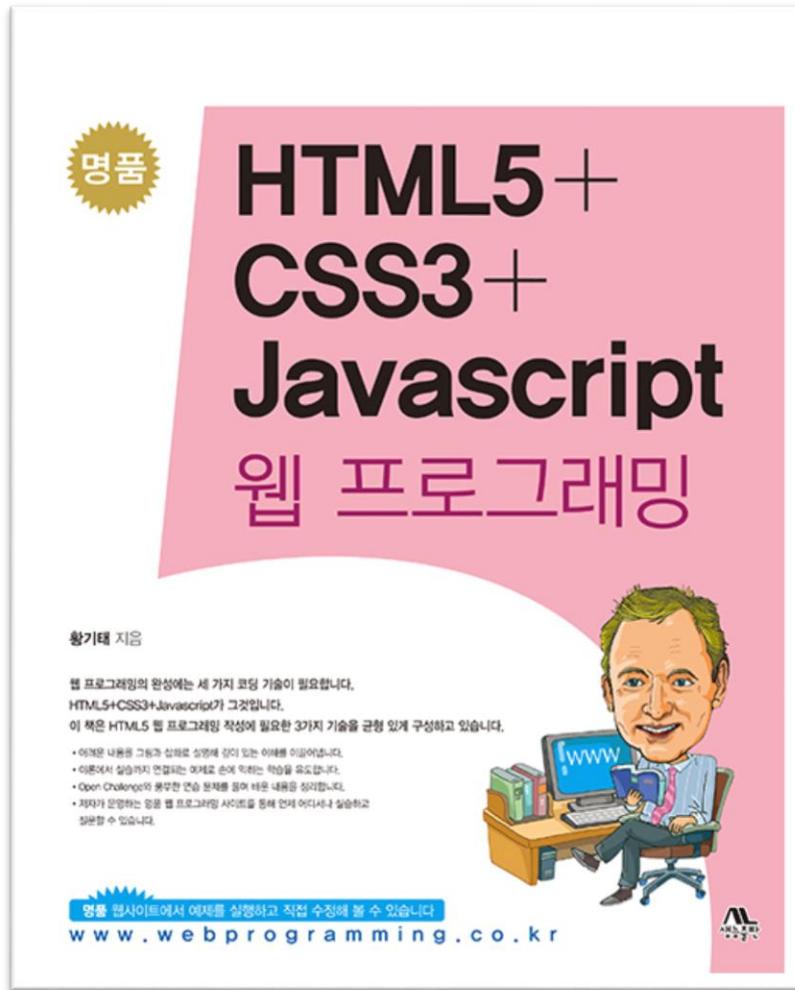


Mobile Simulation



A screenshot of the w3schools.com website. The search bar at the top says "Search w3schools.com:". Below it is a Google Custom Search bar. The main content area features the w3schools logo and the text "JavaScript". Below this are two buttons: "JavaScript Tutorial" and "JavaScript Reference". Further down, another section for "JQuery" is shown with "JQuery Tutorial" and "JQuery Reference" buttons.



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: App**
- **wk14 :**
- **wk15 : Final exam.**



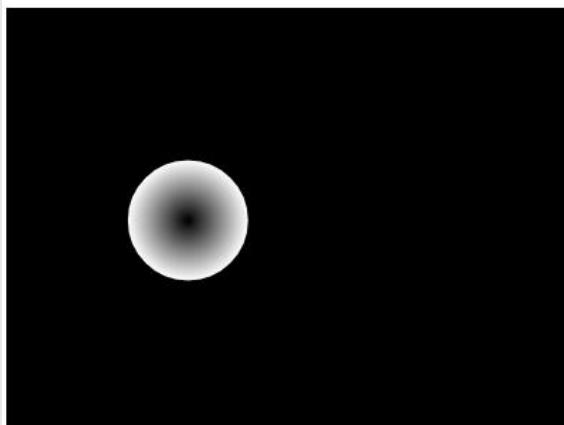
도전과제

1. 아래 그림과 같이 애니메이션으로 black hole SVG를 만드시오.
 2. Black hole은 그래디언트를 적용해서 제작.
 3. Black hole SVG를 드롭 존으로 설정하시오.
 4. 여러가지 물건들을 사라지게 프로그래밍하시오.
- Html 파일을 [**MSnn_SVG_DD.html**](#)로 저장하시오.

Learning Drag & Drop :



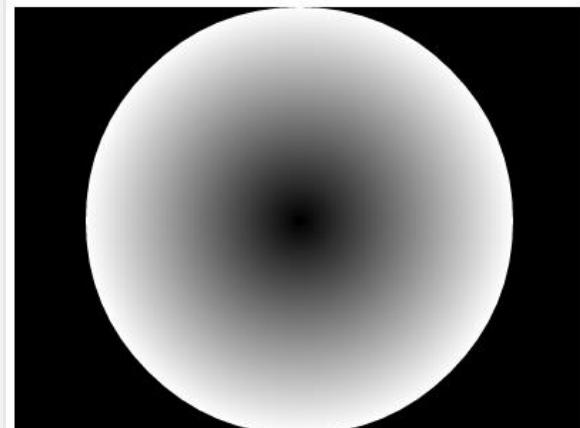
SVG blackhole: drop your things



Learning Drag & Drop :



SVG blackhole: drop your things



과제08. msnn_rpt08.zip

5

[실습과제08] black hole SVG with D&D

- [1] black hole SVG 제작.
- [2] Drag&drop 프로그래밍 추가.
- [3] 파일명: MSnn_SVG_DD.html
- [4] 가점: SVG 제작 및 javascript 프로그래밍 적용 능력.

**** MSnn_SVG_DD.html 파일 및 관련 파일(image 등..)을
MSnn_Rpt08.zip 으로 압축해서 제출하시오.

[제출파일] [msnn_rpt08.zip](#) (11월7일 오후 6시 마감)

[html 파일과 사용된 그림을 압축하여 이메일로 “msnn_rpt08” 제목으로 제출](#)

Email : chaos21c@gmail.com

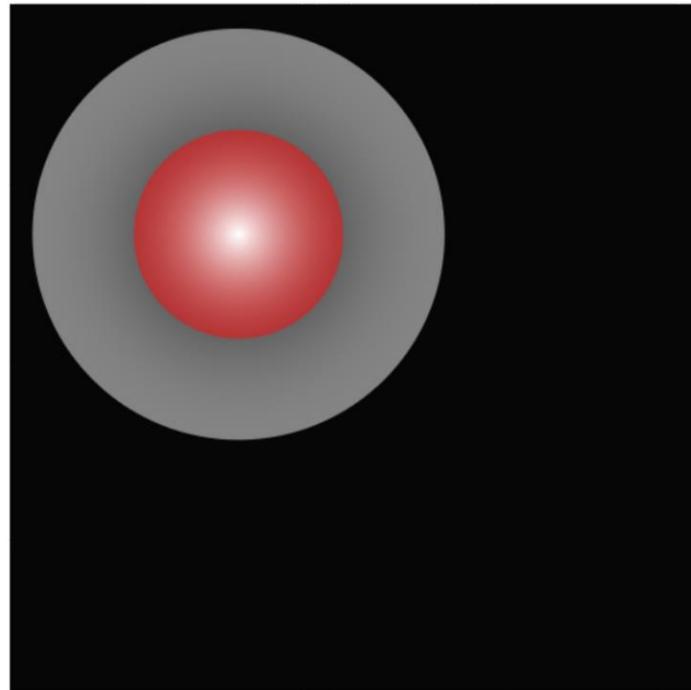
Best report



휴대폰을 블랙홀 중앙으로 던지세요.



블랙홀에 정확히 물건을 넣으세요



CHAPTER 13

Google Map I. Basic

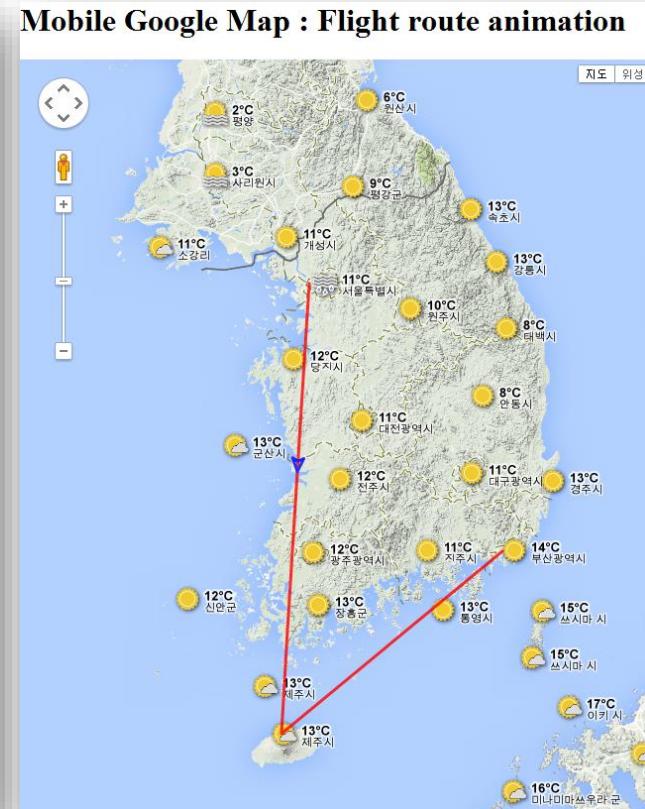
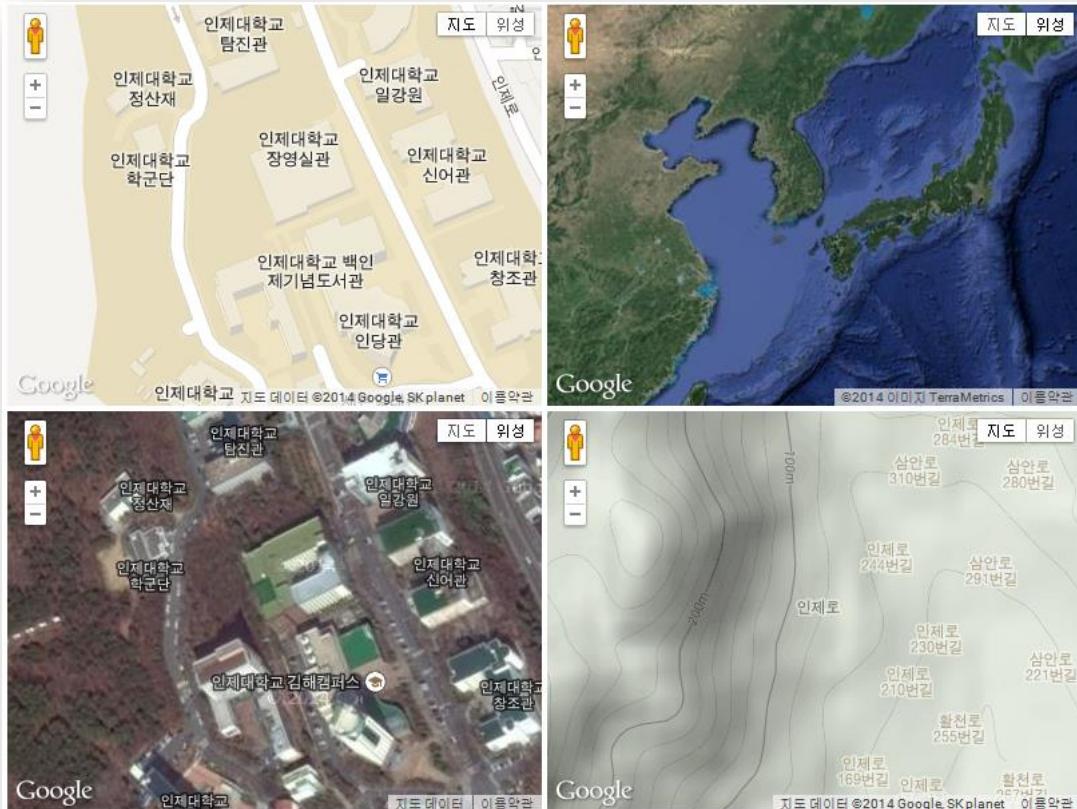
Develop apps using Google Map Api V3 & HTML5 Geolocation.

