

Open API : GoogleMap/Weather

Web Programming

Open API

- API (application programming interface)
 - 운영체제나 시스템, 애플리케이션, 라이브러리 등을 활용해 응용 프로그램을 작성할 수 있게 하는 다양한 인터페이스
 - Window API, Java API, HTML5 API, Android API
- Open API (Web API)
 - API 중에서 플랫폼의 기능 또는 콘텐츠를 외부에서 웹 프로토콜(HTTP)로 호출해 사용할 수 있게 개방(open)한 API
 - 웹을 통해 특정 서비스를 제공
 - 구글맵, 네이버 오픈 API, 날씨 제공 API, 공공 데이터 오픈 API

Ajax

- Asynchronous JavaScript and XML
- 비동기 통신 처리
- 자바스크립트를 이용해 서버와 데이터를 주고 받는 방식
 - 기존 : 동기식. 전체 화면 새로 변경. HTML 데이터 수신
 - Ajax : 화면 일부 변경. XML/HTML/JSON/CSV 데이터 수신
- 비동기처리 : 콜백함수 이용
 - 요청을 보내고, 응답 기다리지 않고 빠져 나와 원래/다른 작업 수행
 - 응답이 오면, 미리 등록해 놓은 콜백함수가 처리 (화면에 반영)
- 구글맵, 유튜브, 페이스북

전송 데이터 형식

- XML (eXtensible Markup Language)
 - 이용자 정의 태그로 감싸서 전달
 - `<myname>khlee</myname>`
- JSON (JavaScript Object Notation)
 - 자바스크립트 객체 표현 방식
 - `{ "key":value, "name":"khlee", "data":[0,1,2] }`
- CSV (Comma Separated Value)
 - 콤마로 분리된 텍스트
 - `"khlee",0,1,2,"yours",3,4,5`

지도를 지원하는 API 및 SDK

- Maps SDK for Android : Android 앱에 지도를 추가
- Maps SDK for iOS : iOS 앱에 지도를 추가
- Maps Static API
 - 웹사이트에 최소한의 코드로 삽입 가능한 간단한 지도 이미지를 추가
- Maps JavaScript API : 웹사이트에 대화형 지도를 추가
 - 자신만의 콘텐츠와 이미지를 사용하여 지도를 맞춤 설정
- Street View API : 360도 파노라마 형식의 실제 이미지를 삽입
- 지도 URL
 - Google 지도를 실행하고 교차 플랫폼 URL 구성표를 사용하여 검색, 또는 경로와 같은 작업을 시작

Maps Static API

- 다른 코드 없이 간단한 HTTP 요청을 사용하여 웹페이지에 Google 지도 이미지를 삽입
- URL을 통한 HTTP 요청에 응답하여 이미지 (GIF, PNG 또는 JPEG)를 반환
- 각 요청에 대해 지도의 위치, 이미지의 크기, 확대/축소 수준, 지도 유형, 지도상의 위치에 선택적 마커의 배치를 지정. 영숫자 문자를 사용하여 마커에 레이블을 붙일 수 있음
- Maps Static API 이미지는 `` 태그의 `src` 속성 내에 삽입되거나 다른 프로그래밍 언어의 이에 상응하는 속성 내에 삽입됨

Maps Static API

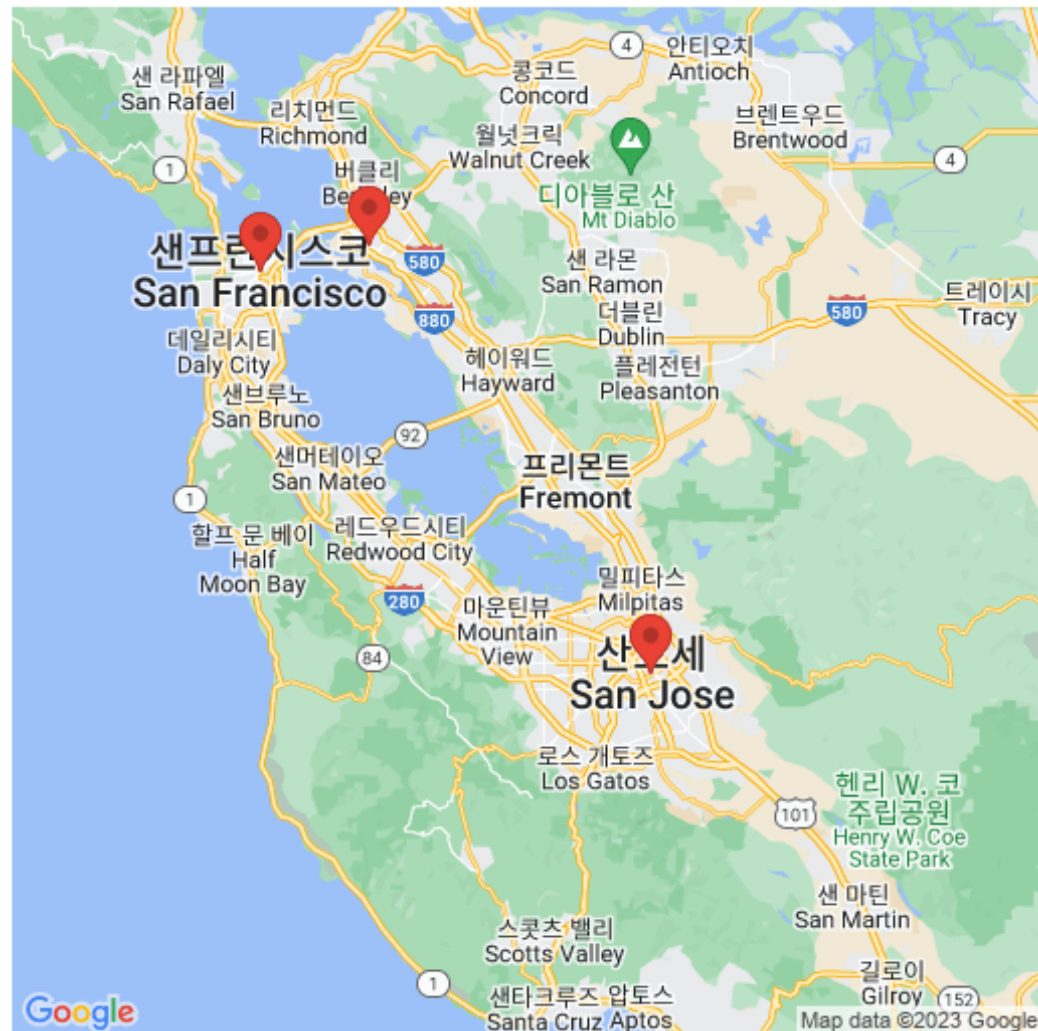
```
<img border=0 src=
"https://maps.googleapis.com/maps/api/staticmap?
center=Brooklyn+Bridge,New+York,NY&
zoom=13&size=600x300&
maptype=roadmap&
markers=color:blue%7Clabel:S%7C40.702147,-74.015794&
markers=color:green%7Clabel:G%7C40.711614,-74.012318&
markers=color:red%7Clabel:C%7C40.718217,-73.998284&
key=YOUR_API_KEY"
Alt= "Manhattan Faraway" />    // %7C ➡ | (pipe)
```

Maps Static API



Maps Static API

```
<body>
<p>  </p> </body>
</body>
```



URL and Parameters

- <https://maps.googleapis.com/maps/api/staticmap?parameters>
- 위치 매개변수
 - center (마커가 없는 경우 필수) : 지도의 중심. 쉼표로 구분된 {latitude,longitude} 쌍 (예: '40.714728,-73.998672') 또는 문자열 (예: ' city hall, new york, ny ')
 - Zoom (마커가 없는 경우 필수) – 지도의 확대 수준. 숫자값
- 지도 매개변수
 - size (필수) : 지도 이미지의 직사각형 크기. {h}x{v} 형식의 문자열(ex. 500x400)
 - scale (선택사항) : scale=1 or 2
 - format (선택사항) : 결과 이미지의 형식 (default PNG)
 - maptype (선택사항) : 지도 유형 (roadmap, satellite, hybrid, terrain)
 - language (선택사항) : 레이블 표시 언어
 - region (선택사항) : 국경선

