LOCATION FEATURE

Html CSS

FEBRUARY 19, 2017

Contents

Location Features

- Location.html

```
<!DOCTYPE html>
<html>

    <head>
      <script src = "https://maps.googleapis.com/maps/api/js"></script>

      <script>
        function loadMap() {

          var mapOptions = {
            center:new google.maps.LatLng(20.593684, 78.96288), zoom:12,
            mapTypeId:google.maps.MapTypeId.ROADMAP
          };

          var map = new google.maps.Map(document.getElementById("sample"),mapOptions);
        }
      </script>

    </head>

    <body onload = "loadMap()">
      <div id = "sample" style = "width:570px; height:580px;"></div>
    </body>

</html>
```

Steps to Load the Map on a Webpage

Follow the steps given below to load a map on your webpage −

Step 1 : Create an HTML Page

- Create a basic HTML page with head and body tags as shown below −

```
<!DOCTYPE html>
<html>

  <head>
  </head>

  <body>
  ..............
  </body>

</html>
```

Step 2 : Load the API

- Load or include the Google Maps API using the script tag as shown below −

```
<!DOCTYPE html>
<html>

  <head>
    <script src = "https://maps.googleapis.com/maps/api/js"></script>
  </head>

  <body>
  ..............
  </body>

</html>
```

Step 3 : Create the Container

To hold the map, we have to create a container element, generally the tag (a generic container) is used for this purpose. Create a container element and define its dimensions as shown below −

```
<div id = "sample" style = "width:900px; height:580px;"></div>
```

Step 4 : Map Options

Before initializing the map, we have to create a **mapOptions** object (it is created just like a literal) and set values for map initialization variables. A map has three main options, namely, **centre**, **zoom**, and **maptypeid**.

- **centre** − Under this property, we have to specify the location where we want to centre the map. To pass the location, we have to create a **LatLng** object by passing the latitude and longitudes of the required location to the constructor.
- **zoom** − Under this property, we have to specify the zoom level of the map.
- **maptypeid**− Under this property, we have to specify the type of the map we want. Following are the types of maps provided by Google −
    - ROADMAP (normal, default 2D map)
    - SATELLITE (photographic map)
    - HYBRID (combination of two or more other types)
    - TERRAIN (map with mountains, rivers, etc.)

Within a function, say, **loadMap()**, create the mapOptions object and set the required values for center, zoom, and mapTypeId as shown below.

```
function loadMap() {

  var mapOptions = {

center:new google.maps.LatLng(17.240498, 82.287598),
    zoom:9,
 mapTypeId:google.maps.MapTypeId.ROADMAP
  };
}
```

Step 5 : Create a Map Object

You can create a map by instantiating the JavaScript class called **Map**. While instantiating this class, you have to pass an HTML container to hold the map and the map options for the required map. Create a map object as shown below.

```
var map = new
google.maps.Map(document.getElementById("sample"),mapOptions);
```

Step 6 : load the map

Finally load the map by calling the loadMap() method or by adding DOM listener.

```
google.maps.event.addDomListener(window, 'load', loadMap);
                or
<body onload = "loadMap()">
```

Example

The following example shows how to load the roadmap of the city named Vishakhapatnam with a zoom value of 12.

Vishakhapatnam.html

```
<!DOCTYPE html>
<html>

  <head>
    <script src = "https://maps.googleapis.com/maps/api/js"></script>

    <script>
      function loadMap() {

        var mapOptions = {
          center:new google.maps.LatLng(17.609993, 83.221436),
          zoom:12,
          mapTypeId:google.maps.MapTypeId.ROADMAP
        };

        var map = new
google.maps.Map(document.getElementById("sample"),mapOptions);
      }

      google.maps.event.addDomListener(window, 'load', loadMap);
    </script>

  </head>

  <body>
    <div id = "sample" style = "width:580px; height:400px;"></div>
  </body>

</html>
```

It produces the following output –

Types of Maps

Google Maps provides four types of maps. They are −

- **ROADMAP** − This is the default type. If you haven't chosen any of the types, this will be displayed. It shows the street view of the selected region.
- **SATELLITE** − This is the map type that shows the satellite images of the selected region.
- **HYBRID** − This map type shows the major streets on satellite images.
- **TERRAIN** − This is the map type that shows the terrain and vegetation