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

API = Application programming interface.

Target #1:

Using Google Map API V3



API Key: Using Google Map API V3

The screenshot shows the Google Developers website with a blue header. On the left is the Google Developers logo, followed by a search bar containing "Google Maps Javascript API" with a magnifying glass icon and an "X" button. To the right of the search bar is a "검색" (Search) button. Below the header, the breadcrumb navigation reads "제품 > Google Maps APIs > For Web > Google Maps Javascript API". The main content area features a large circular icon with a stylized map pin and the text "Google Maps Javascript API" in large white letters. Below this, a subtitle says "Customize maps with your own content and imagery." At the bottom of the main section are two buttons: "GET A KEY" and "VIEW PLANS". At the very bottom of the page, there is a horizontal menu with Korean text: "홈" (Home), "안내" (Information), "참조" (Reference), "샘플" (Samples), and "지원" (Support).

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)

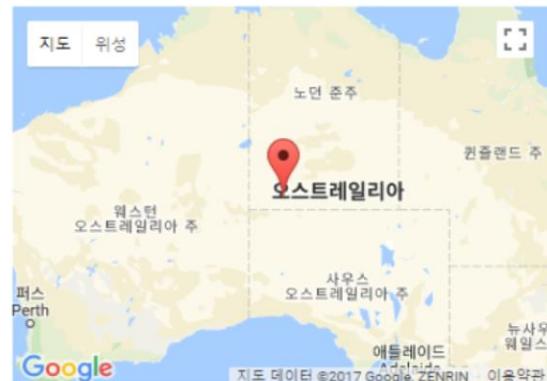
<https://developers.google.com/maps/documentation/javascript/?hl=ko>

Reference (Google Maps JS API)

The screenshot shows the Google Maps JavaScript API documentation page. The URL in the address bar is <https://developers.google.com/maps/documentation/javascript/?hl=ko>. The page title is "Google Maps API". The main navigation menu includes "홈", "문서", and "가격 및 플랜". Below the menu, it says "웹 > Maps JavaScript API". A sub-menu below "Maps JavaScript API" lists "개요", "가이드", "참조", "샘플", and "자원". The main content area contains a message in Korean: "자신만의 콘텐츠와 이미지를 사용하여 지도를 사용자 지정합니다." (Customize maps using your own content and images). There are also tabs for "지도" and "위성".

튜토리얼

다음 튜토리얼 중 하나를 선택하거나 [모든 튜토리얼을 보세요.](#)



마커가 포함된 지도 생성



데이터 시각화

Geolocation in HTML5

HTML5 위치정보 (*Geolocation*)

- 위치정보(**Geolocation**)은 자신의 위치를 웹 사이트와 공유
- 현재 지역의 날씨, 유명한 맛집 등의 정보를 제공받을 수 있다.



geolocation 객체

□ **var geolocation = navigator.geolocation;**

메소드	설명
<u>getCurrentPosition()</u>	사용자의 현재 위치 정보를 반환한다.
<u>watchPosition()</u>	장치의 현재 위치에 대한 정보를 주기적으로 반환한다.
<u>clearWatch()</u>	현재 진행 중인 <u>watchPosition()</u> 실행을 중지한다.

The getCurrentPosition() Method - Return Data

The `getCurrentPosition()` method returns an object if it is successful. The `latitude`, `longitude` and `accuracy` properties are always returned. The other properties below are returned if available.

Property	Description
<code>coords.latitude</code>	The latitude as a decimal number
<code>coords.longitude</code>	The longitude as a decimal number
<code>coords.accuracy</code>	The accuracy of position
<code>coords.altitude</code>	The altitude in meters above the mean sea level
<code>coords.altitudeAccuracy</code>	The altitude accuracy of position
<code>coords.heading</code>	The heading as degrees clockwise from North
<code>coords.speed</code>	The speed in meters per second
<code>timestamp</code>	The date/time of the response

Geolocation 예제

```
<!DOCTYPE html>
<html>
<body>
    <button onclick="getGeolocation()">위치 정보 얻기</button>
    <h2 id="target"> Click the button to get your location.</h2>
    <script>
        var myDiv = document.getElementById("target");
        function getGeolocation() {
            if (navigator.geolocation) {
                navigator.geolocation.getCurrentPosition(showLocation);
            }else {
                myDiv.innerHTML="No gps support";
            }
        }
        function showLocation(location) {
            myDiv.innerHTML = "(위도: " + location.coords.latitude +
                ", 경도: " + location.coords.longitude + ")";
        }
    </script>
</body>
</html>
```

위치 정보 얻기

No gps support

Using Geolocation : my location

위치 정보 얻기

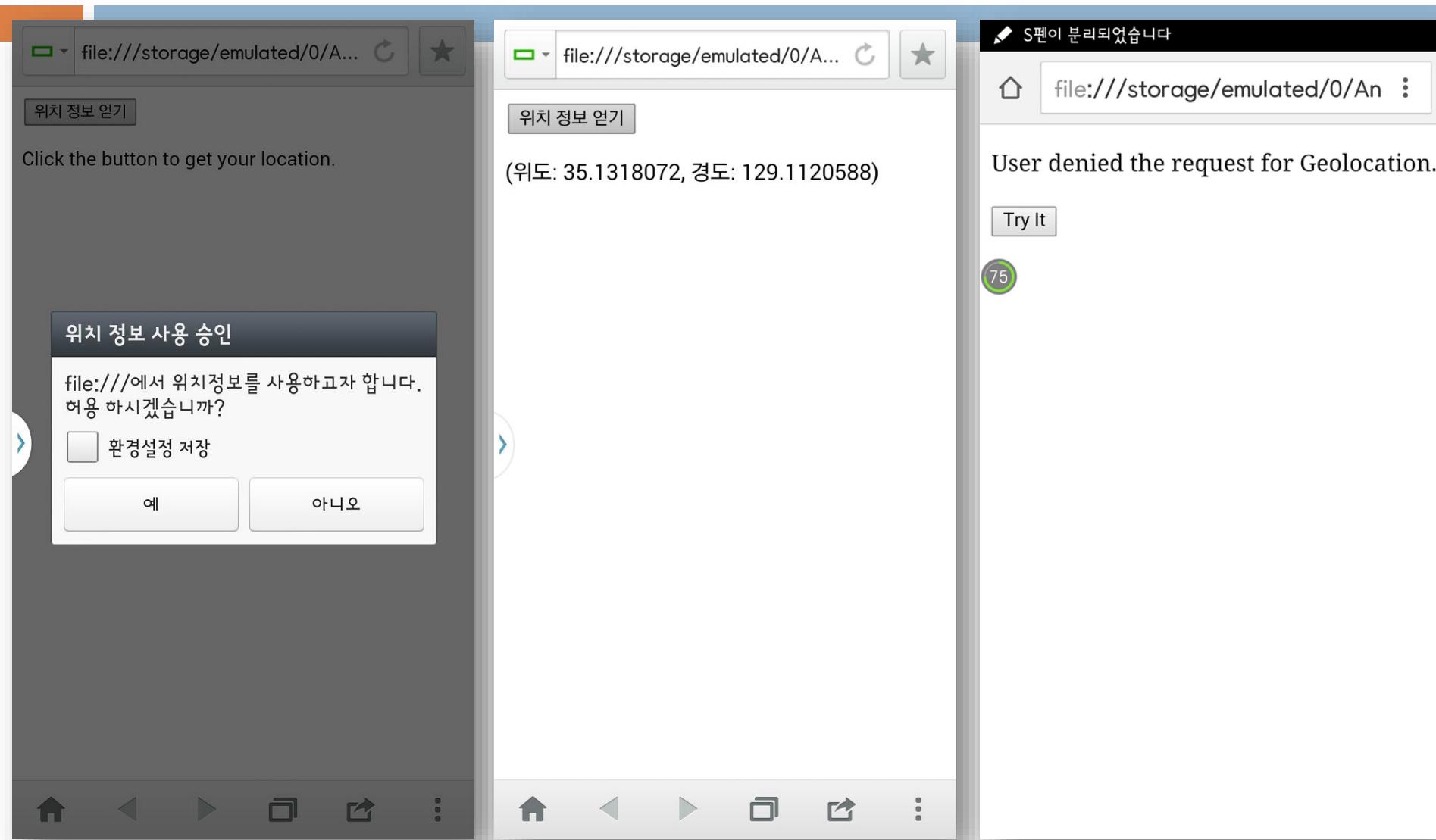
(위도: 35.907757, 경도: 127.766922)

Geolocation 예제 - error

```
<script>
    var myDiv = document.getElementById("target");
    function getGeolocation() {
        if (navigator.geolocation) {
            navigator.geolocation.getCurrentPosition(showLocation, showError);
        }else {
            myDiv.innerHTML="No gps support";
        }
    }

    function showError(error) {
        switch(error.code) {
            case error.PERMISSION_DENIED:
                myDiv.innerHTML = "User denied the request for Geolocation."
                break;
            case error.POSITION_UNAVAILABLE:
                myDiv.innerHTML = "Location information is unavailable."
                break;
            case error.TIMEOUT:
                myDiv.innerHTML = "The request to get user location timed out."
                break;
            case error.UNKNOWN_ERROR:
                myDiv.innerHTML = "An unknown error occurred."
                break;
        }
    }
}
```

Geolocation 예제 – 실행 결과(RP)



이동하면서 위치 정보를 얻기

- geolocation 객체의 **watchPosition()**을 호출
 - **watchPosition()** - 사용자의 현재 위치를 연속하여 출력한다.
 - **clearWatch(id)** - watchPosition() id를 중지한다.

Geolocation object - Other interesting Methods

watchPosition() - Returns the current position of the user and continues to return updated position as the user moves (like the GPS in a car).

clearWatch() - Stops the watchPosition() method.

이동하면서 위치 정보를 얻기

```
<button onclick="startGeolocation()">위치 정보 시작</button>
<button onclick="stopGeolocation()">위치 정보 중지</button>
<h2 id="target"> </h2>

<script>
    var id;
    var myDiv = document.getElementById("target");
    function startGeolocation() {
        if (navigator.geolocation) {
            id = navigator.geolocation.watchPosition(showGeolocation);
        }else {
            myDiv.innerHTML="No gps support";
        }
    }
    function showGeolocation(location) {
        myDiv.innerHTML = "(위도: " + location.coords.latitude +
        ", 경도: " + location.coords.longitude + ")";
    }
    function stopGeolocation() {
        if (navigator.geolocation) {
            navigator.geolocation.clearWatch(id);
        }
        myDiv.innerHTML="Stop location search.";
    }
</script>
```

Geolocation 예제 – 실행 결과

Using Geolocation : moving locations

위치 정보 시작

위치 정보 중지

(위도: 35.907757, 경도: 127.766922)

Using Geolocation : moving locations

위치 정보 시작

위치 정보 중지

Stop location search.