URL and Parameters

- 특성 매개변수
 - map_id (선택사항) : 특정 지도의 식별자를 지정
 - markers (선택사항) : 이미지에 첨부할 마커 정의. 파이프 문자(|) 구분 복수 지정 가능
 - path (선택사항) : 이미지에 오버레이할 둘 이상 연결 지점의 단일 경로 정의
 - visible (선택사항) : 표시를 지정해야 하는 하나 이상의 위치를 지정
 - style (선택사항) : 지도상의 특정 지형지물 (도로, 공원 등)을 변경하여 표시하기 위해 사용자 지정 스타일을 정의
- 키 및 서명 매개변수
 - key (필수)
 - signature (권장) : 권한 확인 디지털 서명
- URL 크기 제한
 - Maps Static API URL은 16,384자(영문 기준)로 제한

Maps Embed API

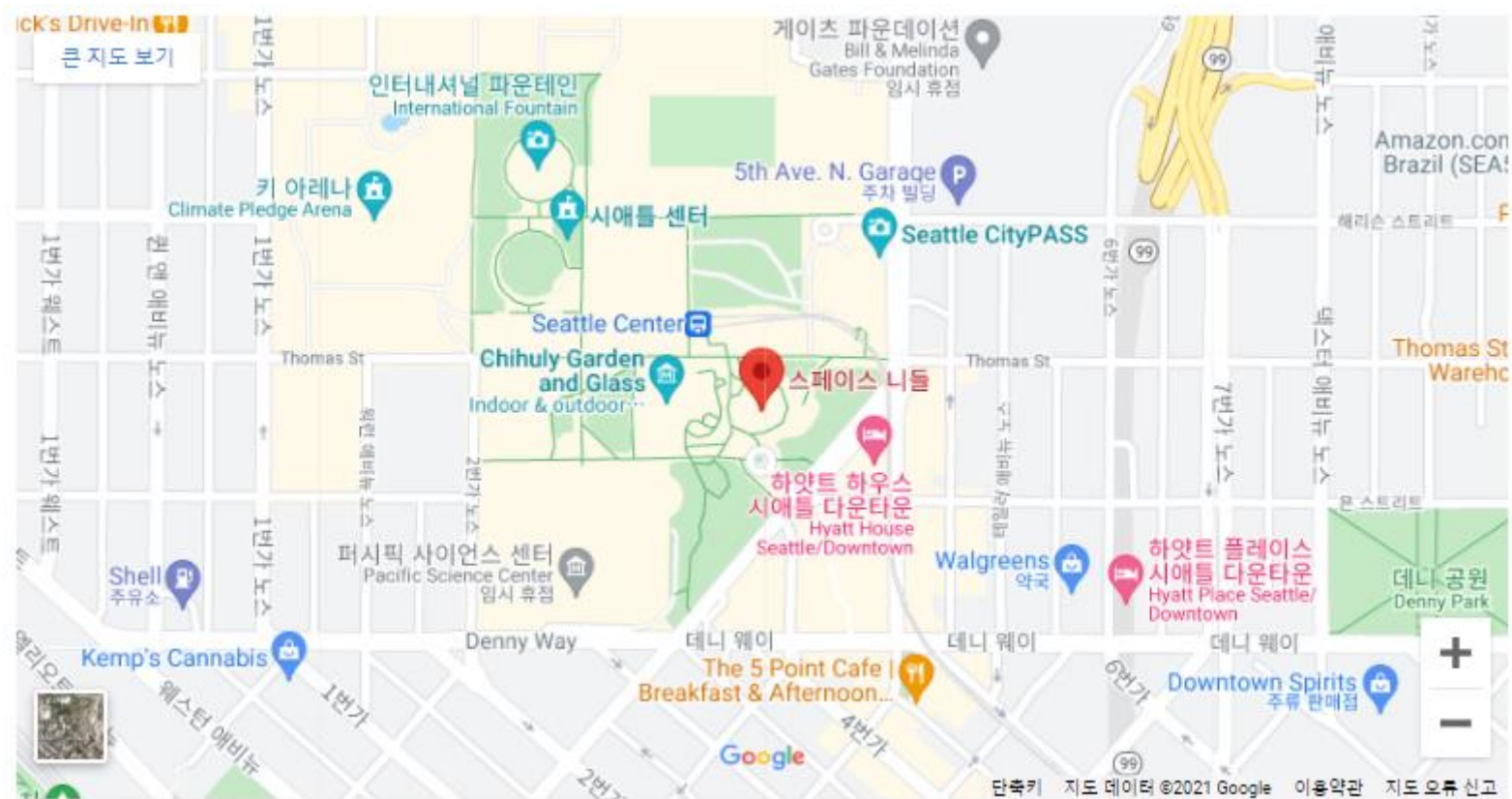
- 웹페이지에 대화형 지도 또는 스트리트 뷰 파노라마를 배치
- 간단한 HTTP 요청을 사용. JavaScript 불필요.

```
<iframe
  width="600" height="450"
  style="border:0"
  loading="lazy"
  allowfullscreen
  src="https://www.google.com/maps/embed/v1/place?key=API_KEY
    &q=Space+Needle,Seattle+WA">
</iframe>
```

Maps Embed API

```
<html>
<body>
<iframe
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3159
.74555268445!2d127.07528725111477!3d37.63167262805309!2m3!1f0!2f0!
3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x357cbbd569641237%3A0x7b8f
e3317caedc65!2z7lSc7Jq46rO87ZWZ6riw7lig64yA7ZWZ6rWQ!5e0!3m2!1sk
o!2skr!4v1659616098174!5m2!1sko!2skr" width="600" height="450"
style="border:0;" allowfullscreen="" loading="lazy" referrerpolicy="no-
referrer-when-downgrade"> </iframe>
</body>
</html>
```

