<!DOCTYPE html>  
<html>  
  <head>  
    <style>  
       /\* Set the size of the div element that contains the map \*/  
      #map {  
        height: 400px;  /\* The height is 400 pixels \*/  
        width: 100%;  /\* The width is the width of the web page \*/  
       }  
    </style>  
  </head>  
  <body>  
    <h3>My Google Maps Demo</h3>  
    <!--The div element for the map -->  
    <div id="map"></div>  
    <script>  
// Initialize and add the map  
function initMap() {  
  // The location of Uluru  
  var uluru = {lat: -25.344, lng: 131.036};  
  // The map, centered at Uluru  
  var map = new google.maps.Map(  
      document.getElementById('map'), {zoom: 4, center: uluru});  
  // The marker, positioned at Uluru  
  var marker = new google.maps.Marker({position: uluru, map: map});  
}  
    </script>  
    <!--Load the API from the specified URL  
    \* The async attribute allows the browser to render the page while the API loads  
    \* The key parameter will contain your own API key (which is not needed for this tutorial)  
    \* The callback parameter executes the initMap() function  
    -->  
    <script async defer  
    src="https://maps.googleapis.com/maps/api/js?key=***YOUR\_API\_KEY***&callback=initMap">  
    </script>  
  </body>  
</html>

Getting started

There are three steps to creating a Google map with a marker on your web page:

1. [Create an HTML page](https://developers.google.com/maps/documentation/javascript/adding-a-google-map#page)
2. [Add a map with a marker](https://developers.google.com/maps/documentation/javascript/adding-a-google-map#map)
3. [Get an API key](https://developers.google.com/maps/documentation/javascript/adding-a-google-map#key)

You need a web browser. Choose a well-known one like Google Chrome (recommended), Firefox, Safari or Internet Explorer, based on your platform.

Step 1: Create an HTML page

Here's the code for a basic HTML web page:

<h3>My Google Maps Demo</h3>

<!--The div element for the map -->

<div id="map"></div>

#map {

width: 100%;

height: 400px;

background-color: grey;

}

<!DOCTYPE html>  
<html>  
  <head>  
    <style>  
      #map {  
        width: 100%;  
        height: 400px;  
        background-color: grey;  
      }  
    </style>  
  </head>  
  <body>  
    <h3>My Google Maps Demo</h3>  
    <!--The div element for the map -->  
    <div id="map"></div>  
  </body>  
</html>

Note that this is a very basic page with a heading level three (h3), a single div element, and a style element. You can add any content you like to the web page.

Try it yourself

At the top right corner of the sample code above are three buttons. Click the left-most button to open the sample in JSFiddle.

Understanding the code

The code below creates an HTML page consisting of a head and a body.

<html>  
 <head>  
 </head>  
 <body>  
 </body>  
</html>

You can add a heading level three above the map using the code below.

<h3>My Google Maps Demo</h3>

The code below defines an area of the page for your Google map.

<div id="map"></div>

At this stage of the tutorial, the div appears as just a grey block, because you have not yet added a map. The code below describes the CSS that sets the size and color of the div.

<style>  
 #map {  
   width: 100%;  
   height: 400px;  
   background-color: grey;  
 }  
</style>

In the above code, the style element sets the div size for your map. Set the div width and height to greater than 0px for the map to be visible. In this case, the div is set to a height of 400 pixels, and width of 100% to display the across the width of your web page. Apply background-color: grey to the div to view the area for your map on the web page.

Step 2: Add a map with a marker

This section shows you how to load the Maps JavaScript API into your web page, and how to write your own JavaScript that uses the API to add a map with a marker on it.