지도(Map)에 위치 표시하기

```
<!DOCTYPE html>
<html>
<body>
    <button onclick="startGeolocation()">지도 보이기</button>
    <button onclick="showMylocation()">지도(E323) 보이기</button>
    <h2 id="target"> </h2>
    <div id="map"> </div>
<script>
    var myDiv = document.getElementById("target");

    function startGeolocation() {
        if (navigator.geolocation) {
            navigator.geolocation.getCurrentPosition(showGeolocation);
        }
    }

    function showGeolocation(location) {
        myDiv.innerHTML = "(위도: " + location.coords.latitude +
        ", 경도: " + location.coords.longitude + ")";
        var pos = location.coords.latitude + "," + location.coords.longitude;
        //pos = "35.249164,128.901881";
        var img_url = "http://maps.googleapis.com/maps/api/staticmap?center=" +
        pos+"&zoom=14&size=400x300&sensor=false";
        document.getElementById("map").innerHTML = "<img src=\"" + img_url + "\">";
    }
}
```

지도(Map)에 위치(E323) 표시하기

```
function showMylocation() {
    pos = "35.249164,128.901881"; // E323
    myDiv.innerHTML=pos;
    var img_url = "http://maps.googleapis.com/maps/api/staticmap?center="
    +pos+"&zoom=14&size=400x300&sensor=false";
    document.getElementById("map").innerHTML = "<img src='"+img_url+">";
}

</script>
```

실행결과

Using Geolocation : show map

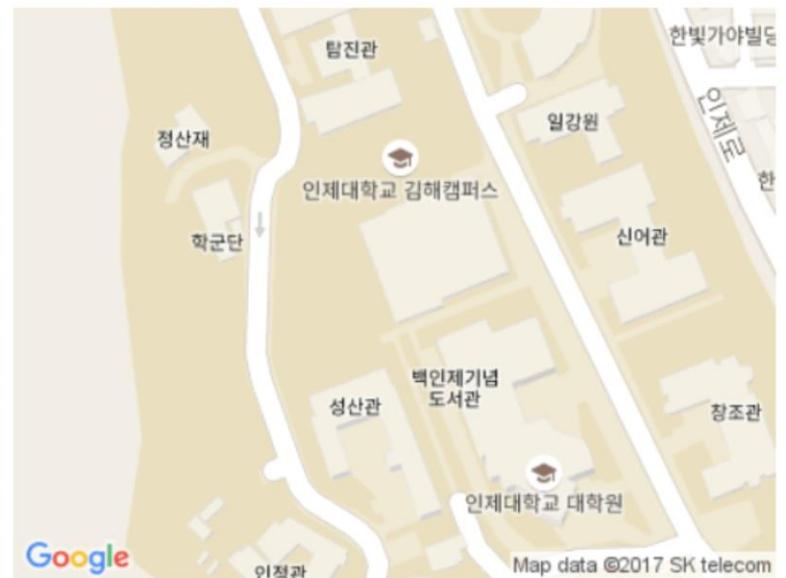
[지도 보이기](#) [지도\(E323\) 보이기](#)

(위도: 35.907757, 경도: 127.766922)



[지도 보이기](#) [지도\(E323\) 보이기](#)

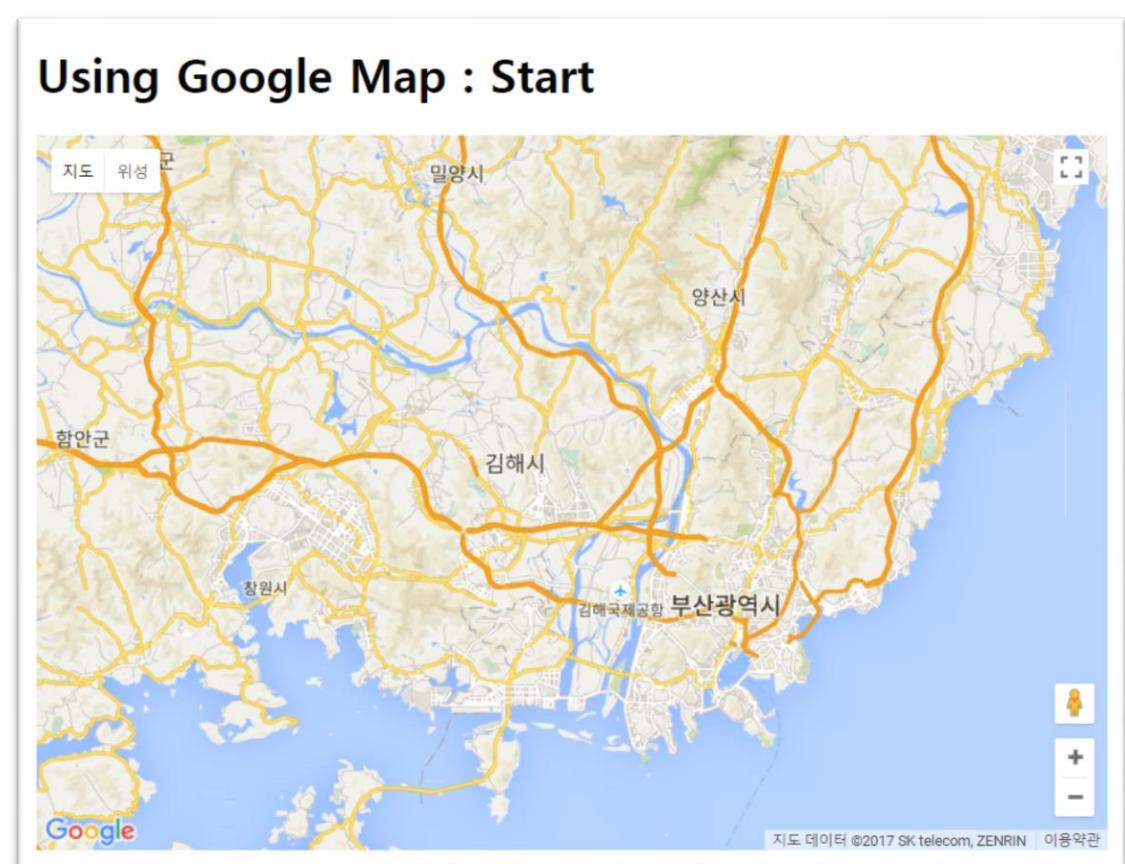
35.249164, 128.901881



Google Map

Javascript API

Basic



1. Create a Basic Google Map

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title> Mobile Simulation: Google Map </title>
<script
  src="http://maps.googleapis.com/maps/api/js?v=3.exp&sensor=false">
</script>

<script>
function initialize() {
  var mapProp = {
    center: new google.maps.LatLng(35.249164, 128.901881), // E323
    zoom:17,
    mapTypeId: google.maps.MapTypeId.ROADMAP
  };
  var map = new google.maps.Map(document.getElementById("googleMap"), mapProp);
}

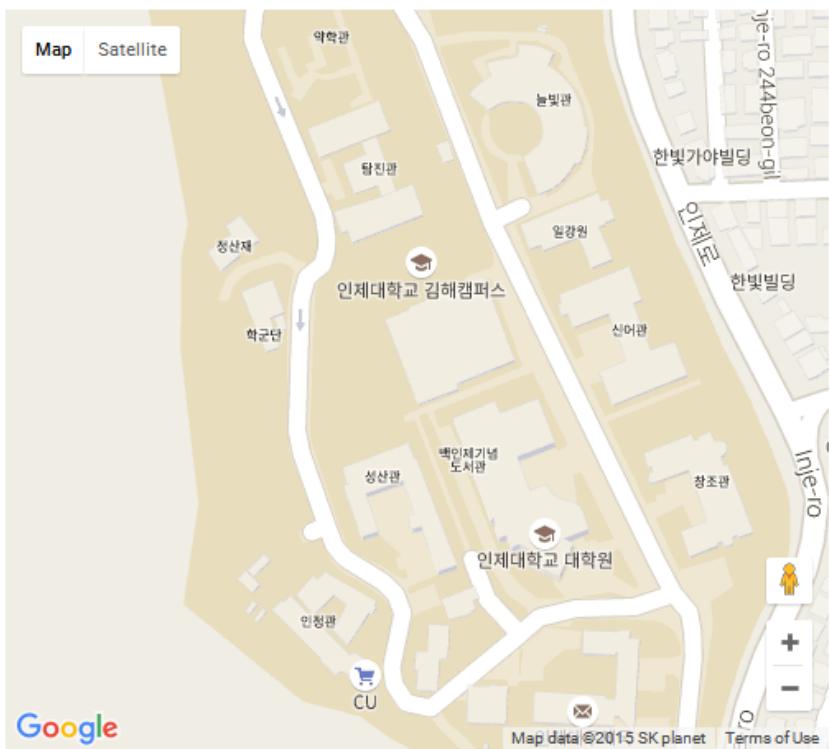
google.maps.event.addDomListener(window, 'load', initialize);
</script>
</head>

<body>
<h1>Using Google Map : Start</h1>
<div id="googleMap" style="width:500px;height:450px;"> </div>
</body>
</html>
```

[Result] 1.1 Localization

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

Using Google Map : Start



Using Google Map : Localization



2. Four types of maps

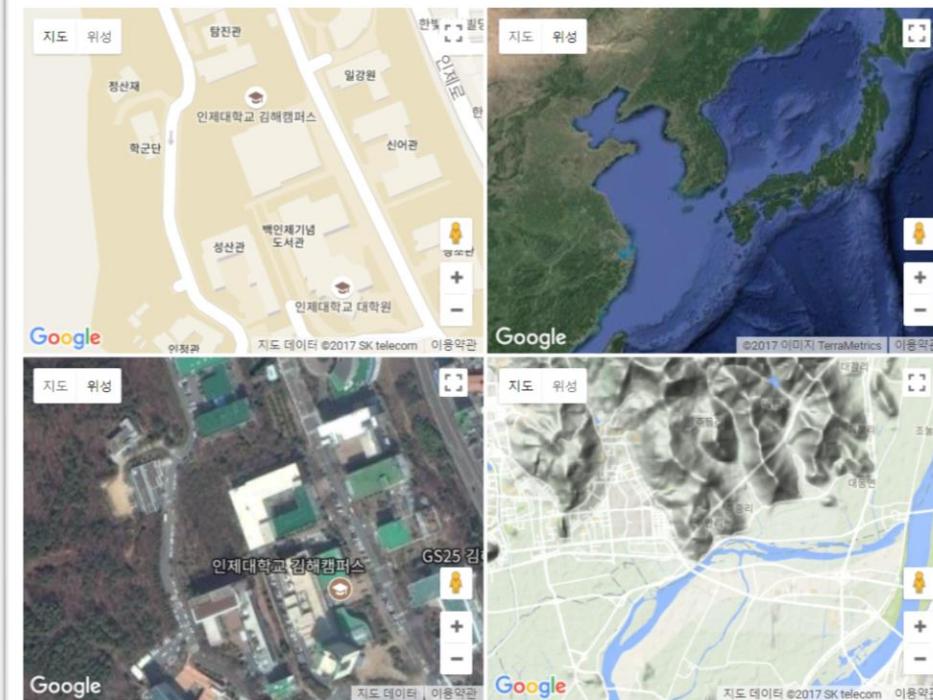
```
function initialize()
{
  var mapProp = {
    center: new google.maps.LatLng(35.249164, 128.901881),
    zoom:17,
    mapTypeId: google.maps.MapTypeId.ROADMAP
  };
  var mapProp2 = {
    center: new google.maps.LatLng(35.249164, 128.901881),
    zoom:4,
    mapTypeId: google.maps.MapTypeId.SATELLITE
  };
  var mapProp3 = {
    center: new google.maps.LatLng(35.249164, 128.901881),
    zoom:17,
    mapTypeId: google.maps.MapTypeId.HYBRID
  };
  var mapProp4 = {
    center: new google.maps.LatLng(35.249164, 128.901881),
    zoom:15,
    mapTypeId: google.maps.MapTypeId.TERRAIN
  };
  var map = new google.maps.Map(document.getElementById("googleMap"),mapProp);
  var map2 = new google.maps.Map(document.getElementById("googleMap2"),mapProp2);
  var map3 = new google.maps.Map(document.getElementById("googleMap3"),mapProp3);
  var map4 = new google.maps.Map(document.getElementById("googleMap4"),mapProp4);
}
```

2. Four types of maps