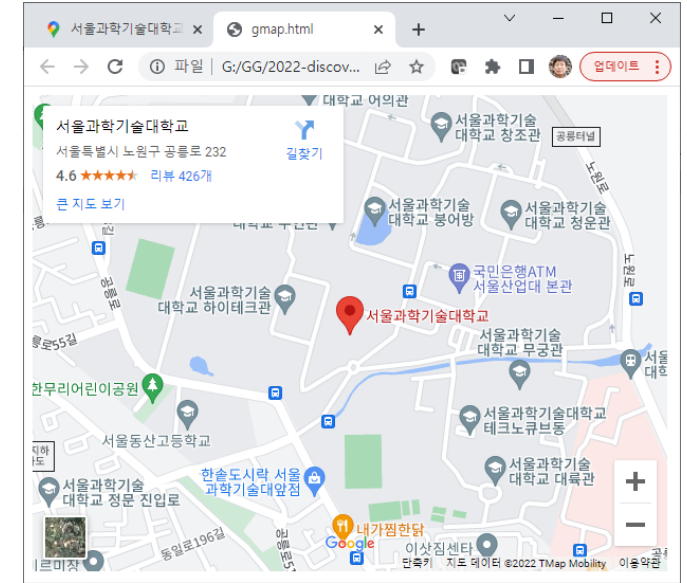
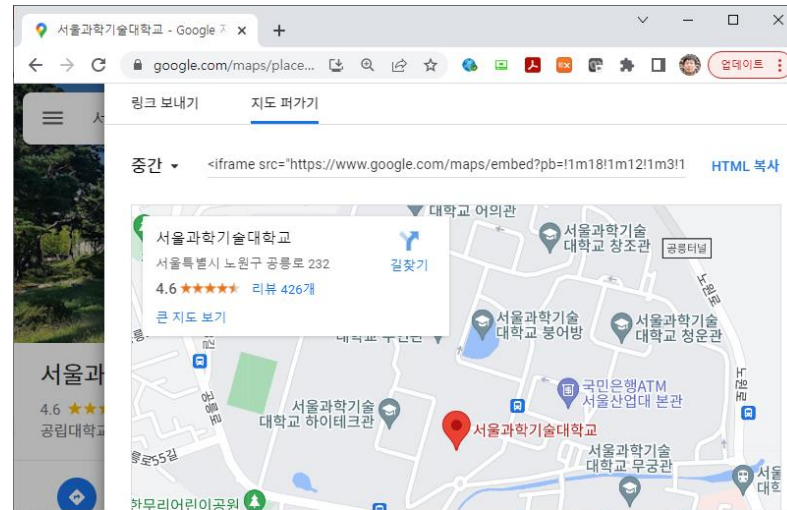
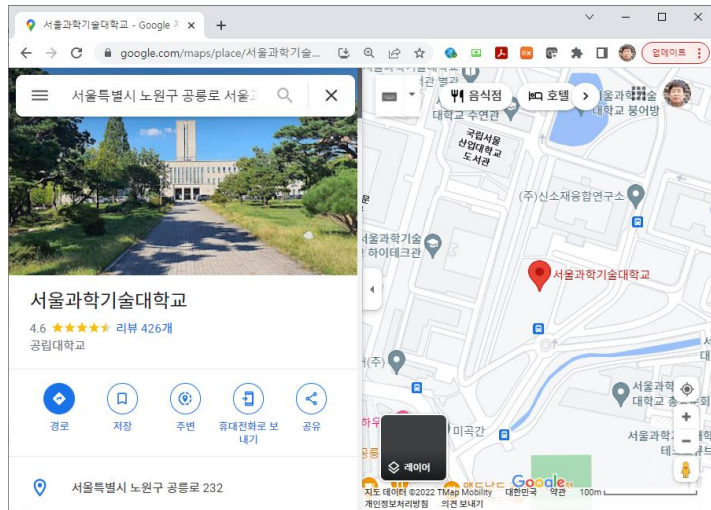
Maps Embed API



이동, 확대/축소, 맵 타입 변경 가능

키없이 페이지에 구글맵 넣기

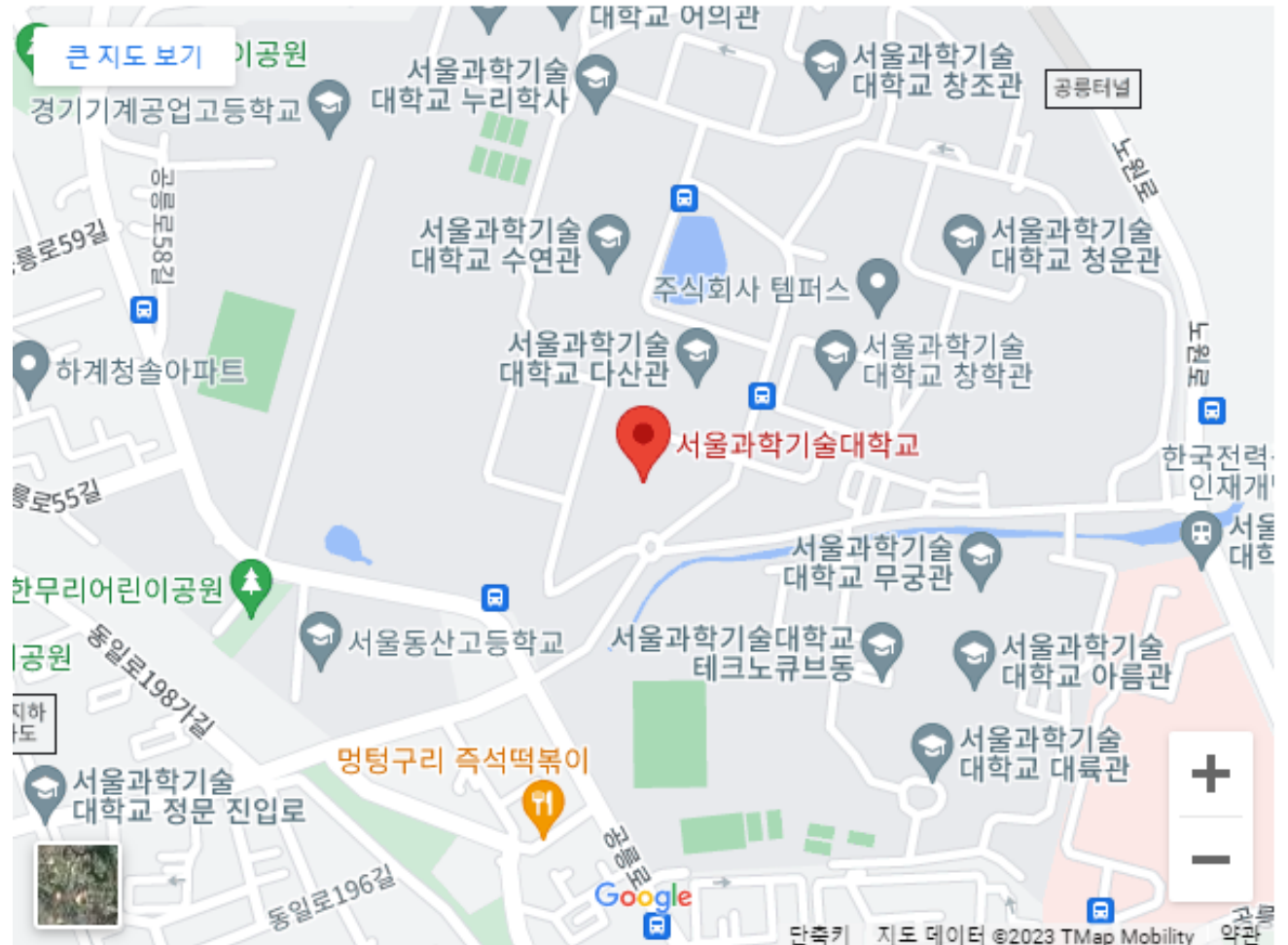
- 구글맵->검색->공유->지도퍼가기->복사->붙여넣기



- <iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3159.74555268445!2d127.07528725111477!3d37.63167262805309!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x357cbbd569641237%3A0x7b8fe3317caedc65!2z7ISc7Jq46rO87ZWZ6riw7lig64yA7ZWZ6rWQ!5e0!3m2!1sko!2skr!4v1659616098174!5m2!1sko!2skr" width="600" height="450" style="border:0;" allowfullscreen="" loading="lazy" referrerpolicy="no-referrer-when-downgrade"></iframe>

키없이 페이지에 구글맵 넣기

```
<html> <body> <iframe  
src="https://www.google.com/maps/e  
mbed?pb=!1m18!1m12!1m3!1d3159.7  
4555268445!2d127.07528725111477!3  
d37.63167262805309!2m3!1f0!2f0!3f0!  
3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0  
x357cbbd569641237%3A0x7b8fe3317  
caedc65!2z7ISc7Jq46rO87ZWZ6riw7lig  
64yA7ZWZ6rWQ!5e0!3m2!1sko!2skr!4  
v1659616098174!5m2!1sko!2skr"  
width="600" height="450"  
style="border:0;" allowfullscreen=""  
loading="lazy" referrerpolicy="no-  
referrer-when-downgrade"> </iframe>  
</body> </html>
```



Maps JavaScript API

- 세부적인 맞춤설정이 가능한 대화형 동적 지도, 위치, 지리정보 환경을 웹 앱에 구축
- The Maps JavaScript API features four basic map types (roadmap, satellite, hybrid, and terrain) which you can modify using layers and styles, controls and events, and various services and libraries

Maps JavaScript API 기능



3D 지도 만들기

Maps JavaScript API를 사용하여 3D 지도를 만드는 방법을 알아봅니다.



