

Google Maps JavaScript API v3

Carlo Taticchi e Ivan Mercanti

developer's guide

[https://developers.google.com/maps/documentation/
javascript/tutorial](https://developers.google.com/maps/documentation/javascript/tutorial)

1. Load the Google API

```
<script src="http://maps.googleapis.com/maps/api/js"></script>
```

The Google Maps API is a JavaScript library!

2. Create a Map Container

In the HTML, create a `<div>` element to hold the map

```
<div id="googleMap"></div>
```

You can use CSS to size the element!

3. Create the map

In a blank javascript document,
declare a map object

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

The code above creates a new map inside the <div> element with id="googleMap", using the parameters that are passed (**mapProp**).

now we have to create the object **mapProp** to define the properties for the map:

```
var mapProp = {  
    center:new google.maps.LatLng(43.1107009, 12.3891720),  
    zoom: 7,  
    mapTypeId: google.maps.MapTypeId.ROADMAP  
};
```

The **mapTypeId** property specifies the map type to display.
The following map types are supported:

- ROADMAP (normal, default 2D map)
- SATELLITE (photographic map)
- HYBRID (photographic map + roads and city names)
- TERRAIN (map with mountains, rivers, etc.)

4. Create the initialize function

this is the function to initialize the map

```
function initialize() {  
  
}
```

5. Put all together

put **mapProp** and **map** inside the **initialize()** function

```
function initialize() {  
  
    var mapProp = {  
        center:new google.maps.LatLng(43.1107009, 12.3891720),  
        zoom: 7,  
        mapTypeId: google.maps.MapTypeId.ROADMAP  
    };  
  
    var map = new google.maps.Map(document.getElementById("googleMap"), mapProp);  
  
}
```


6. Add an Event Listener to Load the Map

this is a DOM listener that will execute the **initialize()** function when the page is loaded

```
google.maps.event.addDomListener(window, 'load', initialize);
```

NB: the listener is declared after the **initialize()** function, not inside it!

7. Style the map

create a stylesheet and add the following lines

```
#googleMap {  
    width: 60%;  
    height: 60%;  
}
```

It is mandatory to specify width and height of the map!

8. Link script and css in the HTML

```
<script src="js/mapscript.js"></script>  
<link rel="stylesheet" href="css/mapstyle.css" type="text/css"/>
```

Done! Now try to open the page and see the result!



Google API Key

Google allows your web site to call any Google API, many thousand times per day. If you plan for heavier traffic, you should get a free API key from Google.

Go to <https://console.developers.google.com> to get a free key.

Google Maps expects to find the API key in the **key** parameter when loading an API:

```
<script src="http://maps.googleapis.com/maps/api/js?key=YOUR_KEY"></script>
```

Directions Service

1. Add the following lines inside the **initialize()** function

```
directionsService = new google.maps.DirectionsService();  
  
var rendererOptions = { map: map }  
  
directionsDisplay = new google.maps.DirectionsRenderer(rendererOptions);  
  
stepDisplay = new google.maps.InfoWindow();  
  
calcRoute();
```

This will initiate the direction service and create a render to display the path on the map

2. Create the **calcRoute()** function

```
function calcRoute() {  
  
    // First, remove any existing markers from the map.  
    for (var i = 0; i < markerArray.length; i++) {  
        markerArray[i].setMap(null);  
    }  
  
    // Now, clear the array itself.  
    markerArray = [];  
}
```

...continue

■ ■ ■

```
// Retrieve the start and end locations and create
// a DirectionsRequest using WALKING directions.
var start = "Stadio Renato Curi, Viale Pietro Conti, 06125 Perugia, Italia"
var end = "Pretola, PG, Italia";
var request = {
    origin: start,
    destination: end,
    travelMode: google.maps.TravelMode.DRIVING
};
```

...continue

■ ■ ■

```
// Route the directions and pass the response to a
// function to create markers for each step.
directionsService.route(request, function(response, status) {
    if (status == google.maps.DirectionsStatus.OK) {
        directionsDisplay.setDirections(response);
        showSteps(response);
    }
});
}
```

