1 GoogleMap

Definitely, Google map is one of popular web applications nowaday and better than Apple map.

Since Aug 2018, Google has changes the google map development policy, no more free policy: developers have to register an account, binging with ones' credit card. This causes trouble for map developers. We try to overcome this problem by replacing GoogleMap by OpenStreetMap, OPenLayer. And integrated with the help of html geolocation, we try to recreate the map app again. (2019/05)

- [Howto](): basic requirement of using Goole Map App
- [Template](): A simple demo of app using Google Map
- [Mark the Position](): anchor the position
- [Marker]()
- [Demo](); Demo of Marker Draggable
- [How Far from Here]() Distance Measurement
- [Area Estimation]() Estimate the Area of polygon
- [Make Note]() List of Locations
- [Map Animating]() Create a Marauders-Maps
- [OpenMap]() Map app comes back!

1.1 Howto

To use the Google Map service, there are some standard procedures to do as folows:

1. **Use javascript library of the Google Map API's**

```
<script type="text/javascript"
    src="http://maps.googleapis.com/maps/api/js?sensor=true&language=tw&v=3" >
</script>
```

option "sensor=true": use mobile device

2. **Initialize the service**

```
function initialize() {
    // Options of Map
    // center position, rate of magnitude, and type of maps given
    var mapOptions = {
        center: new google.maps.LatLng(25.034264,121.389395),
        zoom: 16,
        mapTypeId: google.maps.MapTypeId.ROADP
    };
    // location at which the map is displayed
    var map = new google.maps.Map(document.getElementById("map_canvas"),
        mapOptions);
}
```

3. **Run the code while loaded**

```
<body onload="initialize()">
<div id="map_canvas" style="width: 600px;height: 400px;" />
```

1.2 Note

Here, type of maps is ROADP; other formats are

- MapTypeId.SATELLITE
- MapTypeId.HYBRID
- MapTypeId.TERRAIN
- MapTypeId.ROADMAP, defaulted.

1.3 Template

A simple demo:

```html
<html>
<head>
<script type="text/javascript"
        src="http://maps.googleapis.com/maps/api/js?sensor=true&
language=tw&v=3" >
</script>

<script type="text/javascript">
        function initialize() {
          var mapOptions = {
            center: new google.maps.LatLng(25.034264,121.389395)
,
            zoom: 16,
            mapTypeId: google.maps.MapTypeId.ROADP
          };
          var map = new google.maps.Map(document.getElementById(
"map"),
                mapOptions);
        }

</script>
</head>
<body onload="initialize()">
      <div id="map" style="width: 600px;height: 400px;"/>
</body>
</html>
```

1.4 CSSModification

Except given the option of size of "map_canvas", we can aslo use CSS to set the size of Canvas of Map

```css
<style>
    html, body, #map {
        height: 100%;
        width: 100%;
        margin: 0px;
        padding: 0px;
    }
</style>
```

The app size should be resized according to user's necessary.

1.5 MarkThePosition

1. define the the latitude and lonitude of given position:

```
var CGU_latlng = new google.maps.LatLng(25.034264,121
.389395);
```

2. create marker:

```
var marker = new google.maps.Marker({
    position: CGU_latlng,
    map: gmap,
    title:"Chang-gung University"
});
```

1.6 Marker

```html
<script type="text/javascript">
    window.onload = function () {
        // initialize Google Map
        var latlng = new google.maps.LatLng(25.034264,121.389395
);
        var mapOptions = {
            zoom:12,
            center:latlng,
            mapTypeId: google.maps.MapTypeId.ROADMAP
        };
        var gmap = new google.maps.Map(document.getElementById("
map_canvas"), mapOptions);
        // Show Mark
        var CGU_latlng = new google.maps.LatLng(25.034264,121.38
9395);
        var marker = new google.maps.Marker({
            position: CGU_latlng,
            map: gmap,
            title:"Chang-gung University"
        });
    };
</script>
<body>
  <div id="map_canvas" style="width: 600px;height: 400px;" />
</body>
```

1.7 MarkerDraggable

1. Show the marker at defaulted position while loading;
2. Use mouse to drag the marker;
3. show the **new** Latitude and longitude of the position at which the mark was placed.

1.8 Basic HTML

Create a block to display the lat-long of poisition:

```html
<div id="map_canvas" style="width: 600px;height: 480px;"><
/div><br />
    <label for="latitude">Latitude:</label>
    <input id="latitude" type="text" value="" />
    <label for="longitude">Longitude:</label>
    <input id="longitude" type="text" value="" />
```

This should create the input columns as follows:

```
""" Google Map Here """
```

Latitude: [   ]   Longitude: [   ]

1.9 JavaScript part

"HTML PART" Here....

```javascript
<script type="text/javascript">
  var myCoordsLenght = 6;
  var defaultLat = 25.034264;
  var defaultLng = 121.389395;

  function initialize() {
     var mapOptions = {
        ...
     };
     var map = new google.maps.Map(document.getElementById
("map_canvas"),mapOptions);

     // creates a draggable marker to the given coords
     var myMarker = new google.maps.Marker({
        ...
         draggable: true
     });

     google.maps.event.addListener(myMarker, 'dragend', fu
nction(evt){
         document.getElementById('latitude').value = evt.l
atLng.lat();
         document.getElementById('longitude').value = evt.
latLng.lng().toFixed(myCoordsLenght);
     });

     // centers the map on markers coords
     map.setCenter(myMarker.position);

     // adds the marker on the map
     myMarker.setMap(map);
  }

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

1.10 Completed Codes