마커에 정보 창 추가

사용자가 마커를 클릭하면 추가 정보와 컨텍스트가 포함된 정보 창을 표시합니다.



마커가 포함된 첫 지도 만들기

Maps JavaScript API를 로드하고 마커가 있는 지도를 웹 앱에 추가하는 방법을 알아봅니다.



지도 스타일 맞춤설정

도로, 지형 지물, 관심 장소 등 지도의 모든 측면을 맞춤설정합니다.

지도 유형

도로, 위성, 하이브리드, 지형, 맞춤 지도를 표시합니다.

현지화

지도의 텍스트를 40개 이상의 언어로 자동 현지화합니다.

마커

기본 Google 지도 마커를 표시하거나 나만의 **맞춤 스타일 마커**를 추가합니다.

UI 컨트롤

지도에 표시되는 UI 컨트롤을 맞춤설정합니다.

일정

사용자 상호작용 및 수명 주기 이벤트에 반응하는 코드를 작성합니다.

WebGL 오버레이

WebGL의 기능을 활용하여 벡터 기본 지도에 풍부한 3D 및 2D 경험을 구축합니다.

정보 창

대화형 정보 창을 사용하여 마커에 향상된 컨텍스트와 정보를 추가합니다.

도형

기본 제공 함수를 사용하여 지도에 **다중선**, **다각형** 등 다양한 도형을 그립니다.

맞춤 오버레이

지도에 데이터, 이미지 등을 표시하는 맞춤 오버레이를 만듭니다.

지면 오버레이

이동하거나 확대/축소할 때 지도와 동기화된 상태로 유지하는 맞춤 이미지를 오버레이합니다.

데이터 레이어

지도에 GeoJSON 및 기타 데이터 유형을 다양한 형식으로 표시합니다.

맞춤 스타일 지정

지도의 거의 모든 시각적인 측면에 맞춤 모양 스타일을 지정합니다.

기울기 및 회전

벡터 기본 지도를 프로그래매틱 방식을 통해 3차원적으로 기울이거나 회전합니다.

마커 클러스터링

수많은 마커를 그룹화하여 깔끔한 사용자 환경을 조성합니다.

히트맵

지리적 위치에서 데이터의 밀도를 시각화합니다.

라이브러리

Maps JS API를 로드하여 확장 기능을 추가할 때 선택적 라이브러리를 부트스트랩

라이브러리 개요

Maps JS API를 로드할 때 사용할 가능한 라이브러리를 부트스트랩하는 방법을 알아봅니다.

그리기 라이브러리

사용자에게 지도에 그릴 수 있는 대화형 기능을 제공합니다.

도형 라이브러리

지표면의 도형 데이터를 계산하는 유틸리티 함수를 사용합니다.

장소 라이브러리

주변 장소를 검색하고, Place Autocomplete를 사용 설정하고, 장소 세부정보 및 사진을 검색합니다.

로컬 컨텍스트 라이브러리(베타)

사진 갤러리로 구성된 장소 선택기가 포함된 대화형 지도를 추가합니다.

시각화 라이브러리

히트맵을 사용하여 지리적 위치의 데이터 강도를 시각화합니다.

서비스

기본 제공 데이터 서비스를 사용하여 웹 앱의 기능을 강화

경로 서비스

지도에서 두 지점 간의 경로 및 최대 25개의 경유지를 찾습니다.

거리 행렬 서비스

출발지와 목적지 행렬에 대한 이동 거리와 시간을 가져옵니다.

고도 서비스

해저를 포함하여 지표면의 위치 고도를 가져옵니다.

지오코딩 서비스

주소 또는 장소 ID를 좌표로 변환하거나 그 반대로 변환합니다.

최대 확대/축소 이미지 서비스

특정 위치에서 사용 가능한 이미지의 최대 확대/축소 수준을 쿼리합니다.

스트리트 뷰 서비스

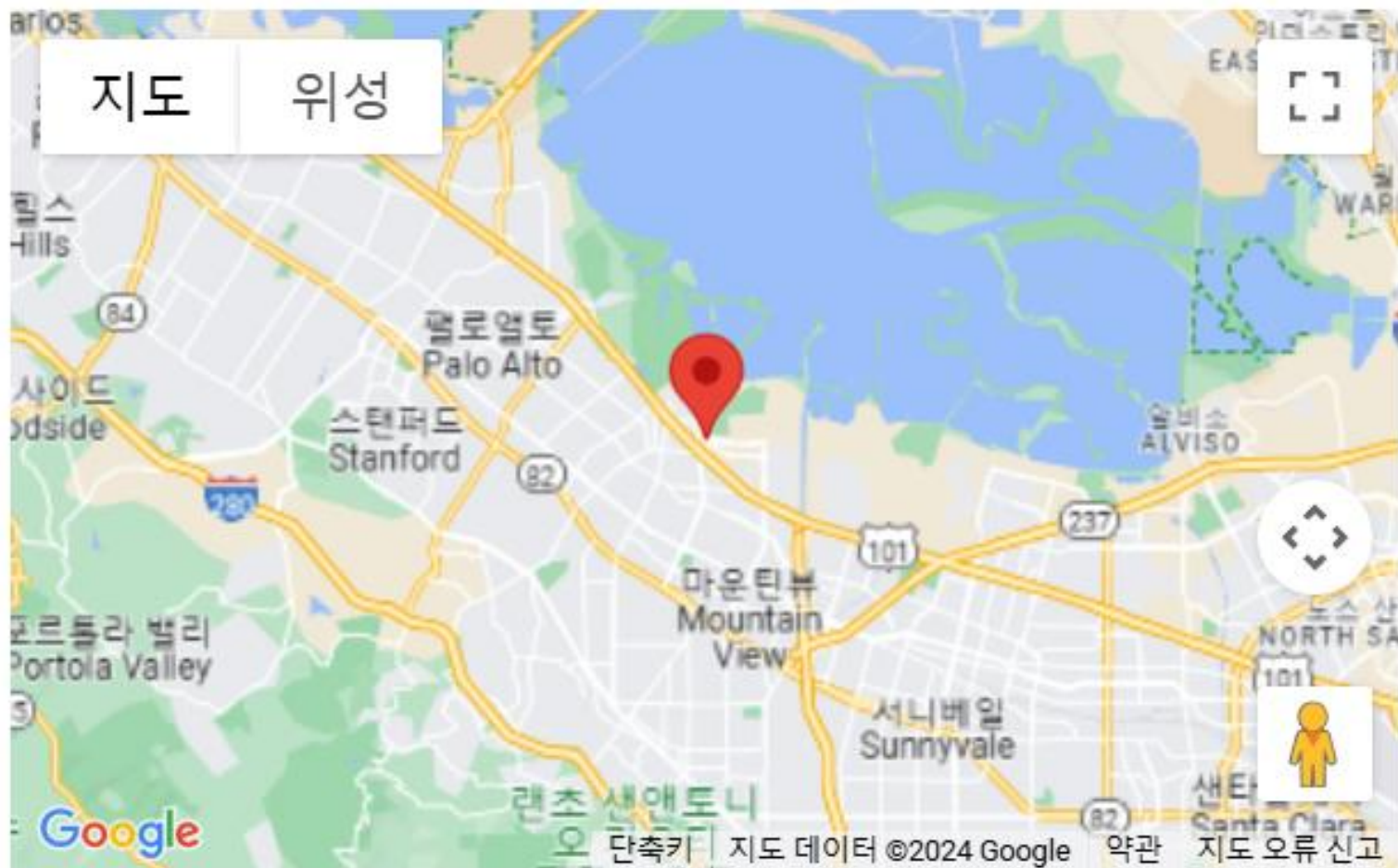
웹 앱에 360도 스트리트 뷰 이미지를 추가하거나 자신만의 이미지를 업로드합니다.