```
<table>
<tr>
<td><div id="googleMap" style="width:400px;height:300px;"> </div> </td>
<td><div id="googleMap2" style="width:400px;height:300px"> </div> </td>
</tr>

<tr>
<td><div id="googleMap3" style="width:400px;height:300px;"> </div> </td>
<td><div id="googleMap4" style="width:400px;height:300px;"> </div> </td>
</tr>
</table>
```

Using Google Map : Types



3. Google Maps - Overlays

Google Maps - Overlays

Overlays are objects on the map that are bound to latitude/longitude coordinates.

Google Maps has several types of overlays:

- Marker - Single locations on a map. Markers can also display custom icon images
- Polyline - Series of straight lines on a map
- Polygon - Series of straight lines on a map, and the shape is "closed"
- Circle and Rectangle
- Info Windows - Displays content within a popup balloon on top of a map
- Custom overlays

3.1.1 Add a marker

```
<script>  
var myCenter=new google.maps.LatLng(35.249164, 128.901881); // E323  
  
function initialize()  
{  
    var mapProp = {  
        center:myCenter,  
        zoom:5,  
        mapTypeId:google.maps.MapTypeId.ROADMAP  
    };  
  
    var map=new google.maps.Map(document.getElementById("googleMap"),  
    mapProp);  
  
    var marker=new google.maps.Marker({  
        position:myCenter,  
    });  
  
    marker.setMap(map);  
}  
  
google.maps.event.addDomListener(window, 'load', initialize);  
</script>
```

Using Google Map : Add an icon



```
<div id="googleMap" style="width:500px;height:450px;"> </div>
```

3.1.2 Animate a marker

```
<script>  
var myCenter=new google.maps.LatLng(35.249164, 128.901881); // E323  
  
function initialize()  
{  
    var mapProp = {  
        center:myCenter,  
        zoom:5,  
        mapTypeId:google.maps.MapTypeId.ROADMAP  
    };  
  
    var map=new google.maps.Map(document.getElementById("googleMap"),  
        mapProp);  
  
    var marker=new google.maps.Marker({  
        position:myCenter,  
        animation:google.maps.Animation.BOUNCE  
    });  
  
    marker.setMap(map);  
}  
  
google.maps.event.addDomListener(window, 'load', initialize);  
</script>
```

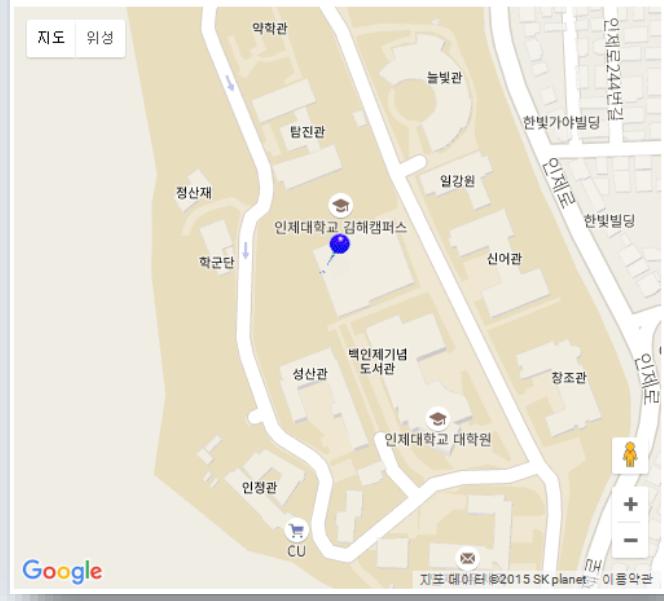
Using Google Map : Add an icon



3.1.3 Add Icon Instead of Marker

```
<script>  
var myCenter=new google.maps.LatLng(35.249164, 128.901881); // E323  
  
function initialize()  
{  
    var mapProp = {  
        center:myCenter,  
        zoom:5,  
        mapTypeId:google.maps.MapTypeId.ROADMAP  
    };  
  
    var map=new google.maps.Map(document.getElementById("googleMap"),  
        mapProp);  
  
    var marker=new google.maps.Marker({  
        position:myCenter,  
        icon: 'image/pin.png'  
    });  
  
    marker.setMap(map);  
}  
  
google.maps.event.addDomListener(window, 'load', initialize);  
</script>
```

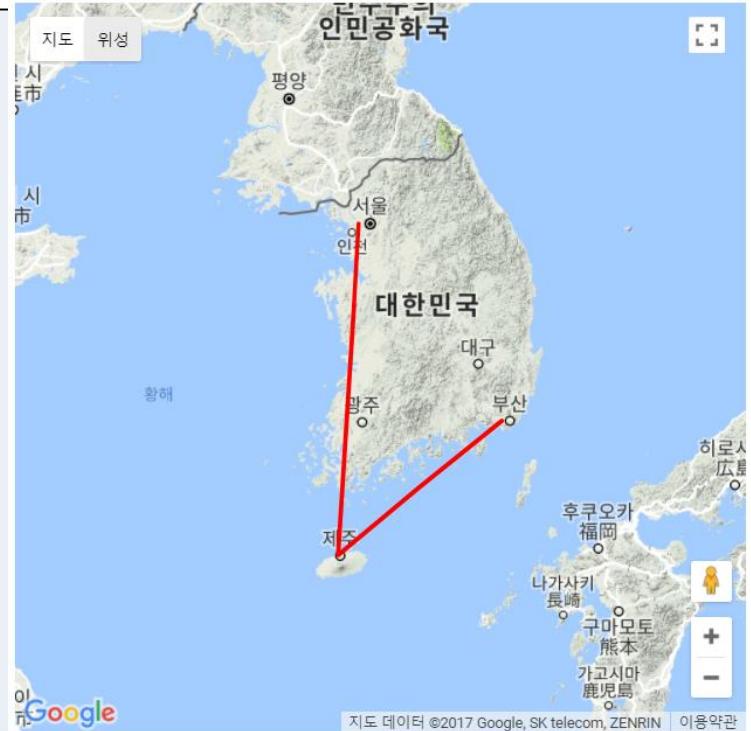
Using Google Map : Add an icon



3.2.1 Add Polyline

Using Google Map : Polyline

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

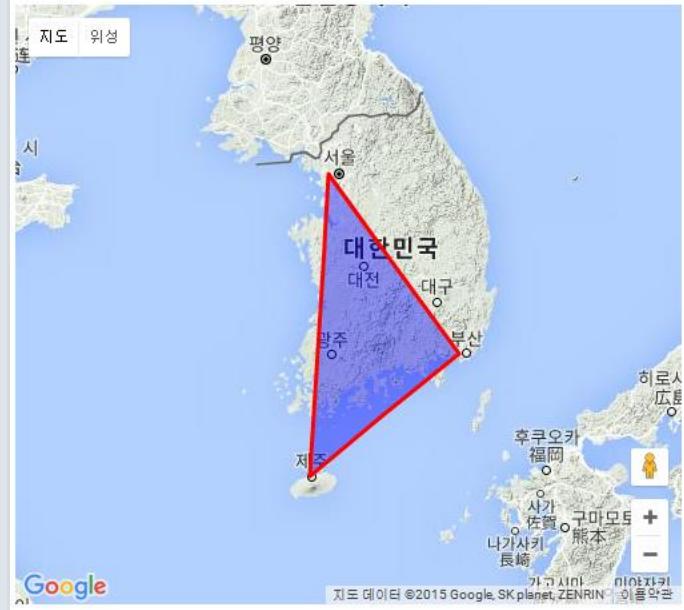


- path - specifies several latitude/longitude coordinates for the line
- strokeColor - specifies a hexadecimal color for the line (format: "#FFFFFF")
- strokeOpacity - specifies the opacity of the line (a value between 0.0 and 1.0)
- strokeWeight - specifies the weight of the line's stroke in pixels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

3.2.2 Add Polygon

```
function initialize() {  
    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.Polygon({  
        path: flightPlanCoordinates,  
        geodesic: true,  
        strokeColor: '#FF0000',  
        strokeOpacity: 1.0,  
        strokeWeight: 3,  
        fillColor:"#0000FF",  
        fillOpacity:0.4  
    });  
  