```html
<html lang="en">
<head>
   <meta charset="utf-8" />
   <script type="text/javascript" src="http://maps.google.com/
maps/api/js?sensor=true"></script>
</head>
<body>
    <div id="map_canvas" style="width: 600px;height: 400px;"><
/div><br />
   <label for="latitude">Latitude:</label>
   <input id="latitude" type="text" value="" />
   <label for="longitude">Longitude:</label>
   <input id="longitude" type="text" value="" />
<script type="text/javascript">
  var myCoordsLenght = 6;
  var defaultLat = 25.034264;
  var defaultLng = 121.389395;

  function initialize() {
    var mapOptions = {
        center: new google.maps.LatLng(defaultLat,defaultLng)
,
        zoom: 16,
        mapTypeId: google.maps.MapTypeId.ROADP
    };
    var map = new google.maps.Map(document.getElementById("ma
p_canvas"),mapOptions);

    var myMarker = new google.maps.Marker({
        position: new google.maps.LatLng(defaultLat, defaultL
ng),
        draggable: true
    });

    google.maps.event.addListener(myMarker, 'dragend', functi
on(evt){
        document.getElementById('latitude').value = evt.latLn
g.lat();
        document.getElementById('longitude').value = evt.latL
ng.lng().toFixed(myCoordsLenght);
    });

    map.setCenter(myMarker.position);
    myMarker.setMap(map);
  }
  google.maps.event.addDomListener(window, 'load', initialize)
;
</script>
</body></html>
```
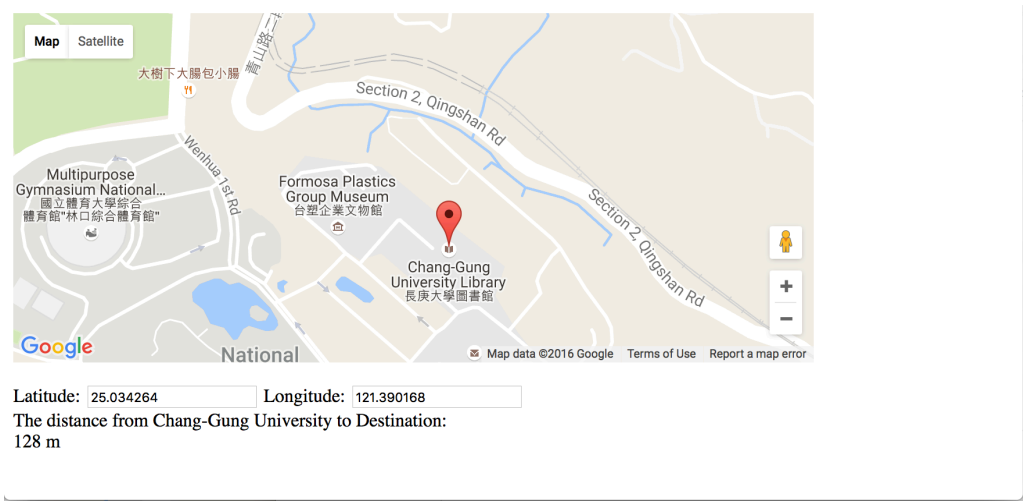
Latitude: 25.034264   Longitude: 121.390168

The distance from Chang-Gung University to Destination:
128 m

1.11 Application

- Make Survey (get data in csv format)
- make map of survey data (by scratch or by Python)

```
-----------------------------------------------------------
-------------------
NameError                                 Traceback (mos
t recent call last)
<ipython-input-3-421e654598d9> in <module>
----> 1 IFrame(src="ntufolium.html", width="800px",
height="500px" )

