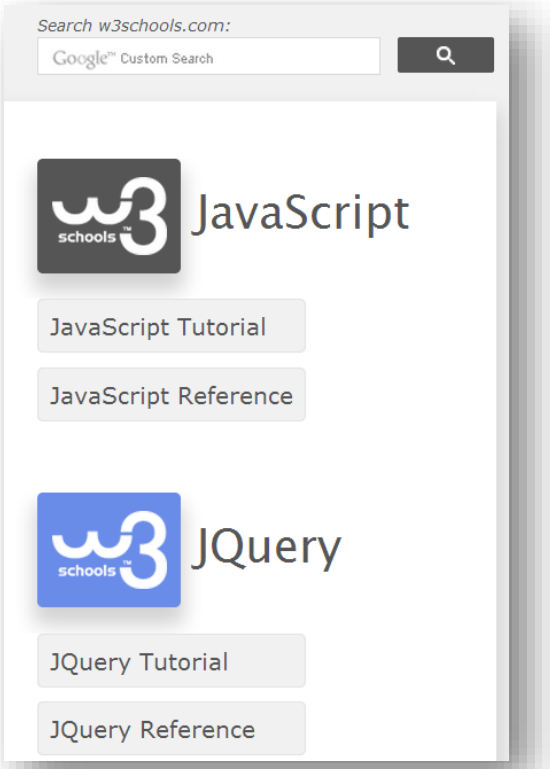
Mobile Simulation

2017-2

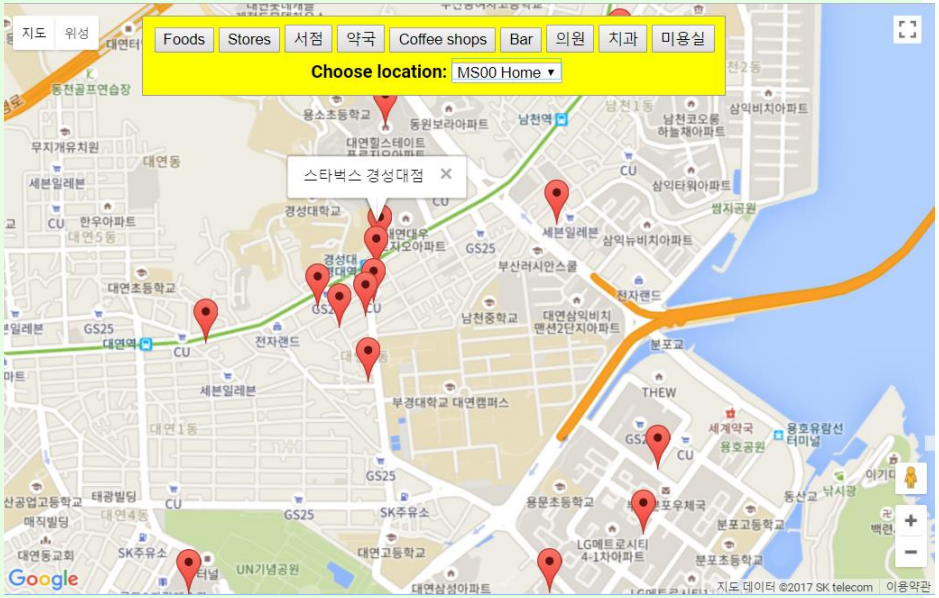


Weekly plan (HTML5, 1st semester 2017)

- **wk01 : Introduction to curriculum & current state of HTML5**
- **wk02 : Making HTML5 documents**
- **wk03 : Table, iframe and media**
- **wk04 : Semantic tag and Form**
- **wk05 : CSS3 I. Basic**
- **wk06 : CSS3 II. Advanced**
- **wk07 : CSS3 III. Animation**
- **wk08 : Mid-term Exam.**
- **wk09 : Javascript : Data types & operators**
- **wk10 : Javascript : Loop & functions**
- **wk11 : Javascript : Core objects**
- **wk12 : Javascript : DOM**
- **wk13 : Javascript : Event handling I**
- **wk14 : Javascript : Event handling II**
- **wk15 : Final exam.**

Weekly plan (Mobile Simulation, 2nd semester 2017)

- **wk01 : Introduction to curriculum & current state**
- **wk02 : Browser Object Model (BOM), installing Brackets editor**
- **wk03 : Canvas graphics I. Basic**
- **wk04 : Canvas graphics II. Image & Transformation**
- **wk05 : Canvas graphics III. Animation**
- **wk06 : Canvas graphics IV. Game**
- **wk07 : 보강 기간에 보강 실시**
- **wk08 : Mid-term Exam.**
- **wk09 : jQuery I. Basic**
- **wk10 : jQuery II. Application**
- **wk11 : SVG, Drag & Drop**
- **wk12 : Google Map I: Intro**
- **wk13 : Google Map II: Apps**
- **wk14 : JS server - node.js**
- **wk15 : Final exam.**



과제09. msnn_rpt09.zip

[실습과제09] Google Map Basic

[1] 실습 결과 그림 3장 저장.
[2] MSnn_NPP_Alert .png
MSnn_Place_Markers.png
MSnn_MapType_UI.png

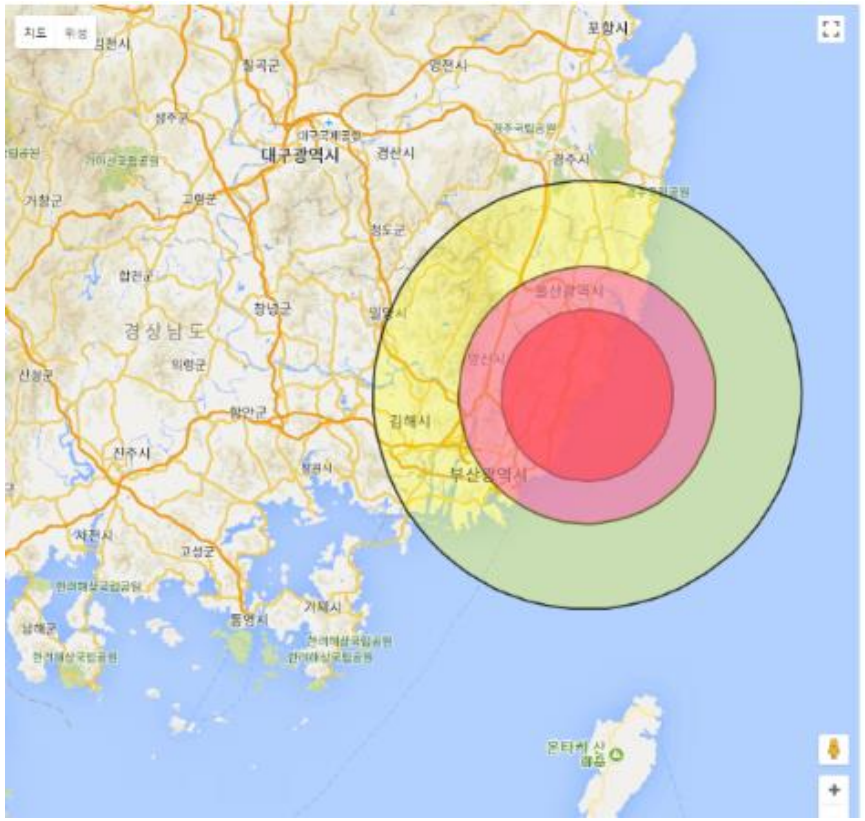
****** MSnn_Rpt09.zip 으로 압축해서 제출하시오.**

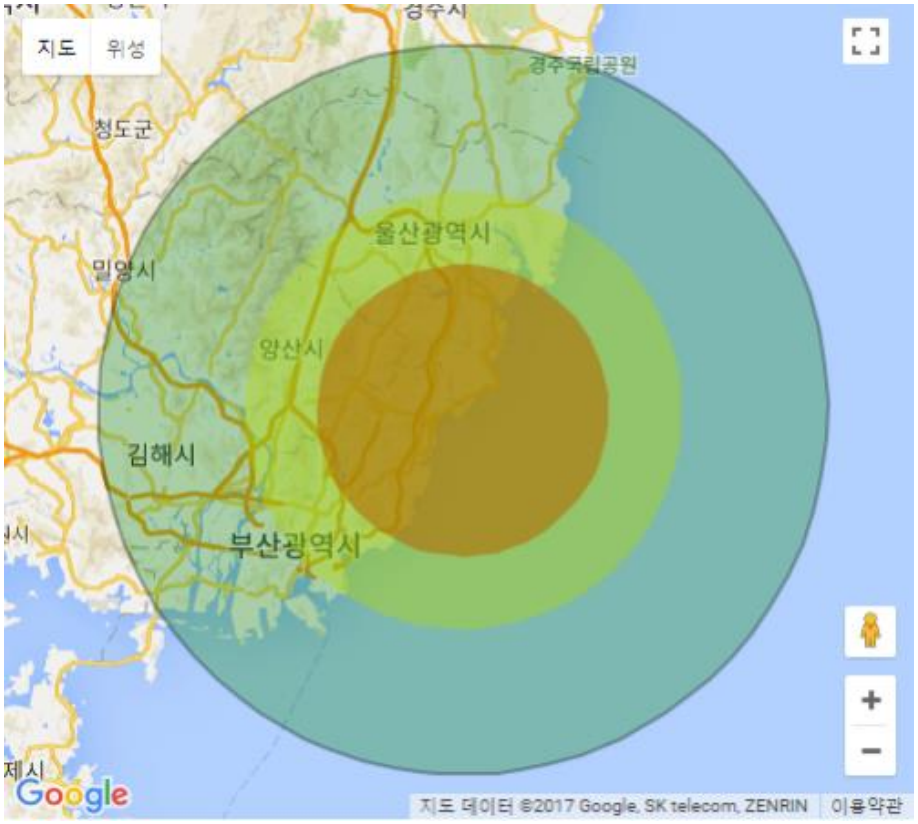
[제출파일] msnn_rpt09.zip (11월14일 오후 6시 마감)

html 파일과 사용된 그림을 압축하여 이메일로 “msnn_rpt09” 제목으로 제출

Email : chaos21c@gmail.com

Result


포항시
칠곡군
경산시
대구광역시
경주시
청도군
경상남도
밀양시
울산광역시
김해시
부산광역시
고성군
거제시
통영시


지도
위성
경주국립공원
청도군
울산광역시
밀양시
양산시
김해시
부산광역시
Google
지도 데이터 ©2017 Google, SK telecom, ZENRIN
이용약관

CHAPTER 13

Google Map II. Apps

Develop apps using Google Map Api V3 & HTML5 Geolocation.

Before you start, you will need a free API key from Google (???).

API = Application programming interface.

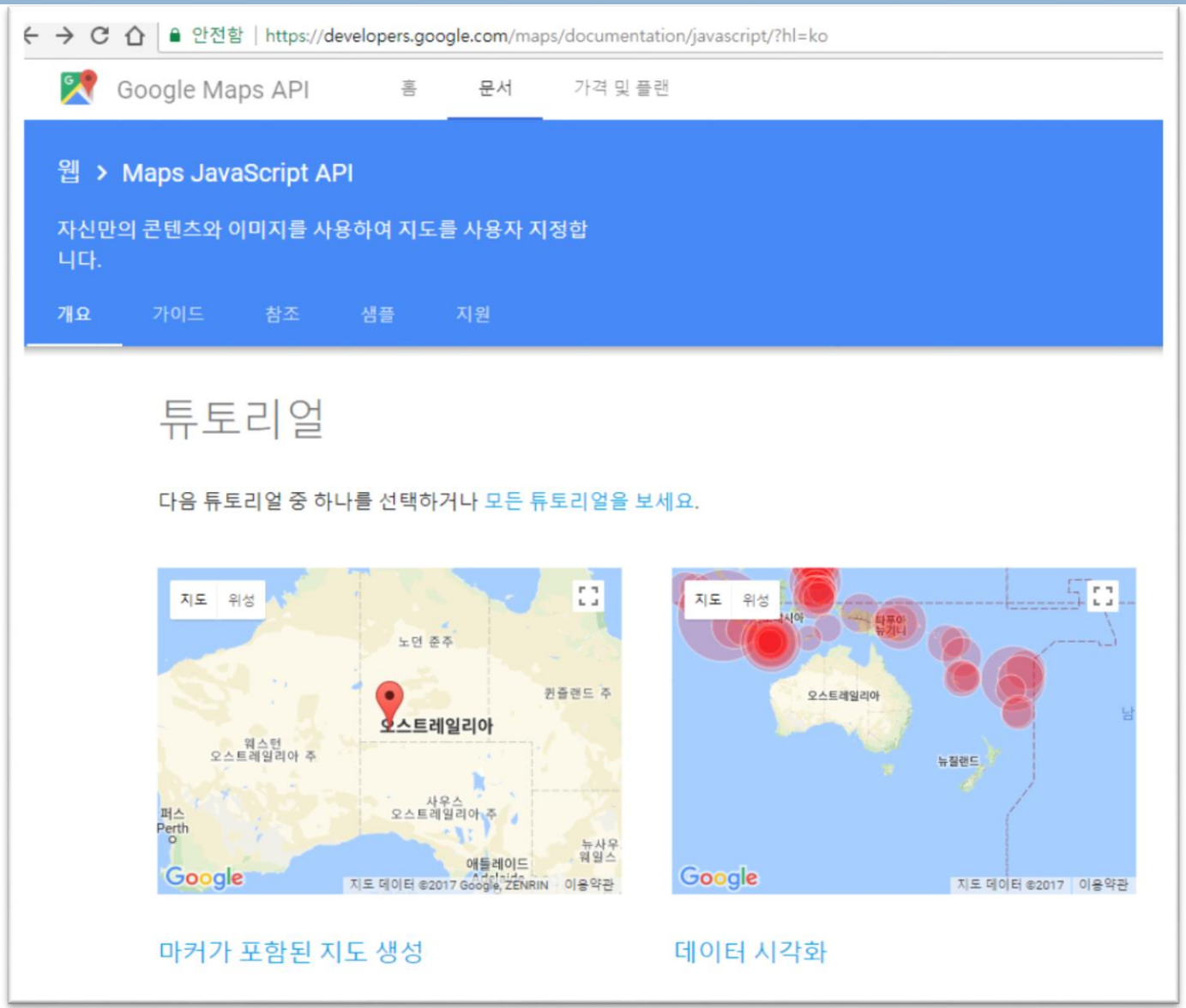
Reference (w3schools.com)

HTML5 Geolocation
http://www.w3schools.com/html/html5_geolocation.asp

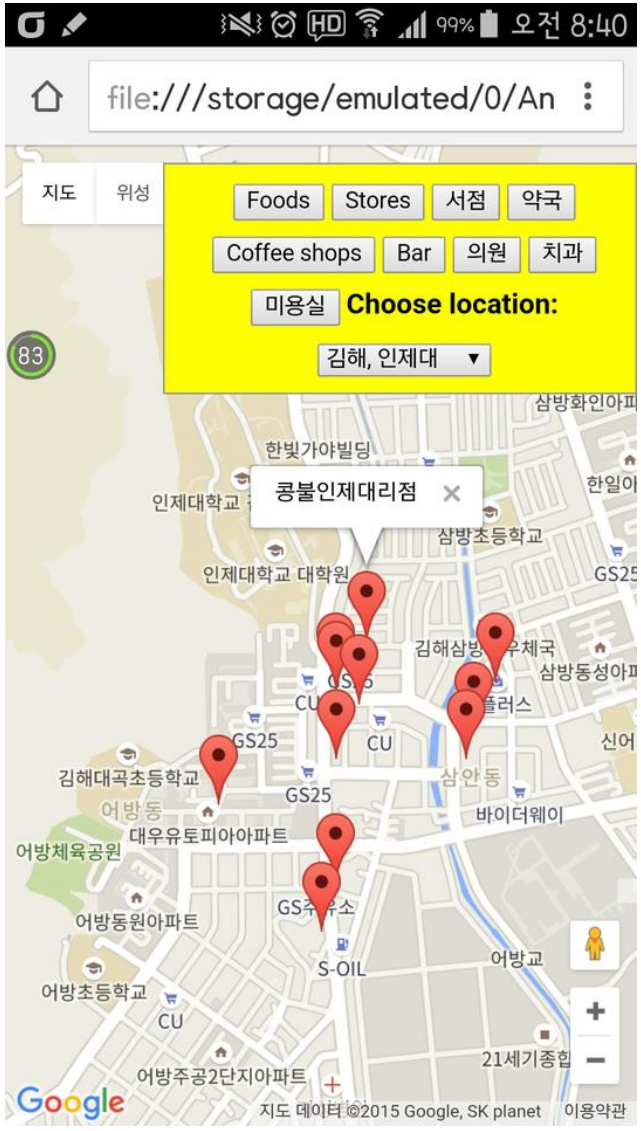
Google Maps API Tutorial
https://www.w3schools.com/graphics/google_maps_intro.asp