    flightPath.setMap(map);  
}
```

Using Google Map : Polygon



- path - specifies several LatLng coordinates for the line (first and last coordinate are equal)
- 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
- fillColor - specifies a hexadecimal color for the area within the enclosed region (format: "#FFFFFF")
- fillOpacity - specifies the opacity of the fill color (a value between 0.0 and 1.0)
- editable - defines whether the line is editable by users (true/false)

3.2.3 Add Circle around myCity, GimHae

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

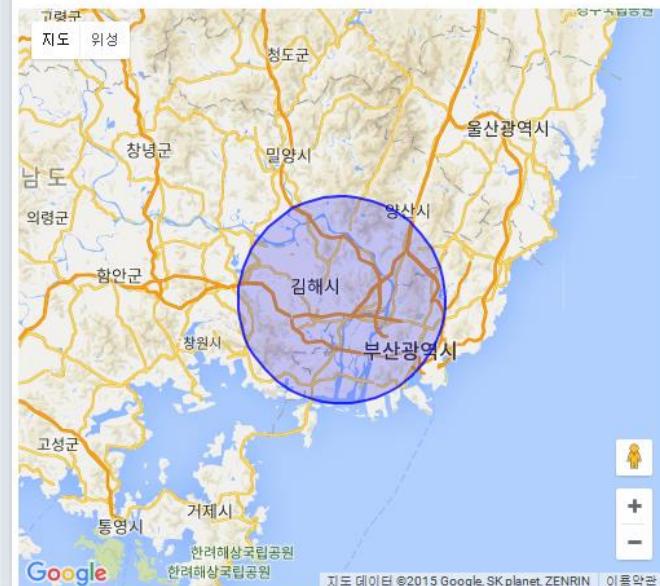
function initialize()
{
  var mapProp = {
    center:e323,
    zoom:9,
    mapTypeId:google.maps.MapTypeId.ROADMAP
  };

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

var myCity=new google.maps.Circle({
  center:e323,
  radius:20000, // meter
  strokeColor:"#0000FF",
  strokeOpacity:0.8,
  strokeWeight:2,
  fillColor:"#0000FF",
  fillOpacity:0.4
});
myCity.setMap(map);
}

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

Using Google Map : Add a Circle



- center - specifies the google.maps.LatLng of the center of the circle
- radius - specifies the radius of the circle, in meters
- strokeColor - specifies a hexadecimal color for the line around the circle (format: "#FFFFFF")
- strokeOpacity - specifies the opacity of the stroke color (a value between 0.0 and 1.0)
- strokeWeight - specifies the weight of the line's stroke in pixels
- fillColor - specifies a hexadecimal color for the area within the circle (format: "#FFFFFF")
- fillOpacity - specifies the opacity of the fill color (a value between 0.0 and 1.0)
- editable - defines whether the circle is editable by users (true/false)

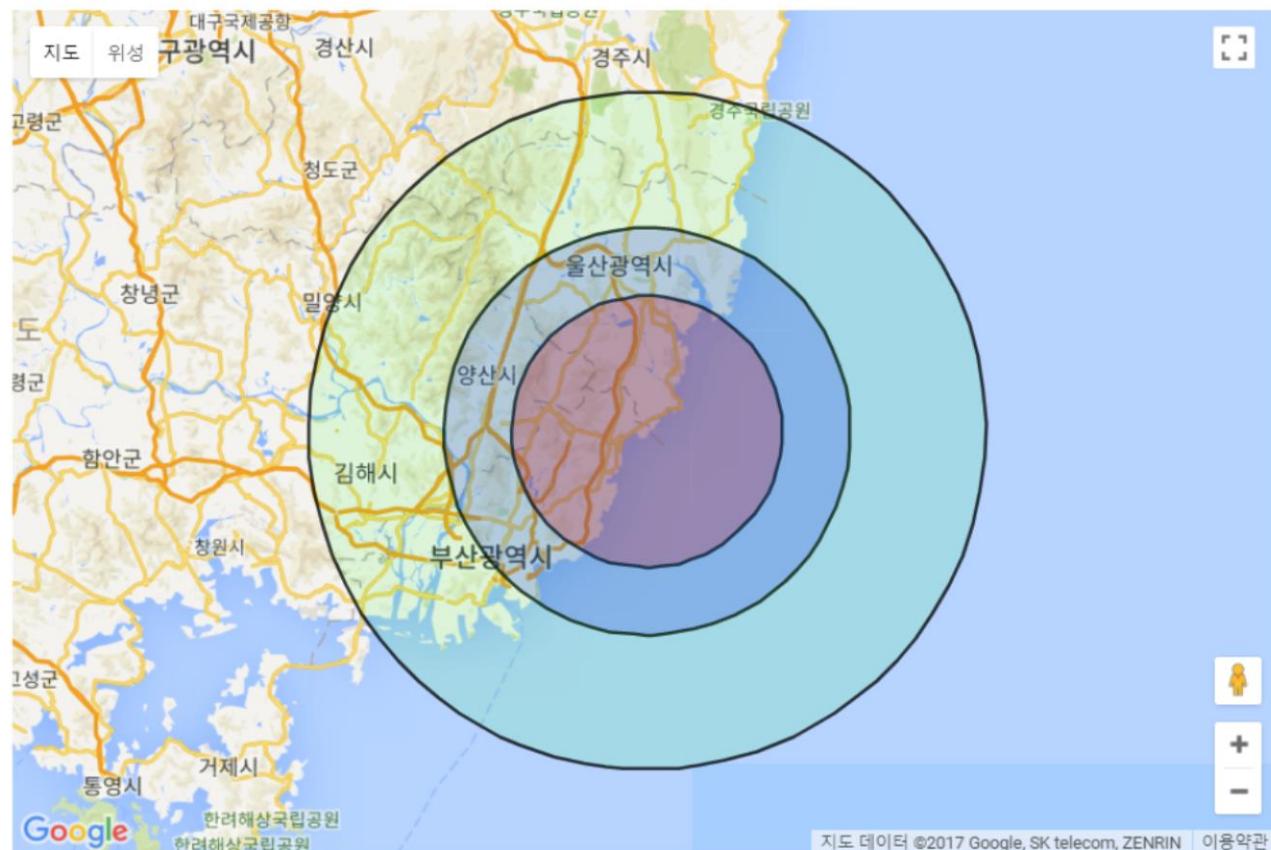
3.2.4 Add Circle around NPP

```
var npp=new google.maps.LatLng(35.327235, 129.301354); // 고리 원전  
var d20km=new google.maps.Circle({  
    center:npp,  
    radius:20000, // meter  
    strokeColor:"#000000",  
    strokeOpacity:0.8,  
    strokeWeight:2,  
    fillColor:"#FF0000",  
    fillOpacity:0.3  
});  
var d30km=new google.maps.Circle({  
    center:npp,  
    radius:30000, // meter  
    strokeColor:"#000000",  
    strokeOpacity:0.8,  
    strokeWeight:2,  
    fillColor:"#0000FF",  
    fillOpacity:0.2  
});  
var d50km=new google.maps.Circle({  
    center:npp,  
    radius:50000, // meter  
    strokeColor:"#000000",  
    strokeOpacity:0.8,  
    strokeWeight:2,  
    fillColor:"#00FF00",  
    fillOpacity:0.1  
});
```

```
d50km.setMap(map);  
d30km.setMap(map);  
d20km.setMap(map);
```

3.2.4 Add Circle around NPP

Using Google Map : From Gori NPP



MSnn_NPP_Alert.png

3.3 Add Info Window

Using Google Map : Info window

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

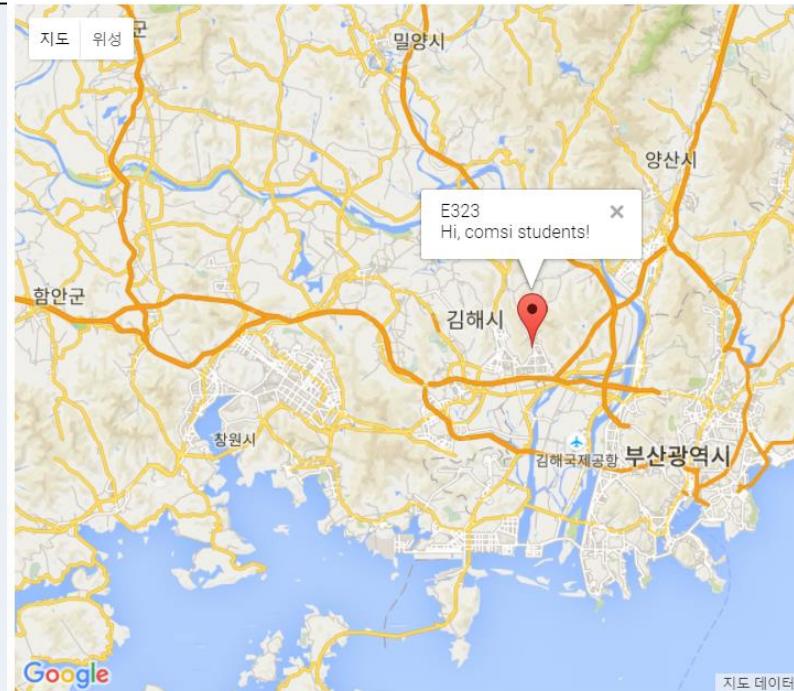
function initialize()
{
  var mapProp = {
    center:e323,
    zoom:9,
    mapTypeId:google.maps.MapTypeId.ROADMAP
  };

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

  var marker=new google.maps.Marker({
    position: e323
  });

  var infowindow = new google.maps.InfoWindow({
  content:"E323 <br> Hi, comsi students!"
  });
  infowindow.open(map,marker);

  marker.setMap(map);
}
google.maps.event.addDomListener(window, 'load', initialize);
</script>
```



4. Google Maps - Events

Events

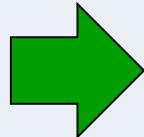
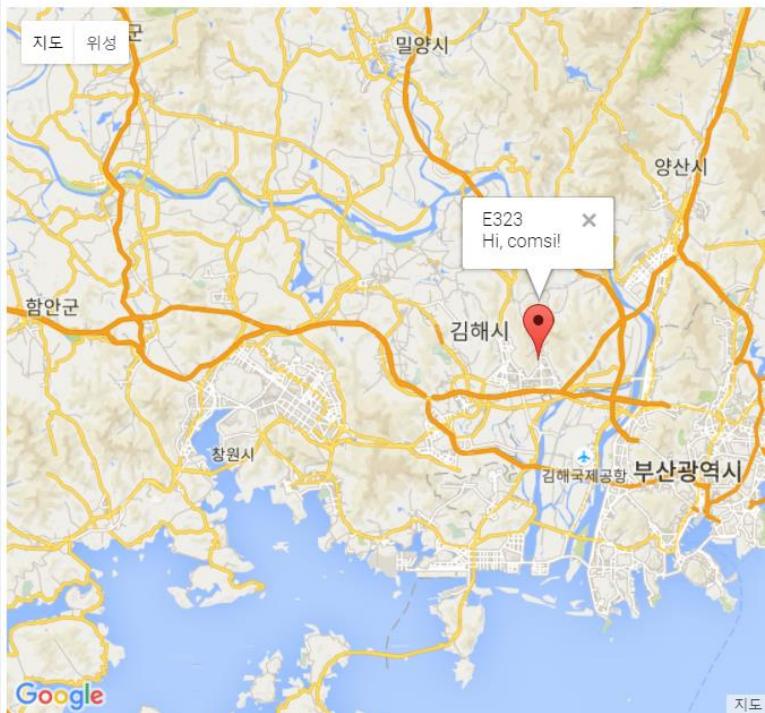
Constructor/Object	Description
MapsEventListener	It has no methods and no constructor. Its instances are returned from addListener(), addDomListener() and are eventually passed back to removeListener()
event	Adds/Removes/Trigger event listeners
MouseEvent	Returned from various mouse events on the map and overlays

4.1 Click The Marker to Zoom

```
// add the below code in <script>
```

```
// Zoom to 17 when clicking on marker
google.maps.event.addListener(marker,'click',function() {
  map.setZoom(17);
  map.setCenter(marker.getPosition());
});
```

Using Google Map : Event zoom



Using Google Map : Event zoom



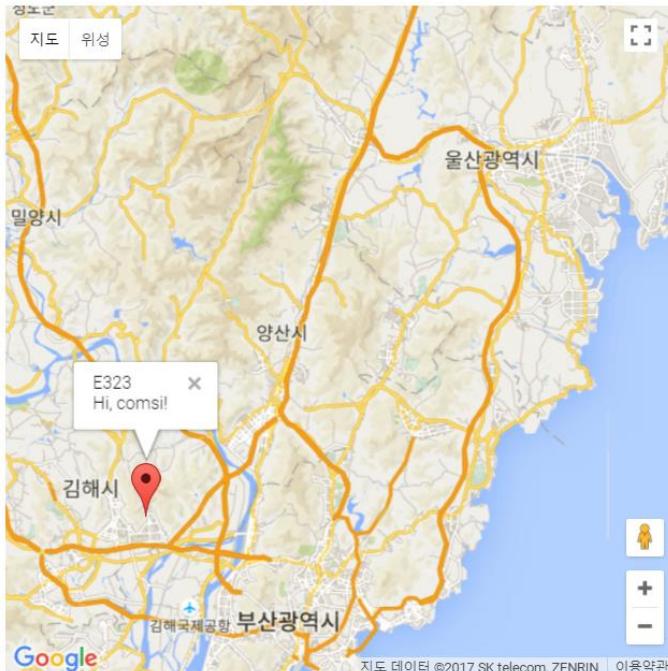
4.2 Pan Back to Marker

```
// add the below code in <script>
```