NameError: name 'IFrame' is not defined
```
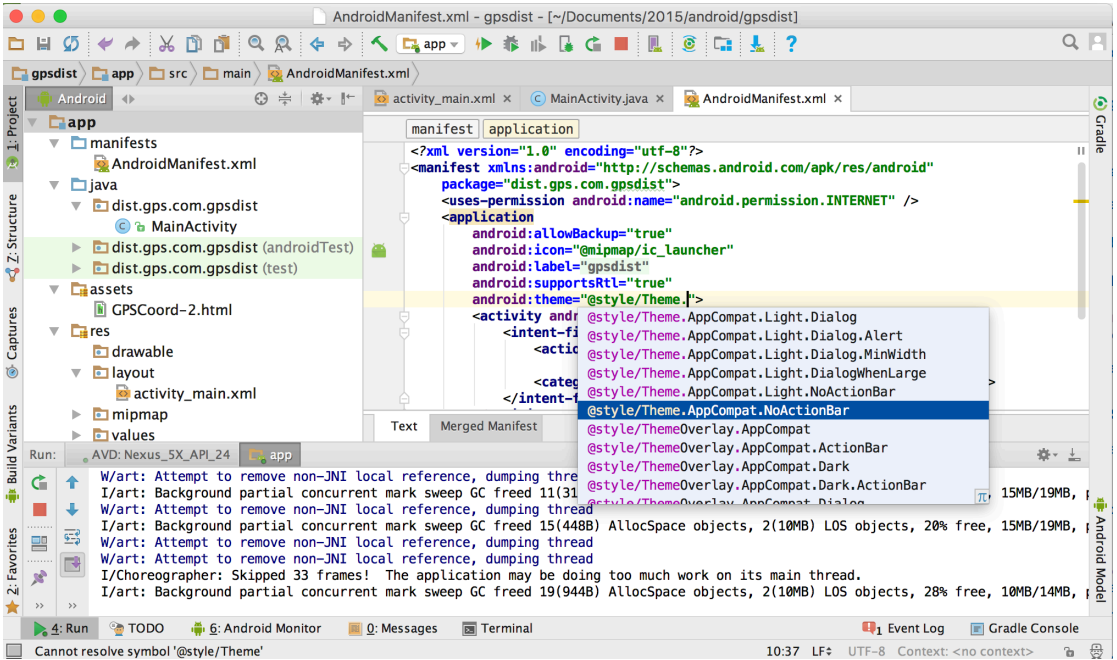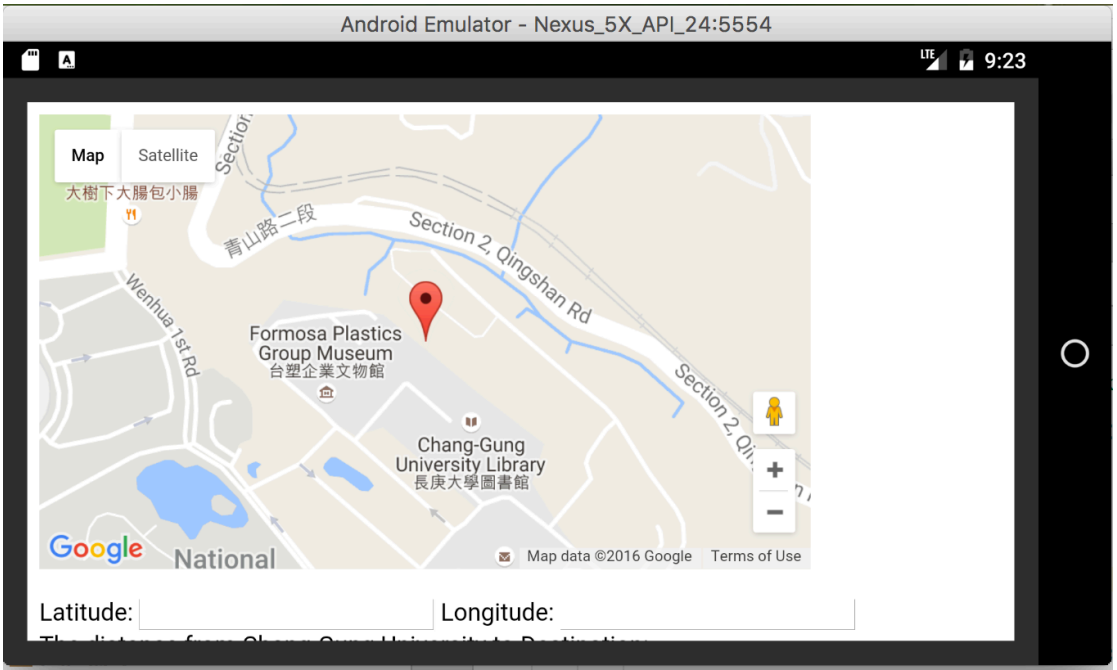
1.12 Note

1. How to get rid of app title bar?

So small to display the unnecessary info. Modify the default setting of Theme in
AndroidManifest.xml:
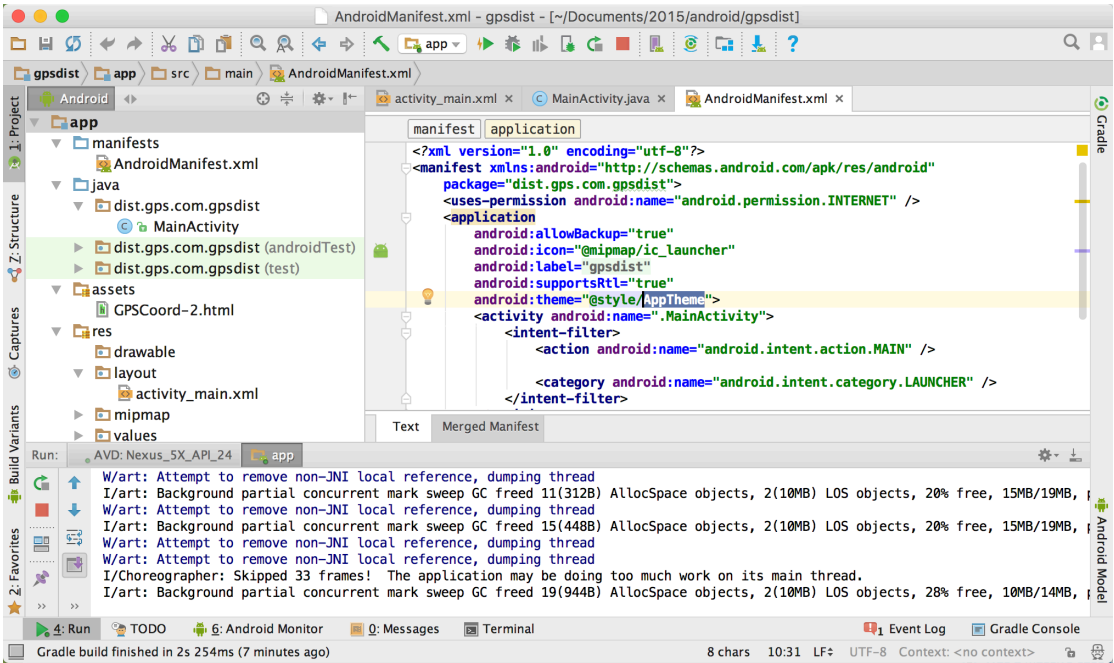
1.13 The result

1.14 DistanceMeasurement

1. To access the function of distance measurement requires geometry libarary:

```
<head>
  <script type="text/javascript"
      src="http://maps.google.com/maps/api/js?sensor=true&v=3&libraries=geometry"></script>
</head>
```

2. where the measurement is placed:

```
<label>
    The distance from Chang-Gung University to Destination: <div id="distanceAB"></div>
<label>
```

3. calculate the distance, set new coordinates, then measure by **"google.maps.geometry.spherical.computeDistanceBetween()"**:

```
google.maps.event.addListener(myMarker, 'dragend', function(evt){
    var newLat=evt.latLng.lat();
    var newLng=evt.latLng.lng().toFixed(myCoordsLenght);
    document.getElementById('latitude').value = newLat;
    document.getElementById('longitude').value = newLng;

    var loc2 = new google.maps.LatLng(newLat, newLng);
    document.getElementById('distanceAB').innerHTML =
        Math.round(google.maps.geometry.spherical.computeDistanceBetween (loc1, loc2))+' m';
});
```

1.15 Sketch

```
                    HTML
            defaulted Lat/Lng
  JavaScript


       Latitude: (defaultLat)  Longitude: (defaultLng)    google
   .maps.event.addListener(,,function(evt))
                    |                    |

   :
                    |                    |
   :
                    ▼                    ▼
   :
                  newLat              newLng
   :    waiting for position changed
                    ▲                    ▲
   :
                    |                    |
   :
              evt.latLng.lat()  ◄-..  evt.latLng.lng()  ◄--------
   ---- ┤     If dragged


   :


   :
            <div id="distanceAB">     </div>
   :
                                        ▲
   :
                                  └ _____
   _____┘


   google.maps.geometry.spherical.computeDistanceBetween(loc1,loc
   2)
```

1.15.1 Make a note via Google Map

[Codes, GPSCoord-3.html (GPSCoord-3.html)](GPSCoord-3.html)

Create an arraay of latitude/longitude list for which we are interested. Move the marker to the place which we select from the HTML options.

- create the html options
  ```html Library

  __
  " ```

Result:

Library

__

- Array in Javascript

```
var loctoLib=[[25.035236, 121.389524],[25.034225,121.390
168]];
var loctoPO=[[25.035236, 121.389524],[25.032514, 121.390
66]];
var loctoNSU=[[25.035236,121.389524],[25.034750,121.3892
45],[25.033991,121.388494],[25.034147,121.388237], [25.033
350,121.387013],[25.032125,121.388022],[25.0317170,121.387
464],
          [25.0319892,121.386692]];
```

- show the marker at the place on the map while option was selected (i.e.
  `[Element]`.onchange):

```
document.getElementById('PosMenu').onchange = function() {
    var index = this.value;

    var loc2 = new google.maps.LatLng(loc[index][0], loc[in
dex][1]);
    document.getElementById('distanceAC').innerHTML =
        Math.round(google.maps.geometry.spherical.computeDis
tanceBetween (loc1, loc2))+'  m';
    var newMarker = new google.maps.Marker({
        position: new google.maps.LatLng(loc[index][0], loc[
index][1]),
        draggable: myMarkerIsDraggable
    });

    // centers the map on markers coords
    var mapOptions = {
        ...
    };

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

    google.maps.event.addListener(newMarker, 'dragend', fun
ction(evt){
        var newLat=evt.latLng.lat();
        var newLng=evt.latLng.lng().toFixed(myCoordsLenght);
        document.getElementById('latitude').value = newLat;
        document.getElementById('longitude').value = newLng;
        var loc2 = new google.maps.LatLng(newLat, newLng);
        document.getElementById('distanceAB').innerHTML =
        Math.round(google.maps.geometry.spherical.computeDis
tanceBetween (loc1, loc2))+'  m';
    });

    map.setCenter(newMarker.position);
    newMarker.setMap(map)
}
```

- create the chosen tracjectory

```
        if (index==0) {
            var coord =[];
            for (i = 0; i < loctoLib.length; i++) {
                coord.push(new google.maps.LatLng(loctoLib[i]
    [0], loctoLib[i][1]));
            }
        } else if (index==1) {
            var coord =[];
            for (i = 0; i < loctoNSU.length; i++) {
                coord.push(new google.maps.LatLng(loctoNSU[i][
    0], loctoNSU[i][1]));
            }
        } else {
            var coord =[];
            for (i = 0; i < loctoPO.length; i++) {
                coord.push(new google.maps.LatLng(loctoPO[i][0]
    , loctoPO[i][1]));
            }
        }
```

1.16 PositionMarker

```
<script type="text/javascript">
window.onload = function () {
    var latlng = new google.maps.LatLng(25.034264,121.389395);
    var mapOptions = {
        zoom:12,
        center:latlng,
        mapTypeId: google.maps.MapTypeId.ROADMAP
    };
    var gmap = new google.maps.Map($("map_canvas"), mapOptions
);

    var Coordinates = [
        new google.maps.LatLng(25.034264,121.389395),
        new google.maps.LatLng(25.034264,121.391395),
        new google.maps.LatLng(25.036264,121.391395),
    ];
    var flightPath = new google.maps.Polyline({
        path: Coordinates,
        strokeColor: "#FF0000",
        strokeOpacity: 1.0,
        strokeWeight: 3,
        map: gmap
    });
};
</script>
</head><body>
<div id="map_canvas" />
```

- Make the trajectory:

```javascript
var TrajPath= new google.maps.Polyline({
    path: coord,
    geodesic: true,
    strokeColor: '#FF0000',
    strokeOpacity: 0.8,
    strokeWeight: 2
});
TrajPath.setMap(map);
```