Syntax

To select one of these map types, you have to mention the respective map type id in the map options as shown below −

```
var mapOptions = {
  mapTypeId:google.maps.MapTypeId.Id of the required map type
};
```

Roadmap

The following example shows how to select a map of type ROADMAP −

```
<!DOCTYPE html>
<html>

  <head>
    <script src = "https://maps.googleapis.com/maps/api/js"></script>

    <script>
      function loadMap() {

        var mapOptions = {
          center:new google.maps.LatLng(17.609993, 83.221436),
          zoom:9,
          mapTypeId:google.maps.MapTypeId.ROADMAP
        };

        var map = new
google.maps.Map(document.getElementById("sample"),mapOptions);
      }

      google.maps.event.addDomListener(window, 'load', loadMap);
    </script>

  </head>

  <body>
    <div id = "sample" style = "width:580px; height:400px;"></div>
  </body>

</html>
```

Satellite

The following example shows how to select a map of type SATELLITE −

```
<!DOCTYPE html>
<html>

  <head>
    <script src = "https://maps.googleapis.com/maps/api/js"></script>

    <script>
      function loadMap() {

        var mapOptions = {
            center:new google.maps.LatLng(17.609993, 83.221436),
            zoom:9,
            mapTypeId:google.maps.MapTypeId.SATELLITE
        };

        var map = new
google.maps.Map(document.getElementById("sample"),mapOptions);
      }

      google.maps.event.addDomListener(window, 'load', loadMap);
    </script>

  </head>

  <body>
    <div id = "sample" style = "width:580px; height:400px;"></div>
  </body>

</html>
```

Hybrid

The following example shows how to select a map of type HYBRID −

```
<!DOCTYPE html>
<html>

  <head>
    <script src = "https://maps.googleapis.com/maps/api/js"></script>

    <script>
      function loadMap() {

        var mapOptions = {
          center:new google.maps.LatLng(17.609993, 83.221436),
          zoom:9,
          mapTypeId:google.maps.MapTypeId.Hybrid
        };

        var map = new
google.maps.Map(document.getElementById("sample"),mapOptions);
      }

      google.maps.event.addDomListener(window, 'load', loadMap);
    </script>

  </head>

  <body>
    <div id = "sample" style = "width:580px; height:400px;"></div>
  </body>

</html>
```

Terrain

The following example shows how to select a map of type TERRAIN −

```
<!DOCTYPE html>
<html>

  <head>
    <script src = "https://maps.googleapis.com/maps/api/js"></script>

    <script>
      function loadMap() {

        var mapOptions = {
          center:new google.maps.LatLng(17.609993, 83.221436),
          zoom:9,
          mapTypeId:google.maps.MapTypeId.TERRAIN
        };

        var map = new
google.maps.Map(document.getElementById("sample"),mapOptions);
      }

      google.maps.event.addDomListener(window, 'load', loadMap);
    </script>

  </head>

  <body>
    <div id = "sample" style = "width:580px; height:400px;"></div>
  </body>

</html>
```

Increase/Decrease the Zoom Value

You can increase or decrease the zoom value of a map by modifying the value of the **zoom** attribute in the the map options.

Syntax

We can increase or decrease the zoom value of the map using the zoom option. Given below is the syntax to change the zoom value of the current map.

```
var mapOptions = {
   zoom:required zoom value
};
```

Example : Zoom 6

The following code will display the roadmap of the city Vishakhapatnam with a zoom value of 6.

```
<!DOCTYPE html>
<html>

  <head>
    <script src = "https://maps.googleapis.com/maps/api/js"></script>

    <script>
      function loadMap() {

        var mapOptions = {
          center:new google.maps.LatLng(17.609993, 83.221436),
                zoom:6,
          mapTypeId:google.maps.MapTypeId.ROADMAP
        };

        var map = new
google.maps.Map(document.getElementById("sample"),mapOptions);
      }
    </script>

  </head>

  <body onload = "loadMap()">
    <div id = "sample" style = "width:587px; height:400px;"></div>
  </body>

</html>
```

Example : Zoom 10

The following code will display the roadmap of the city Vishakhapatnam with a zoom value of 10.

```
<!DOCTYPE html>
<html>

  <head>
    <script src = "https://maps.googleapis.com/maps/api/js"></script>

    <script>
      function loadMap() {

        var mapOptions = {
          center:new google.maps.LatLng(17.609993, 83.221436),
                zoom:10,
          mapTypeId:google.maps.MapTypeId.ROADMAP
        };

        var map = new
google.maps.Map(document.getElementById("sample"),mapOptions);
      }
    </script>

  </head>

  <body onload = "loadMap()">
    <div id = "sample" style = "width:587px; height:400px;"></div>
  </body>

</html>
```

By default, the city names and option names given on the map will be in English. If required, we can display such information in other languages as well. This process is known as **localization**. In this chapter, we will learn how to localize a map.

