

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

Karl Benedict

GEOG 485L/585L - Spring 2017

## 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 now required](#)
- 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 <https://developers.google.com/maps/documentation/javascript/?csw=1>

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
  - Scale Control
  - MapType Control
  - Street View Control
  - Rotate (for maps that contain 45-degree imagery)
  - Fullscreen Control
- 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 ([reference](#))

**Polyline** linear features defined by multiple points with a defined style for the line ([reference](#))

**Polygon** closed features defined by multiple points. Supports multi-polygons, and donuts. Line and fill styles may be specified. ([reference](#))

**(Ground) Overlay Maps** Image-based map layers that replace or overlay Google layers - registered to the map coordinates ([reference](#))

**Info Windows** floating content windows for displaying content defined as HTML, a DOM element, or text string ([reference](#))

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

**Custom Overlays** definition of programmatically controlled layers ([reference](#))

## Services

- Geocoding Service

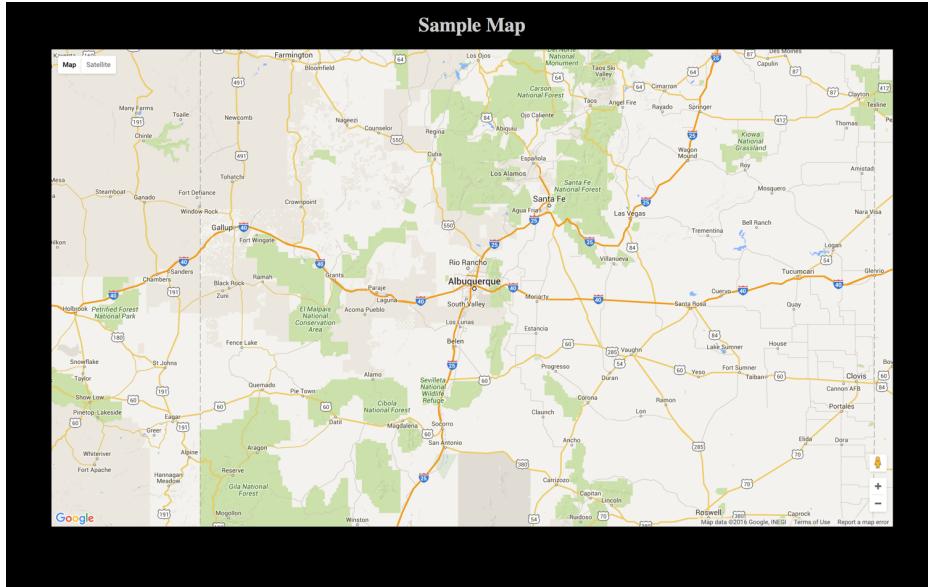
- Forward and reverse geocoding:
  - \* address to LatLon
  - \* LatLon to Nearest Address
- May be biased to current viewport, region
- Directions
  - Based upon an origin, destination, and a variety of additional options
  - Available directions and rendered route
- Distance Matrix
  - Travel distance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

gmaps01.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    <script async defer
18      src="https://maps.googleapis.com/maps/api/js?callback=initialize&key=<mykey>">
19    </script>
20  </body>
21</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 }
12

```

## 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 -->

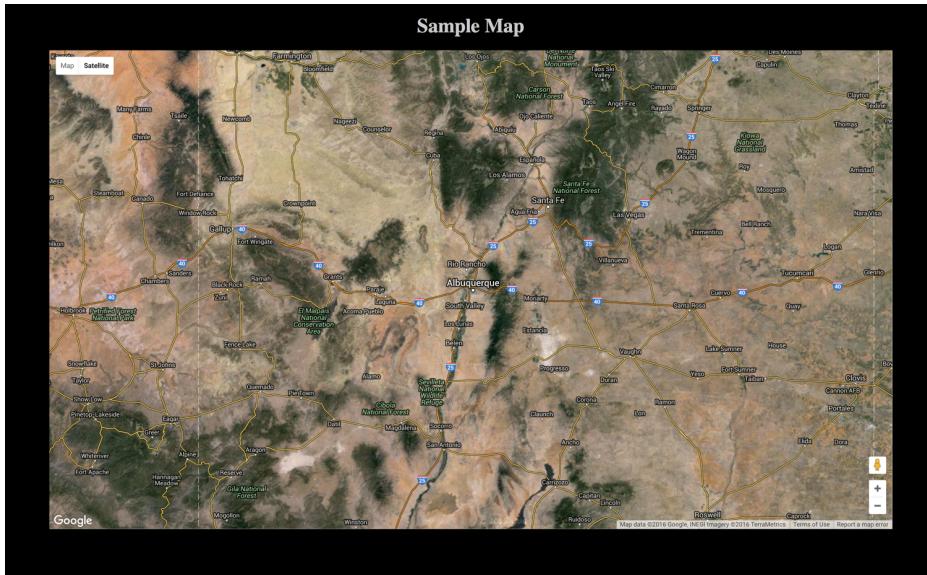
```

```
17
18     <script async defer
19         src="https://maps.googleapis.com/maps/api/js?callback=initialize&key=<mykey>">
20     </script>
21   </body>
22 </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
18  <script async defer
19    src="https://maps.googleapis.com/maps/api/js?callback=initialize&key=<mykey>">
20  </script>
21 </body>
22 </html>