1.17 Complete Code

Show the distance of destiny positions

```html
<html lang="en">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width" />
    <title></title>
    <script type="text/javascript"
        src="http://maps.google.com/maps/api/js?sensor=true&v=3
&libraries=geometry"></script>

<style type="text/css">
    body {
        margin: 10;
        padding: 10
    }
    #map_canvas {
        position: absolute;
        width: 60%;
        height: 60%;
        left:20%;
        right:20%;
        top:30%;
        overflow: auto
    }
</style>
</head>

<body>

    <div id="map_canvas"></div>
    <br />
    <label for="latitude">Latitude:</label>
    <input id="latitude" type="text" value="" />
    <label for="longitude">Longitude:</label>bottom
    <input id="longitude" type="text" value="" />

    <br><label>
    The distance from Chang-Gung University to Destination: <d
iv id="distanceAB"></div>
```

```html
<label>
<br>
   The distance from Chang-Gung University to Destination
   <select id="PosMenu">
        <option value="0">library</option>
        <option value="1">NSU</option>
        <option value="2">PO</option>
   </select>
  <div id="distanceAC"></div>

<script type="text/javascript">

  var myZoom = 16;
  var myMarkerIsDraggable = true;
  var myCoordsLenght = 6;
  var defaultLat = 25.035255529260443;
  var defaultLng = 121.389524;
  var loc1 = new google.maps.LatLng(25.035255529260443, 121.389524);
  var loc = [[25.034225,121.390168],[25.032047,121.386692],[25.032514, 121.390661]];

  function initialize() {
    var mapOptions = {
        center: new google.maps.LatLng(defaultLat,defaultLng),
        zoom: myZoom,
        mapTypeId: google.maps.MapTypeId.ROADP
    };
    var map = new google.maps.Map(document.getElementById("map_canvas"),mapOptions);
    // creates a draggable marker to the given coords    -3.118 -3.118

    var myMarker = new google.maps.Marker({
    position: new google.maps.LatLng(defaultLat, defaultLng),
    draggable: myMarkerIsDraggable
    });


    google.maps.event.addListener(myMarker, 'dragend', function(evt){
        var newLat=evt.latLng.lat();
        var newLng=evt.latLng.lng().toFixed(myCoordsLenght);
    document.getElementById('latitude').value = newLat;
    document.getElementById('longitude').value = newLng;
    var loc2 = new google.maps.LatLng(newLat, newLng);
    document.getElementById('distanceAB').innerHTML =
        Math.round(google.maps.geometry.spherical.computeDistanceBetween (loc1, loc2))+'  m';

    });

    // centers the map on markers coords
    map.setCenter(myMarker.position);
```

```javascript
    // adds the marker on the map
    myMarker.setMap(map);
  }

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

  document.getElementById('PosMenu').onchange = function() {
    var index = this.value; // array indices start at 0
    //alert(loc[index][0]);
    var loc2 = new google.maps.LatLng(loc[index][0], loc[index
][1]);
    document.getElementById('distanceAC').innerHTML =
    Math.round(google.maps.geometry.spherical.computeDistanceB
etween (loc1, loc2))+'  m';
    var newMarker = new google.maps.Marker({
        position: new google.maps.LatLng(loc[index][0], loc[in
dex][1]),
        draggable: myMarkerIsDraggable
    });

    // centers the map on markers coords
    var mapOptions = {
        center: new google.maps.LatLng(defaultLat,defaultLng
),
        zoom: myZoom,
        mapTypeId: google.maps.MapTypeId.ROADP
    };

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

    google.maps.event.addListener(newMarker, 'dragend', functi
on(evt){
        var newLat=evt.latLng.lat();
        var newLng=evt.latLng.lng().toFixed(myCoordsLenght);
        document.getElementById('latitude').value = newLat;
        document.getElementById('longitude').value = newLng;
        var loc2 = new google.maps.LatLng(newLat, newLng);
        document.getElementById('distanceAB').innerHTML =
          Math.round(google.maps.geometry.spherical.computeDis
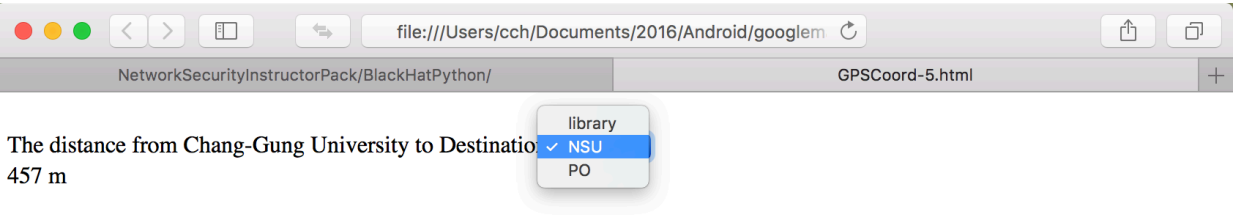tanceBetween (loc1, loc2))+'  m';
    });

    map.setCenter(newMarker.position);

    // adds the marker on the map
    newMarker.setMap(map)
  }

</script>

</body>
</html>
```

1.18 Result

1.19 AreaEstimation

The steps to solve the last practice, mark a region:

1. Find the the (latitude, longitude)'s of defaulted building;
2. make a closed polygon formed by set of given points, found by above;
3. calculate the area of polygon above.

1.20 code1-1

```html
<head>
    <meta charset="utf-8" />
    <title>Laditude and Longitude</title>
    <script type="text/javascript" src="http://maps.google.co
m/maps/api/js?sensor=true" >
    </script>
  </head>
```