Maps JavaScript API

```
<html> <head> <title>Simple Map</title>
<script>
  let map;
  function initMap() {
    map = new google.maps.Map(document.getElementById("map"), {
      center: { lat: -34.397, lng: 150.644 }, zoom: 8, });
  }
</script> </head> <body>
<div id="map"> </div>
<!-- Async script executes immediately and must be after any DOM elements used in callback. -->
<script src="https://maps.googleapis.com/maps/api/js?
      key=YOUR_API_KEY&
      callback=initMap&
      libraries=&v=weekly" async>

</script>
</body> </html>
```

Maps JavaScript API



구글 맵 API 키 가져오기

- 구글 계정 등록
- 구글 개발자 사이트에서 구글 맵 웹용 API 키 생성
 - <https://developers.google.com/>
 - Maps Platform
 - 시작하기
 - 웹(</>) 선택
 - 프로젝트 선택기
 - 프로젝트 만들기
 - 사용자 인증 정보
 - API 키 만들기

구글 맵 API 키 만들기

시작하기 [↗](#)

또는 다음 탭을 클릭하여 필요한 설정 단계를 완료할 수도 있습니다.

1단계
[Set up your project](#)

2단계
Enable APIs or SDKs

3단계
Get an API Key

1. Google Cloud Console의 프로젝트 선택기 페이지에서 프로젝트 만들기를 클릭하여 새 Cloud 프로젝트를 만듭니다.

★ 참고: 이 절차에서 생성한 리소스를 유지하지 않으려면 기존 프로젝트를 선택하지 말고 프로젝트를 새로 만드세요. 작업이 끝나면 프로젝트를 삭제하여 프로젝트와 관련된 모든 리소스를 삭제할 수 있습니다.

[프로젝트 선택기 페이지로 이동](#) [↗](#)

2. Cloud 프로젝트에 결제가 사용 설정되어 있어야 하므로 프로젝트에 결제가 사용 설정되어 있는지 확인합니다 [↗](#).

Google Cloud에서는 300달러의 무료 체험판을 제공하며 Google Maps Platform에서는 매월 200달러의 크레딧을 제공합니다. 자세한 내용은 [결제 계정 크레딧](#) 및 [결제](#)를 참고하세요.



API 키 제한사항

- URL

- 도메인 및 모든 하위 도메인
 - https://example.com
 - https://*.example.com
- http://www.example.com:8000

- IP

- 192.168.0.1, 172.16.0.0/12,
- 2001:db8::1 또는 2001:db8::/64

- Android 앱

- API 키 사용을 지정된 Android 앱으로 제한
- 각 앱의 패키지 이름과 SHA-1 인증서 지문을 추가

이름 *


추가 정보

API Key

key=API_KEY

생성일

키 제한사항

 이 키는 제한되지 않습니다. 무단 사용을 방지하려면 API를 사용할 수 있는 위치와 대상을 제한하는 것이 좋습니다.[자세히 알아보기](#)

애플리케이션 제한사항 설정

애플리케이션 제한사항은 API 키 사용을 특정 웹사이트, IP 주소, Android 애플리케이션 또는 iOS 애플리케이션으로 제한합니다. 키별로 애플리케이션 제한사항 1개를 설정할 수 있습니다.

☒ 없음
☐ 웹사이트
☐ IP 주소
☐ Android 앱
☐ iOS 앱

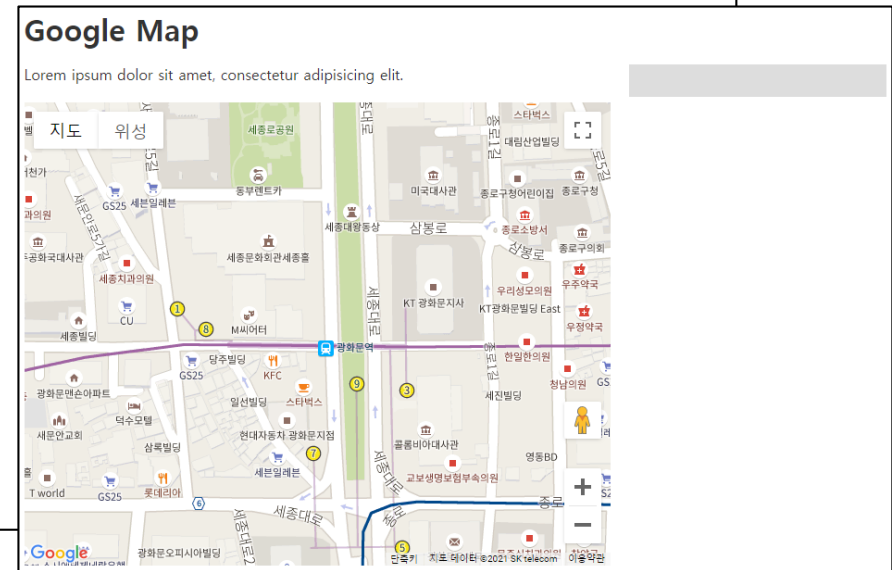
API 제한사항

API 제한사항은 이 키를 호출할 수 있는 사용 설정된 API를 지정합니다.

☒ 키 제한 안함
이 키는 모든 API를 호출할 수 있습니다.
☐ 키 제한

구글 맵 HTML 예제

```
<header>
  <h1>Google Map</h1>
</header>
<main>
  <section>
    <article>
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.</p>
      <div id="map"> <br> &nbsp;지도가 표시될 영역</div>
    </article>
    <aside>
      <form action="">
        <input type="text" id="name" name="name">
      </form>
    </aside>
  </section>
</main>
<script> </script>
```



HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Simple Map</title>
    <script src="https://polyfill.io/v3/polyfill.min.js?features=default"> </script>
    <link rel="stylesheet" type="text/css" href="./style.css" />
    <script src="./index.js"> </script>
  </head>
  <body>
    <div id="map"> </div>
    <!-- Async script executes immediately and must be after any DOM elements used in callback. -->
    <script src="https://maps.googleapis.com/maps/api/js?
      key=YOUR_API_KEY &callback=initMap&libraries=&v=weekly" async> </script>
  </body>
</html>
```

CSS

```
/* Always set the map height explicitly to define the size of the div  
   * element that contains the map. */
```

```
#map {  
    height: 100%;  
}
```

```
/* Optional: Makes the sample page fill the window. */
```

```
html,  
body {  
    height: 100%;  
    margin: 0;  
    padding: 0;  
}
```

google.maps.Map()

- new google.maps.Map(HTMLElement, MapOptions)

Parameter	Description
<i>HTMLElement</i>	Specifies in what HTML element to put the map
<i>MapOptions</i>	A MapOptions object that holds the map initialization variables/options

```
let map;

function initMap() {
  map = new google.maps.Map(document.getElementById("map"), {
    center: { lat: -34.397, lng: 150.644 },
    zoom: 8,
  });
}
```

MapOptions

속성	값	설명
center	LatLng	Required. 기본 맵 센터 지정
mapTypeId	MapTypeId	기본 지도 유형 (HYBRID, ROADMAP, SATELLITE, TERRAIN)
maxZoom	number	지도의 최대 줌 레벨
minZoom	number	지도에 대한 최소 줌 레벨
scaleControl	boolean	기본값은 스케일 제어 / 비활성 상태
styles	Array.<MapStyleType>	스타일은 기본 지도 유형 각각에 적용
zoom	number	Required. 기본 지도 줌 레벨 ((전세계)0-18)
zoomControl	boolean	줌 컨트롤의 활성화 / 비활성화 상태
zoomControlOptions	ZoomControlOptions	줌 컨트롤의 표시 옵션

마커 추가하기