end of **calcRoute()**

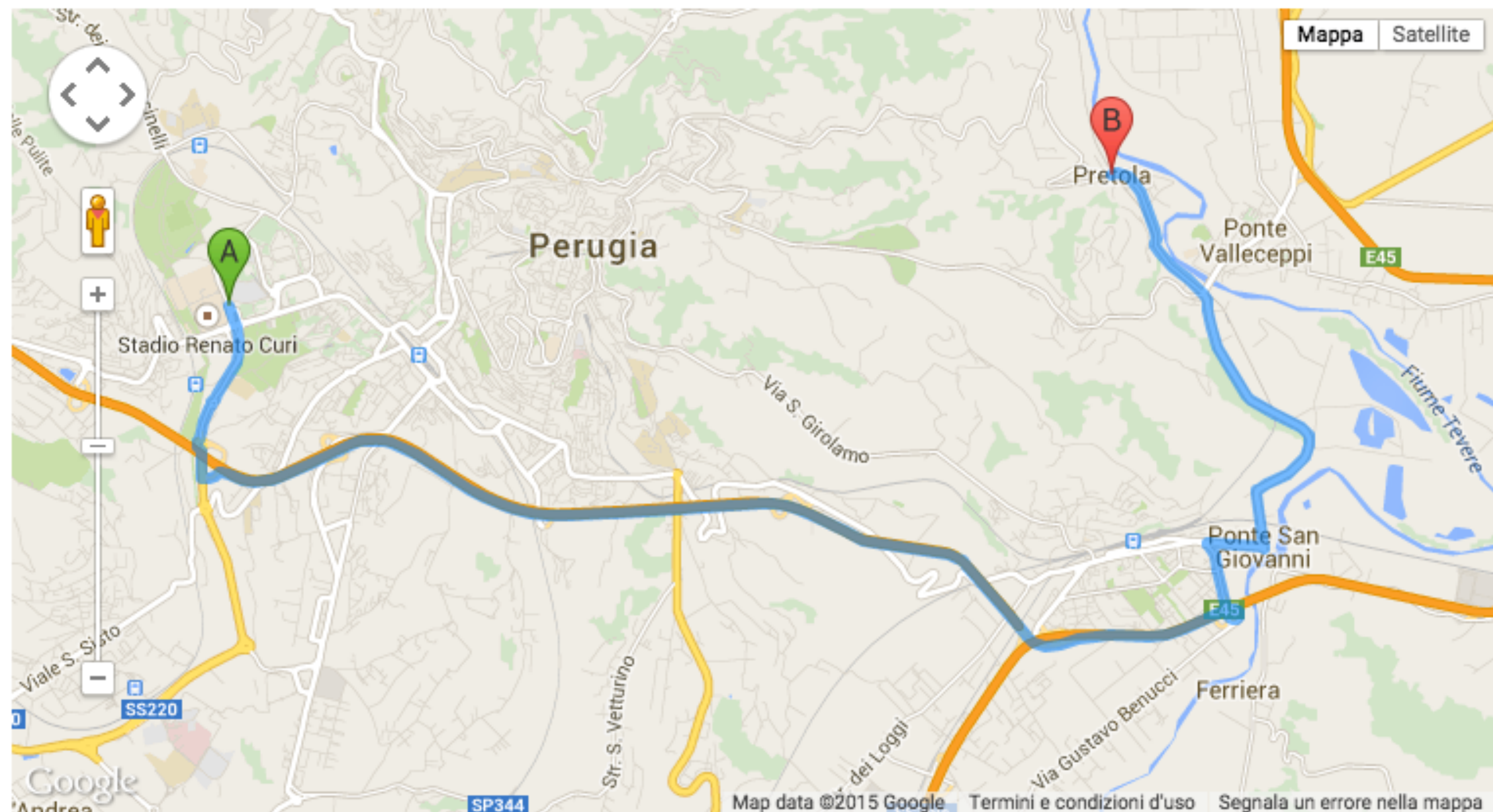
3. Create the **showSteps()** function

```
function showSteps(directionResult) {  
    // For each step, place a marker, and add the text to the marker's  
    // info window. Also attach the marker to an array so we  
    // can keep track of it and remove it when calculating new  
    // routes.  
    var myRoute = directionResult.routes[0].legs[0];  
  
    for (var i = 0; i < myRoute.steps.length; i++) {  
        var marker = new google.maps.Marker({  
            position: myRoute.steps[i].start_location,  
            map: map  
        });  
        attachInstructionText(marker, myRoute.steps[i].instructions);  
        markerArray[i] = marker;  
    }  
}
```

4. Create **attachInstructionText()** function

```
function attachInstructionText(marker, text) {  
  google.maps.event.addListener(marker, 'click', function() {  
    // Open an info window when the marker is clicked on,  
    // containing the text of the step.  
    stepDisplay.setContent(text);  
    stepDisplay.open(map, marker);  
  });  
}
```

put the DOM Listener after this function



The End