1.21 code1-2

```html
<body>
   <div id="map_canvas" style="width: 600px;height: 400px;"></
div><br>
   <label for="latitude">Latitude:</label>
   <input id="latitude" type="text" value="" />
   <label for="longitude">Longitude:</label>
   <input id="longitude" type="text" value="" />
<script type="text/javascript">
  var myZoom = 16;
  var myMarkerIsDraggable = true;
  var myCoordsLenght = 6;
  var defaultLat = 25.034264;
  var defaultLng = 121.389395;

  function initialize() {
    var mapOptions = {
         center: new google.maps.LatLng(defaultLat,defaultLng
),
         zoom: myZoom,
         mapTypeId: google.maps.MapTypeId.ROADP
    };
    var map = new google.maps.Map(document.getElementById("map
_canvas"),mapOptions);

    // creates a draggable marker to the given coords
    var myMarker = new google.maps.Marker({
    position: new google.maps.LatLng(defaultLat, defaultLng),
    draggable: myMarkerIsDraggable
    });

    google.maps.event.addListener(myMarker, 'dragend', functio
n(evt){
    document.getElementById('latitude').value = evt.latLng.lat
().toFixed(myCoordsLenght);
    document.getElementById('longitude').value = evt.latLng.ln
g().toFixed(myCoordsLenght);
    });

    // centers the map on markers coords
    map.setCenter(myMarker.position);

    // adds the marker on the map
    myMarker.setMap(map);
  }

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

1.22 Code 2-1

```
<script>
  // This example creates a simple polygon representing the li
brary building in CGU .
  var myZoom = 16;
  //var myMarkerIsDraggable = true;
  //var myCoordsLenght = 6;
  var defaultLat = 25.034264;
  var defaultLng = 121.389395;

  function initialize() {
    var mapOptions = {
        center: new google.maps.LatLng(defaultLat,defaultLng
),
        zoom: myZoom,
        mapTypeId: google.maps.MapTypeId.TERRAIN
    };
    var map = new google.maps.Map(document.getElementById('map
_canvas'),mapOptions);

    // Define the LatLng coordinates for the polygon's path.
    var PolygonCoords = [
        new google.maps.LatLng(25.034200,121.390527),
        new google.maps.LatLng(25.034020,121.390790),
        new google.maps.LatLng(25.033413,121.390237),
        new google.maps.LatLng(25.033612,121.390001),
        new google.maps.LatLng(25.034200,121.390527)
    ];
    // Construct thepolygon.
    var myPolygon;
    myPolygon= new google.maps.Polygon({
      paths: PolygonCoords,
      strokeColor: '#FF0000',
      strokeOpacity: 0.8,
      strokeWeight: 2,
      fillColor: '#FF0000',
      fillOpacity: 0.35
    });
    myPolygon.setMap(map);
  }
  google.maps.event.addDomListener(window, 'load', initialize)
;
</script>
```

1.23 Code2-2

```
<body>
    <div id="map_canvas" style="width: 600px;height: 400px;"><
/div>
</body>
```

1.24 Code3-1

```html
<script type="text/javascript"
    src="https://maps.googleapis.com/maps/api/js?libraries=geometry&sensor=false" >
</script>
<script type="text/javascript">
    var myZoom = 16;
    var myMarkerIsDraggable = true;
    var myCoordsLenght = 6;
    var defaultLat = 25.034264;
    var defaultLng = 121.389395;
    var map;
    function initialize(){
        var mapOptions = {
          center: new google.maps.LatLng(defaultLat,defaultLng),
          zoom: myZoom,
          mapTypeId: google.maps.MapTypeId.ROADP
        };
        var map = new google.maps.Map(document.getElementById("map_canvas"),mapOptions);
    }
    google.maps.event.addDomListener(window, 'load', initialize);
</script>
```

1.25 Code3-2

```html
<script>
    function test(){
        var arr = new Array()
        arr.push('25.034200,121.390527');
        arr.push('25.034020,121.390790');
        arr.push('25.033413,121.390237');
        arr.push('25.033612,121.390001');
        arr.push('25.034200,121.390527');
        AreaComp(arr);
    }
    function AreaComp(CoordArr){
       var a = new Array();

       for(var i=0; i<CoordArr.length; i++){
          var point = CoordArr[i].split(",");
          a[i] = new google.maps.LatLng(point[0],point[1]);
       }

      mypolygon = new google.maps.Polygon({
         paths: a,
         strokeColor: "#22B14C",
         strokeOpacity: 0.8,
         strokeWeight: 2,
         fillColor: "#22B14C",
         fillOpacity: 0.35
      })

      mypolygon.setMap(map);//until here is ok
      var z =  google.maps.geometry.spherical.computeArea(mypol
ygon.getPath());
      alert(z); //this is not working
    }
</script>
```

1.26 Code3-3

```html
<body onload="test();">
    <div id="map_canvas" style="width: 600px;height: 400px;"><
/div>
</body>
```

2 Practice

Make a squre around your dormitory and estimate the base area of the building.

2.1 Marauders Maps

The Marauder's Map is a magical document that reveals all of Hogwarts School of Witchcraft and Wizardry.

2.2 Animated Map

To create the animated trajetory of object, we make some changes from the last example, make notes.

1. First for

```
<!doctype html>
<html lang="en">
<head>
    ...
<style type="text/css">
    body {
        margin: 10;
        padding: 10
     }
    #map_canvas {
        position: absolute;
        width: 60%;
        height: 60%;
        left:20%;
        right:20%;
        top:30%;
        overflow: auto
    }
    div.vertical-text {
        -webkit-transform:rotate(90deg);
        -moz-transform:rotate(90deg);
        -o-transform: rotate(90deg);
        transform: rotate(90deg);
        transform-origin: left top 0;
        white-space:nowrap;
        display:block;
        bottom:0;
        width:20px;
        height:20px;
        font-family: 'Trebuchet MS', Helvetica, sans-serif;
        font-size:1.em;
        font-weight:normal;
        text-shadow: 0px 0px 1px #333;
    }
</style>
</head>
...
```

Here the syntax for CSS

- `tag {...}` : function on HTML's `<tag>` ;
- `#name {...}` : function on HTML's tag with `id/name="name"` ;
- `div.vertical-text {...}` : function on HTML's tag, `<div class='vertical-text >`;

2. create a moving object, small circle ○, which runs on the chosen trajectory:

```html
<script type="text/javascript">
...
    var lineSymbol = {
        path: google.maps.SymbolPath.CIRCLE,
        scale: 8,
        strokeColor: '#393'
    };
...
    var myMarker = new google.maps.Marker({
     ...
        animation: google.maps.Animation.DROP
    });
    google.maps.event.addListener(myMarker, 'click', toggleBoun
ce);

    function toggleBounce() {
        if (myMarker.getAnimation() != null) {
           myMarker.setAnimation(null);
        } else {
         myMarker.setAnimation(google.maps.Animation.BOUNCE);
        }
     };
```

- Marker can be animated by two ways:
  - DROP: jump to the end;
  - BOUNCE: move while animation is *null*.
- While initialized, set Marker is DROP,
- set Listener on mouse `click` event; if `on`, continuously `jump` ing while animation becomes null.

3. Determine the position of moving marker and refresh map; the respose requires a new function, `animateCircle()` , to refresh the marker position every **20** milli-second:

```html
<script type="text/javascript">
...
    var TrajPath= new google.maps.Polyline({
        ...
        icons: [{
            icon: lineSymbol,
            offset: '100%'
        }],
        ...
        map: map
     });
    //TrajPath.setMap(map);
    animateCircle();
    map.setCenter(newMarker.position);

    // adds the marker on the map
    newMarker.setMap(map);

    function animateCircle() {
      var count = 0;
      window.setInterval(function() {
         count = (count + 1) % 200;

         var icons = TrajPath.get('icons');
         icons[0].offset = (count / 2) + '%';
         TrajPath.set('icons', icons);
      }, 20);
    };

}
</script>
```

2.3 OpenMap

Using OpenMap solution, requires

1. implement javascript `getgeolocation` , (by navigator.geolocation), to check whether the geolocation is supported, MAC safari not supported.
2. acclaim new map at certain latitude/longtitude.

Here, we want to introduce an example, `where am I?` , in which we could find the place we locate now.

Details

1. HTML outline

```html
<html>
  <head>
      <meta charset="utf-8">
      <meta name="viewport" content="width=device-width,
          initial-scale=1.0, maximum-scale=1.0, user-scala
ble=no" />
      <script src="http://www.openlayers.org/api/OpenLayers.
js"></script>
  </head>
  <script>
      //pre-defined javascript fiunctions placed here
      function getLocation(){// check wheter geoloaction su
pported  }
      function getPosition(position){// get the (lat,lng)}

      function getPosition1(lat,lng){ // generate map }
  </script>
  <style>
      body {
          margin: 10;
          padding: 10
      }
      #mapdiv {
          position: relative;
          width: 60.0%;
          height: 60.0%;
          left:20%;
          right:20%;
          top:20%;
          overflow: auto;
      }
  </style>
  <body>
      <!-- main content -->
  </body>
</html>
```

2. generate Open Map:

```
function getPosition1(lat,lng){
        map = new OpenLayers.Map("mapdiv");
        map.addLayer(new OpenLayers.Layer.OSM());

        var lonLat = new OpenLayers.LonLat(lng,lat).trans
form(
                new OpenLayers.Projection("EPSG:4326"), //
transform from WGS 1984
                map.getProjectionObject() // to Spherical M
ercator Projection
        );

        var zoom=17;
        var markers = new OpenLayers.Layer.Markers( "Mark
ers");
        map.addLayer(markers);
        markers.addMarker(new OpenLayers.Marker(lonLat));
        map.setCenter (lonLat, zoom);
    }
```

3. get lat/lng if geolocation supported

```
function getPosition(position){
            // get the (lat,lng)
            lat=position.coords.latitude;
            lng=position.coords.longitude;
            document.getElementById("latitude").value = lat
.toFixed(6) ;
            document.getElementById("longitude").value = ln
g.toFixed(6);

            // map re-generate
            document.getElementById("mapdiv").innerHTML = "
";
            getPosition1(lat,lng);
        }
```

4. check whether geolocation supported,

```
function getLocation(){
        // Check whether browser supports Geolocation API
or not
        if (navigator.geolocation) { // Supported
           // To add PositionOptions
           navigator.geolocation.getCurrentPosition(getPo
sition);
        } else { // Not supported
                alert("Oops! This browser does not suppo
rt HTML Geolocation.");
        }
        getPosition();
    }
```

5. html body, create two columns for latitude and longitude, one button for click to get the location, and finally generate the map where we locate:

```
<h1>Finding Me</h1>
<div id="map_canvas"></div>
<br />
<label for="latitude">Latitude:</label><input id="la
titude" type="text" value="" />
<label for="longitude">Longitude:</label><input id="
longitude" type="" value="" />
<button onclick="getLocation()">Where am I?</button>

<div id="mapdiv"></div>

<script>
    getPosition1(25.048834785146223,121.514365);
</script>
```

While click the buttom, Choose [Allow] in popup windows to allow app to get the location.

2.4 Exercise

First set up a tour map by options, define a menu of selection as follows:

```
Location At <select id="Location">
            <option value="0">View 0</option>
            <option value="1">View 2</option>
            ...
        </select>
```

To display out the map while selection made, use embeded javascript snippt as follow:

```
<script>

    var loc = [[lat0,lon0],[lat1,lon1],...];
    getPosition1(25.048834785146223,121.514365);

    document.getElementById('Location').onchange = func
tion() {
        var index = this.value;
        //
        document.getElementById("latitude").value = loc[
index][0].toFixed(6) ;
        document.getElementById("longitude").value = loc
[index][1].toFixed(6);
        getPosition1(loc[index][0], loc[index][1]);
    };
</script>
```

2.5 Note

How to get lat/lon? try `https://www.gps-coordinates.net` .

2.6 OpenMap v.5

There are several big changes in last OpenLayers, v.5.

1. First includes the required CSS and js libraries packages within `<head>...</head>` :