```
<div id="map">
<script>
function initMap() {
  var map = new google.maps.Map(document.getElementById('map'), {
    center: { lat: 37.571717, lng: 126.976479 }, scrollwheel: false, zoom: 17
  });
  // 마커 객체 생성
  var marker = new google.maps.Marker({
    map: map,
    position: { lat: 37.571717, lng: 126.976479 },
    title: '여기가 내 위치입니다!'
  });
}
</script>
<script src="https://maps.googleapis.com/maps/api/js?
  key=AlzaSyA79F3HwKaIzzGfFmXbW6esal6fqbxTi0I &callback=initMap"
  async defer> </script>
```

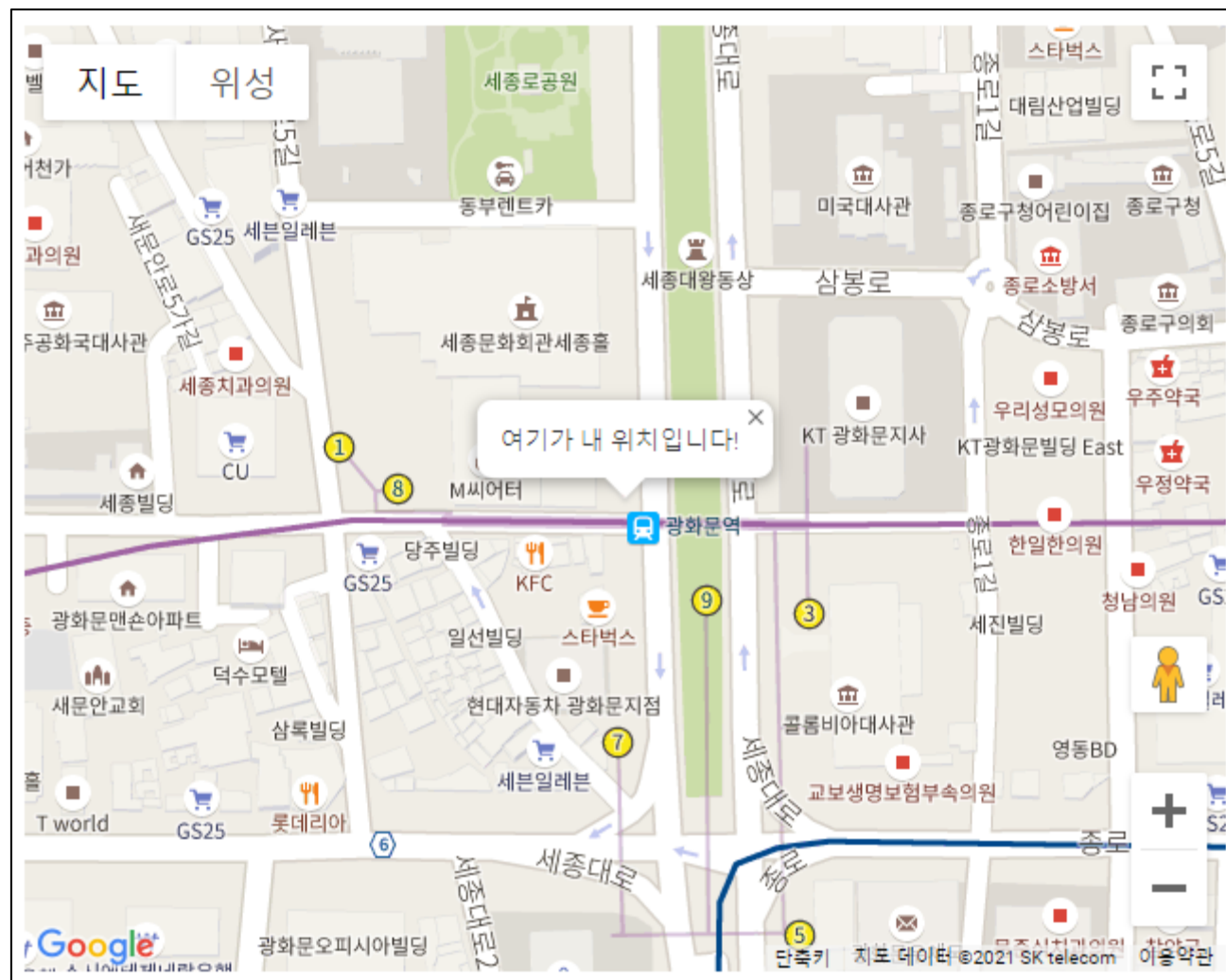
마커 추가하기



정보창 추가하기

```
<script>
function initMap() {
  var map = new google.maps.Map(document.getElementById('map'), {
    center: {lat: 37.571717, lng: 126.976479}, scrollwheel: false, zoom: 17 });
  var marker = new google.maps.Marker({ map: map, position: {lat: 37.571717, lng: 126.976479},
    title: '여기가 내 위치입니다!' });
  // 정보창 객체를 추가
  var infoWindow = new google.maps.InfoWindow({map: map});
  var my_position = {lat: 37.571717, lng: 126.976479};
  infoWindow.setPosition(my_position);
  infoWindow.setContent('여기가 내 위치입니다!');
}
</script>
<script src="https://maps.googleapis.com/maps/api/js?
  key=AlzaSyA79F3HwKaIZZGfFmXbW6esal6fqbxti0I &callback=initMap" async defer>
</script>
```

정보창 추가하기



기능개선 : 스타일 지정

```
// 맵 스타일 정의
var styleArray = [
    { featureType: 'all', stylers: [ { saturation: -80 } ] },
    { featureType: 'road.arterial', elementType: 'geometry',
      stylers: [ { hue: '#00ffee' }, { saturation: 50 } ] },
    { featureType: 'poi.business', elementType: 'labels', stylers: [ { visibility: 'off' } ] }
];

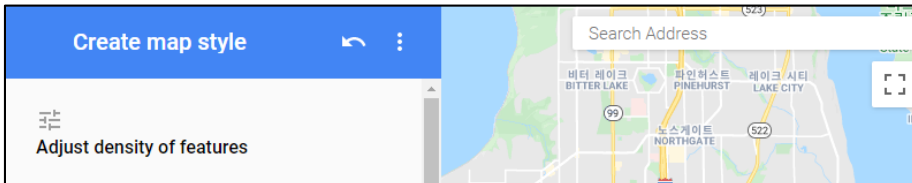
// 위치 정보 변수
var my_position = { lat: 37.667306, lng: 126.699969 };