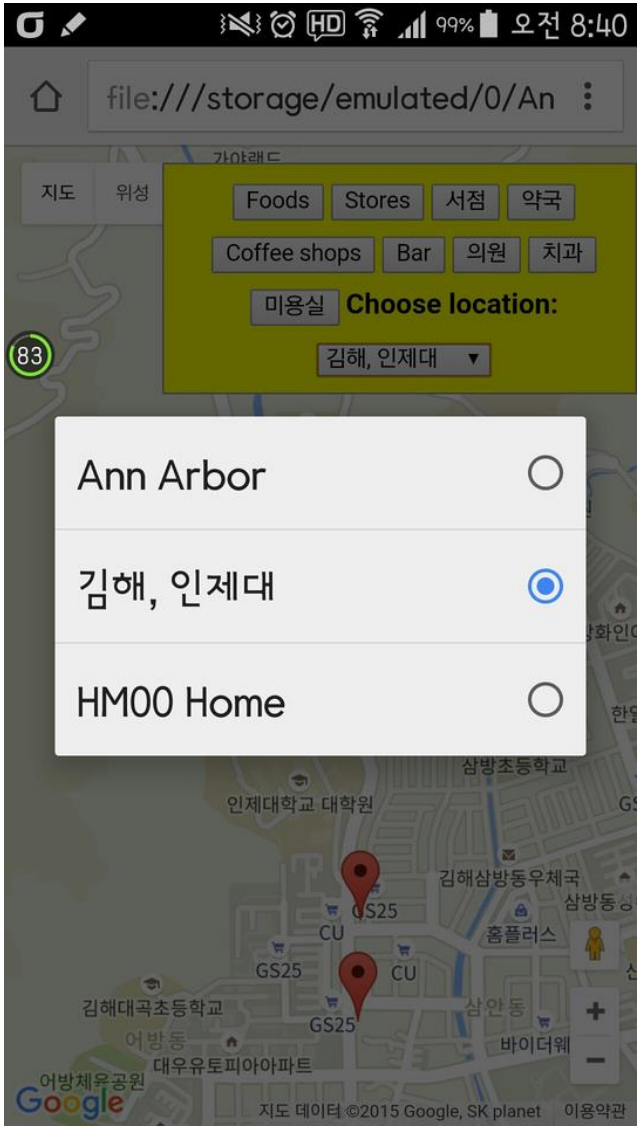
Google Maps JavaScript API v3 (google.com)
https://developers.google.com/maps/documentation/javascript/?hl=ko

Reference (Google Maps JS API)



Target: **Places**



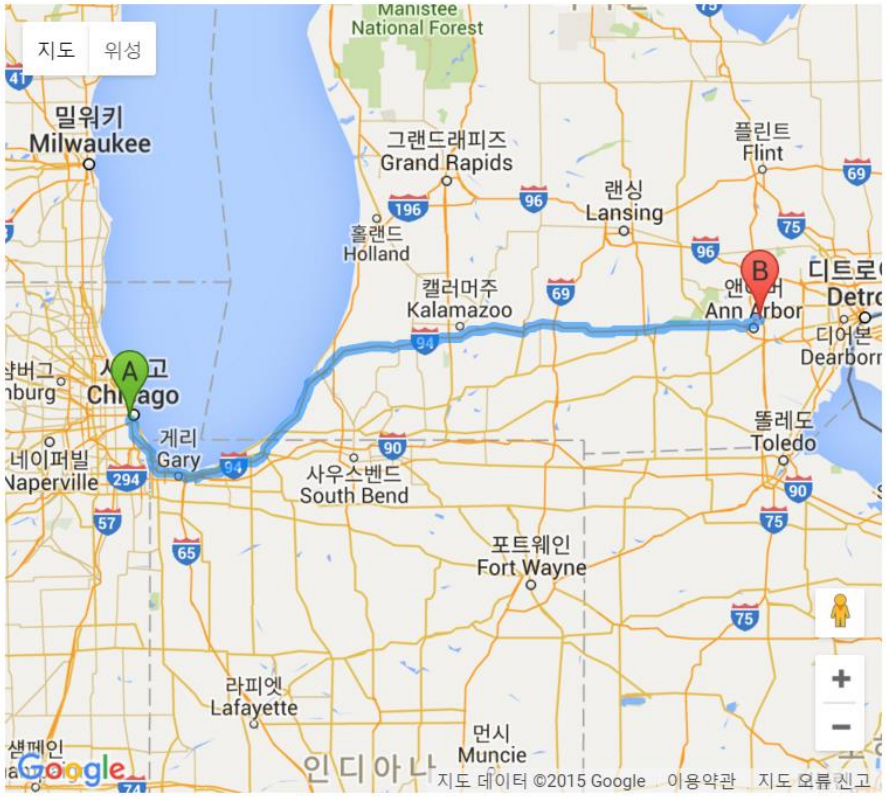


Target (N.A.):

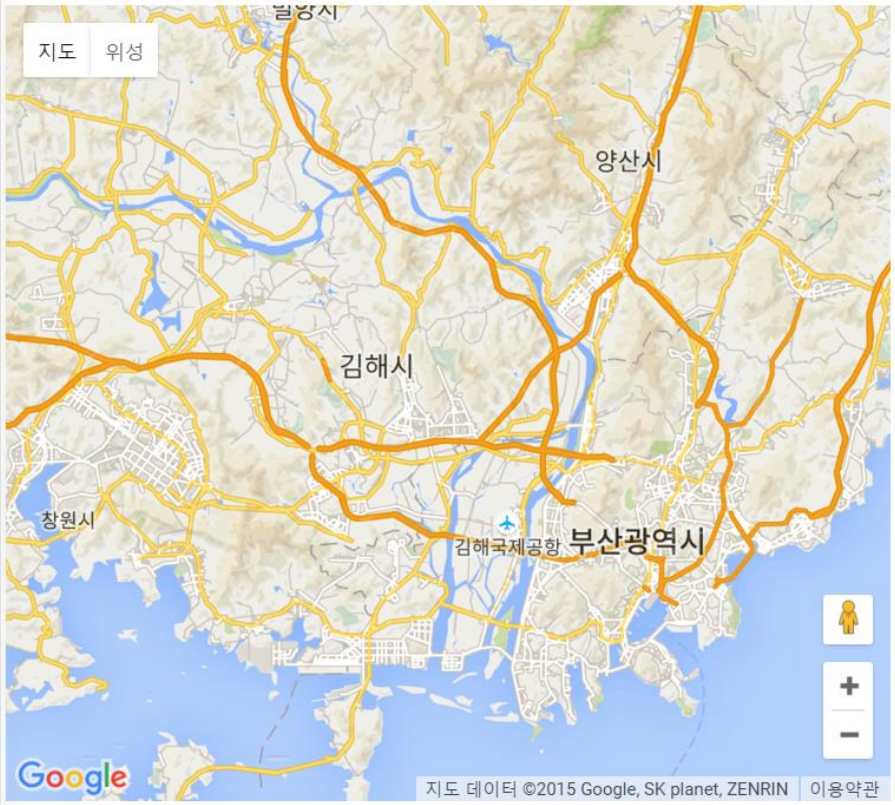
Using Google Map API V3

Target: Directions

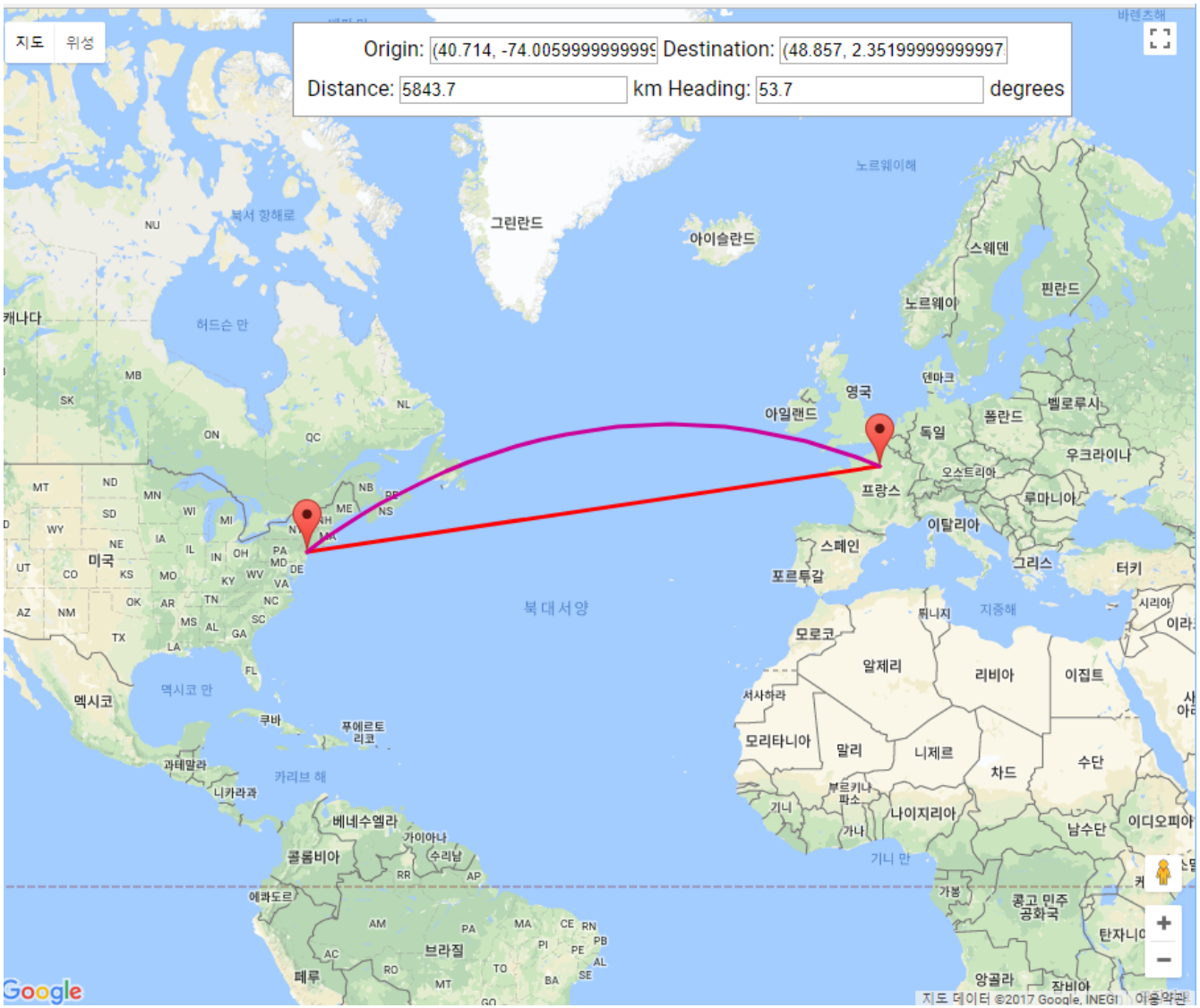
Using Google Map : Directions



Using Google Map : Directions

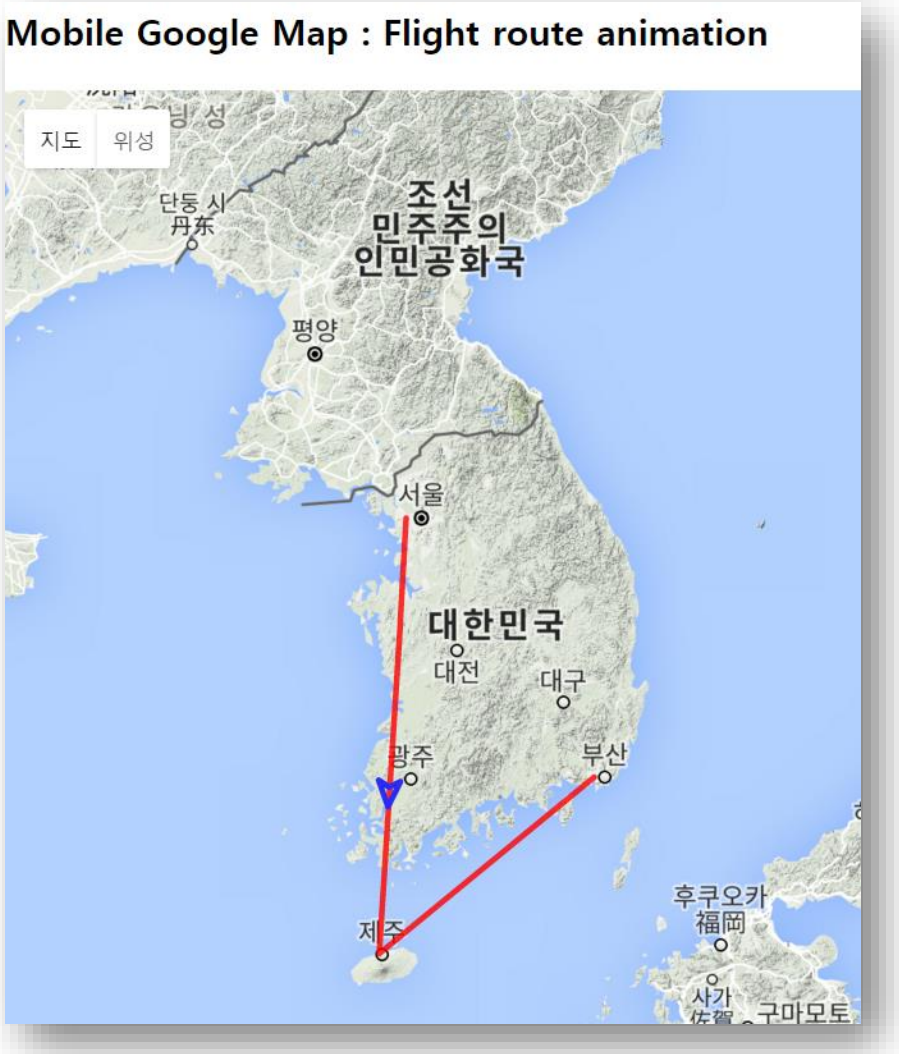


Target: Distance & heading



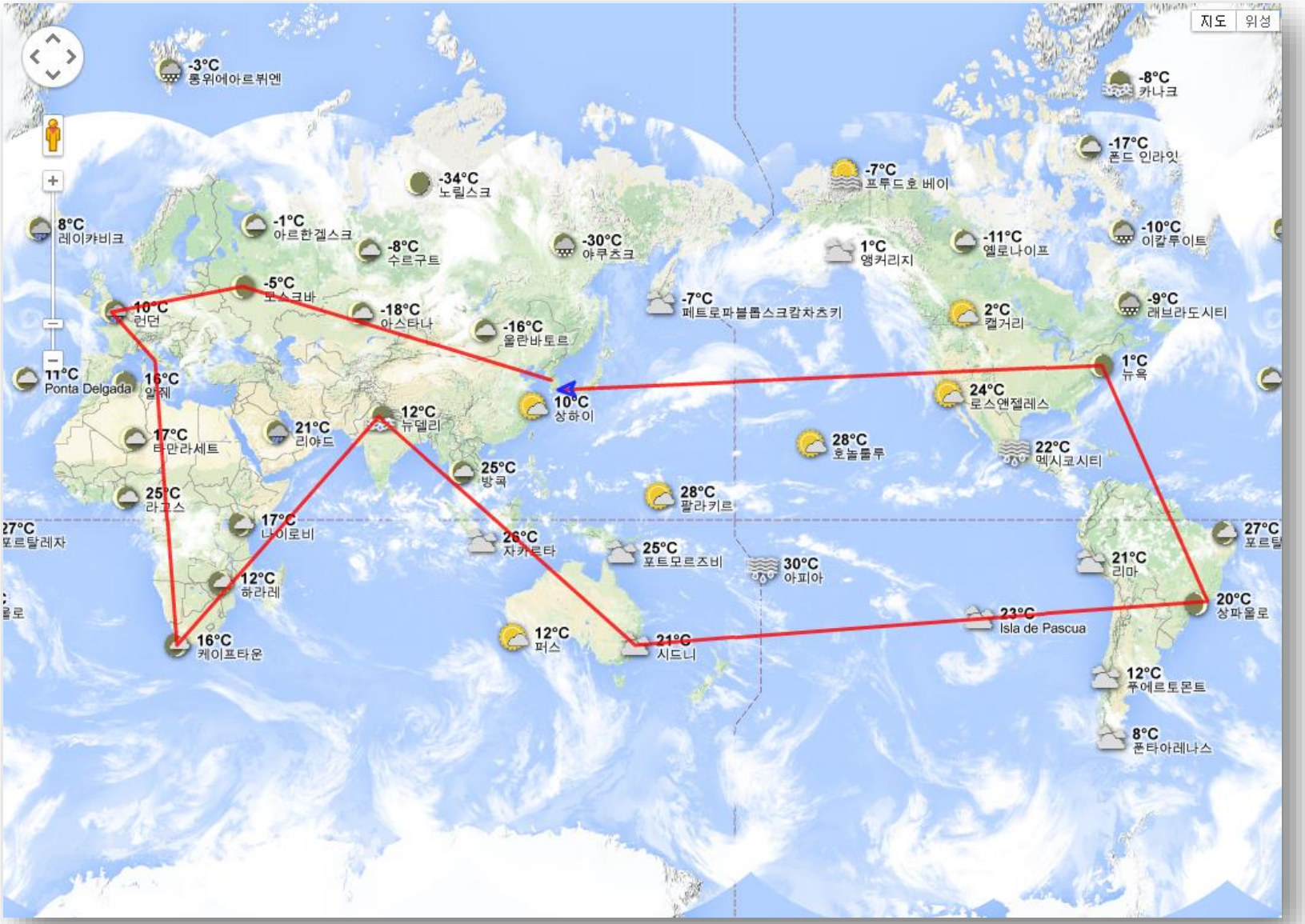
Challenge 1:

