165    </h2>
166    <p class="eventDescription">blah, blah, blah ...</p>
167
168    <h2>
169        <span class="date">10/19/2008</span>
170        Duke City Half Marathon
171        <span class="time">3:09:42</span>
172        (<a href="#event-map" onclick="recenter(map, eventPlaces[0].point, 11)">map</a>)
173    </h2>
174    <p class="eventDescription">blah, blah, blah ...</p>
175
176    <h2>
177        <span class="date">6/21/2008</span>
178        Animas Valley/Steamworks Half Marathon
179        <span class="time">3:14:52</span>
180        (<a href="#event-map" onclick="recenter(map, eventPlaces[1].point, 10)">map</a>)
181    </h2>
182    <p class="eventDescription">blah, blah, blah ...</p>
183
184 </body>
185
186 <!-- Let's put our JavaScript down here --->
187 <!-- Load the external JavaScript file with the map definition code -->
188 <script src="js/mapPage_events.js"></script>
189
190     <!-- Load the API in asynchronous mode and execute the initialize function
191         when done -->
192     <!-- The optional 'key-<API Key>' parameter is not used here but should
193         be for a production map -->
194     <script async defer
195         src="https://maps.googleapis.com/maps/api/js?callback=initialize">
196     </script>
197 </body>
198 </html>

```

gmmaps_events.js

```

1 var map;
2 var eventPlaces;
3
4 function initialize() {
5     // Define a set of global coordinates for use throughout the web site
6     // Place coordinates derived from GNIS database: http://geonames.usgs.gov/pls/gnispublic
7     eventPlaces = [
8         {
9             name: "Albuquerque",
10            point: new google.maps.LatLng(35.0889356,-106.5747462),
11            label: "Albuquerque: Duke City Half Marathon"
12        },
13        {
14            name: "Durango",
15            point: new google.maps.LatLng(37.2752800,-107.8800667),
16            label: "Durango: Animas Valley/Steamworks Half Marathon"
17        },

```

```

18     {
19         name: "San Diego",
20         point: new google.maps.LatLng(32.7153292,-117.1572551),
21         label: "San Diego: San Diego Rock 'n' Roll Marathon"
22     },
23     {
24         name: "San Francisco",
25         point: new google.maps.LatLng(37.7749295,-122.4194155),
26         label: "San Francisco: Nike Women's Marathon"
27     },
28     {
29         name: "Orlando",
30         point: new google.maps.LatLng(28.5383355,-81.3792365),
31         label: "Orlando: Walt Disney World half- and full-marathon"
32     },
33     {
34         name: "Anaheim",
35         point: new google.maps.LatLng(33.8352932,-117.9145036),
36         label: "Anaheim: Disneyland Half Marathon"
37     },
38     {
39         name: "Albuquerque",
40         point: new google.maps.LatLng(35.0889356,-106.5747462),
41         label: "Hot Chocolate 15k"
42     }
43 ];
44
45
46 var myOptions = {
47     zoom: 4,
48     center: eventPlaces[0].point,
49     mapTypeId: google.maps.MapTypeId.ROADMAP};
50
51 map = new google.maps.Map(
52     document.getElementById("event-map"),
53     myOptions);
54
55 addMarkers(map,eventPlaces)
56 }
57
58 function recenter(mapName, latlon, zoomLevel) {
59     mapName.setCenter(latlon);
60     mapName.setZoom(zoomLevel)
61 }
62
63 function addMarkers(mapName, markerArray) {
64     for (index = 0; index < markerArray.length; index++) {
65         myMarker = new google.maps.Marker({
66             position: markerArray[index].point,
67             title: markerArray[index].label
68         });
69         myMarker.setMap(mapName)
70     }
71 }
```

This work by Karl Benedict is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.