```

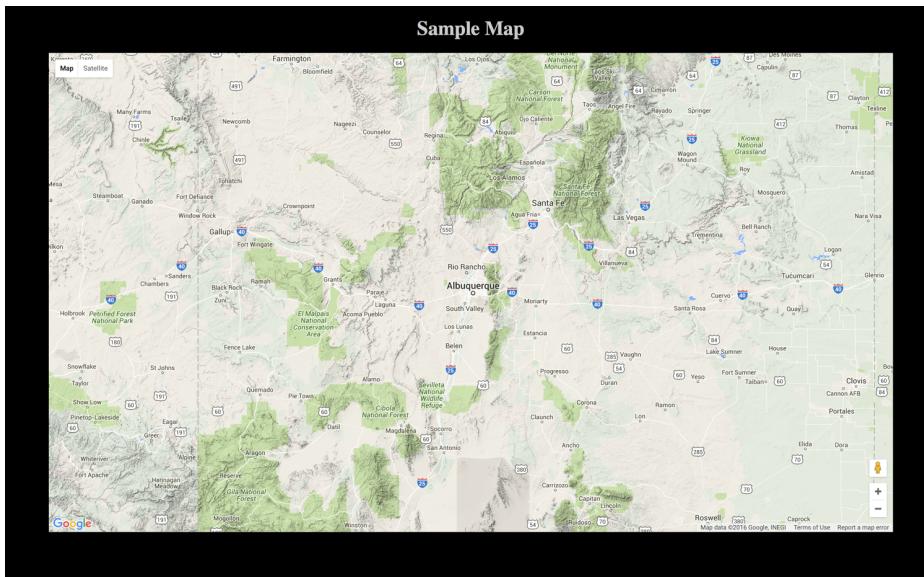
### mapPage\_03.js

```

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 }
12

```

### Simple - Terrain



## Simple - Terrain Code

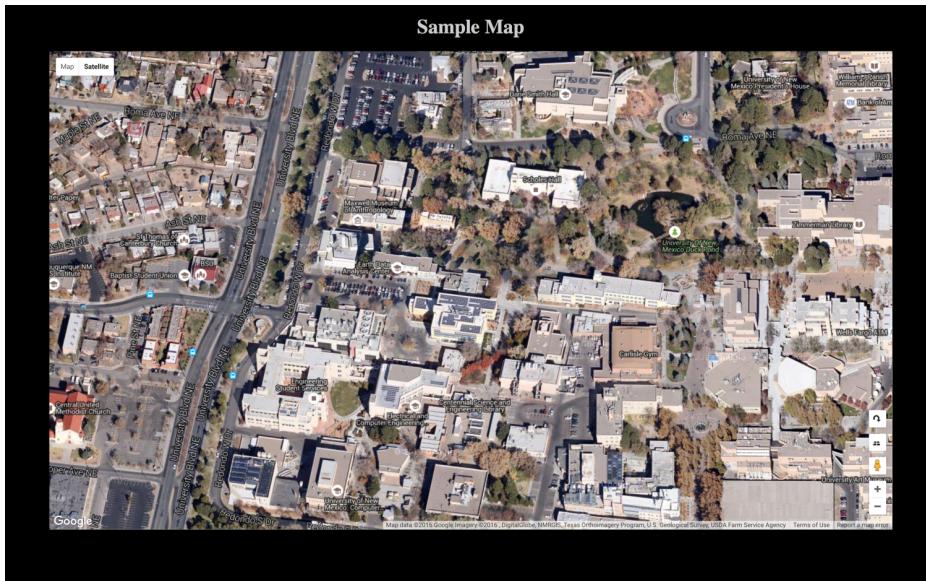
gmaps04.html

```
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
18    <script async defer
19        src="https://maps.googleapis.com/maps/api/js?callback=initialize&key=<mykey>">
20        </script>
21    </body>
22 </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 }
```

## 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
18     <script async defer
19         src="https://maps.googleapis.com/maps/api/js?callback=initialize&key=<mykey>">
20     </script>
21     </body>
22 </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 }
12

```

## 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
18     <script async defer
19         src="https://maps.googleapis.com/maps/api/js?callback=initialize&key=<mykey">
20     </script>
21 </body>
22 </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
18     <script async defer
19         src="https://maps.googleapis.com/maps/api/js?callback=initialize&key=<mykey>">
20     </script>
21 </body>
22 </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
18        <script async defer
19            src="https://maps.googleapis.com/maps/api/js?callback=initialize&key=<mykey>">
20        </script>
21    </body>
22 </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
18     <script async defer
19         src="https://maps.googleapis.com/maps/api/js?callback=initialize&key=<mykey>">
20     </script>
21     </body>
22 </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 }
45

```

## 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
18    <script async defer
19      src="https://maps.googleapis.com/maps/api/js?callback=initialize&key=<mykey>">
20    </script>
21  </body>
22 </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  var bldgInfoContent = '<div class="infoBox">' +
54      '<p>This is the location of Bandelier East and West on the UNM Campus</p>' +
55      '<iframe src="https://www.google.com/maps/embed?pb=!1m0!3m2!1sen!2sus!4v1486322485343!6m8!1m7!1z'
56

```

```
57     var bldgInfoWindow = new google.maps.InfoWindow({  
58         content: bldgInfoContent  
59     });  
60     google.maps.event.addListener(bldgPoly, 'click', function() {  
61         var position = new google.maps.LatLng(35.084438,-106.624317)  
62         bldgInfoWindow.setPosition(position)  
63         bldgInfoWindow.open(map);  
64     });  
65 }  
66  
67
```

---

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