Animation on Google map



Challenge 2:

World travel animation



Google Map Javascript API

Advanced

Mobile app, Animation, Direction, Places, Travel

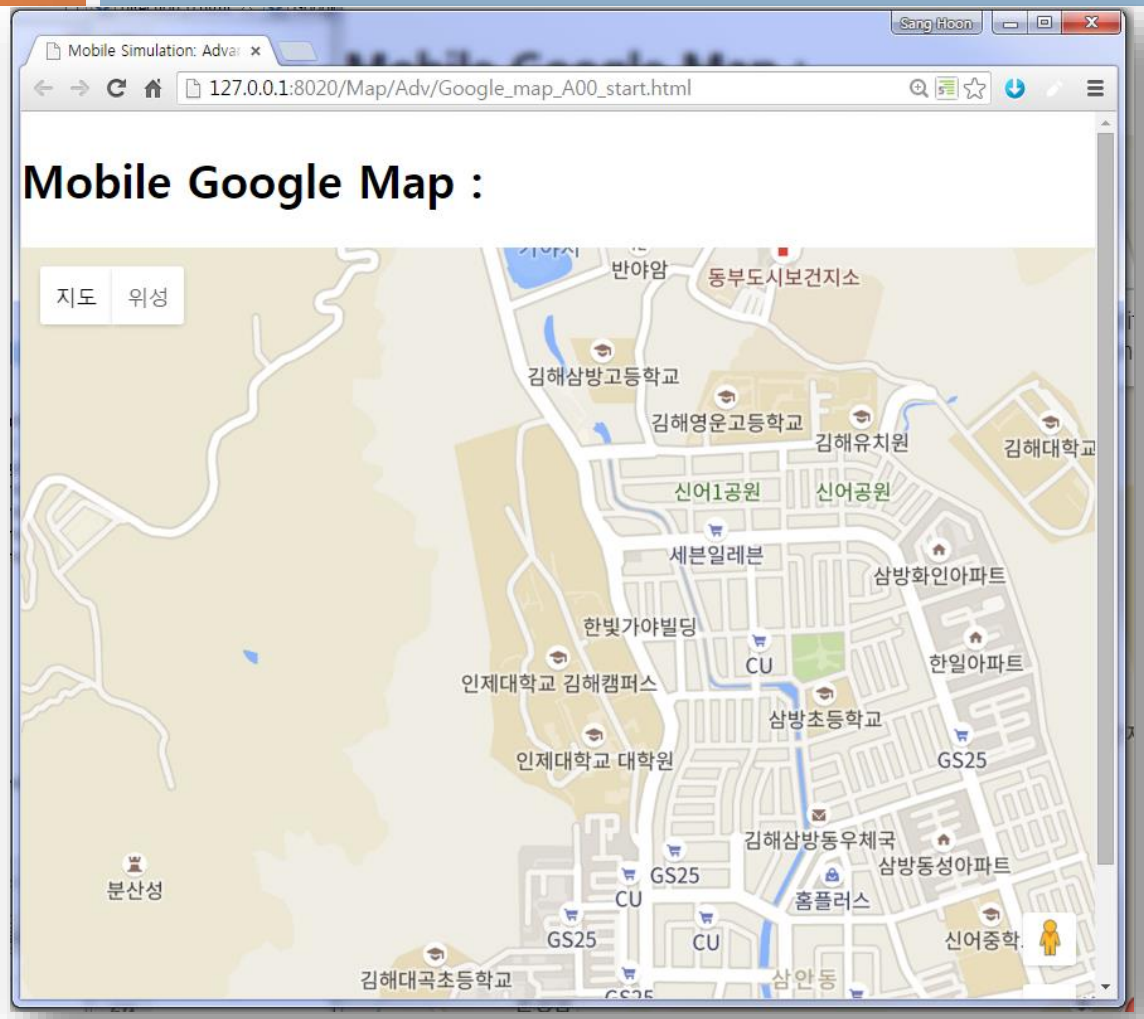
A1.1 Google Map for Mobile

```
<!DOCTYPE html>
<html>
<head>
   <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
    <meta charset="utf-8">
    <title> Mobile Simulation: Advanced Google Map </title>
   <style>
         html, body {
           height: 100%;
           margin: 0;
           padding: 0;
         }
         #map {
           height: 100%;
         }
   </style>

</head>

<body>

    <h2>Mobile Google Map :  </h2>
    <div id="map"> </div>
```

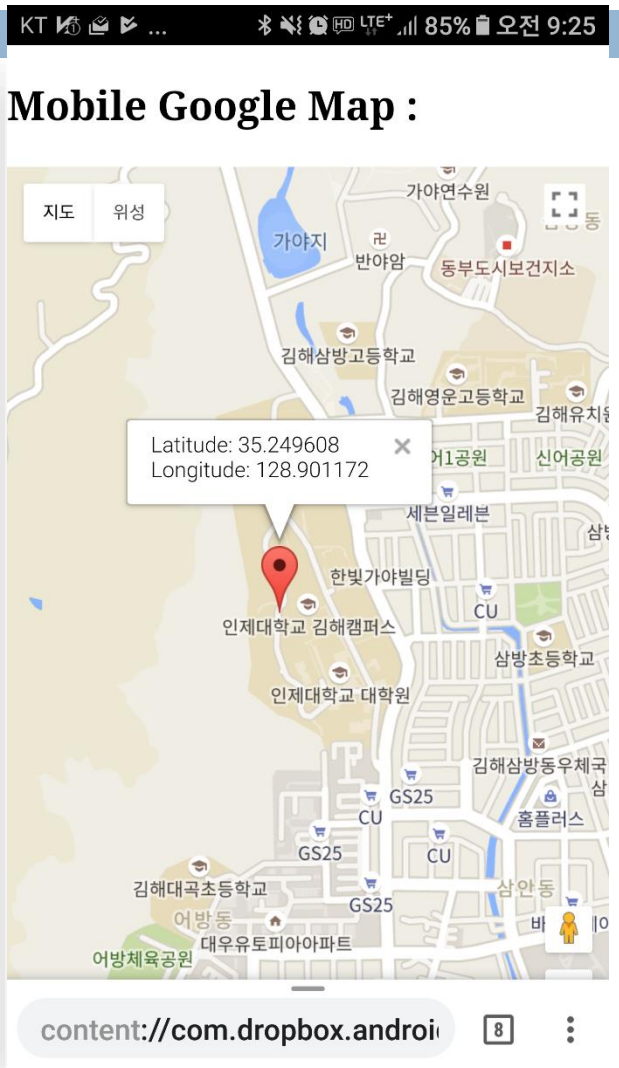
A1.2 Google Map for Mobile

```
<script>
    var e323 = {lat: 35.249164, lng: 128.901881};  // E323
    //var e323=new google.maps.LatLng(35.249164, 128.901881);

    var map;

    function initMap()
    {
      var mapProp = {
        center:e323,
        zoom:15,
        mapTypeId:google.maps.MapTypeId.ROADMAP
      };
      map=new google.maps.Map(document.getElementById("map"), mapProp);

      google.maps.event.addListener(map, 'click', function(event) {
        placeMarker(event.latLng);
      });

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

      var infowindow = new google.maps.InfoWindow({
        content: 'Latitude: ' + location.lat() +
        '<br>Longitude: ' + location.lng()
      });
      infowindow.open(map,marker);
     }
    }
    //google.maps.event.addDomListener(window, 'load', initialize);

</script>

<script
    src="https://maps.googleapis.com/maps/api/js?callback=initMap">
</script>
```

[Result] A1. Mobile setting



PC



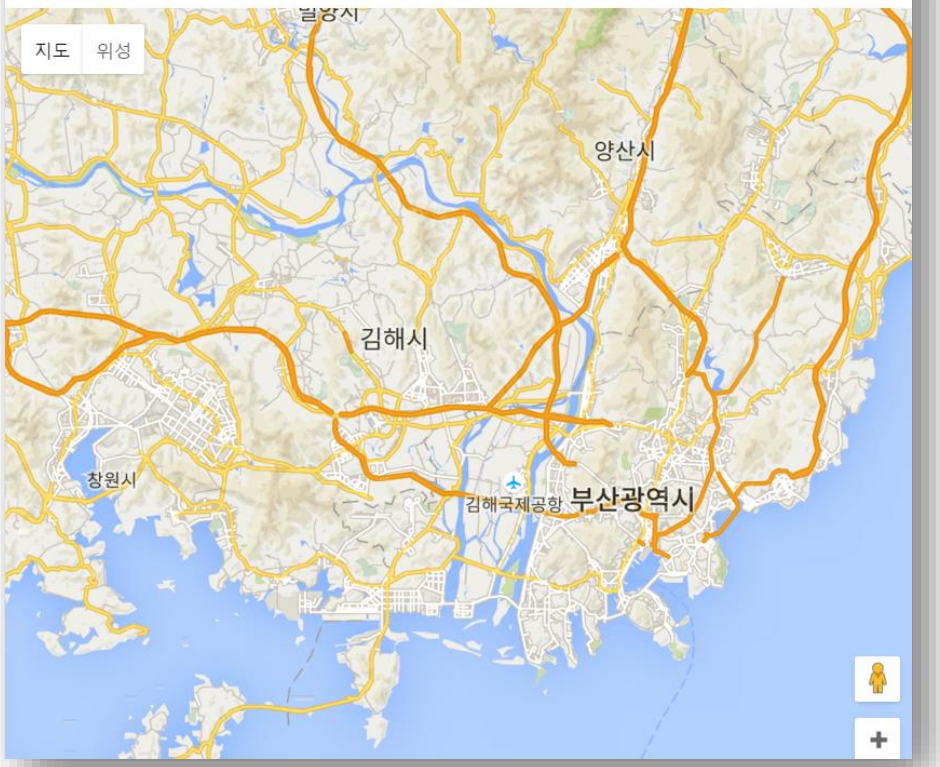
Mobile

A2. Traffic

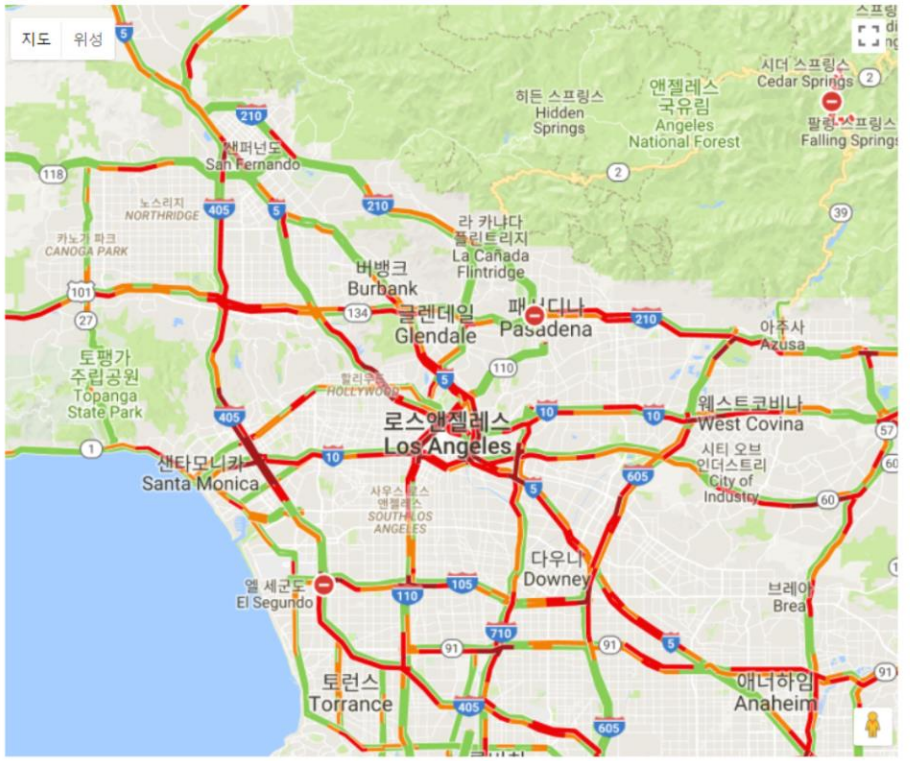
```
<script>
var e323 = {lat: 35.249164, lng: 128.901881};  // E323
var LA = {lat: 34.049245, lng: -118.241043};

function initMap()
{
  var mapProp = {
    center:LA, //e323,
    zoom:10,
    mapTypeId:google.maps.MapTypeId.ROADMAP
  };

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

  var trafficLayer = new google.maps.TrafficLayer();
  trafficLayer.setMap(map);

}

</script>
```

[Result] A2. Traffic

Mobile Google Map : Traffic



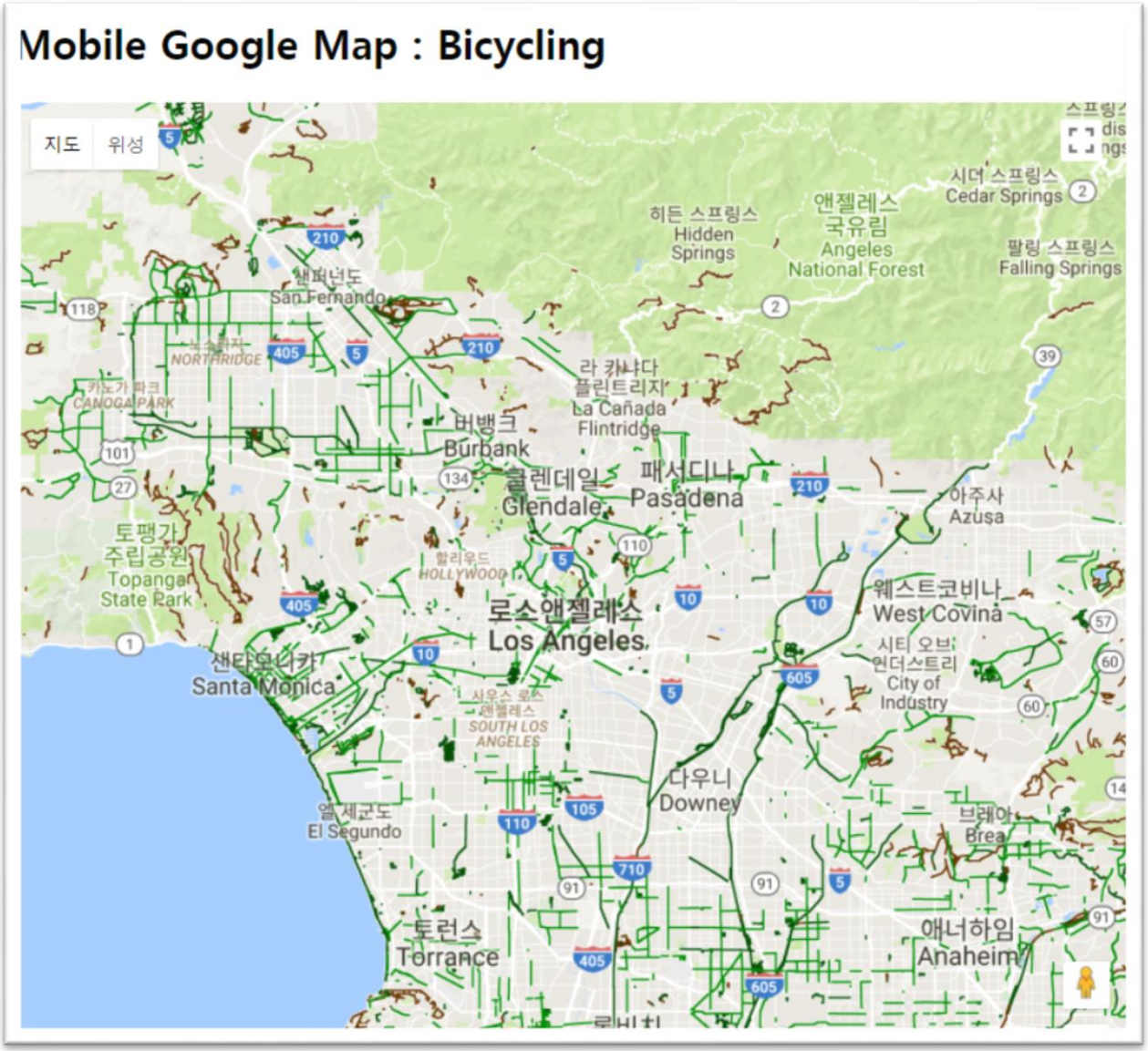
Mobile Google Map : Traffic



A3. Bicycling route

```
<script>
var e323 = {lat: 35.249164, lng: 128.901881};  // E323
var LA = {lat: 34.049245, lng: -118.241043};

function initMap()
{
  var mapProp = {
    center:LA, //e323,
    zoom:10,
    mapTypeId:google.maps.MapTypeId.ROADMAP
  };

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

  var bykeLayer = new google.maps.BicyclingLayer();
  bykeLayer.setMap(map);

}

</script>
```