```html
<link rel="stylesheet" href="https://cdn.rawgit.com/openla
yers/openlayers.github.io/master/en/v5.2.0/css/ol.css" typ
e="text/css">

 <script src="https://polyfill.io/v3/polyfill.min.js?featu
res=requestAnimationFrame,URL,Element.prototype.classList"
></script>
 <script src="https://code.jquery.com/jquery-2.2.3.min.js"
></script>
 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.
com/bootstrap/3.3.6/css/bootstrap.min.css">
 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.
3.6/js/bootstrap.min.js"></script>
 <script src="https://cdn.rawgit.com/openlayers/openlayers
.github.io/master/en/v5.2.0/build/ol.js"></script>
```

2. acclaim the CSS as usual we need:

```css
<style>
    body {
            margin: 10;
            padding: 10
    }
    #mapdiv {
        position: relative;
        width: 90.0%;
        height: 80.0%;
        left:5%;
        right:5%;
        top:10%;
        buttom:5%;
        overflow: auto;
    }

    .popover-content {
      min-width: 180px;
    }
 </style>
```

3. add `popup` above `mapdiv` :

```html
...
<div id="popup" ></div>
<div id="mapdiv" class="mapdiv"></div>
...
```

4. Our main role, ol, add map:

```javascript
var pos = ol.proj.fromLonLat([16.3725, 48.208889]);
var map = new ol.Map({
  target: 'mapdiv',
  layers: [
    new ol.layer.Tile({
      source: new ol.source.OSM()
    })
  ],
  view: new ol.View({
    center: ol.proj.fromLonLat([121.514365,25.04883478
5146223]),
      zoom: 10
  })
});
```

5. define maker and add on the map:

```javascript
var marker = new ol.Feature({
            geometry: new ol.geom.Point(
            ol.proj.fromLonLat([121.514365,25.048
834785146223])
            ),
});
var vectorSource = new ol.source.Vector({ features: [m
arker]});
var markerVectorLayer = new ol.layer.Vector({source: v
ectorSource});
map.addLayer(markerVectorLayer);
```

6. define the pop and it's trigger function:

```javascript
var popup = new ol.Overlay({
  element: document.getElementById('popup')
});
map.addOverlay(popup);

map.on('click', function(evt) {
  var element = popup.getElement();
  var coordinate = evt.coordinate;
  var hdms = ol.coordinate.toStringHDMS(ol.proj.toLonL
at(coordinate));
  var poslonlat=ol.proj.toLonLat(coordinate);
  @@0@@(element).popover({
    placement: 'top',
    animation: false,
    html: true,
    content: '<p>Position clicked was:</p>Longitude:<c
ode>' + poslonlat[0].toFixed(5) +
                              '</code><br>Latitude:<co
de>'+ poslonlat[1].toFixed(5)+'</code>'
  });
  $(element).popover('show');
});
```

what does `hdms` means? you can try to replace `content` by the following and see what happens:

```
        content: '<p>Location clicked was:</p><code>' + hdms + '
    </code>'
```

2.7 Apache Cordova

Apache cordova is an open-source project which supports flexible functions to develop mobil apps with CSS and js seamless on multiple platforms, android included; also a.k.a crosswalk:

Now it is continuously developed under `npm` environment; create a new cordova project as follows after installing npm:

1. `npm install -g cordova`
2. `> cordova create MyApp`

    `> cd MyApp`
    `> cordova platform add browser`

    `> cordova run browser`

To develop template for a basic cordova app is simple as follows:

```html
<html>
  <head>
    <title>Device Properties Example</title>
    <meta name="format-detection" content="telephone=no">
    <meta name="msapplication-tap-highlight" content="no">
    <meta name="viewport" content="initial-scale=1, width=device-width, viewport-fit=cover">

    <link rel="stylesheet" type="text/css" href="css/index.css">


    <script type="text/javascript" charset="utf-8">

    // Wait for Cordova to load
    //
    document.addEventListener("deviceready", onDeviceReady, false);

    // Cordova is ready
    //
    function onDeviceReady() {
        ...
    }

    </script>
  </head>
  <body>
    ...
    <script type="text/javascript" src="js/index.js"></script>
    <script type="text/javascript" charset="utf-8" src="cordova.js"></script>
  </body>
```

Let us to re-do a old openmap(v2) app:

1. define tags in body to display the information:

```html
<body>
 ...
 <p id="geolocation">Finding geolocation...</p>
 ...
</body>
```

2. implement the js part:

```html
<script type="text/javascript" charset="utf-8">
...
function onDeviceReady() {
    navigator.geolocation.getCurrentPosition(onSuccess, onError);
 }

 // onSuccess Geolocation
 //
 function onSuccess(position) {
     var element = document.getElementById('geolocation');
     element.innerHTML = 'Latitude: '+ position.coords.latitude+ '<br />'
     +'Longitude: '+ position.coords.longitude + '<br />'
     +'Altitude: '+ position.coords.altitude + '<br />'
     +'Accuracy: '+ position.coords.accuracy + '<br />'
     +'Altitude Accuracy: '+ position.coords.altitudeAccuracy + '<br />'
     +'Heading: ' + position.coords.heading+ '<br />'
     +'Speed: '+ position.coords.speed + '<br />'
     +'Timestamp: ' + position.timestamp          + '<br />';
 }
</script>
<body>
```

2.8 Exercise

Redo the part in old openmap demo by implementing `clickm me` button to get the current position.

Type *Markdown* and LaTeX: α^2