// Initialize and add the map

function initMap() {

// The location of Uluru

var uluru = {lat: -25.344, lng: 131.036};

// The map, centered at Uluru

var map = new google.maps.Map(

document.getElementById('map'), {zoom: 4, center: uluru});

// The marker, positioned at Uluru

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

}

<h3>My Google Maps Demo</h3>

<!--The div element for the map -->

<div id="map"></div>

/\* Set the size of the div element that contains the map \*/

#map {

height: 400px; /\* The height is 400 pixels \*/

width: 100%; /\* The width is the width of the web page \*/

}

<!-- Replace the value of the key parameter with your own API key. -->

<script async defer

src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCkUOdZ5y7hMm0yrcCQoCvLwzdM6M8s5qk&callback=initMap">

</script>

<!DOCTYPE html>  
<html>  
  <head>  
    <style>  
      /\* Set the size of the div element that contains the map \*/  
      #map {  
        height: 400px;  /\* The height is 400 pixels \*/  
        width: 100%;  /\* The width is the width of the web page \*/  
       }  
    </style>  
  </head>  
  <body>  
    <h3>My Google Maps Demo</h3>  
    <!--The div element for the map -->  
    <div id="map"></div>  
    <script>  
// Initialize and add the map  
function initMap() {  
  // The location of Uluru  
  var uluru = {lat: -25.344, lng: 131.036};  
  // The map, centered at Uluru  
  var map = new google.maps.Map(  
      document.getElementById('map'), {zoom: 4, center: uluru});  
  // The marker, positioned at Uluru  
  var marker = new google.maps.Marker({position: uluru, map: map});  
}  
    </script>  
    <script async defer  
    src="https://maps.googleapis.com/maps/api/js?key=***YOUR\_API\_KEY***&callback=initMap">  
    </script>  
  </body>  
</html>

Try it yourself

At the top right corner of the sample code above are three buttons. Click the left-most button to open the sample in JSFiddle.

Understanding the code

Notice that the above sample no longer contains the CSS that colors the div grey. This is because the div now contains a map.

In the code below, the script loads the API from the specified URL.

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

In the above code, the callback parameter executes the initMap function after the API loads. The async attribute allows the browser to continue rendering the rest of your page while the API loads. The key parameter contains your API key. You don't need your own API key when [experimenting with this tutorial in JSFiddle](https://developers.google.com/maps/documentation/javascript/adding-a-google-map#jsFiddle).  
See [Step 3: Get an API key](https://developers.google.com/maps/documentation/javascript/adding-a-google-map#key) for instructions on getting your own API key later.

The code below contains the initMap function that initializes and adds the map when the web page loads. Use a script tag to include your own JavaScript which contains the initMap function.

<script>  
  function initMap() {  
  }  
</script>

Add the getElementById() function to find the map div on the web page.

The code below constructs a new Google maps object, and adds properties to the map including the center and zoom level. See the documentation for [other property options](https://developers.google.com/maps/documentation/javascript/reference#MapOptions).

{  
  var uluru = {lat: -25.344, lng: 131.036};  
  
  var map = new google.maps.Map(document.getElementById('map'), {  
    zoom: 4,  
    center: uluru  
  });  
  
}

In the above code, new google.maps.Map() creates a new Google maps object. The center property tells the API where to center the map. The map coordinates are set in the order: **latitude**, **longitude**.  
Learn more about [getting latitude/longitude coordinates, or converting an address into geographical coordinates.](https://developers.google.com/maps/documentation/javascript/adding-a-google-map#get_latLng)

The zoom property specifies the zoom level for the map. Zoom: 0 is the lowest zoom, and displays the entire earth. Set the zoom value higher to zoom in to the earth at higher resolutions.

The code below puts a marker on the map. The position property sets the position of the marker.

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

Step 3: Get an API key

This section explains how to authenticate your app to the Maps JavaScript API using your own API key.

Follow these steps to get an API key:

1. Go to the [Google Cloud Platform Console](https://cloud.google.com/console/google/maps-apis/overview).
2. Create or select a project.
3. Click **Continue** to enable the API and any related services.
4. On the **Credentials** page, get an **API key** (and set the API key restrictions). Note: If you have an existing unrestricted API key, or a key with browser restrictions, you may use that key.
5. To prevent quota theft and secure your API key, see [Using API Keys](https://cloud.google.com/docs/authentication/api-keys).
6. (Optional) Enable billing. See [Usage Limits](https://developers.google.com/maps/documentation/javascript/usage) for more information.
7. Copy the entire code of this tutorial from this page, to your text editor. If you don't already have a text editor, here are some recommendations: You can use: Notepad++ (for Windows); TextEdit (for macOS); gedit, KWrite, among others (for Linux machines).
8. Replace the value of the key parameter in the URL with your own API key (that's the API key that you've just obtained).