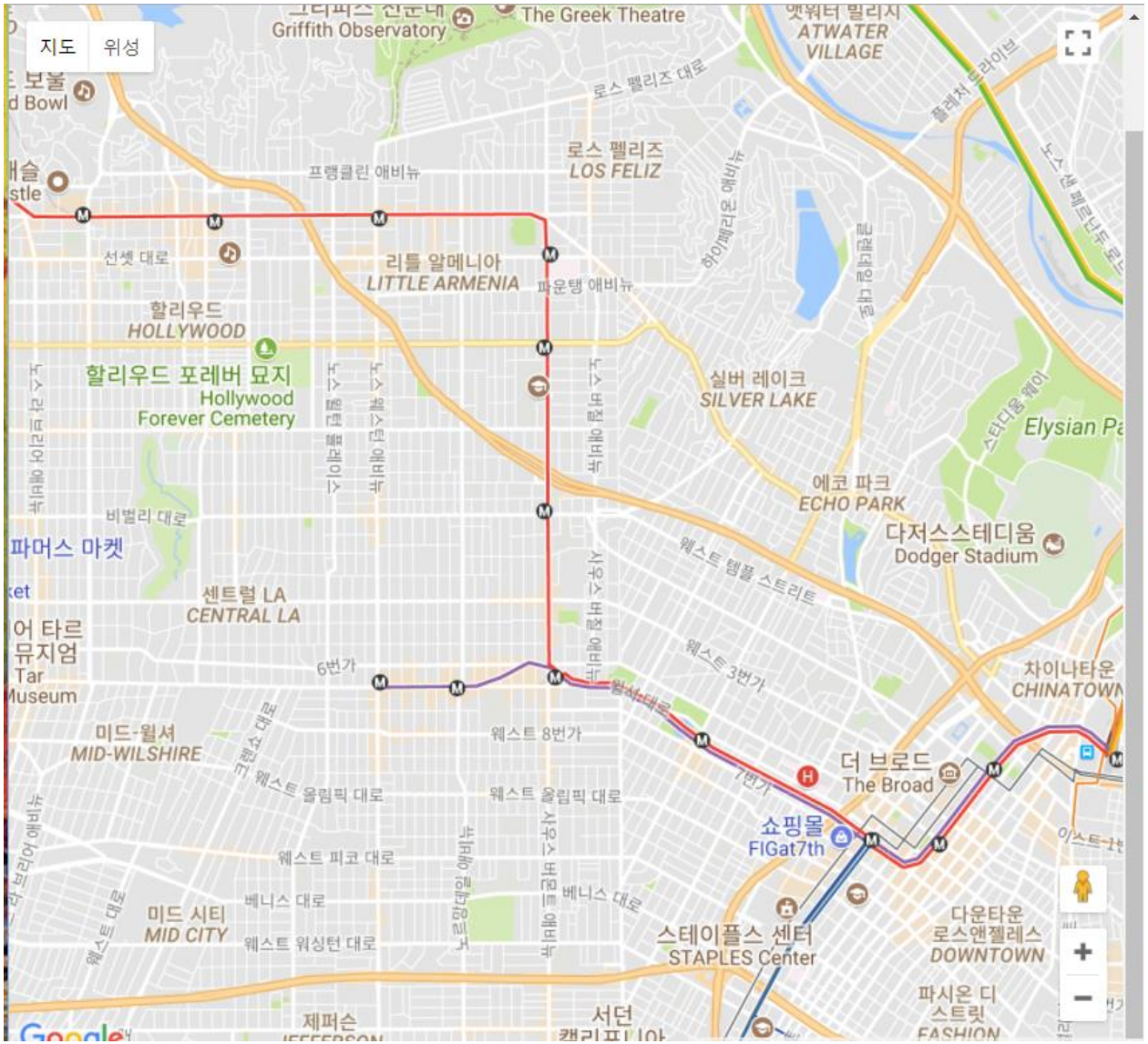
[Result] A3. Bicycling route



A4. Transit

```
<script>
var e323 = {lat: 35.249164, lng: 128.901881};  // E323
var LA = {lat: 34.049245, lng: -118.241043};

function initMap()
{
  var mapProp = {
    center:LA, //e323,
    zoom:10,
    mapTypeId:google.maps.MapTypeId.ROADMAP
  };

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

var aLayer = new google.maps.TransitLayer();
  aLayer.setMap(map);
}

</script>
```

[Result] A4. Transit



Google Map Coverage (ㅠㅠ)

지도 커버리지 세부정보

☆☆☆☆☆

Google 지도 팀은 전세계적인 커버리지를 개선하기 위해 끊임없이 노력하고 있습니다. 다음 표는 최신 커버리지 세부정보를 국가별로 보여줍니다. 일부 데이터(예: 대중교통 경로)는 도시 수준에서 사용할 수 있으며 이 목록에는 나타나지 않습니다. 참고로, 데이터 공급자와의 라이선스 계약이 변경되면 데이터 커버리지가 변경될 수 있습니다.

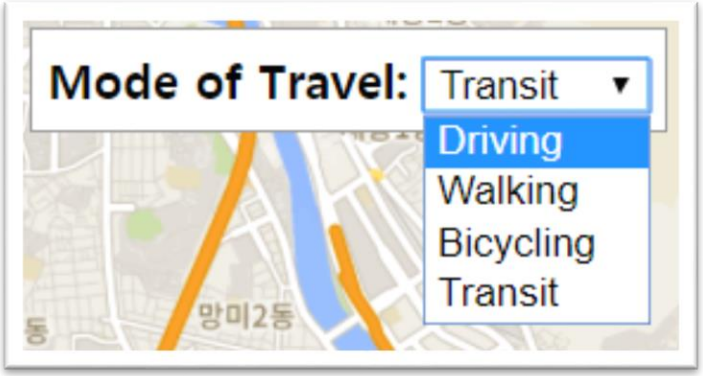
국가 또는 지역을 입력하여 아래 표를 필터링합니다.

지역 코드	국가/지역	지도 타일	지오코딩	교통량 계층	자동차 길찾기	자전거 길찾기	도보 길찾기
AF	아프가니스탄	●	►	-	►	-	►
AL	알바니아	►	►	-	►	-	►
DZ	알제리	►	►	►	►	-	►
AS	미국령 사모아	►	►	-	►	-	►
AD	안도라	►	►	-	►	-	►
US	미국	►	►	►	►	►	►
KR	대한민국	►	►	-	-	-	-

https://developers.google.com/maps/coverage

A5.0 All modes of travels

```
<body>
  <div id="panel">
  <b>Mode of Travel: </b>
  <select id="mode" onchange="calcRoute();">
    <option value="DRIVING">Driving</option>
    <option value="WALKING">Walking</option>
    <option value="BICYCLING">Bicycling</option>
    <option value="TRANSIT">Transit</option>
  </select>
  </div>
  <div id="map-canvas"> </div>

</body>
```



A5.1 All modes of travels

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
    <meta charset="utf-8">
    <title>Travel modes in directions</title>
    <style>
      html, body, #map-canvas {
        height: 100%;
        margin: 0px;
        padding: 0px
      }
      #panel {
        position: absolute;
        top: 5px;
        left: 50%;
        margin-left: -180px;
        z-index: 5;
        background-color: #fff;
        padding: 5px;
        border: 1px solid #999;
      }
    </style>
    <script src="https://maps.googleapis.com/maps/api/js?language=ko"></script>
```

A5.2 All modes of travels

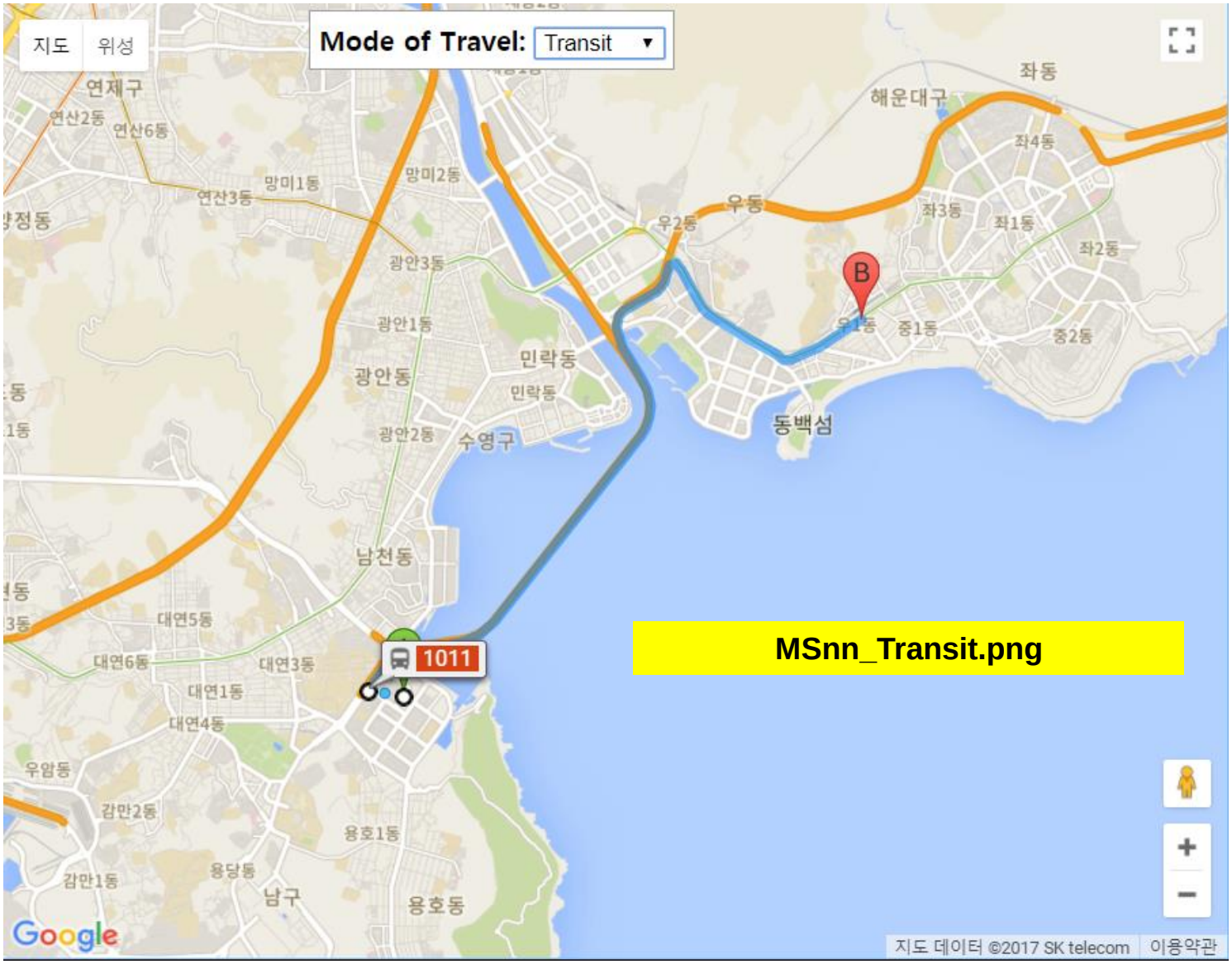
```
<script>
var directionsDisplay;
var directionsService = new google.maps.DirectionsService();
var map;

var e323 = new google.maps.LatLng(35.249164, 128.901881);
var indang = new google.maps.LatLng(35.248137, 128.902949);
var library = new google.maps.LatLng(35.248721, 128.902704);
var studctr = new google.maps.LatLng(35.250496, 128.902470);
var myhome = new google.maps.LatLng(35.132354, 129.111521);

var p1 = new google.maps.LatLng(37.523870, 126.933629);  // seoul 1
var p2 = new google.maps.LatLng(37.527069, 126.913631);  // seoul 2
var b1 = new google.maps.LatLng(35.132163, 129.112248);  // busan 1
var b2 = new google.maps.LatLng(35.163522, 129.158781);  // busan 2

var a1 = new google.maps.LatLng(42.318412, -83.706596);  // ann arbor 1
var a2 = new google.maps.LatLng(42.318839, -83.685928);  // ann arbor 2


var haight = new google.maps.LatLng(37.7699298, -122.4469157);
var oceanBeach = new google.maps.LatLng(37.7683909618184, -122.51089453697205);

function initialize() {
  directionsDisplay = new google.maps.DirectionsRenderer();
  var mapOptions = {
    zoom: 17,
    center: b1, //a1//oceanBeach //e323
  };
  map = new google.maps.Map(document.getElementById('map-canvas'), mapOptions);
  directionsDisplay.setMap(map);
}
```

A5.3 All modes of travels

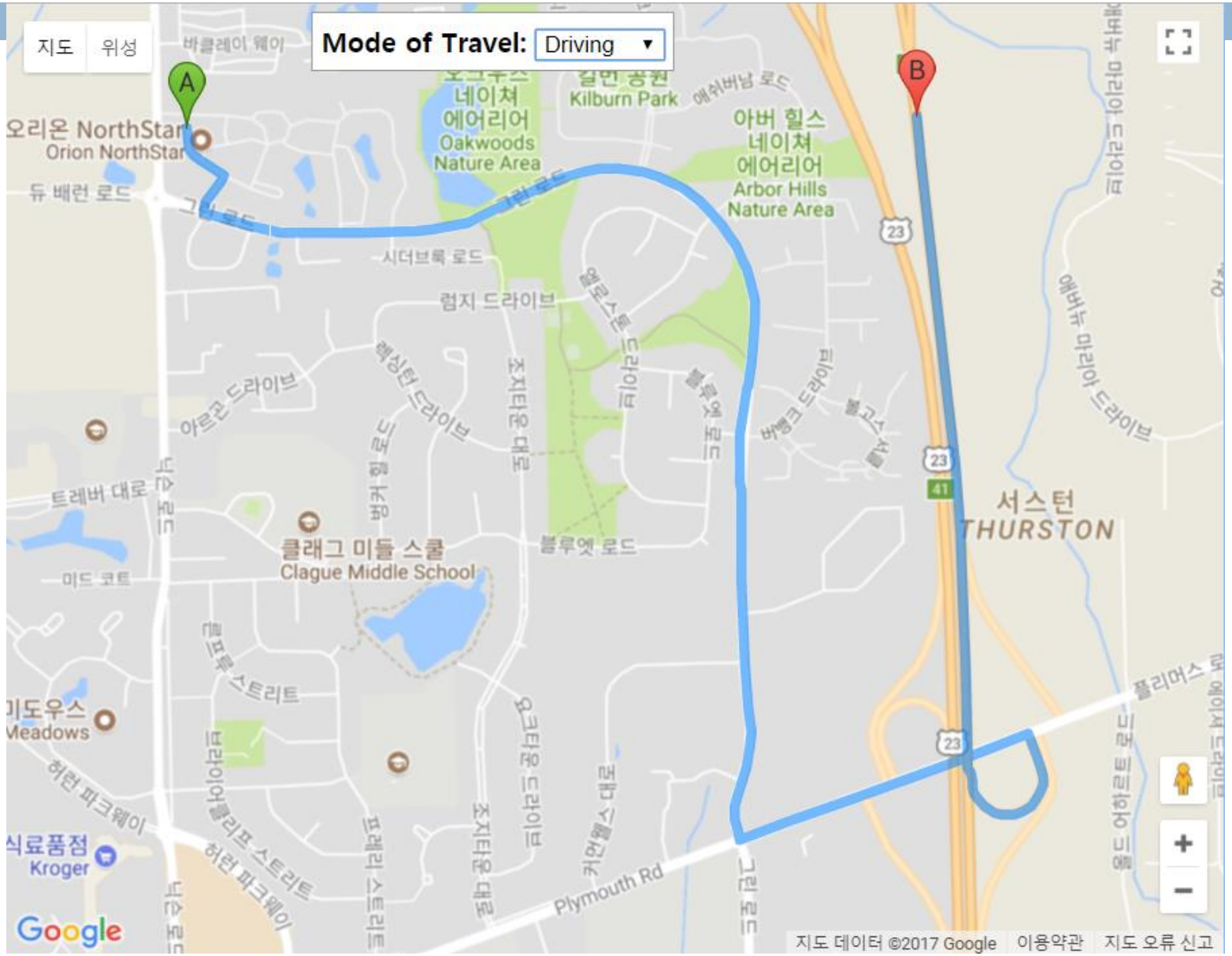
```
function calcRoute() {
  var selectedMode = document.getElementById('mode').value;
  var request = {
      origin: b1, //haight, //e323,
      destination: b2, //oceanBeach, //indang,
      // Note that Javascript allows us to access the constant
      // using square brackets and a string value as its
      // "property."
      travelMode: google.maps.TravelMode[selectedMode]
  };
  directionsService.route(request, function(response, status) {
    if (status == google.maps.DirectionsStatus.OK) {
      directionsDisplay.setDirections(response);
    }
    else{
        alert("Can't find the route you requested.");
    }
  });
}