Localizing a Map

You can customize (localize) the language of the map by specifying the language option in the URL as shown below.

```
<script src = "https://maps.googleapis.com/maps/api/js?language=zh-Hans"></script>
```

Example

Here is an example that shows how to localize GoogleMaps. Here you can see a roadmap of China that is localized to Chinese language.

```
<!DOCTYPE html>
<html>
  <head>
    <script src = "https://maps.googleapis.com/maps/api/js?language=zh-Hans"></script>

    <script>
      function loadMap() {

        var mapOptions = {
          center:new google.maps.LatLng(32.870360, 101.645508),
          zoom:9,
          mapTypeId:google.maps.MapTypeId.ROADMAP
        };

        var map = new
google.maps.Map(document.getElementById("sample"),mapOptions);
      }
    </script>

  </head>

  <body onload = "loadMap()">
    <div id = "sample" style = "width:580px; height:400px;"></div>
  </body>

</html>
```

Google Maps provides a User Interface with various controls to let the user interact with the map. We can add, customize, and disable these controls.

Default Controls

Here is a list of the default controls provided by Google Maps −

- **Zoom** − To increase and decease the zoom level of the map, we will have a slider with + and − buttons, by default. This slider will be located at the corner of left hand side of the map.
- **Pan** − Just above the zoom slider, there will be a pan control for panning the map.
- **Map Type** − You can locate this control at the top right corner of the map. It provides map type options such as Satellite, Roadmap, and Terrain. Users can choose any of these maps.
- **Street view** − Between the pan icon and the zoom slider, we have a pegman icon. Users can drag this icon and place at a particular location to get its street view.

Example

Here is an example where you can observe the default UI controls provided by Google Maps −

Disabling the UI Default Controls

We can disable the default UI controls provided by Google Maps simply by making the **disableDefaultUI** value true in the map options.

Example

The following example shows how to disable the default UI controls provided by Google Maps.

```
<!DOCTYPE html>
<html>

  <head>
    <script src = "https://maps.googleapis.com/maps/api/js"></script>

    <script>
      function loadMap() {

        var mapOptions = {
          center:new google.maps.LatLng(17.609993, 83.221436),
          zoom:5,
          mapTypeId:google.maps.MapTypeId.ROADMAP,
                disableDefaultUI: true
        };

        var map = new
google.maps.Map(document.getElementById("sample"),mapOptions);
      }
    </script>

  </head>
```

```
  <body onload = "loadMap()">
    <div id = "sample" style = "width:580px; height:400px;"></div>
  </body>

</html>
```

Enabling/Disabling All the Controls

In addition to these default controls, Google Maps also provides three more controls as listed below.

- **Scale** − The Scale control displays a map scale element. This control is not enabled by default.
- **Rotate** − The Rotate control contains a small circular icon which allows you to rotate maps containing oblique imagery. This control appears by default at the top left corner of the map. (See 45° Imagery for more information.)
- **Overview** − To increase and decease the zoom level of the map, we have a slider with + and − buttons, by default. This slider is located at the left corner of the map.

In the map options, we can enable and disable any of the controls provided by Google Maps as shown below −

```
{
  panControl: boolean,
  zoomControl: boolean,
  mapTypeControl: boolean,
  scaleControl: boolean,
  streetViewControl: boolean,
  overviewMapControl: boolean
}
```

Example

The following code shows how to enable all the controls −

```
<!DOCTYPE html>
<html>

  <head>
    <script src = "https://maps.googleapis.com/maps/api/js"></script>

    <script>
      function loadMap() {

        var mapOptions = {
          center:new google.maps.LatLng(19.373341, 78.662109),
          zoom:5,
              panControl: true,
              zoomControl: true,
              scaleControl: true,
              mapTypeControl:true,
              streetViewControl:true,
              overviewMapControl:true,
              rotateControl:true
        }

        var map = new
google.maps.Map(document.getElementById("sample"),mapOptions);
      }
    </script>

  </head>

  <body onload = "loadMap()">
    <div id = "sample" style = "width:580px; height:400px;"></div>
  </body>

</html>
```

Control Options

We can change the appearance of Google Maps controls using its control options. For example, the zoom control can be either reduced or enlarged in size. The MapType control appearance can be varied to a horizontal bar or a drop-down menu. Given below is a list of Control options for Zoom and MapType controls.