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

1. Save this file with a name that ends with **.html**, like **google-maps.html**.
2. Load the HTML file in a web browser by dragging it from your desktop onto your browser. Alternatively, double-clicking the file works on most operating systems.

Tips and troubleshooting

**Tip:** Check out the [Store Locator solution](https://developers.google.com/maps/solutions/store-locator) to see more comprehensive examples of using custom markers to visualize data on a map.

* Use the JSFiddle interface to display HTML, CSS and JavaScript code in separate panes. You can run the code and display output in the **Results** pane.
* You can tweak options like style and properties to customize the map. For more information on customizing maps, read the guides to [styling](https://developers.google.com/maps/documentation/javascript/styling), and [drawing on the map](https://developers.google.com/maps/documentation/javascript/overlays).
* Use the **Developer Tools Console** in your web browser to test and run your code, read error reports and solve problems with your code.
* Use the following keyboard shortcuts to open the console in Chrome:  
  Command+Option+J (on Mac), or Control+Shift+J (on Windows).
* Follow the steps below to get the latitude and longitude coordinates for a location on Google Maps.
  1. Open Google Maps in a browser.
  2. Right-click the exact location on the map for which you require coordinates.
  3. Select **What's here** from the context menu that appears. The map displays a card at the bottom of the screen. Find the latitude and longitude coordinates in the last row of the card.
* You can convert an address into latitude and longitude coordinates using the Geocoding service. The developer guides provide detailed information on [getting started with the Geocoding service](https://developers.google.com/maps/documentation/javascript/geocoding#GetStarted).

*Except as otherwise noted, the content of this page is licensed under the*[*Creative Commons Attribution 3.0 License*](https://creativecommons.org/licenses/by/3.0/)*, and code samples are licensed under the*[*Apache 2.0 License*](https://www.apache.org/licenses/LICENSE-2.0)*. For details, see our*[*Site Policies*](https://developers.google.com/terms/site-policies)*. Java is a registered trademark of Oracle and/or its affiliates.*

*Last updated September 25, 2018.*

* [Stack Overflow](http://stackoverflow.com/questions/tagged/google-maps)

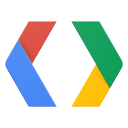
Ask a question under the google-maps tag.

* [GitHub](https://github.com/googlemaps/)

Fork our samples and try them yourself.

* [[](https://googlegeodevelopers.blogspot.com/)Blog](https://googlegeodevelopers.blogspot.com/)

Read about the latest updates, customer stories, and tips.

* [[](https://issuetracker.google.com/issues/new?component=188853&template=788207)Issue Tracker](https://issuetracker.google.com/issues/new?component=188853&template=788207)

Something wrong? Send us a bug report!

* Learn More
  + [FAQ](https://developers.google.com/maps/faq)
  + [API Picker](https://developers.google.com/maps/documentation/api-picker)
  + [Tutorials](https://developers.google.com/maps/documentation/javascript/tutorials/)
* Platforms
  + [Android](https://developers.google.com/maps/apis-by-platform#android)
  + [iOS](https://developers.google.com/maps/apis-by-platform#ios)
  + [Web](https://developers.google.com/maps/apis-by-platform#web_apis)
  + [Web Services](https://developers.google.com/maps/apis-by-platform#web_service_apis)
* Product Info
  + [Pricing and Plans](https://developers.google.com/maps/pricing-and-plans/)
  + [Contact Sales](https://developers.google.com/maps/contact-sales/)
  + [Support](https://developers.google.com/maps/support/)
  + [Terms of Service](https://cloud.google.com/maps-platform/terms)
* [](https://developers.google.com/)
  + [Android](https://developer.android.com/index.html)
  + [Chrome](https://developer.chrome.com/home)
  + [Firebase](https://firebase.google.com/)
  + [Google Cloud Platform](https://cloud.google.com/)
  + [All Products](https://developers.google.com/products/)

Top of Form

             Bahasa Indonesia                        Deutsch                        English                        español                        français                        Português Brasileiro                        Русский                        日本語                        简体中文                        한국어           

Bottom of Form