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

</script>
```

[Result] A5. All modes of travels

현재 가장 많이 이용하는 통학 구간을 구글지도에 표시하여 **MSnn_Transit.png** **로 저장**



[Result] A5. All modes of travels



A6.1 Google map library: geometry

```
<!DOCTYPE html>
<html>
  <head>
    <title>Navigation functions (heading)</title>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
    <meta charset="utf-8">
    <script src="https://maps.googleapis.com/maps/api/js?libraries=geometry&callback=initMap"
        async defer></script>
    <style>
      html, body {
        height: 100%;
        margin: 0;
        padding: 0;
      }
      #map {
        height: 100%;
      }
    #floating-panel {
      position: absolute;
      top: 10px;
      left: 25%;
      z-index: 5;
      background-color: #fff;
      padding: 5px;
      border: 1px solid #999;
      text-align: center;
      font-family: 'Roboto','sans-serif';
      line-height: 30px;
      padding-left: 10px;
    }
    </style>
  </head>
```

A6.2 Google map library: geometry

```
<body>
  <div id="map"></div>
  <div id="floating-panel">
    Origin: <input type="text" readonly id="origin">
    Destination: <input type="text" readonly id="destination"><br>
    Distance: <input type="text" readonly id="distance"> km
    Heading: <input type="text" readonly id="heading"> degrees
  </div>

<script>
    var marker1, marker2;
    var poly, geodesicPoly;

    function initMap() {
      var map = new google.maps.Map(document.getElementById('map'), {
        zoom: 5,
        center: {lat: 34, lng: -40.605}
      });

      map.controls[google.maps.ControlPosition.TOP_CENTER].push(
          document.getElementById('info'));

      marker1 = new google.maps.Marker({
        map: map,
        draggable: true,
        position: {lat: 40.714, lng: -74.006}
      });
      marker2 = new google.maps.Marker({
        map: map,
        draggable: true,
        position: {lat: 48.857, lng: 2.352}
      });
```

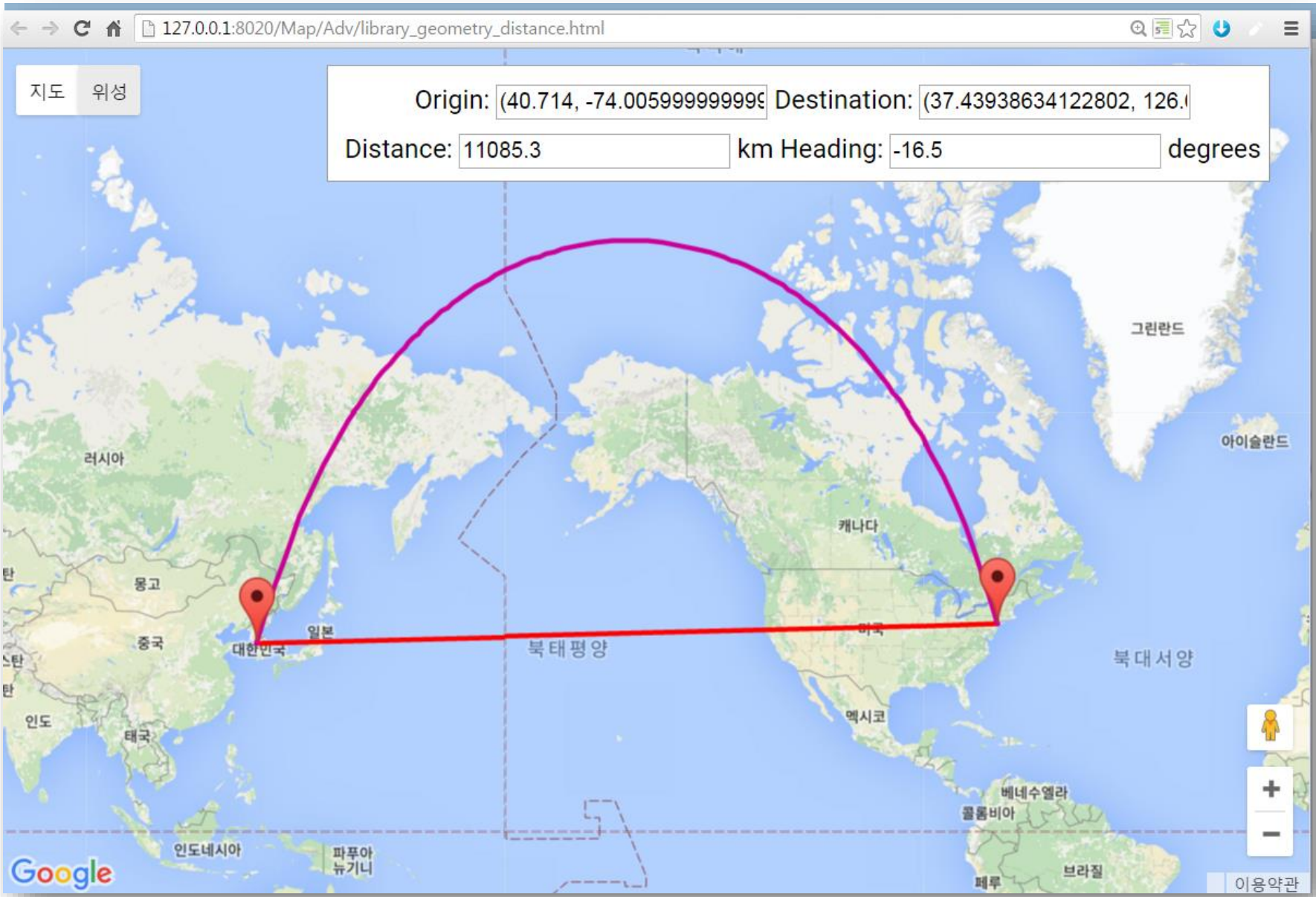
A6.3 Google map library: geometry

```
  var bounds = new google.maps.LatLngBounds(
      marker1.getPosition(), marker2.getPosition());
  map.fitBounds(bounds);

  google.maps.event.addListener(marker1, 'position_changed', update);
  google.maps.event.addListener(marker2, 'position_changed', update);

  poly = new google.maps.Polyline({
    strokeColor: '#FF0000',
    strokeOpacity: 1.0,
    strokeWeight: 3,
    map: map,
  });

  geodesicPoly = new google.maps.Polyline({
    strokeColor: '#CC0099',
    strokeOpacity: 1.0,
    strokeWeight: 3,
    geodesic: true,
    map: map
  });

  update();
}
```

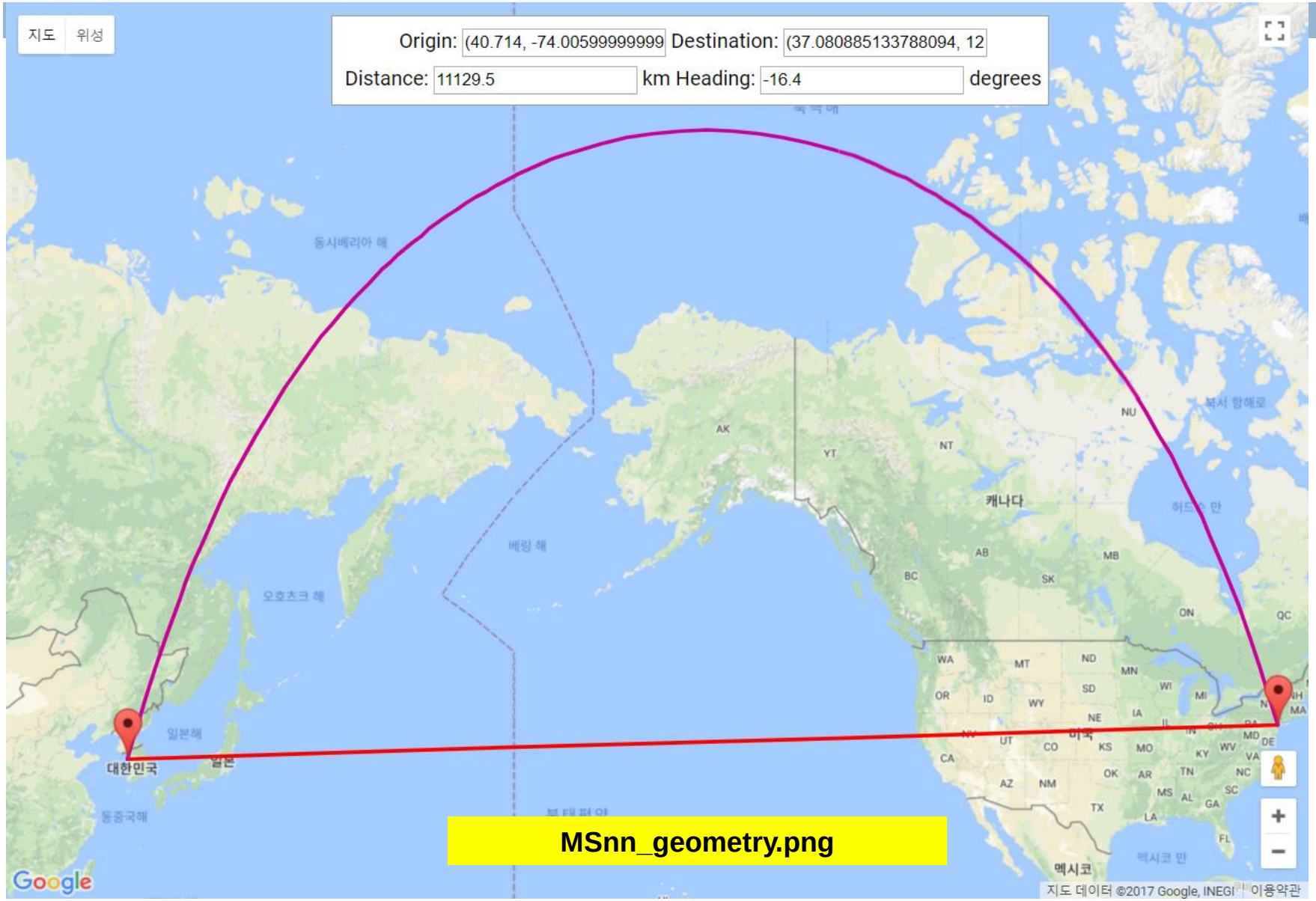
A6.4 Google map library: geometry

```
function update() {
  var path = [marker1.getPosition(), marker2.getPosition()];
  poly.setPath(path);
  geodesicPoly.setPath(path);
  var heading = google.maps.geometry.spherical.computeHeading(path[0], path[1]);
  var distance = google.maps.geometry.spherical.computeDistanceBetween(path[0], path[1]);
  document.getElementById('heading').value = heading.toFixed(1);
  document.getElementById('distance').value = (distance/1000.0).toFixed(1);
  document.getElementById('origin').value = path[0].toString();
  document.getElementById('destination').value = path[1].toString();
}
```

[Result] A6. library: geometry



[Result] A6. library: geometry



A7.1 Google map library: places

```
<!DOCTYPE html>
<html>
  <head>
    <title>Place searches</title>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
    <meta charset="utf-8">
    <script src="https://maps.googleapis.com/maps/api/js?libraries=places&callback=initMap"
    async defer></script>

    <style>
      html, body {
        height: 100%;
        margin: 0;
        padding: 0;
      }
      #map {
        height: 100%;
      }
      #floating-panel {
        position: absolute;
        top: 10px;
        left: 15%;
        background-color: #ff0;
        padding: 5px;
        border: 1px solid #999;
        text-align: center;
        font-family: 'Roboto','sans-serif';
        line-height: 30px;
        padding-left: 10px;
        position: absolute;
        z-index: 5;
      }

    </style>
```

A7.2 Google map library: places

```
<script>
  var map;
  var markers =[];
  var infowindow;
  var service;
  var target_location; // name of check-up location
  var e323 = {lat: 35.249164, lng: 128.901881}; // 35.249164, 128.901881)
  var annarbor = {lat: 42.318388, lng: -83.706521};//windwood dr. , ann arbor, mi
  //var pyrmont = {lat: -33.867, lng: 151.195};

  function initMap() {

    target_location=e323;    // annarbor

    map = new google.maps.Map(document.getElementById('map'), {
      center: target_location, //target_location, //pyrmont,ann arbor,
      zoom: 15
    });

    infowindow = new google.maps.InfoWindow();

    show_places('cafe');

    var onChangeHandler = function() {
      var obj = JSON.parse(document.getElementById('target').value);
      //console.log(obj.lat);
      target_location = {lat:obj.lat, lng:obj.lng};
      // select type of places
      show_places('cafe');
    };

    document.getElementById('target').addEventListener('change', onChangeHandler);
```

Types of places

accounting
airport
amusement_park
aquarium
art_gallery
atm
bakery
bank
bar
beauty_salon
bicycle_store
book_store
bowling_alley
bus_station
cafe
campground
car_dealer
car_rental
car_repair
car_wash

casino
cemetery
church
city_hall
clothing_store
convenience_store
courthouse
dentist
department_store
doctor
electrician
electronics_store
embassy
establishment
finance
fire_station
florist
food
funeral_home
furniture_store

gas_station
general_contractor
grocery_or_supermarket
gym
hair_care
hardware_store
health
hindu_temple
home_goods_store
hospital
insurance_agency
jewelry_store
laundry
lawyer
library
liquor_store
local_government_office
locksmith
lodging
meal_delivery

A7.3 Google map library: places

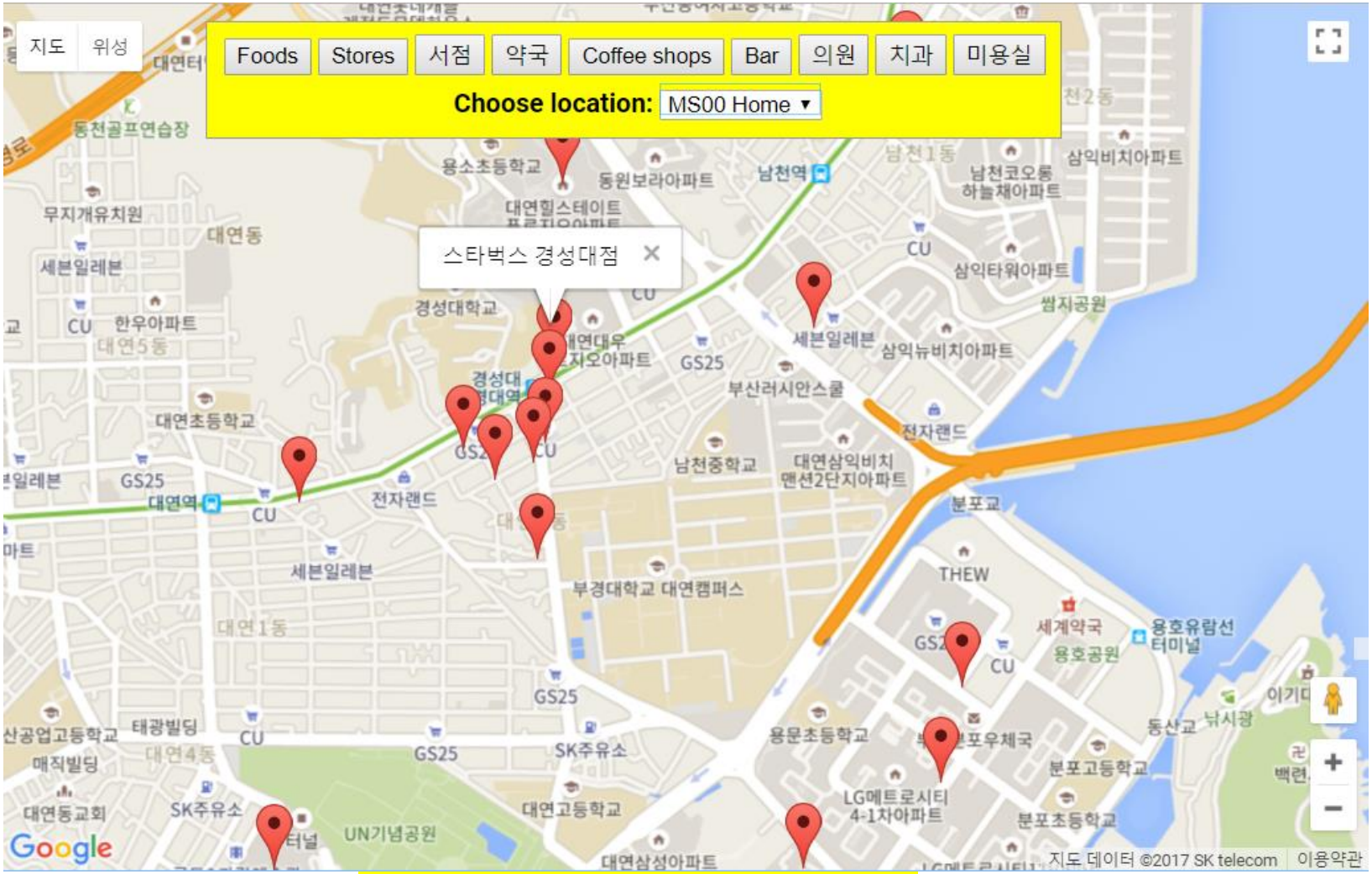
```
function show_places(whatplace) {
    clearMarkers();  // clear all previous markers

    map = new google.maps.Map(document.getElementById('map'), {
      center: target_location, //target_location, //pyrmont,ann arbor,
      zoom: 15
    });

    service = new google.maps.places.PlacesService(map);
    service.nearbySearch({
      location: target_location, //target_location, //annarbor, //pyrmont,
      radius: 2000,
      types: [whatplace]  // store, food
    }, callback);
}

function callback(results, status) {

  if (status === google.maps.places.PlacesServiceStatus.OK) {
    for (var i = 0; i < results.length; i++) {
      createMarker(results[i]);
    }
  }
}
```

A7.4 Google map library: places