Sr.No.	Control Name	Control Options
1	Zoom control	• google.maps.ZoomControlStyle.SMALL • google.maps.ZoomControlStyle.LARGE • google.maps.ZoomControlStyle.DEFAULT
2	MapType control	• google.maps.MapTypeControlStyle.HORIZONTAL_BAR • google.maps.MapTypeControlStyle.DROPDOWN_MENU • google.maps.MapTypeControlStyle.DEFAULT

Example

The following example demonstrates how to use the control options −

```
<!DOCTYPE html>
<html>

  <head>
    <script src = "https://maps.googleapis.com/maps/api/js"></script>

    <script>
      function loadMap() {

        var mapOptions = {
          center:new google.maps.LatLng(19.373341, 78.662109),
          zoom:5,
          mapTypeControl: true,

          mapTypeControlOptions: {
            style: google.maps.MapTypeControlStyle.DROPDOWN_MENU, mapTypeIds: [
              google.maps.MapTypeId.ROADMAP,
              google.maps.MapTypeId.TERRAIN
            ]
          },

          zoomControl: true,

          zoomControlOptions: {
```

```
            style: google.maps.ZoomControlStyle.SMALL
         }
      }

      var map = new
google.maps.Map(document.getElementById("sample"),mapOptions);
   }
  </script>

 </head>

 <body onload = "loadMap()">
  <div id = "sample" style = "width:580px; height:400px;"></div>
 </body>

</html>
```

Control Positioning

You can change the position of the controls by adding the following line in the control options.

position:google.maps.ControlPosition.**Desired_Position**,

Here is the list of available positions where a control can be placed on a map −

- TOP_CENTER
- TOP_LEFT
- TOP_RIGHT
- LEFT_TOP
- RIGHT_TOP
- LEFT_CENTER
- RIGHT_CENTER
- LEFT_BOTTOM
- RIGHT_BOTTOM
- BOTTOM_CENTER
- BOTTOM_LEFT
- BOTTOM_RIGHT

Example

The following example shows how to place the MapTypeid control at the top centre of the map and how to place the zoom control at the bottom centre of the map.

```
<!DOCTYPE html>
<html>

  <head>
    <script src = "https://maps.googleapis.com/maps/api/js"></script>

    <script>
      function loadMap() {

        var mapOptions = {
          center:new google.maps.LatLng(19.373341, 78.662109),
          zoom:5,
          mapTypeControl: true,

          mapTypeControlOptions: {
            style: google.maps.MapTypeControlStyle.DROPDOWN_MENU,
            position:google.maps.ControlPosition.TOP_CENTER,

            mapTypeIds: [
              google.maps.MapTypeId.ROADMAP,
              google.maps.MapTypeId.TERRAIN
            ]
          },

          zoomControl: true,

          zoomControlOptions: {
            style: google.maps.ZoomControlStyle.SMALL,
            position:google.maps.ControlPosition.BOTTOM_CENTER
          }
        }

        var map = new
google.maps.Map(document.getElementById("sample"),mapOptions);
      }
    </script>

  </head>

  <body onload = "loadMap()">
    <div id = "sample" style = "width:580px; height:400px;"></div>
  </body>

</html>
```

We can draw objects on the map and bind them to a desired latitude and longitude. These are called overlays. Google Maps provides various overlays as shown below.

- Markers

- Polylines
- Polygons
- Circle and rectangle
- Info window
- Symbols

To mark a single location on the map, Google Maps provides **markers**. These markers use a standard symbol and these symbols can be customized. This chapter explains how to add markers, and how to customize, animate, and remove them.