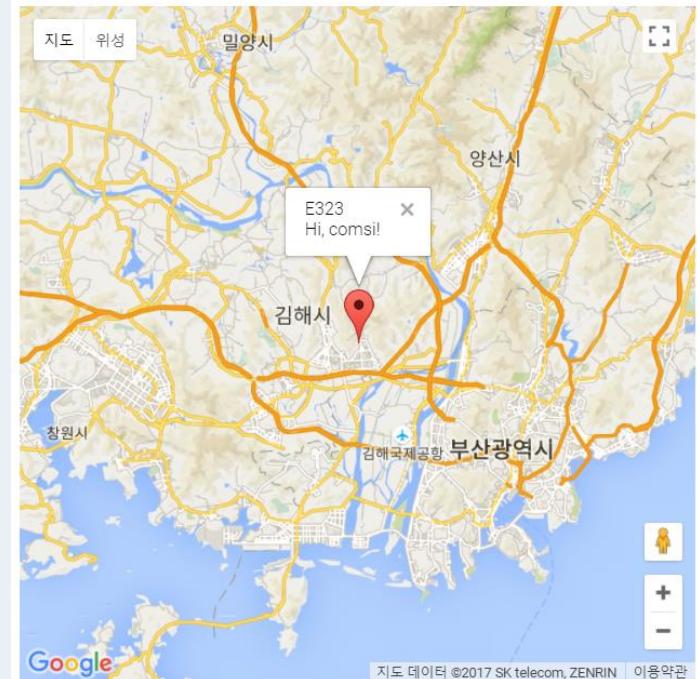
```
// add an event handler to the map for changes to the 'center' property and pan the map back to  
// the marker after 3 seconds on a center_changed event:
```

```
google.maps.event.addListener(map,'center_changed',function() {  
  window.setTimeout(function() {  
    map.panTo(marker.getPosition());  
  },3000);  
});
```

Using Google Map : Pan back



Using Google Map : Pan back



4.3 Open an InfoWindow When Clicking on The Marker

```
// add the below code in <script>  
// Click on the marker to show an infowindow  
google.maps.event.addListener(marker, 'click', function() {  
  infowindow.open(map,marker);  
});
```

Using Google Map : Open Info Window



Using Google Map : Open Info Window



4.4 Set Markers and Open InfoWindow for Each Marker

```
// add the below code in <script>  
// Place markers and open infowindow for each marker
```

```
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);  
}
```

// 그 동안 작업했던 overlay를 모두 지우고 시작.

```
<div id="googleMap" style="width:600px;height:600px;"> </div>
```

Using Google Map : Place markers



MSnn_Place_Markers.png

5. Google Maps – Controls

Google Maps - The Default Controls

When showing a standard Google map, it comes with the default control set:

- Zoom - displays a slider or "+/-" buttons to control the zoom level of the map
- Pan - displays a pan control for panning the map
- MapType - lets the user toggle between map types (roadmap and satellite)
- Street View - displays a Pegman icon which can be dragged to the map to enable Street View

Google Maps - More Controls

In addition to the default controls, Google Maps also has:

- Scale - displays a map scale element
- Rotate - displays a small circular icon which allows you to rotate maps
- Overview Map - displays a thumbnail overview map reflecting the current map viewport within a wider area

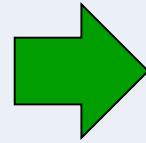
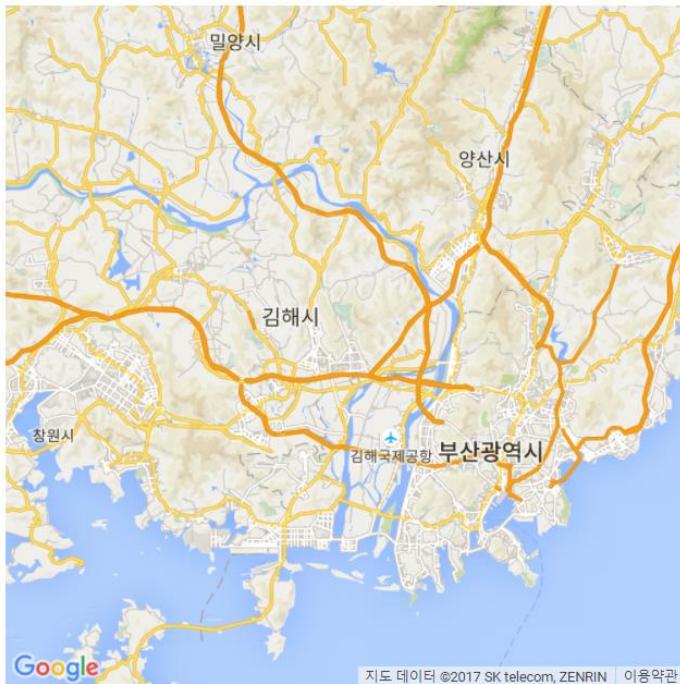
You can specify which controls to show when creating the map (inside MapOptions) or by calling setOptions() to change the map's options.

https://www.w3schools.com/graphics/google_maps_controls.asp

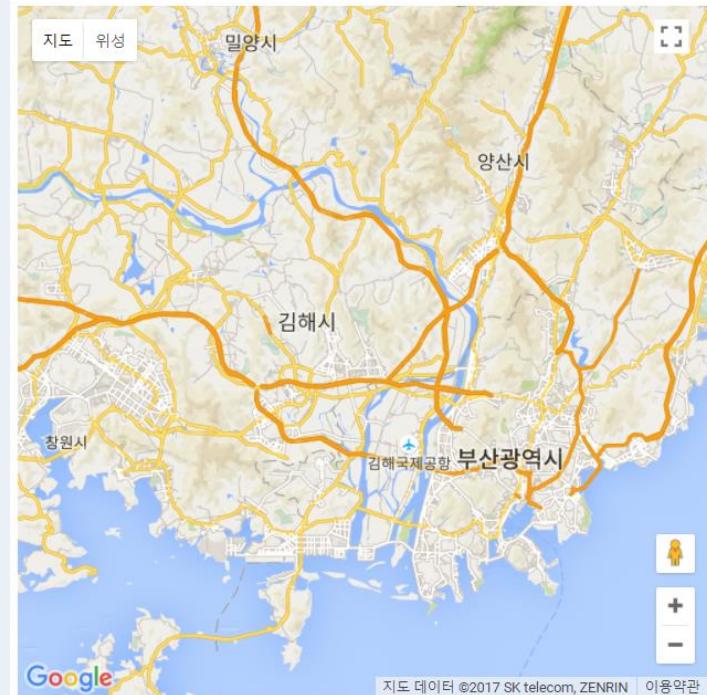
5.1 Disabling The Default Controls

```
// add the below code in <script>  
var mapProp = {  
  center: new google.maps.LatLng(35.249164, 128.901881), // E323  
  disableDefaultUI:true,  
  zoom:17,  
  mapTypeId: google.maps.MapTypeId.ROADMAP  
};
```

Using Google Map : Enable controls



Using Google Map : Enable controls

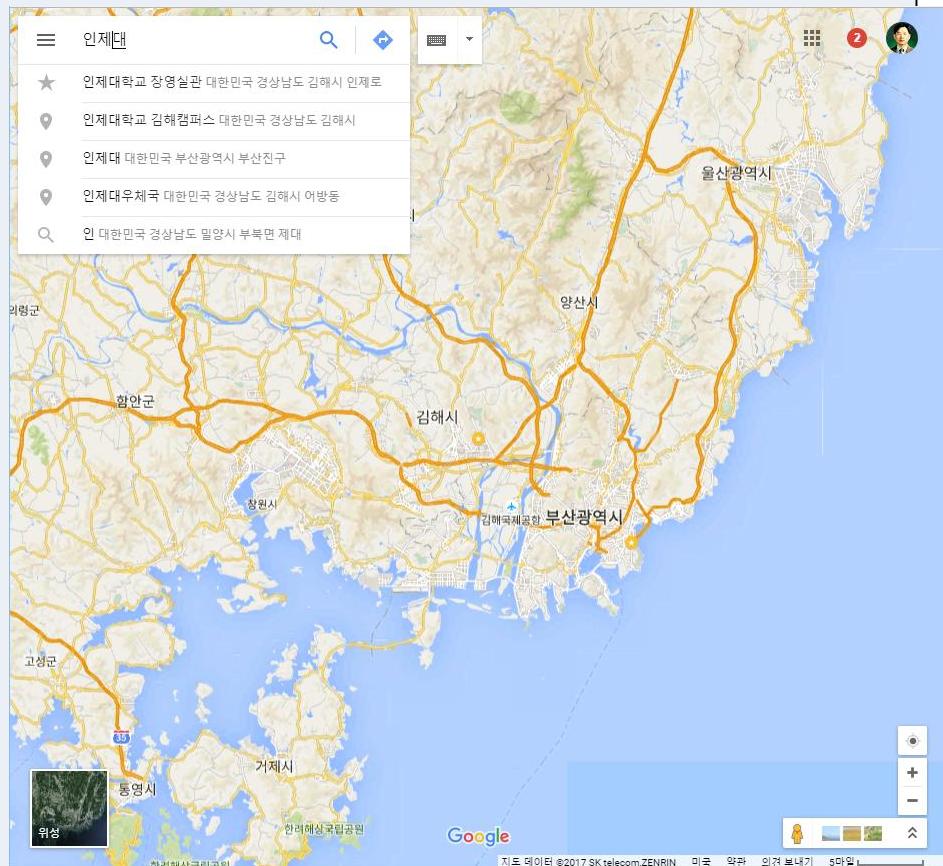
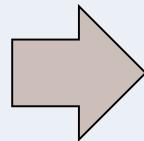
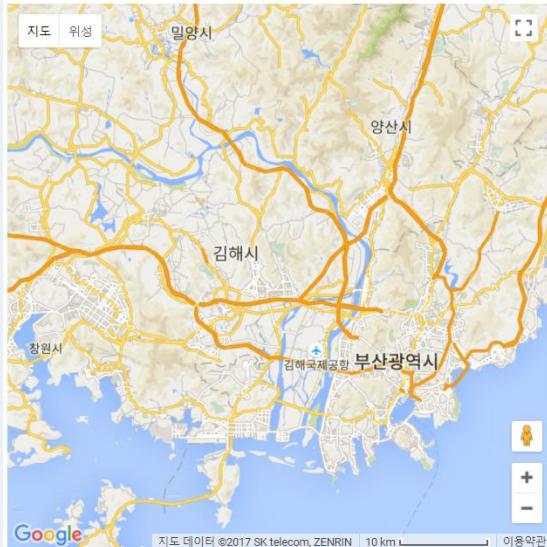


5.2 Turn On All Controls

```
// add the below code in <script>
```

```
var mapProp = {  
  center: new google.maps.LatLng(35.249164, 128.901881), // E323  
  panControl:true,  
  zoomControl:true,  
  mapTypeControl:true,  
  scaleControl:true,  
  streetViewControl:true,  
  overviewMapControl:true,  
  rotateControl:true,  
  zoom:17,  
  mapTypeId: google.maps.MapTypeId.ROADMAP  
};
```