```
function createMarker(place) {
  var placeLoc = place.geometry.location;
  var marker = new google.maps.Marker({
    map: map,
    position: place.geometry.location
  });
  markers.push(marker);

  google.maps.event.addListener(marker, 'click', function() {
    infowindow.setContent(place.name);
    infowindow.open(map, this);
  });
}
```

```
// Sets the map on all markers in the array.
function setMapOnAll(map) {
  for (var i = 0; i < markers.length; i++) {
    markers[i].setMap(map);
  }
}

// Removes the markers from the map, but keeps them in the array.
function clearMarkers() {
  setMapOnAll(null);
}
// Shows any markers currently in the array.
function showMarkers() {
  setMapOnAll(map);
}
// Deletes all markers in the array by removing references to them.
function deleteMarkers() {
  clearMarkers();
  markers = [];
}
```

[Result] A7. Google map : places

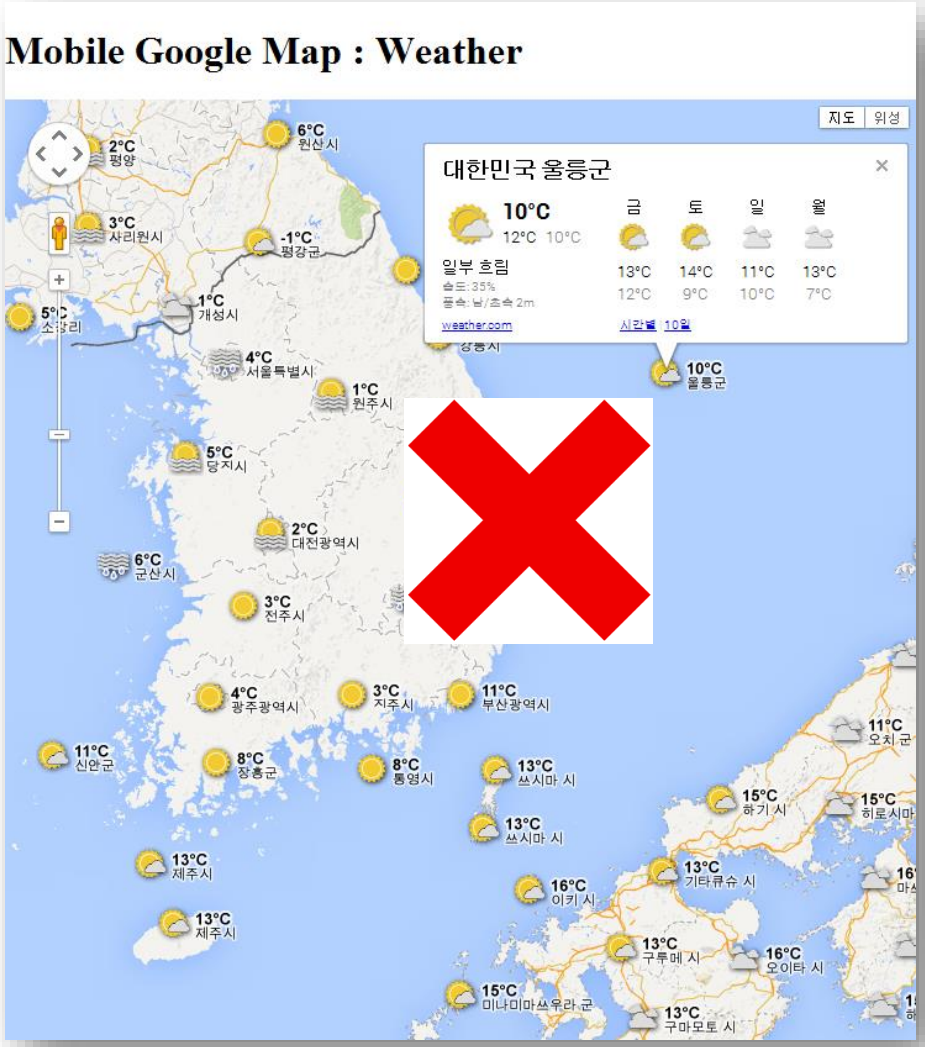


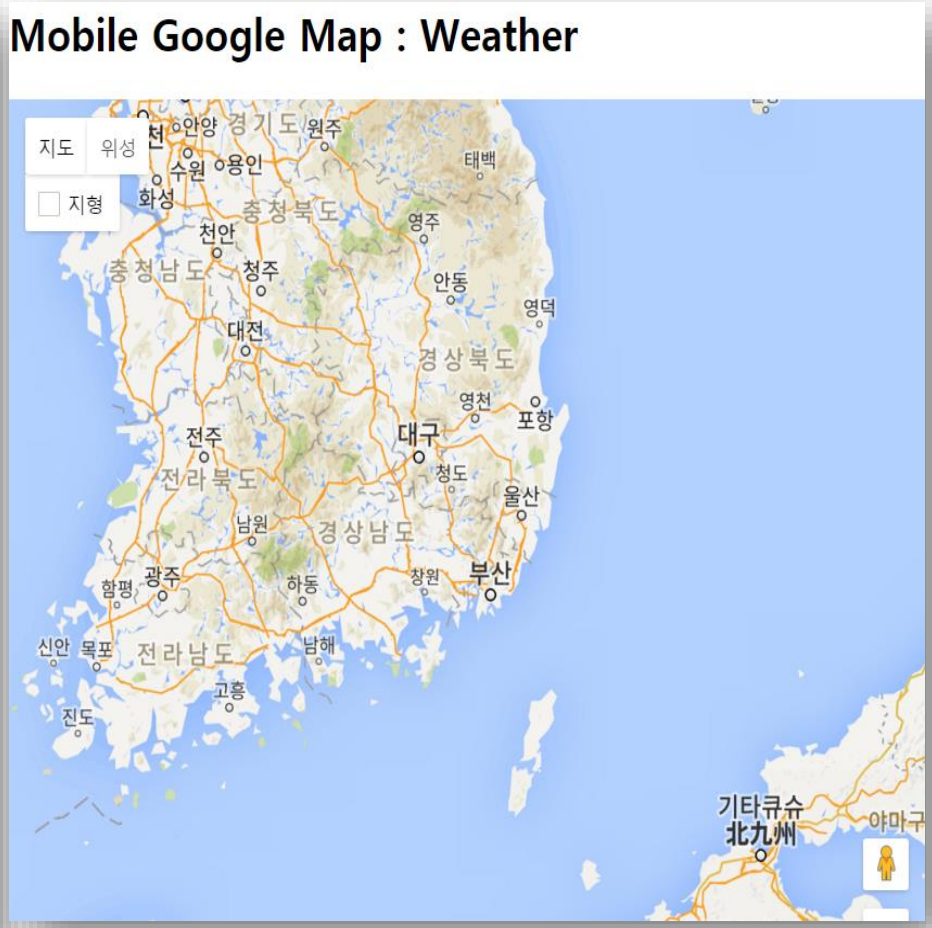
집 주변 **places** 추가.
MSnn_MyHome.png

A8. Weather with cloud

```
<script
   src="http://maps.googleapis.com/maps/api/js?v=3.exp&language=ko
   &sensor=false&libraries=weather">
</script>

<script>
var e323=new google.maps.LatLng(35.249164, 128.901881);  // E323

function initialize()
{
  var mapProp = {
    center:e323,
    zoom:7,
    mapTypeId:google.maps.MapTypeId.ROADMAP
  };

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

  var weatherLayer = new google.maps.weather.WeatherLayer ({
   temperatureUnits:
   google.maps.weather.TemperatureUnit.CELCIUS
  });
  weatherLayer.setMap(map);

  var cloudLayer = new google.maps.weather.CloudLayer();
  cloudLayer.setMap(map);
```

[Result] A8. Weather with cloud

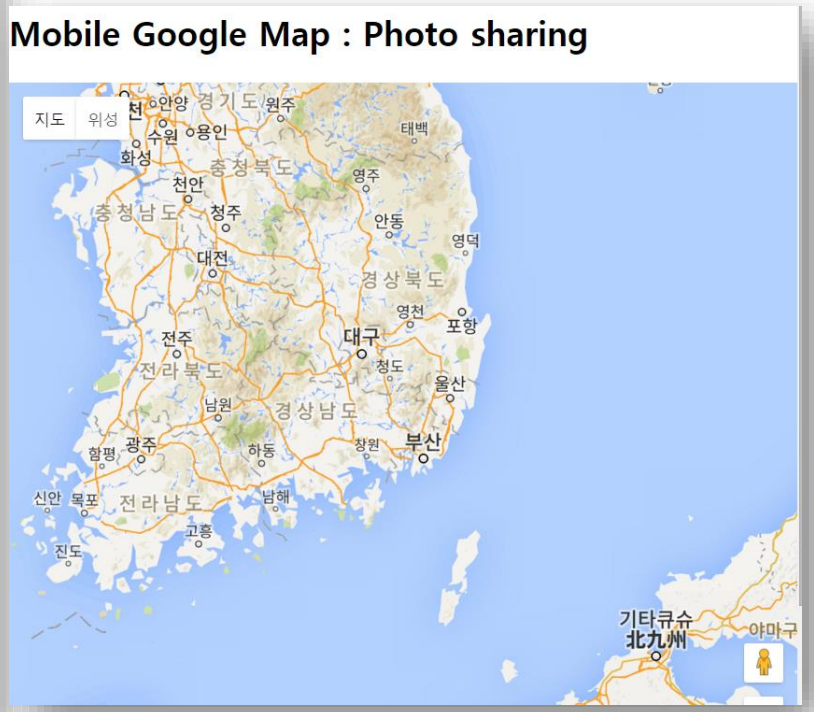




A9. Photo sharing with Panoramio

```
<script
   src="http://maps.googleapis.com/maps/api/js?v=3.exp&language=ko
   &sensor=false&libraries=panoramio">
</script>

<script>
var e323=new google.maps.LatLng(35.249164, 128.901881);  // E323

function initialize()
{
  var mapProp = {
    center:e323,
    zoom:7,
    mapTypeId:google.maps.MapTypeId.ROADMAP
  };

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

  var panoramioLayer = new google.maps.panoramio.PanoramioLayer({
    suppressInfoWindows:false
  });
  panoramioLayer.setMap(map);
```

[Result] A9. Photo sharing with Panoramio



Google Map Javascript API

Challenge stage

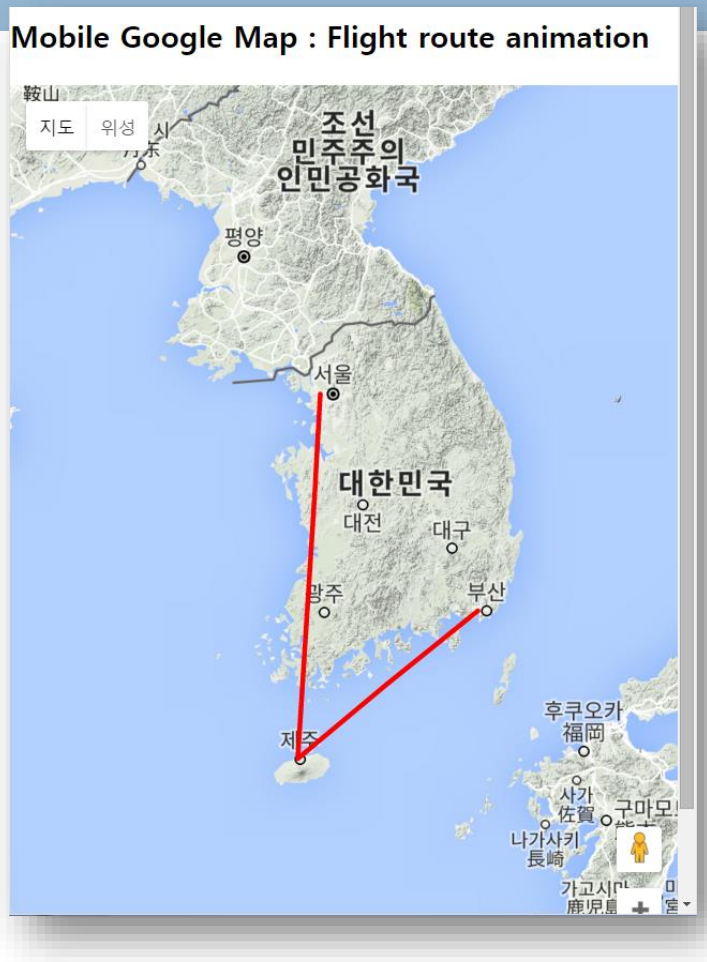
3.2.1 Add Polyline

```
function initMap() {
  var mapProp = {
    zoom: 6,
    center: new google.maps.LatLng(35.840212, 127.125173),  // 전주
    mapTypeId: google.maps.MapTypeId.TERRAIN
  };

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

// the path of Korean flight between Gimpo, Jeju, and Busan, Korea.
  var flightPlanCoordinates = [
    new google.maps.LatLng(37.558909, 126.803083),  // Gimpo airport
    new google.maps.LatLng(33.507758, 126.491434),  // Jeju airport
    new google.maps.LatLng(35.173431, 128.946659),  // Gimhae airport
  ];
  var flightPath = new google.maps.Polyline({
    path: flightPlanCoordinates,
    geodesic: true,
    strokeColor: '#FF0000',
    strokeOpacity: 1.0,
    strokeWeight: 3
  });

  flightPath.setMap(map);
}
```

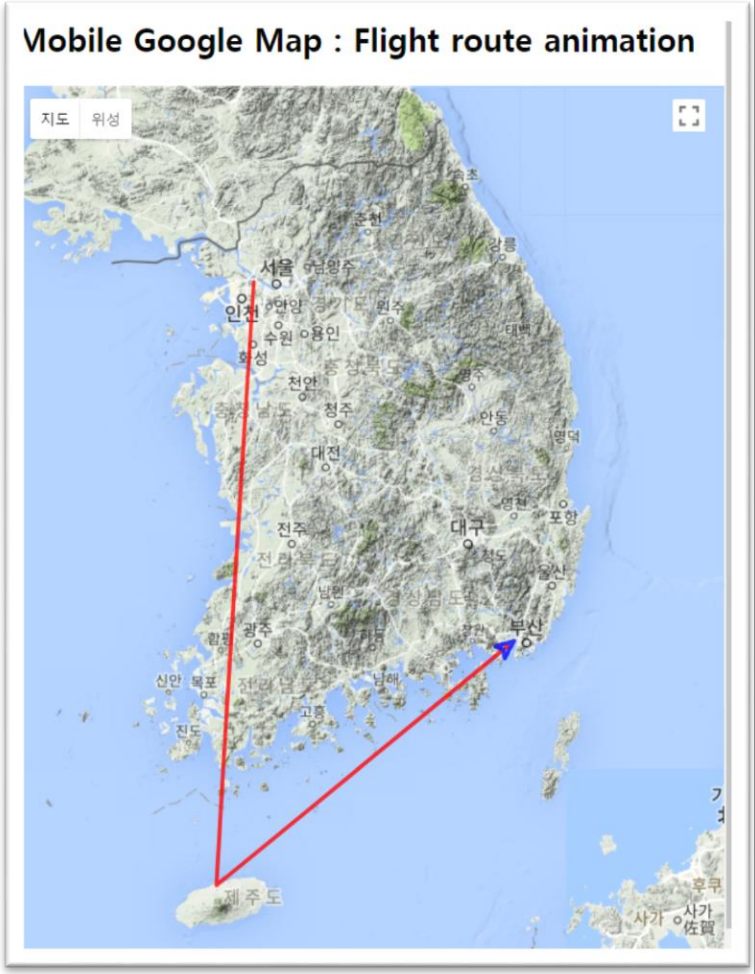
- path - specifies several latitude/longitude coordinates for the line
- strokeColor - specifies a hexadecimal color for the line (format: "#FFFFFF")
- strokeOpacity - specifies the opacity of the line (a value between 0.0 and 1.0)
- strokeWeight - specifies the weight of the line's stroke in pixels
- editable - defines whether the line is editable by users (true/false)

Challenge C1. Mobile version

```
<script>
var map;
var flightPath;

function addPolyline() {
// the path of Korean flight between Gimpo, Jeju, and Busan, Korea.
  var flightPlanCoordinates = [
    new google.maps.LatLng(37.558909, 126.803083),  // Gimpo airport
    new google.maps.LatLng(33.507758, 126.491434),  // Jeju airport
    new google.maps.LatLng(35.173431, 128.946659),  // Gimhae airport
  ];
  flightPath = new google.maps.Polyline({
    path: flightPlanCoordinates,
    geodesic: true,
    strokeColor: '#FF0000',
    strokeOpacity: 1.0,
    strokeWeight: 3
  });

  flightPath.setMap(map);
}

function initialize() {
  var mapProp = {
    zoom: 6,
    center: new google.maps.LatLng(35.840212, 127.125173),  // 전주
    mapTypeId: google.maps.MapTypeId.TERRAIN
  };