Adding a Simple Marker

You can add a simple marker to the map at a desired location by instantiating the marker class and specifying the position to be marked using latlng, as shown below.

```
var marker = new google.maps.Marker({
  position: new google.maps.LatLng(19.373341, 78.662109),
  map: map,
});
```

Example

The following code sets the marker on the city Hyderabad (India).

```
<!DOCTYPE html>
<html>

  <head>
    <script src = "https://maps.googleapis.com/maps/api/js"></script>

    <script>
      function loadMap() {

        var mapOptions = {
          center:new google.maps.LatLng(19.373341, 78.662109),
          zoom:7
        }

        var map = new
google.maps.Map(document.getElementById("sample"),mapOptions);

        var marker = new google.maps.Marker({
          position: new google.maps.LatLng(17.088291, 78.442383),
          map: map,
```

```
        });
      }
    </script>

  </head>

  <body onload = "loadMap()">
    <div id = "sample" style = "width:580px; height:400px;"></div>
  </body>

</html>
```

Animating the Marker

After adding a marker to the map, you can go further and add animations to it such as **bounce** and **drop**. The following code snippet shows how to add bounce and drop animations to a marker.

```
//To make the marker bounce`
animation:google.maps.Animation.BOUNCE

//To make the marker drop
animation:google.maps.Animation.Drop
```

Example

The following code sets the marker on the city Hyderabad with an added animation effect −

```
<!DOCTYPE html>
<html>

  <head>
    <script src = "https://maps.googleapis.com/maps/api/js"></script>

    <script>
      function loadMap() {

        var mapOptions = {
          center:new google.maps.LatLng(17.377631, 78.478603),
          zoom:5
        }

        var map = new
google.maps.Map(document.getElementById("sample"),mapOptions);

        var marker = new google.maps.Marker({
          position: new google.maps.LatLng(17.377631, 78.478603),
          map: map,
          animation:google.maps.Animation.Drop
```

```
        });
      }
    </script>

  </head>

  <body onload = "loadMap()">
    <div id = "sample" style = "width:580px; height:400px;"></div>
  </body>

</html>
```

Customizing the Marker

You can use your own icons instead of the default icon provided by Google Maps. Just set the icon as **icon:'ICON PATH'**. And you can make this icon draggable by setting **draggable:true**.

Example

The following example shows how to customize the marker to a desired icon −

```
<!DOCTYPE html>
<html>

  <head>
    <script src = "https://maps.googleapis.com/maps/api/js"></script>

    <script>
      function loadMap() {

        var mapOptions = {
          center:new google.maps.LatLng(17.377631, 78.478603),
          zoom:5
        }

        var map = new
google.maps.Map(document.getElementById("sample"),mapOptions);

        var marker = new google.maps.Marker({
          position: new google.maps.LatLng(17.377631, 78.478603),
          map: map,
          draggable:true,
          icon:'/scripts/img/logo-footer.png'
        });

        marker.setMap(map);
      }
```

```
    </script>

  </head>

  <body onload = "loadMap()">
    <div id = "sample" style = "width:580px; height:400px;"></div>
  </body>

</html>
```

Removing the Marker

You can remove an existing marker by setting up the marker to null using the **marker.setMap()** method.

Example

The following example shows how to remove the marker from a map −

```
<!DOCTYPE html>
<html>

  <head>
    <script src = "https://maps.googleapis.com/maps/api/js"></script>

    <script>
      function loadMap() {

        var mapOptions = {
          center:new google.maps.LatLng(17.377631, 78.478603),
          zoom:5
        }

        var map = new
google.maps.Map(document.getElementById("sample"),mapOptions);

        var marker = new google.maps.Marker({
          position: new google.maps.LatLng(17.377631, 78.478603),
          map: map,
          animation:google.maps.Animation.Drop
        });

        marker.setMap(null);
      }
    </script>

  </head>
```

```
  <body onload = "loadMap()">
    <div id = "sample" style = "width:580px; height:400px;"></div>
  </body>

</html>
```

Adding a Window

Info Window is used to add any kind of information to the map. For instance, if you want to provide information about a location on the map, you can use an info window. Usually the info window is attached to a marker. You can attach an info window by instantiating the **google.maps.InfoWindow** class. It has the following properties −

- **Content** − You can pass your content in String format using this option.
- **position** − You can choose the position of the info window using this option.
- **maxWidth** − By default, the info window's width will be stretched till the text is wrapped. By specifying maxWidth, we can restrict the size of the info window horizontally.