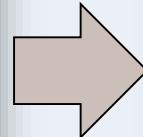
Using Google Map : All controls



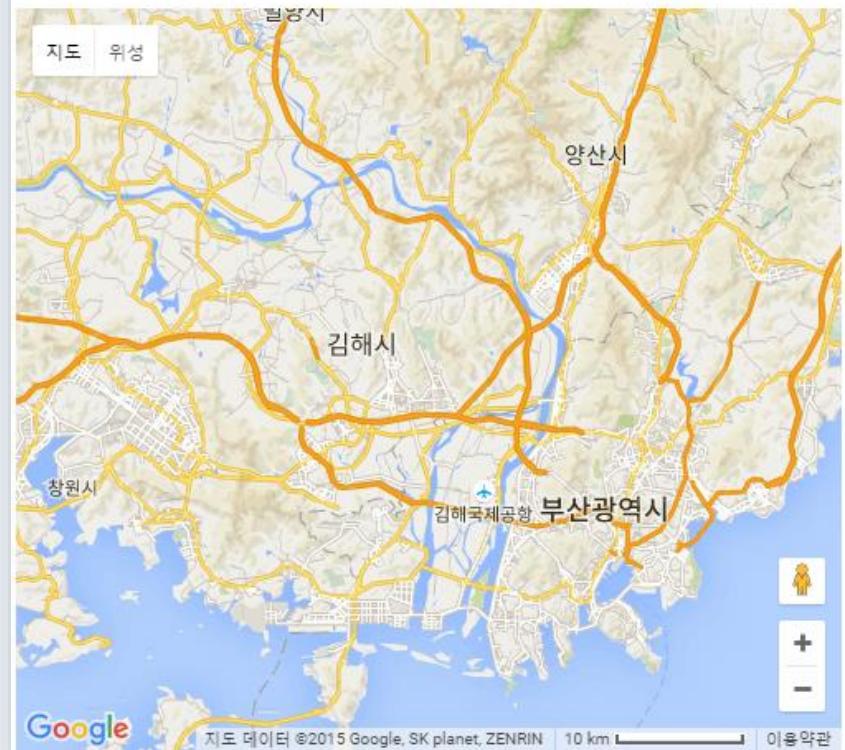
5.3.1 Modifying Controls – small Zoom

```
// add the below code in <script>
var mapProp = {
  center: new google.maps.LatLng(35.249164, 128.901881), // E323
  zoom:17,
  zoomControl:true,
  zoomControlOptions: {
    style:google.maps.ZoomControlStyle.SMALL
  },
  mapTypeId: google.maps.MapTypeId.ROADMAP
};
```

Using Google Map : Small Zoom

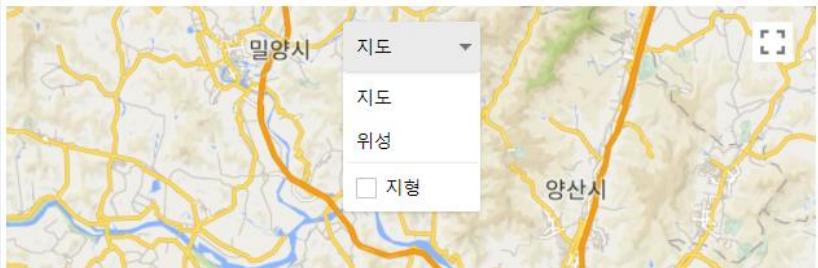


Using Google Map : All controls

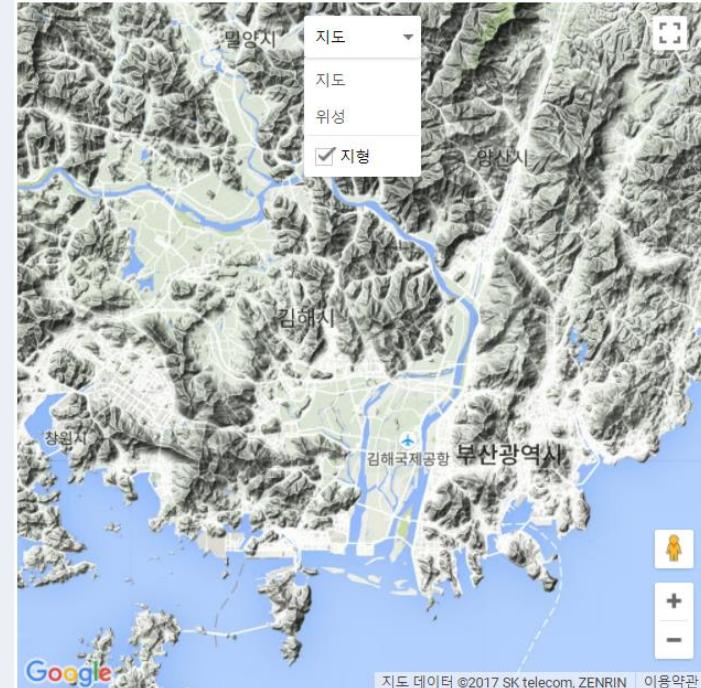


5.3.2 Modifying Controls – dropdown mapType

```
// add the below code in <script>
var mapProp = {
  center: new google.maps.LatLng(35.249164, 128.901881), // E323
  zoom:17,
  mapTypeControl:true,
  mapTypeControlOptions: {
    style:google.maps.MapTypeControlStyle.DROPDOWN_MENU
    //position:google.maps.ControlPosition.TOP_CENTER
  },
  mapTypeId: google.maps.MapTypeId.ROADMAP
};
```



Using Google Map : Dropdown mapType



과제09. msnn_rpt09.zip

49

[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.

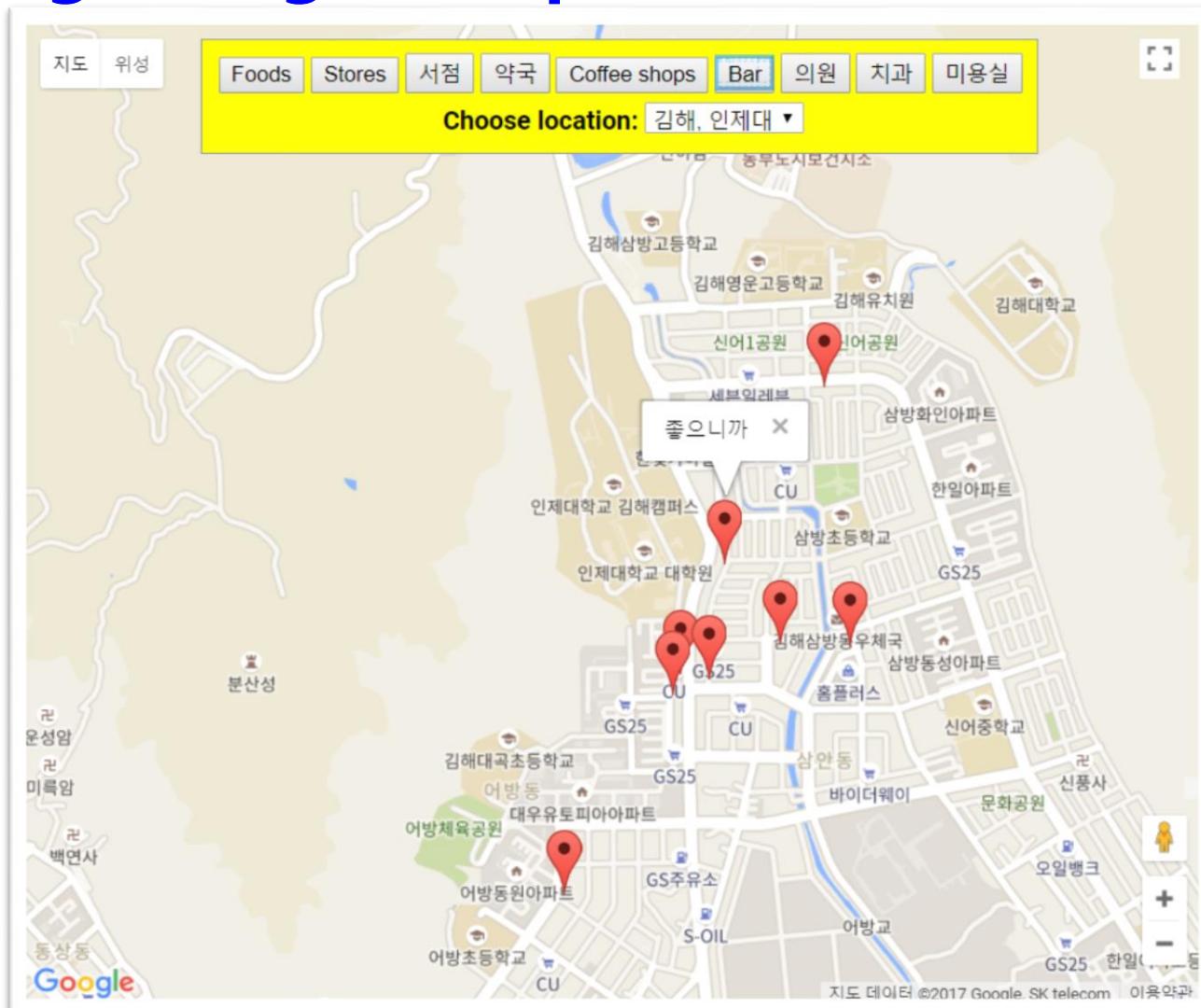
[제출파일] [msnn_rpt09.zip](#) (11월14일 오후 6시 마감)

[html 파일과 사용된 그림을 압축하여 이메일로 “msnn_rpt09” 제목으로 제출](#)

Email : chaos21c@gmail.com

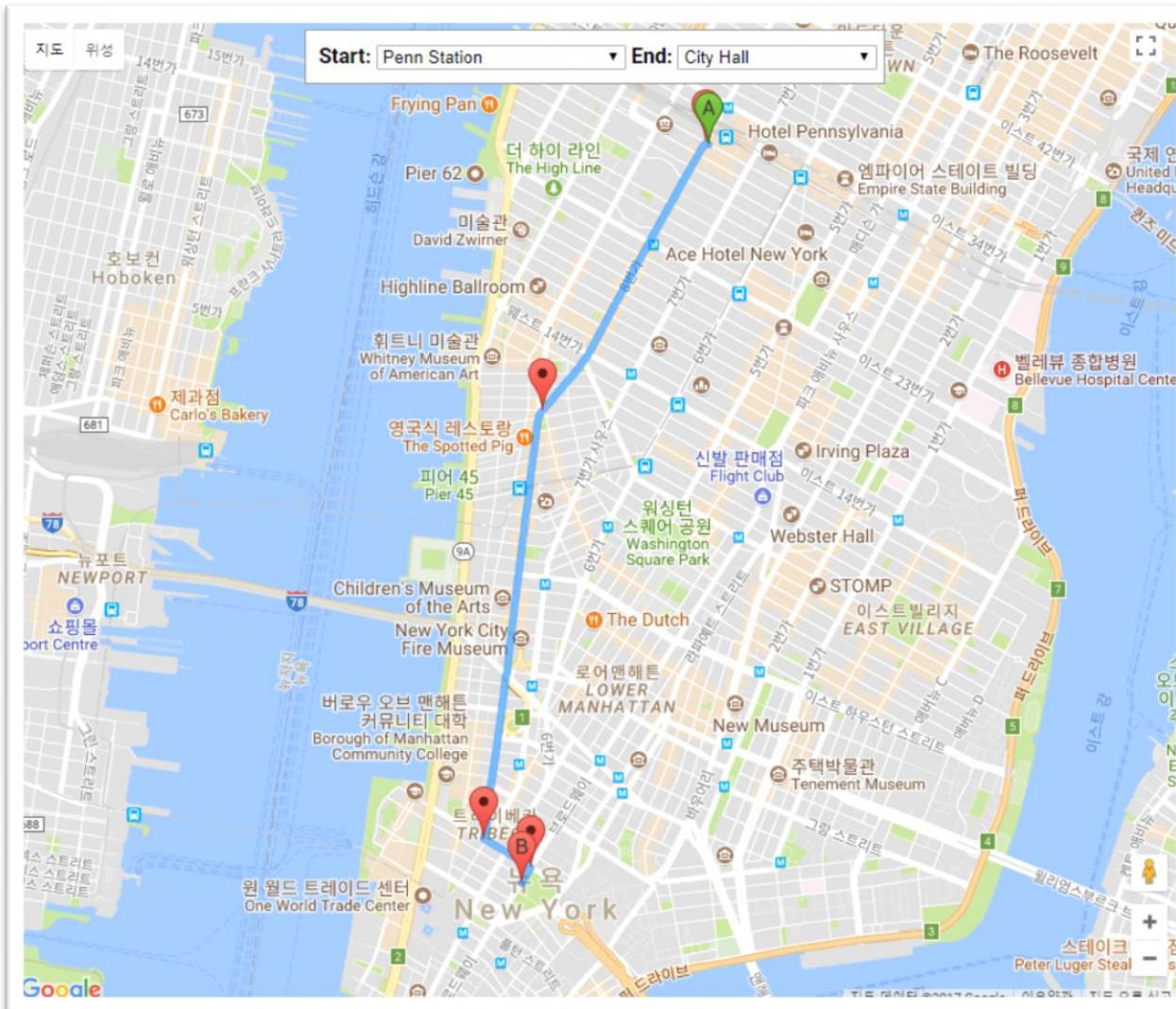
Target #2:

Using Google Map API V3



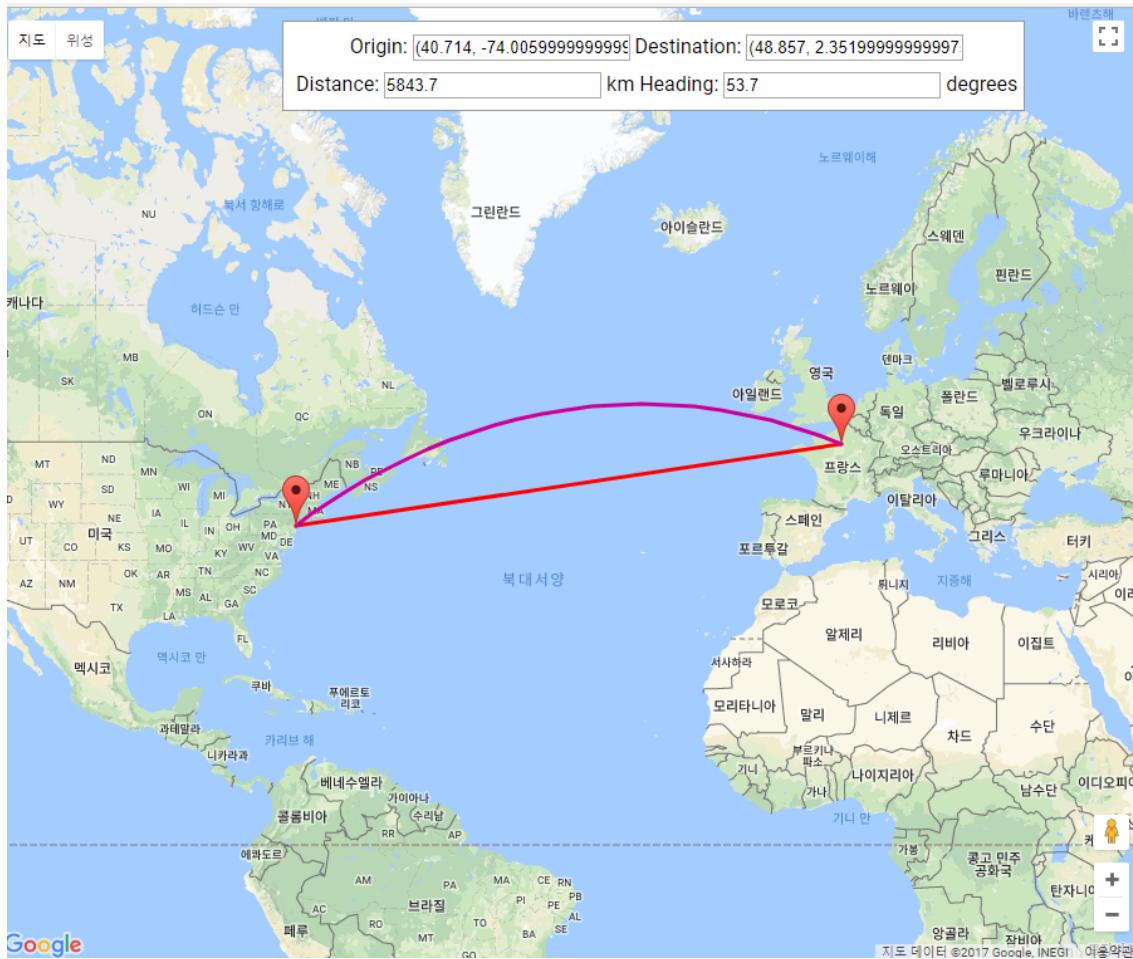
Target #2:

Using Google Map API V3

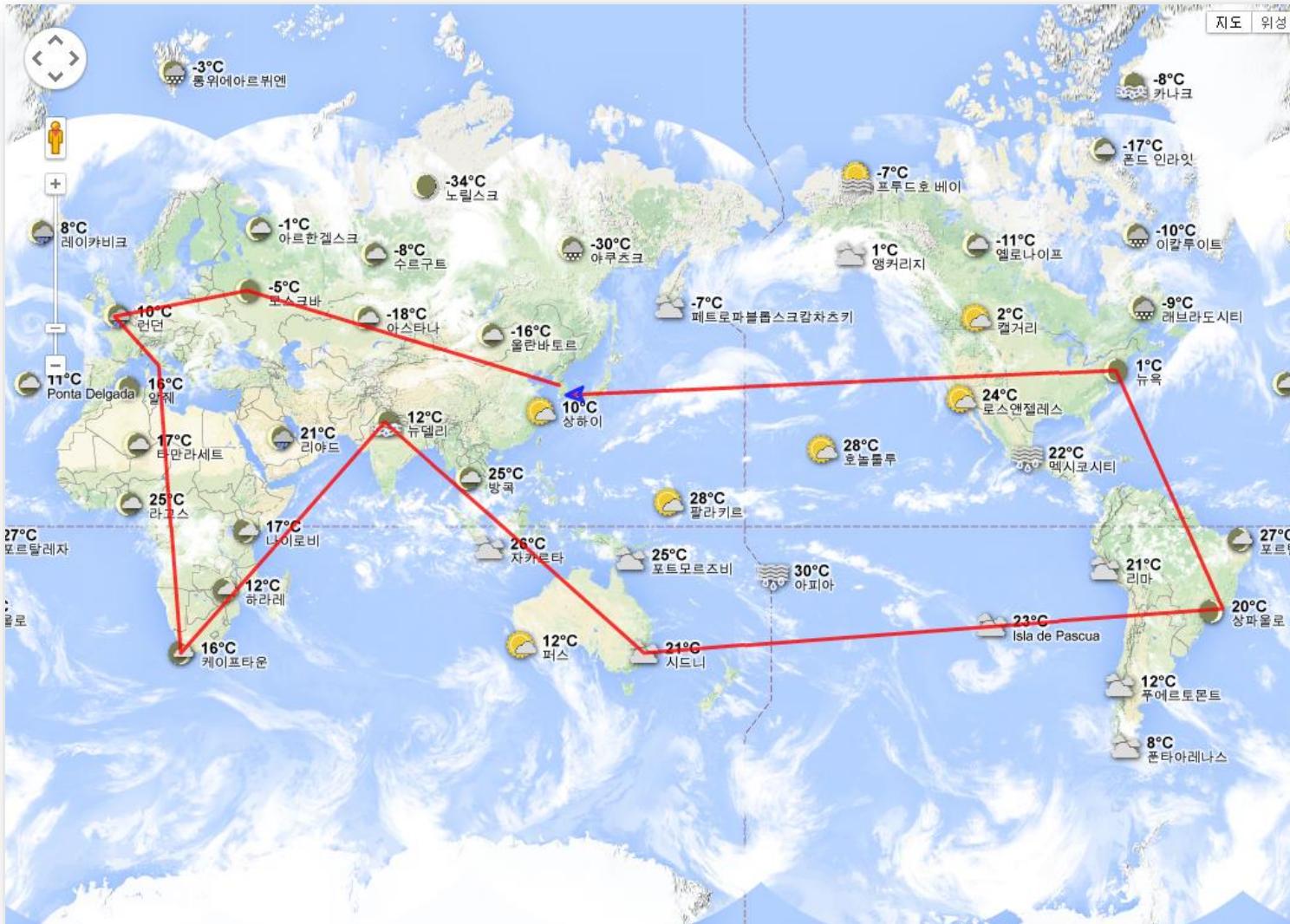


Target #2:

Using Google Map API V3



Challenge: Using Google Map API V3



교재 WEB 강의 소개



**명품 HTML5+
CSS3+ Javascript 웹 프로그래밍**

**명품 HTML5 +
CSS3 +
Javascript**

웹 프로그래밍

HTML5로 여러분의 무한한 상상력을 표현해 보세요!

Sir Tim Berners-Lee (1955.6.8 ~)



명품 웹 프로그래밍 소개

“웹 프로그래밍을
가장 쉽게 익힐 수 있는 책”

처음 웹 프로그래밍을 공부하는 입문자들도
모든 주제를 직관적으로 이해하고
빠르게 파악할 수 있습니다.

자세히보기 →



강력한 Q&A 피드백 제공

“빠르고, 간결하고, 정확한
저자의 직접적인 답변”

‘이거 이해가 잘 안되는데.. 물어볼 사람도 없고..’
더이상 고민하지 마세요.
명품 웹 프로그래밍 홈페이지에서는
누구나 저자가 직접 답변해주는
Q&A 게시판을 이용할 수 있습니다.

자세히보기 →



즉석 실행 가능한 예제 프로그램

“백문이 불여일견, 백견이 불여일타(打)!”

코드로만 설명되어 있는 예제들,
결과 화면이 있어도 이해가 잘 안되시죠?
예제 소스를 바탕으로, 내맘대로 수정한
코드를 즉석으로 웹 페이지로
변환해주는 예제 프로그램을 통해
모든 코드를 빠르고 쉽게
이해할 수 있습니다.



자세히보기 →

Notice

Test

2017-01-16 15:32

Know-How

Test

2017-01-17 14:04 관리자

관련 WEB 강의 소개 – w3schools.com

The screenshot shows the homepage of w3schools.com. At the top, there's a navigation bar with links for TUTORIALS, REFERENCES, and EXAMPLES. Below the navigation, there are three main sections: HTML, CSS, and JavaScript.

HTML Section: This section is titled "HTML: The language for building web pages". It features a "LEARN HTML" button and a "HTML REFERENCE" button. A code example is shown:

```
<!DOCTYPE html>
<html>
<title>HTML Tutorial</title>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

A "Try it Yourself" button is located at the bottom of this section.

CSS Section: This section is titled "CSS: The language for styling web pages". It features a "LEARN CSS" button and a "CSS REFERENCE" button. A code example is shown:

```
body {
    background-color: lightblue;
}
h1 {
    color: white;
    text-align: center;
}
p {
    font-family: verdana;
    font-size: 20px;
}
```

A "Try it Yourself" button is located at the bottom of this section.

JavaScript Section: This section is titled "JavaScript: The language for programming web pages". It features a "LEARN JAVASCRIPT" button and a "JAVASCRIPT REFERENCE" button. A code example is shown:

```
<script>
function myFunction() {
    var x = document.getElementById("demo");
    x.innerHTML = "Hello";
}
```

<http://www.w3schools.com>