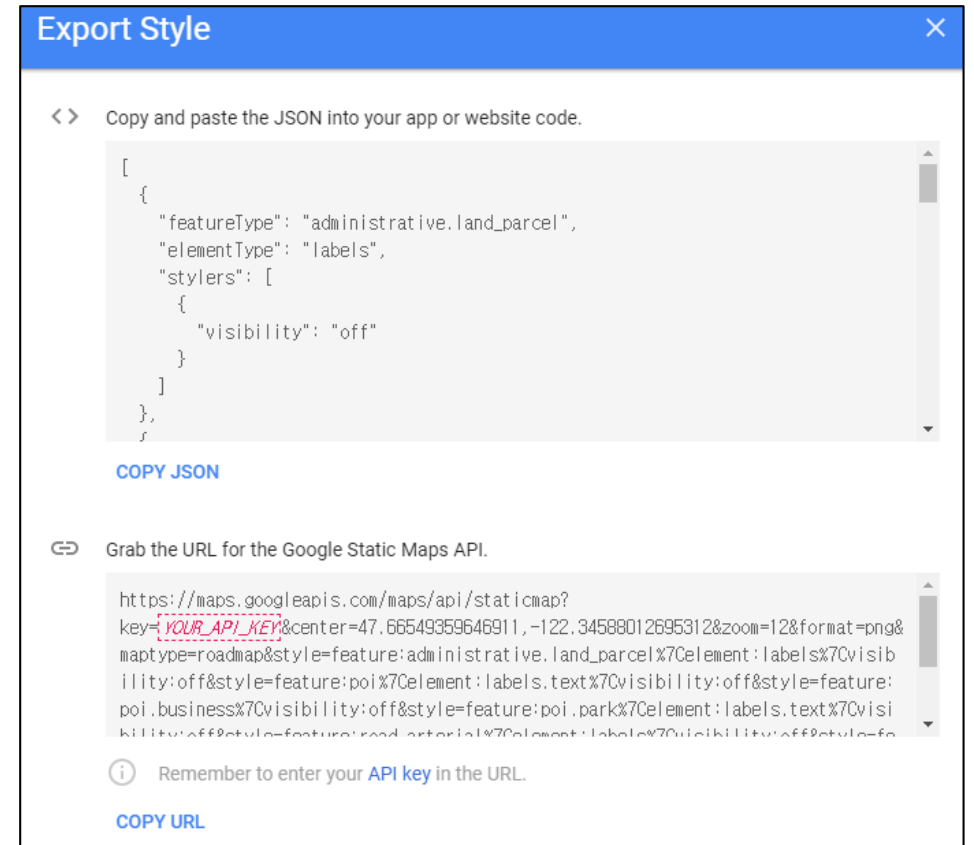
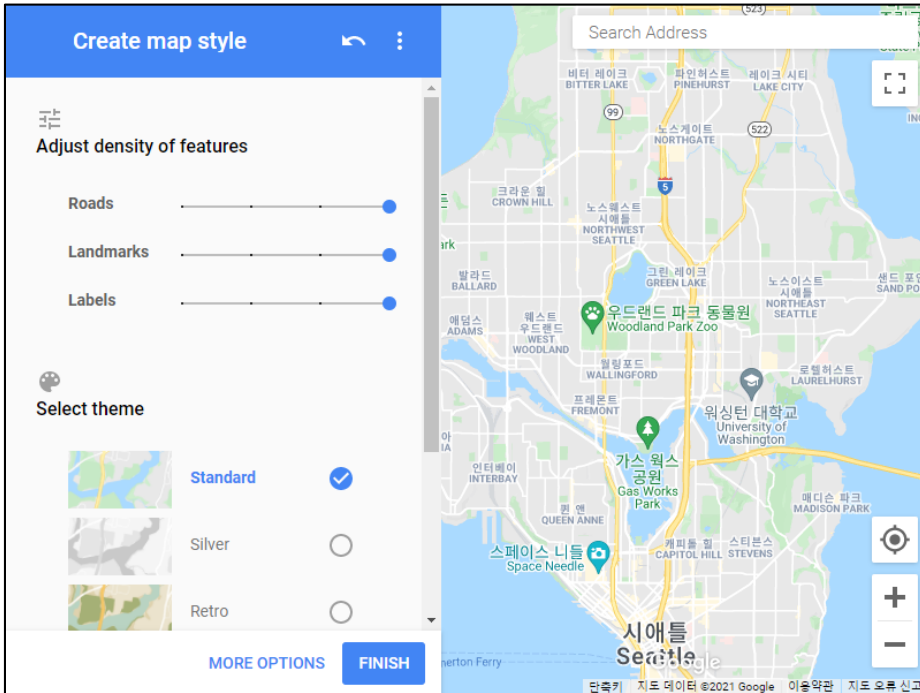
// 맵 객체를 생성하고 id='map'에 지도 표시
var map = new google.maps.Map(document.getElementById('map'), {
    center: my_position, scrollwheel: false, zoom: 17,
    styles: styleArray
});
```

스타일 지정



스타일 지정 지도 마법사

- <https://mapstyle.withgoogle.com/>
 - Create
 - 선택
 - finish
- 



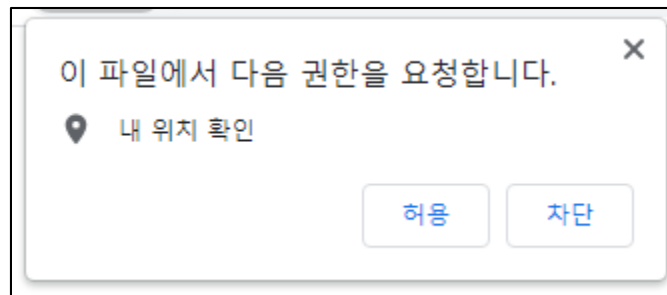
스타일 지정 지도 마법사 출력 (JSON,URL)

```
[ { "featureType": "administrative.land_parcel", "elementType": "labels", "stylers":  
[ { "visibility": "off" } ] }, { "featureType": "poi", "elementType": "labels.text",  
"stylers": [ { "visibility": "off" } ] }, { "featureType": "poi.business", "stylers":  
[ { "visibility": "off" } ] }, { "featureType": "poi.park", "elementType": "labels.text",  
"stylers": [ { "visibility": "off" } ] }, { "featureType": "road.arterial",  
"elementType": "labels", "stylers": [ { "visibility": "off" } ] }, { "featureType":  
"road.highway", "elementType": "labels", "stylers": [ { "visibility": "off" } ] },  
{ "featureType": "road.local", "stylers": [ { "visibility": "off" } ] }, { "featureType":  
"road.local", "elementType": "labels", "stylers": [ { "visibility": "off" } ] } ]
```