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

  addPolyline();
}


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

</script>
```



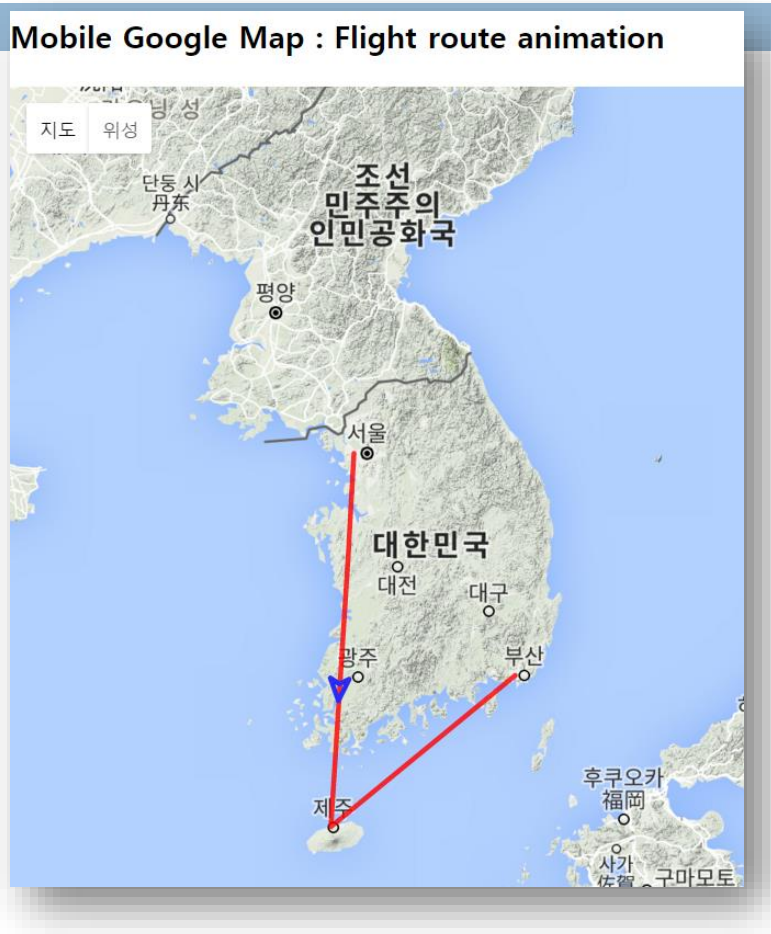
Challenge C2.1
Route animation




```
<script>
var map;
var flightPath;
// Defining arrow symbol
var arrowSymbol = {
   strokeColor: '#00F',
   scale: 3,
   path: google.maps.SymbolPath.FORWARD_CLOSED_ARROW
};
function addAnimatedPolyline() {
// the path of Korean flight between Gimpo, Jeju, and Busan, Korea.
  var flightPlanCoordinates = [
    new google.maps.LatLng(37.558909, 126.803083),  // Gimpo airport
    new google.maps.LatLng(33.507758, 126.491434),  // Jeju airport
    new google.maps.LatLng(35.173431, 128.946659),  // Gimhae airport
  ];
  flightPath = new google.maps.Polyline({
    path: flightPlanCoordinates,
    //geodesic: true,
    icons: [{
      icon: arrowSymbol,
      offset: '100%'
    }],
    strokeColor: '#FF0000',
    strokeOpacity: 0.8,
    strokeWeight: 3,
    map: map
  });
  // Calling the arrow animation function
  animateArrow();
}
```

Challenge C2.2 Route animation

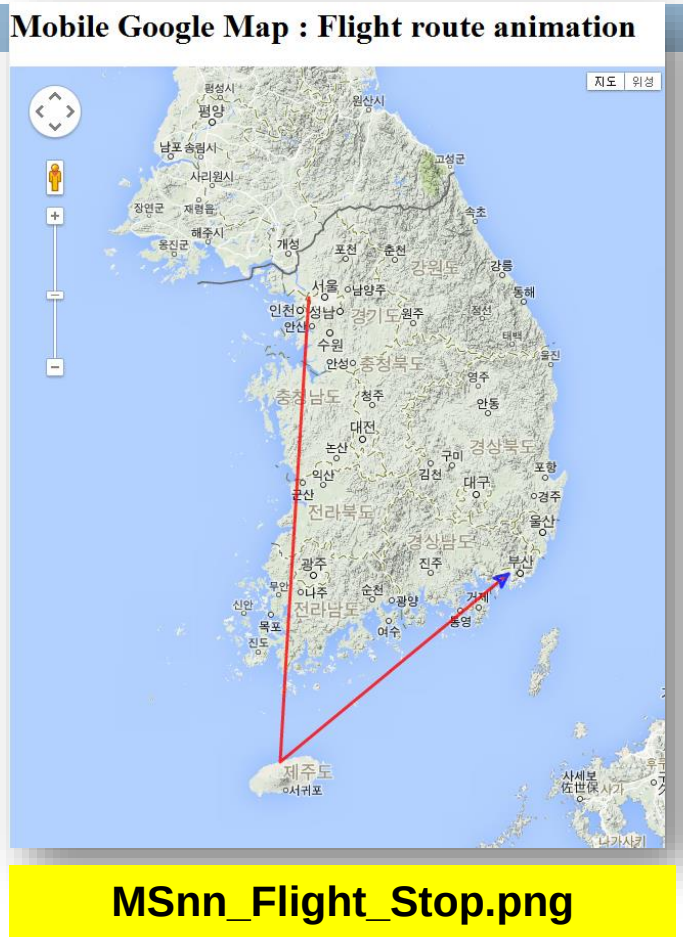
```
function animateArrow() {
  var counter = 0;
  var intervalID=window.setInterval(function() {
  counter = (counter + 1) % 200;
  var arrows = flightPath.get('icons');
  arrows[0].offset = (counter / 2) + '%';
  flightPath.set('icons', arrows);
  }, 50);
}

function initialize() {
 var mapProp = {
  zoom: 7,
  center: new google.maps.LatLng(35.840212, 127.125173),  // 전주
  mapTypeId: google.maps.MapTypeId.TERRAIN
 };

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

  addAnimatedPolyline();

}
```



Challenge C3. Stop animation at home

```
function animateArrow() {
    var counter = 0;
    var intervalID=window.setInterval(function() {
    counter = (counter + 1) % 200;
    var arrows = flightPath.get('icons');
    arrows[0].offset = (counter / 2) + '%';
    flightPath.set('icons', arrows);
    }, 50);
}
```

```
function animateArrow() {
    var counter = 0;
    var intervalID=window.setInterval(function() {
    counter = (counter + 1) % 201;
    var arrows = flightPath.get('icons');
    arrows[0].offset = (counter / 2) + '%';
    flightPath.set('icons', arrows);
    if(counter==200){
    clearInterval(intervalID);
    }
    }, 50);
}
```



MSnn_Flight_Stop.png

Challenge C4.1 Animation by distance (%)

```
 6   <meta charset="utf-8" />
 7   <title> Mobile Simulation: Google Map </title>
 8 - <style type="text/css">
 9       html { height: 100% }
10       body { height: 100%; margin: 0; }
11       #googleMap { width: 100%; height: 100%; }
12   </style>
13   <script
14       src="http://maps.googleapis.com/maps/api/js?v=3.exp&language=ko
15       &sensor=false&libraries=geometry">
16   </script>
17
18 - <script>
19   // global variables
20   var map;
21   var flightPlanCoordinates;
22   var flightPath;
23   var pointDistances;
24   var id;
25   // Defining arrow symbol
26 - var arrowSymbol = {
27       strokeColor: '#00F',
28       scale: 4,
29       path: google.maps.SymbolPath.FORWARD_CLOSED_ARROW
30   };
```

Challenge C4.2 Animation by distance (%)

```
function initialize() {
  var mapProp = {
    zoom: 7,
    center: new google.maps.LatLng(35.840212, 127.125173),  // 전주
    mapTypeId: google.maps.MapTypeId.TERRAIN
  };

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

  // the path of Korean flight between Gimpo, Jeju, and Busan, Korea.
  flightPlanCoordinates = [
    new google.maps.LatLng(37.558909, 126.803083),  // Gimpo airport
    new google.maps.LatLng(33.507758, 126.491434),  // Jeju airport
    new google.maps.LatLng(35.173431, 128.946659),  // Gimhae airport
  ];

  // point distances from beginning in %

    map.setCenter(flightPlanCoordinates[0]);
    var sphericalLib = google.maps.geometry.spherical;

    var wholeDist;
    pointDistances = [0];  // 1st distance = 0
    for (var i = 1; i < flightPlanCoordinates.length; i++) {
        var partialDistance = sphericalLib.computeDistanceBetween(
                                            flightPlanCoordinates[i],
                                            flightPlanCoordinates[i - 1]);
        pointDistances[i] = pointDistances[i - 1] + partialDistance;
        console.log('pointDistances[' + i + ']: ' + pointDistances[i]);
    }
    // convert to percentage offset
    wholeDist = pointDistances[pointDistances.length - 1];
    for (var i = 0; i < flightPlanCoordinates.length; i++) {
        pointDistances[i] = 100 * pointDistances[i] / wholeDist;
        console.log('pointDistances[' + i + ']: ' + pointDistances[i]);
    }
    console.log('whole distance: ' + wholeDist);

  //pointDistances=evalDistanceRatio();
  addAnimatedPolyline();
}
```

Challenge C4.3 Animation by distance (%)

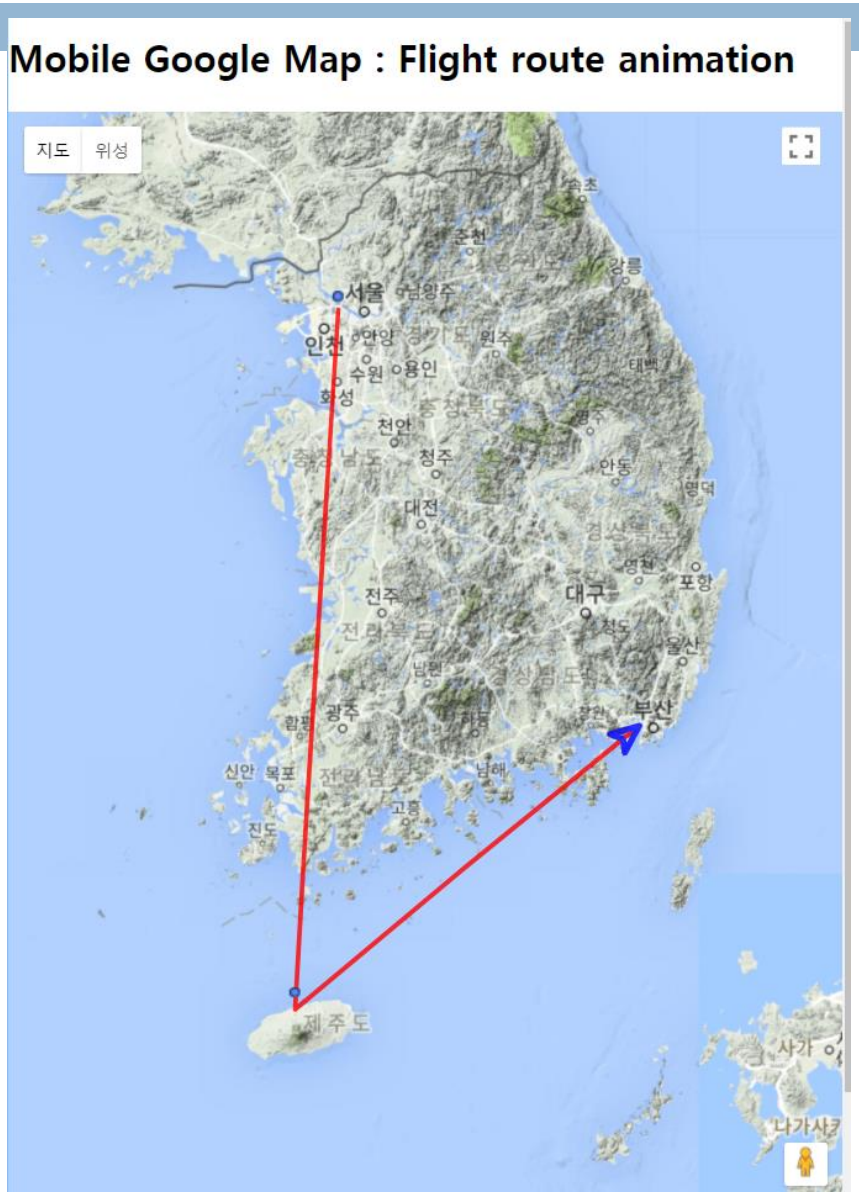
```
function addAnimatedPolyline() {
  //pointDistances=evalDistanceRatio();

  flightPath = new google.maps.Polyline({
    path: flightPlanCoordinates,
    geodesic: true,
    icons: [{
      icon: arrowSymbol,
      offset: '100%'
    }],
    strokeColor: '#FF0000',
    strokeOpacity: 0.8,
    strokeWeight: 3,
    map: map
  });
  // Calling the arrow animation function
  animateArrow();
}

/*
function evalDistanceRatio() {
    // point distances from beginning in %
    var sphericalLib = google.maps.geometry.spherical;
    map.setCenter(flightPlanCoordinates[0]);
    var wholeDist;
    pointDistances[0] = 0;
    for (var i = 1; i < flightPlanCoordinates.length; i++) {
        var partialDistance = sphericalLib.computeDistanceBetween(
                                              flightPlanCoordinates[i],
                                              flightPlanCoordinates[i - 1]);
        pointDistances[i] = pointDistances[i - 1] + partialDistance;
        console.log('pointDistances[' + i + ']: ' + pointDistances[i]);
    }
    // convert to percentage offset
    wholeDist = pointDistances[pointDistances.length - 1];
    for (var i = 0; i < flightPlanCoordinates.length; i++) {
        pointDistances[i] = 100 * pointDistances[i] / wholeDist;
        console.log('pointDistances[' + i + ']: ' + pointDistances[i]);
    }
    console.log('whole distance: ' + wholeDist);
    return pointDistances;
}
*/
```

Challenge C4.4 Animation by distance (%)

```
function animateArrow() {
    /* var counter = 0;
    pointDistances = evalDistanceRatio();

    var intervalID=window.setInterval(function() {
    counter = (counter + 1) % 201;
    var arrows = flightPath.get('icons');
    arrows[0].offset = (counter / 2) + '%';
    flightPath.set('icons', arrows);
    if(counter==200){
        clearInterval(intervalID);
    }
    }, 50); */
```

```
    var count = 0;
    var offset;
    var sentiel = -1;

    id = window.setInterval(function () {
        count = (count + 1) % 201;
        offset = count /2;

        for (var i = pointDistances.length-1; i > sentiel; i--) {
            if (offset > pointDistances[i]) {
                console.log('create marker');
                var marker = new google.maps.Marker({
                    icon: {
                        url:"https://maps.gstatic.com/intl/en_us/mapfiles/markers2/measle_blue.png",
                        size: new google.maps.Size(16,16),
                        anchor: new google.maps.Point(4,4)
                    },
                    //icon: 'image/pin_s_tr.png',
                    position: flightPath.getPath().getAt(i),
                    title: flightPath.getPath().getAt(i).toUrlValue(6),
                    animation:google.maps.Animation.BOUNCE,
                    map: map
                });

                sentiel++;
                break;
            }
        }
        // we have only one icon
        var icons = flightPath.get('icons');
        icons[0].offset = (offset) + '%';
        flightPath.set('icons', icons);

        if (flightPath.get('icons')[0].offset == "100%") {
            window.clearInterval(id);
        }