Example

The following example shows how to set the marker and specify an info window above it −

```
<!DOCTYPE html>
<html>

  <head>
    <script src = "https://maps.googleapis.com/maps/api/js"></script>

    <script>
      function loadMap() {

        var mapOptions = {
          center:new google.maps.LatLng(17.433053, 78.412172),
          zoom:5
        }

        var map = new
google.maps.Map(document.getElementById("sample"),mapOptions);

        var marker = new google.maps.Marker({
          position: new google.maps.LatLng(17.433053, 78.412172),
          map: map,
          draggable:true,
          icon:'/scripts/img/logo-footer.png'
        });

        marker.setMap(map);
```

```
        var infowindow = new google.maps.InfoWindow({
          content:"388-A , Road no 22, Jubilee Hills, Hyderabad Telangana, INDIA-500033"
        });

        infowindow.open(map,marker);
      }
    </script>

  </head>

  <body onload = "loadMap()">
    <div id = "sample" style = "width:580px; height:400px;"></div>
  </body>

</html>
```

Adding an Event Listener

You can add an event listener using the method **addListener()**. It accepts parameters such as object name on which we want to add the listener, name of the event, and the handler event.

Example

The following example shows how to add an event listener to a marker object. The program raises the zoom value of the map by 5 each time we double-click on the marker.

```
<!DOCTYPE html>
<html>

  <head>
    <script src = "https://maps.googleapis.com/maps/api/js"></script>

    <script>
      var myCenter = new google.maps.LatLng(17.433053, 78.412172);
      function loadMap(){

        var mapProp = {
          center: myCenter,
          zoom:5,
          mapTypeId: google.maps.MapTypeId.ROADMAP
        };

        var map = new
google.maps.Map(document.getElementById("googleMap"),mapProp);

        var marker = new google.maps.Marker({
          position: myCenter,
          title:'Click to zoom'
        });
```

```
        marker.setMap(map);

        //Zoom to 7 when clicked on marker
        google.maps.event.addListener(marker,'click',function() {
          map.setZoom(9);
          map.setCenter(marker.getPosition());
        });
      }
    </script>

  </head>

  <body onload = "loadMap()">
    <div id = "googleMap" style = "width:580px; height:400px;"></div>
  </body>

</html>
```

Opening an Info Window on Click

The following code opens an info window on clicking the marker −

```
<!DOCTYPE html>
<html>

  <head>
    <script src = "https://maps.googleapis.com/maps/api/js"></script>

    <script>
      var myCenter = new google.maps.LatLng(17.433053, 78.412172);
      function loadMap(){

        var mapProp = {
          center:myCenter,
          zoom:4,
          mapTypeId:google.maps.MapTypeId.ROADMAP
        };

        var map = new
google.maps.Map(document.getElementById("googleMap"),mapProp);

        var marker = new google.maps.Marker({
          position:myCenter,
        });

        marker.setMap(map);

        var infowindow = new google.maps.InfoWindow({
          content:"Hi"
```

```
        });

        google.maps.event.addListener(marker, 'click', function() {
          infowindow.open(map,marker);
        });
      }
    </script>

  </head>

  <body onload = "loadMap()">
    <div id = "googleMap" style = "width:580px; height:400px;"></div>
  </body>

</html>
```

Removing the Listener

You can remove an existing listener using the method **removeListener()**. This method accepts the listener object, therefore we have to assign the listener to a variable and pass it to this method.

Example

The following code shows how to remove a listener −

```
<!DOCTYPE html>
<html>

  <head>
    <script src = "https://maps.googleapis.com/maps/api/js"></script>

    <script>
      var myCenter = new google.maps.LatLng(17.433053, 78.412172);
      function loadMap(){

        var mapProp = {
          center:myCenter,
          zoom:4,
          mapTypeId:google.maps.MapTypeId.ROADMAP
        };

        var map = new
google.maps.Map(document.getElementById("googleMap"),mapProp);

        var marker = new google.maps.Marker({
          position:myCenter,
        });

        marker.setMap(map);
```

```
        var infowindow = new google.maps.InfoWindow({
          content:"Hi"
        });

        var myListener = google.maps.event.addListener(marker, 'click', function() {
          infowindow.open(map,marker);
        });

        google.maps.event.removeListener(myListener);
      }
    </script>

  </head>

  <body onload = "loadMap()">
    <div id = "googleMap" style = "width:580px; height:400px;"></div>
  </body>

</html>
```