# Using Google Map Javascript API

## Introduction

This tutorial demonstrates how to build a basic Google Map using its JavaScript API.

## Objective

To create a map and add a custom marker that points to the Google India office in Bangalore.

## Summary

This tutorial contains three steps to achieve the objective:

1. [Create an HTML page](#_d5swuzt89cus).
2. [Add a map with a custom marker](#_av9sozd8akap).
3. [Authenticate the map using the API key](#_dsnphc5pqy2r).

## Create an HTML page

This section lists the steps to create an HTML page to display the map on a browser.

1. Use any text editor of your choice and add a basic HTML structure as shown here:  
   <!DOCTYPE html>

<html>

<head>

</head>

<body>

</body>

</html>

1. Add the following meta tag inside the head element to specify the character encoding for the HTML page:  
   <meta charset="UTF-8">
2. Add a title for the HTML page inside the head element.

<title>Google Maps JavaScript API Tutorial</title>

1. Add the following div element inside the body element to use it as a container for the map on the HTML page:

<div id="my\_map" style="height:900px; width:100%"></div>  
The id attribute assigns a unique ID to the div, and the style attribute sets the size of the div.

1. Save the file as *MyMap.html* and validate the content in the file. It should match the following markup:  
   <!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Google Maps JavaScript API Tutorial</title>

</head>

<body>

<div id="my\_map" style="height:900px; width:100%"></div>

</body>

</html>

## Add a map with a custom marker

This section lists the steps to create the map and add a custom marker positioned at the Google India office location in Bangalore.

1. Open *MyMap.html* in a text editor.
2. Add the following script element inside the body element to load the Google Maps JavaScript API:  
   <script async defer src="https://maps.googleapis.com/maps/api/js?key=YOUR\_API\_KEY&callback=initMap">

</script>  
Here, the script tag contains the following attributes:

* The src attribute points to the URL for loading Google Map JavaScript API.
  + The URL needs an API key (YOUR\_API\_KEY) to authenticate the access. This is explained later in the [Authenticate the map](#_dsnphc5pqy2r) section.
  + The callback parameter in the URL is used for initializing the initMap function to construct and load the map on the HTML page.
* The async and defer attributes are used to load the script on the browsers. The async attribute takes a higher priority on the modern browsers and it loads the script while the page continues the parsing. While some of the old browsers that don’t support it use the defer attribute to load the script when the page has finished parsing.

1. Add a new script element just before the script element of the previous step.  
   <script>

</script>

1. Add the initMap function inside the new script element to construct the map object based on the specified properties.  
   function initMap() {

// Properties to zoom and center the map at Bangalore, India.

var properties = {  
 zoom:13,   
 center:{lat:12.9716, lng:77.5946}

}

// Create the map object inside the div element using the defined properties.

var map = new google.maps.Map(

document.getElementById('my\_map'), properties);

}  
In the above code, the initMap function uses the following variables to construct the map:

* The properties variable declares the map zoom level and center coordinates.
* The map variable declares a new map object inside the div element using the map options set in the properties variable.

1. Add the following code inside the initMap function to create a custom marker that points to the location of Google India office in Bangalore:  
   // Add a custom marker pointing to Google India office in Bangalore.

var marker = new google.maps.Marker  
({  
 position: {lat:12.993766, lng:77.660181}, map:map, icon:'http://maps.google.com/mapfiles/kml/pal5/icon62.png'  
});  
In the above code, the marker variable declares a new marker object and uses the position property to place it on the map. It also uses the icon property to assign a custom image to the marker.

1. Save the changes and validate the content in the HTML file. It should match the following markup:  
   <!DOCTYPE html>  
   <html>  
    <head>  
    <meta charset="UTF-8">

<title>Google Maps JavaScript API Tutorial</title>

</head>

<body>

<div id="my\_map" style="height:900px; width:100%"></div>

<script>

function initMap() {

// Properties to zoom and center the map at Bangalore, India.

var properties = {  
 zoom:13,   
 center:{lat:12.9716, lng:77.5946}

}

// Create the map object using the defined properties.

var map = new google.maps.Map(

document.getElementById('my\_map'), properties);

// A custom marker pointing to Google India office in Bangalore.

var marker = new google.maps.Marker ({  
 position: {lat:12.993766, lng:77.660181}, map:map, icon:'http://maps.google.com/mapfiles/kml/pal5/icon62.png'  
 });

}

</script>

<script async defer src="https://maps.googleapis.com/maps/api/js?key=YOUR\_API\_KEY&callback=initMap">

</script>

</body>

</html>

## Authenticate the Map

This section lists the steps to generate a key to authenticate the code and gain access to Google Map JavaScript API.

1. Go to the [Google Cloud Platform Console](https://cloud.google.com/console/google/maps-apis/overview).
2. Use the project dropdown to create or select a project.
3. (Optional) Link a billing account to the project to avoid a runtime pop-up, and a ‘For development purpose only’ watermark all over the map. For more information, see [Enable billing for a new project](https://cloud.google.com/billing/docs/how-to/modify-project#enable_billing_for_an_existing_project).
4. Click the menu button, and select **API & Services** > **Library**.
5. Find the **Maps JavaScript API** and enable it.
6. Click the menu button, and select **API & Services** > **Credentials**.
7. On the **Credentials** page, click **Create Credentials** and select **API key**.  
   Note: If you are using an existing project that already contains an API key, you may use it. For more information, see [Using API Keys](https://cloud.google.com/docs/authentication/api-keys).
8. Copy the key.
9. Open *MyMap.html* in a text editor.
10. Locate the following code and replace YOUR\_API\_KEY with the copied key:  
    <script async defer src="https://maps.googleapis.com/maps/api/js?key=YOUR\_API\_KEY&callback=initMap">

</script>

1. Save and load the file in a browser.

## Result

You should see a Google Map zoomed in and centered at Bangalore with a custom marker positioned at the Google India office location in the city.