    }, 100);
}
```

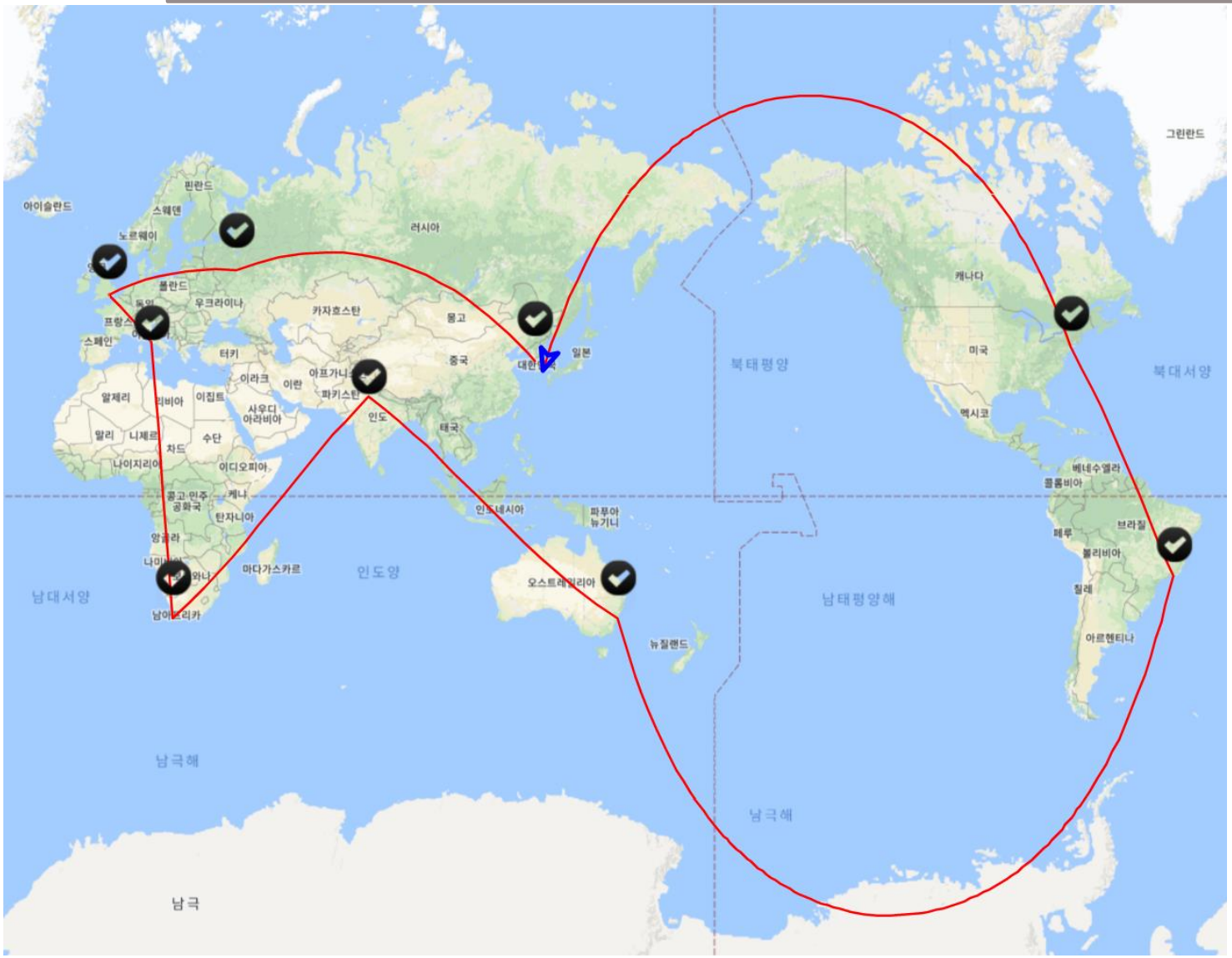
Challenge C4.5 Animation by distance (%)



도전과제

1. 아래 그림과 같이 세계 일주 여행 지도를 만드시오.
2. 꼭 가고 싶은 곳의 위치를 4 곳 이상 찾아서 추가하시오.
3. 구굴맵의 다양한 기능을 활용하시오. (아이콘, 애니메이션 변경)

html 파일을 **MSnn_Google_Travel.html** 로 저장하시오.



도전과제 - code#1

```
<meta name="viewport" content="initial-scale=1.0, user-scalable=no" />

<meta charset="utf-8" />
<title> Mobile Simulation: Google Map </title>
<style type="text/css">
    html {
        height: 100%
    }

    body {
        height: 100%;
        margin: 0;
    }

    #googleMap {
        width: 100%;
        height: 100%;
    }

</style>
<script src="http://maps.googleapis.com/maps/api/js?language=ko&libraries=geometry">


</script>
```

도전과제 - code#2

```
<script>
    var line;

    var map;
    var pointDistances;
    var airport = ["인천공항", "모스코바", "런던 히드로", "로마 레오나르도 다빈치",
        "케이프타운", "뉴델리", "시드니", "리오데자네이로", "뉴욕JFK", "김해공항"
    ];

    function initialize() {
        var mapOptions = {
            center: new google.maps.LatLng(2.881766, 101.626877),
            zoom: 1,
            mapTypeId: google.maps.MapTypeId.HTBRID
        };

        map = new google.maps.Map(document.getElementById('googleMap'), mapOptions);

        var lineCoordinates = [
            new google.maps.LatLng(37.448987, 126.451371), // Inchon airport
            new google.maps.LatLng(55.758679, 37.620822), // Moscova
            new google.maps.LatLng(51.470045, -0.454344), // London
            new google.maps.LatLng(41.801392, 12.243144), // Roma
            new google.maps.LatLng(-33.971405, 18.602069), // Cape Town
            new google.maps.LatLng(28.556167, 77.099958), // New Deli
            new google.maps.LatLng(-33.939830, 151.175271), // Sidney
            new google.maps.LatLng(-22.813375, -43.249450), // Rio
            new google.maps.LatLng(40.756916, -73.988200), // NY
            new google.maps.LatLng(35.173431, 128.946659), // Gimhae airport
        ];

        map.setCenter(lineCoordinates[0]);
```

도전과제 - code#3

```
// point distances from beginning in %
var sphericalLib = google.maps.geometry.spherical;

pointDistances = [0];
for (var i = 1; i < lineCoordinates.length; i++) {
    var partialDistance = sphericalLib.computeDistanceBetween(
        lineCoordinates[i],
        lineCoordinates[i - 1]);
    pointDistances[i] = pointDistances[i - 1] + partialDistance;
    console.log('pointDistances[' + i + ']: ' + pointDistances[i]);
}
// convert to percentage offset
wholeDist = pointDistances[pointDistances.length - 1];
for (var i = 0; i < lineCoordinates.length; i++) {
    pointDistances[i] = 100 * pointDistances[i] / wholeDist; // unit:%
    console.log('pointDistances[' + i + ']: ' + pointDistances[i]);
}
console.log('whole distance: ' + wholeDist);
```

도전과제 - code#4

```
    // Defining arrow symbol
    var arrowSymbol = {
        strokeColor: '#00F',
        scale: 4,
        path: google.maps.SymbolPath.FORWARD_CLOSED_ARROW
    };

    line = new google.maps.Polyline({
        path: lineCoordinates,
        geodesic: true,
        strokeColor: '#FF0000',
        strokeOpacity: 1.0,
        strokeWeight: 2,
        icons: [{
            icon: arrowSymbol,
            offset: '100%'
        }],
        map: map
    });

    animateCircle();
}

var image = {
    url: 'image/mark32.png',
    // This marker is 20 pixels wide by 32 pixels tall.
    size: new google.maps.Size(48, 48),
    // The origin for this image is 0,0.
    origin: new google.maps.Point(0, 0),
    // The anchor for this image is the base of the flagpole at 0,32.
    anchor: new google.maps.Point(16, 32)
};
```

도전과제 - code#5

```
var id;  // timer object
function animateCircle() {
    var count = 0;
    var offset;
    var sentiel = -1;

    id = window.setInterval(function() {
        count = (count + 1) % 401;
        offset = count / 4;

        for (var i = pointDistances.length - 1; i > sentiel; i--) {
            if (offset > pointDistances[i]) {
                console.log('create marker');
                var marker = new google.maps.Marker({
                    icon: image,
                    position: line.getPath().getAt(i),
                    title: airport[i] + " : " + line.getPath().getAt(i).toUrlValue(5),
                    animation: google.maps.Animation.BOUNCE,
                    map: map
                });

                sentiel++;
                break;
            }
        }

        // we have only one icon
        var icons = line.get('icons');
        icons[0].offset = (offset) + '%';
        line.set('icons', icons);

        if (line.get('icons')[0].offset == "100%") {
            window.clearInterval(id);
        }

    }, 20);
}

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

과제10. msnn_rpt10.zip

[실습과제10] Google Map apps
[1] 실습 결과 그림 4장 저장.
[2] MSnn_Transit.png, MSnn_geometry.png
MSnn_MyHome.png, MSnn_Flight_Stop.png

[3.1] 나의 세계 일주 여행 지도를 만드시오.
[3.2] 꼭 가고 싶은 곳의 위치를 4 곳 이상 찾아서 추가하시오.
[3.3] 구굴맵의 다양한 기능을 활용하시오. (아이콘, 애니메이션 변경)

html 파일을 MSnn_Google_Travel.html 로 저장하시오.

**** MSnn_Rpt10.zip 으로 압축해서 제출하시오.

[제출파일] msnn_rpt10.zip (11월21일 오후 6시 마감)

html 파일과 사용된 그림을 압축하여 이메일로 “msnn_rpt10” 제목으로 제출

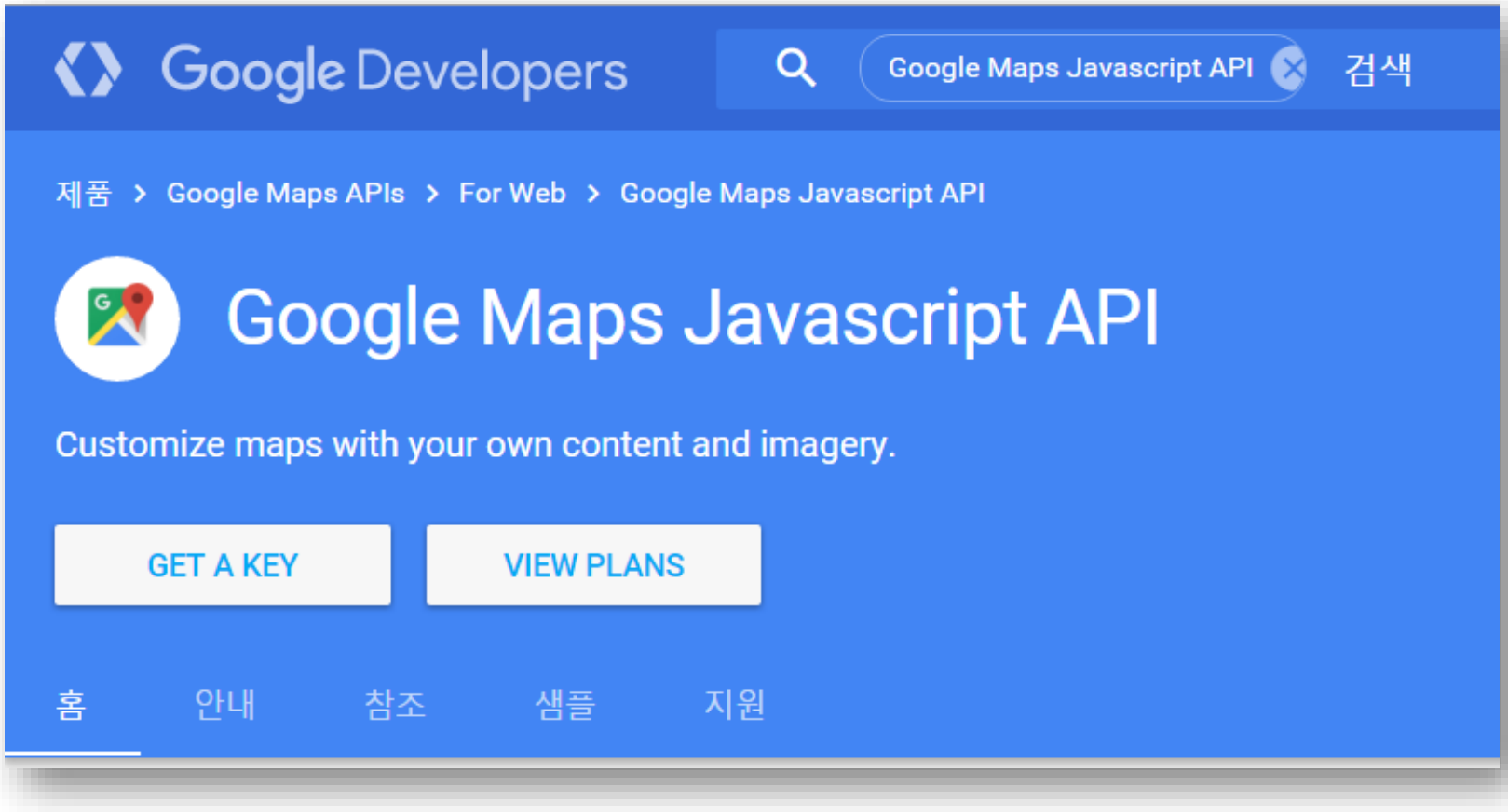
Email : chaos21c@gmail.com

Result


그린란드
아이슬란드
노르웨이
스웨덴
핀란드
러시아
영국
프랑스
스페인
이탈리아
폴란드
우크라이나
카자흐스탄
몽고
중국
일본
북한
터키
이라크
이란
아프가니스탄
파키스탄
인도
태국
사우디 아라비아
알제리
리비아
이집트
니제르
차드
수단
나이지리아
에티오피아
케냐
콩고 민주 공화국
탄자니아
앙골라
나미비아
보츠와나
남아프리카
마다가스카르
베네수엘라
콜롬비아
브라질
페루
볼리비아
칠레
아르헨티나
인도네시아
파푸아 뉴기니
오스트레일리아
뉴질랜드
북대서양
남대서양
인도양
남극해

API Key:

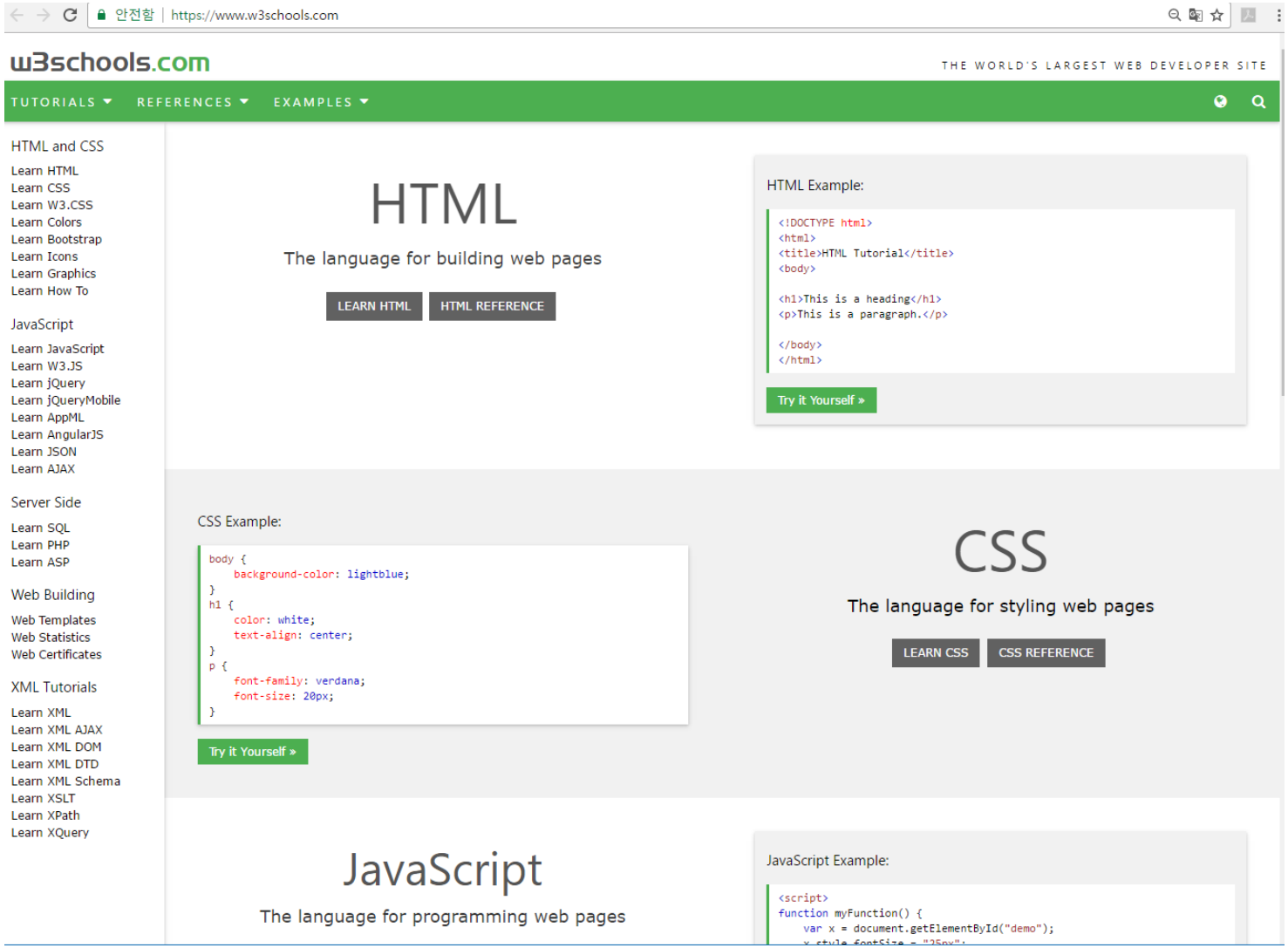
Using Google Map API V3



교재 WEB 강의 소개

http://webprogramming.co.kr

관련 WEB 강의 소개 – w3schools.com



http://www.w3schools.com