```
https://maps.googleapis.com/maps/api/staticmap?key=YOUR_API_KEY&center=47.66549  
359646911,-122.34588012695312&zoom=12&format=png&maptype=roadmap&  
style=feature:administrative.land_parcel%7Celement:labels%7Cvisibility:off&styl  
e=feature:poi%7Celement:labels.text%7Cvisibility:off&style=feature:poi.business  
%7Cvisibility:off&style=feature:poi.park%7Celement:labels.text%7Cvisibility:off  
&style=feature:road.arterial%7Celement:labels%7Cvisibility:off&style=feature:ro  
ad.highway%7Celement:labels%7Cvisibility:off&style=feature:road.local%7Cvisibil  
ity:off&style=feature:road.local%7Celement:labels%7Cvisibility:off&size=480x360
```

Geolocation

- 모바일 디바이스 : GPS 센서
- PC : IP 기반 정보
- 권한 허용
- HTML5 Geolocation API
 - `getCurrentPosition()` : 현재 위치 요청
 - `watchPosition()` : 위치 경로 추적



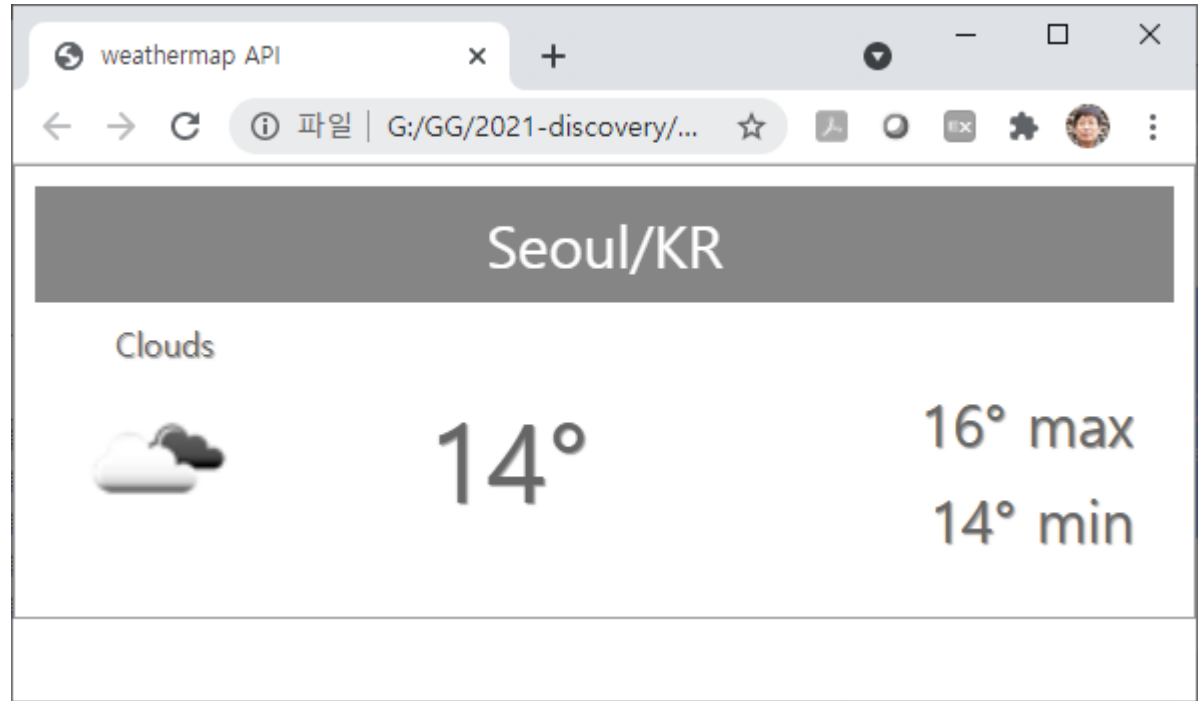
```
function initMap() {  
    var map = new google.maps.Map(document.getElementById('map'), {  
        center: {lat: -34.397, lng: 130.644}, scrollwheel: false, zoom: 17 });  
    // 정보창 객체를 추가  
    var infoWindow = new google.maps.InfoWindow({map: map});  
    // HTML5 위치 정보 요청  
    if (navigator.geolocation) { // 지오로케이션 사용 가능한 경우  
        navigator.geolocation.getCurrentPosition(function(position) { // 위치 얻기 성공  
            var pos = { lat: position.coords.latitude, lng: position.coords.longitude };  
            infoWindow.setPosition(pos); infoWindow.setContent('여기가 내 위치입니다!');  
            map.setCenter(pos);  
        }, function() { // 연결 실패  
            handleLocationError(true, infoWindow, map.getCenter());  
        });  
    } else { // 지오로케이션 사용 불가능한 경우  
        handleLocationError(false, infoWindow, map.getCenter());  
    }  
}  
// 지오로케이션 오류 처리  
function handleLocationError(browserHasGeolocation, infoWindow, pos) {  
    infoWindow.setPosition(pos); infoWindow.setContent(browserHasGeolocation ?  
        '오류: 지오로케이션 연결 실패' : '오류: 브라우저에서 지오로케이션을 지원하지 않음');  
}
```


Geolocation 결과 (PC)



Weathrt API

- openweather API
- jQuery
- JSONP
- JSON.stringify



OpenWeather API 개요

- 데이터 형식 : JSON, XML, HTML
- API 홈 : openweathermap.org
- 요구 사항 : 키 발급 후 사용
- 사용료 : 무료, 과금

키 발급 받기

- 사이트 접속
- 회원 가입
- 로그인
- 무료 옵션 선택
- API Keys 생성
- 키 복사 사용

API 사용 방법

- 도시명으로 사용
 - <http://api.openweathermap.org/data/2.5/weather?q=seoul,kr>
- 도시 id로 사용
 - <http://api.openweathermap.org/data/2.5/weather?id=183547>
- 위도, 경도 값으로 사용
 - <http://api.openweathermap.org/data/2.5/weather?lat=위도값&lon=경도값>

jQuery code

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.3/jquery.min.js" > </script>
<script>
var url = 'http://api.openweathermap.org/data/2.5/weather?
          q=seoul&APPID=e11462160015cffa69954c9f67741b7b';
$.getJSON(url, function(data) {
    var sys = data.sys;           // 국가명, 일출/일몰
    var city = data.name;        // 도시명
    var weather = data.weather;  // 날씨 객체
    var main = data.main;        // 온도 기압 관련 객체
    var wmain = weather[0].main; // 구름 상태(Cloudiness)
    var w_id = weather[0].id;    // 날씨상태 id 코드
    ...
})
</script>
```

JavaScript 요청/응답

요청페이지

```
<body>
<script>function test(input) { document.body.innerHTML += JSON.stringify(input); }
</script> <script
src="http://api.openweathermap.org/data/2.5/weather?q=seoul&APPID=e11462160015cffa69954c9
f67741b7b&callback=test" > </script>
</body>
```

결과

```
{
  "coord": {"lon": 126.9778, "lat": 37.5683},
  "weather": [{"id": 804, "main": "Clouds", "description": "overcast clouds", "icon": "04n"}],
  "base": "stations",
  "main": {"temp": 298.96, "feels_like": 299.27, "temp_min": 298.38, "temp_max": 299.86, "pressure": 1008, "humidity": 64, "sea_level": 1008, "grnd_level": 1002},
  "visibility": 10000,
  "wind": {"speed": 0.65, "deg": 45, "gust": 2.05},
  "clouds": {"all": 99},
  "dt": 1628774089,
  "sys": {"type": 1, "id": 5509, "country": "KR", "sunrise": 1628714711, "sunset": 1628764160},
  "timezone": 32400,
  "id": 1835848,
  "name": "Seoul",
  "cod": 200
}
```

Open Weather API x +

← → ↻ 주의 요함 | comunix.seoultech.ac.kr/~khlee/dynamicweb/ch12/openwaether-javascript.html

```
{"coord":{"lon":126.9778,"lat":37.5683},"weather":[{"id":800,"main":"Clear","description":"clear sky","icon":"01n"}],"base":"stations","main":{"temp":297.71,"feels_like":298.02,"temp_min":296.91,"temp_max":299.86,"pressure":1008,"humidity":69},"speed":1.03,"deg":150,"clouds":{"all":0},"dt":1628776574,"sys":{"type":1,"id":8105,"country":"KR","sunrise":1628714711,"sunset":1628764160},"timezone":32400,"id":183}
```

Elements Console Sources **Network** Performance Memory Application >> 1

● ○ 🔍 ☐ Preserve log ☐ Disable cache No throttling ▼ 📶 ⬆️ ⬇️

Filter ☐ Hide data URLs **All** Fetch/XHR JS CSS Img Media Font Doc WS Wasm Manifest Other

☐ Has blocked cookies ☐ Blocked Requests

50 ms 100 ms 150 ms 200 ms 250 ms

Name × Headers Preview **Response** Initiator Timing Cookies

1 weather?q=seoul&APPID.. test({"coord":{"lon":126.9778,"lat":37.5683},"weather":[{"id":800,"main":"Clear",

4 requests | 3.3 kB transferred { } Line 1, Column 1

Jsonp

- JSONP (JSON with Padding)
- 요청 URL에 콜백함수 이름을 추가하여 서버로 요청을 보냄
- 서버는 JSON 데이터로 응답
- 서버는 callback(data)의 형태의 데이터 응답
- 응답을 받으면 클라이언트의 콜백함수가 처리
- JSON.stringify() 메소드
 - JavaScript 값이나 객체를 **JSON** 문자열로 변환